PRO Inżynieria Oprogramowania

**Temat**: „System obsługi zgłoszeń serwisowych dla IT", "Opis: Platforma do zgłaszania awarii, przydzielania techników i monitorowania napraw.”

Hubert Michna, w67259, 6IIZ/2022-GPL04

# **Opis inżynierii wymagań**

## **Cel projektu**

Celem tego projektu było przygotowanie prostego i intuicyjnego systemu, który pozwoli użytkownikom na tworzenie zgłoszeń dotyczących problemów IT. Takie zgłoszenia będą mogły być obsługiwane przez wiele osób w celu pomocy użytkownikom.

## **Wymagania funkcjonalne**

Wymagania funkcjonalne jakie spełnia ten projekt to:

* System kont użytkowników.
* Możliwość tworzenia zgłoszeń.
* Możliwość przeglądania istniejących zgłoszeń.
* Możliwość dodawania wiadomości do zgłoszeń, tak żeby były widoczne dla innych użytkowników systemu.
* Edytowanie statusów zgłoszeń, przez co można na pierwszy rzut oka pogrupować zgłoszenia ze względu na ich tematyka i stopień ważności.

## **Wymagania niefunkcjonalne**

Wymagania funkcjonalne jakie spełnia ten projekt to:

* Prostota obsługi dla użytkowników.
* Wysoka responsywność, która pozwala na poprawne i efektywne na urządzeniach o mniejszej szerokości ekranu, telefonach i tabletach.
* Skalowalność.
* Wysokie bezpieczeństwo zbieranych.
* Wysoka niezawodność.
* Możliwość tworzenia kopii zapasowych.

# **Opis przewidywanych zagrożeń**

Przewidywane zagrożenia z jakimi może spotkać się ten projekt to:

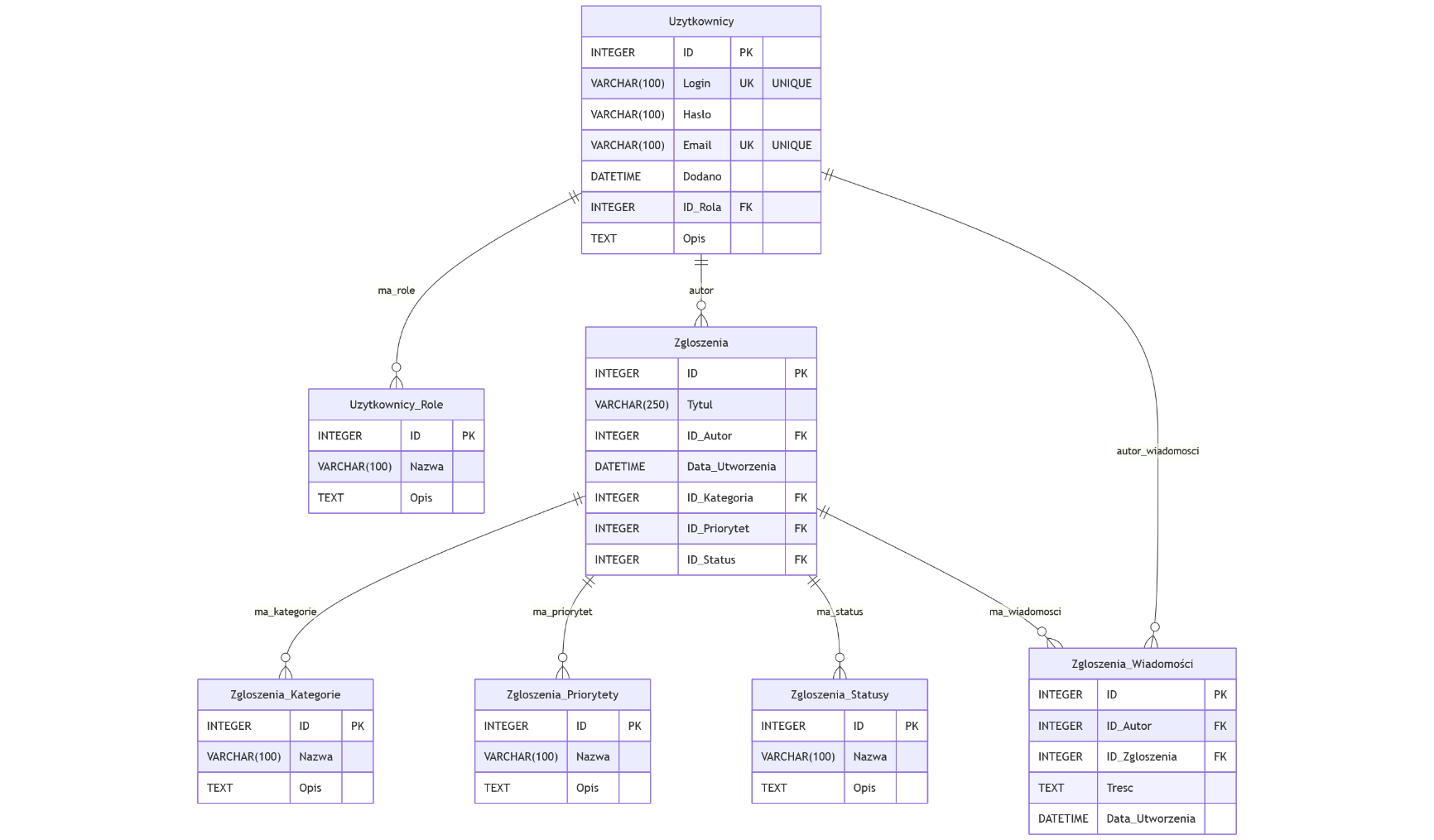
* Przekroczenie czasu dostarczenia oprogramowania.
* Przekroczenie dopuszczonego budżetu.
* Błędy techniczne uniemożliwiające poprawne działanie oprogramowania.
* Zmienny zakres projektu.
* Problemy z testowaniem oprogramowania.

# **Dokumentację UML projektu**

Niektóre z poniższych diagramów zostały przygotowane i wygenerowane poprzez Mermaid.js.

## **Diagram ERD**

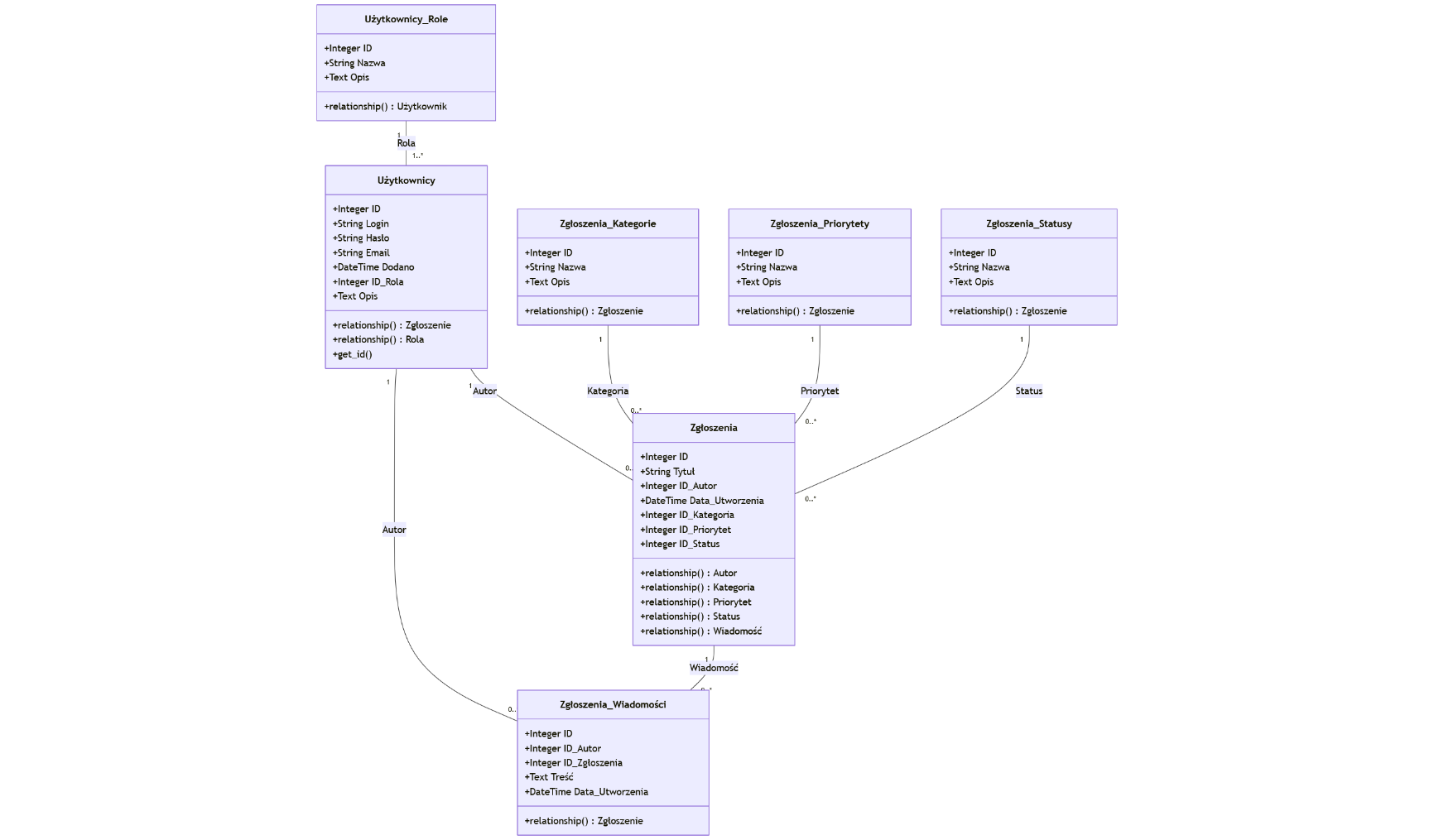
Poniższy diagram przedstawia relacje pomiędzy tabelami/modelami.



Rysunek - Diagram ERD

## **Diagram klas**

Poniższy diagram przedstawia najważniejsze klasy w tym oprogramowaniu oraz ich relacje ze sobą.



Rysunek - Diagram klas

## **Diagram przypadków użycia**

Poniżej znajduje się diagram przypadków użycia, z podziałem na zwykłego, nie zarejestrowanego/nie zalogowanego użytkownika i zalogowanego użytkownika.



Rysunek - Diagram przypadków użycia

# **Opis wytworzonego oprogramowania**

Przygotowana została aplikacja internetowa napisana w języku Python z wykorzystaniem frameworka Flask, inne istotne frameworki z jakich korzysta to oprogramowanie to: flask\_login, flask\_sqlalchemy, wtforms i flask\_wtf. W celu zapewnienia spójnego i responsywnego wyglądu wykorzystano framework Bootstrap 5.

Oprogramowanie wykorzystuje wzorzec projektowy „Application Factory”, w celu umożliwienia podziału aplikacji na modułu, prostszą rozbudowę i wygodniejsze testowanie.

Aplikacja jest rozdzielona na następujące moduły:

* „**Main**” – Główny moduł ogólnego przeznaczenia.
* „**Konta**” – Moduł, gdzie znajduje się wszystko związane z kontami użytkowników, czyli np. logowanie, rejestracja czy edycja konta.
* „**Zgłoszenia**” – Moduł, gdzie znajduje się wszystko związane ze zgłoszeniami, tutaj użytkownicy mogą m.in. dodawać nowe zgłoszenia, wyświetlać i edytować je oraz dodawać swoje wiadomości, które będą zapisane w historii zgłoszeń.

Osoby korzystające z tego oprogramowania można podzielić na:

* Zwykłych, niezalogowanych użytkowników – Mogą się zarejestrować i wyświetlać zgłoszenia, ale nie mogą dodawać wiadomości ani edytować zgłoszeń. Mogą je tylko wyświetlać.
* Zalogowanych użytkowników – Mogą zmienić dane swojego konta, mają dostęp do tych samych możliwości co niezalogowani użytkownicy, ale mogą też dodatkowo tworzyć i edytować zgłoszenia.

# **Kompletne testy oprogramowania**

Przeprowadzone testy jednostkowe i testy integracyjne wraz z konfiguracją znajdują się w „Flask\_Aplikacja/Aplikacja/Testy”. Testy zostały przygotowane z wykorzystaniem frameworka pytest.

Na poniższym zrzucie ekranu znajduje się pomyślny wynik przeprowadzonych testów:

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

Rysunek - Testy

# **Weryfikację oraz walidację wytworzonego oprogramowania**

Weryfikacja i walidacja została wykona. Weryfikacja pozwoliła na wychwycenie błędów kodu aplikacji i wynikających z nich potencjalnych niebezpieczeństw/problemów z działaniem aplikacji. Kod został utworzony zgodnie ze standardami i dostępną dokumentacja. Potwierdzeniem tego jest znajdujący się powyżej screen z pozytywnym wynikiem przeprowadzonych testów

Walidacja również została wykonana, dostarczone oprogramowanie spełnia przedstawione oczekiwania i zostało zaakceptowane jako poprawne. Zostało przeprowadzone prototypowanie i demonstracje gotowych fragmentów aplikacji jak i całości systemu.

# **Możliwe plany rozwoju oprogramowania**

To oprogramowanie można w przyszłości rozbudować między innymi o:

* Możliwość filtrowania zgłoszeń, żeby nie trzeba było samemu szukać interesujących nas zgłoszeń.
* Dodanie stronicowania przy widoku zgłoszeń.
* Rozbudowanie formularzy o bardziej intuicyjne komunikaty z uwagami/błędami co do danych wprowadzonych przez użytkowników.
* Możliwość wysyłania załączników.
* Możliwość wysyłania powiadomień, poprzez np. pocztę elektroniczną.
* Możliwość oznaczania innych użytkownikach we własnych wiadomościach.
* Wprowadzenie rozbudowanego systemu uprawnień.
* Przygotowanie bardziej odpowiednich statycznych komunikatów tekstowych.
* Poprawienie i dostosowanie warstwy wizualnej.
* Obsługa ciemnego motywu.
* Mechanizmy zwiększające dostępność cyfrową.

# **Podsumowanie**

W ramach tego projektu przygotowano aplikację webową przygotowaną w Pythonie, z wykorzystaniem frameworka Flask, stosując wzorzec projektowy „Application Factory”. Język Python został wybrany ze względu na swoją prostotę, wygodę obsługi, dużą ilość przydatnych bibliotek oraz ponieważ ta technologia pokrywa się z innymi projektami realizowanymi w tym semestrze.

To oprogramowanie działa poprawnie i odpowiednio realizuję temat projektu.

# **Kod źródłowy**

Kod źródłowy programu jest dostępny na [GitHub’ie](https://github.com/Hubsik456/PRO_Inzynieria_Oprogramowania).

Znajduje się tam też plik „requirements.txt” z informacją o wszystkich używanych bibliotekach.