

Inżynieria wymagań

1. Macierz kompetencji zespołu

Kompetencje	Radosław Brońka	Kinga Surma	Kacper Pabian
Programowanie Python	Posiada	Posiada	Posiada
Technologie Webowe	Posiada	Posiada (podstawy)	Posiada
Bazy danych	Posiada	Posiada (podstawy)	Nie posiada
Znajomość UML	Nie posiada	Posiada	Posiada (podstawy)
Testowanie programowania	Posiada (podstawy)	Posiada (podstawy)	Posiada
Obsługa Gita	Posiada	Posiada	Posiada

2. Pytania oraz odpowiedzi do zmodelowania projektu:

Pytanie	Odpowiedź	Uwagi
Jak przebiega proces autoryzacji i identyfikacji pracowników w systemie?	Skanowanie kodu QR przez pracownika, wykonanie zdjęcia twarzy i weryfikacja w bazie danych za pomocą algorytmów uczenia maszynowego.	
Co ma zapewnić system?	Automatyzacja logowania wejść i wyjść z funkcją weryfikacji tożsamości pracowników za pomocą kodu QR i biometrii	
Jaką zapewniamy skuteczność wykrywania twarzy?	>90% skuteczności.	
Jak ma wyglądać frontend aplikacji?	Widok główny- ekran podglądu kamery podstawowego systemu, powiadomienia o pozytywnej/negatywnej weryfikacji, webowy panel sterowania dla admina.	
Czy ma być generowany aport dla administratora	Tak, okresowo	PDF, format tekstowy
Co ma zawierać raport?	moment wejścia, wyjście, przypadki współdzielenia kart, mismatche twarzy, statystyki czasu prac	

3. Format danych

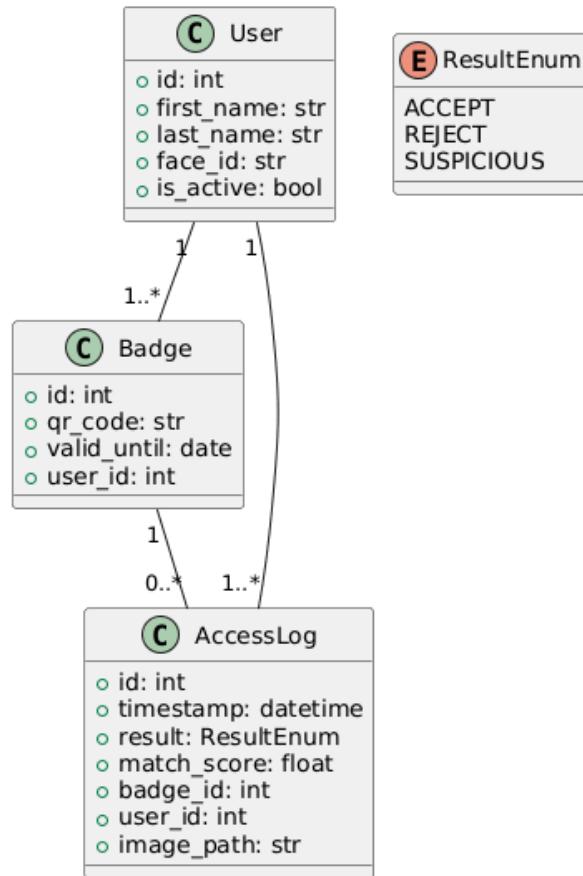
Database (User & Badge)	Input (Skaner & Kamera)	Report (AccessLog)
User id (int) face_id (str) is_active (bool) first_name (str) last_name (str) Badge id (int) qr_code (str) valid_until (date) user_id (int - FK)	Scanner scanned_code (str) Camera captured_image (binary/blob) System current_timestamp (datetime)	timestamp (datetime) - czas zdarzenia result (ResultEnum: ACCEPT, REJECT, SUSPICIOUS) match_score (float) badge_id (int) user_id (int) image_path (str) id (int) - unikalne ID logu

4.1. Modelowanie systemu za pomocą tabeli

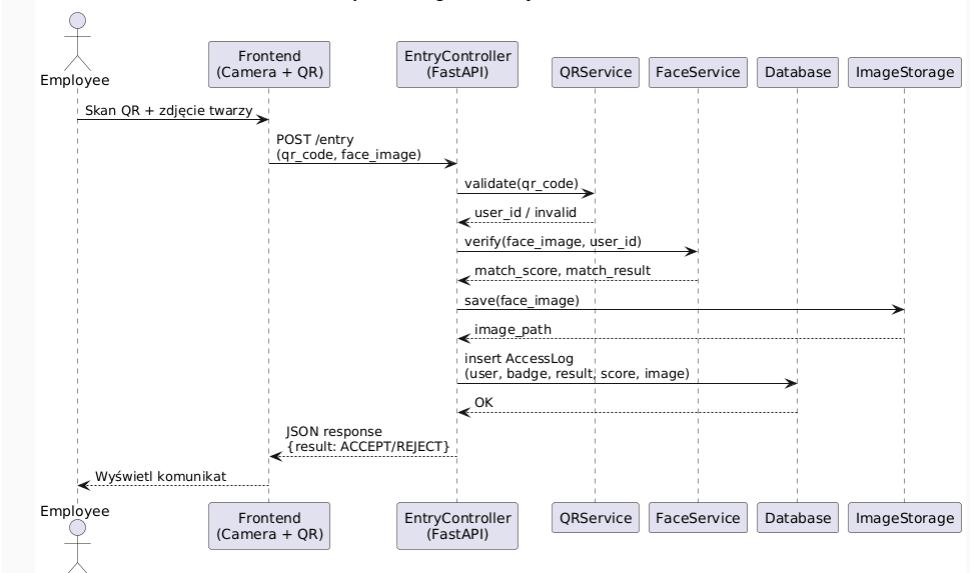
Pole	Treść
Aktorzy	Pracownik (ubiegający się o wejście), Administrator (zarządzający), System (automatyczna bramka)
Opis	Celem działania modelowanego systemu jest bezpieczna weryfikacja tożsamości osoby próbującej wejść na teren fabryki. System łączy weryfikację fizycznej przepustki (QR) z biometrycznym rozpoznawaniem twarzy. Wejście jest autoryzowane tylko wtedy, gdy oba czynniki są zgodne i aktywne.
Dane	Obraz z kamery (klatka wideo/zdjęcie), ciąg znaków z kodu QR, rekord pracownika z bazy danych (wzorzec twarzy, status uprawnień).
Wyzwalacz	Zeskanowanie kodu QR przez czytnik (kamera) i wykrycie twarzy w polu widzenia kamery.
Odpowiedź	Komunikat o otwarciu bramki lub komunikat odmowy dostępu. Raport w systemie: log wejścia zawierający timestamp, ID pracownika, zdjęcie z weryfikacji oraz wynik dopasowania (score).
Uwagi	System musi działać w czasie rzeczywistym. W przypadku niezgodności twarzy z kodem QR (próba użycia cudzej karty) generowany jest specjalny alert bezpieczeństwa.

4.2. Przedstawienie modelowanego systemu za pomocą diagramów UML

Entrance Identification System - Class Diagram



Sequence Diagram - Entry Verification Flow



5. Sugerowany język implementacji: Backend(Python) + Frontend(HTML+ CSS)

Do implementacji backendu systemu wybrano język Python w połączeniu z frameworkiem FastAPI. Python został wybrany ze względu na czytelność kodu oraz szybkość i elastyczność w tworzeniu rozbudowanych aplikacji webowych, co jest szczególnie istotne przy projektach wymagających integracji z systemami przetwarzania obrazów i rozpoznawania twarzy. Istniejące biblioteki Pythona umożliwiają łatwą integrację z usługami rozpoznawania twarzy. FastAPI zapewnia wydajne, nowoczesne REST API, obsługę asynchronicznych operacji oraz automatyczne generowanie dokumentacji, co przyspiesza rozwój i testowanie.

Frontend systemu oparto na standardowych technologiach webowych: HTML, CSS i JavaScript. Takie rozwiązanie umożliwia tworzenie interfejsu użytkownika działającego w przeglądarce internetowej. JavaScript pozwala na bezpośredni dostęp do kamery urządzenia oraz integrację z bibliotekami do skanowania kodów QR, co umożliwia natychmiastową weryfikację tożsamości pracownika. HTML i CSS zapewniają prosty, elastyczny i intuicyjny interfejs administracyjny do zarządzania uprawnieniami, pracownikami i logami zdarzeń.

Całość tworzy system, który zapewnia szybkie i niezawodne działanie w czasie rzeczywistym przy jednoczesnym skupieniu na funkcjonalności systemu z nieskomplikowanym GUI.