

# Obsługa błędów

Projekt na przedmiot zespołowy

Nazwa zespołu:	Pju-Pju Mahen
Kierownik zespołu:	Wojciech Niestrój
Backend:	Wojciech Niestrój
Frontend:	Dariusz Achciński, Krzysztof Biernacki
Analiza i dokumentacja:	Radosław Dąbrowski
Tester:	Radosław Dąbrowski
Typ projektu:	Web

## Spis treści

1.	Analiza wymagań.....	3
1.1.	Opis dziedziny problemowej .....	3
1.2.	Tytuł systemu .....	3
1.3.	Cel i opis podstawowych funkcjonalności systemu.....	3
1.4.	Słownik pojęć.....	4
1.5.	Systemowe diagramy przypadków użycia dla poszczególnych aktorów.....	5
1.6.	Diagram DDL.....	8
2.	Interfejs .....	9
2.1.	Panel logowania .....	9
2.2.	Po wprowadzeniu błędnego hasła .....	9
2.3.	Ekran rejestracji nowego użytkownika.....	10
2.4.	Błędnie wpisane hasła przy rejestracji .....	10
2.5.	Wymagania przy rejestracji .....	11
2.6.	Panel po zalogowaniu.....	11

## 1. Analiza wymagań

### 1.1. Opis dziedziny problemowej

Firma XYZ pragnie mieć system informatyczny, dzięki któremu osoby pracujące na tym systemie będą mogły zgłaszać błędy występujące w programie zakupionym w danej firmie, a także śledzić stan naprawy danego błędu.

Osoba pracująca w firmie (Pracownik) będzie mógł zgłosić swój błąd przez specjalnie przygotowany formularz zgłoszeniowy. Będzie mógł zawrzeć w nim krótki opis błędu, nadać priorytet, a także dodać zrzut ekranu danego błędu.

Zgłoszenie to następnie trafi do jednostki firmy odpowiedzialnej za dany program. Tam developer może zmienić priorytet błędu i przekazać zgłoszenie do programisty/ów. Programiści następnie naprawiają daną usterkę i przekazują poprawiony program do działu testerów. Testerzy następnie sprawdzają, czy nie pojawiły się nowe błędy w związku z naprawą. Jeżeli takowych nie ma zgłoszenie dostaje status „naprawione” oraz informację w jakim czasie zostanie wprowadzona poprawka do programu. Jeżeli natomiast pojawiły się błędy to problem zostaje jeszcze raz skierowany do działu programistów. Cała procedura następnie przebiega jak za pierwszym razem.

Całość wykonywanych operacji obserwuje Kierownik. Ma on wpływ na każdy z etapów zgłoszenia błędu (np. jeżeli tak uzna może zmienić priorytet błędu).

### 1.2. Tytuł systemu

Aplikacja internetowa „Obsługa błędów”

### 1.3. Cel i opis podstawowych funkcjonalności systemu

Cele systemu:

- Dodanie zgłoszenia
- Edycja danych zgłoszenia (takich jak priorytet itp.)
- Komunikacja pomiędzy poszczególnymi działami
- Komunikacja użytkownik-developer

## 1.4. Słownik pojęć

Konto – przypisane do każdego użytkownika unikalny login oraz dane użytkownika. Dostęp do konta zabezpieczony jest hasłem.

Kierownik – jest to użytkownik, który posiada wszystkie prawa. Dogląda wszelkich prac dotyczących programu.

Użytkownik – zalogowana osoba, pracująca w firmie, w której wykorzystywany jest nasz program. Może wysłać zgłoszenie błędu.

Developer – osoba zajmująca się komunikacją z użytkownikiem. Sprawdza, a następnie przekazuje zgłoszenie błędu do programisty.

Programista – osoba odpowiedzialna za naprawę błędu wynikającego ze zgłoszenia.

Tester – osoba, do której trafia poprawiona wersja programu. Ma za zadanie sprawdzić poprawność działania poprawionego programu.

Zgłoszenie błędu – formularz, w którym zawarte są najważniejsze informacje dotyczące występującego błędu (opis błędu, priorytet błędu, zrzut ekranu z błędem, itp.).

Priorytet zgłoszenia – jest to status nadawany przy zgłoszeniu błędu. Priorytet może być wysoki, co oznacza, że bez naprawy tego błędu program nie może być obsługiwany, średni, co znaczy, że program może działać bez naprawy błędu jednak to działanie jest dość uciążliwe, lub niski, jeżeli program działa poprawnie, a błąd nie wpływa znacząco na pracę.

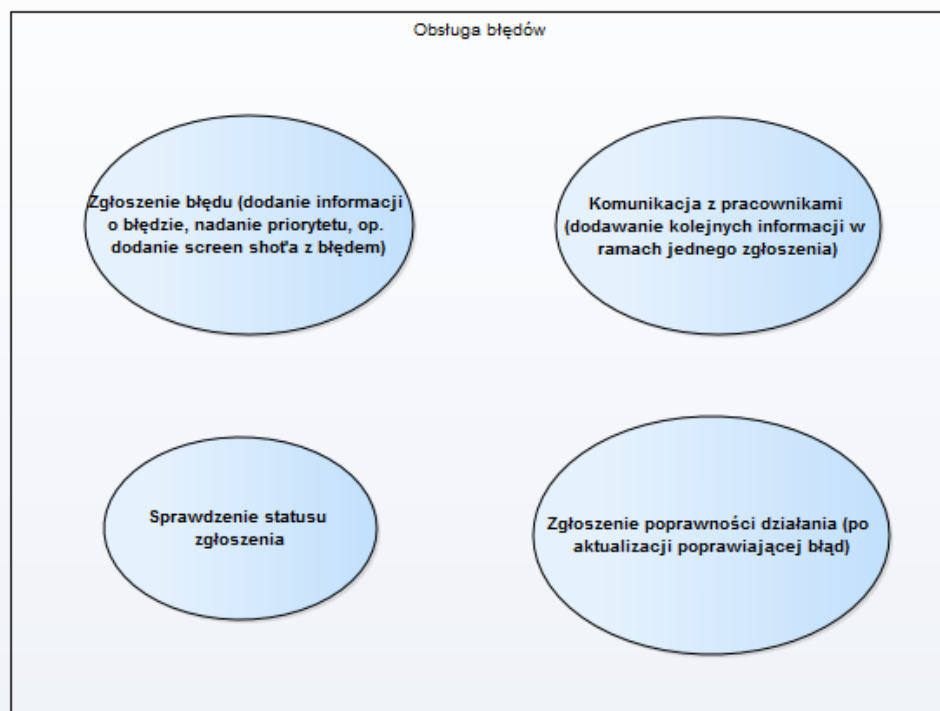
Status zgłoszenia – jest to informacja dotycząca zgłoszenia. Odnosi się ona do aktualnie wykonywanych prac w związku z danym zgłoszeniem, np. jeżeli zgłoszenie zostało odebrane już przez programistów i prowadzą oni prace związane z naprawą to zgłoszenie to otrzymuje status „W trakcie naprawy”. Status ten może podejrzeć użytkownik, który dane zgłoszenie wysłał.

Aktualizacja/poprawka/naprawa – ciąg czynności powodujących usunięcie błędu z programu.

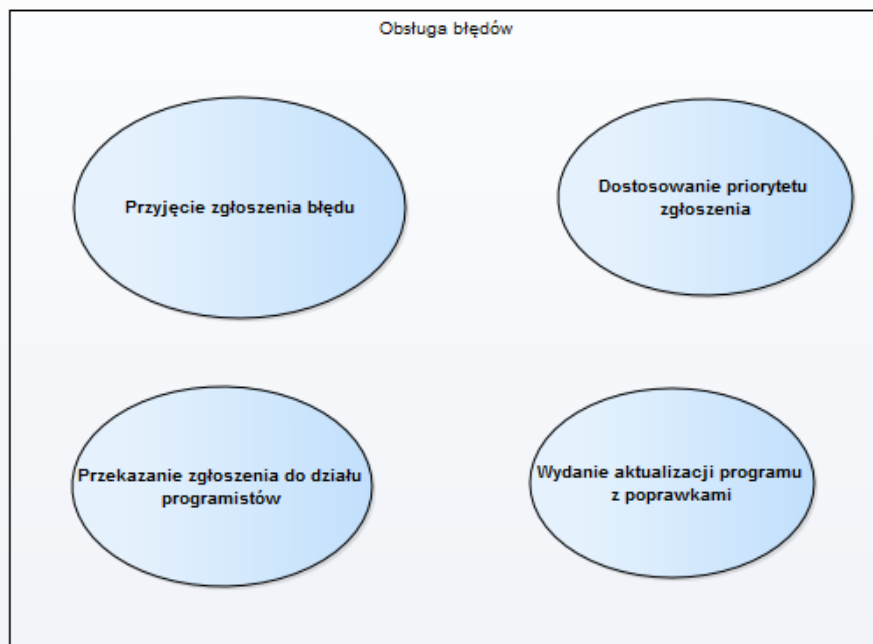
## 1.5. Systemowe diagramy przypadków użycia dla poszczególnych aktorów

Każdy użytkownik, programista, developer, tester, kierownik przed możliwością wykonywania swoich czynności musi zalogować się na swoje konto.

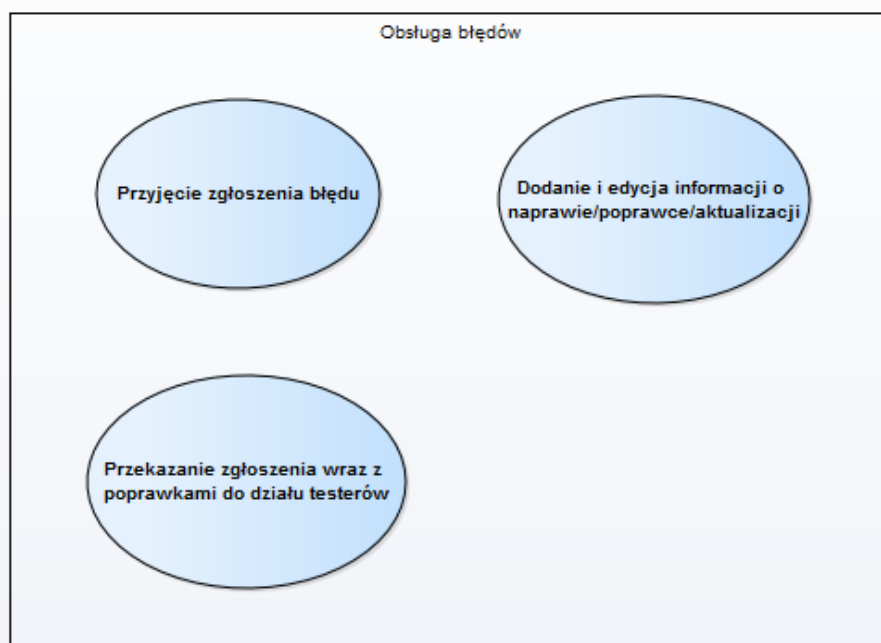
Systemowy diagram przypadków użycia dla użytkownika:



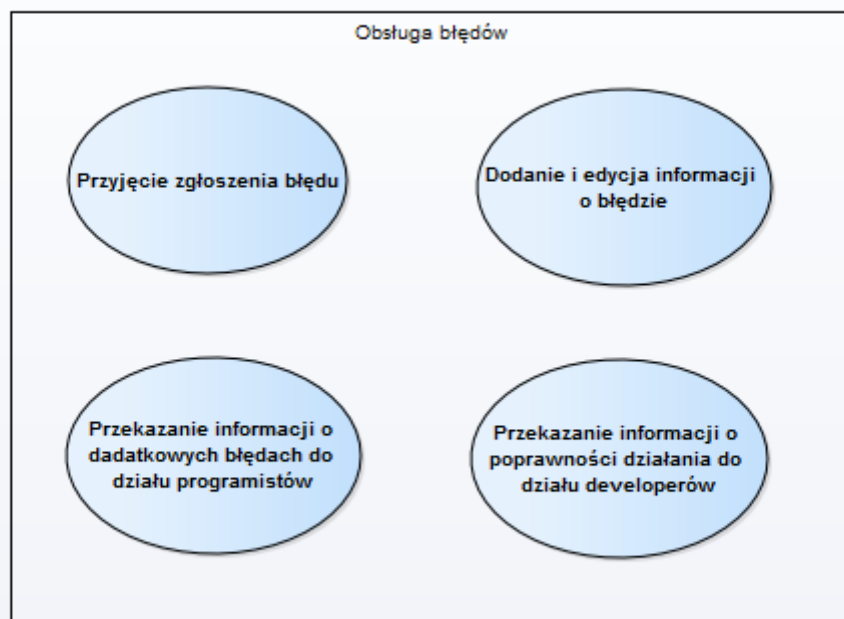
Systemowy diagram przypadków użycia dla developera:



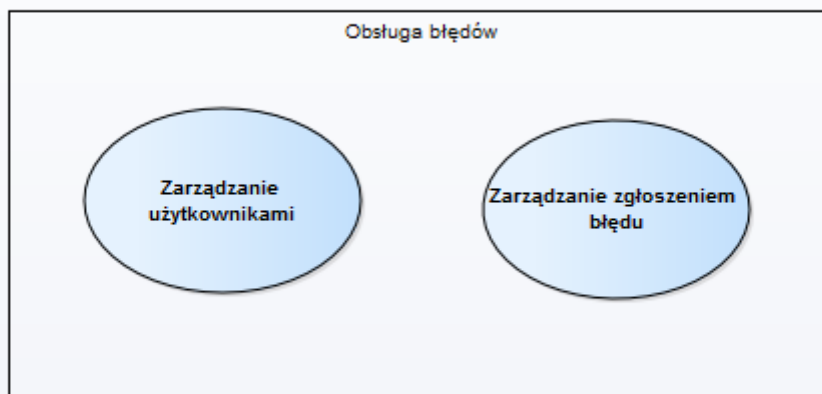
Systemowy diagram przypadków użycia dla programisty:



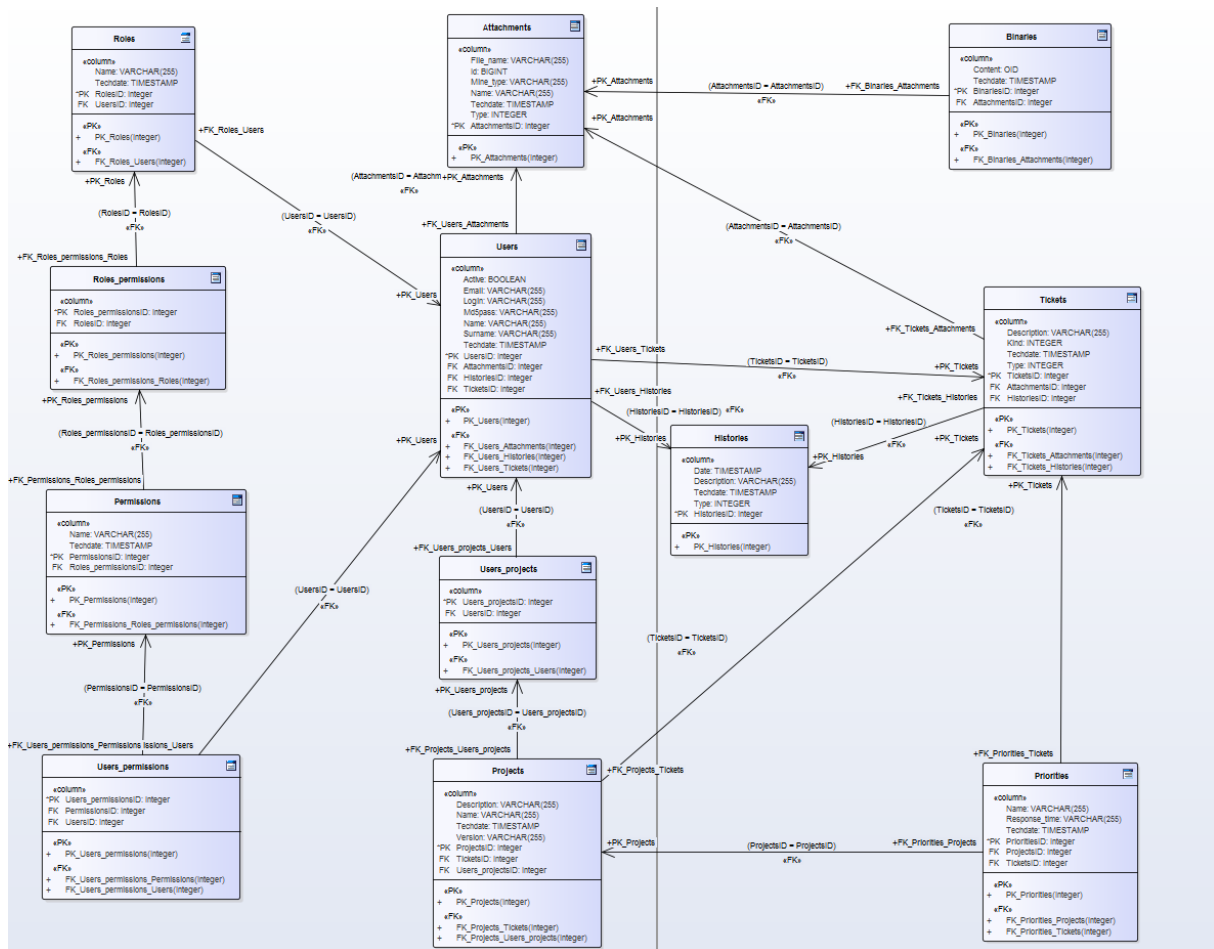
Systemowy diagram przypadków użycia dla testera:



Systemowy diagram przypadków użycia dla kierownika:



## 1.6. Diagram DDL





## 2. Interfejs

### 2.1. Panel logowania

Obsługa błędów

Zaloguj

Zarejestruj

### Logowanie

Login

Hasło

Zaloguj

### 2.2. Po wprowadzeniu błędnego hasła

Obsługa błędów

Zaloguj

Zarejestruj

Hasło nieprawidłowe

### Logowanie

Login

Hasło

Zaloguj

## 2.3. Ekran rejestracji nowego użytkownika

Obsługa błędów

[Zaloguj](#) [Zarejestruj](#)

### Rejestracja

Imię

Login

Email

Hasło

Wpisz ponownie hasło

## 2.4. Błędnie wpisane hasła przy rejestracji

Obsługa błędów

[Zaloguj](#) [Zarejestruj](#)

### Rejestracja

Hasła nie zgadzają się ze sobą

Imię

Login

Email

Hasło

Wpisz ponownie hasło

## 2.5. Wymagania przy rejestracji

Obsługa błędów

Zaloguj

Zarejestruj

### Rejestracja

Imię

fas

Login

fas

Email

fsafsa

Hasło

...

Wpisz ponownie hasło

.....

Zarejestruj

! Uwzględnij znak „@” w adresie e-mail. W adresie „fsafsa” brakuje znaku „@”.

## 2.6. Panel po zalogowaniu

Po zalogowaniu użytkownik może zgłosić błąd, przeglądając przy tym listę zgłoszonych przez siebie błędów.

E-Lista [Dashboard](#) [Logout](#)

ID

Nazwa

Opis

Nazwisko zgłaszającego

ID	Nazwa	Opis	Nazwisko Zgłaszającego	Edytuj/Usuń
----	-------	------	------------------------	-------------

