Aplikacja z systemem ticketowym

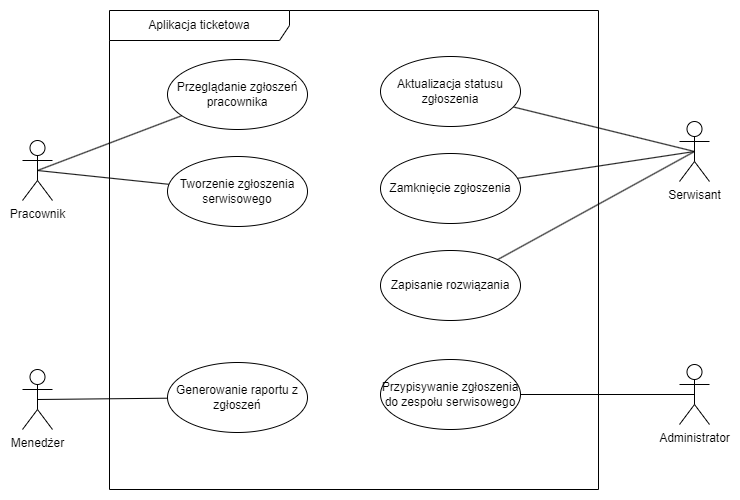
## 1.Krótki opis aplikacji

Aplikacja ticketowa to narzędzie zaprojektowane do efektywnego zarządzania zgłoszeniami serwisowymi i zadaniami w firmach. Jej głównym celem jest usprawnienie procesu zgłaszania problemów, monitorowania ich rozwiązywania oraz komunikacji między użytkownikami a zespołem wsparcia.

**Korzyści z wdrożenia aplikacji:**

* **Zwiększona efektywność:** Dzięki automatyzacji i przejrzystości procesów, czas potrzebny na rozwiązanie zgłoszeń jest znacznie skrócony.
* **Lepsza organizacja pracy:** Możliwość przypisywania zadań i monitorowania ich realizacji pomaga w lepszym zarządzaniu zasobami.
* **Poprawa komunikacji:** Bezpośrednia komunikacja w aplikacji zmniejsza liczbę nieporozumień i przyspiesza rozwiązywanie problemów.
* **Analiza i optymalizacja:** Dzięki szczegółowym raportom, firmy mogą lepiej zrozumieć swoje procesy i wprowadzać usprawnienia.
* **Zachowywanie rozwiązań problemów:** Umożliwia szybsze rozwiązanie problemów, które się powtarzają

## 2.DPU



## 3.Scenariusze

#### 1. Tworzenie zgłoszenia serwisowego

**Aktor:** Pracownik

**Opis:** Pracownik, napotykając problem techniczny, loguje się do aplikacji i tworzy nowe zgłoszenie serwisowe, podając szczegóły problemu i ewentualne załączniki.

**Kroki:**

1. Pracownik loguje się do aplikacji.
2. Wybiera opcję "Nowe zgłoszenie".
3. Wypełnia formularz zgłoszenia, podając tytuł, opis problemu i załączniki.
4. Wybiera priorytet zgłoszenia.
5. Przesyła zgłoszenie.

**Warunki początkowe:** Pracownik jest zalogowany w systemie.

**Warunki końcowe:** Nowe zgłoszenie zostaje zapisane w systemie i jest widoczne dla odpowiednich działów.

#### 2. Przypisywanie zgłoszenia do zespołu serwisowego

**Aktor:** Administrator

**Opis:** Administrator przegląda nowe zgłoszenia i przypisuje je do odpowiednich członków zespołu serwisowego.

**Kroki:**

1. Administrator loguje się do aplikacji.
2. Przegląda listę nowych zgłoszeń.
3. Wybiera zgłoszenie do przypisania.
4. Wybiera odpowiedniego członka zespołu serwisowego.
5. Przypisuje zgłoszenie do wybranego członka zespołu.

**Warunki początkowe:** Administrator jest zalogowany w systemie i przegląda listę zgłoszeń.

**Warunki końcowe:** Zgłoszenie jest przypisane do konkretnego członka zespołu serwisowego.

#### 3. Aktualizacja statusu zgłoszenia

**Aktor:** Serwisant

**Opis:** Serwisant aktualizuje status zgłoszenia, informując o postępach w jego rozwiązaniu.

**Kroki:**

1. Serwisant loguje się do aplikacji.
2. Przegląda listę przypisanych zgłoszeń.
3. Wybiera zgłoszenie do aktualizacji.
4. Zmienia status zgłoszenia (np. "W toku", "Rozwiązane").
5. Dodaje notatki dotyczące podjętych działań.
6. Zapisuje zmiany.

**Warunki początkowe:** Serwisant jest zalogowany w systemie i ma przypisane zgłoszenia do obsługi.

**Warunki końcowe:** Status zgłoszenia jest zaktualizowany, a zgłaszający pracownik otrzymuje powiadomienie o zmianach.

#### 4. Przeglądanie zgłoszeń pracownika

**Aktor:** Pracownik

**Opis:** Pracownik przegląda zgłoszenia, które utworzył, aby sprawdzić ich status i notatki dodane przez serwisantów.

**Kroki:**

1. Pracownik loguje się do aplikacji.
2. Przegląda listę swoich zgłoszeń.
3. Wybiera zgłoszenie, które chce sprawdzić.
4. Sprawdza status zgłoszenia i notatki dodane przez serwisantów.

**Warunki początkowe:** Pracownik jest zalogowany w systemie.

**Warunki końcowe:** Pracownik zapoznaje się z aktualnym statusem i notatkami dotyczącymi wybranego zgłoszenia.

#### 5. Generowanie raportu z zgłoszeń

**Aktor:** Menedżer

**Opis:** Menedżer generuje raport z liczby zgłoszeń, czasu realizacji oraz efektywności zespołu serwisowego.

**Kroki:**

1. Menedżer loguje się do aplikacji.
2. Przechodzi do sekcji raportów.
3. Wybiera zakres dat i typ raportu (np. liczba zgłoszeń, czas realizacji).
4. Generuje raport.
5. Przegląda wygenerowany raport i eksportuje go do pliku (np. PDF, Excel).

**Warunki początkowe:** Menedżer jest zalogowany w systemie.

**Warunki końcowe:** Raport jest wygenerowany i gotowy do analizy lub eksportu.

#### 6. Zamknięcie zgłoszenia

**Aktor:** Serwisant

**Opis:** Serwisant zamyka zgłoszenie po rozwiązaniu problemu, a zgłaszający otrzymuje powiadomienie o zamknięciu.

**Kroki:**

1. Serwisant loguje się do aplikacji.
2. Przegląda listę przypisanych zgłoszeń.
3. Wybiera zgłoszenie do zamknięcia.
4. Aktualizuje status na "Zamknięte".
5. Dodaje końcową notatkę opisującą rozwiązanie problemu.
6. Zapisuje zmiany.

**Warunki początkowe:** Serwisant jest zalogowany w systemie i problem został rozwiązany.

**Warunki końcowe:** Zgłoszenie jest zamknięte, a zgłaszający pracownik otrzymuje powiadomienie o zamknięciu zgłoszenia.

#### 7. Zapisanie rozwiązania

**Aktor:** Serwisant

**Opis:** Serwisant zapisuje szczegółowe rozwiązanie problemu w zgłoszeniu, aby udokumentować podjęte kroki i ułatwić przyszłe rozwiązywanie podobnych problemów.

**Kroki:**

1. Serwisant loguje się do aplikacji.
2. Przegląda listę przypisanych zgłoszeń.
3. Wybiera zgłoszenie, które zostało rozwiązane.
4. Przechodzi do sekcji "Rozwiązanie" w szczegółach zgłoszenia.
5. Wprowadza szczegółowy opis podjętych kroków i metod użytych do rozwiązania problemu.
6. Opcjonalnie dodaje załączniki, takie jak zdjęcia, dokumenty lub linki do zasobów pomocniczych.
7. Zapisuje wprowadzone informacje.

**Warunki początkowe:** Serwisant jest zalogowany w systemie i zgłoszenie zostało rozwiązane.

**Warunki końcowe:** Szczegółowe rozwiązanie problemu jest zapisane w systemie i dostępne dla przyszłych odniesień.

## 4.Projekt bazy danych

### Struktura bazy danych

1. **Tabela Users**
   * user\_id (INT, PK, AUTO\_INCREMENT) - Unikalny identyfikator użytkownika
   * username (VARCHAR(50)) - Nazwa użytkownika
   * password (VARCHAR(255)) - Hasło użytkownika (powinno być hashowane)
   * email (VARCHAR(100)) - Adres email użytkownika
   * role (ENUM('employee', 'admin', 'technician', 'manager')) - Rola użytkownika
   * department\_id (INT, FK) - Identyfikator działu, do którego należy użytkownik
2. **Tabela Departments**
   * department\_id (INT, PK, AUTO\_INCREMENT) - Unikalny identyfikator działu
   * department\_name (VARCHAR(100)) - Nazwa działu
3. **Tabela Tickets**
   * ticket\_id (INT, PK, AUTO\_INCREMENT) - Unikalny identyfikator zgłoszenia
   * title (VARCHAR(255)) - Tytuł zgłoszenia
   * description (TEXT) - Opis problemu
   * status (ENUM('new', 'in\_progress', 'resolved', 'closed')) - Status zgłoszenia
   * priority (ENUM('low', 'medium', 'high')) - Priorytet zgłoszenia
   * created\_at (DATETIME) - Data i czas utworzenia zgłoszenia
   * updated\_at (DATETIME) - Data i czas ostatniej aktualizacji zgłoszenia
   * created\_by (INT, FK) - Identyfikator użytkownika, który utworzył zgłoszenie
   * assigned\_to (INT, FK) - Identyfikator użytkownika, któremu przypisano zgłoszenie (opcjonalnie)
   * department\_id (INT, FK) - Identyfikator działu, do którego zgłoszenie jest przypisane
4. **Tabela Comments**
   * comment\_id (INT, PK, AUTO\_INCREMENT) - Unikalny identyfikator komentarza
   * ticket\_id (INT, FK) - Identyfikator zgłoszenia, do którego komentarz należy
   * user\_id (INT, FK) - Identyfikator użytkownika, który dodał komentarz
   * comment (TEXT) - Treść komentarza
   * created\_at (DATETIME) - Data i czas utworzenia komentarza
5. **Tabela Attachments**
   * attachment\_id (INT, PK, AUTO\_INCREMENT) - Unikalny identyfikator załącznika
   * ticket\_id (INT, FK) - Identyfikator zgłoszenia, do którego załącznik należy
   * file\_path (VARCHAR(255)) - Ścieżka do pliku załącznika
   * uploaded\_at (DATETIME) - Data i czas przesłania załącznika

### Relacje między tabelami

* Jeden User może utworzyć wiele Tickets (relacja jeden-do-wielu).
* Jeden User może być przypisany do wielu Tickets (relacja jeden-do-wielu).
* Jeden Ticket może mieć wiele Comments (relacja jeden-do-wielu).
* Jeden Ticket może mieć wiele Attachments (relacja jeden-do-wielu).
* Jeden Department może mieć wielu Users (relacja jeden-do-wielu).
* Jeden Department może mieć wiele Tickets (relacja jeden-do-wielu).

