Dokumentacja inżynierii wymagań

Członkowie zespołu:

- Kacper Ślęzak
- Gabriela Solich
- Martyna Peukert

I. Macierz kompetencji

Kompetencje	Kacper Ślęzak	Gabriela Solich	Martyna Peukert
Python	Posiada	Posiada	Posiada
FastAPI	Posiada	Posiada (podstawowy)	Posiada (podstawowe)
React	Nie posiada	Posiada	Posiada
JavaScript	Posiada (podstawowe)	Posiada (podstawowy)	Posiada
OpenCV	Posiada	Nie posiada	Nie posiada
UI/UX	Nie posiada	Posiada (podstawowe)	Posiada
PostgreSQL	Posiada	Nie posiada	Nie posiada
Docker	Posiada	Posiada (podstawowe)	Nie posiada
Znajomość UML	Posiada (podstawowe)	Posiada (podstawowe)	Posiada (podstawowe)
Git	Posiada	Posiada	Posiada

II. Tabela pytań

Pytanie	Odpowiedź	Uwagi
Jakim sprzętem dysponujemy?	Laptop, kamera USB / IP	Musi to być interfejs webowy na urządzeniu przy wejściu.
Jak to ogólnie ma działać?	Po podaniu QR Code sprawdzamy twarz, czy się zgadza w bazie.	Jeden pracownik nie może przejść na 3 przepustki ze swoją twarzą. Wymagane zapisywanie zarysów twarzy w bazie danych.
Kto będzie użytkownikiem systemu?	Administracja, pracownicy	Musi zostać zrobiony PANEL ADMINISTRACYJNY w celu konfiguracji ustawień i pracowników według zasad firmowych.
Co musi się znajdować w PANELU ADMINISTRACYJNYM?	1) dodać pracownika 2) uzupełnić dane + zdjęcie/wideo 3) aktualizacja danych 4) nadać/zabrać uprawnienia po wejściu ---> QR Code ważny X minut 5) sprawdzić raport dla pracownika 6) generowanie QR code 7) generowanie raportów	
Jaki okres czasu mają być przechowywane logi?	Do zamknięcia firmy	
Jak pracownik ma otrzymać swój kod QR?	Drogę mailowa	Stały kod QR dla każdego pracownika, który dostanie je przez drogę mailową.
Jaki ma być maksymalny czas przetwarzania?	5 sekund	
Jaka ma być trafność identyfikacji?	min 90%	
Jakie raporty mają być generowane?	Poprawnych wejść, a także niepoprawnych.	
Czy wszystkie wymagania klienta są możliwe do spełnienia?	TAK	

III. Ustalony format danych wejściowych

Pracownik	Wejście (Terminal)	Control - warunek	Raport
employee_uid (UUID unique)	timestamp (epoch)	Sprawdź scanned_uid vs employee_uid (czy aktywny)	Timestamp (epoch) - generowany po każdej próbie. employee_id (FK)
is_active (Boolean)	scanned_uid (UUID)	Sprawdź scanned_face vs embedding_vector	status (Enum)
embedding_vector (LargeBinary)	scanned_face (ImageFrame)	Całkowity czas przetwarzania < 5 sekund.	reason (String)

IV. Założenia funkcjonalne i niefunkcjonalne

A. Funkcjonalne
   1. Proces rejestracji i aktywacji
      a) Rejestracja biometryczna twarzy
      b) Generowanie kodów QR
      c) Wydanie kodów QR
      d) Skanowanie kodów QR
      e) Aktywacja / dezaktywacja dostępu dla pracownika przez administratora
   2. Proces weryfikacji
      a) Wyszukiwanie danych z bazy danych
      b) Porównanie wzorców
      c) Decyzja autoryzacyjna
   3. Rejestracja czasu i zdarzeń
      a) Zapis wejść poprawnych
      b) Zapis prób wejść niepoprawnych
      c) Generowanie raportów

B. Niefunkcjonalne
   1. Produktowe
      a) Dostęp 24h na dobę
      b) Uwierzytelnianie przebiega w ciągu 5 s
      c) Dokładność rozpoznania twarzy na poziomie 90%
   2. Organizacyjne
      a) Uwierzytelnianie użytkownika za pomocą kodu QR.
      b) Weryfikacja tożsamości na podstawie zdjęcia twarzy.
   3. Zewnętrzne
      a) Spełnienie wymagań RODO podczas przechowywania danych biometrycznych w bazie danych.

V. Modelowanie systemu za pomocą tabeli

Aktorzy	Administrator (Pracownik HR/IT zarządzający systemem) , Pracownik (Użytkownik końcowy, pracownik fabryki).
Opis	Celem systemu jest zastąpienie podatnego na nadużycia systemu kart magnetycznych. System weryfikuje tożsamość pracownika za pomocą dwuskładnikowego uwierzytelniania (2FA) : 1. Skan kodu QR (coś, co masz) , 2. Weryfikacja twarzy (ktoś, kim jesteś). Ma to na celu wyeliminowanie fałszowania ewidencji czasu pracy
Dane	Baza danych pracowników (dane osobowe, status) , Baza osadzeń biometrycznych (wektory twarzy) , Dziennik zdarzeń (logi wejść/odmów).
Wyzwalacz	Pracownik prezentuje swój indywidualny kod QR do kamery Terminala Dostępowego.
Odpowiedź	Pozytywna: Wyświetlenie komunikatu "Dostęp Przyznany" , zapisanie pomyślnego wejścia w dzienniku zdarzeń. Negatywna: Wyświetlenie komunikatu "Odmowa Dostępu" , zapisanie nieudanej próby (wraz z powodem) w dzienniku.
Uwagi	Kluczowym wymaganiem jest realizacja Weryfikacji (1:1) , a nie Identyfikacji (1:N). System nie odpowiada na pytanie "Kto to jest?", tylko sprawdza "Czy to jest osoba X, za którą się podajesz (na podstawie QR)?

VI. Przedstawienie modelowanego systemu za pomocą diagramów UML

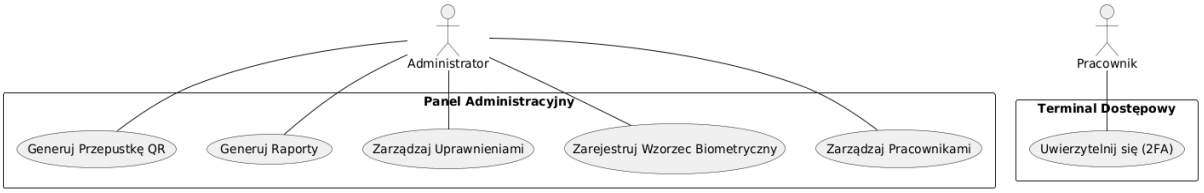


Diagram: Przypadków Użycia

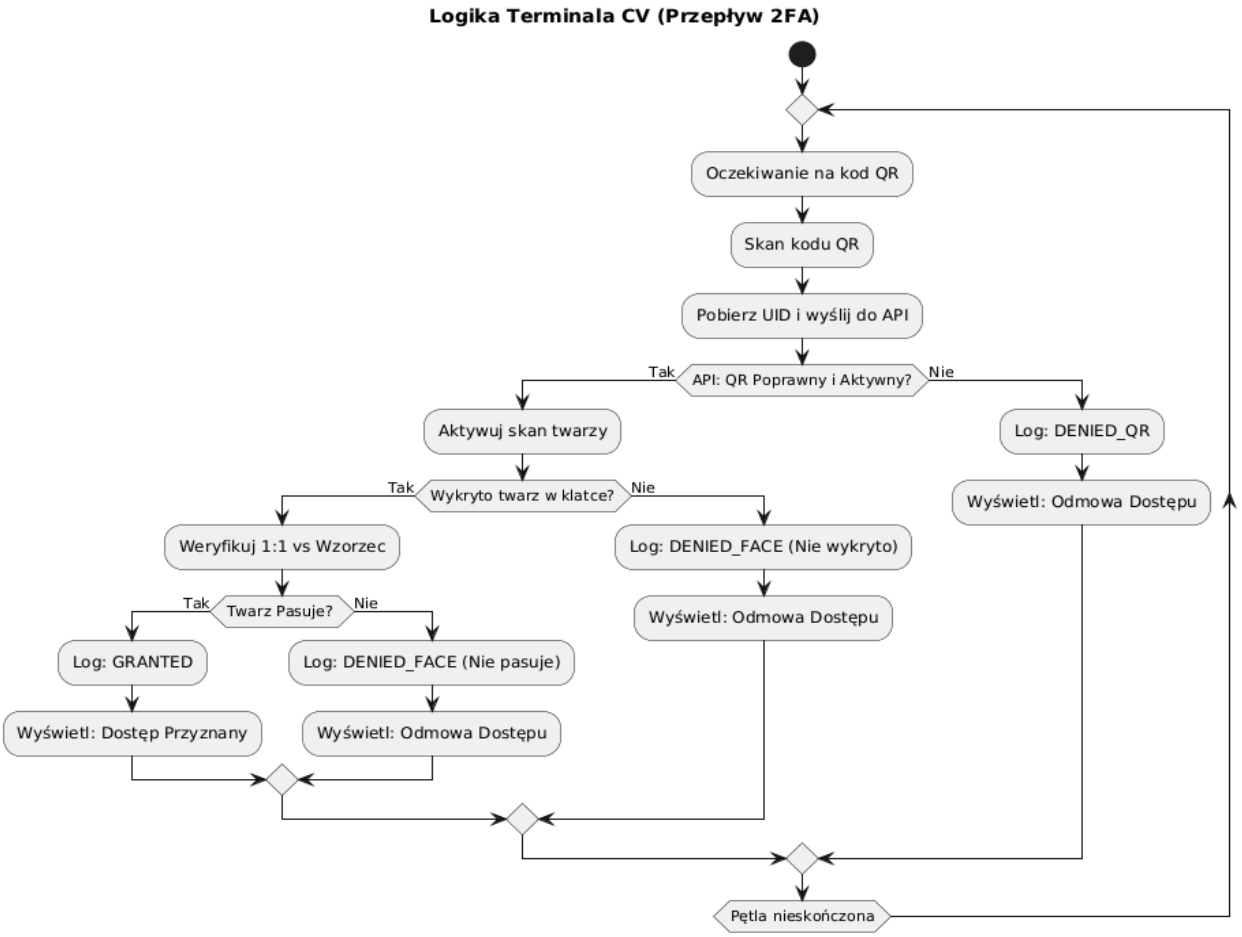


Diagram: Aktywności

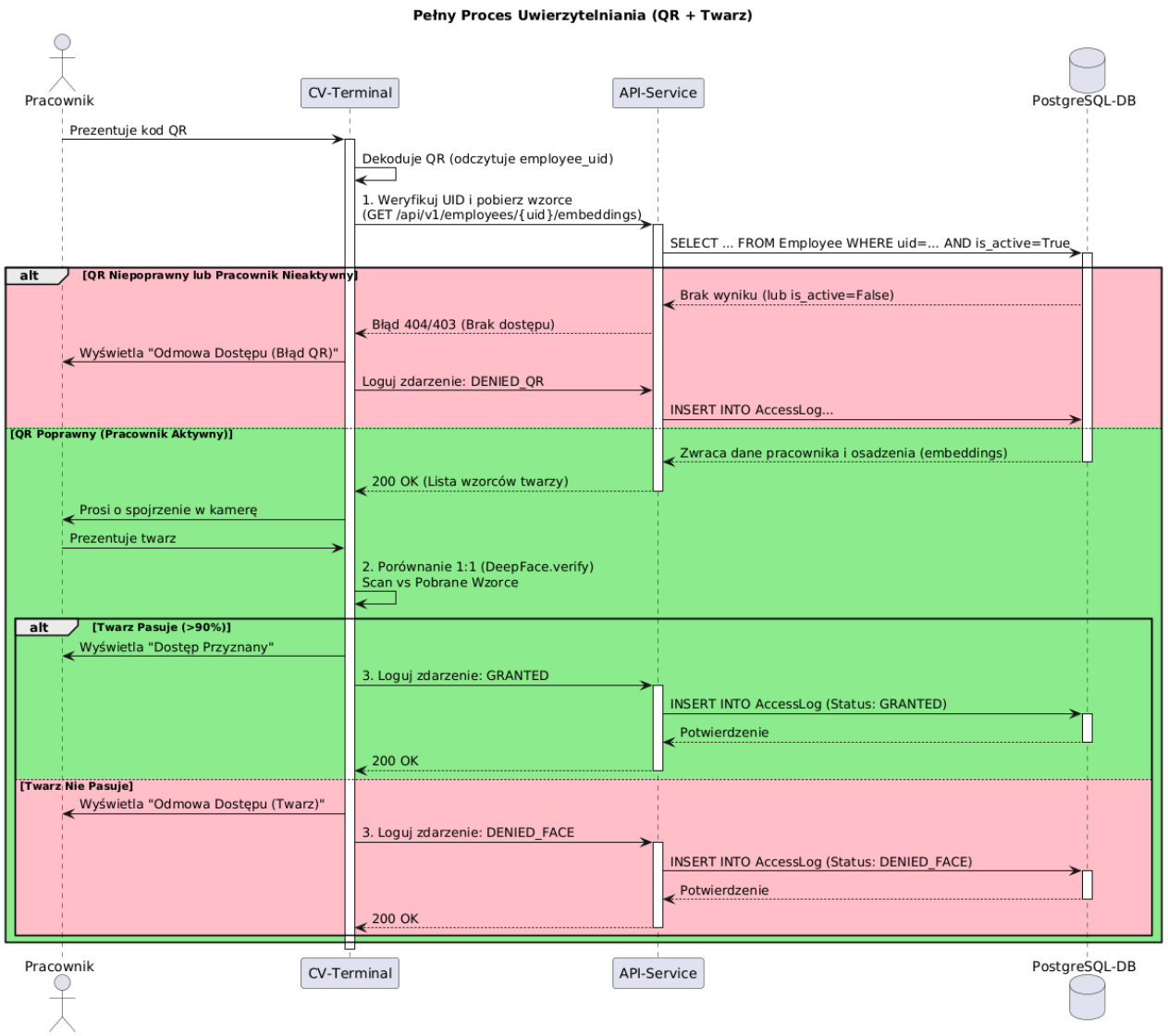


Diagram: Sekwencji

VII. Sugerowane Technologie i Języki Implementacji

- Frontend (FaceOn-Admin-UI): React
- Aplikacja Terminala (FaceOn-CV-Terminal): Python (z biblioteką OpenCV)
- Backend (FaceOn-API-Service): Python (z frameworkiem FastAPI)
- Baza Danych: PostgreSQL
- Model Weryfikacji Biometrycznej

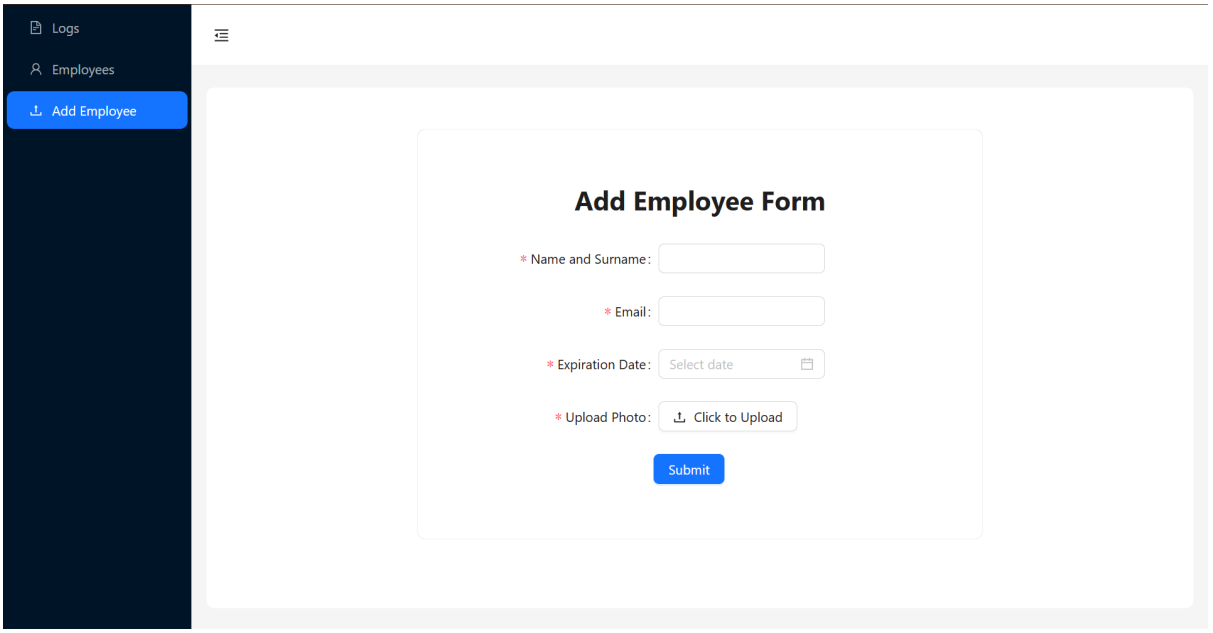
Uzasadnienie:

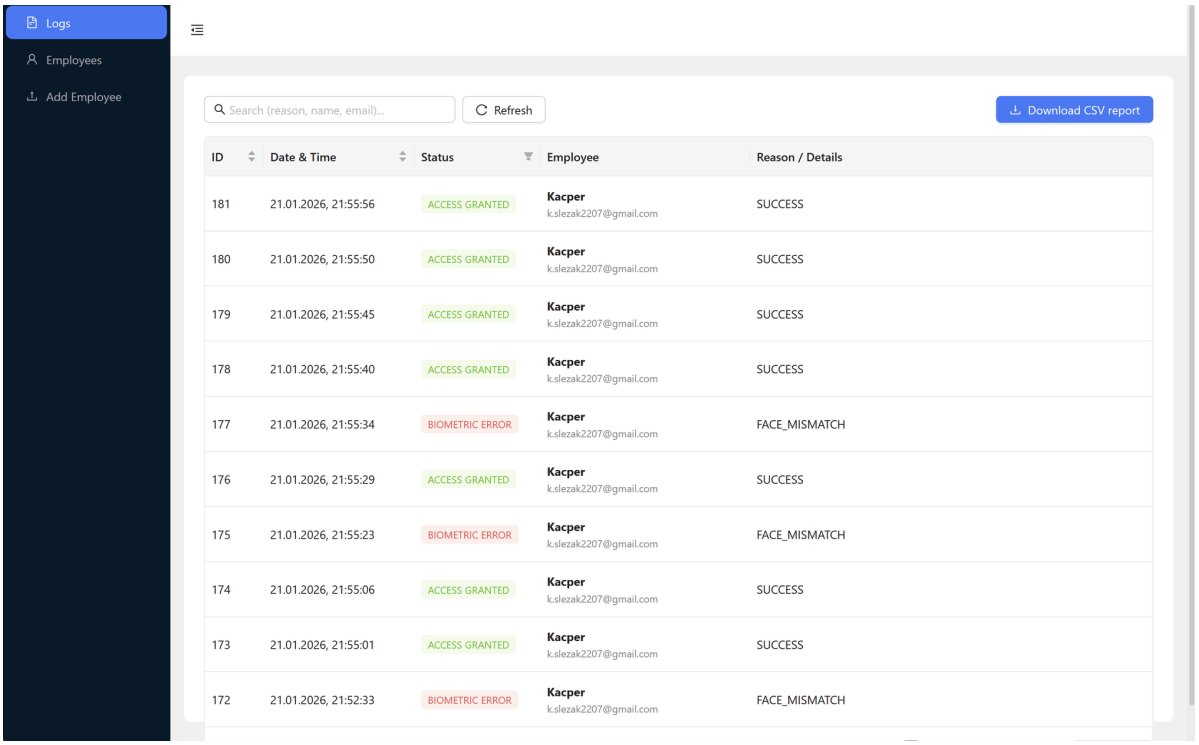
Python (FastAPI, OpenCV): Został wybrany jako główny język dla backendu i aplikacji terminala.

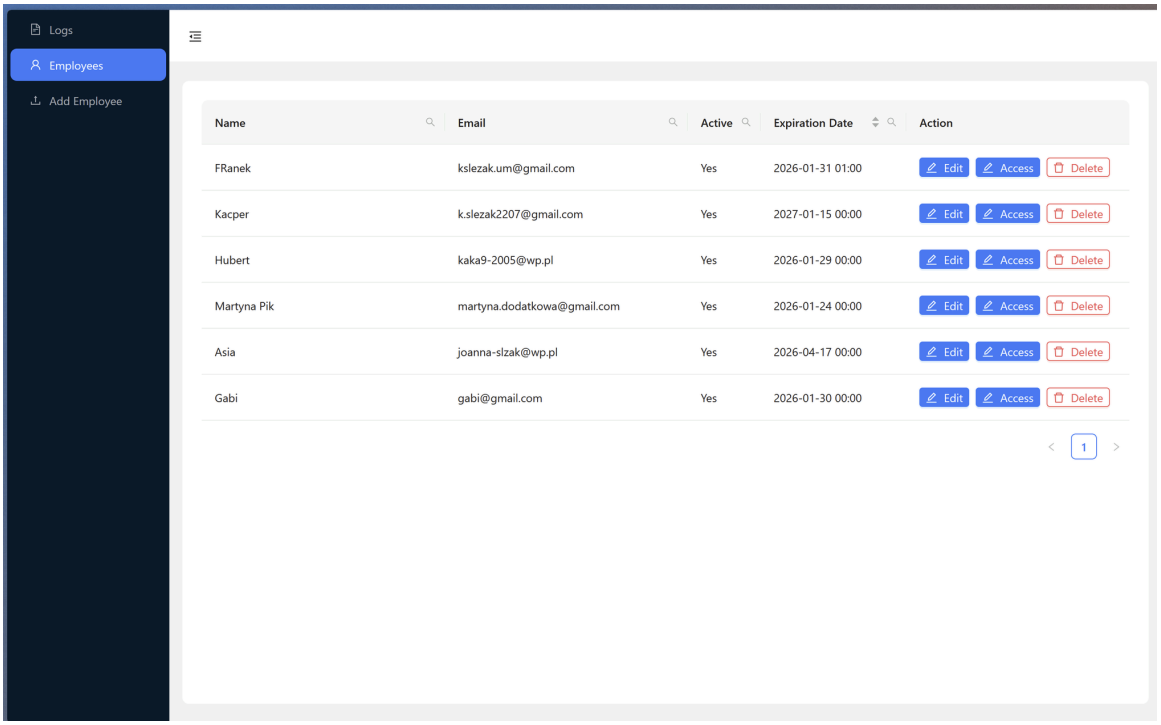
- FastAPI gwarantuje bardzo wysoką wydajność, kluczową dla spełnienia wymagania przetwarzania poniżej 5 sekund. Jego natywne wsparcie dla programowania asynchronicznego idealnie nadaje się do obsługi żądań HTTP i operacji I/O (np. komunikacji z bazą danych).
- React: Został wybrany do budowy interfejsu FaceOn-Admin-UI. Jest to nowoczesna biblioteka, która pozwala na szybkie tworzenie złożonych, responsywnych paneli użytkownika. Umożliwi to sprawne zaimplementowanie wszystkich funkcji administracyjnych, takich jak dodawanie pracowników, generowanie raportów, czy zarządzanie uprawnieniami.
- PostgreSQL: Wybrano jako system zarządzania bazą danych. Jest to potężne, otwarte i niezawodne rozwiązanie SQL. Oferuje zaawansowane możliwości, takie jak obsługa typów danych (np. UUID, Boolean) oraz potencjalnie rozszerzeń do przechowywania i indeksowania wektorów biometrycznych (embedding_vector), co jest kluczowe dla naszego systemu.

VIII. Minimum Viable Product (MVP)

Celem MVP systemu FaceOn było stworzenie działającego prototypu systemu kontroli dostępu opartego na uwierzytelnianiu dwuskładnikowym (2FA): kod QR oraz biometria twarzy, zastępującego tradycyjne karty magnetyczne.







Panel Administracyjny

- **Zarządzanie Pracownikami:** Zaimplementowano pełny CRUD (Create, Read, Update, Delete). Na załączonym screenie widać listę pracowników z polami takimi jak "Name", "Email", "Active" oraz datą wygaśnięcia uprawnień.
- **Wgląd w status:** Administrator widzi, czy pracownik jest aktywny ("Active: Yes") i do kiedy ma ważną przepustkę, co realizuje wymaganie nadawania/odbierania uprawnień.
- **Logi i Raportowanie:** Zgodnie z wymaganiem generowania raportów, system rejestruje zdarzenia. Screen z logami pokazuje dokładną datę, status ("ACCESS GRANTED", "BIOMETRIC ERROR") oraz powód decyzji ("SUCCESS", "FACE_MISMATCH").
- **Eksport Danych:** Zaimplementowano funkcję "Download CSV report", co pozwala na zewnętrzną analizę danych (widoczny przycisk na screenie 2).

1. Logika Terminala i Weryfikacja

- **Proces 2FA:** Logi potwierdzają działanie algorytmu opisanego w diagramie aktywności. System rozróżnia sytuacje, gdzie kod QR jest poprawny, ale twarz nie pasuje ("FACE_MISMATCH"), od pełnego sukcesu.
- **Rejestracja zdarzeń:** Każda próba wejścia (udana lub nie) jest odkładana w bazie danych, co realizuje wymaganie zapisu wejść poprawnych i niepoprawnych.

2. Zgodność z Wymaganiami Niefunkcjonalnymi - *osiągnięte cele:*

- **Czas przetwarzania:** < 5 sekund
- **Trafność identyfikacji:** min. 90% trafności
- **Dostęp:** 24h na dobę
- **Bezpieczeństwo danych pracowników:** Spełnienie wymagań podczas przechowywania danych biometrycznych w bazie danych.

3. Architektura i Technologie

- **Frontend:** React
- **Backend:** Python + FastAPI
- **Baza Danych:** PostgreSQL

4. **Weryfikacja scenariuszy użycia na podstawie logów systemowych**

Zaimplementowany mechanizm logowania potwierdza poprawność działania logiki biznesowej dla obu kluczowych ścieżek przepływu sterowania:

- **Ścieżka poprawna:** Wpis o godzinie 21:55:56 potwierdza pomyślną weryfikację użytkownika 'Kacper'. System poprawnie zinterpretował zgodność kodu QR oraz biometrii, nadając status ACCESS GRANTED z powodem SUCCESS.
- Ścieżka odrzucenia: Wpis o godzinie 21:52:33 demonstruje skuteczność zabezpieczenia przed nieautoryzowanym użyciem przepustki. Mimo prezentacji poprawnego kodu QR, system wykrył niezgodność biometryczną, generując status BIOMETRIC ERROR z kodem błędu FACE_MISMATCH. Dowodzi to realizacji wymagania dotyczącego eliminacji fałszowania ewidencji.