# 1 ВСТУП

## 1.1 Огляд продукту

Програмна система призначена для автоматизації процесів керування сервісом розкрою листових матеріалів. Вона орієнтована на малий та середній бізнес, який надає послуги з розкрою за індивідуальними кресленнями клієнтів. Продукт охоплює повний цикл: від прийому замовлення із завантаженням креслень у форматі DXF — до планування розміщення моделей на аркуші та формування виробничих файлів.

Система реалізована як веб-орієнтоване клієнт-серверне рішення. Клієнтська частина створена за допомогою Blazor WebAssembly, що забезпечує зручний, інтерактивний інтерфейс користувача. Серверна частина — на базі ASP.NET Core Web API — відповідає за обробку бізнес-логіки, взаємодію з базою даних PostgreSQL 15 та управління збереженням файлів. Для структурування запитів і розширення системи застосовано патерн CQRS у поєднанні з бібліотекою MediatR, що покращує підтримуваність та тестованість коду. Використання Entity Framework Core забезпечує ефективну роботу з базою даних, а реалізація власного механізму конвертації DXF у SVG дозволяє інтегрувати перегляд креслень без використання платних сторонніх бібліотек.

Таким чином, система поєднує керування замовленнями, адміністрування ресурсів, інтерактивне планування розкрою та підготовку виробничих файлів у єдиній платформі, що дозволяє підвищити ефективність і прозорість виробничих процесів.

## 1.2 Мета

Метою створення програмної системи є автоматизація процесів обробки замовлень на розкрій листових матеріалів, підвищення ефективності та точності планування розміщення моделей на аркуші, а також зменшення витрат часу і ресурсів, пов’язаних із ручною обробкою креслень та координацією між працівниками.

## 1.3 Межі

Розроблювана програмна система призначена виключно для автоматизації процесів обробки замовлень на розкрій листових матеріалів у межах сервісного підприємства або виробничого підрозділу. Система не охоплює фінансовий облік, логістику, управління персоналом або інші підсистеми підприємства, що не належать безпосередньо до задач розкрою.

Функціональність системи обмежується роботою з векторними кресленнями у форматі DXF.

Початкова реалізація не включає модуль автоматичної оптимізації розміщення моделей на листі, а забезпечує лише ручне планування через інтерактивний редактор. Також система орієнтована на локальне файлове сховище для креслень, із перспективою переходу на хмарні рішення у наступних релізах.

Таким чином, межі системи визначені функціональністю, необхідною для прийому замовлень, їх планування, візуалізації, формування файлів та супроводу процесу обробки замовлення до його виконання.

## 1.4 Означення та абревіатури

DXF – Drawing Exchange Format

SVG – Scalable Vector Graphics

API – Application Programming Interface

HTTP – HyperText Transfer Protocol

CQRS – Command Query Responsibility Segregation

REST – Representational State Transfer

JSON – JavaScript Object Notation

JWT – JSON Web Tokens

# 

# 2 ЗАГАЛЬНИЙ ОПИС

## 2.1 Перспективи продукту

Розроблюване програмне забезпечення є частиною інформаційної системи, призначеної для автоматизації процесів обліку замовлень, планування розкрою листових матеріалів та управління ресурсами (матеріалами, товщинами, верстатами). Система спрямована на використання в малих та середніх виробничих сервісах, які надають послуги з розкрою за кресленнями замовників.

Основною перевагою розроблюваного продукту є поєднання гнучкого веб-інтерфейсу з інструментами інтерактивного планування, що дозволяє користувачам у реальному часі взаємодіяти з кресленнями та розташовувати їх на аркушах для подальшого виробництва. При цьому система зберігає інформацію про кожне замовлення, пов’язує його з відповідним клієнтом, забезпечує контроль статусів і формує виробничі файли у стандартних форматі DXF.

## 2.2 Функції продукту

Програмний продукт забезпечує комплексну підтримку процесів, пов’язаних із сервісом розкрою листових матеріалів. Основною функцією є створення та збереження замовлень клієнтів, які включають креслення у форматі DXF, вибір матеріалу, товщини та кількості копій. Система дозволяє клієнтам формувати замовлення через веб-інтерфейс та відстежувати статус їх обробки.

Окремим ключовим елементом системи є редактор аркушів розкрою, який дозволяє працівнику вручну розміщувати моделі на віртуальному листі верстата. Користувач розташовує кожну копію моделей замовлення, після чого система генерує новий файл аркуша у форматі DXF і формує актуальне SVG-прев’ю.

Система підтримує керування чергами обробки для кожного верстата, враховуючи їх сумісність з конкретними матеріалами та товщинами. При плануванні розкрою система оновлює статус замовлень, відображаючи їх поточний стан — від прийому до завершення обробки.

Для забезпечення належного адміністрування система дозволяє управляти довідниками матеріалів, товщин та верстатів. Адміністратор має доступ до інтерфейсів для налаштування та контролю доступу, зокрема, управління акаунтами працівників сервісу.

Таким чином, програмний продукт реалізує функціональність для трьох категорій користувачів — клієнта, працівника сервісу та адміністратора, надаючи інструменти для керування усім життєвим циклом замовлення — від подачі до передачі результатів у виробництво.

## 2.3 Характеристика користувачів

Клієнт:

* фізична або юридична особа, яка подає замовлення на розкрій матеріалу;
* може створювати нові замовлення, прикріплювати креслення у форматі DXF, обирати тип матеріалу та товщину, вказувати кількість копій;
* має можливість переглядати статус обробки своїх замовлень та переглядати прев’ю файлу.

Працівник сервісу:

* відповідальний за формування аркушів розкрою, призначення моделей на верстати, редагування позицій моделей на аркуші;
* має доступ до редактора розміщення моделей та черги аркушів;
* може змінювати статуси замовлень залежно від етапу обробки.

Адміністратор сервісу:

* виконує управління конфігурацією системи: додає або редагує перелік доступних матеріалів, їх товщин, конфігурацій верстатів;
* має повний доступ до всієї інформації в системі.

## 2.4 Загальні обмеження

Розроблювана система у версії MVP має низку усвідомлених обмежень, які зумовлені технічними, часовими та стратегічними рамками поточного етапу реалізації. Вони визначають фокус і межі відповідальності рішення на першому етапі впровадження.

Обмеження MVP:

* система не виконує автоматичну оптимізацію розміщення моделей на листі, усі дії здійснюються вручну оператором;
* підтримується лише один формат креслень – DXF, інші формати не конвертуються і не приймаються;
* система не здійснює складський облік залишків матеріалу, облік ведеться на рівні вибору матеріалу при плануванні, без кількісного контролю;
* відсутня багатомовність інтерфейсу – реалізовано лише одну мову;
* платіжна система або облік оплати за замовлення – відсутні й не входять до обсягу функціоналу;
* ідентифікація клієнтів за юридичними особами, контрактами або ІПН – не реалізується, авторизація здійснюється тільки за e-mail/паролем.

## 2.5 Припущення та залежності

Під час проєктування і розробки системи зроблено низку технічних та функціональних припущень. Передбачається, що користувачі мають стабільний доступ до інтернету та працюють у сучасному браузері, який підтримує запуск клієнтських веб-застосунків на основі WebAssembly.

Серверна частина функціонує в середовищі з підтримкою технологій .NET, баз даних PostgreSQL та інструментів для обробки запитів і валідації. Взаємодія між компонентами базується на стандартних механізмах веб-API.

Система розрахована на роботу з файлами креслень у стандартному форматі DXF, що вважається загальноприйнятим у відповідній галузі. Також передбачено подальше розширення функціоналу, зокрема перехід до хмарного зберігання файлів та впровадження алгоритмів оптимізації розкрою.

Загалом, розробка базується на сучасному технологічному стеку з відкритим програмним забезпеченням і припускає можливість подальшої модифікації та масштабування.

# 3 КОНКРЕТНІ ВИМОГИ

## 3.1 Вимоги до зовнішніх інтерфейсів

## 3.1.1 Інтерфейс користувача

Інтерфейс має бути адаптивним і забезпечувати коректне відображення на сучасних браузерах: Google Chrome, Mozilla Firefox, Microsoft Edge. Для мобільних пристроїв підтримка може бути обмеженою, але передбачена в перспективних випусках.

Інтерфейс повинен бути інтуїтивно зрозумілим для користувачів із базовими навичками роботи з ПК.

Інтерфейс динамічно адаптується до ролі користувача, обмежуючи доступ до функцій відповідно до прав.

## 3.1.2 Програмний інтерфейс

Усі запити до API поділяються за ролями користувачів (адміністратор, працівник сервісу, замовник), що забезпечує контрольований доступ до функціоналу.

API включає такі функціональні групи ендпоінтів:

* обробка замовлень (створення, оновлення, перегляд, статуси);
* керування матеріалами, товщинами та верстатами;
* завантаження креслень та отримання прев’ю;
* формування та управління аркушами розкрою.

Кожен ендпоінт документується засобами Swagger/OpenAPI, що дозволяє легко інтегрувати систему з іншими сервісами або використовувати її як частину більшої інформаційної платформи.

## 3.1.3 Комунікаційний протокол

Комунікація між клієнтською та серверною частинами програмної системи здійснюється за допомогою протоколу HTTP/HTTPS, що є стандартом для веб-застосунків. Для забезпечення конфіденційності, цілісності та безпеки передавання даних використовується шифрування через HTTPS.

Обмін даними відбувається у форматі JSON, що забезпечує зручну структурування інформації та підтримується більшістю сучасних технологій. Для передавання креслень у форматі DXF використовується формат MultipartFormData, який дозволяє передавати файли разом з метаданими.

Аутентифікація запитів реалізована через JWT. Кожен запит до захищених API має містити заголовок Authorization із токеном доступу, що підтверджує особу користувача та його роль у системі.

Комунікація відбувається в архітектурі "запит-відповідь", що передбачає ініціацію клієнтом дії та отримання результату із відповідним кодом статусу (наприклад, 200 OK, 400 Bad Request, 401 Unauthorized тощо). Це забезпечує передбачувану поведінку клієнта та сервера і спрощує обробку помилок.

## 3.1.4 Обмеження пам’яті

Програмна система не має критичних вимог до обсягу оперативної пам’яті, однак для стабільної роботи клієнтської частини рекомендується щонайменше 4 ГБ ОЗП на стороні користувача, особливо при обробці складних креслень. Для серверної частини, враховуючи необхідність обробки запитів, зберігання та обробки файлів DXF і SVG, рекомендовано не менше 8 ГБ ОЗП на сервері при одночасній роботі з декількома замовленнями.

У разі перевищення лімітів пам’яті на клієнтській стороні можлива деградація інтерфейсу редактора або затримки при відображенні великої кількості елементів. На сервері — затримка відповіді або зростання часу обробки запитів. Тому рекомендовано застосування інструментів моніторингу ресурсів при продуктивному використанні системи.

## 3.1.5 Операції

Програмне забезпечення має підтримувати низку основних операцій, необхідних для повноцінного циклу обробки замовлень на розкрій листових матеріалів. Усі операції мають бути доступні через веб-інтерфейс залежно від ролі користувача.

Основні операції, які виконує система:

* створення та перегляд замовлення: клієнт має можливість створити нове замовлення, прикріпити креслення у форматі DXF, вказати матеріал, товщину та кількість копі, замовлення зберігається на сервері, де стає доступним для обробки;
* завантаження та перегляд креслення: після завантаження файлу DXF система автоматично створює SVG-прев’ю, яке може бути переглянуте як клієнтом, так і працівником сервісу;
* розміщення моделей на аркуші: працівник сервісу може створити аркуш розкрою, обрати верстат і матеріал, після чого розмістити замовлені моделі вручну за допомогою редактора;
* генерація файлів розкрою: після розміщення система автоматично створює оновлений DXF-файл аркуша та SVG-файл для попереднього перегляду, з урахуванням координат розміщення кожної моделі;
* керування чергою виробництва: кожен лист розкрою має статус обробки (очікує, у процесі, виконано), черга верстата формується автоматично відповідно до створених листів;
* адміністрування довідників: адміністратор має можливість додавати, редагувати або видаляти записи про матеріали, товщини та верстати, а також керувати користувачами системи.

Усі операції передбачають перевірку прав доступу відповідно до ролі користувача. Інтерфейс побудовано таким чином, щоб забезпечити мінімальну кількість кроків для виконання кожної операції та швидкий доступ до необхідної функціональності.

## 3.1.6 Функції продукту

Функції клієнта (замовника) наступні:

* MF-1: створення нового замовлення через веб-інтерфейс із вказанням матеріалу, товщини та кількості копій;
* MF-2: завантаження креслення у форматі DXF для кожного замовлення;
* MF-3: перегляд списку власних замовлень із фільтрацією за статусом;
* MF-4: ознайомлення з поточним статусом виконання кожного замовлення.

Функції працівника сервісу наступні:

* MF-5: перегляд усіх замовлень, що очікують на розміщення для обробки;
* MF-6: створення листа (аркуша розкрою) із вибором матеріалу, товщини та верстата;
* MF-7: розміщення моделей на аркуші;
* MF-8: вибір кількості копій моделі, що має бути розміщена;
* MF-9: видалення або переміщення моделей на аркуші в редакторі;
* MF-10: збереження розміщення на аркуші, що призводить до генерації плану та оновлення статусів замовлень;
* MF-11: призначення аркуша на конкретний верстат із подальшим додаванням у чергу;
* MF-12: завершення обробки листа, що змінює статус усіх пов’язаних замовлень на Done;
* MF-13: видалення листа з поверненням замовлень у статус Created.

Функції адміністратора сервісу наступні:

* MF-14: створення та редагування довідника матеріалів із вказанням типу та доступних товщин;
* MF-15: створення та редагування верстатів із вказанням максимальних розмірів листа та списку допустимих матеріалів;
* MF-16: призначення матеріалів верстатам;
* MF-17: управління обліковими записами працівників сервісу.

Системні функції (спільні):

* MF-18: аутентифікація та авторизація користувачів;
* MF-19: автоматичне оновлення статусу замовлення залежно від дій оператора;
* MF-20: захищене зберігання файлів креслень на сервері (без публічних URL);
* MF-21: генерація SVG-прев’ю на основі DXF-файлу;
* MF-22: завантаження DXF-файлів працівником або адміністратором для подальшого використання.

## 3.1.7 Припущення та залежності

Припущення системи наступні:

* A-1: усі користувачі системи мають доступ до інтернету через сучасний веб браузер (Chrome, Firefox, Edge);
* A-2: клієнти завантажують креслення у форматі DXF, що відповідає мінімальному стандарту підтримки (наприклад, 2D полілінії, без вкладених блоків);
* A-3: для кожного замовлення припускається використання одного типу матеріалу й однієї товщини;
* A-4: усі верстати попередньо сконфігуровані адміністратором сервісу з актуальними параметрами (допустимі матеріали, розміри листів тощо);
* A-5: розміщення моделей на листі у редакторі здійснюється вручну – автоматичне оптимізоване розміщення у версії MVP не передбачено;
* A-6: статуси замовлень змінюються лише через визначені дії у системі (збереження листа, завершення обробки замовлення, видалення тощо).

Залежності системи наступні:

* D-1: облікові записи користувачів керуються внутрішньою системою авторизації на основі ASP.NET Identity та JWT токенів;
* D-2: візуальний редактор базується на можливостях Blazor WebAssembly та обробки SVG;
* D-3: дані про матеріали, товщини та верстати зберігаються в централізованій базі даних і повинні бути актуальними для коректної роботи логіки створення листів;
* D-4: кожен аркуш розкрою може бути оброблений лише на одному верстаті, обраному вручну оператором;
* D-5: завантажені DXF-файли зберігаються у хмарному файловому сховищі (наприклад, Azure Blob Storage або AWS S3), доступ до файлів здійснюється виключно через авторизовані API з перевіркою ролі користувача та приналежності до замовлення. Пряма адреса до файлу не публікується.

## 3.2 Властивості програмного продукту

У процесі розробки програмного забезпечення використовуються сучасні засоби та фреймворки, які забезпечують модульність, масштабованість, безпечну архітектуру та зручну підтримку коду. Нижче наведено основні технології та інструменти, якими користувався здобувач при реалізації програмної системи.

Мова програмування: C# — основна мова розробки як для клієнтської, так і для серверної частин.

Серверна частина: ASP.NET Core Web API — для побудови RESTful-сервісу, що обробляє запити клієнта та взаємодіє з базою даних.

Клієнтська частина: Blazor WebAssembly — для створення SPA-застосунку з виконанням C#-коду безпосередньо у браузері, без потреби у JavaScript.

База даних та ORM: PostgreSQL 15 — реляційна СУБД для зберігання усіх структурованих даних. Entity Framework Core — ORM для зручної роботи з базою через моделі .NET.

Патерни та інфраструктура логіки: MediatR — для реалізації шаблону CQRS та чіткого розділення запитів і команд.

Тестування: xUnit — фреймворк для написання модульних тестів, який дозволяє перевіряти окремі компоненти серверної логіки.

## 3.3 Атрибути програмного продукту

Атрибути програмного продукту описують його якісні характеристики — наскільки він відповідає вимогам до функціональності, надійності, безпеки, зручності використання та адаптивності. Для розробленої системи визначено ключові атрибути.

Надійність — система стабільно виконує основні операції, зокрема створення замовлень, обробку креслень та формування аркушів розкрою. Передбачена валідація вхідних даних, перевірка файлів, обробка помилок на рівні сервісів і запитів, що мінімізує ймовірність збоїв при некоректному введенні.

Доступність — продукт є веб-застосунком і доступний через браузер на будь-якому пристрої з підключенням до Інтернету. Не потребує встановлення спеціального ПЗ. Підтримуються всі сучасні браузери.

Безпека — використовується рольова модель доступу. Аутентифікація реалізована через JWT-токени, які передаються в кожному запиті. Кожен користувач має обмежений доступ до функцій відповідно до своєї ролі. Захищено збереження файлів та бази даних.

Супроводжуваність — система побудована на модульних принципах з чітким розділенням відповідальності, що дозволяє легко підтримувати та розширювати окремі компоненти без впливу на інші. Уся логіка структурована, використовуються сервіси з інтерфейсами.

Переносимість — завдяки використанню стандартів веб-технологій, система може бути розгорнута на різних операційних системах та середовищах. Відсутність залежності від апаратного забезпечення користувача.

Продуктивність — інтерфейс на Blazor WebAssembly працює у браузері без звернення до сервера на кожну дію, що забезпечує швидкий відгук та ефективну взаємодію користувача з редактором. Серверна частина побудована з урахуванням асинхронної обробки запитів та масштабування.

## 3.4 Вимоги бази даних

База даних системи має забезпечувати зберігання структурованої інформації про замовлення, клієнтів, верстати, матеріали, товщини, аркуші розкрою та розміщення моделей. Для реалізації цієї функціональності використовується PostgreSQL 15. Доступ до бази даних здійснюється через ORM-фреймворк Entity Framework Core. Структура таблиць побудована з урахуванням нормалізації даних до третьої нормальної форми, що забезпечує відсутність надлишковості, уніфіковане зберігання інформації та логічну узгодженість сутностей. Вимоги до бази включають підтримку зв’язків типу «один-до-багатьох» та «багато-до-багатьох», забезпечення цілісності даних, ефективну роботу з транзакціями та можливість масштабування.