

1. Macierz kompetencji zespołu.

Kompetencje	Igor Walasek	Stanisław Ptaszyński	Maciej Kochan
Programowanie C++	Posiada (podstawa)	Posiada (podstawa)	Posiada (podstawa)
Programowanie python	Posiada	Posiada	Posiada
Programowanie java	Posiada	Posiada	Posiada
Programowanie JS	Posiada	Posiada	Posiada
HTML + CSS	Posiada	Posiada	Posiada
SQL	Posiada	Posiada (podstawa)	Posiada (podstawa)
Spring	Nie posiada	Nie posiada	Nie posiada
Spark	Posiada	Posiada	Nie posiada
Flask	Nie posiada	Posiada (podstawa)	Posiada (podstawa)
nodeJS	Posiada	Nie posiada	Nie posiada
Devops	Nie posiada	Nie posiada	Nie posiada
UNIX	Posiada	Posiada (podstawa)	Posiada (podstawa)
Znajomość proxy	Nie posiada	Nie posiada	Nie posiada
Znajomość UML	Posiada (podstawa)	Nie posiada	Posiada (podstawa)
Testowanie oprogramowania	Posiada (podstawa)	Nie posiada	Nie posiada
AI (overall)	Posiada (podstawa)	Posiada (podstawa)	Posiada (podstawa)
Docker	Nie posiada	Nie posiada	Nie posiada
Kubernetes	Nie posiada	Nie posiada	Nie posiada
AWS	Nie posiada	Nie posiada	Nie posiada

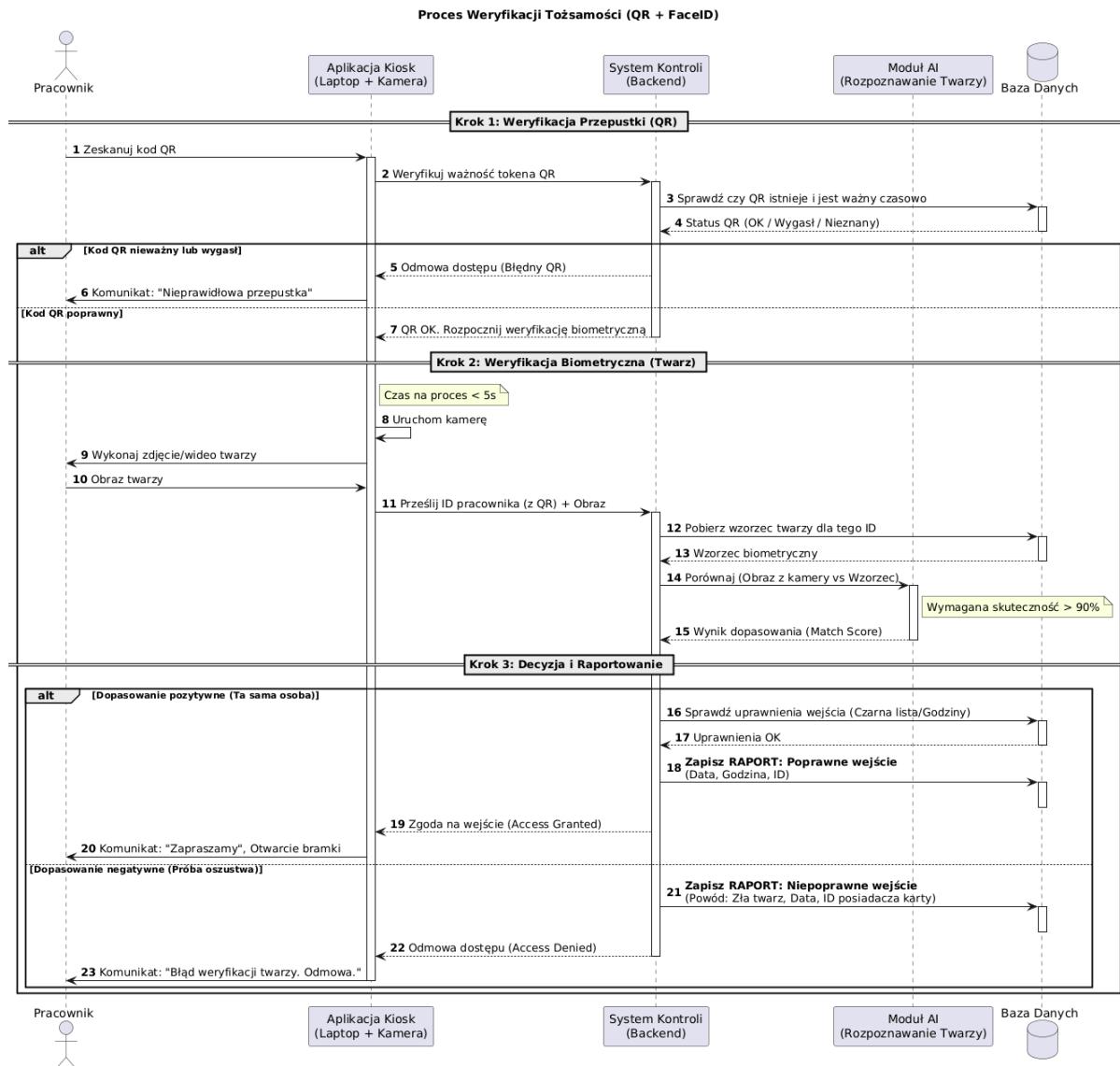
2. Zestaw pytań i odpowiedzi dotyczących projektu.

Pytanie	Odpowiedź	Uwagi
Dostępne zaplecze sprzętowe?	Aplikacja na laptopie, kamera	
Co rozumieć przez bazę twarzy?	Baza min. 20 pracowników i QRCode	
Co rozumiemy przez raportowanie?	Poprawne wejścia i niepoprawne wejścia	poprawne (data, godzina) niepoprawne (niepoprawna "twarz", powód odmowy, data, godzina, id pracownika)
Jak ma działać rozpoznawanie twarzy?	Za pomocą wideo, jednego zdjęcia bądź kilku	
Jak wygląda panel administracyjny?	dodaj pracownika, uzupełnić dane + zdjęcie/wideo, aktualizacja danych, nadać uprawnienia do wejścia, sprawdzić raport dla pracownika, generowanie przepustek	
Ille jest ważny QRCode?	Kilka minut	
Jaki jest dostęp do kamery?	USB/IP(dowolność)	
Czas przetwarzania?	5s	
Skuteczność identyfikacji?	min. 90%	
Przez jaki czas przechowywane są raporty?	6 miesięcy	

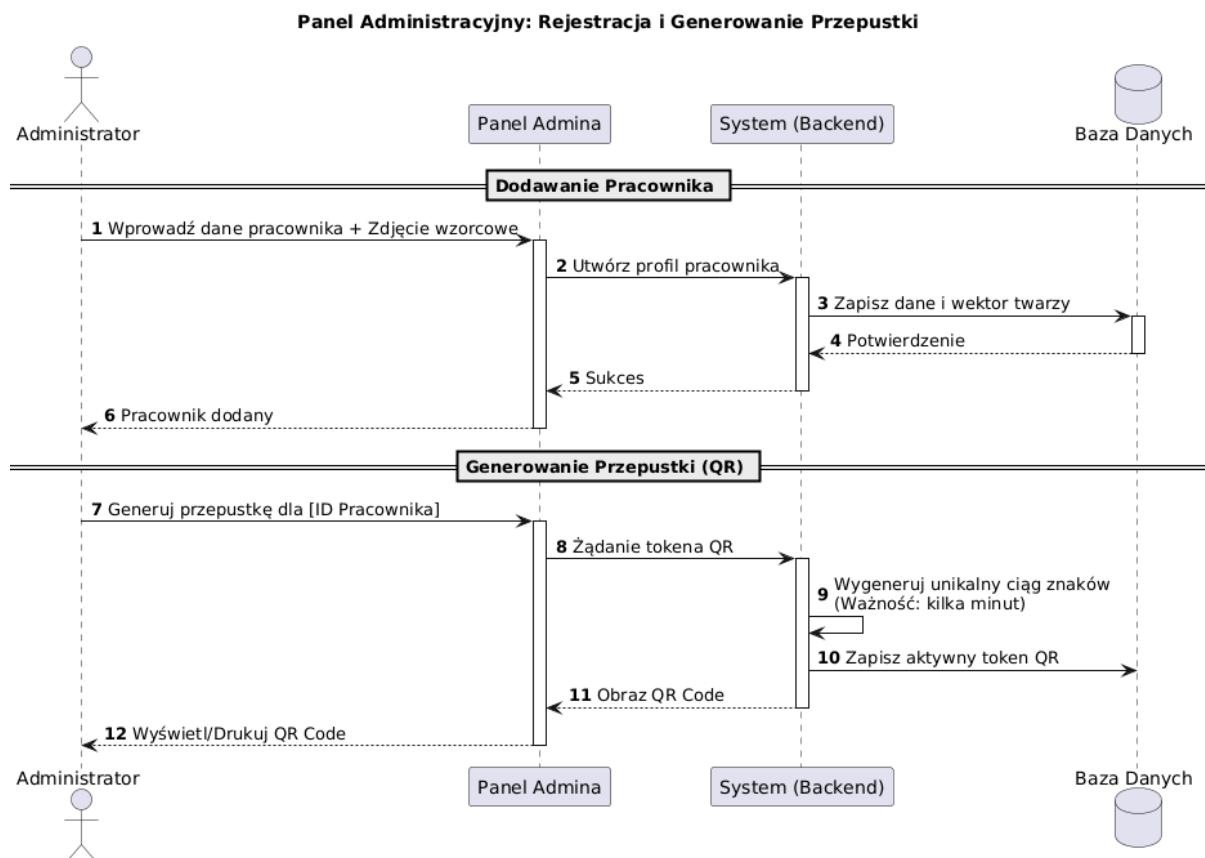
3. Ustalony format danych wejściowych

Users (Baza Pracowników)	Entry Request (Dane z Bramki)	Control – Warunki	Report
id (int/string - unikalne ID pracownika)	qrToken (string - format: QR_{id}_{timestamp})	1. Walidacja QR: Czy token istnieje w bazie i czy nie wygasł?	timestamp
uuid (string - UUID v4, klucz główny)	gatePhoto (binary/blob - plik obrazu .jpg z kamery)	2. Status: Czy flaga blocked == false?	userId (int/string)
name (string - imię i nazwisko)	timestamp (systemowy moment odebrania żądania)	3. Biometria: Czy wektor twarzy z gatePhoto pasuje do photoPath (Python script)?	userName (string)
role (enum: 'admin', 'worker')		4. Rola: Czy użytkownik ma odpowiednie uprawnienia (opcjonalne)?	success (boolean)
photoPath (string - ścieżka do pliku na serwerze)			reason (string - np. "Brak twarzy", "OK")
blocked (boolean - status blokady)			

4. Diagramy sekwencji.

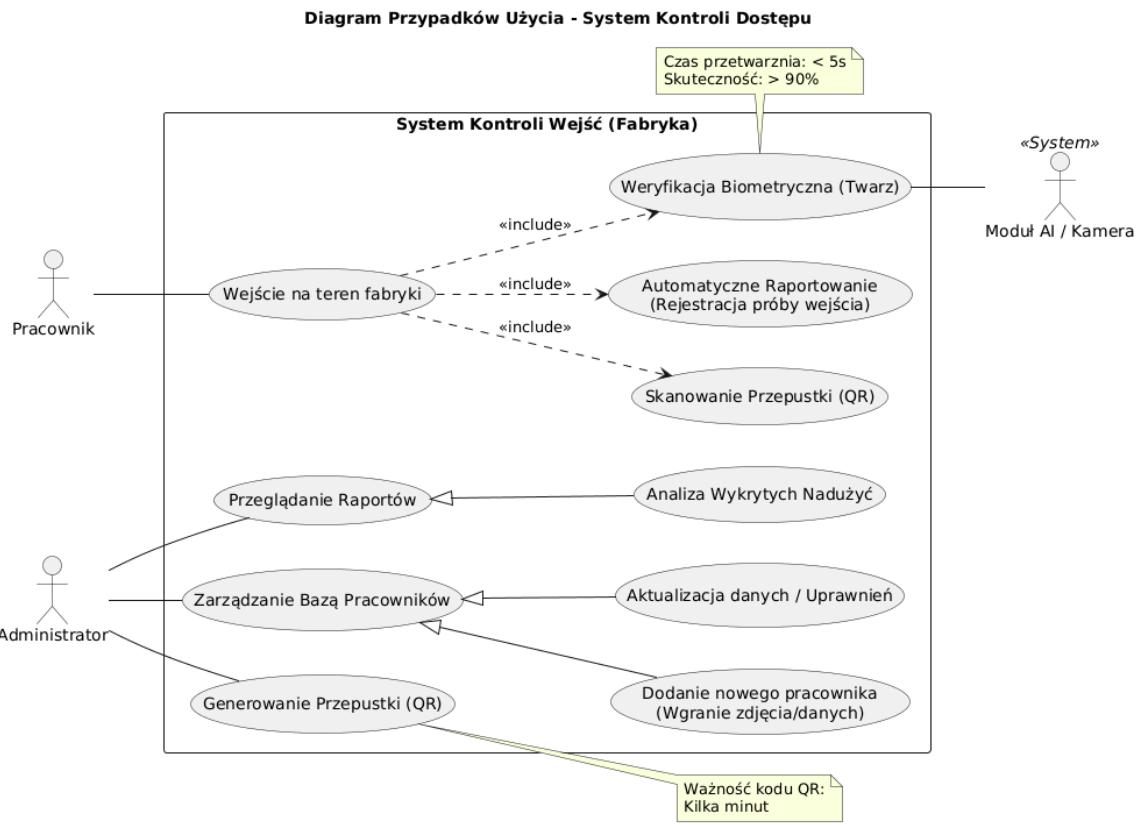


Rys. 1 diagram sekwencji



Rys. 2 pomocniczy diagram sekwencji

5. Przypadki użycia



Rys 3. przypadki użycia aplikacji

6. Opis wybranej technologii.

Jako środowisko uruchomieniowe aplikacji wybrano **Node.js**. Wybór ten minimalizuje ryzyko projektowe poprzez dopasowanie stacku technologicznego do profilu umiejętności zespołu. Ponadto, architektura Node.js zapewnia wydajną interoperacyjność z zewnętrznymi skryptami, co jest krytyczne dla integracji z podsystemem biometrycznym opartym na języku **Python**. Zastosowanie Pythona jest obligatoryjne dla funkcjonalności rozpoznawania twarzy, z uwagi na dostępność zaawansowanych bibliotek z dziedziny Computer Vision.