Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: Draft 2/**System do zarządzania zgłoszeniami**

Autorzy: **Jakub Ociepa, Joachim Welk**

Grupa: I1-220B

Kierunek: informatyka

Rok akademicki: 2022/2023

Poziom i semestr: I/4

Tryb studiów: stacjonarne

1. Spis treści

[Dokumentacja projektu zaliczeniowego 1](#_Toc134655767)

[Przedmiot: Inżynieria oprogramowania 1](#_Toc134655768)

[2 Odnośniki do innych źródeł 4](#_Toc134655769)

[3 Słownik pojęć 5](#_Toc134655770)

[4 Wprowadzenie 6](#_Toc134655771)

[4.1 Cel dokumentacji 6](#_Toc134655772)

[4.2 Przeznaczenie dokumentacji 6](#_Toc134655773)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc134655774)

[4.4 Analiza SWOT organizacji 6](#_Toc134655775)

[5 Specyfikacja wymagań 7](#_Toc134655776)

[5.1 Charakterystyka ogólna 7](#_Toc134655777)

[5.1.1 Definicja produktu 7](#_Toc134655778)

[5.1.2 Podstawowe założenia 7](#_Toc134655779)

[5.1.3 Cel biznesowy 7](#_Toc134655780)

[5.1.4 Użytkownicy 7](#_Toc134655781)

[5.1.5 Korzyści z systemu 7](#_Toc134655782)

[5.1.6 Ograniczenia projektowe i wdrożeniowe 8](#_Toc134655783)

[5.2 Wymagania funkcjonalne 8](#_Toc134655784)

[5.2.1 Lista wymagań 8](#_Toc134655785)

[5.2.2 Diagramy przypadków użycia 9](#_Toc134655786)

[5.2.3 Szczegółowy opis wymagań 9](#_Toc134655787)

[5.3 Wymagania niefunkcjonalne 11](#_Toc134655788)

[6 Zarządzanie projektem 13](#_Toc134655789)

[6.1 Zasoby ludzkie 13](#_Toc134655790)

[6.2 Harmonogram prac 13](#_Toc134655791)

[6.3 Etapy/kamienie milowe projektu 13](#_Toc134655792)

[7 Zarządzanie ryzykiem 14](#_Toc134655793)

[7.1 Lista czynników ryzyka 14](#_Toc134655794)

[7.2 Ocena ryzyka 14](#_Toc134655795)

[7.3 Plan reakcji na ryzyko 14](#_Toc134655796)

[8 Zarządzanie jakością 16](#_Toc134655797)

[8.1 Scenariusze i przypadki testowe 16](#_Toc134655798)

[9 Projekt techniczny 17](#_Toc134655799)

[9.1 Opis architektury systemu 17](#_Toc134655800)

[9.2 Technologie implementacji systemu 17](#_Toc134655801)

[9.3 Diagramy UML 17](#_Toc134655802)

[9.3.1 Diagram(-y) klas 17](#_Toc134655803)

[9.3.2 Diagram(-y) czynności 17](#_Toc134655804)

[9.3.3 Diagramy sekwencji 17](#_Toc134655805)

[9.3.4 Inne diagramy 17](#_Toc134655806)

[9.4 Charakterystyka zastosowanych wzorców projektowych 17](#_Toc134655807)

[9.5 Projekt bazy danych 17](#_Toc134655808)

[9.5.1 Schemat 17](#_Toc134655809)

[9.5.2 Projekty szczegółowe tabel 17](#_Toc134655810)

[9.6 Projekt interfejsu użytkownika 18](#_Toc134655811)

[9.6.1 Lista głównych elementów interfejsu 18](#_Toc134655812)

[9.6.2 Przejścia między głównymi elementami 18](#_Toc134655813)

[9.6.3 Projekty szczegółowe poszczególnych elementów 18](#_Toc134655814)

[9.7 Procedura wdrożenia 18](#_Toc134655815)

[10 Dokumentacja dla użytkownika 19](#_Toc134655816)

[11 Podsumowanie 20](#_Toc134655817)

[11.1 Szczegółowe nakłady projektowe członków zespołu 20](#_Toc134655818)

[12 Inne informacje 21](#_Toc134655819)

# Odnośniki do innych źródeł

* + [github](https://github.com/A4CH1M/IO)

# Słownik pojęć

1. System: Oprogramowanie, które ma na celu zapewniać mechanizm do rejestrowania i monitorowania zgłoszeń, a także umożliwienie klientom wygodnego ich składania.
2. Zgłoszenie: Informacja przekazana przez klienta dotycząca problemu, pytania lub żądania, które wymaga reakcji i rozwiązania.
3. Administrator systemu: Osoba odpowiedzialna za zarządzanie i utrzymanie systemu, w tym konfigurację, monitorowanie, rozwiązywanie problemów i udzielanie wsparcia użytkownikom.
4. Wymagania funkcjonalne: Precyzyjne opisy funkcji i możliwości, które system musi posiadać, aby spełnić oczekiwania użytkowników.
5. Wymagania niefunkcjonalne: Wymagania dotyczące jakości, wydajności, bezpieczeństwa, dostępności i innych aspektów systemu, które nie są bezpośrednio związane z jego funkcjonalnościami.
6. Interfejs użytkownika: Elementy, które umożliwiają interakcję między użytkownikiem a systemem, takie jak okna, strony, formularze itp.
7. Baza danych: Zbiór danych przechowywanych i zarządzanych w strukturalny sposób, służący do przechowywania informacji systemowych, takich jak zgłoszenia, informacje o klientach itp.
8. Harmonogram: Planowanie i ustalanie kolejności działań oraz terminów, które mają być wykonane w trakcie projektu.
9. Szkolenie: Proces przekazywania wiedzy i umiejętności użytkownikom systemu, aby mogli skutecznie korzystać z jego funkcjonalności.
10. Wdrożenie: Proces wprowadzenia systemu do działania w środowisku produkcyjnym i udostępnienie go użytkownikom.
11. Serwis: Działania utrzymania i konserwacji systemu po jego wdrożeniu, obejmujące wsparcie techniczne, rozwiązywanie problemów, dostarczanie aktualizacji itp.

# Wprowadzenie

## Cel dokumentacji

Dokumentacja ma na celu ułatwienie procesu wytworzenia i wdrożenia systemu. Zawiera ona informacje niezbędne do zrozumienia założeń projektowych.

## Przeznaczenie dokumentacji

- kierownik projektu

- zespół programistów

- zespół architektów inżynierów oprogramowania

## Opis organizacji lub analiza rynku

Organizacja będzie wydawcą i dystrybutorem oprogramowania użytkowego, odpowiedzialnym za jego utrzymanie, poprawną eksploatację i naprawę błędów.

## Analiza SWOT organizacji

|  |  |  |
| --- | --- | --- |
|  | pozytywne | negatywne |
| wewnętrzne | - młody, zdolny zespół programistów  - wykorzystanie najnowszych technologi i narzędzi | - ograniczone środki finansowe  - brak doświadczenia na rynku |
| zewnętrzne | - raptowna informatyzacja rynku  - potrzeba nadzoru informatycznych zasobów przedsiębiorstw | - silna konkurencja, posiadająca lojalną bazę klientów i dobrą reputację  - szybkie zmiany technologiczne |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

System do zarządzania zgłoszeniami usterek oprogramowania dystrybułowanymi przez organizację.

### Podstawowe założenia

System ten ma na celu umożliwienie klientom wygodnego składania zgłoszeń, a także zapewniać mechanizm do ich rejestrowania i monitorowania. Będzie on umożliwiał skuteczniejsze zarządzanie zgłoszeniami, dzięki czemu będzie można lepiej zrozumieć potrzeby klientów oraz szybciej i sprawniej na nie reagować. Dzięki temu firma będzie mogła zwiększyć swoją efektywność i poprawić jakość obsługi klienta.

### Cel biznesowy

System pozwoli docelowemu użytkownikowi na usystematyzowanie, zarządzanie i ewidencjonowanie zgłoszeń klientów organizacji. Gruntownie ułatwi pracownikom wykonywanie powierzonych obowiązków, zwiększając efektywność. Dzięki temu przedsiębiorstwo poprawi jakość świadczonych usług.

### Użytkownicy

1. Klienci firmy
2. Pracownicy firmy
3. Kierownictwo firmy
4. Administratorzy systemu

### Korzyści z systemu

1. Klienci firmy - możliwość złożenia zgłoszenia i monitorowania jego postępu
2. Pracownicy firmy - możliwość płynniejszego przyjmowania, przetwarzania i realizacji zgłoszeń klientów. Dostęp do ujednoliconej i uporządkowanej bazy zgłoszeń umożliwiający zwiększenie produktywności i komfortu pracy.
3. Kierownictwo firmy - wygodny dostęp do zapotrzebowań, roszczeń klientów oraz nadzór nad pracownikami i jakością obsługi klienta.
4. Administratorzy systemu - zapewnienie centralnego miejsca, w którym będą mogli zarządzać zgłoszeniami oraz dostępem użytkowników do systemu, co pozwoli na szybkie wykrycie i reagowanie na ewentualne zagrożenia.

### Ograniczenia projektowe i wdrożeniowe

1. Przepisy prawne: System musi być zgodny z obowiązującymi przepisami prawnymi, takimi jak RODO (Ogólne Rozporządzenie o Ochronie Danych Osobowych) lub inne regulacje dotyczące prywatności i ochrony danych. Konieczne jest przestrzeganie tych przepisów w celu zapewnienia bezpiecznego i legalnego przetwarzania danych osobowych.
2. Specyficzne technologie: Istnieją ograniczenia dotyczące wyboru konkretnych technologii, narzędzi i frameworków do budowy systemu. Może to wynikać z wymagań klienta, zaleceń ekspertów branżowych, dostępności zasobów lub innych czynników. Na przykład, klient może wymagać użycia konkretnej technologii programistycznej lub narzędzia do zarządzania projektami.
3. Bazy danych: System musi współpracować z określoną bazą danych lub posiadać ograniczenia dotyczące typu i rozmiaru bazy danych. Może to wynikać z istniejącej infrastruktury klienta, zaleceń bezpieczeństwa lub innych czynników.
4. Protokoły komunikacyjne: System musi obsługiwać określone protokoły komunikacyjne, które są wymagane do integracji z innymi systemami lub urządzeniami. Na przykład, może być konieczne obsługiwane protokołu HTTP, SOAP, REST itp.
5. Aspekty zabezpieczeń: System musi spełniać określone wymagania dotyczące zabezpieczeń, takie jak uwierzytelnianie, autoryzacja, szyfrowanie danych, zarządzanie sesjami użytkowników. Wymagania te mogą wynikać z norm branżowych, przepisów prawa lub oczekiwań klienta.
6. Zgodność ze standardami: System musi być zgodny z określonymi standardami, takimi jak standardy dotyczące interfejsów programistycznych (API), formatów danych, zarządzania projektami, jakości oprogramowania itp. Zgodność ta może być konieczna dla integracji z innymi systemami lub wymagań klienta.
7. Powiązania z innymi aplikacjami: System może wymagać integracji z istniejącymi aplikacjami lub systemami w organizacji. Wymagania te mogą dotyczyć interfejsów integracyjnych, formatów danych, protokołów komunikacyjnych itp.
8. Platforma sprzętowa i system operacyjny: System musi działać na określonej platformie sprzętowej (np. serwerach, urządzeniach mobilnych) oraz systemie operacyjnym (np. Windows, Linux, iOS, Android). Ograniczenia te mogą wynikać z infrastruktury klienta, preferencji technologicznych lub innych czynników.
9. Inne komponenty niezbędne do współpracy: System może wymagać zainstalowania i konfiguracji innych komponentów, takich jak serwery baz danych, serwery aplikacji, serwery poczty, usługi sieciowe itp. Konieczne jest spełnienie wymagań tych komponentów i ich współpraca z systemem.
10. Uzasadnienie tych ograniczeń może wynikać z wymagań klienta, obowiązujących przepisów prawnych, dostępności zasobów technologicznych, wymagań bezpieczeństwa, zgodności z istniejącą infrastrukturą itp.

## Wymagania funkcjonalne

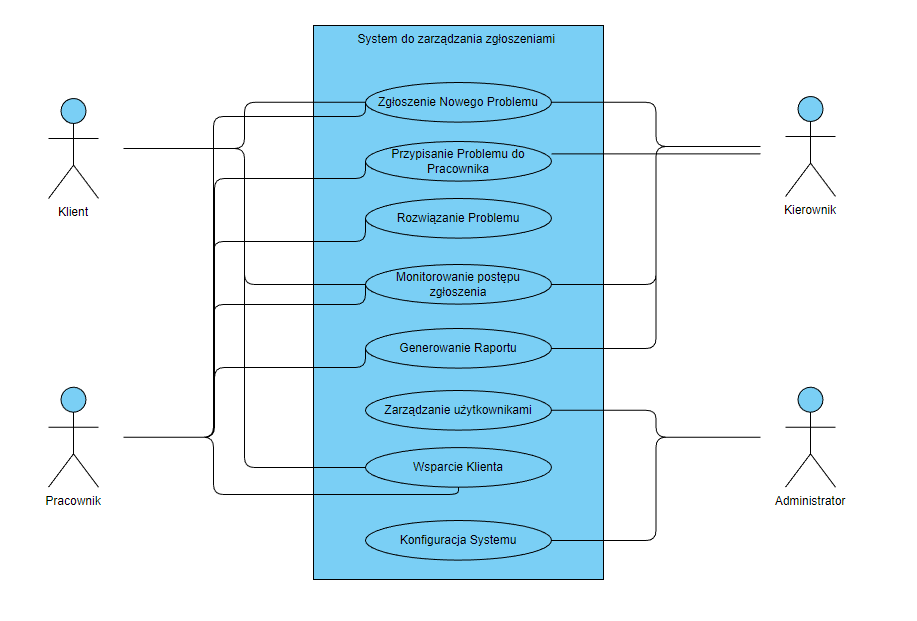
### Lista wymagań

1. Zgłoszenie nowego problemu - użytkownik ma możliwość zgłoszenia nowego problemu, w którym zawarte są szczegóły dotyczące problemu oraz informacje kontaktowe.
2. Przypisanie problemu do pracownika - pracownik ma możliwość przypisania zgłoszonego problemu do siebie, co oznacza, że to on jest odpowiedzialny za jego rozwiązanie.
3. Zamknięcie zgłoszenia - pracownik ma możliwość wprowadzenia rozwiązania dla zgłoszonego problemu, co pozwala na zakończenie zgłoszenia.
4. Monitorowanie postępu zgłoszenia - użytkownik ma możliwość śledzenia postępu zgłoszenia, w tym statusu zgłoszenia, przypisanego pracownika, wprowadzonego rozwiązania oraz daty zamknięcia zgłoszenia.
5. Generowanie raportów - pracownicy mają dostęp do narzędzi umożliwiających generowanie raportów dotyczących zgłoszeń, w tym raportów dotyczących ilości zgłoszeń, czasu potrzebnego na rozwiązanie zgłoszeń, przypisanych pracowników i innych kluczowych wskaźników.
6. Zarządzanie użytkownikami - administratorzy systemu mają możliwość zarządzania użytkownikami, w tym tworzenia nowych kont, resetowania haseł, zmiany uprawnień i innych zadań związanych z zarządzaniem użytkownikami.
7. Wsparcie klienta - użytkownicy mają możliwość skorzystania z funkcji wsparcia klienta, w tym kontaktu z pracownikami firmy w celu uzyskania pomocy w zgłaszaniu problemów i rozwiązywaniu ich.
8. Konfiguracja systemu - administratorzy systemu mają możliwość konfiguracji systemu, w tym dostosowania go do potrzeb firmy, wprowadzania zmian w formularzach zgłoszeń, definiowania reguł biznesowych i innych zadań związanych z konfiguracją systemu.

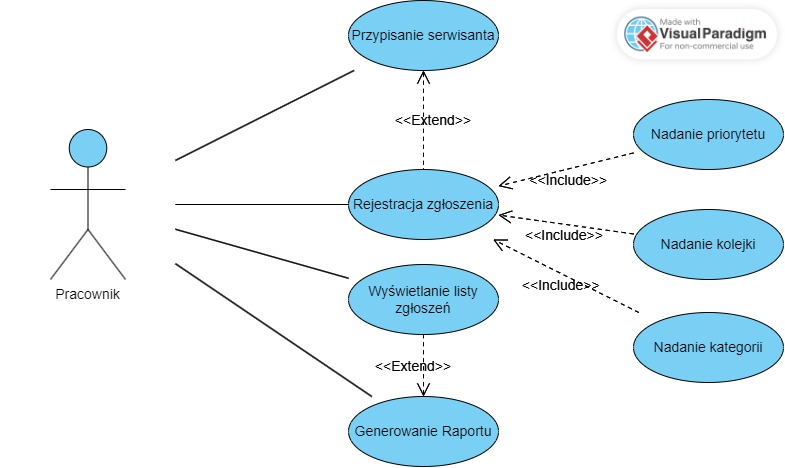
### Diagramy przypadków użycia

Tutaj same diagramy – bez specyfikacji, ale każdy diagram z tytułem i na osobnej stronie

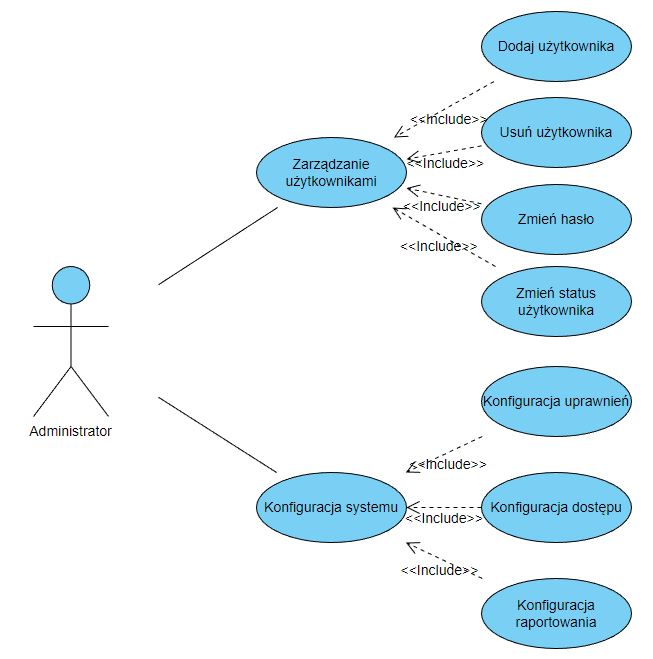
**Całościowy diagram przypadków użycia dla systemu zarządzania zgłoszeniami.**



**Diagram przypadku użycia dla pracownika**



**Diagram przypadku użycia dla administratora**



### Szczegółowy opis wymagań

* Numer: 1
* Nazwa: Zgłoszenie nowego problemu
* Uzasadnienie biznesowe:
  + System ma za zadanie zarządzać zgłoszeniami w zorganizowany sposób.
* Użytkownicy:
  + Klienci
  + Pracownicy
* Scenariusze:

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa scenariusza | Zgłoszenie samodzielne w systemie | Zgłoszenie poprzez serwisanta drogą mailową | Zgłoszenie poprzez serwisanta drogą telefoniczną |
| Warunki początkowe | Przedstawiciel klienta musi być zarejestrowany w bazie danych oraz być upoważniony do zakładania zgłoszeń serwisowych | Przedstawiciel klienta musi być zarejestrowany w bazie danych oraz być upoważniony do zakładania zgłoszeń serwisowych | Przedstawiciel klienta musi być zarejestrowany w bazie danych oraz być upoważniony do zakładania zgłoszeń serwisowych |
| Przebieg działań | -Klient loguje się do systemu  -Zakłada zgłoszenie  -Zgłoszenie trafia do bazy ze statusem “do przyjęcia”  -Po weryfikacji przez serwisanta zgłoszenie może zostać zatwierdzone i przekazane dalej | -Klient wysyła maila z wyczerpującym opisem problemu  -Serwisant ewidencjonuje zgłoszenie w bazie i informuje klienta  -Zgłoszenie czeka na przekazanie dalej | -Klient kontaktuje się z serwisantem i wyczerpująco opisuje problem  -Serwisant rejestruje zgłoszenie w bazie i rozłącza się  -Zgłoszenie czeka na przekazanie dalej |
| Efekty | Zgłoszenie może zostać przekazane do realizacji do odpowiedniego serwisanta | Zgłoszenie może zostać przekazane do realizacji do odpowiedniego serwisanta | Zgłoszenie może zostać przekazane do realizacji do odpowiedniego serwisanta |
| Wymagania niefunkcjonalne | Całodobowy dostęp do systemu,  bezpieczeństwo danych | Zgłoszenia przyjmowane w czasie pracy serwisu | Zgłoszenia przyjmowane w czasie pracy serwisu |
| Częstotliwość | 3 | 4 | 4 |
| Istotność | 4 | 5 | 5 |

* Numer: 2
* Nazwa: Przypisanie problemu do pracownika
* Uzasadnienie biznesowe:
  + Przypisanie konkretnego serwisanta do zgłoszenia umożliwia zachowanie spójności i przejrzystości w systemie oraz pomaga długoterminowo pozwalając na bardziej efektywne rozwiązywanie podobnych problemów w przyszłości.
* Użytkownicy:
  + Pracownicy
  + Kierownicy
* Scenariusze:

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa scenariusza | Samodzielne przypisanie sobie zgłoszenia | Przypisanie zadania przez kierownika | Przypisanie zadania przez pracownika odpowiedzialnego za serwis |
| Warunki początkowe | Zgłoszenie może istnieć lub być w procesie rejestracji | Zgłoszenie musi istnieć | Zgłoszenie może istnieć lub być w procesie rejestracji |
| Przebieg działań | -Pracownik rejestruje nowe zgłoszenie lub znajduje zadanie oczekujące na realizację  -Pracownik przypisuje sobie zadanie do wykonania  -Pracownik przystępuje do realizacji zgłoszenia | -Kierownik wyświetla listę zgłoszeń  -Kierownik wybiera zgłoszenie, które ma zostać przypisane.  -Kierownik wyświetla listę pracowników dostępnych do przypisania zadania.  -Kierownik wybiera odpowiedniego pracownika z listy.  -Kierownik przypisuje wybrane zadanie do wybranego pracownika.  -Pracownik otrzymuje powiadomienie o przypisanym zadaniu.  -Pracownik rozpoczyna pracę nad problemem. | -Pracownik odpowiedzialny za serwis ma zgłoszenie, które wymaga rozwiązania, a nie jest do nikogo przypisane i nie jest stanie zrealizować go samodzielnie.  -Pracownik przegląda listę współpracowników i identyfikuje osobę, która jest dostępna i może pomóc w realizacji zadania.  -Pracownik kontaktuje się z współpracownikiem i potwierdza, że może przypisać mu nowe zadanie.  -Pracownik przypisuje zadanie współpracownikowi poprzez system.  -Współpracownik otrzymuje powiadomienie o przypisanym zadaniu.  -Współpracownik rozpoczyna pracę nad problemem. |
| Efekty | -Zgłoszenie jest przypisane do pracownika  -Klient widzi postęp w sprawie | -Zgłoszenie jest przypisane do pracownika  -Klient widzi postęp w sprawie | -Zgłoszenie jest przypisane do pracownika  -Klient widzi postęp w sprawie |
| Wymagania niefunkcjonalne | - Kontrolowane przez odpowiednie uprawnienia  - Rejestracja infor-macji dotyczących przypisania zgłoszenia | -Minimalne opóźnienia  i responsywność  - Kontrolowane przez odpowiednie uprawnienia  - Rejestracja informacji dotyczących przypisania zgłoszenia | -Kontrolowane przez odpowiednie uprawnienia  - Rejestracja informacji dotyczących przypisania zgłoszenia |
| Częstotliwość | 3 | 2 | 5 |
| Istotność | 5 | 5 | 5 |

* Numer: 3
* Nazwa: Monitorowanie postępu zgłoszenia
* Uzasadnienie biznesowe:
  + System musi pozwalać na wgląd do postępu zgłoszenia zarówno przez klienta, jak i pracownika, by zwiększyć efektywność pracy i skuteczność wdrożonego rozwiązania.
* Użytkownicy:
  + Klienci
  + Pracownicy
  + Kierownicy
  + Administratorzy
* Scenariusze:

|  |  |  |
| --- | --- | --- |
| Nazwa scenariusza | Samodzielne monitorowanie przez klienta | Monitorowanie postępów zgłoszenia przez pracowników |
| Warunki początkowe | Zgłoszenie musi istnieć | Zgłoszenie musi istnieć |
| Przebieg działań | -Klient loguje się do systemu za pomocą swoich danych uwierzytelniających.  -Klient wyszukuje zgłoszenie, które chce monitorować.  -System wyświetla informacje o stanie zgłoszenia, takie jak aktualny status, daty ważne dla zgłoszenia, przypisanych pracowników, dodane komentarze itp. | -Pracownik otwiera panel monitorowania zgłoszeń.  -System wyświetla listę zgłoszeń wraz z informacjami o ich stanie, czasie trwania, przypisanych pracownikach itp.  -Pracownik może filtrować i sortować zgłoszenia, aby skoncentrować się na tych o największym priorytecie lub najdłuższym czasie oczekiwania.  -Pracownik może dodawać komentarze, aktualizować statusy zgłoszeń lub przypisywać nowe zadania pracownikom w celu monitorowania i zarządzania postępem zgłoszeń. |
| Efekty | Klient wie, jakie postępy zostały poczynione w jego zgłoszeniu | Pracownicy mogą znaleźć dostępne zgłoszenia lub zobaczyć postęp “odziedziczonego” zgłoszenia  Kierownicy mogą zobaczyć jakie postępy w pracy robią ich podwładni |
| Wymagania niefunkcjonalne | - dostępny dla klientów przez całą dobę, 7 dni w tygodniu  -Dane klienta powinny być zabezpieczone np.: szyfrowaniem i autoryzacją | - Zabezpieczenia takie jak audyt logów i uwierzytelnianie dwuskładnikowe  - Panel kontrolny powinien działać sprawnie, umożliwiając administratorom szybkie wykonywanie operacji |
| Częstotliwość | 3 | 3 |
| Istotność | 3 | 3 |

* Numer: 4
* Nazwa: Generowanie raportów
* Uzasadnienie biznesowe:
  + Kontrola i zarządzanie efektywnością procesu.
* Użytkownicy:
  + Klienci
  + Pracownicy
  + Administratorzy
* Scenariusze:

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa scenariusza | Generowanie raportu dla konkretnego klienta | Automatyczne generowanie raportów okresowych | Indywidualne generowanie raportów |
| Warunki początkowe | -Klient  musi być zarejestrowany i posiadać ważne konto w systemie.  -Klient musi posiadać zgłoszenia | - Administrator lub upoważniony użytkownik musi skonfigurować harmonogram generowania raportów okresowych, określając częstotliwość, datę, godzinę i inne parametry generacji. | -Użytkownik musi mieć uprawnienia do generowania raportów.  - Użytkownik musi wybrać odpowiednie parametry generacji, takie jak zakres dat, kryteria filtrowania, grupowanie danych itp. |
| Przebieg działań | -Pracownik wybiera opcję generowania raportu.  -System wyświetla formularz do wprowadzenia parametrów raportu, takich jak wybrany klient, okres czasu, rodzaj danych itp.  -Pracownik wprowadza odpowiednie parametry i inicjuje generowanie raportu.  -System przetwarza żądanie  i generuje raport zawierający żądane dane klienta.  -System wyświetla raport w formie interaktywnej lub udostępnia go do pobrania w odpowiednim formacie (np. PDF, XLS). | -System zaprogramowany jest do automatycznego generowania raportów okresowych (np. co tydzień, miesiąc).  -System zbiera niezbędne dane z bazy danych lub innych źródeł informacji.  -Na podstawie zaprogramowanych reguł i ustawień, system generuje raporty z wybranymi danymi i formatem.  -Wygenerowane raporty są automatycznie zapisywane lub wysyłane do określonych odbiorców (np. pracowników, kierownictwa) w wyznaczonym czasie. | -Pracownik wybiera opcję generowania raportów.  -System udostępnia różne rodzaje raportów do wyboru (np. statystyki klientów, wykorzystanie zasobów, efektywność pracy).  -Pracownik wybiera odpowiedni raport i konfiguruje parametry, takie jak zakres dat, filtrowanie danych itp.  -System generuje raport zgodnie z wybranymi parametrami  i prezentuje go pracownikowi w interfejsie aplikacji.  -Pracownik może eksportować raport do różnych formatów lub udostępnić go innym zainteresowanym osobom. |
| Efekty | Raport jest wygenerowany | Raport jest wygenerowany | Raport jest wygenerowany |
| Wymagania niefunkcjonalne | - możliwość indywidualnego generowania raportów na żądanie, umożliwiając wybór określonych parametrów, filtrów, grupowania i innych kryteriów  -intuicyjność i łatwość obsługi | - możliwość konfiguracji harmonogramu generowania, umożliwiając ustalenie częstotliwości, daty lub godziny  -intuicyjność i łatwość obsługi | - możliwość indywidualnego generowania raportów na żądanie, umożliwiając wybór określonych parametrów, filtrów, grupowania i innych kryteriów  -intuicyjność  i łatwość obsługi |
| Częstotliwość | 3 | 4 | 5 |
| Istotność | 3 | 5 | 4 |

* Numer: 5
* Nazwa: Zarządzanie użytkownikami
* Uzasadnienie biznesowe:
  + Zapewnienia bezpiecznego i uporządkowanego dostępu do systemu, umożliwiając precyzyjne kontrolowanie uprawnień i zabezpieczeń, co przekłada się na poprawę efektywności i ochronę danych firmy oraz klientów.
* Użytkownicy:
  + Administratorzy
* Scenariusze:

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa scenariusza | Rejestracja nowego użytkownika | Zarządzanie uprawnieniami użytkowników | Resetowanie hasła użytkownika |
| Warunki początkowe | -Formularz rejestracyjny powinien być dostępny przez stronę internetową lub aplikację mobilną.  - Nowy użytkownik musi podać unikalny identyfikator, który nie jest już używany przez innego użytkownika. | - Tylko użytkownicy posiadający uprawnienia administratora powinni mieć dostęp do panelu administracyjnego.  - Uprawnienia użytkowników muszą być przypisane i zarządzane przez administratora. | - Użytkownik musi zgłosić prośbę o resetowanie hasła poprzez odpowiedni formularz lub funkcję w systemie.  -System powinien dostarczyć użytkownikowi odpowiednie narzędzia lub procedury do zresetowania hasła. |
| Przebieg działań | -Nowy użytkownik odwiedza stronę rejestracyjną systemu.  -Użytkownik wypełnia formularz rejestracyjny, wprowadzając swoje dane osobowe, adres e-mail i wybierając unikalną nazwę użytkownika oraz hasło.  -System weryfikuje wprowadzone dane i sprawdza, czy wybrana nazwa użytkownika jest dostępna.  -Jeśli dane są poprawne, administrator rejestruje nowego użytkownika  i przypisuje mu odpowiednie uprawnienia i rolę.  -Użytkownik otrzymuje potwierdzenie rejestracji na podany adres e-mail lub inny sposób komunikacji. | -Administrator wybiera opcję zarządzania użytkownikami lub uprawnieniami.  -System wyświetla listę użytkowników wraz z ich rolami  i uprawnieniami.  -Administrator może dodawać, usuwać lub edytować użytkowników oraz przypisywać im odpowiednie role  i uprawnienia.  -Zmiany w uprawnieniach użytkowników są zapisywane i mają wpływ na ich dostęp do poszczególnych funkcji i danych w systemie. | -Użytkownik zgłasza zapomniane hasło lub prosi o resetowanie hasła.  -System sprawdza tożsamość użytkownika, np. poprzez adres e-mail, pytania bezpieczeństwa itp.  -Jeśli użytkownik zostaje potwierdzony, system generuje tymczasowe hasło lub link do resetowania hasła.  -Użytkownik otrzymuje tymczasowe hasło lub link do resetowania hasła drogą elektroniczną lub innym sposobem komunikacji.  -Użytkownik loguje się za pomocą tymczasowego hasła lub korzysta z linku do resetowania hasła i wprowadza nowe hasło.  -System zapisuje zmiany i potwierdza użytkownikowi zakończenie procesu resetowania hasła. |
| Efekty | -Nowy użytkownik powinien otrzymać potwierdzenie rejestracji  -Nowy użytkownik powinien mieć pełny dostęp do funkcjonalności systemu zgodnie  z przypisanymi mu uprawnieniami | -Zmiany w uprawnieniach użytkowników powinny być skutecznie zastosowane w systemie | - Użytkownik ma możliwość zresetowania swojego hasła |
| Wymagania niefunkcjonalne | -zabezpieczenia takie jak weryfikacja adresu e-mail i potwierdzenia tożsamości  - szybki i efektywny, minimalizując czas oczekiwania użytkownika na potwierdzenie rejestracji i umożliwiając szybkie rozpoczęcie korzystania z systemu | -Odpowiednie mechanizmy uwierzytelniania i autoryzacji  -Umożliwienie konfiguracji i dostosowania różnych poziomów uprawnień oraz ich hierarchii | -zabezpieczenia takie jak weryfikacja adresu e-mail  i potwierdzenia tożsamości  -szybki i efektywny, minimalizując czas oczekiwania użytkownika na potwierdzenie rejestracji |
| Częstotliwość | 3 | 5 | 1 |
| Istotność | 4 | 5 | 3 |

## Wymagania niefunkcjonalne

1. Bezpieczeństwo:

* Wymagana jest autoryzacja użytkowników i zapewnienie poufności danych.
* Wszelka wymiana danych między użytkownikami a systemem musi odbywać się w sposób szyfrowany.
* System powinien być odporny na ataki zewnętrzne.

1. Wydajność:

* System powinien obsługiwać co najmniej 1000 użytkowników jednocześnie.
* Czas odpowiedzi na żądania użytkowników nie powinien przekraczać 2 sekund.

1. Skalowalność:

* System powinien umożliwiać łatwe dodawanie nowych funkcjonalności i integracji z innymi systemami.

1. Dostępność:

* System powinien być dostępny 24/7, z minimalnym czasem przestoju na potrzeby konserwacji.
* W przypadku awarii system powinien automatycznie przechodzić na tryb awaryjny.

1. Użyteczność:

* Interfejs użytkownika powinien być intuicyjny i łatwy w obsłudze.
* System powinien umożliwiać dostosowywanie ustawień do potrzeb użytkownika.
* System powinien być zgodny z zasadami dostępności.

1. Wymagania techniczne:

* System powinien korzystać z szybkiej i niezawodnej bazy danych, która umożliwi skuteczne zarządzanie dużą ilością danych.
* System powinien być modułowy, aby ułatwić rozwój i utrzymanie.
* System powinien korzystać z najnowszych wersji języków programowania oraz frameworków.

1. Wymagania prawne i regulacyjne:

* System powinien być zgodny z obowiązującymi przepisami prawa, w tym z RODO.

1. Wymagania dotyczące dokumentacji:

* System powinien posiadać dokumentację techniczną i użytkownika.
* Wszelkie zmiany w systemie powinny być opisane w dokumentacji.

# Zarządzanie projektem

## Zasoby ludzkie

Projekt Manager: Osoba odpowiedzialna za zarządzanie całością projektu. Koordynuje działania zespołu, nadzoruje postęp prac, monitoruje budżet i terminy. Projekt Manager jest kluczowym punktem kontaktu dla klienta i zarządza zasobami ludzkimi w projekcie.

Analityk biznesowy: Osoba odpowiedzialna za analizę wymagań klienta i przekształcenie ich w szczegółowe specyfikacje. Współpracuje z klientem i zespołem programistycznym w celu zapewnienia zrozumienia i spełnienia oczekiwań klienta.

Projektant UI/UX: Specjalista od projektowania interfejsu użytkownika (UI) i doświadczenia użytkownika (UX). Projektant tworzy intuicyjne i atrakcyjne interfejsy, które są łatwe w obsłudze i zapewniają pozytywne doświadczenia użytkownikom.

Programiści: Zespół programistów, którzy zajmują się implementacją oprogramowania. Mogą to być programiści back-end, zajmujący się logiką biznesową i bazami danych, oraz programiści front-end, odpowiedzialni za tworzenie interfejsu użytkownika.

Testerzy: Osoby odpowiedzialne za przeprowadzanie testów oprogramowania w celu sprawdzenia poprawności działania, wykrycia błędów i zapewnienia jakości. Testerzy mogą być odpowiedzialni zarówno za testy funkcjonalne, jak i testy wydajności, bezpieczeństwa itp.

Konsultanci lub eksperci branżowi: W zależności od specyfiki projektu, może być potrzebna wiedza lub doświadczenie ekspertów z określonej dziedziny lub branży. Tacy eksperci mogą doradzać w zakresie funkcjonalności, wymagań czy zgodności z regulacjami branżowymi.

## Harmonogram prac

Etap 1: Analiza wymagań

Zadanie 1: Zebranie i analiza wymagań systemowych – 2 tyg.

Zadanie 2: Sporządzenie dokumentu z wymaganiami funkcjonalnymi i niefunkcjonalnymi – 3 tyg.

Zadanie 3: Przygotowanie diagramów przypadków użycia - 2 tyg.

Etap 2: Projektowanie architektury systemu

Zadanie 4: Opracowanie architektury systemu - 2 tyg.

Zadanie 5: Utworzenie diagramów komponentów i diagramów sekwencji - 3 tyg.

Zadanie 6: Wybór odpowiednich technologii i narzędzi - 1 tydz.

Etap 3: Implementacja systemu

Zadanie 7: Stworzenie interfejsu użytkownika - 4 tyg.

Zadanie 8: Implementacja logiki biznesowej – 8 tyg.

Zadanie 9: Integracja z bazą danych – 2 tyg.

Zadanie 10: Testowanie i debugowanie – 2 tyg.

Etap 4: Testowanie systemu

Zadanie 11: Przeprowadzenie testów jednostkowych – 2 tyg.

Zadanie 12: Testowanie funkcjonalności systemu – 3 tyg.

Zadanie 13: Wykrywanie i usuwanie błędów - 2 tyg.

Zadanie 14: Optymalizacja systemu – 4 tyg.

Etap 5: Wdrożenie i szkolenie

Zadanie 15: Konfiguracja i instalacja systemu na docelowej infrastrukturze – 2 tyg.

Zadanie 16: Przygotowanie dokumentacji użytkownika - 2 tyg.

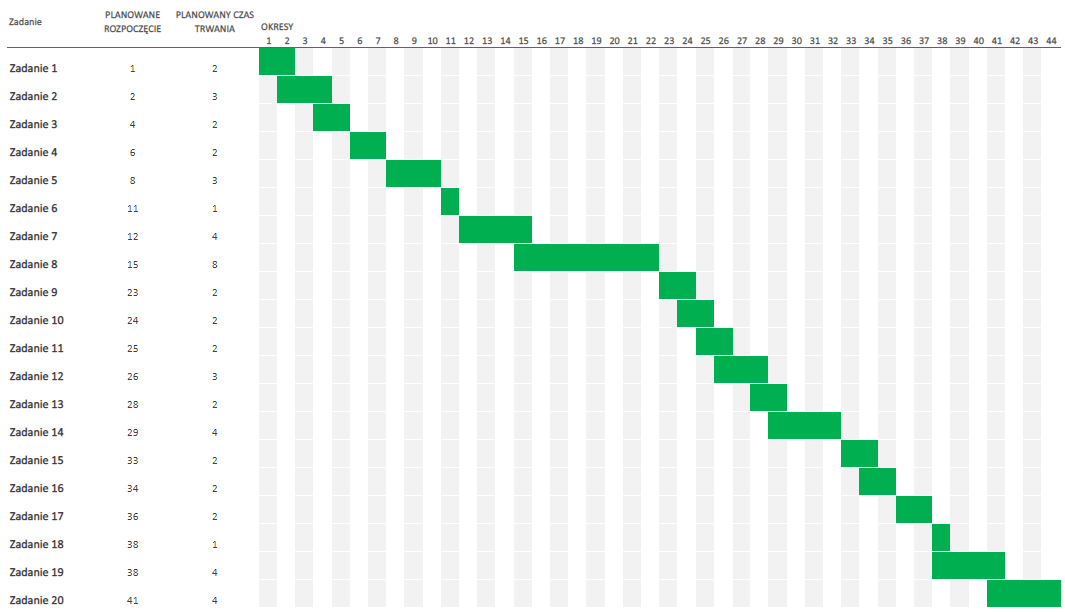
Zadanie 17: Szkolenie personelu w zakresie obsługi systemu – 2 tyg.

Etap 6: Rozruch i monitorowanie

Zadanie 18: Uruchomienie systemu w środowisku produkcyjnym – 1 tydz.

Zadanie 19: Monitorowanie działania systemu i reagowanie na ewentualne problemy – 4 tyg.

Zadanie 20: Wydanie ewentualnych poprawek i aktualizacji



## Etapy/kamienie milowe projektu

1. Analiza wymagań

* Zakończenie zbierania wymagań systemowych
* Ukończenie analizy i dokumentacji wymagań
* Zatwierdzenie dokumentu z wymaganiami przez klienta

1. Projektowanie architektury systemu

* Zakończenie opracowywania architektury systemu
* Ukończenie diagramów komponentów i diagramów sekwencji
* Przygotowanie dokumentacji architektonicznej

1. Implementacja systemu

* Ukończenie interfejsu użytkownika
* Implementacja głównej logiki biznesowej
* Integracja z bazą danych
* Pierwsza wersja działającego systemu

1. Testowanie systemu

* Przeprowadzenie testów jednostkowych
* Testowanie funkcjonalności systemu
* Wykrycie i poprawienie większości błędów

1. Wdrożenie i szkolenie

* Skonfigurowanie i zainstalowanie systemu w środowisku produkcyjnym
* Ukończenie dokumentacji użytkownika
* Szkolenie personelu w zakresie obsługi systemu

1. Rozruch i monitorowanie

* Uruchomienie systemu w pełni produkcyjnym środowisku
* Monitorowanie działania systemu i reakcja na ewentualne problemy
* Wprowadzenie ewentualnych poprawek i aktualizacji

# Zarządzanie ryzykiem

## Lista czynników ryzyka

* zpóźnienia w harmonogramie projektu, co może prowadzić do przekroczenia budżetu i opóźnienia w wdrożeniu systemu.
* Problemy techniczne, takie jak problemy z integracją systemu z istniejącymi systemami w firmie, co może prowadzić do problemów z funkcjonalnością.
* Brak zaangażowania zespołu projektowego, co może prowadzić do niskiej jakości produktu.
* Brak akceptacji klientów, co może prowadzić do niskiej adopcji systemu i niskiej skuteczności w zarządzaniu zgłoszeniami.
* Zagrożenie cyberbezpieczeństwa, takie jak ataki hakerskie lub wycieki danych, co może prowadzić do utraty danych klientów i wizerunku firmy.
* Niedostateczne szkolenie personelu, co może prowadzić do błędów w obsłudze systemu i zmniejszenia jego skuteczności.
* Problemy z dostawcami, takie jak opóźnienia w dostawie sprzętu lub oprogramowania, co może prowadzić do opóźnień w harmonogramie projektu i przekroczenia budżetu.

## Ocena ryzyka

* Ryzyko techniczne:
  + Prawdopodobieństwo: Średnie
  + Wpływ: Wysoki
* Ryzyko czasowe:
  + Prawdopodobieństwo: Wysokie
  + Wpływ: Średni
* Ryzyko finansowe:
  + Prawdopodobieństwo: Niskie
  + Wpływ: Wysoki
* Ryzyko zmiany wymagań:
  + Prawdopodobieństwo: Wysokie
  + Wpływ: Średni
* Ryzyko ludzkie:
  + Prawdopodobieństwo: Średnie
  + Wpływ: Średni

## Plan reakcji na ryzyko

* Ryzyko techniczne:
  + Przeprowadzenie testów jednostkowych, integracyjnych i akceptacyjnych w celu zminimalizowania ryzyka wystąpienia błędów w kodzie.
  + Zatrudnienie doświadczonych programistów, którzy posiadają odpowiednie umiejętności techniczne.
  + Wdrożenie procedur bezpieczeństwa i prywatności danych.
* Ryzyko czasowe:
  + Przygotowanie realistycznego harmonogramu projektu z uwzględnieniem dodatkowych okresów na przeglądanie i weryfikację postępów projektu.
  + Przeprowadzenie regularnych spotkań z zespołem projektowym, aby monitorować postęp projektu i rozwiązywać problemy na bieżąco.
* Ryzyko finansowe:
  + Przygotowanie realistycznego budżetu projektu, który uwzględnia koszty związane z zarządzaniem ryzykiem.
  + Wprowadzenie ścisłej kontroli kosztów i monitorowanie budżetu projektu na bieżąco.
* Ryzyko zmiany wymagań:
  + Przeprowadzenie szczegółowej analizy wymagań klienta przed rozpoczęciem projektu.
  + Utrzymywanie regularnego kontaktu z klientem i informowanie go o każdej zmianie w harmonogramie projektu.
  + Wprowadzenie procedur zmiany wymagań, które pozwolą na ich szybkie i skuteczne zarządzanie.
* Ryzyko ludzkie:
  + Zapewnienie odpowiedniego szkolenia i wsparcia dla zespołu projektowego.
  + Tworzenie dobrych warunków pracy dla zespołu, aby zapobiec wypaleniu zawodowemu.
  + Przygotowanie planu awaryjnego na wypadek rezygnacji któregoś z członków zespołu.

# Zarządzanie jakością

## Scenariusze i przypadki testowe

1. Rejestracja użytkownika

* Kategoria: funkcjonalność
* Opis: Testowanie procesu rejestracji nowego użytkownika
* Tester: Administrator
* Termin: Po stworzeniu modułu zarządzania użytkownikami
* Narzędzia wspomagające: Brak
* Przebieg działań:
  + Otwórz panel zarządzania użytkownikami
  + Wprowadź wymagane dane użytkownika i kliknij przycisk "Zarejestruj"
* Warunki wstępne:
* Dane wejściowe: Dane rejestracyjne: imię, nazwisko, adres e-mail, hasło
* Warunek zaliczenia testu: Nowy użytkownik zostaje poprawnie zarejestrowany i jego dane są zapisane w systemie
* Zestaw danych testowych: Brak

1. Dodanie nowego zgłoszenia

* Kategoria: funkcjonalność
* Opis: Testowanie procesu dodawania nowego zgłoszenia przez pracownika
* Tester: Pracownik (serwisant)
* Termin: Po stworzeniu modułu dodawania i zarządzania zgłoszeniami, panelu zarządzania użytkownikami i klientami
* Narzędzia wspomagające: Brak
* Przebieg działań:
  + Pracownik otwiera moduł zarządzania zgłoszeniami
  + Wybiera opcję dodania nowego zgłoszenia
  + Wprowadza wymagane informacje dotyczące zgłoszenia.
  + Klika “Zatwierdź”.
  + System weryfikuje poprawność danych.
  + System zapisuje zgłoszenie w bazie danych.
  + System generuje unikalny numer zgłoszenia oraz przypisuje status i przypisuje je do zgłoszenia.
  + System wysyła potwierdzenie otrzymania zgłoszenia do klienta.
* Warunki wstępne:
  + Ukończony i funkcjonujący moduł dodawania i zarządzania zgłoszeniami
  + Połączenie z bazą danych
  + Stworzony klient testowy
* Dane wejściowe:
  + Dane klienta: nazwa klienta, imię i nazwisko zgłaszającego
  + Opis zgłoszenia: tytuł, opis problemu
  + Informacje dodatkowe: kategoria, priorytet, kolejka, serwisant
* Warunek zaliczenia testu: Zgłoszenie zostaje poprawnie zapisane w systemie, przypisane pracownikowi i klient otrzymuje potwierdzenie rejestracji zgłoszenia.
* Zestaw danych testowych: Brak

1. Generowanie raportu

* Kategoria: funkcjonalność
* Opis: Testowanie procesu generowania raportu dla konkretnego klienta
* Tester: Administrator
* Termin: Po stworzeniu modułu raportów i panelu zarządzania użytkownikami
* Narzędzia wspomagające: Program do obsługi plików raportowych
* Przebieg działań:
  + Administrator otwiera modułu raportów
  + Administrator wybiera klienta, dla którego chce wygenerować raport
  + Administrator wybiera okres, za który chce wygenerować raport
  + Administrator inicjuje generowanie raportu
  + System generuje raport na podstawie wybranych parametrów
  + System prezentuje wygenerowany raport administratorowi
  + Administrator eksportuje raport do pliku wyjściowego
  + Administrator otwiera wyeksportowany plik i sprawdza jego poprawność
* Warunki wstępne:
  + Konto administratora istnieje
  + Istnieje profil klienta testowego
  + Klient testowy posiada testowe zgłoszenia
  + Administrator ma dostęp do panelu administracyjnego
  + Narzędzie wspomagające (Narzędzie X) jest zainstalowane i skonfigurowane
* Dane wejściowe:
  + Wybrany klient: identyfikator klienta.
  + Okres generowania raportu: data początkowa i data końcowa.
* Warunek zaliczenia testu: Wygenerowany raport zawiera poprawne dane dla wybranego klienta i okresu oraz wyeksportowany plik posiada poprawne dane
* Zestaw danych testowych: Brak

1. Logowanie/wylogowanie użytkownika

* Kategoria: funkcjonalność
* Opis: Testowanie procesu logowania użytkownika
* Tester: Użytkownik testowy
* Termin: Po stworzeniu modułu zarządzania użytkownikami
* Narzędzia wspomagające: Brak
* Przebieg działań:
  + Wprowadź niepoprawne dane logowania i kliknij przycisk "Zaloguj się"
  + Wprowadź poprawne dane logowania i kliknij przycisk "Zaloguj się"
  + Wyloguj się
  + Wprowadź wielokrotnie niepoprawne dane logowania i kliknij przycisk "Zaloguj się"
* Warunki wstępne: Zarejestrowany użytkownik w systemie
* Dane wejściowe: Dane logowania (identyfikator, hasło)
* Warunek zaliczenia testu:
  + Zarejestrowany użytkownik może poprawnie zalogować się do systemu za pomocą swojego adresu e-mail i hasła
  + Niepoprawne wprowadzenie danych logowania spowoduje błąd
  + Użytkownik poprawnie się wyloguje
  + Wprowadzenie niepoprawnych danych wystarczającą liczbę razy spowoduje zablokowanie konta.
* Zestaw danych testowych: Brak

1. Monitorowanie postępu zgłoszenia

* Kategoria: funkcjonalność
* Opis: Testowanie możliwości monitorowania postępu zgłoszenia przez klienta
* Tester: Klient testowy
* Termin: Po stworzeniu modułu zarządzania użytkownikami
* Narzędzia wspomagające: Brak
* Przebieg działań:
  + Klient loguje się do systemu.
  + Klient przechodzi do sekcji "Moje zgłoszenia".
  + Klient wybiera zgłoszenie, które chce monitorować.
  + System wyświetla klientowi informacje o postępie zgłoszenia.
  + Klient może śledzić aktualizacje dotyczące zgłoszenia.
* Warunki wstępne:
  + Klient testowy ma aktywne konto w systemie i posiada zgłoszenia.
  + Zgłoszenie ma przypisanego klienta i jest w trakcie przetwarzania.
* Dane wejściowe: Wybrane zgłoszenie: numer zgłoszenia.
* Warunek zaliczenia testu: Klient otrzymuje poprawne informacje dotyczące postępu zgłoszenia i może je monitorować w czasie rzeczywistym.
* Zestaw danych testowych: Brak

# Projekt techniczny

## Opis architektury systemu

z ew. rysunkami pomocniczymi

Architektura systemu oparta jest na modelu klient-serwer, w którym komunikacja odbywa się między klientami a serwerem. Poniżej przedstawiam opis architektury klient-serwer:

* Klient:
  + Klient to aplikacja, która jest uruchamiana na komputerze użytkownika.
  + Odpowiada za interakcję z użytkownikiem, prezentację interfejsu użytkownika oraz przetwarzanie danych wprowadzonych przez użytkownika.
  + Komunikuje się z serwerem, przesyłając żądania i odbierając odpowiedzi.
* Serwer
  + Serwer to centralna jednostka odpowiedzialna za przetwarzanie żądań klientów i udostępnianie im odpowiednich zasobów i usług.
  + Odpowiada za zarządzanie danymi, logiką biznesową oraz zabezpieczeniami systemu.
  + Przyjmuje żądania od klientów, przetwarza je i zwraca odpowiedzi.
* Protokół komunikacyjny:
  + Komunikacja między klientem a serwerem odbywa się za pomocą określonego protokołu komunikacyjnego, takiego jak HTTP, TCP/IP itp.
  + Protokół zapewnia struktury danych, formaty komunikacji oraz reguły przesyłania i odbierania danych.
* Warstwy architektury
  + Architektura klient-serwer może być rozbudowana o dodatkowe warstwy, takie jak warstwa prezentacji, warstwa logiki biznesowej i warstwa danych.
  + Warstwa prezentacji odpowiada za prezentację interfejsu użytkownika klienta.
  + Warstwa logiki biznesowej zawiera logikę przetwarzania danych i zarządzania biznesowymi procesami.
  + Warstwa danych zajmuje się zarządzaniem danymi, w tym przechowywaniem, odczytem i zapisem danych w bazie danych lub innym magazynie danych.

## Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

|  |  |
| --- | --- |
| Technologia | Uzasadnienie |
| C++ | C++ to język programowania ogólnego przeznaczenia, który oferuje wysoką wydajność i efektywne zarządzanie zasobami. Jest szeroko stosowany w tworzeniu aplikacji o dużej złożoności i wymagających wysokiej wydajności. |
| QT Framework | Qt to rozbudowany framework do tworzenia aplikacji w języku C++. Zapewnia narzędzia i biblioteki do budowy interfejsu użytkownika, zarządzania zdarzeniami, obsługi wielowątkowości, a także dostęp do funkcji systemowych i grafiki. |
| MYSQL | MySQL jest systemem zarządzania bazą danych. Zapewnia niezawodne przechowywanie danych i wysoką wydajność. |
| Git | Git jest rozproszonym systemem kontroli wersji. Umożliwia śledzenie zmian w kodzie, współpracę programistów i zarządzanie kodem źródłowym. |

## Diagramy UML

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie

diagramy przypadków użycia umieszczone w punkcie 5.2.2, a nie tutaj.

### Diagram(-y) klas

Diagram Klasy Pracowników

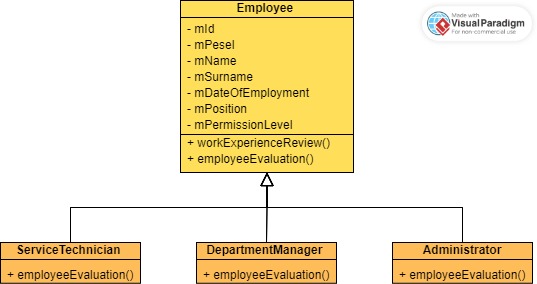
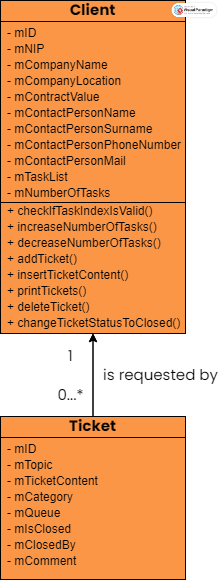


Diagram Klient/Zgłoszenie



### Diagram(-y) czynności

Diagram Założenia zgłoszenia

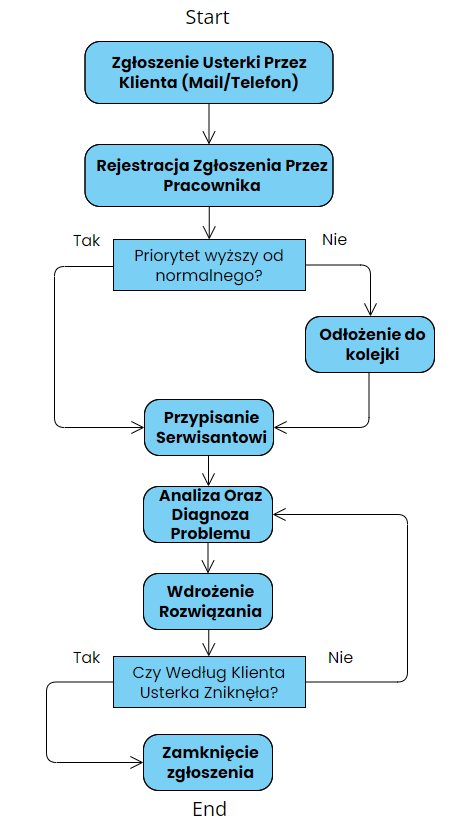
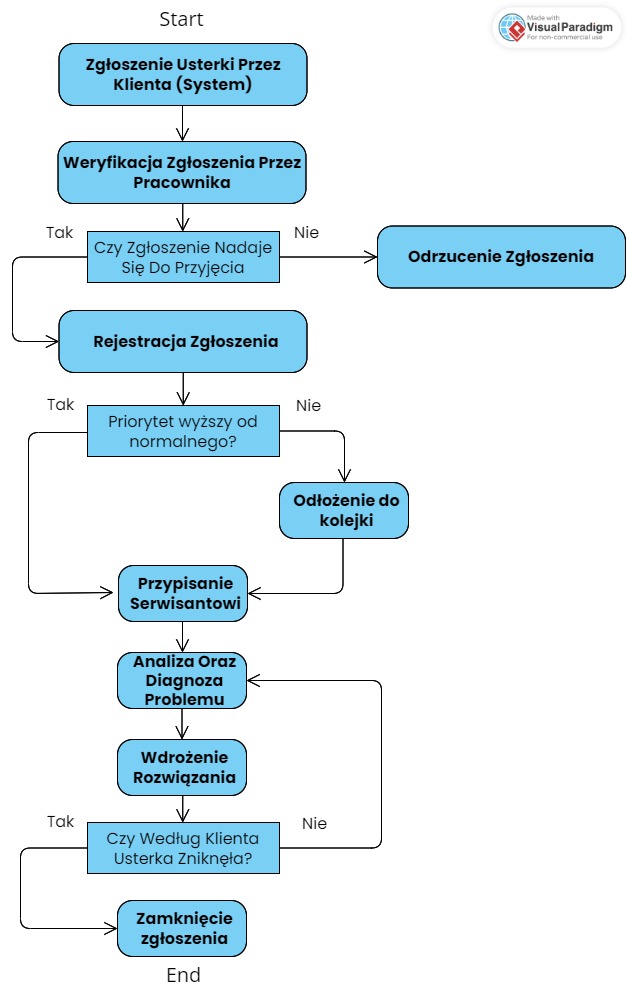
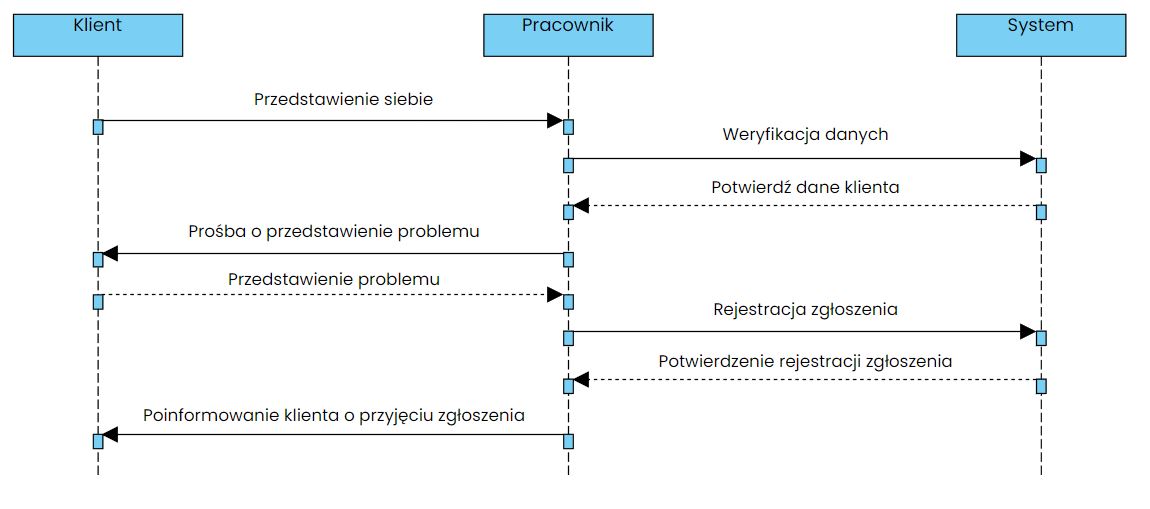
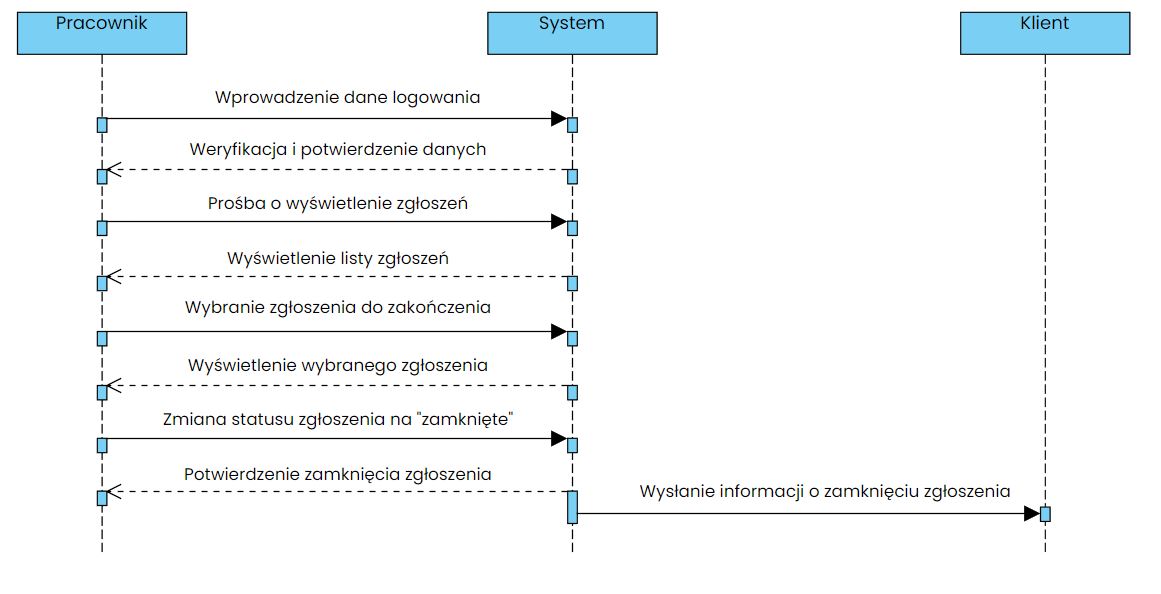


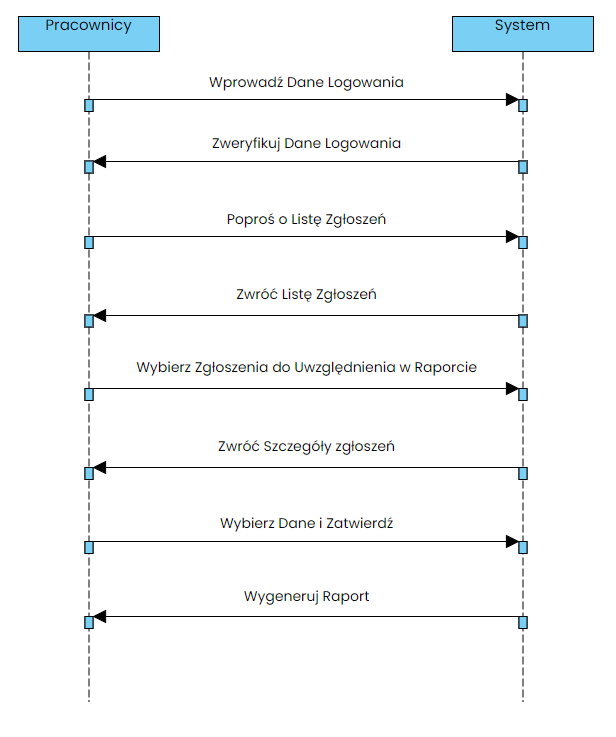
Diagram założenia zgłoszenia

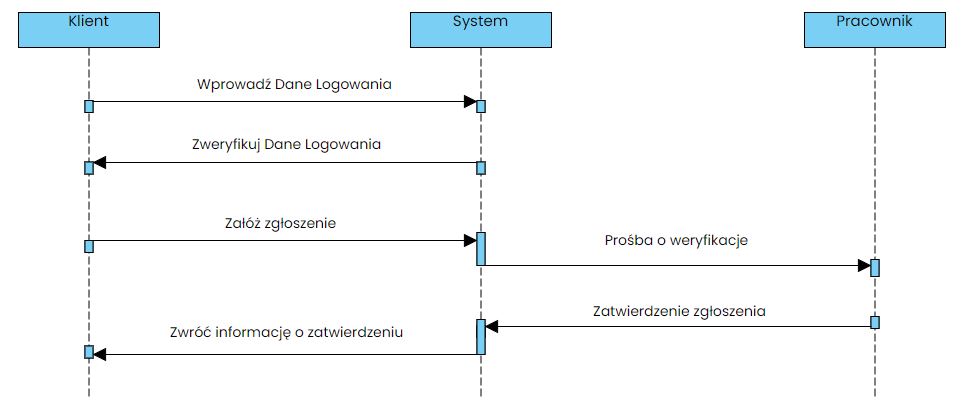


### Diagramy sekwencji









### Inne diagramy

Diagram maszyny stanowej zamknięcia zgłoszenia

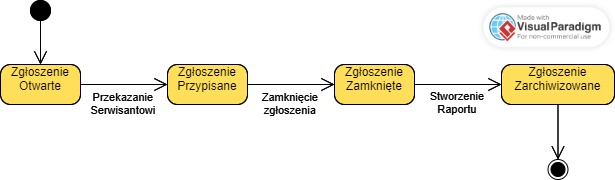


Diagram maszyny stanowej logowania

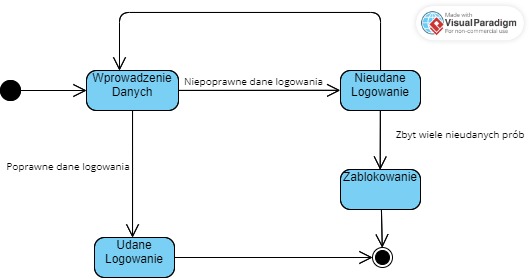
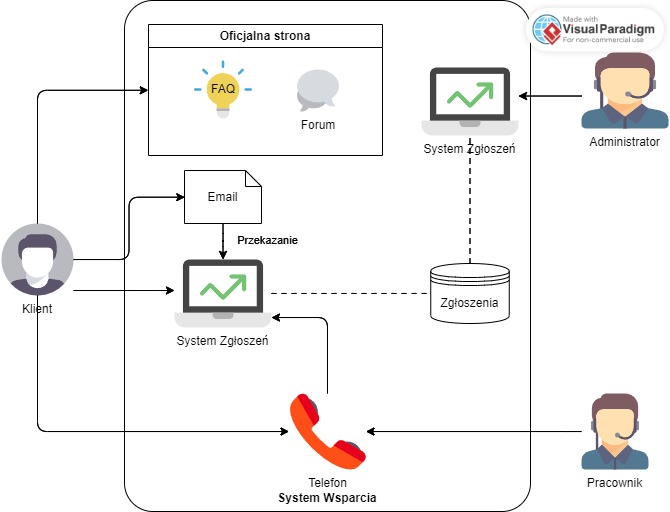


Diagram modelu biznesowego

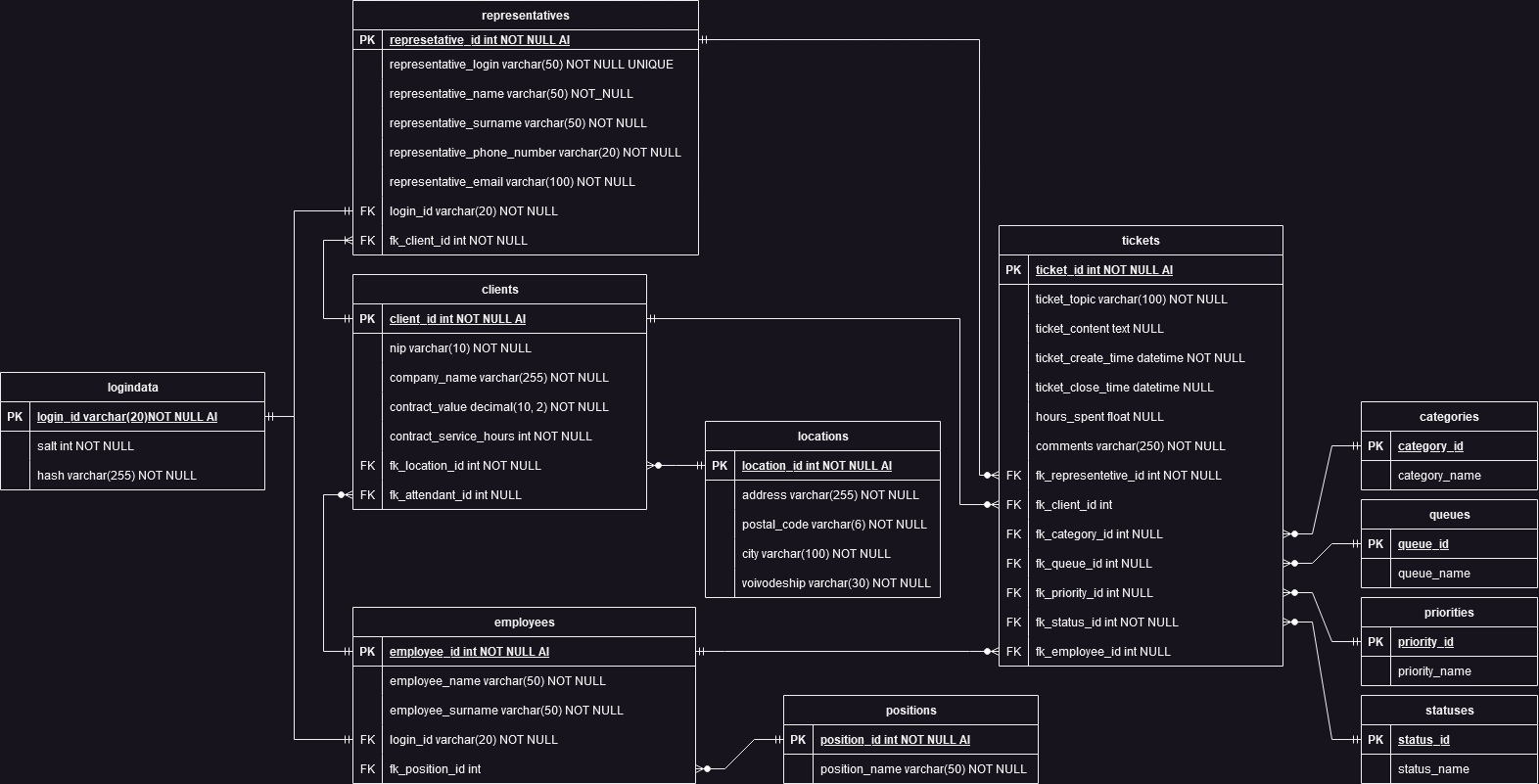


## Charakterystyka zastosowanych wzorców projektowych

Nie wykorzystaliśmy.

## Projekt bazy danych

### Schemat



### Projekty szczegółowe tabel

Tabela clients:

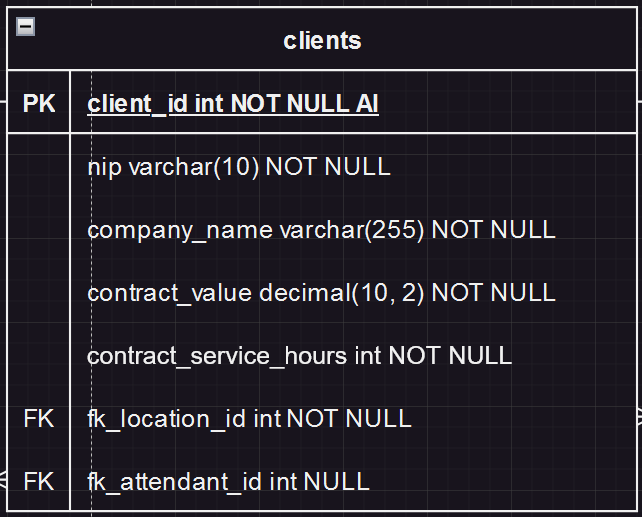


Tabela representatives:

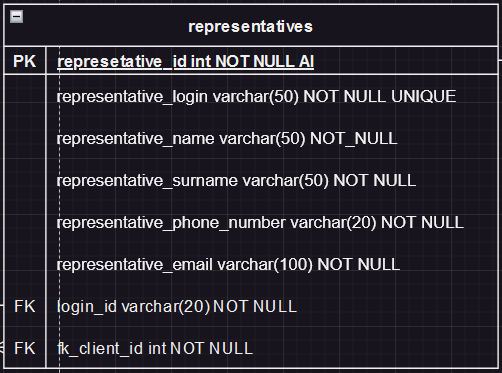


Tabela locations:

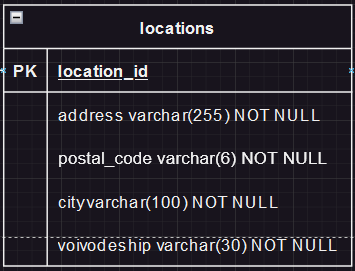
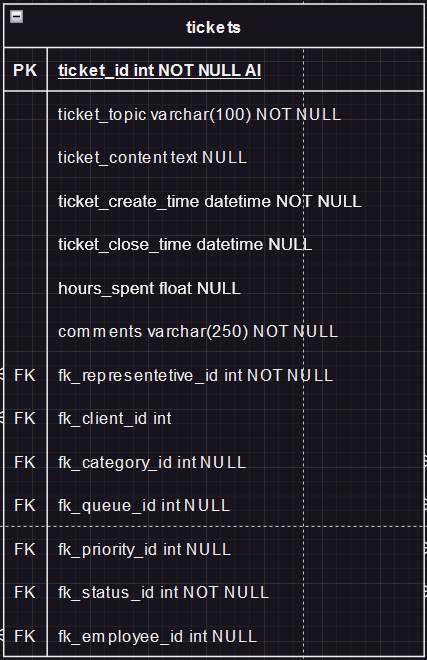


Tabela tickets:



Tabele uzupełniające tabelę tickets

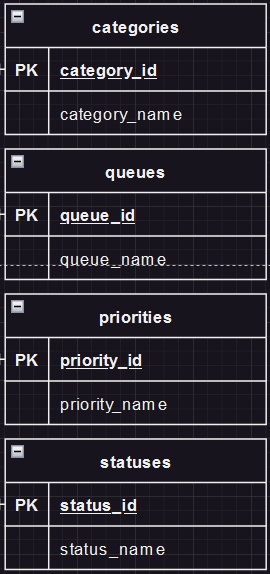


Tabela employees

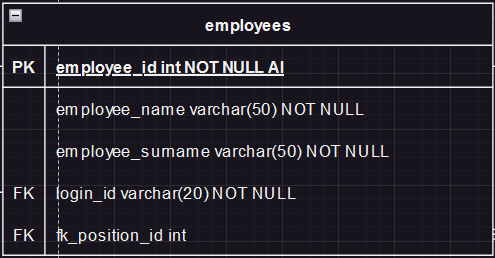


Tabela positions:

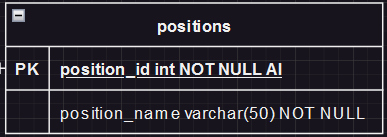
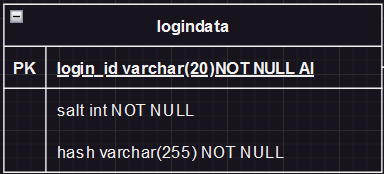


Tabela logindata:



## Projekt interfejsu użytkownika

### Lista głównych elementów interfejsu

* **Strona logowania**: Umożliwia użytkownikom wprowadzenie swoich danych logowania i uzyskanie dostępu do systemu.
* **Panel główny**: Centralne miejsce, które prezentuje podsumowanie najważniejszych informacji i aktywności użytkownika po zalogowaniu.
* **Strona klientów**: Prezentuje listę klientów wraz z informacjami kontaktowymi, historią zgłoszeń, statusami itp.
* **Strona zgłoszeń**: Wyświetla listę zgłoszeń wraz z ich szczegółami, umożliwia przeglądanie, edycję, dodawanie i usuwanie zgłoszeń.
* **Okno tworzenia zgłoszenia**: Pozwala użytkownikowi na wprowadzenie danych dotyczących nowego zgłoszenia, takie jak tytuł, opis, priorytet itp.
* **Okno edycji zgłoszeni**a: Umożliwia użytkownikowi modyfikację danych zgłoszenia, takie jak zmiana statusu, przypisanie do innego użytkownika itp.
* **Strona raportów i statystyk**: Prezentuje różne raporty i statystyki dotyczące zgłoszeń, klientów, wydajności itp.
* **Okno powiadomień**: Wyświetla powiadomienia o ważnych informacjach, aktualizacjach lub komunikatach systemowych.
* **Strona ustawień**: Pozwala użytkownikowi dostosować ustawienia systemu, takie jak preferencje powiadomień, preferencje wyglądu itp.
* **Aktywność systemowa**: Rejestruje historię działań i zmian w systemie, umożliwiając użytkownikowi śledzenie zdarzeń i audyt.

### Przejścia między głównymi elementami

1. Logowanie użytkownika:
   1. Użytkownik wprowadza dane logowania (nazwa użytkownika i hasło) na stronie logowania.
   2. System sprawdza poprawność danych logowania.
   3. Jeśli dane są poprawne, użytkownik zostaje przekierowany do panelu głównego. W przeciwnym razie wyświetlany jest komunikat o błędzie logowania.
2. Przeglądanie zgłoszeń:
   1. Użytkownik wybiera opcję "Zgłoszenia" z menu nawigacyjnego.
   2. System wyświetla listę zgłoszeń na stronie zgłoszeń.
   3. Użytkownik może przeglądać szczegóły zgłoszenia, takie jak tytuł, opis, status itp., klikając na odpowiednią pozycję.
3. Dodawanie nowego zgłoszenia:
   1. Użytkownik wybiera opcję "Nowe zgłoszenie" na stronie zgłoszeń.
   2. System wyświetla okno tworzenia zgłoszenia.
   3. Użytkownik wprowadza niezbędne informacje, takie jak tytuł, opis, priorytet itp., a następnie zatwierdza zgłoszenie.
   4. System dodaje nowe zgłoszenie do listy zgłoszeń.
4. Edycja zgłoszenia:
   1. Użytkownik wybiera konkretną pozycję zgłoszenia na stronie zgłoszeń.
   2. System wyświetla okno edycji zgłoszenia z danymi zgłoszenia.
   3. Użytkownik wprowadza żądane zmiany, takie jak zmiana statusu, przypisanie do innego użytkownika itp., a następnie zapisuje zmiany.
   4. System aktualizuje dane zgłoszenia na liście zgłoszeń.
5. Przeglądanie raportów i statystyk:
   1. Użytkownik wybiera opcję "Raporty" z menu nawigacyjnego.
   2. System wyświetla stronę raportów i statystyk.
   3. Użytkownik może wybierać różne raporty i statystyki, które chce przeglądać, na przykład wykresy, liczby zgłoszeń w różnych kategoriach itp.
6. Ustawienia użytkownika:
   1. Użytkownik wybiera opcję "Ustawienia" z menu nawigacyjnego.
   2. System wyświetla stronę ustawień, gdzie użytkownik może dostosować preferencje, takie jak powiadomienia, preferencje wyglądu itp.

### Projekty szczegółowe poszczególnych elementów

dla 5-7 głównych elementów

każdy element od nowej strony z następującą minimalną zawartością:

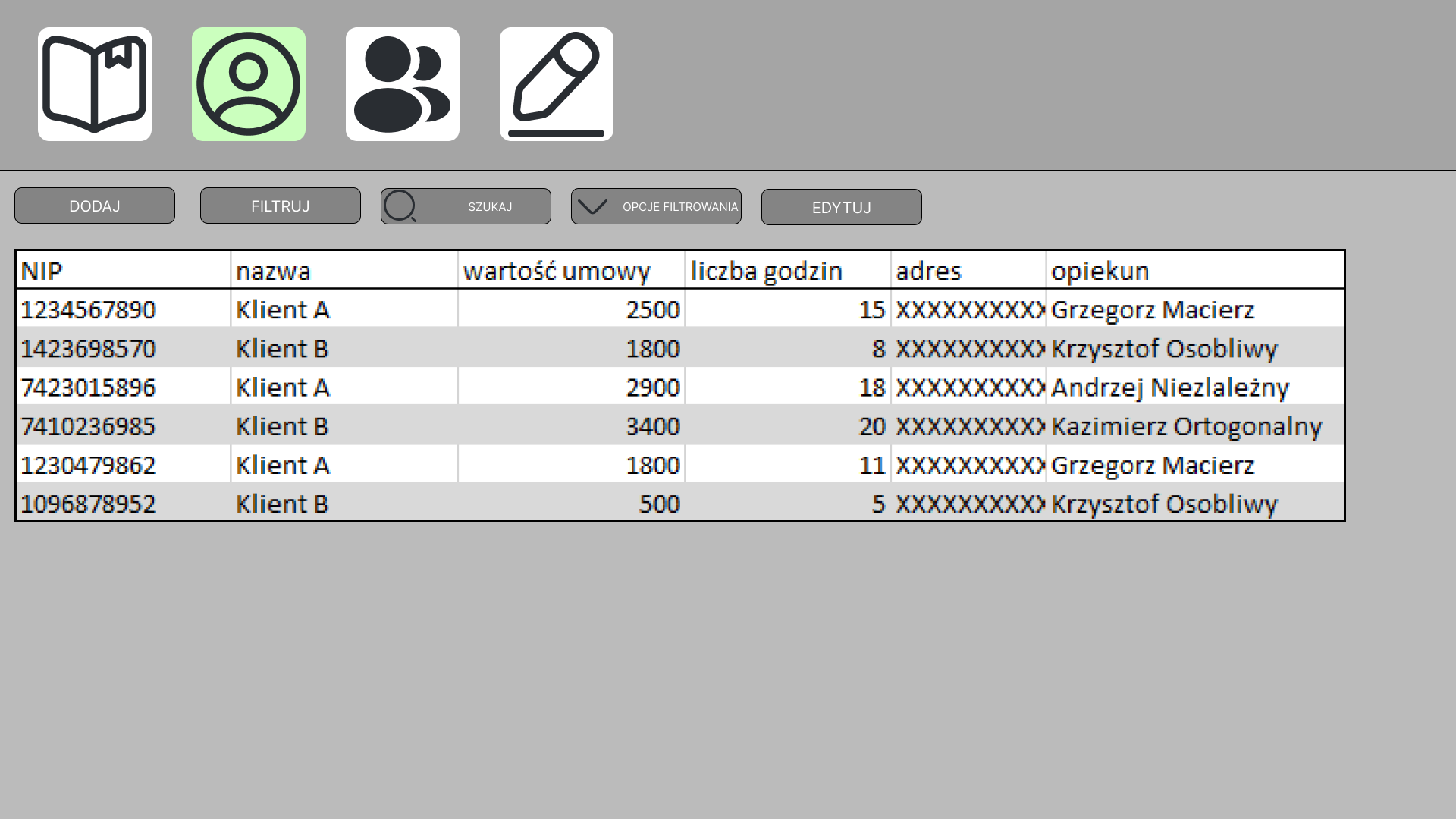
* numer – ID elementu
* nazwa – np. formularz danych produktu
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi informacjami (nie pusty!!!)
* opcjonalnie:
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco 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.

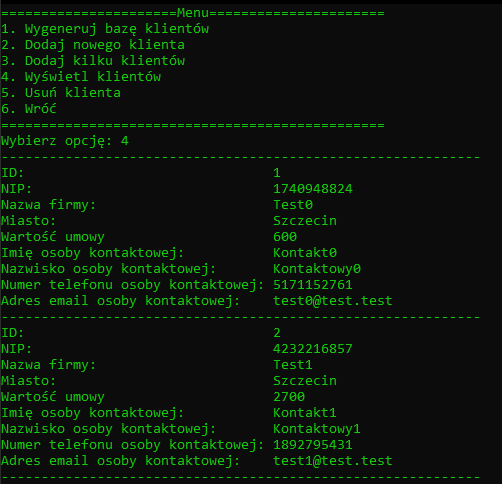
1. Strona logowania



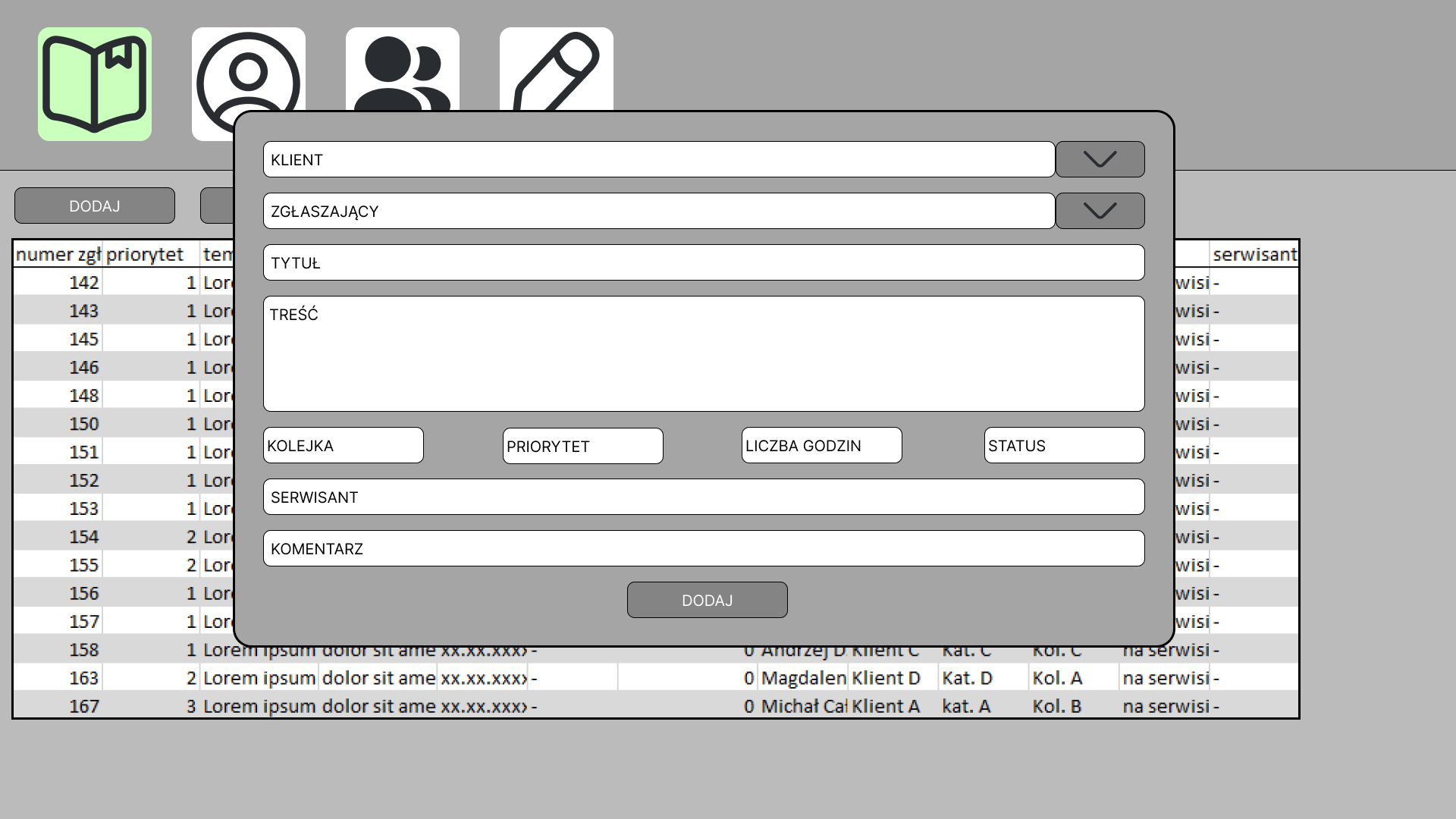


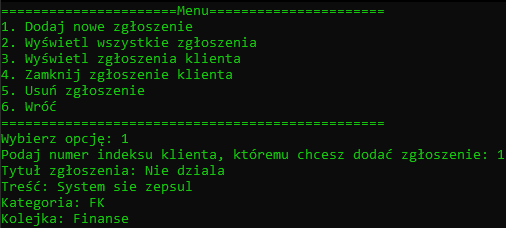
1. Strona klientów



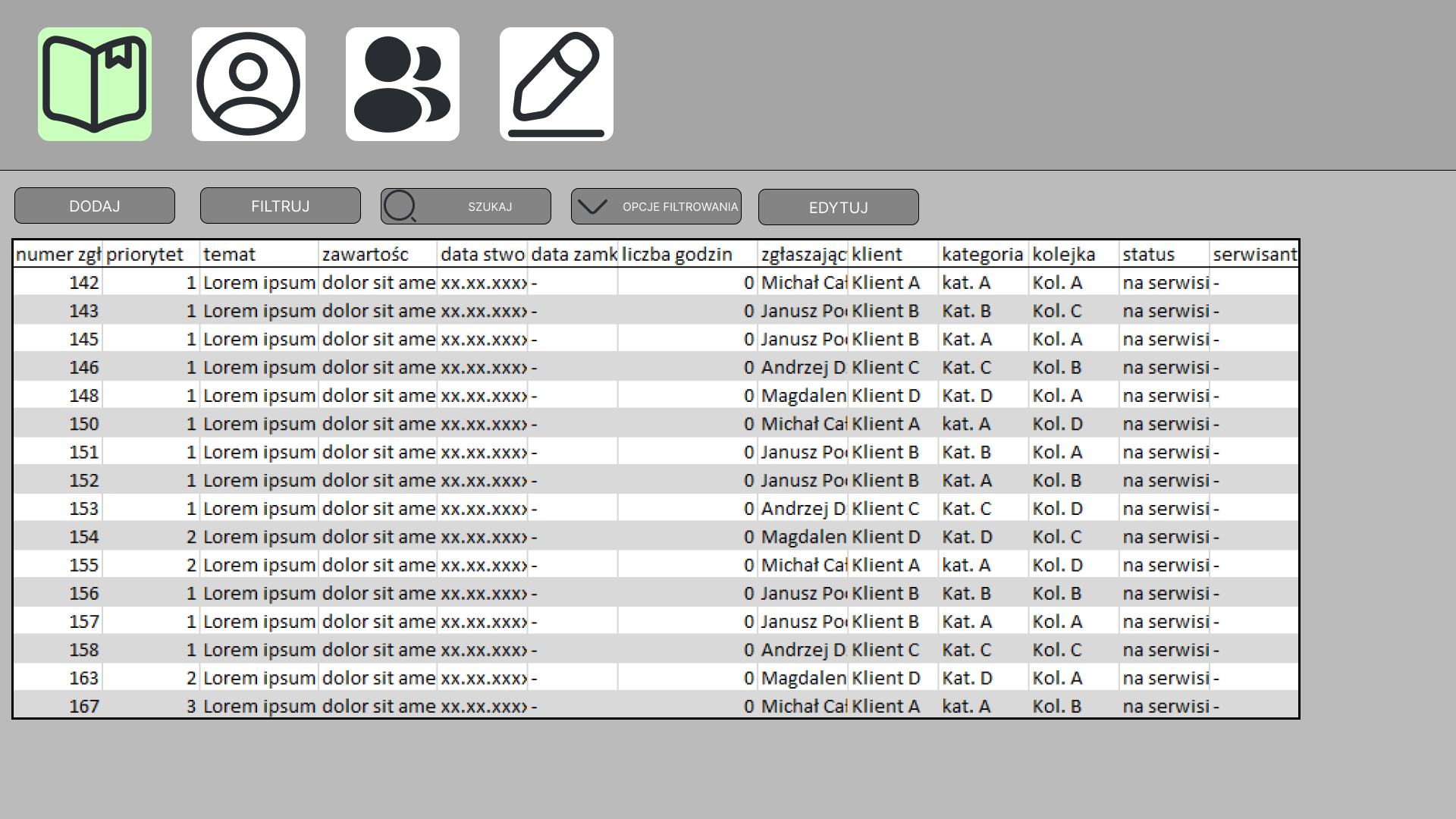


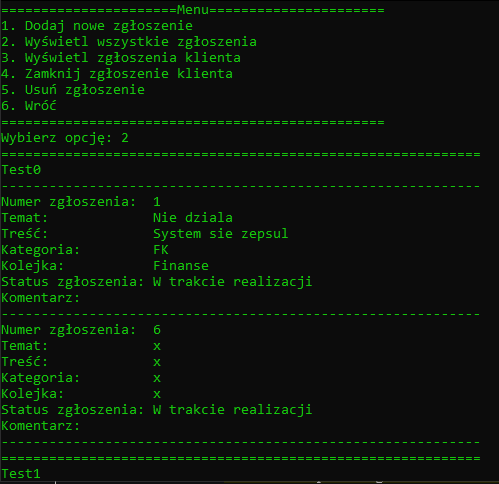
1. Okno tworzenia zgłoszenia



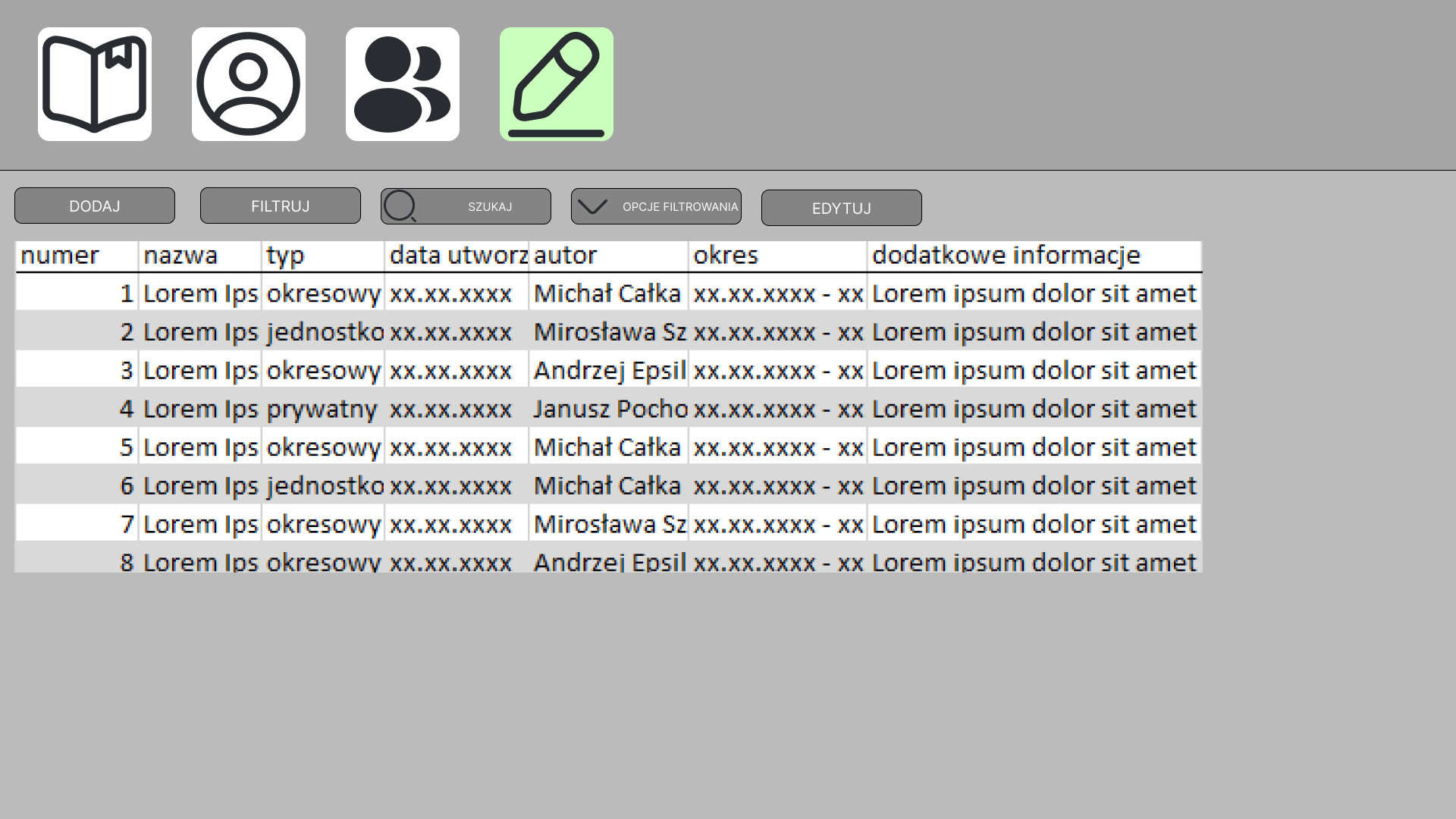


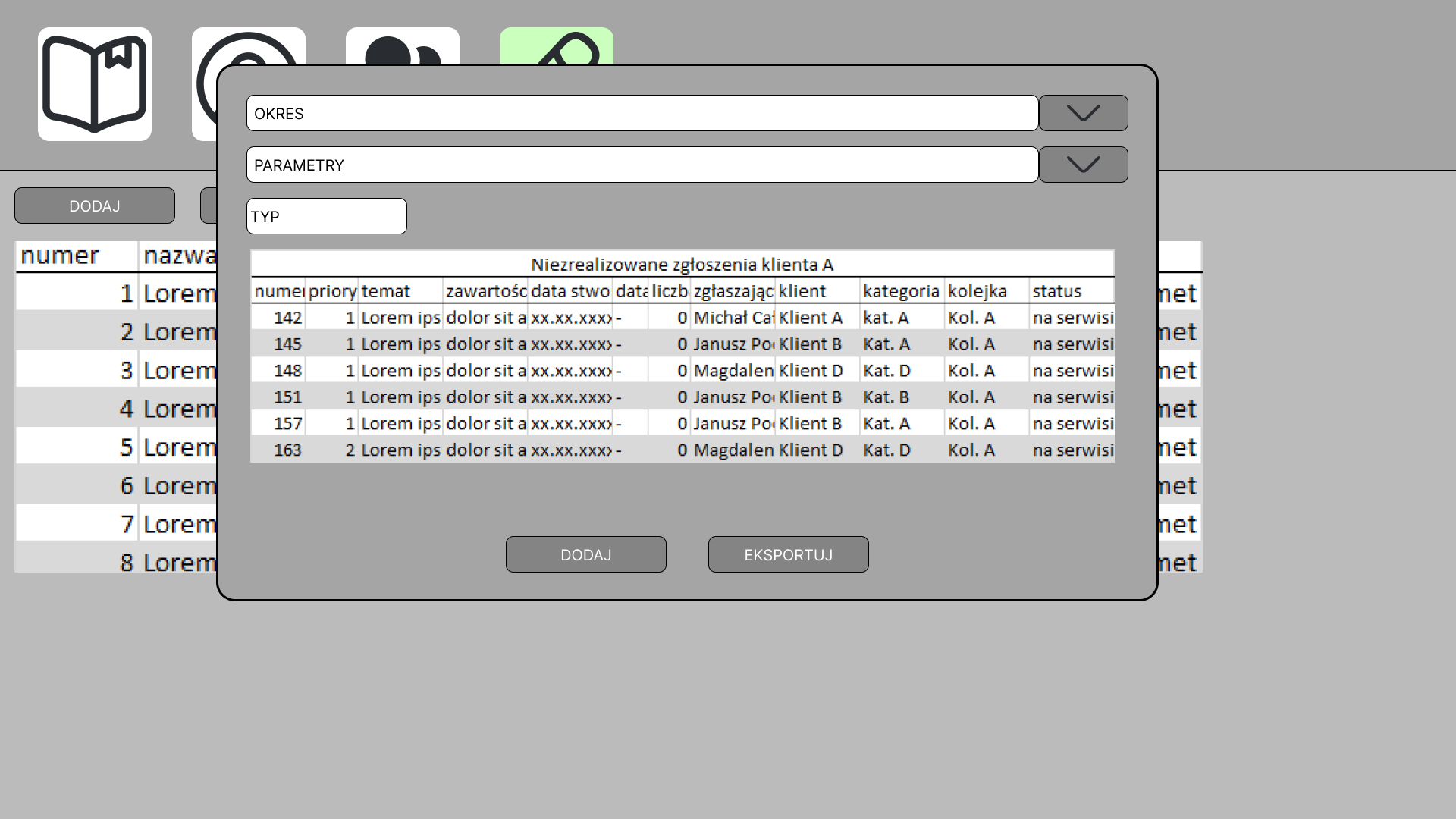
1. Strona zgłoszeń





1. Okno raportów





## Procedura wdrożenia

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

1. Analiza wymagań wdrożenia: Przeprowadzenie szczegółowej analizy wymagań dotyczących wdrożenia systemu. Określenie zasobów, harmonogramu, planu szkoleń i innych czynników kluczowych dla procesu wdrożenia.
2. Przygotowanie środowiska: Zapewnienie odpowiedniego środowiska do wdrożenia systemu, takiego jak serwery, infrastruktura sieciowa, oprogramowanie bazodanowe itp. Skonfigurowanie środowiska w zgodności z wymaganiami systemu.
3. Instalacja i konfiguracja systemu: Instalacja oprogramowania systemu na odpowiednich serwerach i stacjach roboczych. Konfiguracja systemu zgodnie z określonymi wymaganiami, takimi jak dostęp do bazy danych, ustawienia sieciowe, uprawnienia użytkowników itp.
4. Testowanie systemu: Przeprowadzenie testów w celu sprawdzenia poprawności funkcjonowania systemu. Testowanie może obejmować testy jednostkowe, testy integracyjne, testy wydajności, testy akceptacyjne itp. Wykrywanie i rozwiązywanie ewentualnych błędów i problemów.
5. Migracja danych: Przeniesienie istniejących danych z dotychczasowych systemów do nowego systemu. Konwersja danych, importowanie i sprawdzanie poprawności danych w nowym środowisku.
6. Szkolenie użytkowników: Przeprowadzenie szkoleń dla użytkowników systemu w celu zapoznania ich z funkcjonalnościami, interfejsem i procesami systemu. Zapewnienie odpowiednich materiałów szkoleniowych i wsparcia dla użytkowników.
7. Przebieg wdrożenia: Uruchomienie systemu w środowisku produkcyjnym. Monitorowanie i śledzenie działania systemu po wdrożeniu, w celu zapewnienia jego poprawnej funkcjonalności i reakcji na potrzeby użytkowników.
8. Ocena i dostosowanie: Przeprowadzenie oceny wdrożenia systemu, zbieranie opinii od użytkowników i zainteresowanych stron. Wprowadzanie ewentualnych dostosowań, poprawek i ulepszeń do systemu na podstawie zgromadzonych informacji i opinii.
9. Konserwacja i wsparcie: Zapewnienie długoterminowej konserwacji i wsparcia systemu po jego wdrożeniu. Świadczenie usług wsparcia technicznego, rozwiązywanie bieżących problemów i dostarczanie aktualizacji i poprawek.

# Dokumentacja dla użytkownika

-

# Podsumowanie

## 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

|  |  |
| --- | --- |
| Joachim Welk | Jakub Ociepa |
|  |  |
| 55% | 45% |

# Inne informacje

Brak