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)

**ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ**

У цьому документі вжито наступні позначення та скорочення:

**API** (англ. Application Programming Interface) – інтерфейс прикладного програмування

**ETL**(англ. Extract, Transform, and Load) – процес у складуванні (зберіганні) даних, який охоплює Виймання даних із зовнішніх джерел, Перетворення даних та їх Завантаження даних у кінцеве сховище

**IDEF0** – методологія функціонального моделювання (Function modeling) і графічна нотація, призначена для формалізації і опису бізнес-процесів

**Prozorro** – пілотний проект електронної системи публічних закупівель

**SQL** (англ. Structured Query Language)  – мова структурованих запитів

**АІС**– автоматизована інформаційна система

**АПК** – апаратно-програмний комплекс

**АТК** - апаратно-технічний комплекс

**АРМ** – автоматизоване робоче місце

**БД** – база даних

**БДР** – безпека дорожнього руху

**ГО** – громадська організація

**ЕК** – електронна картка

**Система** – система «Прозоре будівництво» та публічний портал CoST

**Комплект ЗІП** – комплект запасних частин та приладів

**КСЗІ** – комплексна система захисту інформації

**МНІ** – магнітний носій інформації

**ППР** – планово-попереджувальні роботи

**САД** – Служба автомобільних доріг

**СЕВ ОВВ** – система електронної взаємодії органів виконавчої влади

**СКБД** – система керування базою даних

**СПЗ** – спеціалізоване програмне забезпечення

**ТЗ** – технічне завдання

# **ЗАГАЛЬНІ ПОЛОЖЕННЯ**

## Призначення ТЗ

Розробити програмне забезпечення, яке забезпечить безпеку операційної системи шляхом шифрування конфіденційних файлів і дозволить розшифровувати їх лише після успішного підтвердження особистості користувача за допомогою розпізнавання обличчя.

## Підстава для виконання робіт

Роботи з розроблення ТЗ здійснюються на підставі виконання курсової роботи з дисципліни: ”Системне програмне забезпечення”.

## Контактні дані Замовника та Виконавця робіт зі створення ТЗ

### Замовник

Назва: НУ «Львівська Політехніка »

Адреса: вулиця Степана Бандери, 12, Львів, Львівська область, 79000

Код ЄДРПОУ: 02071010

Телефон: (032) 258-25-37

### Виконавець робіт зі створення ТЗ

Назва: ФОП Герега Ростислав Олегович

Код ЄДРПОУ: 23735402

Адреса: 79040, м. Львів, вул. Люблінська, 4

Телефон: (098) 118 49 40

## Планові терміни початку та закінчення робіт зі створення ТЗ

Дата початку робіт: 1 заняття

Дата закінчення робіт: 8 заняття

# **ПРИЗНАЧЕННЯ ТА МЕТА СТВОРЕННЯ Системи**

## Призначення Системи

Система призначена для:

Розробки програмного забезпечення, яке забезпечить безпеку операційної системи шляхом шифрування конфіденційних файлів і дозволить розшифровувати їх лише після успішного підтвердження особистості користувача за допомогою розпізнавання обличчя. Це програмне забезпечення буде забезпечувати високий рівень захисту для конфіденційної інформації, використовуючи передові технології шифрування та системи розпізнавання обличчя. Таким чином, тільки авторизованим користувачам буде дозволено отримати доступ до зашифрованих файлів, що гарантує високий рівень безпеки для даних.

.

## Мета створення Системи

Метою створення Системи, яка забезпечить безпеку операційної системи шляхом шифрування конфіденційних файлів і дозволить розшифровувати їх лише після успішного підтвердження особистості користувача за допомогою розпізнавання обличчя.

## Переваги у результаті впровадження Системи

В результаті створення Системи планується досягти:

* Інтегроване розпізнавання облич що дає можливість підтвердження особистості користувача за допомогою модуля face\_recognition. Це може забезпечити додатковий рівень безпеки, оскільки доступ до розшифрованих файлів буде надано лише після успішного підтвердження особистості.
* Широкий спектр можливостей може які можуть поєднувати в собі не тільки шифрування файлів, але і підтвердження особистості через обробку облич. Це робить його більш універсальним та гнучким у використанні.
* Відкритість та розширюваність за допомогою можливості вільно розширювати його функціональність та вдосконалювати, виходячи зі специфічних потреб аудиторії.
* Контроль над безпекою дає повний контроль над захистом даних. Необхідно вибрати найбільш підходящі методи шифрування для забезпечення безпеки, що дозволяє створити більш гнучку та індивідуальну систему.

# **ВИМОГИ ДО Системи**

## Вимоги до Системи в цілому

Система має включати функціональні вимоги, що дозволяють користувачеві вибирати файли для шифрування та розшифрування через інтуїтивний графічний інтерфейс, а також забезпечувати можливість реєстрації обличчя для подальшої ідентифікації. Також важливо, щоб система використовувала симетричне шифрування з AES для захисту конфіденційних файлів і дозволяла їх розшифрування лише після успішної ідентифікації користувача за допомогою розпізнавання обличчя.

У нефункціональних вимогах важливо враховувати надійність системи, її швидкодію та ефективність, а також інтуїтивно зрозумілий графічний інтерфейс користувача, що сприяє зручній взаємодії з програмою. Додатково, система повинна бути стійкою до помилок та збоїв, щоб забезпечити безперебійну роботу.

Щодо інтеграційних вимог, система має бути реалізована мовою програмування Python та використовувати модуль face\_recognition для ідентифікації користувача. Також важливо використовувати Docker Compose для зручного підняття TCP сервера для взаємодії з базою даних PostgreSQL. Для забезпечення багатопоточності роботи програми система повинна використовувати бібліотеку threading.

У вимогах до тестування необхідно провести ретельне тестування системи на різних тестових наборах даних, а також виконати тести безпеки для перевірки стійкості системи до можливих атак. Тестування допоможе переконатися в надійності та безпеки розробленої системи.

### Вимоги до структури та функціонування систем, перелік підсистем

До складу FaceLock, повинні входити наступні функціональні підсистеми:

* Підсистема «Вхід за допомогою розпізнавання обличчя»;
* Підсистема «Вхід за допомогою паролю»;
* Підсистема «Шифрування»;
* Підсистема «Розшифрування»;
* Підсистема «Управлінням акаунтом»;
* Підсистема «Управління зашифрованими файлами»;

Користувачам системи надається можливість вибору файлів для шифрування або розшифрування, що забезпечується шляхом інтуїтивного графічного інтерфейсу користувача. Додатково, користувачам дозволяється зареєструвати своє обличчя для подальшої ідентифікації, що сприяє підвищенню рівня безпеки доступу до системи. Система забезпечує безпеку файлів шляхом застосування симетричного шифрування з використанням стандарту AES, що гарантує надійний захист конфіденційної інформації. Після успішної ідентифікації особистості користувача за допомогою модуля face\_recognition, система дозволяє розшифрувати обрані файли, вимагаючи введення вірного ключа, тим самим забезпечуючи доступ лише авторизованим користувачам до конфіденційної інформації.

### Не функціональні вимоги

Система має забезпечувати високий рівень безпеки, зокрема шляхом надійного шифрування та захищеного зберігання ключів. Крім того, вимагається точність модуля розпізнавання обличчя в ідентифікації користувача для забезпечення надійності системи. Програма повинна працювати ефективно та швидко, а графічний інтерфейс користувача має бути інтуїтивно зрозумілим для зручної взаємодії з програмою. Крім того, важливо, щоб програмне забезпечення було стійким до помилок та збоїв, щоб забезпечити безперебійну роботу і надійність в процесі його використання.

### Вимоги до інтеграції

* Програмне забезпечення повинно бути реалізоване мовою програмування Python.
* Модуль face\_recognition повинен бути інтегрований у систему для проведення процесу ідентифікації користувача.
* Використання TCP сервера, який піднімається в Docker Compose для зручної роботи з PostgreSQL.
* Сервер використовує бібліотеку threading для забезпечення багатопоточності роботи програми.
* Робота з базою даних здійснюється за допомогою ORM.