Зміст

[**1** **ЗАГАЛЬНІ ПОЛОЖЕННЯ** 4](#_Toc161948029)

[1.1 Призначення ТЗ 4](#_Toc161948030)

[1.2 Підстава для виконання робіт 4](#_Toc161948031)

[1.3 Контактні дані Замовника та Виконавця робіт зі створення ТЗ 4](#_Toc161948032)

[1.3.1 Замовник 4](#_Toc161948033)

[1.3.2 Виконавець робіт зі створення ТЗ 4](#_Toc161948034)

[1.4 Планові терміни початку та закінчення робіт зі створення ТЗ 4](#_Toc161948035)

[**2** **ПРИЗНАЧЕННЯ ТА МЕТА СТВОРЕННЯ Системи** 5](#_Toc161948036)

[2.1 Призначення Системи 5](#_Toc161948037)

[2.2 Мета створення Системи 5](#_Toc161948038)

[2.3 Переваги у результаті впровадження Системи 5](#_Toc161948039)

[**3** **ВИМОГИ ДО Системи** 6](#_Toc161948040)

[3.1 Вимоги до Системи в цілому 6](#_Toc161948041)

[3.1.1 Вимоги до структури та функціонування систем, перелік підсистем 7](#_Toc161948042)

[3.1.2 Не функціональні вимоги 7](#_Toc161948043)

[3.1.3 Вимоги до інтеграції 8](#_Toc161948044)

## Аналіз технологій, інструментів, програмних бібліотек та математичних моделей

##### **Мова** програмування

Мова програмування Python вибрана для цього проекту з кількох причин. По-перше, Python має простий і лаконічний синтаксис, що робить розробку швидшою та ефективнішою, особливо на етапі прототипування і розвитку програми. Вона також має велику кількість бібліотек і фреймворків, що спрощують роботу з обробкою зображень (для модуля розпізнавання обличчя), криптографією (для шифрування файлів), та роботою з базами даних.

Другою причиною є широке сприйняття Python у сфері машинного навчання та обробки зображень. Використання Python для модуля розпізнавання обличчя дозволяє легко інтегрувати сторонні бібліотеки та фреймворки, такі як OpenCV або TensorFlow, що полегшує реалізацію функціональності ідентифікації особистості.

Нарешті, Python є більш гнучкою мовою, що сприяє швидкому розвитку і змінам у проекті, що може бути важливим у випадку постійних оновлень та вдосконалень в системі без значного збоку часу і зусиль. У порівнянні з C++ або Java, Python зазвичай потребує менше коду для досягнення тих самих результатів, що полегшує процес розробки та збільшує продуктивність команди розробників.

##### Бібліотеки та інструменти

* OpenCV: Бібліотека OpenCV (Open Source Computer Vision Library) надає широкий набір інструментів для роботи з зображеннями та відео, що включає обробку зображень, відстеження облич та багато іншого.
* face\_recognition: Ця бібліотека реалізує простий інтерфейс для обробки зображень та визначення облич людей на них. Вона базується на бібліотеці dlib, яка використовує машинне навчання для розпізнавання облич.
* SQLAlchemy: Це потужний ORM (Object-Relational Mapping) для роботи з базами даних у Python. Ви можете використовувати SQLAlchemy для спрощення взаємодії з базою даних PostgreSQL.»;

##### Шифрування

- cryptography: Це бібліотека для криптографічних операцій у Python. Вона надає реалізації різних алгоритмів шифрування, включаючи AES, який може бути використаний для шифрування файлів у вашому проекті.

##### Графічний інтерфейс

- PyQt6: PyQt є набором Python-зв'язків для Qt, який надає можливості для розробки крос-платформених графічних інтерфейсів. Ви можете використовувати PyQt6 для створення зручного інтерфейсу користувача для вашого проекту.

* 1. **Сервер та база даних**

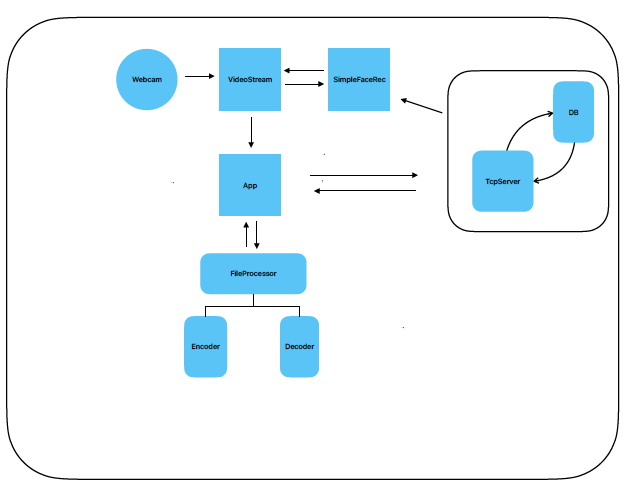
- Docker Compose: Використання Docker Compose для контейнеризації вашого додатку разом із сервером та базою даних дозволить легко розгорнути та керувати середовищем вашого проекту.

- PostgreSQL: Це потужна об'єктно-реляційна система керування базами даних, яка забезпечить надійне зберігання та управління даними вашого проекту.

## Структурна схема реалізації проекту

На данному етапі розробки програми розгляається дві структури проекту. Одна з використанням стандартного функціоналу OpevCV для заходлення відео та окремого потоку для детекторів. Або реалізація за допомогою сучасних методів буферизації та конвеєризації відео потоку за допомогою GStreamer Pipeline, що дозволить роздвоювати відео потік без втрати частоти зображень в секунду.

##### Структурна схема реалізації за допомогою функціоналу OpenCV



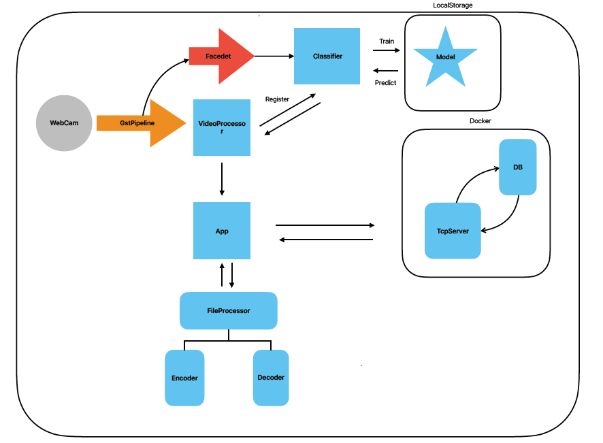
##### Переваги реалізації за допомогою функціоналу OpenCV

* Легка реалізація: Використання OpenCV у сполученні з Python дозволяє легко звертатися до відеопотоку або файлів відео за допомогою методу VideoCapture(). Це дає змогу швидко та ефективно отримувати кадри з відеопотоку або відеофайлів для подальшої обробки. Такий простий доступ до відеофайлів допомагає значно спростити реалізацію системи розпізнавання обличчя.
* Надійне розпізнавання обличчя за допомогою бібліотеки face\_recognition: Бібліотека face\_recognition - це потужний і простий у використанні інструмент для розпізнавання облич. Вона пропонує зручний API для роботи з обличчями, яке включає в себе функції розпізнавання та ідентифікації. Ця бібліотека побудована на базі dlib та має високу точність розпізнавання, що робить її ідеальним вибором для проектів, пов'язаних з ідентифікацією облич.
* Не потрібно тренувати модель та зберігати її на компютері: Однією з ключових переваг бібліотеки face\_recognition є те, що вона використовує готові моделі для розпізнавання облич. Це означає, що користувачеві не потрібно тренувати власні моделі або зберігати великі обсяги даних на своєму комп'ютері. Вона просто використовується для обробки вхідних зображень і забезпечує точні результати без необхідності великого обсягу обчислень або об'ємних даних. Це значно спрощує розробку та підтримку системи розпізнавання облич, особливо для проектів з обмеженими ресурсами.

##### Недоліки реалізації за допомогою функціоналу OpenCV

* Не стабільний FPS при використанні детекторів через копіювання або розділення відео потоку: Під час використання Python для обробки відеопотоку за допомогою бібліотек, таких як OpenCV, може виникати проблема зі стабільністю кадрів на секунду (FPS). Особливо це стає проблемою при використанні детекторів облич з великою кількістю обчислень. Це може призвести до нестабільної роботи системи або втрати частини кадрів, що може вплинути на точність розпізнавання та взаємодію з користувачем.
* Потрібно передавати закодовані зображення на сервер та зберігати їх в базу: При використанні Python для розпізнавання облич у відеопотоці, для подальшої обробки або аналізу знадобиться передача оброблених зображень на сервер та збереження їх в базу даних. Додатково, необхідно обробляти та зберігати дані у форматі, який дозволяє ефективне зберігання та швидкий доступ до них, що може вимагати додаткових ресурсів та зусиль для реалізації і підтримки.

##### Структурна схема реалізації за допомогою GStreamer Pipeline



##### Переваги реалізації за допомогою GStreame Pipeline

* Висока якість та стабільність потоку: Використання GStreamer Pipeline забезпечує високу якість та стабільність відеопотоку. Ця платформа оптимізована для роботи з мультимедійними данними, що дозволяє підтримувати високу швидкодію та якість відтворення.Також це сучасне рішення що використовується для транслювання/обробки/рендерингу відеопотоку у провідних компаніях світу.
* Можливітсь легко змінювати джерело потоку: GStreamer Pipeline дозволяє легко змінювати джерело вхідного відеопотоку. Це дозволяє підключати різні джерела відео (наприклад, веб-камери, відеофайли, IP-камери) без значних змін у коді.
* Легка можливість маштабування та заміни модулів для обробки відео: GStreamer Pipeline має гнучку архітектуру, яка дозволяє легко маштабувати систему та замінювати модулі для обробки відео в процесі розвитку проекту. Це дозволяє швидко внести зміни в систему та адаптувати її до нових потреб або вимог користувачів.
* Не потрібно робити запит в базу, не зберігається особиста інформація в базі:

У разі використання GSreamer ми зможемо відмовитись від викоирстання face\_recogniton та отримувати кропси з лице за допомогою GStreamer та в подальшому натренувати модель за допомогою OpenCV, що дозволить не загружати зображення в базу та одразу мати доступ до моделі.

##### Недоліки реалізації за допомогою GStreamer Pipeline

* Натренована модель зможе показувати лише число схожості, а не розпізнавання: Одним з недоліків використання GStreamer Pipeline є те, що натренована модель для розпізнавання облич може показувати лише число схожості між обличчям у відеопотоці та обличчями збереженими при тренуванні, а не саме ім'я або ідентифікатор особи. Це може вимагати додаткового коду для обробки результатів та визначення конкретних осіб на основі отриманих числових значень схожості.
* Час на тренування власної моделі: Ще одним недоліком є те, що для реалізації системи розпізнавання облич за допомогою GStreamer Pipeline може бути необхідний час на тренування власної моделі. Це може бути часомоємким процесом, особливо якщо потрібно натренувати модель на слабкому компютері.
* .Зберігання моделі: Недоліком є необхідність зберігання натренованої моделі для подальшого використання. Це може вимагати додаткового простору на диску та організації процесу зберігання та оновлення моделі. Також може виникати проблема з підтримкою та сумісністю версій моделі, що може ускладнити розвиток та підтримку системи в майбутньому.

## Інтерфейс роботи програми

##### Інтерфейс екрану для регістрації обличчя

