

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Програмний додаток для налаштування моделі перед друком на 3D-принтері»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології  
проектування»

**Виконавець роботи:** студент групи ІТ-61-7 Новохатько Ігор Олександрович

**Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК  
з оцінкою**

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 р.

Науковий керівник

\_\_\_\_\_

(підпис)

Парфененко Ю.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

\_\_\_\_\_

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук  
Секція інформаційних технологій проектування  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
«\_\_» \_\_\_\_\_ 2020 р.

**1 ЗАВДАННЯ**  
2 НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

*Новохатько Ігор Олександрович*

**1 Тема роботи** Програмний додаток для налаштування моделі перед друком на 3D-принтері

**керівник роботи** Парфененко Юлія Вікторівна \_\_\_\_\_,

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

**2 Строк подання студентом роботи** «1» червня 2020 р.

**3 Вхідні дані до роботи** Методика тріангуляції 3D моделі, методика згладжування 3D моделі, методика слайсингу 3D моделі

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** 1) Аналіз предметної області, 2) Проектування web-додатку для налаштування моделі перед друком на 3D-принтері, 3) Розробка web-додатку для налаштування моделі перед друком на 3D-принтері

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** аналіз предметної області, мета і задачі дипломного проекту, порівняння аналогів, функціональні вимоги до web-додатку, моделювання роботи web-додатку, декомпозиція першого рівня, декомпозиція під процесів, діаграма варіантів використання, архітектура додатку, засоби реалізації, демонстрація додатку, висновки, оприлюднення роботи

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_ 01.10.2019 \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Оформлення планування робіт	До 08.03.2020	
2.	Оформлення технічного завдання	До 15.03.2020	
3.	Проведення аналізу предметної області	До 22.03.2020	
4.	Проведення структурно-функціонально моделювання процесів	До 10.04.2020	
5.	Розробка додатку	До 11.05.2020	
6.	Тестування додатку	До 17.05.2020	
7.	Здача пояснювальної записки та файлів розробленого проекту	До 01.06.2020	

Студент

\_\_\_\_\_

(підпис)

Новохатько І.О.

Керівник роботи

\_\_\_\_\_

(підпис)

Парфененко Ю.В.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Програмний додаток для налаштування моделі перед друком на 3D-принтері».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 20 найменувань, додатків. Загальний обсяг роботи – 63 сторінки, у тому числі 33 сторінки основного тексту, 2 сторінки списку використаних джерел, 21 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку для налаштування моделі перед друком на 3D-принтері.

В роботі проведено огляд останніх досліджень та публікацій, які стосуються 3D друку, аналіз аналогів, моделювання та проектування додатку.

У роботі виконано розробку додатку, а саме таких його частин:

- модуль завантаження моделі;
- модуль відтворення завантаженої моделі;
- модуль збереження отриманих результатів;
- модуль слайсингу 3D моделі;
- модуль згладжування 3D моделі;
- модуль особистого кабінету користувача;
- модуль адміністратора системи.

Результатом проведеної роботи є розроблений web-додаток для налаштування моделі перед друком на 3D-принтері.

Ключові слова: web-додаток, слайсинг, тріангуляція, WebGL.

## ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Огляд останніх досліджень і публікацій.....	7
1.2 Аналіз програмних продуктів - аналогів .....	10
1.3 Постановка задачі.....	12
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ.....	14
2.1 Моделювання WEB-ДОДАТКУ в IDEF0 .....	14
2.2 Проектування web-додатку.....	17
2.3 Проектування моделі бази даних .....	19
3 РОЗРОБКА WEB-ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ.....	20
3.1 Архітектура web-додатку .....	20
3.2 Програмна реалізація.....	21
3.3 Використання програмного додатку.....	26
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	33
ДОДАТОК А Технічне завдання.....	35
ДОДАТОК Б Планування робіт .....	45
ДОДАТОК В Лістинг програмного коду.....	Ошибка! Закладка не определена.

## ВСТУП

Головне призначення 3D-принтерів полягає у їх використанні для роботи і навчання. Цей пристрій може вирішити безліч завдань: створення прототипів, моделей, прес-форм, друк серійних виробів і деталей.

Для архітекторів або дизайнерів це дає можливість виготовляти макети моделей і створювати щось нове. Це дозволить фахівцям попередньо проаналізувати проект до початку реалізації і презентувати його в найвигіднішому світлі.

Для художників, виготовлення оригінальних інсталяції, для майстрів-лялькарів – виготовлення рухомих та нерухомих моделей, для модельєрів – спрощується виготовлення одягу та взуття, цілих костюмів та аксесуарів до них. Сучасні 3D-принтери дозволяють друкувати кольорові об'єкти, декількома матеріалами та навіть повністю кольорові моделі.

Перед друком необхідно виконати ряд налаштувань 3D моделі, що зумовлює необхідність розробки програмного додатку, який би поєднував можливості трансформування, зміни розмірів друкуючої поверхні та слайсингу моделі.

Метою роботи є розробка програмного додатку для підготовки 3D-моделі до друку, а також аналіз моделі на наявність різного виду помилок.

Основними задачами являються:

- дослідження предметної області та формування технічного завдання;
- вибір методів розробки програмного додатку для підготовки 3D-моделі до друку;
- моделювання роботи програмного додатку;
- розробка програмного додатку, тестування та перенесення на хостинг.

Практичне значення розробленого додатку полягатиме у його використанні для тих, хто займається друком на 3D принтері.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

Можливості 3D-друку практично безмежні. Технологія тривимірного друку дозволяє на порядок прискорити винахідницьку діяльність, рішення задач підготовки виробництва, а в ряді випадків вона вже активно застосовуються і для виробництва готової продукції.

Медики завдяки 3D-друку здатні створювати імпланти, шини, значно комфортніші ніж гіпси, також біонічні протези. Біодрук використовує живі клітини для друку органів щоб в подальшому можна було замінити ними нездорові органи.

Деякі завдяки 3D-друку змогли створити цілком надрукований електрокар. Електрокар Strati має багато відмінностей від класичних авто, це може сильно змінити індустрію автомобілебудування.

Також з'явився друк титанових елементів, вони нестимуть навантаження на корпус. Була розроблена технологія друку швидкого плазмового осадження.

Насправді для того, щоб надрукувати модель потрібно мати відповідні навички. Необхідно знайти потрібну модель, проаналізувати модель, створити підтримки, якщо це необхідно, знати, які параметри у принтера, для кожної моделі необхідні свої параметри друку, температура, швидкість друку, якість філаменту та інше [1].

Для прикладу можна розглянути слайсинг моделі в десктопному додатку Photon Workshop, спочатку модель завантажується в додаток, частина параметрів налаштовані, тобто необхідно лише ввести параметри друку [2].

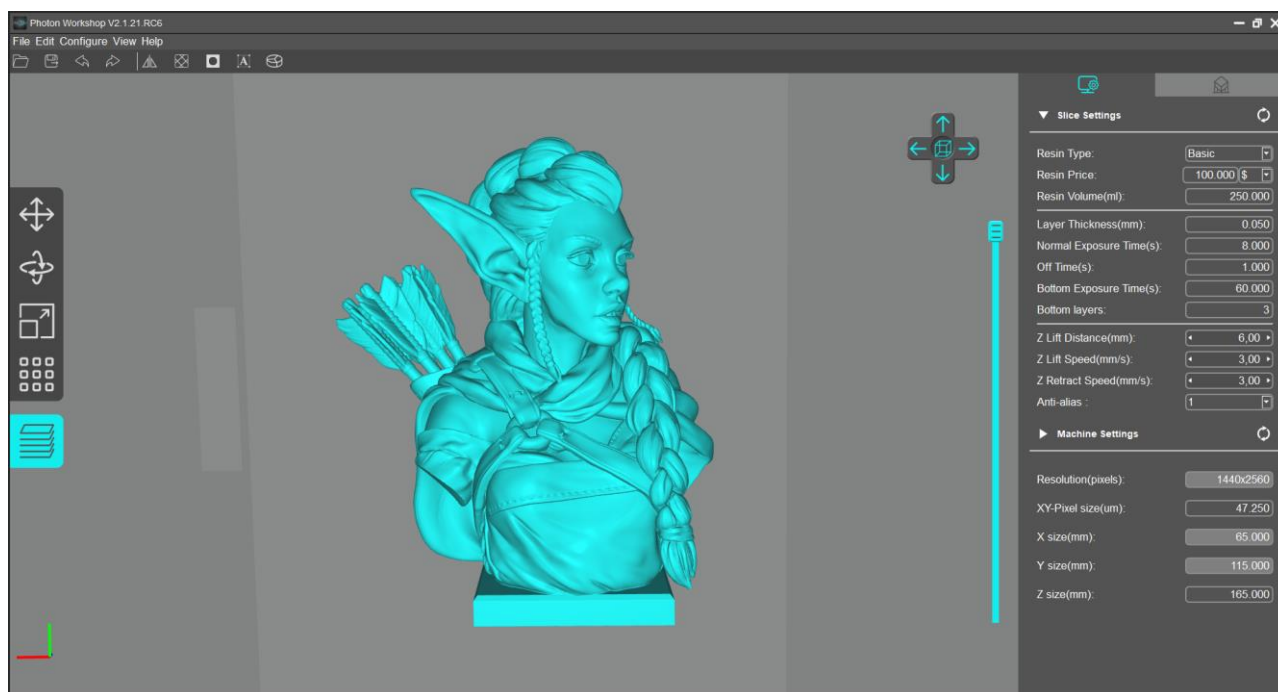


Рисунок 1.1 – Photon Workshop

За допомогою такого додатку можна завантажити модель, проаналізувати, зробити слайсинг, але для зручності використання web-додатку було б значно кращою ідеєю. [3].

Також для кожного типу принтерів існує спеціальне програмне забезпечення. В особливості для FDM принтерів програмне забезпечення універсальне і відрізняється лише алгоритмами роботи, тим часом як SLA принтери використовують програмне забезпечення тільки від компанії виготовлювача, так само принтери типу DLP та інші [3].

В ході аналізу одною з важливих функцій необхідних для реалізації є перегляд моделі в 2D форматі, це дозволить визначити слабкі місця моделі та більш детально переглянути процес друку (рис.1.2).

Після слайсингу модель можна переглянути в 2D форматі.

Саме функції перегляду, параметризації моделі та слайсингу можна перенести до web-додатку.



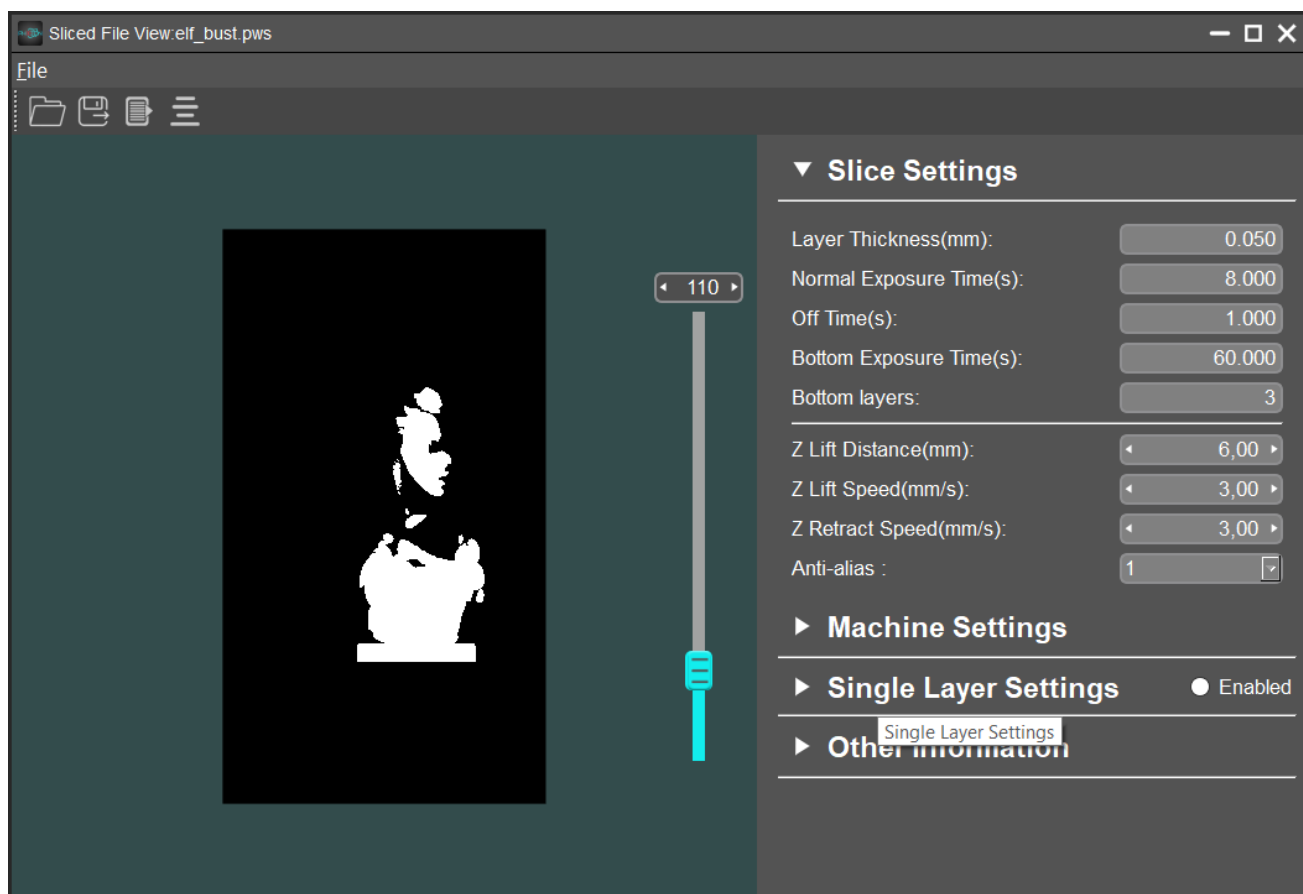


Рисунок 1.2 – Перегляд моделі в 2D форматі

Так як існує багато програм для створення 3D моделей, не всі можуть експортувати модель, не пошкодивши модель.

Вище описаний функціонал доступний лише в десктопних додатках, web-додатки зі схожим функціоналом можуть працювати лише з принтерами типу FDM [4].

Тому актуальною є задача розробки додатка з такими функціями, щоб їм можна було скористатися, не завантажуючи спеціальне програмне забезпечення.

Для виконання аналізу моделей використовують різні методики, в результаті користувач може переглянути моделі в 2D форматі. Прикладом є рис. 1.2, шар за шаром, також переглянути модель в 3D форматі, зберегти результати в форматі stl, png, zip [5].

Для того, щоб спростити розробку, було використано конструктори, плагін point in polygon, бібліотеку three.js, WebGL.

## 1.2 Аналіз програмних продуктів - аналогів

Аналогові додатки в основному потребують реєстрацію, за додатковий функціонал необхідно додатково оформити підписку.

Але головна відмінність таких додатків те, що такі додатки працюють з принтерами типу FDM. Розроблюваний додаток розрахований на роботу з принтерами типу DLP [6].

Розглянемо функції web-додатків:

- Astroprint (рис.1.3);
- 3DPrinterOS (рис.1.4);
- SliceCrafter (рис.1.5);

Astroprint являє собою проект, за допомогою якого можна встановити одноіменну систему на Raspberry PI, підключивши wifi модель дозволить управляти принтером за допомогою web-додатку. За додаткову плату дозволить робити слайсинг моделей, завантажувати їх та посилати на друк, можна робити відеозапис друку, перевіряти роботу принтера з будь-якої точки світу, але це коштує не мало грошей.

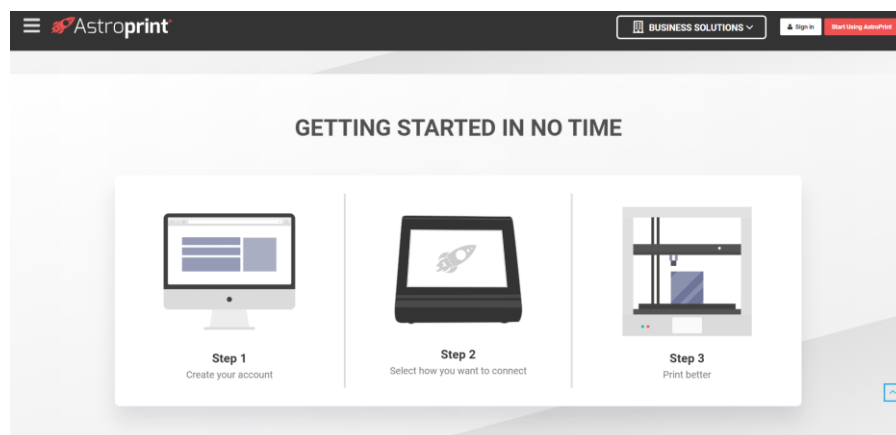


Рисунок 1.3 – Web-додаток Astroprint

3DPrinterOS - це хмарна платформа, яка включає в себе додаток для нарізки, а також інші функції, необхідні для 3D-друку, такі як додаток для ремонту. Таким

чином, платформа являє собою простий спосіб управління файловими машинами та користувачами по всьому бізнесу. У 3DPrinterOS є три додатки для нарізки: «Cloud Slicer», «Slicer 2» і виділений «Makerbot Slicer». Асортимент принтерів, підтримуваних цією платформою, дуже великий.

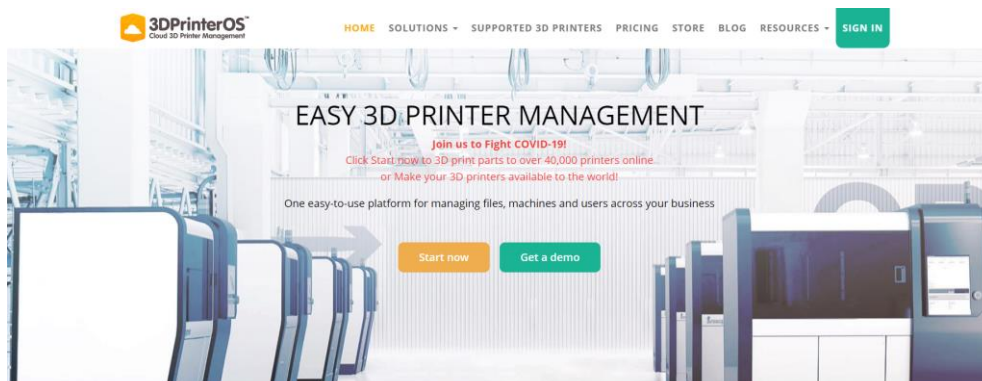


Рисунок 1.4 – Web-додаток 3DPrinterOS

IceSL - це не просто програмне забезпечення для нарізки, а інструмент для тривимірного моделювання, створений за допомогою скриптів на мові Lua. Ця ж технологія дозволяє ефективно нарізати і генерувати інструкції 3D-принтера (за допомогою G-коду), уникаючи дорогого етапу створення сітки. [7] Також є можливість завантажити IceSL-слайсер, який орієнтований виключно на нарізку. SliceCrafter від тієї ж компанії є онлайн-версію IceSL-слайсера, але менш потужну. У них є ряд функцій, детально описаних на їхньому сайті.

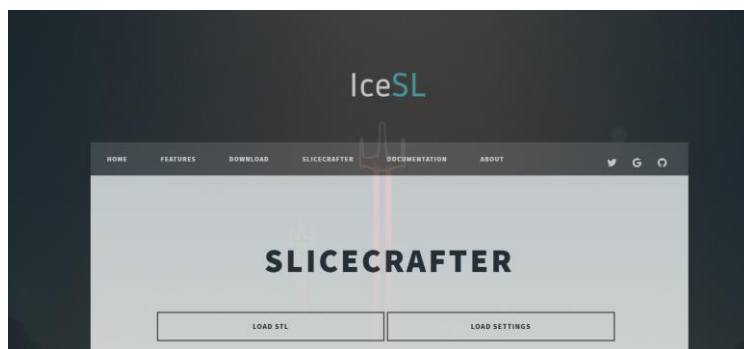


Рисунок 1.5 – Web-додаток SliceCrafter

Альтернативні додатки мають значно більший функціонал, але над такими додатками працюють команди розробників, а також за такі можливості необхідно платити. Для кожного типу принтеру необхідні різні алгоритми слайсингу моделі, а також не кожен із вище перерахованих вище web-додатків підходять для кожної моделі принтеру, таким чином web-додатки мають обмеження щодо типів принтерів [8].

Порівняльна таблиця для вищеперерахованих додатків знаходиться нижче.

Таблиця 1.1

Характеристика	IceSL	<u>3DPrinterOS</u>	Astroprint
Зручний інтерфейс	+	+	-
Реєстрація користувачів	-	+	+
Працездатність	-	+	+
Швидкий зворотній зв'язок	-	-	+

В результаті аналізу було встановлено, що більшість web-додатків такого типу мають схожі функції, але також мають свої відмінності, але додатку з необхідними функціями для DLP принтерів немає, тому виникла необхідність створення додатку такого типу.

### 1.3 Постановка задачі

В ході виконання кваліфікаційної роботи бакалавра буде розроблено програмний додаток для підготовки 3D моделей до друку який реалізовано в якості web-додатку, а також аналізу моделі на наявність браку.

Web-додаток повинен виконувати ряд задач:

- завантаження моделі для подальшої підготовки до друку та аналізу;

- трансформація моделі ( за необхідності);
- графічне відображення моделі;
- перегляд окремих шарів в 2D форматі;
- зміна розмірів друкуючої поверхні;
- зміна параметрів слайсингу;
- слайсинг моделі.

Для реалізації поставленого завдання були виділені задачі для дослідження.

Розробка інтерфейсу полягає в формуванні шаблону та його реалізації, а також адаптація інтерфейсу.

Розробка додатку полягає в реалізації функціональної частини для організації модулю.

Розробка модулю завантаження моделі дасть змогу користувачу завантажити свою модель в додаток.

Розробка модулю слайсингу моделі представляє собою модуль який відповідає за слайсинг моделі використовуючи необхідні параметри.

Розробка модулю перегляду окремих шарів дасть можливість після слайсингу розглянути модель пошарово в форматі 2D.

Відладка та тестування дозволить виявити помилки та недоліки.

Технічне завдання на розробку програмного додатку представлене в додатку А, а планування робіт – у додатку Б.

## 2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ

### 2.1 Моделювання WEB-ДОДАТКУ в IDEF0

Процес проектування web-додатку необхідно розпочинати з розробки контекстної діаграми IDEF0 [9]. Дане рішення прийнято через те, що потрібно розуміти чіткі та зрозумілі твердження, що описують систему її взаємодію з навколишнім середовищем. До контекстної діаграми входять:

- Вхід: дані, що використовуються системою для отримання результату(вихідних даних).
- Вихід: дані, які отримуються в результаті роботи системи.
- Управління: інформація, управляюча діями системи.
- Механізми: ресурси, що виконують роботу в системі.

Проаналізувавши головні елементи контекстної діаграми, для налаштування моделі перед друком на 3D-принтері, на основі цього сформовано наступний перелік:

- Вхід: файли 3D-моделі в форматі stl.
- Вихід: відслайсована модель в форматі png або svg.
- Управління: методика аналізу моделі, параметризація моделі, слайсинг.
- Механізми: користувач, three.js, WebGL, інформаційна система.

На основі цих даних, була розроблена контекстна діаграма, що представлена на рис. 2.1.

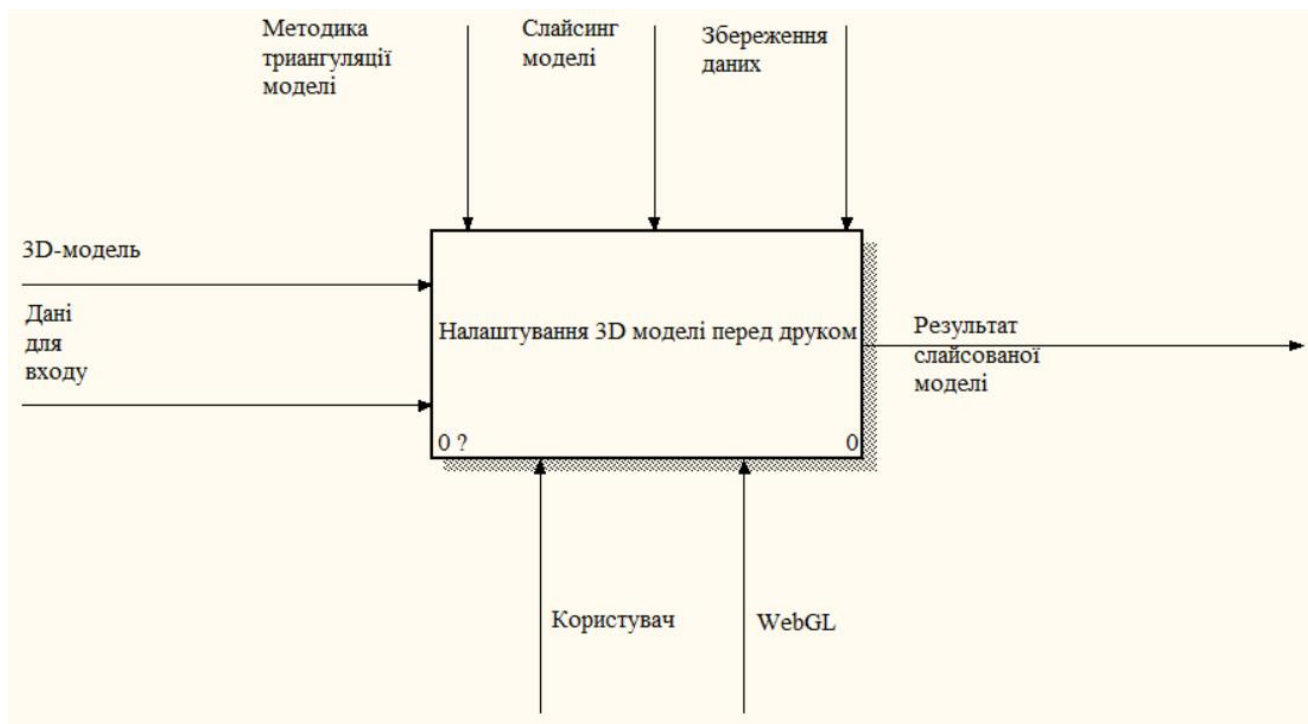


Рисунок 2.1 - Контекстна діаграма

Таке представлення показує лише загальні можливості. Для того, щоб побачити більш деталізовану модель необхідно зробити декомпозицію (рис. 2.2).

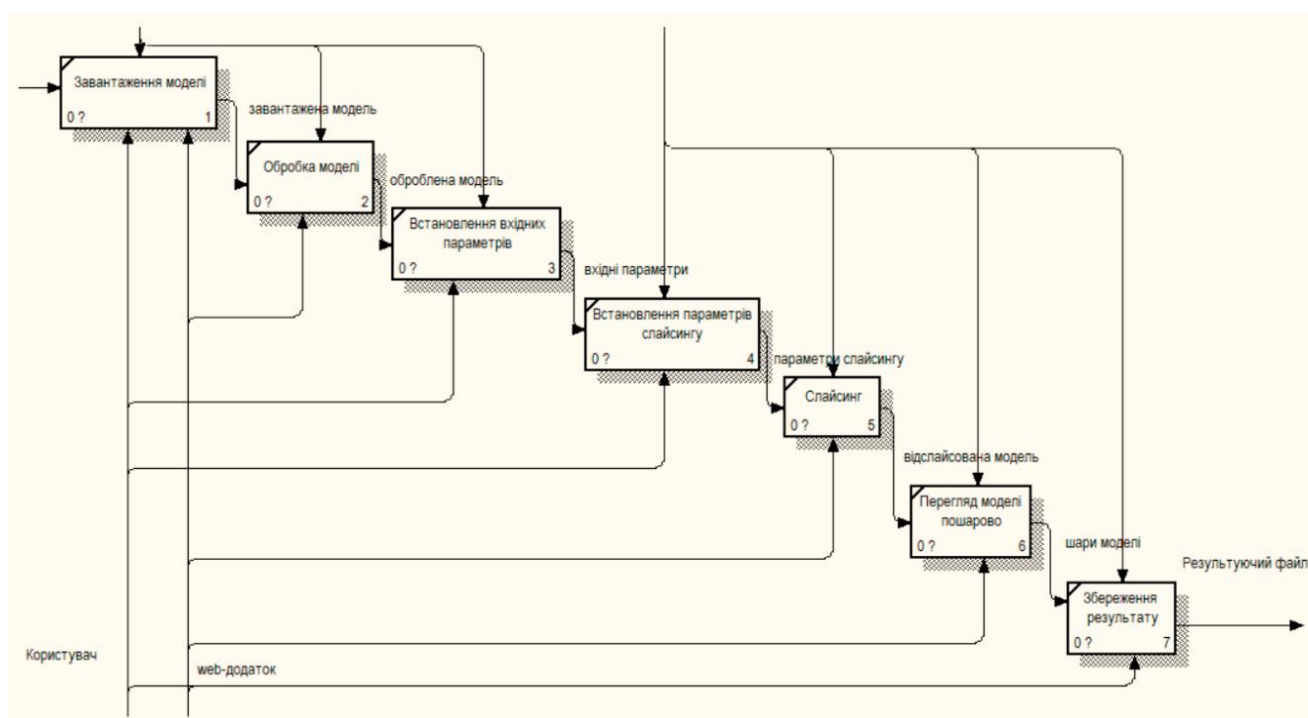


Рисунок 2.2 – Декомпозиція контекстної діаграми

Процес підготовки моделі до друку можна розділити на: аналіз складності моделі, параметризація, слайсинг. Декомпозиції зображені нижче (рисунки 2.2-2.4), декомпозиція зроблена в нотації IDEF3.

До першого підпроцесу (аналіз складності моделі) можна віднести:

- Вхід: завантажена модель.
- Вихід: звіт про результати аналізу.
- Управління: відтворення моделі в форматі 3D, методика аналізу пошкоджень моделі.
- Механізми: користувач, web-додаток.

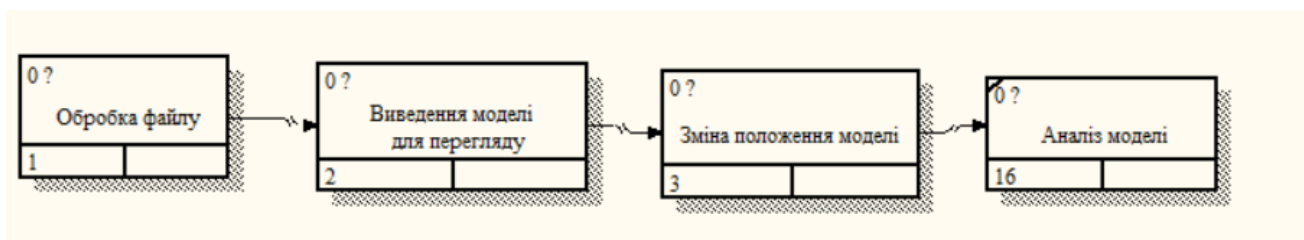


Рисунок 2.3 – Аналіз складності моделі

До другого підпроцесу (параметризація) були сформовані наступні дані:

- Вхід: завантажена модель.
- Вихід: параметризована модель відповідно до вподобань користувача.
- Управління: панель параметрів.
- Механізми: WebGL, three.js, web-додаток.

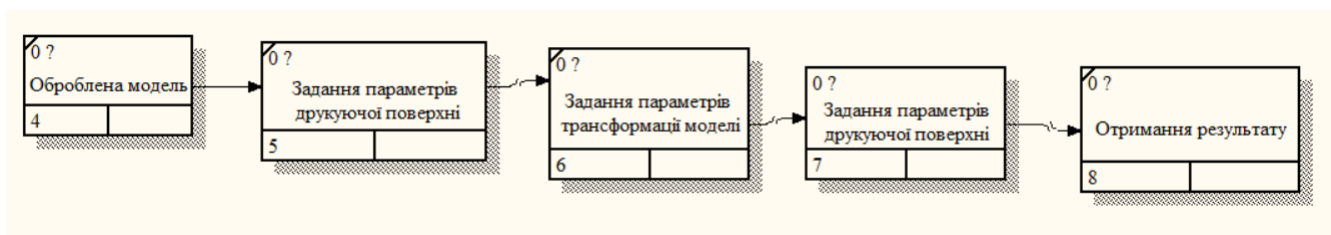


Рисунок 2.4 – Параметризація



До третього підпроцесу (слайсинг) були сформовані наступні дані:

- Вхід: завантажена модель.
- Вихід: слайсована модель.
- Управління: методика слайсингу моделі для відповідного типу принтеру.
- Механізми: Користувач, web-додаток.

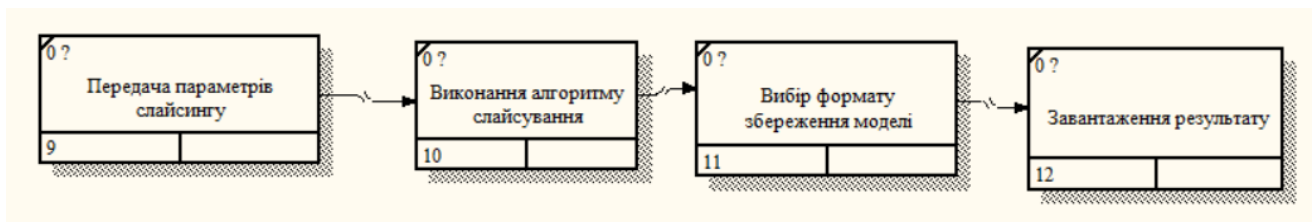


Рисунок 2.5 – Слайсинг

## 2.2 Проектування програмного додатку

Подальша розробка потребує діаграми варіантів використання (Use Case Diagram). Дана діаграма відображає взаємодію акторів та варіантів використання. Діаграма варіантів використання описує, з точки зору чинного особи, групу дій в системі, які призводять до конкретного результату. Розробка даного типу діаграм необхідна за потреби розуміння взаємодії між акторами та варіантами використання.

Для розроблюваної системи було визначено наступних акторів:

- Відвідувач – має доступ до основного функціоналу.
- Користувач з правом читання-запису – має доступ до основного функціоналу та можливість завантаження моделей для швидкого доступу.
- Адміністратор – має доступ до управління вільним місцем та зайнятим дисковим простором.
- WebGL, three.js – відображає завантажену модель.

Після визначення акторів в системі, формується перелік варіантів використання.

- Завантаження моделі.
- Перегляд моделі.
- Параметризація моделі.
- Слайсинг.
- Збереження результатів слайсингу.
- Авторизація/Реєстрація.

На основі сформованих даних про варіанти використання та акторів було розроблено Use Case діаграму, яка представлена рис. 2.6

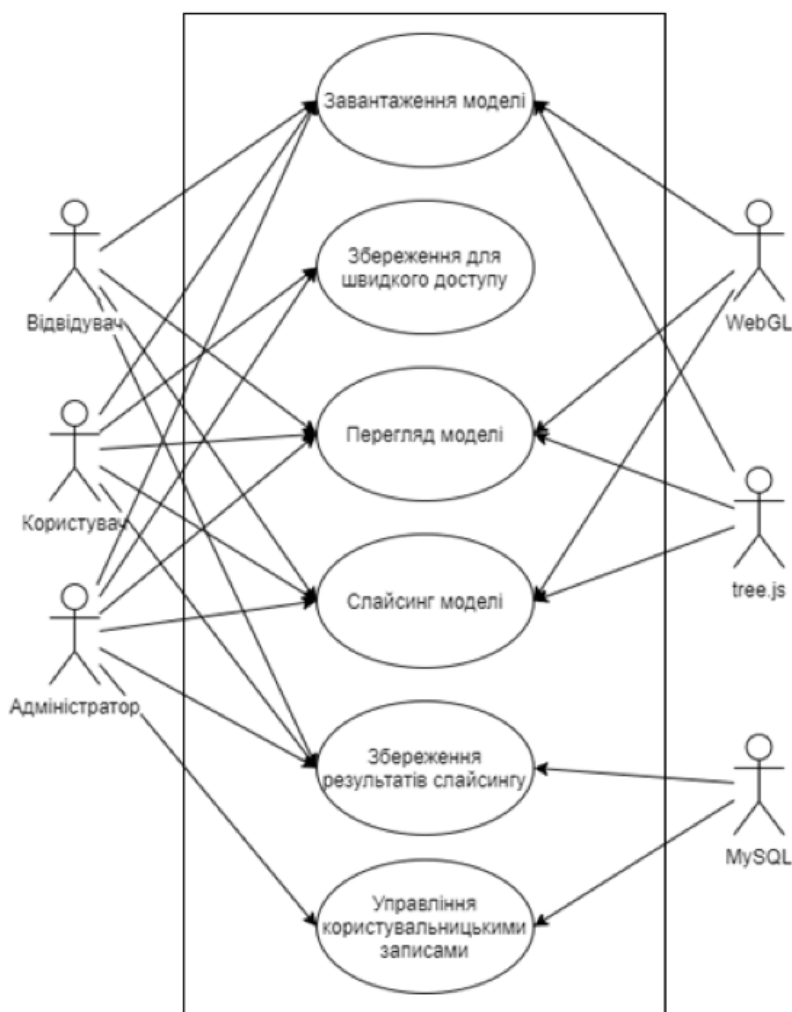


Рисунок 2.6 – Діаграма варіантів використання програмного додатку

Відвідувачі можуть завантажувати модель, переглядати модель, зберігати результати слайсингу.

Користувачі можуть завантажувати модель, переглядати модель, зберігати результати слайсингу також можуть зберігати моделі для швидкого доступу.

Адміністратор може завантажувати модель, переглядати модель, зберігати результати слайсингу, зберігати моделі для швидкого доступу також управління записами користувачів та вільним місцем на диску.

WebGL разом з three.js забезпечують роботу з моделлю.

MySQL забезпечує авторизацію/реєстрацію користувачів.

### 2.3 Проектування моделі бази даних

При проектуванні бази даних було розроблено ER-діаграму, яка відображає таблиці та зв'язки між ними (рис. 2.7).

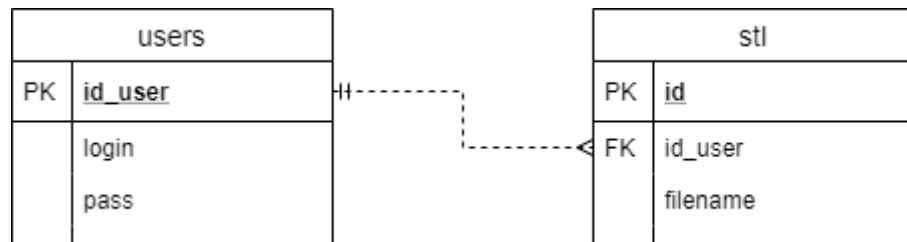


Рисунок 2.7 – ER-діаграма бази даних

База даних містить в собі таблицю “users” та таблицю “stl”. До першої таблиці заносяться дані користувача, в даному випадку логін та пароль, ця таблиця має зв'язок з таблицею “stl”, вона в свою чергу містить назви файлів, які користувач завантажив.

Інтерфейс користувача передбачає управління файлами, тобто користувач може видаляти файли, якщо вони непотрібні.

## 3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ДЛЯ НАЛАШТУВАННЯ МОДЕЛІ ПЕРЕД ДРУКОМ НА 3D-ПРИНТЕРІ

### 3.1 Архітектура програмного додатку

Додаток виконується клієнтською частиною, швидкість завантаження та обробки моделі залежать від можливостей ПК клієнта.

Головною перевагою додатку являється те, що після завантаження сторінки він буде виконувати всі основні функції навіть без наявності доступу до мережі, в такому випадку web-сервер не є обов'язковою частиною.

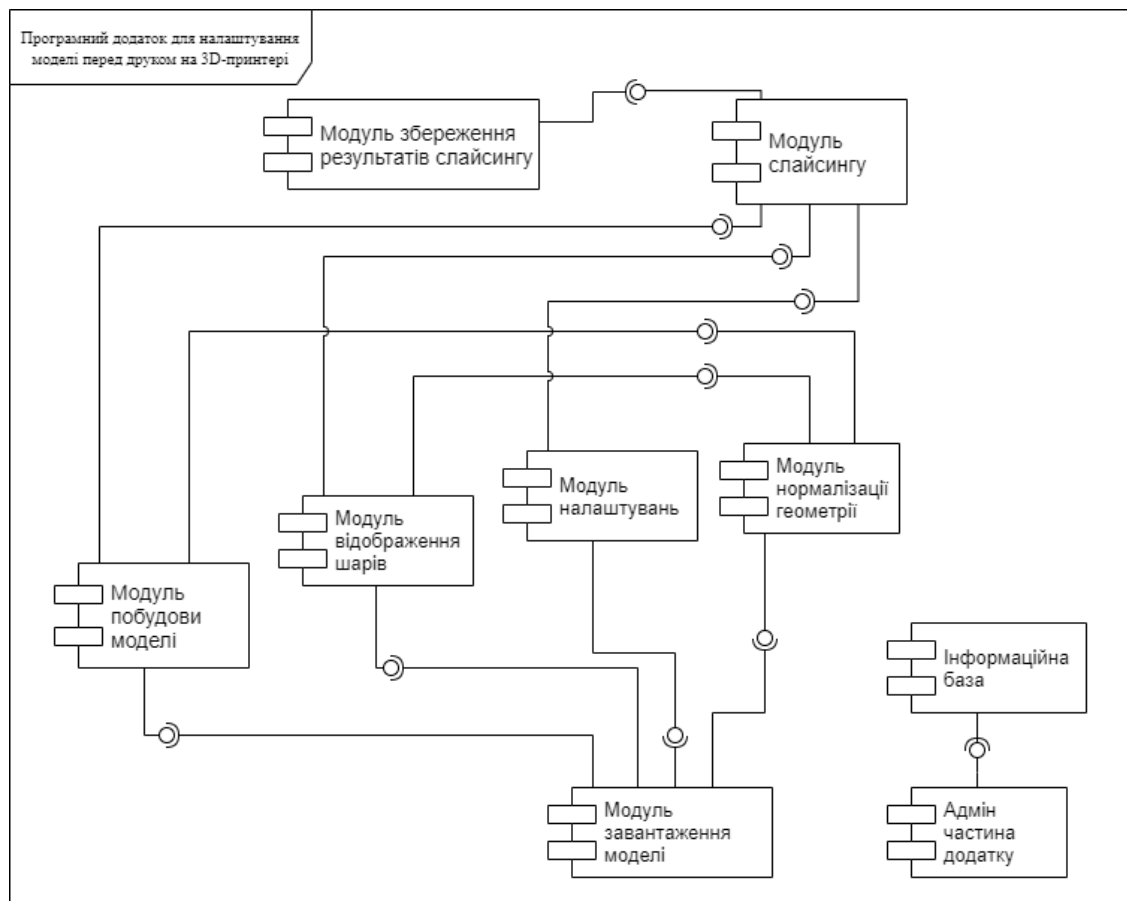


Рисунок 3.1 – Архітектура web-додатку

Початкові дані поступають з модуля завантаження моделі після чого модель відтворюється, нормалізує геометрію, модуль відображення шарів розраховує їх кількість та передає до модулю слайсингу в свою чергу результати слайсингу передаються модулю збереження результатів.

## 3.2 Програмна реалізація

Спочатку було реалізовано інтерфейс користувача, щоб не ускладнювати цю задачу було використано Bootstrap. Це дозволило створити зручний та зрозумілий інтерфейс не прикладаючи великих зусиль, тобто дозволило зменшити витрати часу та вдосконалити backend.

На рисунку 3.2 продемонстровано фінальний макет головної сторінки.

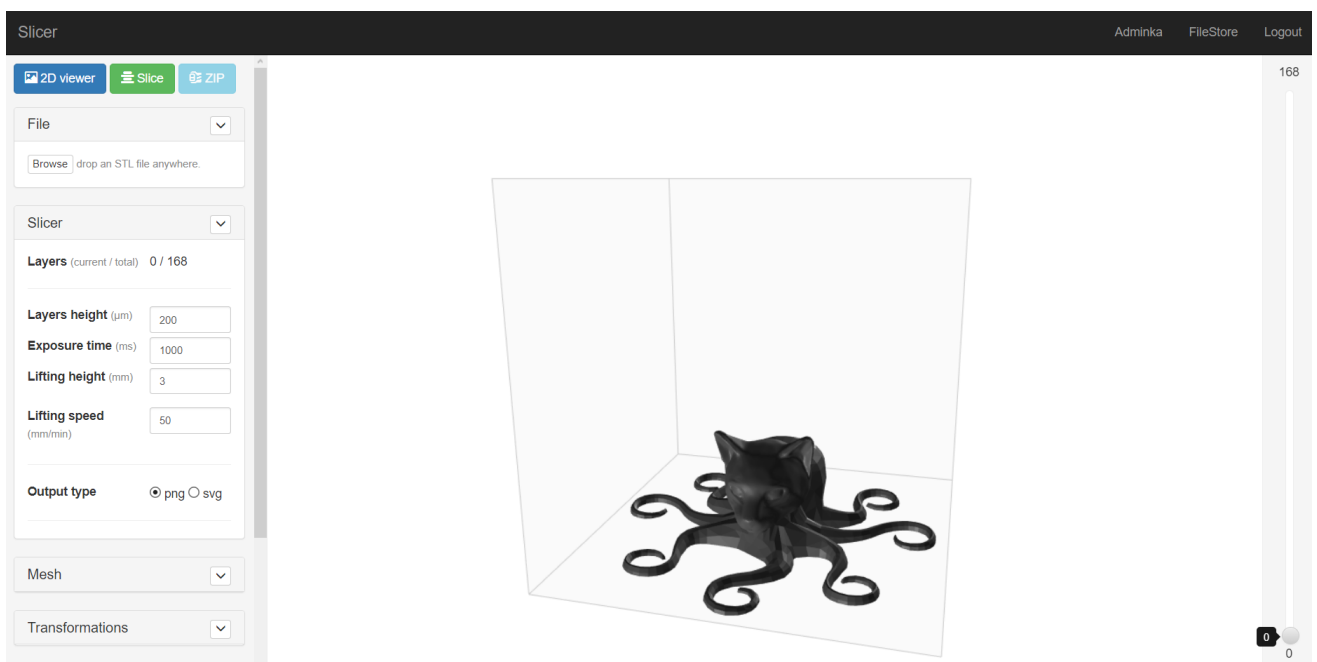


Рисунок 3.2 – Створення елементів

Для елементів “button” додано клас з Bootstrap, після чого стилі до кнопок підтягнулися стилі автоматично. Фрагмент можна побачити нижче продемонстровані стилі для кнопок.

Фрагмент програмного коду стилів кнопки “Success”

```
.btn-success {
  background-image: -webkit-linear-gradient(top, #5cb85c 0%, #419641 100%);
  background-image: -o-linear-gradient(top, #5cb85c 0%, #419641 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#5cb85c),
to(#419641));
  background-image: linear-gradient(to bottom, #5cb85c 0%, #419641 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff5cb85c',
endColorstr='#ff419641', GradientType=0);
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
```

```
background-repeat: repeat-x;
border-color: #3e8f3e;
}
```

В результаті отримали кнопки:

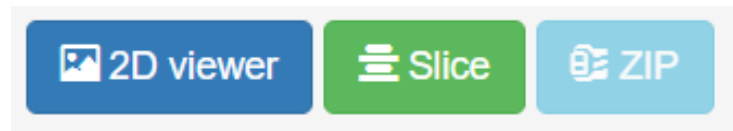


Рисунок 3.3 – Створені кнопки

Таким чином було створено інші елементи панелі параметрів.(рисунок 3.4)

 The image shows a vertical panel with a light gray background. At the top, there are three buttons: '2D viewer' (blue), 'Slice' (green), and 'ZIP' (light blue). Below these is a 'File' section with a dropdown arrow and a 'Browse' button followed by the text 'drop an STL file anywhere.'. The next section is 'Slicer' with a dropdown arrow. Underneath is 'Layers (current / total) 0 / 0'. Then there are several input fields: 'Layers height (μm)' with value '100', 'Exposure time (ms)' with value '1000', 'Lifting height (mm)' with value '3', and 'Lifting speed (mm/min)' with value '50'. Below these are radio button options: 'Output type' with 'png' selected, 'Make ZIP file' with 'yes' selected, and 'Speed mode' with 'no' selected. Finally, there is a 'Speed mode delay (ms)' input field with value '10'.

Рисунок 3.4 - Панель параметрів

Після створення панелі параметрів реалізували полотно canvas за допомогою JavaScript. Фрагмент коду знаходиться нижче.

Фрагмент програмного коду реалізації canvas

```
function Viewer3D(settings) {
  // self alias
  var self = this;

  SLAcer.Viewer.call(self, settings);
  _.defaultsDeep(self.settings, Viewer3D.globalSettings);

  self.controls = new THREE.OrbitControls(self.camera, self.canvas);
  self.controls.addEventListener('change', function() {
    self.render();
  });
  self.controls.noKeys = true;

  self.light = new THREE.AmbientLight(0x000000);
  self.scene.add(self.light);

  self.setBuildVolume(self.settings.buildVolume);

  self.view = new SLAcer.ViewControls({
    target : self.buildVolumeObject,
    controls: self.controls,
    camera : self.camera,
    margin : 10
  });

  var lights = [];
  lights[0] = new THREE.PointLight(0xffffff, 1, 0);
  lights[1] = new THREE.PointLight(0xffffff, 1, 0);
  lights[2] = new THREE.PointLight(0xffffff, 1, 0);
  lights[0].position.set(0, 2000, 0);
  lights[1].position.set(1000, 2000, 1000);
  lights[2].position.set(-1000, -2000, -1000);
  self.scene.add( lights[0] );
  self.scene.add( lights[1] );
  self.scene.add( lights[2] );
  self.setView(this.settings.view);
  self.render();
}
```

Полотно завантажується та до нього підключаються бібліотека `three.js` також бібліотека `tessy.js`, файли реалізації відображення моделі `viewer3d.js`, бібліотека перегляду моделі `OrbitControls.js` повний список бібліотек зображено на рисунку 3.5.

Основна частина коду знаходиться у додатку В.

Фрагмент програмного коду tessy.js

```
var tessy = new libtess.GluTesselator();
// tessy.gluTessProperty(libtess.gluEnum.GLU_TESS_WINDING_RULE,
libtess.windingRule.GLU_TESS_WINDING_POSITIVE);
tessy.gluTessCallback(libtess.gluEnum.GLU_TESS_VERTEX_DATA,
vertexCallback);
tessy.gluTessCallback(libtess.gluEnum.GLU_TESS_BEGIN, begincallback);
tessy.gluTessCallback(libtess.gluEnum.GLU_TESS_ERROR, errorcallback);
tessy.gluTessCallback(libtess.gluEnum.GLU_TESS_COMBINE, combinecallback);
tessy.gluTessCallback(libtess.gluEnum.GLU_TESS_EDGE_FLAG, edgecallback);
```

Viewer3d.js відповідає за відображення полотна canvas та зв'язана з бібліотекою three.js, за допомогою такого зв'язку дозволяє відобразити завантажену користувачем модель на полотні.

Фрагмент програмного коду бібліотеки OrbitControls.js

```
var state = STATE.NONE;
// for reset
this.target0 = this.target.clone();
this.position0 = this.object.position.clone();
// so camera.up is the orbit axis
var quat = new THREE.Quaternion().setFromUnitVectors( object.up, new
THREE.Vector3( 0, 1, 0 ) );
var quatInverse = quat.clone().inverse();
// events
var changeEvent = { type: 'change' };
var startEvent = { type: 'start' };
var endEvent = { type: 'end' };
this.rotateLeft = function ( angle ) {
    if ( angle === undefined ) {
        angle = getAutoRotationAngle();
    }
    thetaDelta -= angle;
};
```

Зв'язок з бібліотекою OrbitControls.js, фрагмент коду зображено вище (код 3.4), дозволяє управляти орбітами та повертати модель для того щоб роздивитися модель з усіх сторін, а tessy.js згладжує модель додаючи більше полігонів.

Таким чином реалізація завантаження та відображення моделі закінчена.



Реалізація слайсингу моделі заключається в використанні та поєднанні бібліотек `earcut.js` та `poly2tri.js`.

Фрагмент програмного коду бібліотеки `earcut.js`

```
function earcut(data, holeIndices, dim) {
  dim = dim || 2;
  var hasHoles = holeIndices && holeIndices.length,
      outerLen = hasHoles ? holeIndices[0] * dim : data.length,
      outerNode = linkedList(data, 0, outerLen, dim, true),
      triangles = [];
  if (!outerNode) return triangles;
  var minX, minY, maxX, maxY, x, y, size;
  if (hasHoles) outerNode = eliminateHoles(data, holeIndices, outerNode, dim);
  // if the shape is not too simple, we'll use z-order curve hash later; calculate polygon
  bbox
  if (data.length > 80 * dim) {
    minX = maxX = data[0];
    minY = maxY = data[1];
    for (var i = dim; i < outerLen; i += dim) {
      x = data[i];
      y = data[i + 1];
      if (x < minX) minX = x;
      if (y < minY) minY = y;
      if (x > maxX) maxX = x;
      if (y > maxY) maxY = y;
    }
  }
}
```

Бібліотека `earcut.js` реалізує модифікований алгоритм нарізання шарів, оптимізований хешуванням кривої z-порядку та розширений для обробки отворів, скручених багатокутників, не гарантує правильності триангуляції, але намагається завжди дати прийнятні результати.

Бібліотека `poly2tri.js` являє собою "Алгоритм розгортки для обмеженої триангуляції Делоне" В. Домітера та Б. Заліка, також відповідає за триангуляцію.

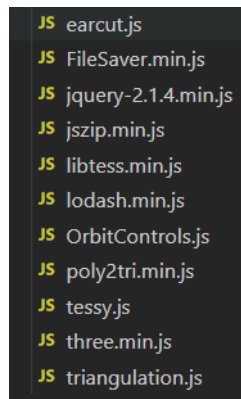


Рисунок 3.10 – Бібліотеки використані для роботи з 3D моделями та збереженням файлів

### 3.3 Використання програмного додатку

В першу чергу додаток будуть використовувати спеціалісти, які займаються друком 3D моделей на 3D принтерах типу DLP. Основна роль додатку полягає в тому, щоб спеціаліст міг проінспектувати необхідну модель на наявність дефектів при друку, а також використати індивідуальні налаштування принтеру для більш точного та якісного аналізу.

Відкривши додаток спеціаліст завантажить необхідну модель (рис.3.12).

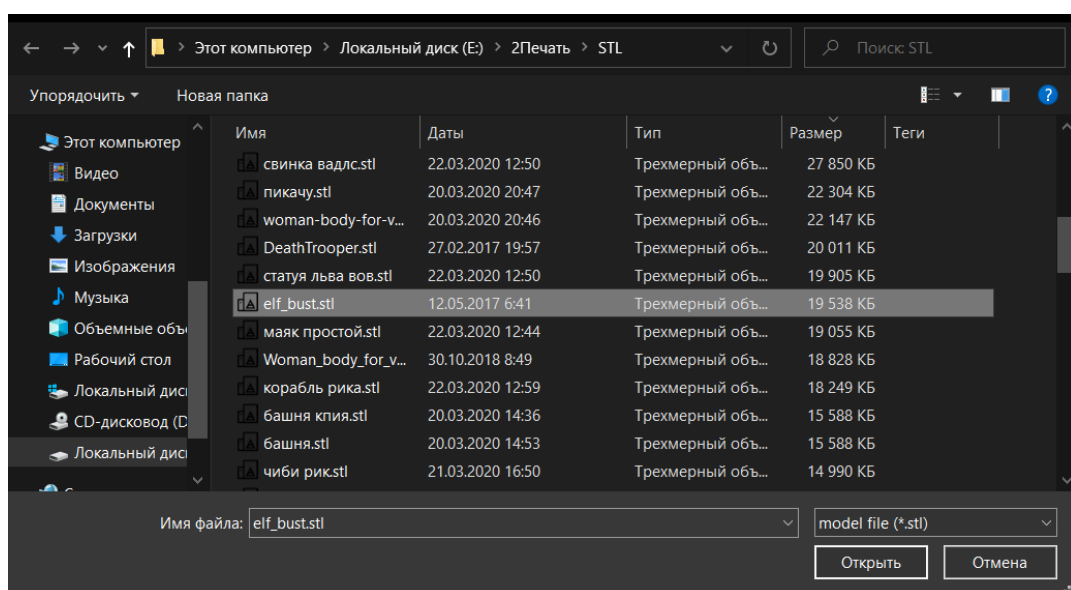


Рисунок 3.12 – Вибір необхідного файлу

Для прикладу було обрано модель створену за допомогою ZBrush (рисунок 3.13).



Рисунок 3.13 – Завантажена модель

Після завантаження необхідно розмістити модель в правильне положення. Для цього в боковій панелі налаштувань є пункт Transformations (рисунок 3.14), за допомогою нього можна віддзеркалити, масштабувати зі збереженням пропорцій або без, перемістити відносно друкуючої площі та повертати відносно будь-якої осі.

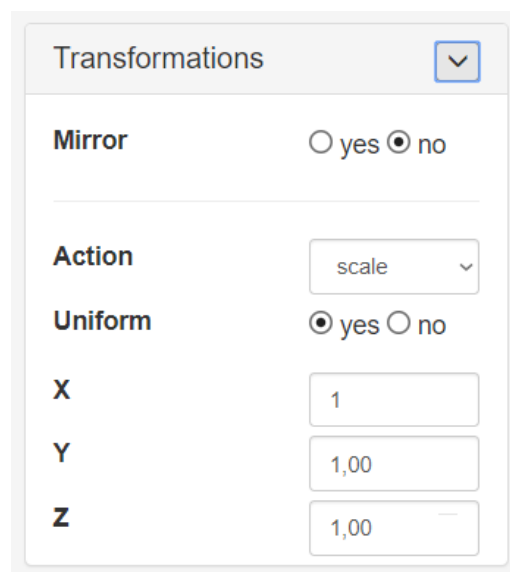


Рисунок 3.14 – Пункт Transformations в панелі налаштувань

Модель було встановлено в необхідне положення за допомогою опції rotate, повернуто на 90 градусів по осі X (рисунок 3.15).

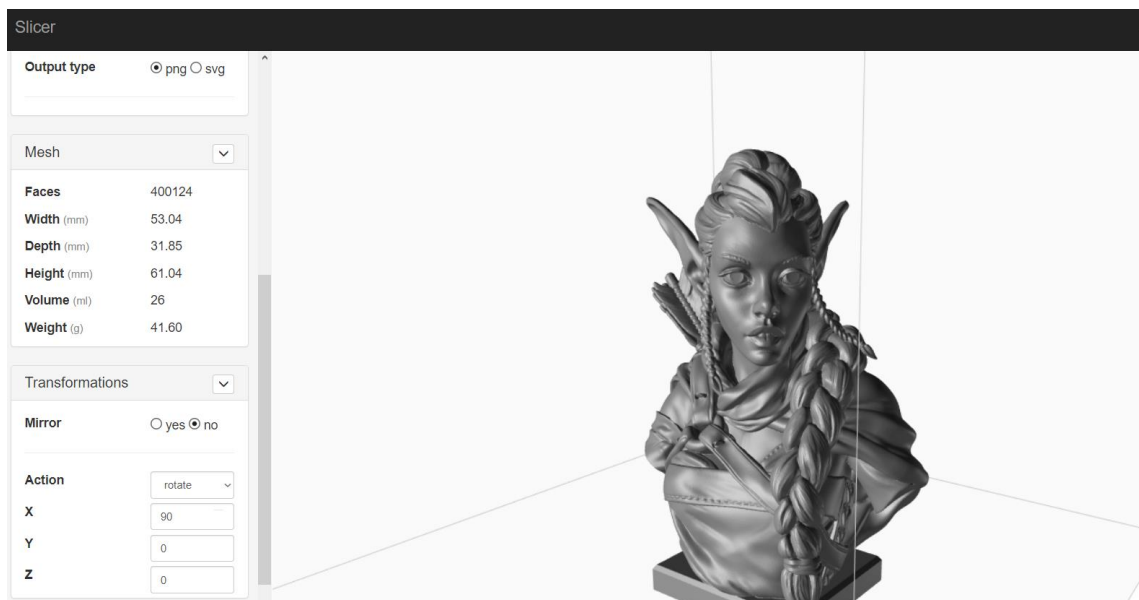


Рисунок 3.15 – Модель після повороту

Закінчивши всі необхідні маніпуляції налаштовуються індивідуальні параметри принтеру, виконується це в пунктах Screen та Build Volume (рисунок 3.16).

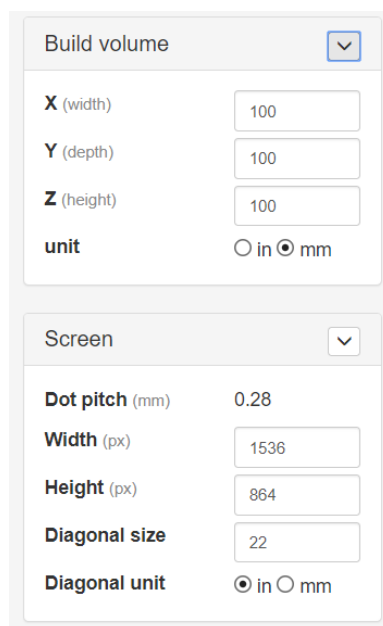


Рисунок 3.16 - Пункти Screen та Build Volume

Пункт Screen відповідає налаштуванням екрану, користувачем вказується діагональ, кількість пікселів по висоті та ширині.

Пункт Build Volume являє собою розміри друкуючої спроможності принтеру.

Останні параметри, які необхідно змінити являються параметри слайсингу в пункті Slicer (рисунок 3.17).

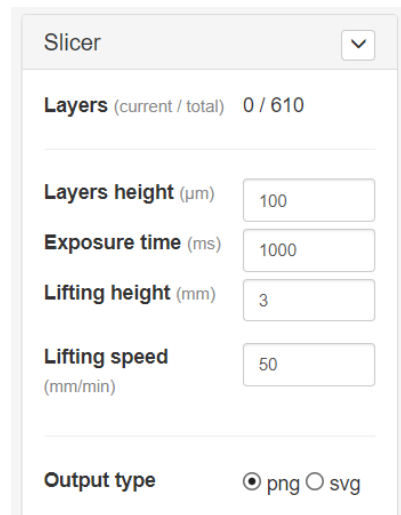


Рисунок 3.17 - Пункт Slicer

Цей пункт містить налаштування висоти шару, час засвічування шару, висота підйому моделі, швидкість підйому та тип вихідних файлів.

Після натискання клавіши Slice відбувається слайсинг моделі, в процесі на моделі відображається проходження кожного шару (рисунок 3.18).

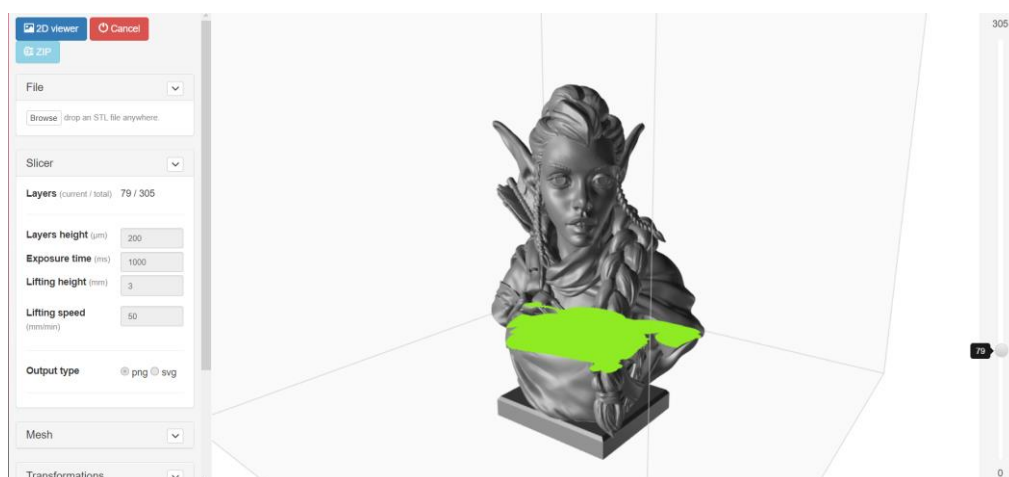


Рисунок 3.18 – Слайсинг моделі

В кінці процесу користувач може вибрати необхідний шар за допомогою панелі справа та переглянути його окремо натиснувши кнопку 2D viewer (рисунок 3.19).

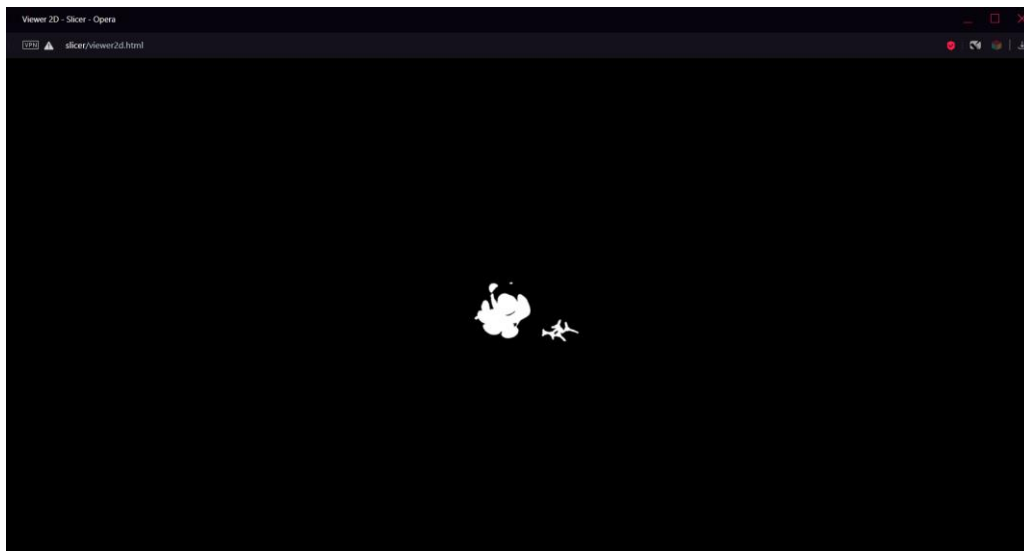


Рисунок 3.19 – Перегляд шару 79 в окремому вікні

Якщо є необхідність завантажити всі шари та переглянути їх, користувач повинен натиснути на кнопку ZIP, додаток упакує всі файли в архів та завантажить користувачу (рисунок 3.20).

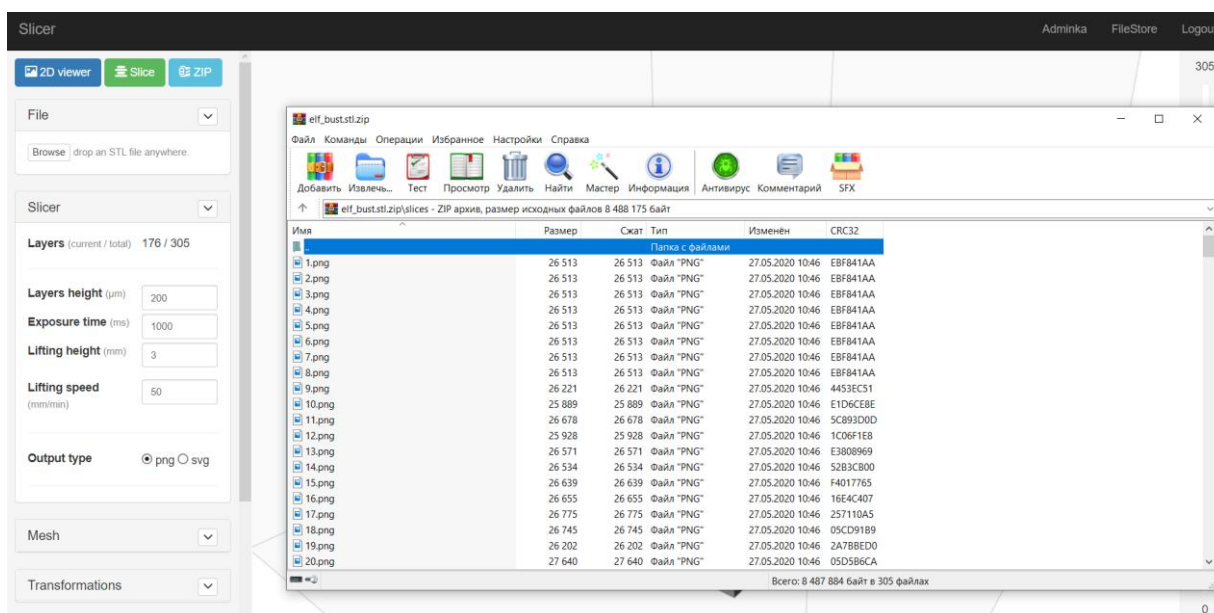


Рисунок 3.20 – Архів даних

Для швидкого доступу до моделей користувач може пройти реєстрацію та в особистому кабінеті завантажувати необхідні моделі (рисунок 3.21).

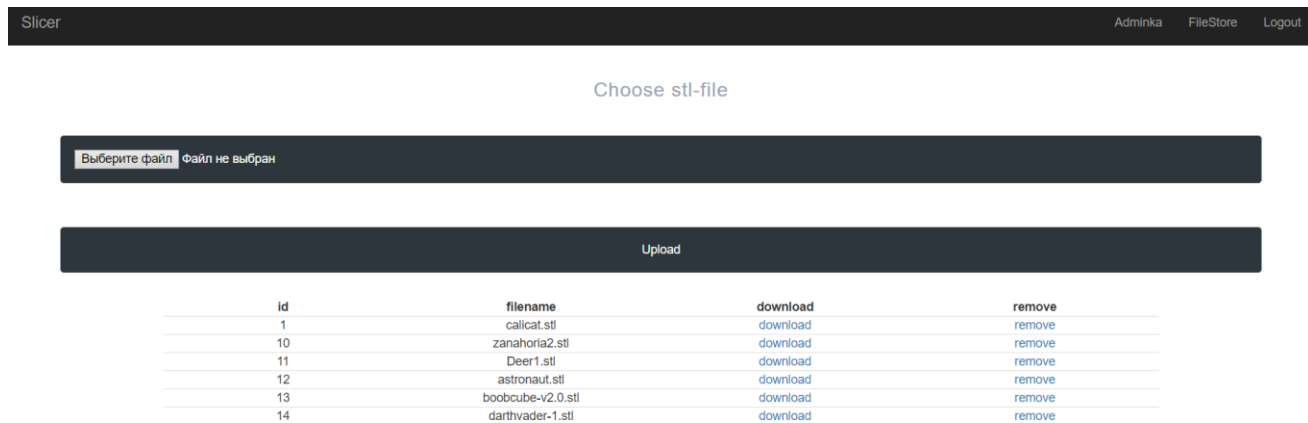


Рисунок 3.21 – Особистий кабінет користувача

Після виконання необхідних корекцій та виправлення помилок для зменшення кількості багів та непередбачуваних результатів реалізація додатку була закінчена.

## ВИСНОВКИ

В ході виконання дипломної роботи було досліджено та проаналізовано головні елементи та функціональні можливості. Мається на увазі методи завантаження та відображення моделі, алгоритми роботи слайсингу моделі та необхідні для цього параметри.

Було вирішено наступні задачі:

- пошук та аналіз інформації згідно тематики кваліфікаційної роботи;
- пошук та аналіз аналогів;
- розроблені діаграми варіантів використання;
- розроблені контекстні діаграми;
- створено структуру задач по створенню продукту, організаційну структуру, матрицю відповідальності, часові межі представлені у вигляді часової шкали та діаграми Ганта, проаналізовано можливі ризики та шляхи їх попередження або усунення (Додаток Б).



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Цифровая светодиодная проекция (dlp) [Электронный ресурс]. – Режим доступа: <https://3ddevice.com.ua/dlp-3d-pechat/> – 17.04.2020р.
- 2 Методология IDEF0 [Электронный ресурс]. – Режим доступа: <https://itteach.ru/bpwin/metodologiya-idef0> – 17.04.2020р. – Методология IDEF0.
- 3 Технологии DLP и SLA и что есть качество [Электронный ресурс]. – Режим доступа: <https://itteach.ru/bpwin/metodologiya-idef0> – 17.04.2020р.
- 4 Responsibility Assignment Matrix [Электронный ресурс]. – Режим доступа: <http://acqnotes.com/acqnote/careerfields/responsibility-assignment-matrix1> – 19.04.2020р. – Responsibility Assignment Matrix (RAM) - AcqNotes.
- 5 DLP [Электронный ресурс]. – Режим доступа: <http://www.3d-format.ru/technologies/dlp/> – 19.04.2020р.
- 6 Отличия, плюсы и минусы - технологии 3d печати [Электронный ресурс]. – Режим доступа: <https://getfab.ru/post/620/> – 18.04.2020р.
- 7 3D-принтеры: зачем они нужны и как они работают [Электронный ресурс]. – Режим доступа: <https://zoom.cnews.ru/publication/item/54392> – 18.04.2020р.
- 8 Принцип работы 3d-принтеров [Электронный ресурс]. – Режим доступа: <http://3dpr.ru/printsip-raboty-3d-printera> – 02.01.2020р.
- 9 Возможности применения 3d-принтеров [Электронный ресурс]. – Режим доступа: <http://3dpr.ru/primenenie-3d-printerov> – 01.04.2019.
- 10 Сравнение технологий 3d-печати [Электронный ресурс]. – Режим доступа: <http://3dpr.ru/sravnenie-tehnologij-3d-pechati> – 18.01.2020р.
- 11 Цифровая светодиодная проекция (DLP) [Электронный ресурс]. – Режим доступа: <https://3dtoday.ru/blogs/inliner/build-a-diy-dlp-printer-entry-the-choice-of-projector-translated-artic> – 18.04.2020р.
- 12 Алгоритм слайсингу [Электронный ресурс]. – Режим доступа: <https://github.com/substack/point-in-polygon/blob/master/index.js> – 18.04.2018р.

- 13 Э. Кэнесс, К. Фонда. Low-cost 3D Printing for Science, Education and Sustainable Development / М. Дзеннаро, 2017 – 288с.
- 14 LittleTinyH Books. 3D печать. Коротко и максимально ясно, 2016 – 312с.
- 15 Tactile Picture Books Project, 3D – Книги для детей с недостатками зрения, 2019
- 16 Горьков Дмитрий, 3D-печать в малом бизнесе, 2015
- 17 ГЭОТАР-Медиа, 3D-печать в медицине, 2019 – 240с.
- 18 Горьков Дмитрий, Первые шаги в 3D-печати, 2018 – 267с.
- 19 Рэдвуд Бен / Гаррэт Брайан / Шофер Филемон, 3D Printing Gold Rush: How to Profit from 3D Printing – The Next Technology Revolution, 2018
- 20 Рэдвуд Бен / Гаррэт Брайан / Шофер Филемон, 3D-печать. Практическое руководство, 2020

## **ДОДАТОК А**

**ТЕХНІЧНЕ ЗАВДАННЯ**  
**на розробку web-додатку «Програмний додаток для налаштування моделі**  
**перед друком на 3D-принтері»**

**Суми 2020**

# **1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ WEB-ДОДАТКУ**

## **Призначення web-додатку**

Даний програмний додаток повинен представляти web-додаток slicer 3d моделей. Основне призначення даного додатку – це аналіз моделей для друку, виявлення помилок моделі, можливість переглянути кожний шар, та детально проаналізувати модель пошарово.

## **Мета створення web-додатку**

Можливість завантажити модель в web-додаток, використовуючи параметри зробити слайсинг моделі, зберегти слайси в форматі svg, png.

## **Цільова аудиторія**

До цільової аудиторії web-додаток можна віднести наступні групи:

1. Проектувальники.
2. 3D моделери.
3. Макетчик.
4. Інші зацікавлені сторони.

# **2 ВИМОГИ ДО WEB-ДОДАТКУ**

## **Вимоги до структури й функціонування web-додатку**

Програмний додаток має бути реалізована у вигляді web-додатку із зручним та зрозумілим у використанні інтерфейсом, це дозволить швидко створити слайсинг моделі. Додаток повинен мати весь необхідний функціонал для правильної роботи.

### **Вимоги до персоналу**

Для роботи з web-додатком потрібно мати навички побудови 3d моделей, мати досвід роботи з 3D принтером.

### **Вимоги до збереження інформації**

Інформація про користувачів має зберігатися в базі даних. У програмному додатку має бути можливість завантаження 3d моделей користувача для швидкого доступу до необхідних ресурсів.

Моделі, які пройшли етап слайсингу можна зберегти в форматі svg, png.

### **Вимоги до розмежування доступу**

Оскільки програмний додаток представлена у вигляді web-додатку, то доступ матимуть декілька груп для кожної впроваджені певні обмеження.

Користувачів програмний додаток можна розділити на 4 групи:

1. Користувачі, які не зареєстрували аккаунт.
2. Користувачі, які мають аккаунт.
3. Адміністратор.

Користувачі, які не зареєстрували аккаунт можуть користуватися ресурсом але не зможуть завантажити свої моделі, в цілому такі користувачі мають майже повноцінний доступ до ресурсу.

Після реєстрації користувачі отримують змогу завантажувати свої моделі та ділитися ними з іншими користувачами. Користувачам надається доступ до особистого кабінету, де вони мають можливість переглядати моделі, завантажувати нові або видаляти непотрібні.

Після авторизації адміністратор має змогу проводити операції з базою даних, тобто слідкувати за реєстрацію аккаунтів, та доступним дисковим простором, очищати дисковий простір та управляти аккаунтами користувачів.

## 2.1 Вимоги до функцій

### Основні вимоги

#### Структура web-додатку

Дана додаток повинен містити.

Головна сторінка має область перегляду моделі та бокове меню за допомогою якого користувач може налаштувати параметри слайсингу та зберегти модель в своєму аккаунті, якщо користувач пройшов реєстрацію.

Особистий кабінет в якому користувач може переглянути список завантажених ним моделей.

#### Навігація

Інтерфейс додатку повинен бути простий і зрозумілий таким чином користувач одразу буде знати, які можливості надає додаток.

Навігація додатку має надати користувачеві доступ до необхідних розділів з відповідною інформацією. Посилання на сторінки мають мати заголовки та загальноприйняті позначки.

#### Наповнення (контент)

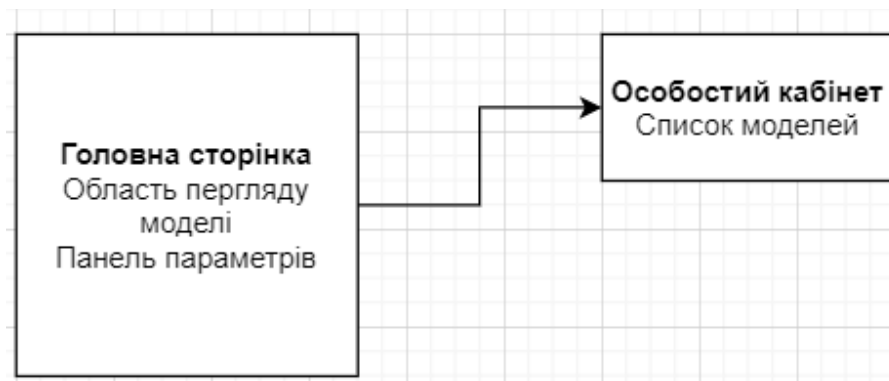
Область перегляду моделі та панель налаштування являють собою основний контент також особистий кабінет користувача за допомогою web-інтерфейсу.

Параметри, які може задавати користувач:

- висота шару;
- час засвічування шару;
- висоту підйому;
- швидкість підйому;
- формат вихідних файлів;
- збереження файлів в архіві;
- колір моделі;
- трансформація моделі.

## Система навігації (карта)

Зв'язок розділів в додатку представлено на рисунку А.1.



**Рисунок А.1 – Карта додатку**

## Вимоги до функціональних можливостей

Web-додаток повинен надавати можливість авторизації, це надає користувачам певні привілежії.

### Функціональні можливості розділів

Головна сторінка містить такий функціонал:

- Завантаження моделі;
- Перегляд моделі;
- Авторизація користувачів;
- Слайсинг моделі;
- Параметри слайсингу.

Не авторизований користувач може завантажити модель, переглянути її, виявити дефекти моделі, зробити слайсинг, а також завантажити результат слайсингу. Також є можливість підрахувати вартість виготовлення моделі та кількість пластику витраченого на виготовлення.

Авторизований користувач матиме можливість доступу до особистого кабінету, тобто змогу завантажувати моделі для швидкого доступу.

Обмеження на завантажувані моделі:

- об'єм файлу не більше 100мб;
- формат файлу stl;
- назва файлу не більше 64 символів.

Таблиця 1 – Потреби користувача

<b>ID</b>	<b>Потреби користувача</b>	<b>Джерело</b>
UN-01	Завантаження моделі користувача	Клієнт
UN-02	Перегляд моделі	Клієнт
UN-03	Налаштування висоти шару	Клієнт
UN-04	Налаштування часу засвічування	Клієнт
UN-05	Налаштування швидкості підйому	Клієнт
UN-06	Збереження вихідних файлів	Клієнт
UN-07	Збереження в архів	Клієнт
UN-08	Масштабування моделі	Клієнт
UN-09	Поворот моделі	Клієнт
UN-10	Розміщення моделі	Клієнт
UN-11	Редагування розмірів друкуючої платформи	Клієнт
UN-12	Задання параметрів екрану	Клієнт
UN-13	Вибір кольору моделі та шару	Клієнт

Таблиця 2 – Системні вимоги

<b>ID</b>	<b>Системні вимоги</b>	<b>Опис</b>
SR-01	Наявність модуля завантаження моделі	Надає завантажити свою модель в форматі stl

Продовження таблиці 2

<b>ID</b>	<b>Системні вимоги</b>	<b>Опис</b>
SR-01	Перегляд 3D моделі	Забезпечує перегляд завантаженої моделі



SR-01	Модуль параметрів для слайсингу	Містить параметри, які змінюють модель та впливають на слайсинг
SR-01	Модуль перегляду шару	Надає можливість переглянути окремий шар моделі в форматі 2D
SR-01	Модуль сітки	Забезпечує зчитування сітки моделі з файлу

### Загальні вимоги

Web-додаток повинна мати повністю робочий функціонал. Дизайн додатку повинен містити мінімалістичний стиль. Приклад розташування елементів головної сторінки зображено на рисунку А.2.



Рисунок А.2 – Головна сторінка

### Типові навігаційні та інформаційні елементи

- Авторизація/Реєстрація
- Основне поле контенту.

### **Авторизація/Реєстрація**

Вікно авторизації/реєстрації користувача повинне викликатися при натисканні кнопки авторизації/реєстрації.

### **Основне поле контенту**

Основне поле контенту має бути розташоване в центрі сторінки додатка. У цьому полі має відображатися зміст розділу відповідно до певного розділу.

## **2.2 Вимоги до видів забезпечення**

### **Вимоги до інформаційного забезпечення**

Реалізація додатку відбувається з використанням:

- MySQL 5.7;
- PHP 7.4.4;
- CSS3;
- HTML 5.0;
- JavaScript.

### **Вимоги до лінгвістичного забезпечення**

Мова додатку повинна бути українською.

### **Вимоги до програмного забезпечення**

Користування даним додатком можливе лише при підключенні до мережі Інтернет.

Для роботи з web-додатком програмне забезпечення повинне задовольняти таким вимогам:

- Браузер Opera 36, Chrome 44.0.2403.157, Mozilla Firefox 73.0.1.

### 3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ САЙТУ

Докладний опис етапів роботи можна побачити в табл. А.1.

**Таблиця А.1 – Поетапне створення додатку**

№	Склад і зміст роботи	Строк розробки (у робочих днях)
1	Аналіз принципів побудови моделей Аналіз десктопних додатків та їх функціоналу	5
2	Аналіз функціональних та нефункціональних вимог	2
3	Створення макету головної сторінки та макету авторизації/реєстрації	2
4	Реалізація макету Тестування та виправлення помилок.	7
5	Розробка БД Реалізація функціоналу Тестування	10
6	Фінальне тестування та виправлення помилок	2
7	Оформлення ПЗ Створення інструкції для користувача	2
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	30

#### **4 ВИМОГИ ДО СКЛАДУ Й ЗМІСТУ РОБІТ ІЗ ВВЕДЕННЯ ДОДАТКУ В ЕКСПЛУАТАЦІЮ**

Для повноцінного функціонування додаток повинен знаходитися на хостингу.

На хостинг будуть перенесені файли, які містять код який реалізує необхідний функціонал зазначений в ТЗ.

## ДОДАТОК Б

### ПЛАНУВАННЯ РОБІТ

#### 1 Ідентифікація ідеї проекту

В основі 3D - друку лежить принцип пошарового нанесення будь-якого матеріалу за допомогою лазерів та інших приладів.

Такий підхід відрізняє ряд найважливіших переваг перед традиційними техніками виробництва. Він створює об'єкти будь-якої геометрії без необхідності в численних допоміжних інструментах на кшталт ливарних форм. Іншими словами, якщо підприємець вирішує змінити зовнішній вигляд товару, йому більше не потрібно міняти всю лінію виробництва. Нова ціна питання - кілька правок, внесених в комп'ютерну модель. Для того щоб це зробити необхідно мати спеціальне програмне забезпечення, яке допоможе створити спеціальну модель.

Метою проекту є розроблення Web-додатку для створення спеціалізованих моделей для 3D-друку.

#### 2 Деталізація мети методом SMART

**S.** Розробити Web-додаток який перетворює 3D-модель в спеціальний формат, тобто перетворення моделі в машинний код

**M.** Результатом виконання проекту, має бути Web-додаток, отримати доступ до якого, можна ввівши адресу сайту до адресного рядку, в свою чергу, Web-додаток мусить бути розміщений на хостингу, пов'язані з ним бази даних(БД), форми та модулі мають бути вірно налаштовані та під'єднані.

**А.** Проект потребує розробника із знанням HTML, CSS, мови програмування JavaScript, PHP, MySQL, яке використовується для розроблення даного додатку.

**Р.** Мету реально досягнути, так як розроблення Web-додатку здійснюється за допомогою Visual Studio Code, не вимагає надзвичайно складних дій або велику кількість ресурсів.

**Т.** Обмеженість в часі зумовлена рішенням замовника, щоб отримати продукт проекту згідно плану.

### 3 Описання фази розробки ІТ—проекту

#### 3.1 Планування змісту структури робіт ІТ—проекту (WBS)

Структурна декомпозиція робіт (work breakdown structure, WBS) - це ієрархічна структура робіт, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

Частина WBS представлена на рисунку Б.1. Повна версія знаходиться в кінці звіту.

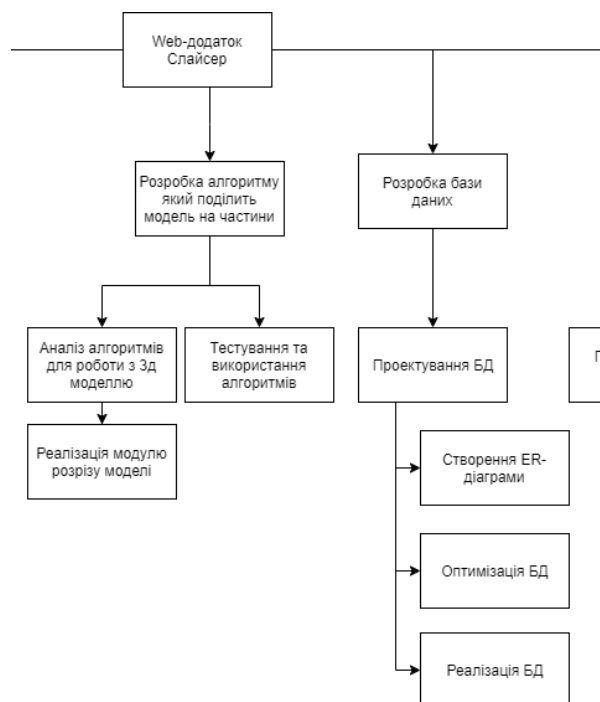


Рисунок Б.1 – WBS структура проекту

### 3.2 Планування структури організації, для впровадження готового проекту (OBS)

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту. На верхньому рівні OBS проекту знаходиться керівник та команда управління проектом; на наступному рівні – виконавці. Останнім рівнем OBS-структури є відповідальні особи виконавців. Це не обов'язково повинні бути керівники, а ті співробітники, яким доручено безпосередньо організувати і відповідати перед виконавцем за виконання конкретного елемента WBS-структури.

Частина OBS структури представлена на рисунку Б.2. Повна версія знаходиться в кінці звіту.

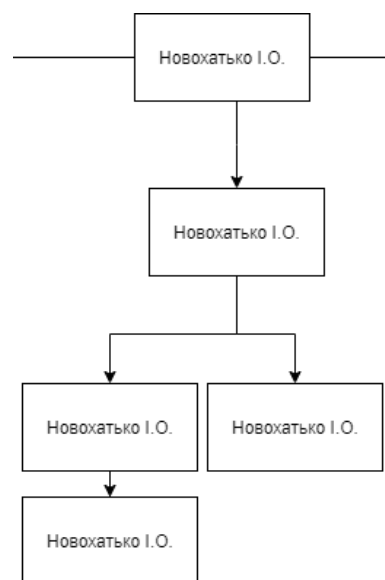


Рисунок 2.2 – OBS структура

### 3.3 Побудова матриці відповідальності (виконавців пакетів робіт)

Матриця відповідальності (Responsibility Assignment Matrix) забезпечує опис і узгодження структури відповідальності за виконання пакетів робіт. Вона

являє собою форму опису розподілу відповідальності за реалізацію робіт проекту із зазначенням ролі кожного з виконавців. Будується на основі WBS та OBS.

У таблиці Б.1 показано матрицю відповідальності проекту.

Таблиця Б.1 – RAM

			Новохатько	Шендрік	Парфененко
	Deliverable or Task		Project Team		
1	<b>Аналіз способів перегляду 3D моделей</b>		I		
2	Аналіз алгоритмів для перегляду 3д моделей		I		
3	Аналіз типів файлів, які можна використати		A	R	I
4	Реалізація модулю роботи з файлами		A		R
5	Тестування та вибір кращого алгоритму		A	I	
6	<b>Додання моделей різних форматів</b>		A	I	R
7	Аналіз способів реалізації		A		R
8	Реалізація модулю розрізу моделі		A		R
9	Тестування та використання алгоритмів		A	I	
10	<b>Аналіз алгоритмів для роботи з 3д моделлю</b>		A	R	
11	Реалізація модулю розрізу моделі		A		R
12	Тестування та використання алгоритмів		A	I	
13	<b>Проектування БД</b>		A	I	R



## Продовження таблиці Б.1

			Новохатько І.О.	Шендрік Віра	Парфененко Юлія
	Deliverable or Task		Project Team		
13	<b>Проектування БД</b>		A	I	R
14	Створення ER-діаграми		A	R	
15	Оптимізація БД		A		R
16	Реалізація БД		A	I	
17	<b>Проектування дизайну</b>		A		R
18	Створення дизайну додатку		A		R
19	Тестування елементів		A	I	
20	Оптимізація елементів додатку		A	I	R
21	Планування методів покращення		A	R	
22	Кінцеве тестування		A		R
23	Залиття додатку на хостинг		A	I	

## 4 Побудова календарного графіку виконання ІТ—проекту (включаючи побудову часткових мережевих моделей у вигляді діаграм Ганта)

Діаграма Ганта (а Gantt Chart) - це візуальний спосіб відображення запланованих завдань. Горизонтальні графіки широко використовуються для планування проектів будь-яких розмірів в різних галузях і сферах. Це зручний спосіб показати, яка робота планується до виконання в певний день і час. Gantt Charts також допомагають командам і менеджерам проектів контролювати дати початку і закінчення будь-якого проекту. Все в одному просторі.

Діаграма Ганта представлена на рисунку 2.5

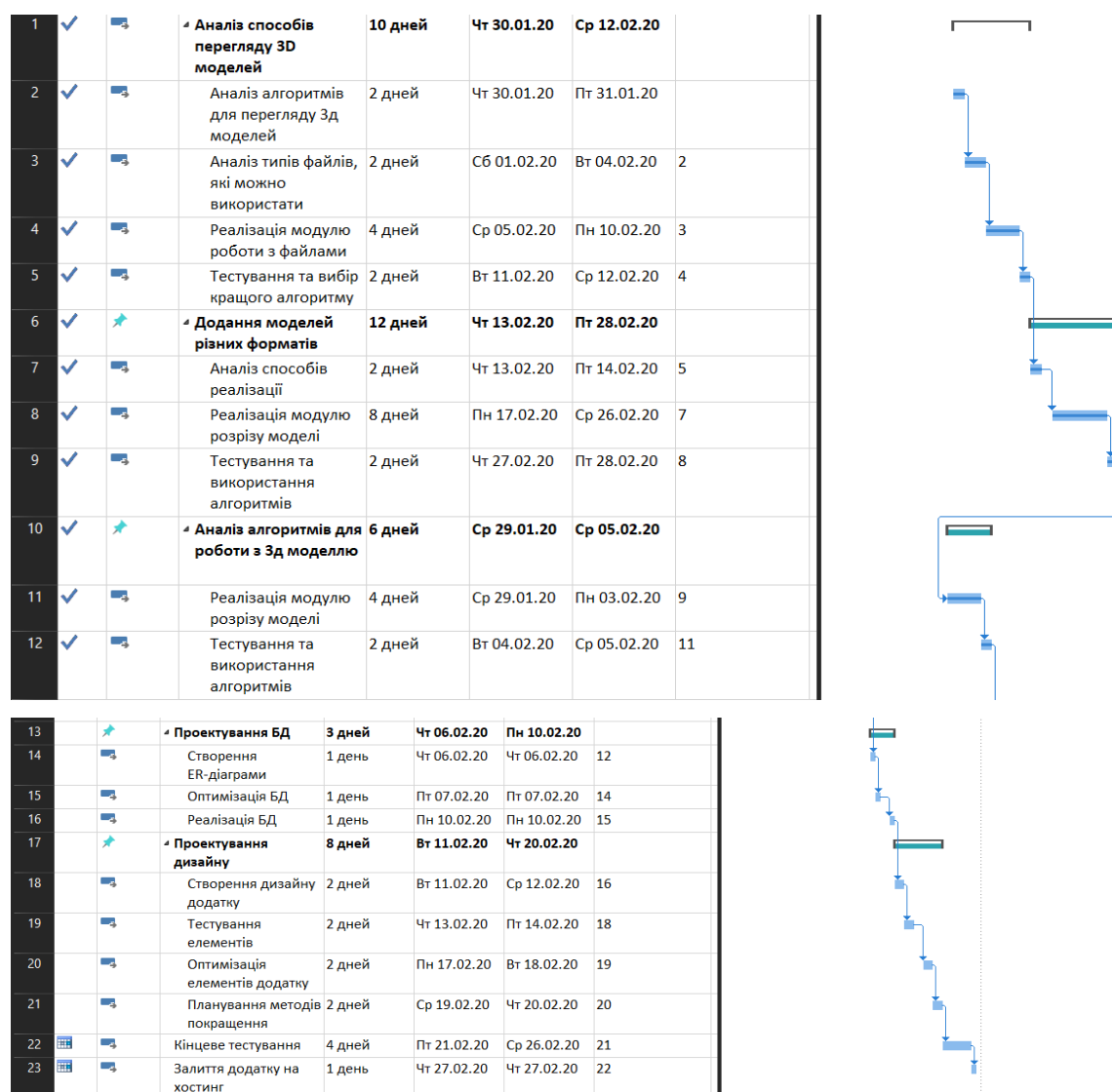


Рисунок 4 – Діаграма Ганта

## 5 Ідентифікація ризиків

Ризик – це імовірна подія, яка у випадку своєї появи негативно або позитивно вплине на проект.

Процес управління ризиками включає наступні етапи:

1. ідентифікація
2. процес оцінювання ризиків, який включає в себе якісний, кількісний аналіз.
3. заходи реагування на ризики
4. моніторинг заходів і ризиків

Ризики проекту представлені у таблиці Б.2

Таблиця Б.2 – Risk Register

№	Risk description	Impact	Probability	RV	Mitigation
1	Project is not well-defined	4	2	M	Complete a business case if not already provided and ensure purpose is well defined on Project Charter and PID.
2	Unplanned work to be performed at a late stage of the project	4	3	M	1 Document all assumptions made in planning and communicate to the project manager before project kick off
3	Pressure to arbitrarily reduce task durations and or run tasks in parallel which would increase risk of errors.	3	1	L	Show project plan to interested parties and patiently explain that a reduction in lead time will lead to unavoidable errors and poor product quality.
4	Low customer involvement	2	1	L	Explain to the customer that the higher his involvement

					in the project, the better product he will receive.
5	Difficult or impossible tasks	5	3	H	Upgrade staff or replace / exclude a task

## Рівні критичності

- 1 Negligible
- 2 Low
- 3 Medium
- 4 High
- 5 Critical

Таблиця Б.3 – Probability / Impact Matrix

5					
4					
3				2	5
2		4		1	
1			3		
Probability / Impact	1	2	3	4	5

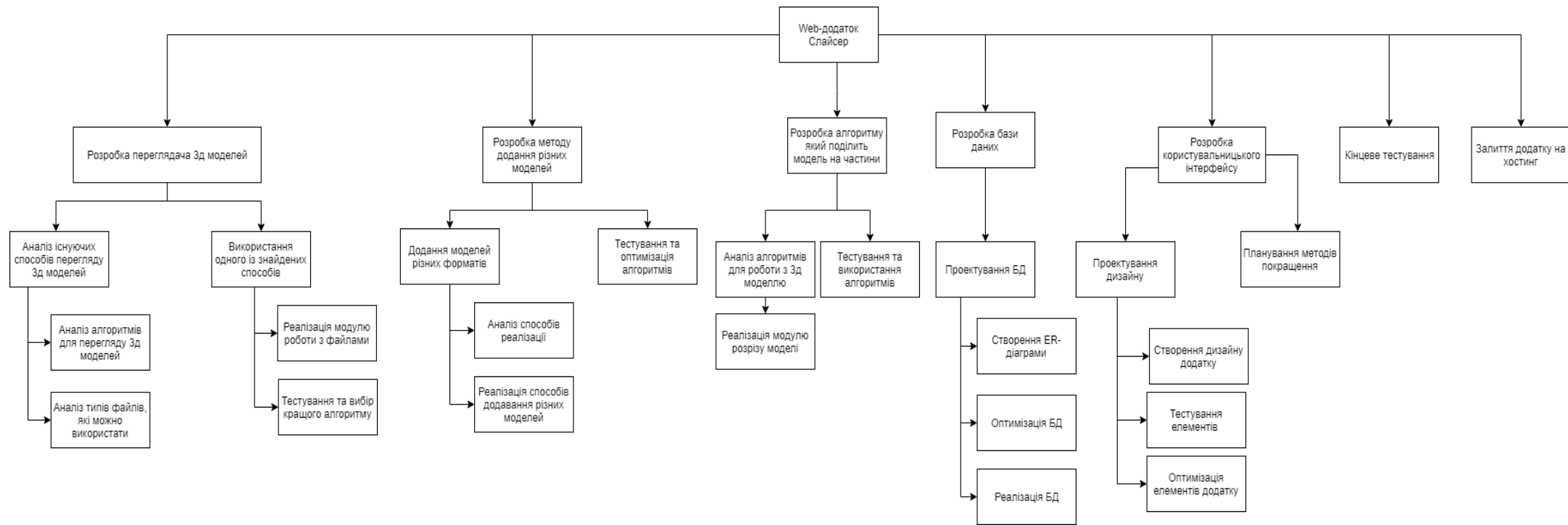


Рисунок Б.1 – WBS діаграма

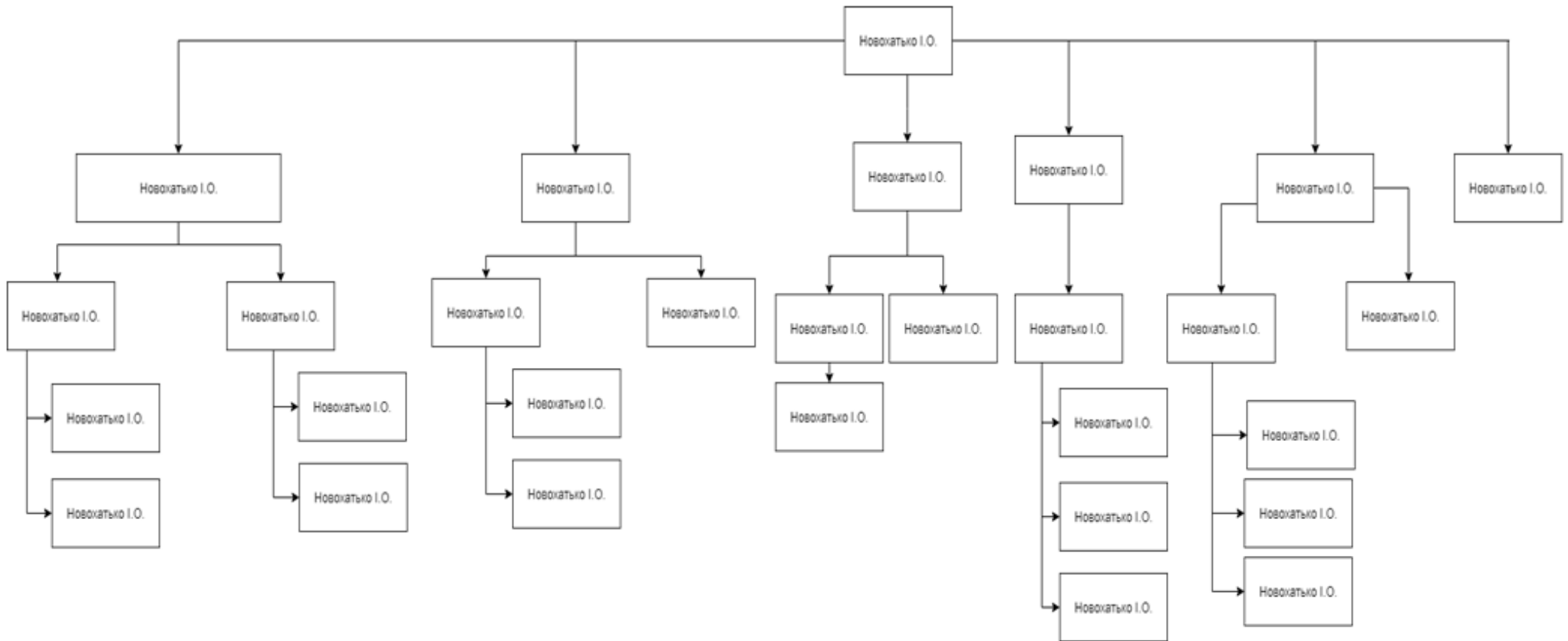


Рисунок Б.2 – OBS діаграмма

## ДОДАТОК В

### ЛІСТИНГ ПРОГРАМНОГО КОДУ

#### OrbitControls.js

```
THREE.OrbitControls = function ( object, domElement ) {  
    this.object = object;  
    this.domElement = ( domElement !== undefined ) ? domElement : document;  
    // API  
    // Set to false to disable this control  
    this.enabled = true;  
    // "target" sets the location of focus, where the control orbits around  
    // and where it pans with respect to.  
    this.target = new THREE.Vector3();  
    // center is old, deprecated; use "target" instead  
    this.center = this.target;  
    // This option actually enables dollying in and out; left as "zoom" for  
    // backwards compatibility  
    this.noZoom = false;  
    this.zoomSpeed = 1.0;  
    // Limits to how far you can dolly in and out  
    this.minDistance = 0;  
    this.maxDistance = Infinity  
    // Set to true to disable this control  
    this.noRotate = false;  
    this.rotateSpeed = 1.0;  
    // Set to true to disable this control  
    this.noPan = false;  
    this.keyPanSpeed = 7.0; // pixels moved per arrow key push  
    // Set to true to automatically rotate around the target  
    this.autoRotate = false;  
    this.autoRotateSpeed = 2.0; // 30 seconds per round when fps is 60  
    // How far you can orbit vertically, upper and lower limits.  
    // Range is 0 to Math.PI radians.  
    this.minPolarAngle = 0; // radians
```

```

this.maxPolarAngle = Math.PI; // radians
// How far you can orbit horizontally, upper and lower limits.
// If set, must be a sub-interval of the interval [ - Math.PI, Math.PI ].
this.minAzimuthAngle = - Infinity; // radians
this.maxAzimuthAngle = Infinity; // radians
// Set to true to disable use of the keys
this.noKeys = false;
// The four arrow keys
this.keys = { LEFT: 37, UP: 38, RIGHT: 39, BOTTOM: 40 };
// Mouse buttons
this.mouseButtons = { ORBIT: THREE.MOUSE.LEFT, ZOOM: THREE.MOUSE.MIDDLE, PAN: THREE.MOUSE.RIGHT
};

//////////
// internals
var scope = this;
var EPS = 0.000001;
var rotateStart = new THREE.Vector2();
var rotateEnd = new THREE.Vector2();
var rotateDelta = new THREE.Vector2();
var panStart = new THREE.Vector2();
var panEnd = new THREE.Vector2();
var panDelta = new THREE.Vector2();
var panOffset = new THREE.Vector3();
var offset = new THREE.Vector3();
var dollyStart = new THREE.Vector2();
var dollyEnd = new THREE.Vector2();
var dollyDelta = new THREE.Vector2();
var theta;
var phi;
var phiDelta = 0;
var thetaDelta = 0;
var scale = 1;
var pan = new THREE.Vector3();
var lastPosition = new THREE.Vector3();
var lastQuaternion = new THREE.Quaternion();
var STATE = { NONE : -1, ROTATE : 0, DOLLY : 1, PAN : 2, TOUCH_ROTATE : 3, TOUCH_DOLLY : 4, TOUCH_PAN : 5
};

var state = STATE.NONE;
// for reset
this.target0 = this.target.clone();
this.position0 = this.object.position.clone();

```



```

// so camera.up is the orbit axis
var quat = new THREE.Quaternion().setFromUnitVectors( object.up, new THREE.Vector3( 0, 1, 0 ) );
var quatInverse = quat.clone().inverse();
// events
var changeEvent = { type: 'change' };
var startEvent = { type: 'start' };
var endEvent = { type: 'end' };
this.rotateLeft = function ( angle ) {
    if ( angle === undefined ) {
        angle = getAutoRotationAngle();
    }
}

```

Earcut.js

```

function earcut(data, holeIndices, dim) {
    dim = dim || 2;
    var hasHoles = holeIndices && holeIndices.length,
        outerLen = hasHoles ? holeIndices[0] * dim : data.length,
        outerNode = linkedList(data, 0, outerLen, dim, true),
        triangles = [];
    if (!outerNode) return triangles
    var minX, minY, maxX, maxY, x, y, size;
    if (hasHoles) outerNode = eliminateHoles(data, holeIndices, outerNode, dim);
    // if the shape is not too simple, we'll use z-order curve hash later; calculate polygon bbox
    if (data.length > 80 * dim) {
        minX = maxX = data[0];
        minY = maxY = data[1];
        for (var i = dim; i < outerLen; i += dim) {
            x = data[i];
            y = data[i + 1];
            if (x < minX) minX = x;
            if (y < minY) minY = y;
            if (x > maxX) maxX = x;
            if (y > maxY) maxY = y;
        }
        // minX, minY and size are later used to transform coords into integers for z-order calculation
        size = Math.max(maxX - minX, maxY - minY);
    }
    earcutLinked(outerNode, triangles, dim, minX, minY, size);
    return triangles;
}

```

```

// create a circular doubly linked list from polygon points in the specified winding order
function linkedList(data, start, end, dim, clockwise) {
  var i, last;
  if (clockwise === (signedArea(data, start, end, dim) > 0)) {
    for (i = start; i < end; i += dim) last = insertNode(i, data[i], data[i + 1], last);
  } else {
    for (i = end - dim; i >= start; i -= dim) last = insertNode(i, data[i], data[i + 1], last);
  }
  if (last && equals(last, last.next)) {
    removeNode(last);
    last = last.next;
  }
  return last;
}

// eliminate colinear or duplicate points
function filterPoints(start, end) {
  if (!start) return start;
  if (!end) end = start;
  var p = start,
      again;
  do {
    again = false;
    if (!p.steiner && (equals(p, p.next) || area(p.prev, p, p.next) === 0)) {
      removeNode(p);
      p = end = p.prev;
      if (p === p.next) return null;
      again = true;
    } else {
      p = p.next;
    }
  } while (again || p !== end);
  return end;
}

// main ear slicing loop which triangulates a polygon (given as a linked list)
function earcutLinked(ear, triangles, dim, minX, minY, size, pass) {
  if (!ear) return;
  // interlink polygon nodes in z-order
  if (!pass && size) indexCurve(ear, minX, minY, size);
  var stop = ear,
      prev, next;
  // iterate through ears, slicing them one by one

```

```

while (ear.prev !== ear.next) {
  prev = ear.prev;
  next = ear.next;
  if (size ? isEarHashed(ear, minX, minY, size) : isEar(ear)) {
    // cut off the triangle
    triangles.push(prev.i / dim);
    triangles.push(ear.i / dim);
    triangles.push(next.i / dim);
    removeNode(ear);
    // skipping the next vertice leads to less sliver triangles
    ear = next.next;
    stop = next.next;
    continue;
  }
  ear = next;
  // if we looped through the whole remaining polygon and can't find any more ears
  if (ear === stop) {
    // try filtering points and slicing again
    if (!pass) {
      earcutLinked(filterPoints(ear), triangles, dim, minX, minY, size, 1);

      // if this didn't work, try curing all small self-intersections locally
    } else if (pass === 1) {
      ear = cureLocalIntersections(ear, triangles, dim);
      earcutLinked(ear, triangles, dim, minX, minY, size, 2);
      // as a last resort, try splitting the remaining polygon into two
    } else if (pass === 2) {
      splitEarcut(ear, triangles, dim, minX, minY, size);
    }
    break;
  }
}
}

// check whether a polygon node forms a valid ear with adjacent nodes
function isEar(ear) {
  var a = ear.prev,
      b = ear,
      c = ear.next;
  if (area(a, b, c) >= 0) return false; // reflex, can't be an ear
  // now make sure we don't have other points inside the potential ear
  var p = ear.next.next;

```

```

while (p !== ear.prev) {
  if (pointInTriangle(a.x, a.y, b.x, b.y, c.x, c.y, p.x, p.y) &&
    area(p.prev, p, p.next) >= 0) return false;
  p = p.next;
}
return true;
}

function isEarHashed(ear, minX, minY, size) {
  var a = ear.prev,
      b = ear,
      c = ear.next;
  if (area(a, b, c) >= 0) return false; // reflex, can't be an ear
  // triangle bbox; min & max are calculated like this for speed
  var minTX = a.x < b.x ? (a.x < c.x ? a.x : c.x) : (b.x < c.x ? b.x : c.x),
      minTY = a.y < b.y ? (a.y < c.y ? a.y : c.y) : (b.y < c.y ? b.y : c.y),
      maxTX = a.x > b.x ? (a.x > c.x ? a.x : c.x) : (b.x > c.x ? b.x : c.x),
      maxTY = a.y > b.y ? (a.y > c.y ? a.y : c.y) : (b.y > c.y ? b.y : c.y);
  // z-order range for the current triangle bbox;
  var minZ = zOrder(minTX, minTY, minX, minY, size),
      maxZ = zOrder(maxTX, maxTY, minX, minY, size);
  // first look for points inside the triangle in increasing z-order
  var p = ear.nextZ;
  while (p && p.z <= maxZ) {
    if (p !== ear.prev && p !== ear.next &&
      pointInTriangle(a.x, a.y, b.x, b.y, c.x, c.y, p.x, p.y) &&
      area(p.prev, p, p.next) >= 0) return false;
    p = p.nextZ;
  }
  // then look for points in decreasing z-order
  p = ear.prevZ;
  while (p && p.z >= minZ) {
    if (p !== ear.prev && p !== ear.next &&
      pointInTriangle(a.x, a.y, b.x, b.y, c.x, c.y, p.x, p.y) &&
      area(p.prev, p, p.next) >= 0) return false;
    p = p.prevZ;
  }
  return true;
}

// go through all polygon nodes and cure small local self-intersections
function cureLocalIntersections(start, triangles, dim) {
  var p = start;

```

```

do {
  var a = p.prev,
      b = p.next.next;

  if (!equals(a, b) && intersects(a, p, p.next, b) && locallyInside(a, b) && locallyInside(b, a)) {
    triangles.push(a.i / dim);
    triangles.push(p.i / dim);
    triangles.push(b.i / dim);
    // remove two nodes involved
    removeNode(p);
    removeNode(p.next);
    p = start = b;
  }
  p = p.next;
} while (p !== start);
return p;
}

// try splitting polygon into two and triangulate them independently
function splitEarcut(start, triangles, dim, minX, minY, size) {
  // look for a valid diagonal that divides the polygon into two
  var a = start;
  do {
    var b = a.next.next;
    while (b !== a.prev) {
      if (a.i !== b.i && isValidDiagonal(a, b)) {
        // split the polygon in two by the diagonal
        var c = splitPolygon(a, b);
        // filter colinear points around the cuts
        a = filterPoints(a, a.next);
        c = filterPoints(c, c.next);
        // run earcut on each half
        earcutLinked(a, triangles, dim, minX, minY, size);
        earcutLinked(c, triangles, dim, minX, minY, size);
        return;
      }
      b = b.next;
    }
    a = a.next;
  } while (a !== start);
}

// link every hole into the outer loop, producing a single-ring polygon without holes

```

```

function eliminateHoles(data, holeIndices, outerNode, dim) {
  var queue = [],
      i, len, start, end, list;
  for (i = 0, len = holeIndices.length; i < len; i++) {
    start = holeIndices[i] * dim;
    end = i < len - 1 ? holeIndices[i + 1] * dim : data.length;
    list = linkedList(data, start, end, dim, false);
    if (list === list.next) list.steiner = true;
    queue.push(getLeftmost(list));
  }
  queue.sort(compareX);
  // process holes from left to right
  for (i = 0; i < queue.length; i++) {
    eliminateHole(queue[i], outerNode);
    outerNode = filterPoints(outerNode, outerNode.next);
  }
  return outerNode;
}

function compareX(a, b) {
  return a.x - b.x;
}

// find a bridge between vertices that connects hole with an outer ring and link it
function eliminateHole(hole, outerNode) {
  outerNode = findHoleBridge(hole, outerNode);
  if (outerNode) {
    var b = splitPolygon(outerNode, hole);
    filterPoints(b, b.next);
  }
}

// David Eberly's algorithm for finding a bridge between hole and outer polygon
function findHoleBridge(hole, outerNode) {
  var p = outerNode,
      hx = hole.x,
      hy = hole.y,
      qx = -Infinity,
      m;
  // find a segment intersected by a ray from the hole's leftmost point to the left;
  // segment's endpoint with lesser x will be potential connection point
  do {
    if (hy <= p.y && hy >= p.next.y) {

```

```
var x = p.x + (hy - p.y) * (p.next.x - p.x) / (p.next.y - p.y);  
if (x <= hx && x > qx) {  
  qx = x;  
  if (x === hx) {  
    if (hy === p.y) return p;  
    if (hy === p.next.y) return p.next;  
  }  
}
```