**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**Сумський державний університет**
**Центр заочного, дистанційного та вечірнього навчання**
**Кафедра інформаційних технологій**

«До захисту допущено»
Т.в.о. завідувача кафедри

_____Світлана ВАЩЕНКО

____ _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
**на здобуття освітнього ступеня бакалавр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: _____

*Web-додаток для візуального контролю концептуальних моделей*

Здобувача (ки) групи ІТ-91 Короля Дениса Олеговича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ _____
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник _____к.т.н. доц. Неня Віктор Григорович _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

**Суми – 2023**

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

**ЗАТВЕРДЖУЮ**
Т.в.о зав. кафедри ІТ

Світлана ВАЩЕНКО
«_____»_____2023 р.

З А В Д А Н Н Я
**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Король Денис Олегович*

**1 Тема роботи** *Web-додаток для візуального контролю концептуальних моделей*

керівник роботи *Неня Віктор Григорович, к.т.н., доцент*,

затверджені наказом по університету від « *29* » травня 2023 р. №0588-VI

**2 Строк подання студентом роботи** « 07 » червня 2023 р.

**3 Вхідні дані до роботи** технічне завдання на розробку web-додатку з підтримки діяльності магазину мобільних телефонів

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, проектування web-додатку, розробка web-додатку

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

_____
_____

**6. Консультанти розділів роботи:**

| Розділ | Консультант | Підпис, дата | |
|---|---|---|---|
| | | Завдання видав | Завдання прийняв |
| | | | |

**7.Дата видачі завдання**_____

## КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Оформлення планування робіт | 15.03.2023-25.03.2023 | |
| 2 | Оформлення технічного завдання | 26.03.2023-10.04.2023 | |
| 3 | Проведення аналізу предметної області | 11.04.2023-20.04.2023 | |
| 4 | Проведення проектування web-додатку | 21.04.2023-30.04.2023 | |
| 5 | Розробка web-додатку | 1.05.2023-23.05.2023 | |
| 6 | Тестування web-додатку | 24.05.2023-26.05.2023 | |
| 7 | Оформлення пояснювальної записки | 27.05.2023-31.05.2023 | |

**Студент** _____ Король Д.О.
(підпис)

**Керівник роботи** _____ к.т.н., доц. Неня В.Г.
(підпис)

# РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток для візуального контролю концептуальних моделей».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 23 найменувань, трьох додатків. Загальний обсяг роботи складає 121 сторінки, у тому числі 45 сторінки основного тексту, 2 сторінки списку використаних джерел, 71 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку для візуального контролю концептуальних моделей.

У першому розділі проведено аналіз предметної області за тематикою даного проекту. Розглянуто методи та підходи до візуального контролю концептуальних моделей, а також проаналізовано існуючі web-додатки у цій сфері. Визначено потрібний функціонал та основні вимоги до розроблюваного додатку. Було визначено мету, задачі проекту та засоби його реалізації.

У другому розділі проведено проектування web-додатку для візуального контролю концептуальних моделей. Виконано аналіз можливих технологій та інструментів для розробки, визначено архітектуру системи та інтерфейс користувача. Розроблено основні компоненти додатку та їх взаємодію. Також проведено проектування бази даних для зберігання концептуальних моделей та зв’язків між ними.

Третій розділ присвячено реалізації web-додатку для візуального контролю концептуальних моделей. Описано процес розробки функціональних модулів додатку, їх інтеграцію та тестування. Проведено валідацію та верифікацію розробленого додатку згідно з вимогами, встановленими в другому розділі.

Результатом роботи є розроблений web-додаток для візуального контролю концептуальних моделей. Додаток забезпечує зручний інтерфейс

для створення, редагування та відображення концептуальних моделей. Він дозволяє користувачам спростити процес взаємодії з моделями, забезпечує широкі можливості візуалізації та аналізу концептуальних моделей.

Практичне значення роботи полягає в створенні web-додатку, який сприяє поліпшенню процесу візуального контролю концептуальних моделей. Використання цього додатку дозволяє збільшити продуктивність та ефективність роботи з концептуальними моделями, спрощує їх редагування та аналіз.

КЛЮЧОВІ СЛОВА: WEB-ДОДАТОК, ВІЗУАЛЬНИЙ КОНТРОЛЬ, КОНЦЕПТУАЛЬНІ МОДЕЛІ, ІНТЕРФЕЙС КОРИСТУВАЧА.

**ЗМІСТ**

# ВСТУП

Розробка Web-додатку для візуального контролю концептуальних моделей має велику важливість у сучасному інформаційному суспільстві. Концептуальні моделі використовуються в різних галузях, таких як бізнес-аналітика, системний аналіз, проектування програмного забезпечення та інші. Ці моделі представляють абстрактні концепції, відображають структуру, функціональність та взаємозв'язки системи.

Однак, зростання складності та обсягу концептуальних моделей ставить перед фахівцями великі виклики щодо їх управління та контролю. Традиційні методи редагування та аналізу моделей, які базуються на локальних програмах або інструментах, можуть бути недостатніми для ефективної роботи з великими обсягами даних та колективною співпрацею.

Web-додаток для візуального контролю концептуальних моделей надає значні переваги та вирішує ці проблеми. Він забезпечує централізований доступ до моделей з будь-якого пристрою, що має підключення до Інтернету. Користувачі можуть одночасно працювати з моделями, спільно редагувати та взаємодіяти, спрощуючи комунікацію та співпрацю в команді.

Крім того, web-додаток надає потужні можливості візуалізації та аналізу концептуальних моделей. Завдяки web-інтерфейсу та сучасним технологіям, користувачі можуть легко навігувати по моделях, збільшувати та зменшувати їх масштаб, використовувати інтерактивні елементи для виявлення зв'язків та структури моделей.

Розробка web-додатку для візуального контролю концептуальних моделей сприяє покращенню ефективності роботи з моделями, спрощує процес їх редагування та аналізу, полегшує комунікацію та співпрацю між фахівцями. Він розширює можливості управління концептуальними моделями та сприяє підвищенню продуктивності та якості проектування.

Web-додаток для візуального контролю концептуальних моделей є проектом, спрямованим на розробку інструменту, який забезпечує зручну та ефективну можливість візуального контролю концептуальних моделей. Головною метою даного проекту є створення web-додатку, який надає користувачам можливість створювати, редагувати та аналізувати концептуальні моделі з використанням візуальних засобів.

Для досягнення поставленої мети проекту необхідно виконати наступні завдання:

– виконати аналіз предметної області та обрати актуальні засоби для реалізації web-додатку;

– виконати структурно-функціональне моделювання та планування проєкту;

– розробити систему налаштувань робочого середовища web-додатку;

– розробити інструменти, включаючи елементи побудови та налаштування, необхідні для відображення концептуальних моделей;

– реалізувати модулі для експорту та імпорту створеної концептуальної моделі.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень

Відповідно до останніх досліджень у галузі розробки web-додатків для візуального контролю концептуальних моделей, спостерігається зростаючий інтерес до розробки таких інструментів та їх застосування в різних сферах. Дослідники активно вивчають можливості web-технологій для створення зручного та ефективного середовища для візуалізації та управління концептуальними моделями [1].

Один з цікавих напрямків досліджень пов’язаний з розробкою інтерактивних Web-додатків, що використовують технології віртуальної реальності (VR) та доповненої реальності (AR). Такі додатки дозволяють користувачам взаємодіяти з концептуальними моделями у віртуальному чи змішаному середовищі, що відкриває нові можливості для їх використання та сприяє збільшенню ефективності та якості візуального контролю [2].

Дослідження також акцентують увагу на аспектах ергономіки та інтерфейсного дизайну web-додатків для візуального контролю концептуальних моделей. Вивчаються підходи до оптимізації навігації, взаємодії та візуалізації, з метою поліпшення користувацького досвіду та забезпечення зручного та ефективного сприйняття моделей [3].

Щодо статистики використання web-додатків для візуального контролю концептуальних моделей, згідно з дослідженнями провідних аналітичних компаній, спостерігається постійне зростання популярності та використання таких додатків. За останні роки web-технології значно покращилися, що дозволило розробникам створювати більш потужні та функціональні web-додатки для візуального контролю концептуальних моделей.

## 1.2 Аналіз існуючих продуктів-аналогів

Web-додатки для візуального контролю концептуальних моделей відіграють важливу роль у наукових дослідженнях та практичних застосуваннях. Вони сприяють поліпшенню розуміння та візуалізації складних концептуальних моделей, що забезпечує краще сприйняття, аналіз та взаємодію з цими моделями [4]. Науковий стиль та реалізація додатків-аналогів дозволяють досягнути наступних важливих результатів:

1. Поліпшення процесу візуалізації: використання web-технологій та візуальних елементів дозволяє створювати привабливі та інтуїтивно зрозумілі інтерфейси, які допомагають користувачам ефективно сприймати та розуміти концептуальні моделі.

2. Забезпечення взаємодії та ефективності: web-додатки дозволяють впроваджувати різноманітні функціональні можливості для взаємодії з концептуальними моделями, такі як масштабування, збільшення, переміщення та маніпуляція об’єктами моделі. Це сприяє покращенню продуктивності та ефективності користувачів у вивченні, аналізі та редагуванні моделей [5].

Зручність та доступність: web-додатки забезпечують можливість доступу до концептуальних моделей через web-браузер без необхідності встановлення додаткового програмного забезпечення. Це робить їх доступними з різних пристроїв та платформ, що забезпечує зручність та гнучкість використання.

3. Підвищення співпраці та обміну інформацією: web-додатки дозволяють користувачам спільно працювати над концептуальними моделями, обмінюватися даними та коментарями, що підвищує ефективність колективної роботи та сприяє обміну знаннями [6].

Тому в якості аналогів було обрано 3 web-додатки для створення та візуального контролю концептуальних моделей.

Першим було розглянуто пограмний продукт Lucidchart [7]. Він є потужним web-додатком, який надає можливості для візуального контролю концептуальних моделей. Даний web-додаток пропонує інтуїтивний інтерфейс та набір інструментів для створення, редагування та відображення концептуальних моделей у web-середовищі. Додаток підтримує різні типи концептуальних моделей, включаючи діаграми потоку даних, UML-діаграми та схеми баз даних. Він також має можливість спільної роботи над моделями, експорту в різні формати і інтеграції з іншими web-додатками. На рисунку 1.1 представлено приклад робочого процесу в web-додатку «Lucidchart».

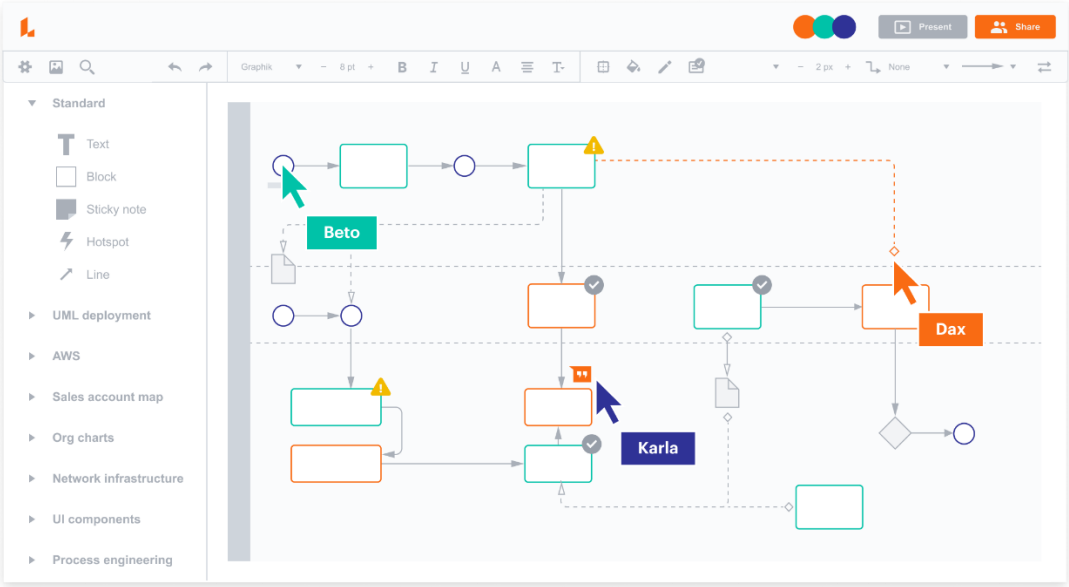


Рисунок 1.1 – Приклад робочого процесу в web-додатку «Lucidchart»

Наступним було проаналізовано онлайн сервіс Draw.io [8]. Він є web-додатком, спеціально розробленим для візуалізації концептуальних моделей. Він надає користувачам можливість імпортувати концептуальні моделі з різних форматів і створювати візуальні представлення цих моделей у web-середовищі. Додаток підтримує різні типи візуалізацій, такі як діаграми класів, діаграми секцій та діаграми потоку роботи. Він також має функціонал для навігації по моделях, масштабування та детального редагування. На рисунку 1.2 представлено приклад робочого процесу в web-додатку «Draw.io».

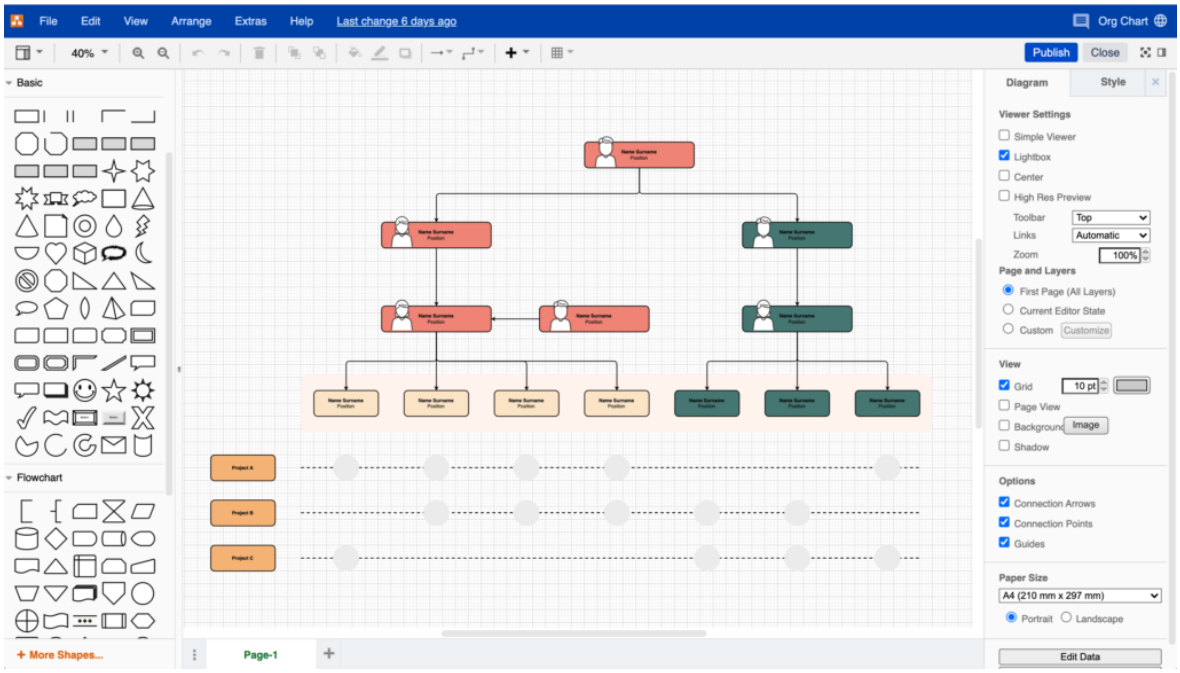


Рисунок 1.2 – Приклад робочого процесу в web-додатку «Draw.io»

Далі було розглянуто програмний продукт Creately [9]. Він є web-додатком, що дозволяє створювати та редагувати концептуальні моделі в онлайн-середовищі. Цей додаток пропонує широкий спектр інструментів для створення діаграм, включаючи варіанти для моделювання процесів, структури даних та взаємодій між об’єктами. Він надає можливості спільної роботи над моделями, збереження моделей у хмарних сховищах та генерацію коду з моделей. На рисунку 1.3 представлено приклад робочого процесу в web-додатку «Creately».

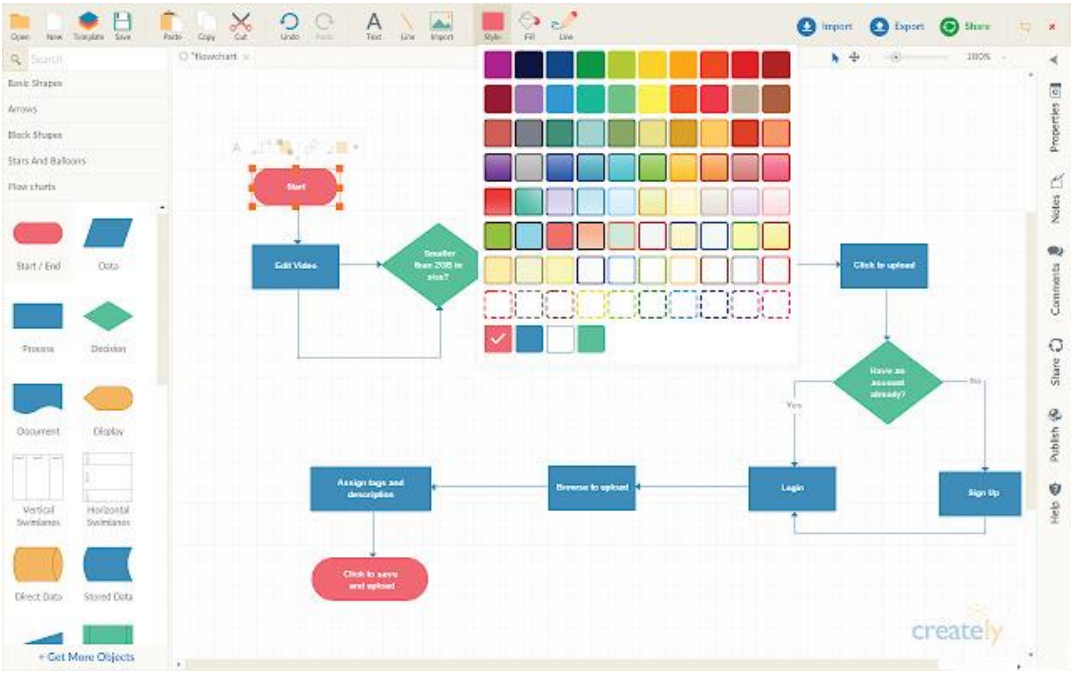


Рисунок 1.3 – Приклад робочого процесу в web-додатку «Creately»

Розглянуті web-додатки аналоги для візуального контролю концептуальних моделей мають свої недоліки, такі як залежність від інтернет-підключення, обмеження швидкості та продуктивності, вразливість до кібератак, обмежена функціональність та проблеми зі сумісністю браузерів. Однак, ці недоліки не зменшують загальної важливості та користі від використання таких додатків. Таблиця 1.1 створена для більш детальної порівняльної характеристики додатків-аналогів.

Таблиця 1.1 – Порівняльна характеристика додатків-аналогів для візуального контролю концептуальних моделей

| Характеристика | Додатки-аналоги | | | Власна розробка |
|---|---|---|---|---|
| | Lucidchart | Draw io | Creately | |
| Функціональність (0-10) | 8 | 8 | 8 | 8 |
| Інтеграція та експорт (0-10) | 9 | 9 | 9 | 9 |

Продовження табл. 1.1.

| | | | | |
|---|---|---|---|---|
| Ціна та доступність (0-10) | 6 | 10 | 8 | 10 |
| Комунікація та спільна робота (0-10) | 10 | 8 | 8 | 8 |
| Навчання та підтримка (0-10) | 8 | 8 | 8 | 8 |
| Інтеграція з іншими програмами (0-10) | 6 | 6 | 6 | 9 |
| Відкритість формату даних | 6 | 6 | 6 | 9 |

У результаті порівняльного аналізу web-додатків для візуального контролю концептуальних моделей, таких як Draw.io, Lucidchart та Creately, було виявлено ряд суттєвих відмінностей та характеристик, що впливають на їх використання та ефективність. Кожен з додатків має свої переваги і недоліки, які варто враховувати при розробці web-додатку для візуального контролю концептуальних моделей.

## 1.3 Постановка задачі

Метою проекту є розробка web-додатку для візуального контролю концептуальних моделей, що надасть користувачам можливість взаємодіяти з концептуальними моделями в онлайн-середовищі. Додаток повинен мати зручний інтерфейс та широкий набір функціональних можливостей для створення, редагування, візуалізації та аналізу концептуальних моделей.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

– визначити актуальність та провести аналіз предметної області;

– обрати методики та засоби розробки, які найкраще відповідають потребам проєкту;

– провести аналіз існуючих аналогів web-додатків для візуального контролю концептуальних моделей та сформулювати технічне завдання для розробки додатку;

– розробити і реалізувати візуалізацію концептуальних моделей у web-додатку, забезпечивши можливість створення, редагування та відображення моделей у зручному для користувача і відкритому форматі даних;

– забезпечити зручну навігацію та пошук інструментів в для полегшення їх використання в процесі створення концептуальних моделях;

– розробити механізми для імпорту та експорту концептуальних моделей;

– оформити супровідну документацію.

Виконання цих завдань дозволить розробити web-додаток, який забезпечуватиме зручний та ефективний візуальний контроль концептуальних моделей у наукових та практичних цілях.

## 1.4 Вибір засобів реалізації web-додатку

Для реалізації web-додатку для візуального контролю концептуальних моделей можна використовувати різні фронтенд технології, такі як React, Angular і Vue.js. Кожна з цих технологій має свої переваги та особливості, які можна врахувати при виборі.

React – це бібліотека JavaScript, яка зосереджена на розробці інтерфейсу користувача. Вона відома своєю простотою використання та широкою підтримкою спільноти розробників. React забезпечує швидку рендерінг віртуального DOM та дозволяє створювати компоненти, які легко повторно використовувати. Однак, для повноцінної реалізації додатку можуть знадобитися додаткові бібліотеки та інструменти [10].

Angular – це повноцінний фреймворк для розробки web-додатків, розроблений компанією Google. Він надає широкий набір інструментів та функціональності для побудови складних додатків. Angular пропонує потужний систему шаблонів, двостороннє зв’язування даних та механізми маршрутизації. Однак, використання Angular може бути складнішим для новачків та вимагати більше часу на навчання [11].

Vue.js – це прогресивний фреймворк JavaScript, який комбінує легкість використання з потужною функціональністю. Він має простий синтаксис та швидкий вивід на екран, що робить його привабливим для швидкого прототипування та розробки. Vue.js також забезпечує можливість компонентного підходу та динамічне зв’язування даних. Однак, Vue.js може бути менш підтримуваним в порівнянні з React або Angular [12].

Для більш детального аналізу було створено таблицю порівнянь frontend-технологій, яка надасть чітке розуміння про вибір технології для розробки web-додатку (табл 1.2).

Таблиця 1.2 – Порівняння frontend-технологій для розробки web-додатку

| **Особливості** | **React** | **Angular** | **Vue.js** |
|---|---|---|---|
| Складність | Середня | Висока | Низька |
| Швидкість запуску серверу | Швидка | Середня | Швидка |
| Спільнота розробників | Велика | Велика | Середня |
| Підтримка компанією | Ні | Google | Ні |
| Документація | Легка для вивчення | Легка для вивчення | Легка для вивчення |
| Навчання | Легко | Складно | Легко |
| Використання великих проєктів | Так | Так | Так |
| Спрощена розробка прототипів | Так | Ні | Так |
| Функціональність | Середня | Висока | Середня |

Узагальнюючи, React був обраний для розробки web-додатку для візуального контролю концептуальних моделей через його численні переваги. React відомий своєю високою продуктивністю та швидким рендерінгом, що забезпечує плавну взаємодію з додатком. Він також має велику спільноту розробників, що забезпечує відповідну підтримку та документацію.

Враховуючи ці переваги, React є привабливим вибором для розробки web-додатку для візуального контролю концептуальних моделей, оскільки він забезпечує продуктивну розробку, високу швидкість роботи та гнучкість для створення інтерактивного та ефективного інтерфейсу користувача.

# 2 ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ПРОЄКТУ

Створення контекстної діаграми є важливим етапом в розробці web-додатку для візуального контролю концептуальних моделей, оскільки вона дозволяє уявити і відобразити взаємозв’язки між різними компонентами та модулями системи. Контекстна діаграма надає високорівневий огляд архітектури додатку та допомагає розробникам зрозуміти його структуру та функціональні можливості.

Контекстна діаграма допомагає чітко визначити межі системи та її взаємодію з зовнішніми елементами. Вона вказує, які компоненти належать до додатку та як вони взаємодіють з іншими системами або користувачами. Також діаграма дозволяє розробникам побачити, які компоненти системи існують та як вони пов’язані між собою та відображає взаємодію додатку з користувачами та зовнішніми системами. Вона показує, які дані передаються між компонентами та які функції доступні користувачам [13].

На рисунку 2.1 представлено контекстну діаграму «Реалізація коцептуальних моделей».

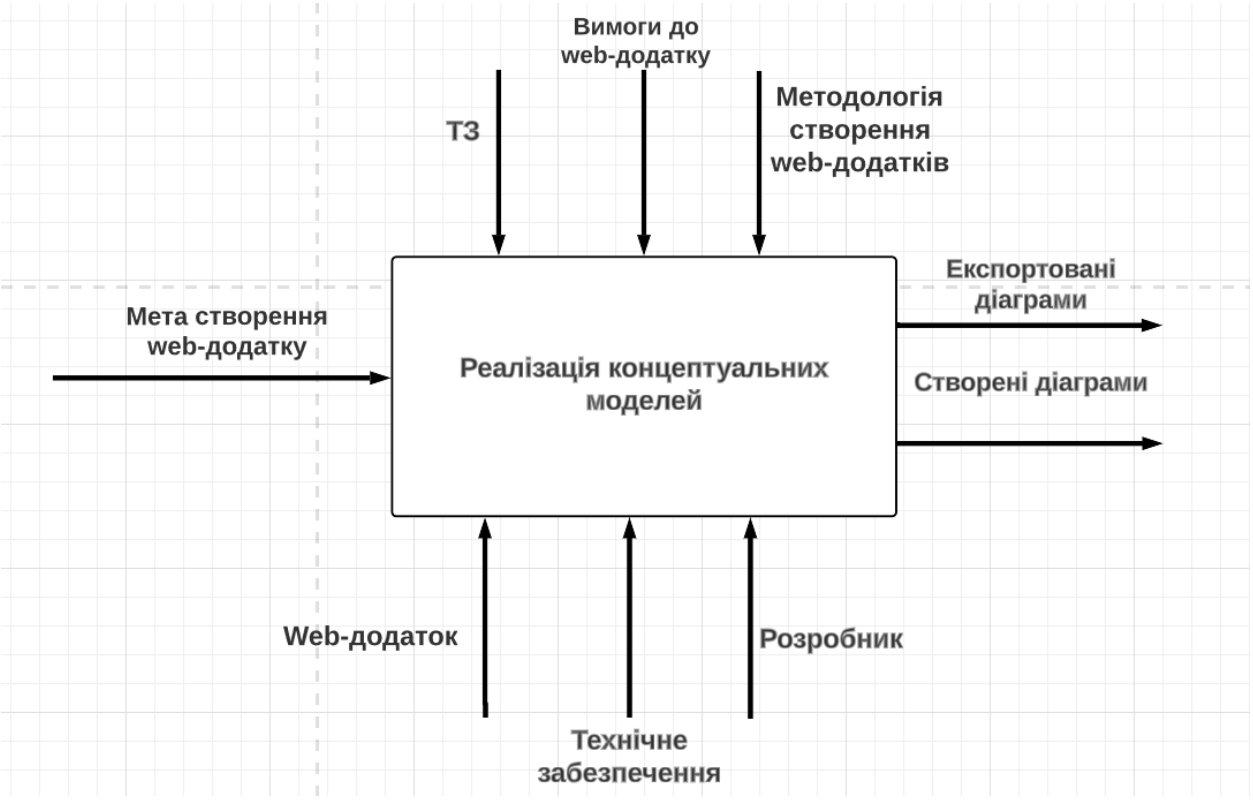


Рисунок 2.1 – Контекстна діаграма «Реалізація концептуальних моделей»

У діаграмі визначені такі головні елементи, як:

– Вхідні дані: мета створення web-додатку;

– Вихідні дані: експортовані діаграми, створені діаграми;

– Управління: ТЗ (технічне завдання), вимоги до web-додатку та методологія створення web-додатків;

– Механізми: web-додаток, розробник, технічне забезпечення.

Діаграма декомпозиції блоку є важливим інструментом в розробці web-додатку для візуального контролю концептуальних моделей, оскільки вона дозволяє розбити складну систему на менші, керовані блоки та продемонструвати взаємозв’язки та взаємодію між ними. Ця діаграма допомагає розробникам зрозуміти структуру та організацію системи, визначити функціональність кожного блоку, а також легше управляти розробкою та тестуванням [14]. На рисунку 2.2 представлено діаграму декомпозиції блоку.

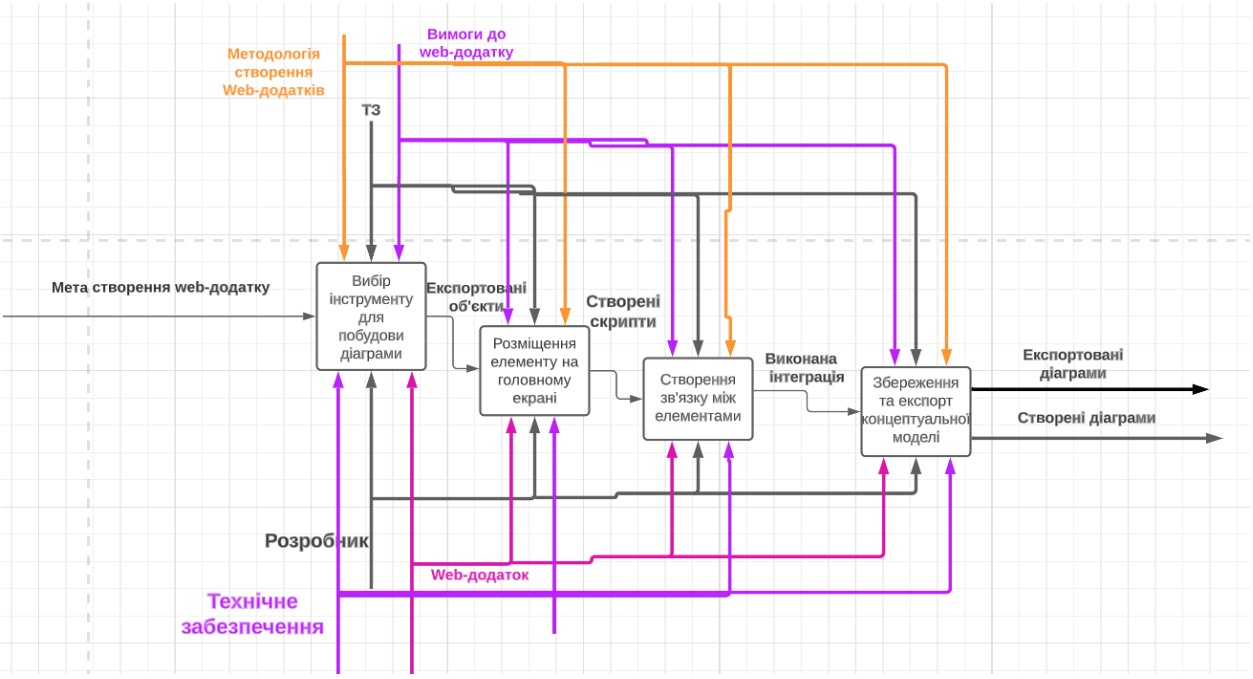


Рисунок 2.2 – Діаграма декомпозиції блоку

Діаграма варіантів використання блоку має важливе значення в розробці web-додатку для візуального контролю концептуальних моделей, оскільки вона дозволяє систематично описати різні сценарії використання системи та взаємодію користувачів з нею. Ця діаграма допомагає розробникам зрозуміти потреби користувачів, їх очікування та способи взаємодії з додатком. На рисунку 2.3 представлено діаграму варіантів використання web-додатку.

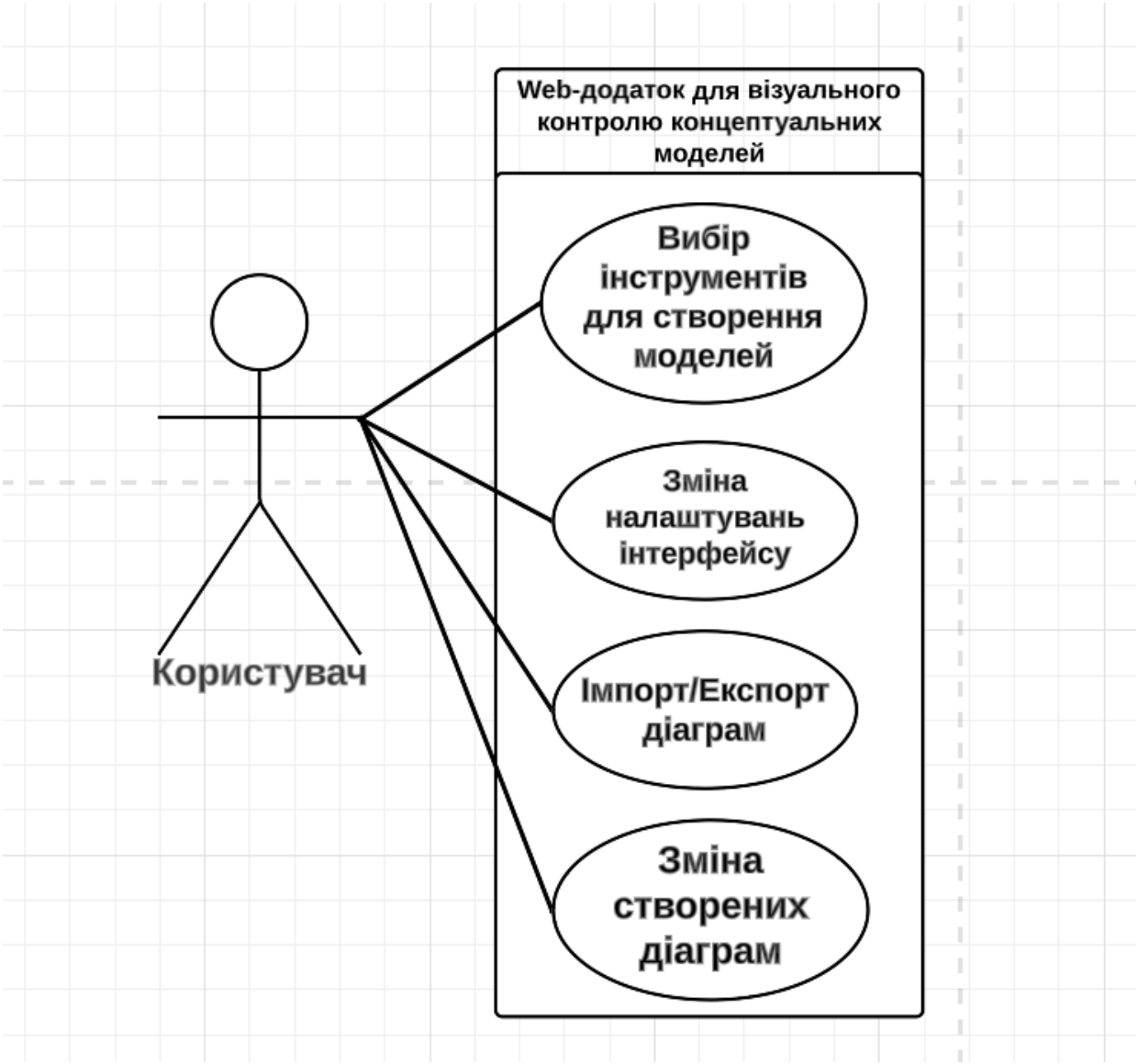


Рисунок 2.3 – Діаграма варіантів використання web-додатку

Одним з ключових етапів перед розробкою web-додатку для візуального контролю концептуальних моделей є побудова діаграми Ганта, яка дозволяє візуально представити часовий графік виконання завдань з урахуванням обмежень використання ресурсів. Крім того, було проведено оцінку ризиків, що можуть виникнути під час моделювання додатку. Детальний план робіт наведений в додатку Б.

# 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

Етап програмної реалізації починається з створення проекту в середовищі Visual Studio Code [15]. Для цього необхідно в терміналі написати наступні команди:

```
npx create-react-app my-app
 cd my-app
 npm start
```

Create React App не обробляє back-end логіку або бази даних, він лише надає команди для складання front-end, тому можна використовувати його з будь-якою back-end технологією.

Коли проєкт готовий до розгортання в продакшені, запуск команди `npm run build` створить оптимізовану збірку застосунку в папці build.

На рисунку 3.1 представлено вигляд терміналу створеного додатку.

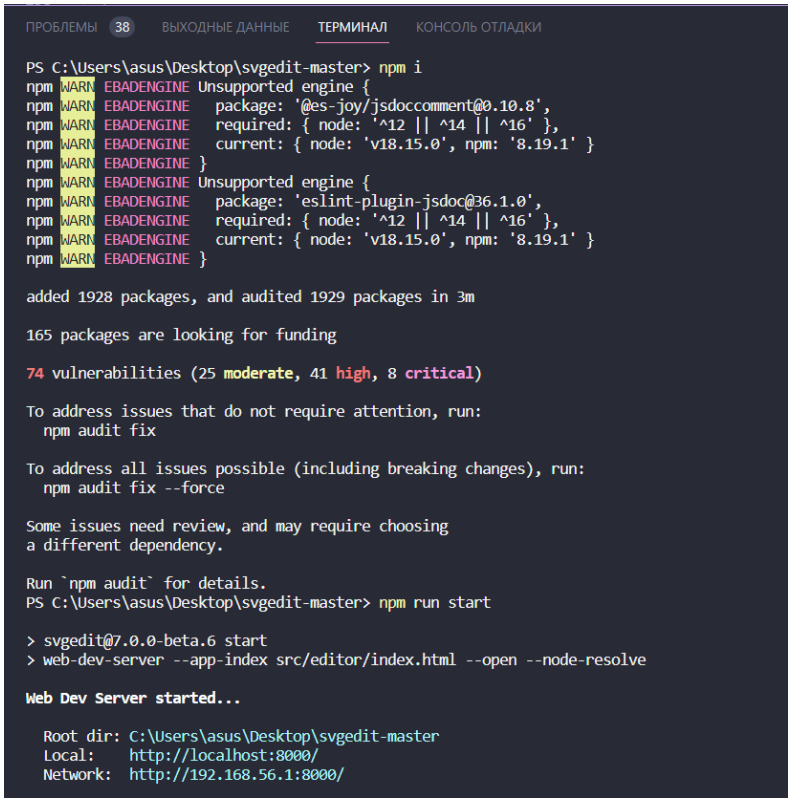


Рисунок 3.1 – Вигляд терміналу створеного додатку

Після того як проект був створений, необхідно розробити необхідні функції додатку, які необхідні для створення діаграм та інтерфейсу користувача. Головним елементом проєкту є компонент, який поєднує в собі весь функціонал додатку. На рисунку 3.2 представлено вигляд розробки компоненту «MainMenu.js» в середовищі Visual Studio Code.

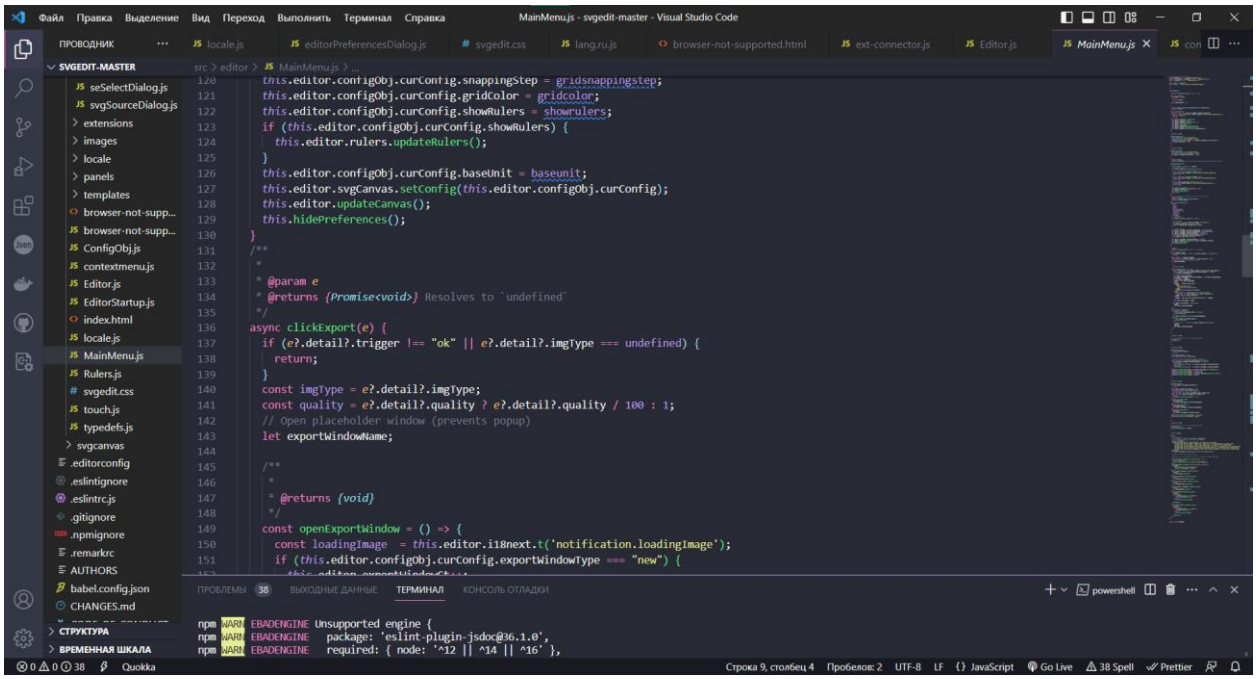


Рисунок 3.2 – Вигляд компоненту «MainMenu.js» в середовищі VS Code

Далі виконуємо розробку необхідних компонентів для функціоналу додатку. Приклад створення панелі налаштувань для web-додатку представлено на рисунку 3.3.

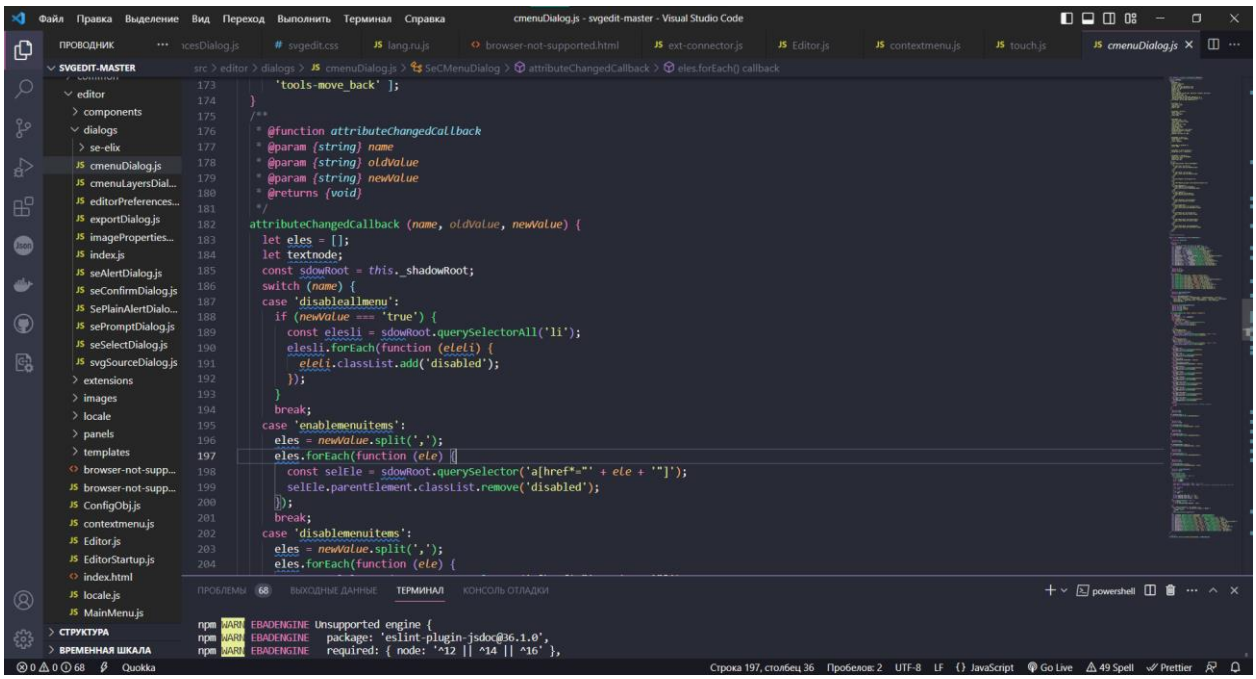


Рисунок 3.3 – Приклад створення компоненту «cmenuDialog.js»

Для перевірки роботи компонентів web-додатку необхідно створити контролер для перевірки підтримки браузера. На рисунку 3.4 представлено створення компоненту «browser-not-support»

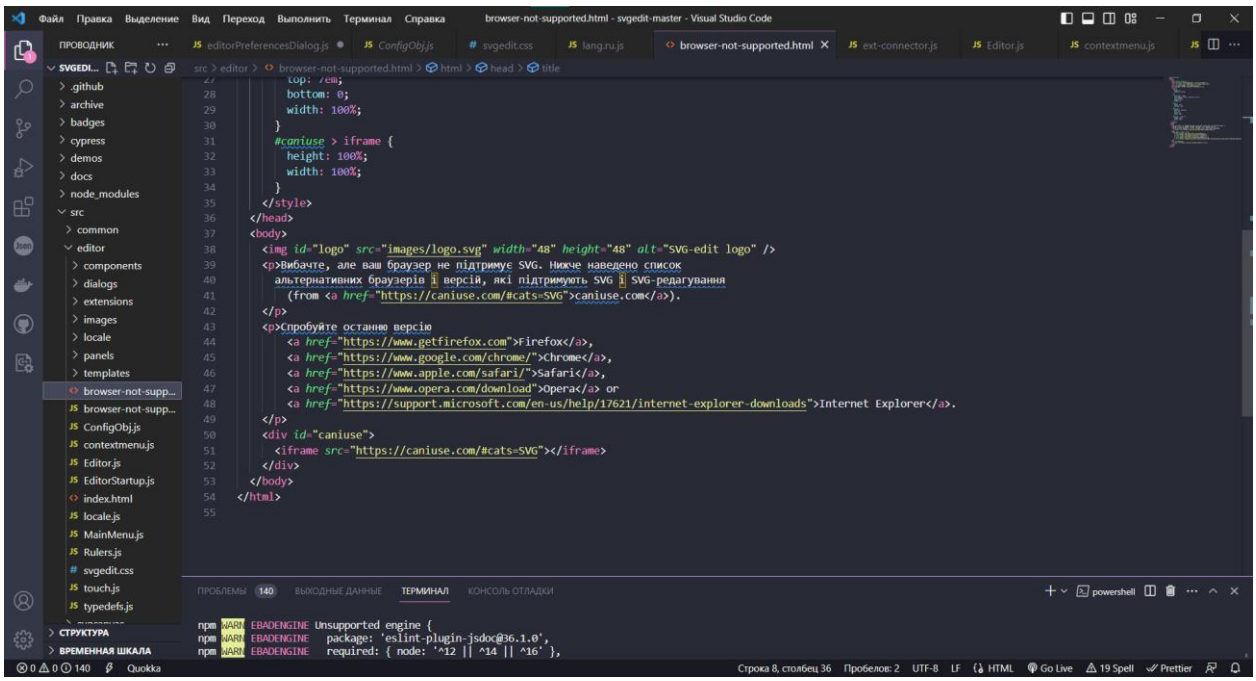


Рисунок 3.4 – Процес створення компоненту «browser-not-support»

Також для інтерфейсу було створено компоненти «panels» які будуть виконувати функцію підключення функцій до відповідних панелей. На рисунку 3.5 представлено результат створення компоненту «TopPanel.js».

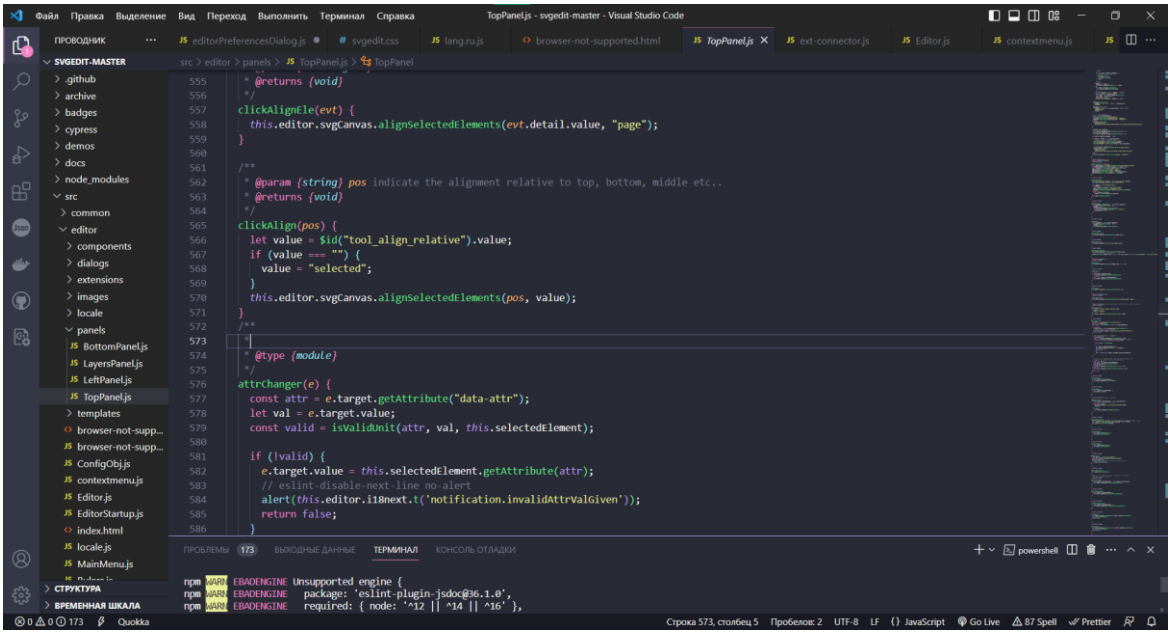


Рисунок 3.5 – Результат створення компоненту «TopPanel.js»

Для можливості опрацювання інформації концептуальних моделей для них обрано формат svg [16-22], який відноситься до класу xml. Формат xml дуже поширений, а відтак має гарну підтримку у вигляді готових бібліотек.

Векторна графіка формату svg стандартизована і є складовою HTML5 [23], а тому реалізована в усіх сучасних браузерах. Якість зображення не залежить від масштабування і завжди забезпечується максимально можлива чіткість зображення.

Особливість концептуальних моделей полягає у тому, що використовуються зображення лише простих геометричних фігур: прямокутник, еліпс, відрізок, стрілки. Зображення моделі виконується у одному чорному кольорі, зауваження – червоним, а відповідь – синім.

Для збереження створеної та рецензованої діаграми спроектовано компонент «svgSourceDialog», розробка якого показана на рисунку 3.6.

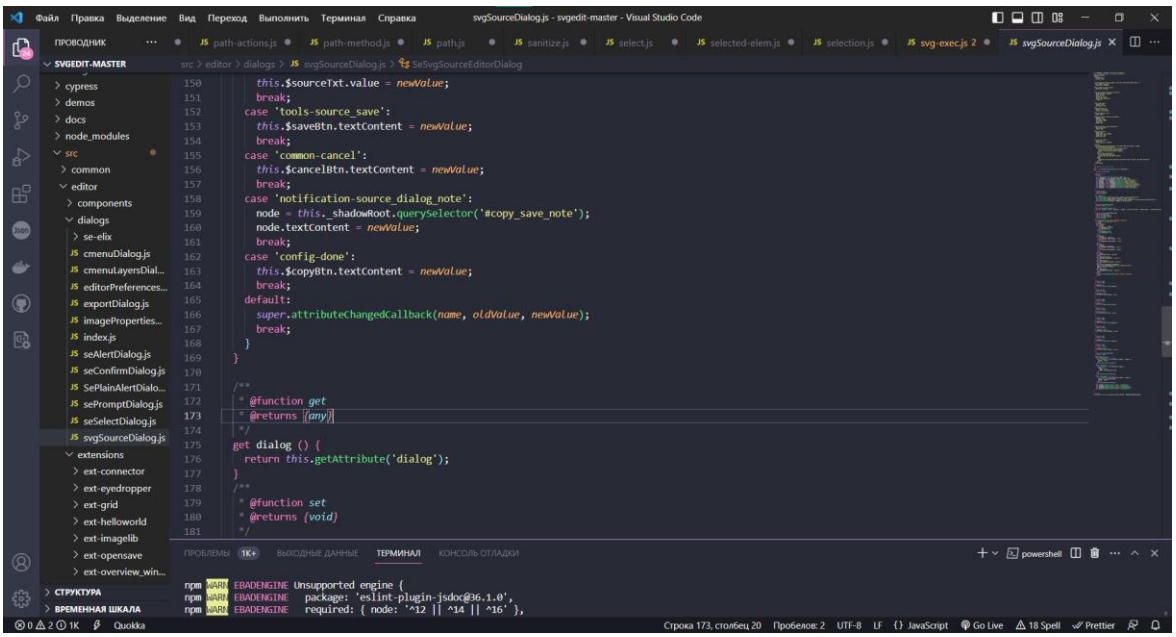

Рисунок 3.6 – Процес створення компоненту «svgSourceDialog»

Також для користування web-додатком за допомогою планшетів або телефонів було створено компонент «touch». На рисунку 3.7 представлено процес розробки компоненту.

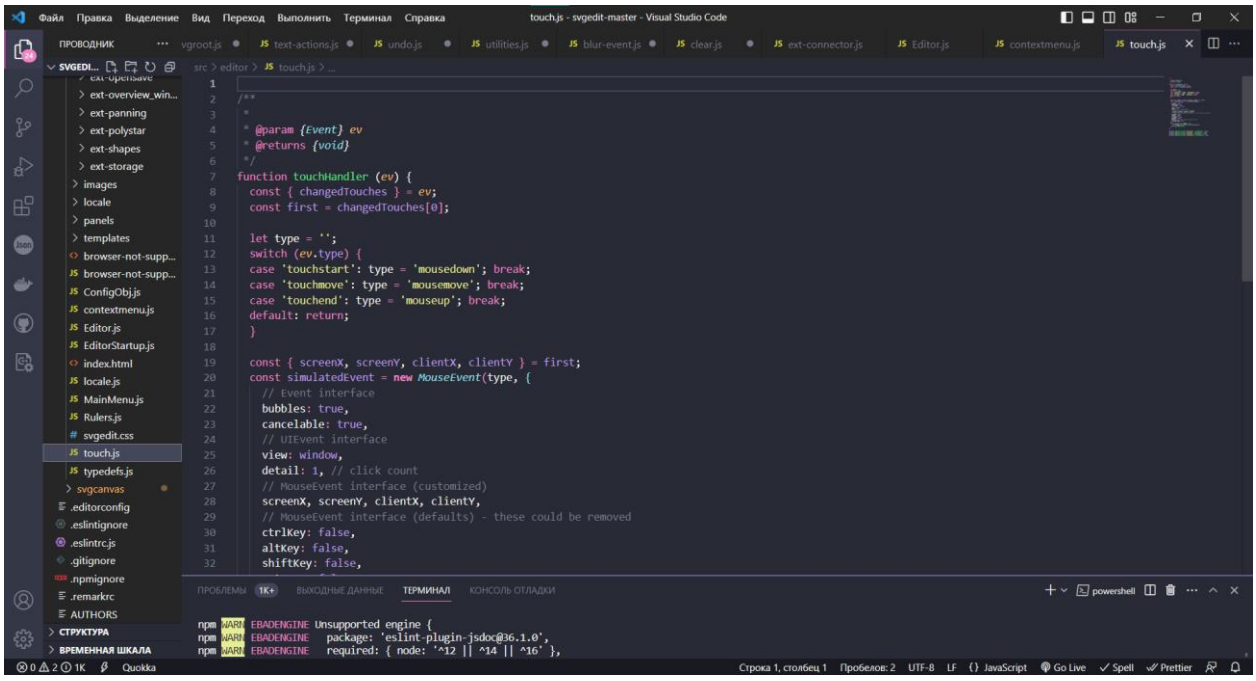

Рисунок 3.7 – Процес створення компоненту «touch.js»

Програмний код основних компонентів web-додатку для візуального контролю концептуальних моделей описано у додатку В.

## 3.2 Демонстрація роботи web-додатку

Головний інтерфейс web-додатку складається з 5 областей, а саме:

– панель інструментів;
– панель налаштувань робочого середовища та web-додатку;
– панель вибору розміру робочого середовища та кольору;
– панель робочого середовища;
– панель керування шарами робочого середовища.

Панель інструментів: ця область інтерфейсу надає користувачеві доступ до різноманітних інструментів та функцій, які можна використовувати в процесі роботи з додатком. Вона може містити кнопки, значки або текстові посилання, що забезпечують швидкий доступ до часто використовуваних операцій. На рисунку 3.8 представлено вигляд панелі інструментів.

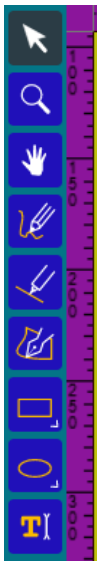

Рисунок 3.8 – Вигляд панелі інструментів

Панель налаштувань робочого середовища та web-додатку: ця область інтерфейсу дозволяє користувачеві налаштовувати параметри робочого середовища та веб-додатку відповідно до своїх потреб. Вона може містити опції налаштувань, які стосуються вигляду інтерфейсу, робочого середовища та інших аспектів додатку. На рисунку 3.9 представлено вигляд панелі налаштувань робочого середовища.

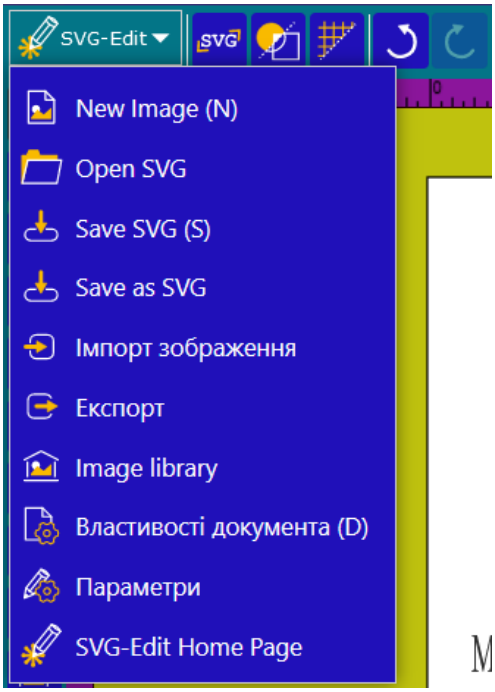


Рисунок 3.9 – Вигляд панелі налаштувань робочого середовища

Панель вибору розміру робочого середовища та кольору: ця область інтерфейсу надає можливість користувачеві змінювати розмір робочого середовища, тобто простору, в якому відбувається редагування або відображення контенту. Крім того, вона також дозволяє вибрати кольорову схему або тему, що впливає на зовнішній вигляд концептуальної моделі. На рисунку 3.10 представлено вигляд панелі вибору розміру робочого середовища.


Рисунок 3.10 – Вигляд панелі вибору розміру робочого середовища

Панель робочого середовища: ця область інтерфейсу представляє собою основну робочу зону, де користувач взаємодіє з вмістом та функціями додатку. Панель містить візуальні компоненти, графічні елементи та інші інструменти, необхідні для виконання завдань, пов’язаних з редагуванням, створенням або переглядом створених концептуальних моделей. На рисунку 3.11 представлено вигляд панелі робочого середовища.

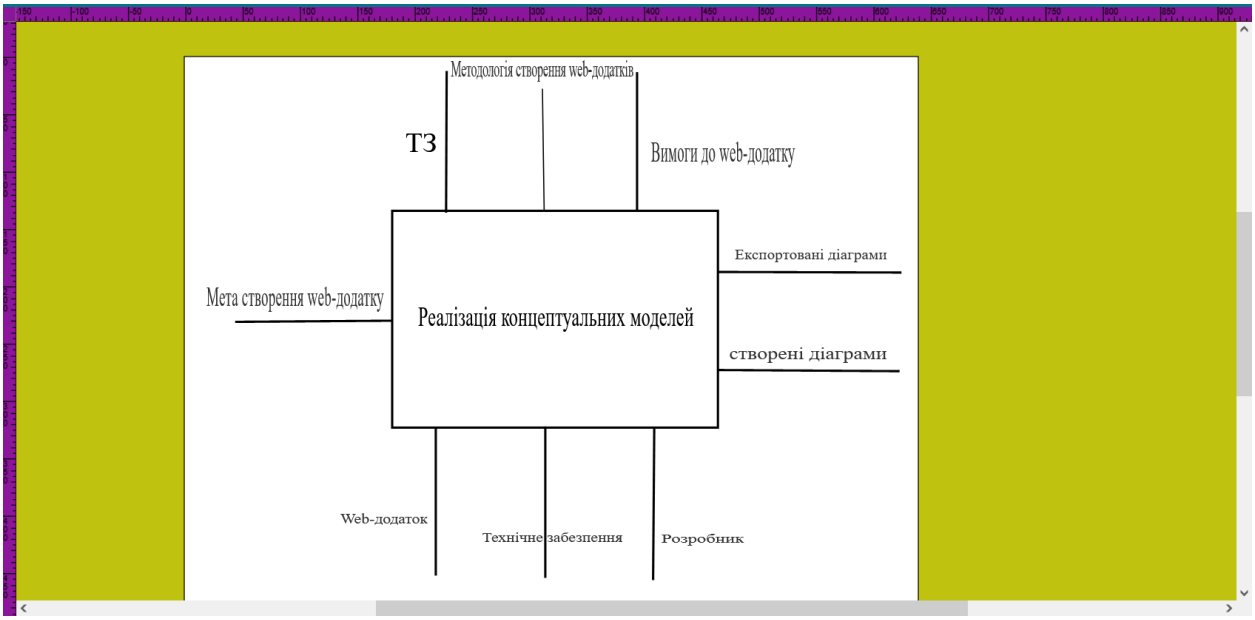


Рисунок 3.11 – Вигляд панелі робочого середовища

Панель керування шарами робочого середовища: ця область інтерфейсу дозволяє користувачеві керувати шарами в робочому середовищі. Шари використовуються для організації та управління різними елементами конуептуальної моделі. Користувач може додавати, видаляти, змінювати порядок та взаємодіяти з шарами за допомогою цієї панелі. На рисунку 3.12 представлено вигляд керування шарами робочого середовища.

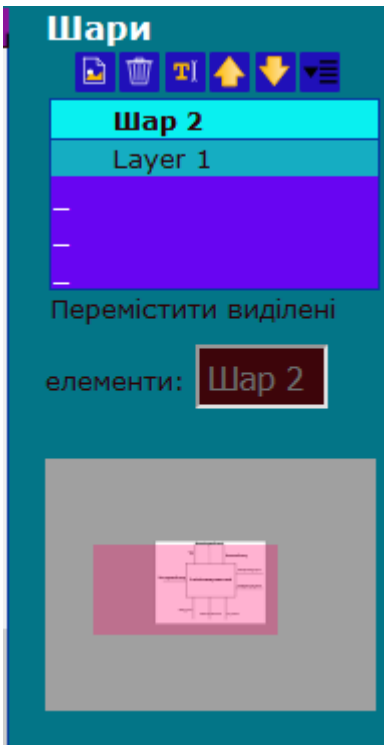


Рисунок 3.13 – Вигляд панелі керування шарами робочого середовища

Зазначені області інтерфейсу спрямовані на забезпечення зручності та ефективності користування web-додатком, дозволяючи користувачу налаштовувати параметри, вибирати оптимальний режим роботи та здійснювати необхідні дії для досягнення своїх цілей. Загальний вигляд web-додатку представлено на рисунку 3.14.

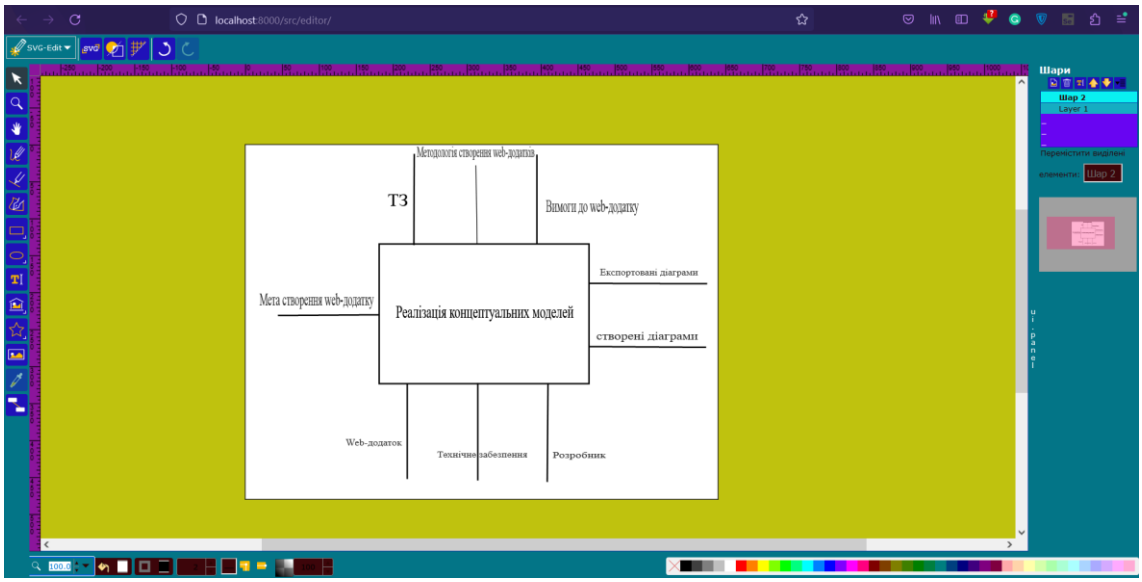


Рисунок 3.14 – Загальний вигляд web-додатку

Після того як проєкт виконав операцію збірки та компіляції, за адресою http://localhost:8000/src/editor/ користувач зможе використовувати web-додаток. Спочатку користувач може обрати тип відображення робочого полотна (сітка чи чисте полотно) на панелі налаштувань робочого середовища. На рисунку 3.15 представлено вигляд іконки «Показати/Приховати сітку» та результат застосування даного інструменту (рис 3.16).

Рисунок 3.15 – Вигляд іконки «Показати/Приховати сітку»

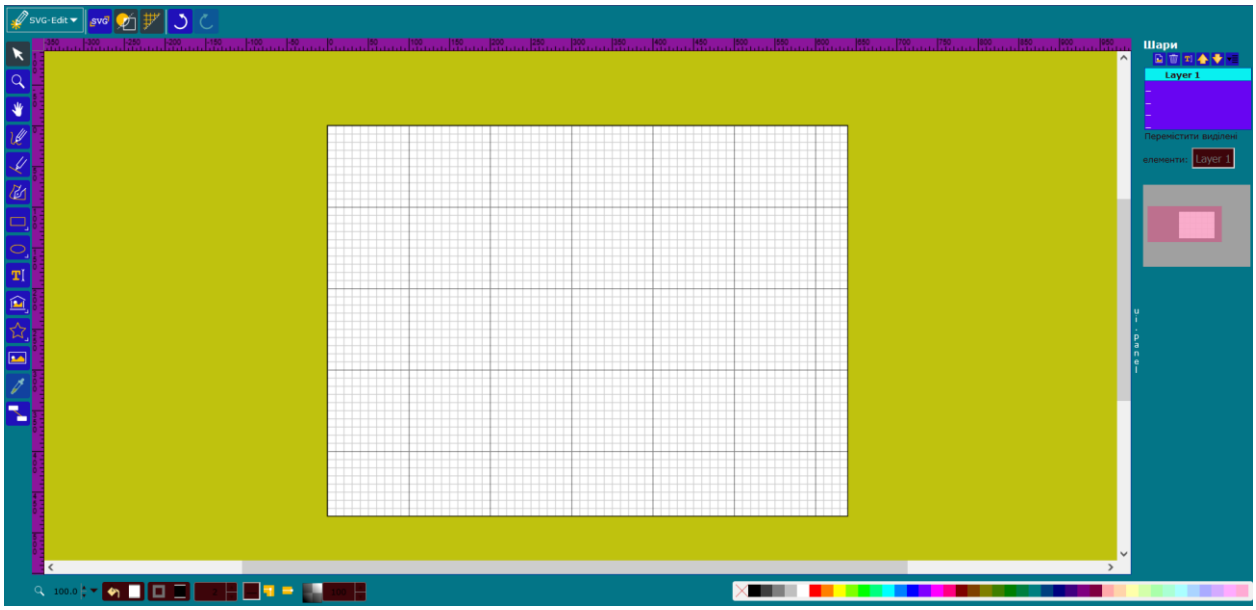


Рисунок 3.16 – Результат застосування інструменту «Показати/Приховати сітку»

Наступний інструмент – «Виділити». Він надає можливість виділити всі необхідні для користувача елементи та виконати потрібну операцію (перемістити, видалити, скопіювати). На рисунку 3.17 представлено вигляд іконки інструменту «Виділити» та приклад застосування інструменту (рис 3.18).

Рисунок 3.17 – Вигляд іконки інструменту «Виділити»

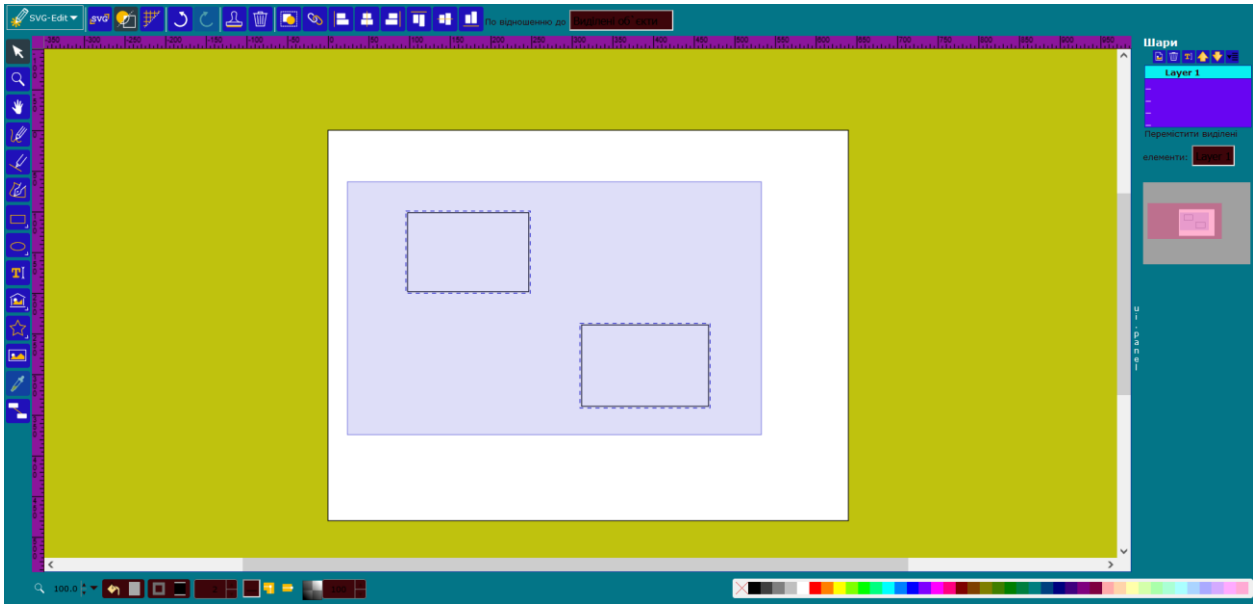

Рисунок 3.18 – Приклад застосування інстументу

Лупа - інструмент, який збільшує деталі вибраної області на моделі, дозволяючи користувачу докладніше розглянути та вивчити деталі цієї області. На рисунку 3.19 представлено вигляд іконки інструменту «Лупа» та приклад застосування інструменту (рис 3.20).

Рисунок 3.19 – Вигляд іконки інструменту «Лупа»

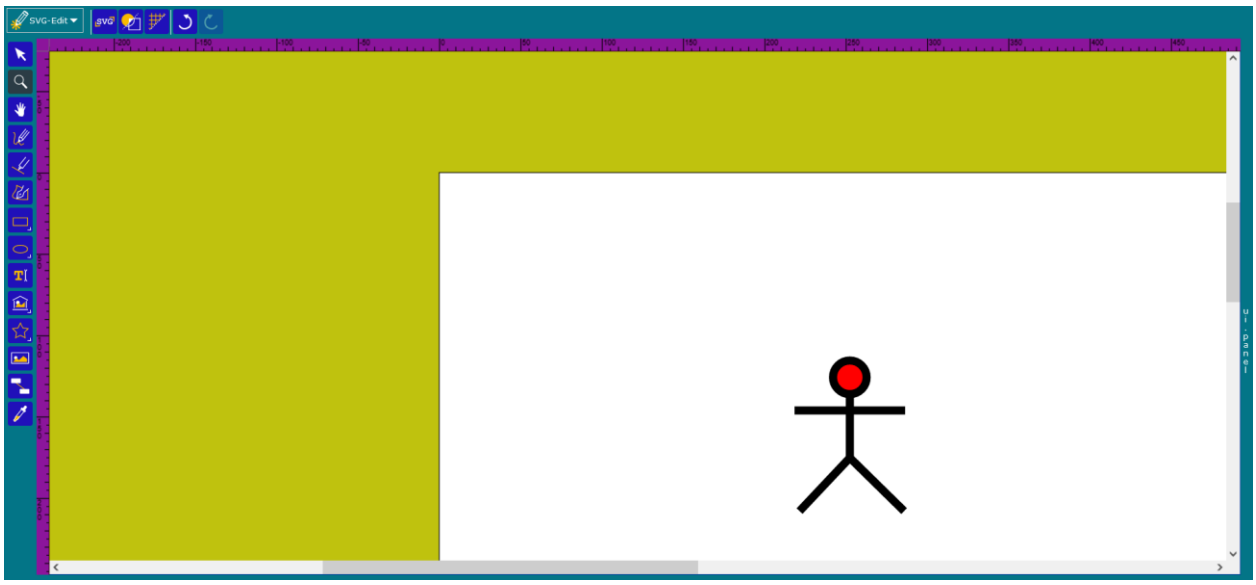

Рисунок 3.20 – Приклад застосування інструменту «Лупа»

Панорамування - цей інструмент дозволяє користувачу переміщатись по моделі, переглядаючи різні частини без зміни масштабу. Він дозволяє зручно навігувати по великих та складних моделях, досліджуючи їх різні аспекти. На рисунку 3.21 представлено вигляд іконки інструменту «Панорамування».

Рисунок 3.21 – Вигляд іконки інструменту «Панорамування»

Подальші інстументи будуть демонстуватися при процесі створення діаграми декомпозиції.

Для побудови діаграми декомпозиції необхідно використати інструменти «Прямокутник», «Лінія» ,«Олівець», «Текст» та «Створення зв’язку між об’єктами».

Прямокутник - інструмент, який дозволяє користувачу малювати прямокутні фігури на моделі. Це може бути використано для виділення, позначення або групування певних об’єктів або областей на моделі. На рисунку 3.22 представлено вигляд іконки інструменту «Прямокутник».

Рисунок 3.22 – Вигляд іконки інструменту «Прямокутник»

Лінія – інструмент, який дозволяє користувачу малювати прямі лінії між двома точками на моделі. Використовується для показу взаємозв'язків, зв'язків або інших типів зв'язків між об'єктами на моделі. На рисунку 3.23 представлено вигляд іконки інструменту «Лінія».

Рисунок 3.23 – Вигляд іконки інструменту «Лінія».

Олівець – інструмент, який дозволяє користувачу малювати вільні лінії або нотатки безпосередньо на моделі. Він може використовуватись для позначення або підкреслення певних деталей, створення коментарів або малюнків для пояснення концептуальної моделі. На рисунку 3.24 представлено вигляд іконки інструменту «Олівець».

Рисунок 3.24 – Вигляд іконки інструменту «Олівець»

Текст – інструмент, який дозволяє користувачу додавати текстові елементи до моделі. Використовується для створення нотаток, пояснень, описів або міток, щоб детальніше пояснити або інтерпретувати певні аспекти моделі. На рисунку 3.25 представлено вигляд іконки інструменту «Текст».

Рисунок 3.25 – Вигляд іконки інструменту «Текст»

Створення зв’язку між об’єктами - цей інструмент дозволяє користувачу створювати лінії або зв’язки між різними об’єктами на моделі, що вказують на їх взаємозв’язок або залежність. Використовується для показу структури, взаємодії або потоку даних між елементами моделі. На рисунку 3.25 представлено вигляд іконки інструменту «Створення зв’язку між об’єктами».

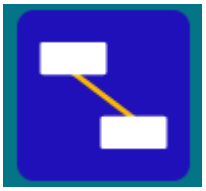

Рисунок 3.26 – Вигляд іконки інструменту
«Створення зв’язку між об’єктами»

На початку побудови треба визначити, на скільки блоків будувати декомпозицію. У даному випадку визначено 4 блоки. За допомогою інструменту «Прямокутник» розміщуємо блоки на робочому полотні (рис.3.27).

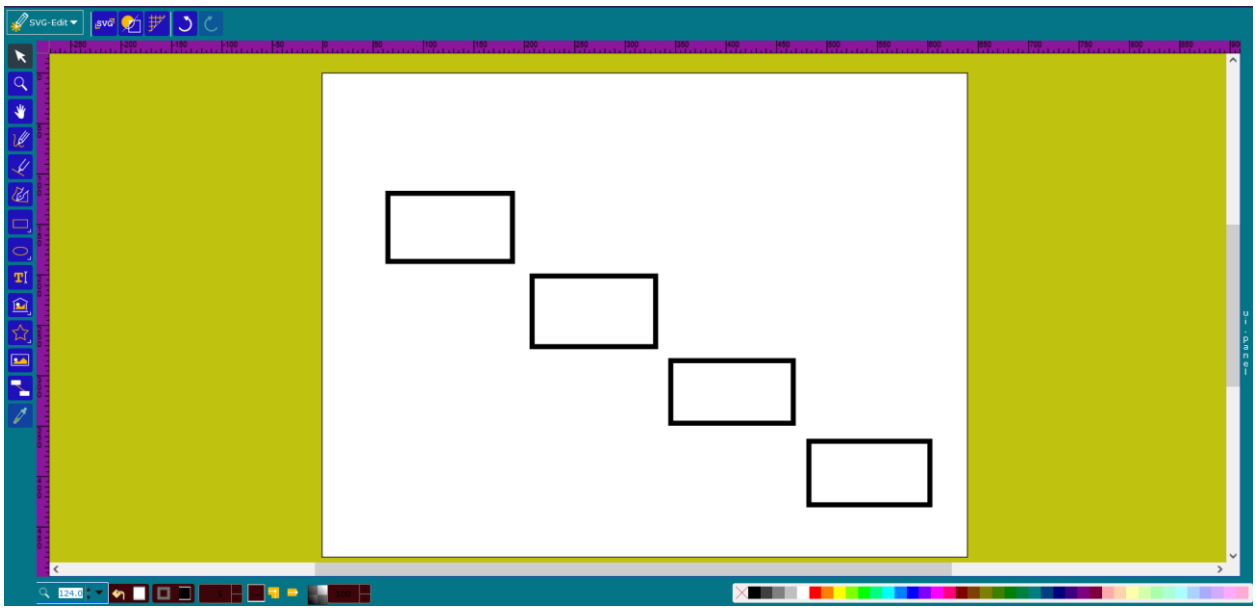

Рисунок 3.27 – Результат розміщення 4 блоків

Після цього необхідно підписати блоки по ієрархії виконання роботи. Дану операцію можна виконати за допомогою інструменту «Текст». На рисунку 3.28 представлено результат виконання операції підпису блоків.

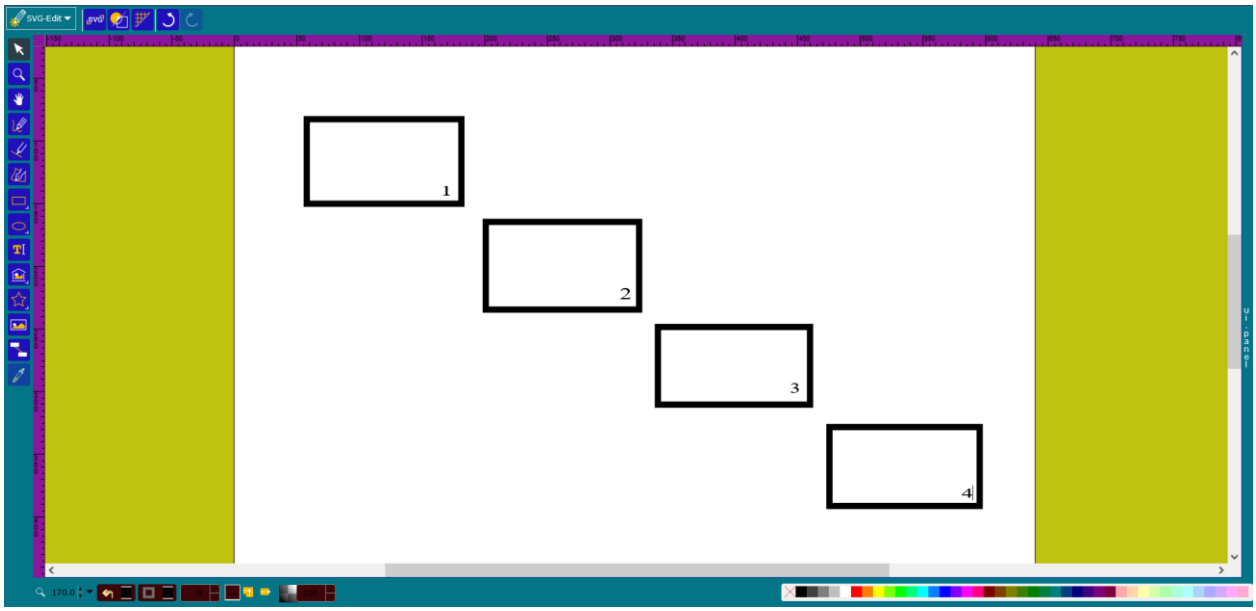


Рисунок 3.28 – Результат виконання операції підпису блоків

Тепер необхідно вказати, яку операцію виконує кожен блок, та додати дані, необхідні для декомпозиції (вхідні дані, керування, механізми, вихідні

дані). Для цього використовуємо інструмент «Текст». На рисунку 3.29 представлено результат додавання тексту до блоків.

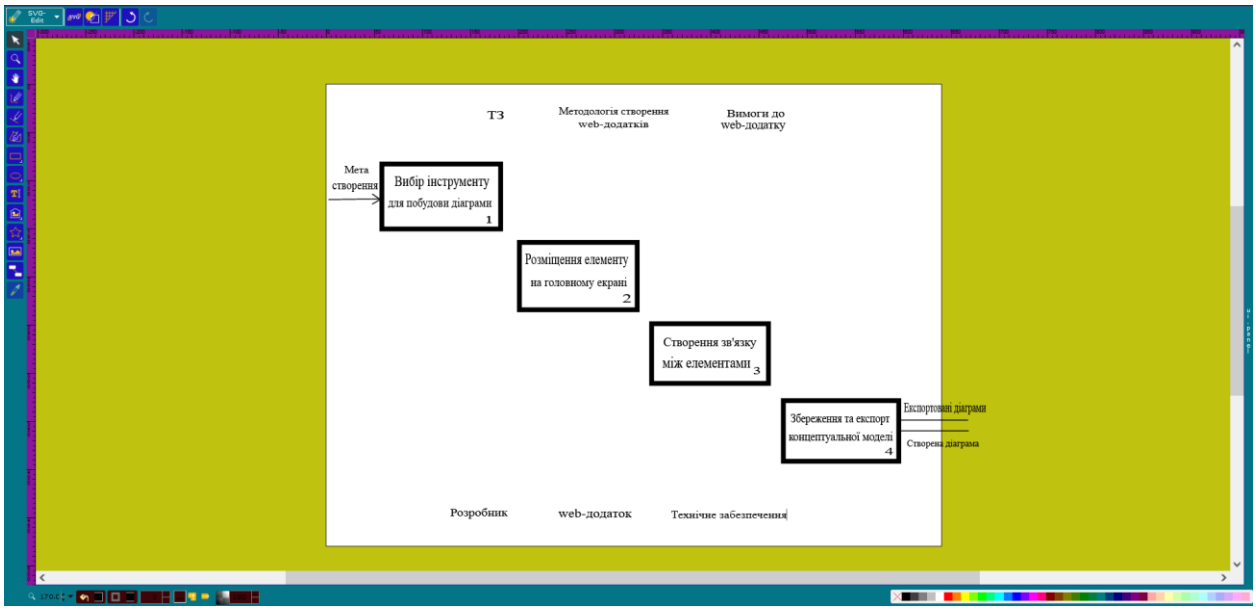


Рисунок 3.29 – Результат додавання тексту до блоків

Після цього необхідно поєднати компоненти між собою. Для цього використаємо інструмент «Створення зв’язку між об’єктами». Для створення зв’язку необхідно натиснути на елемент, від якого буде йти зв’язок та натиснути елемент, з яким потрібно створити зв’язок. На рисунку 3.30 представлено результат створення зв’язків між блоками.

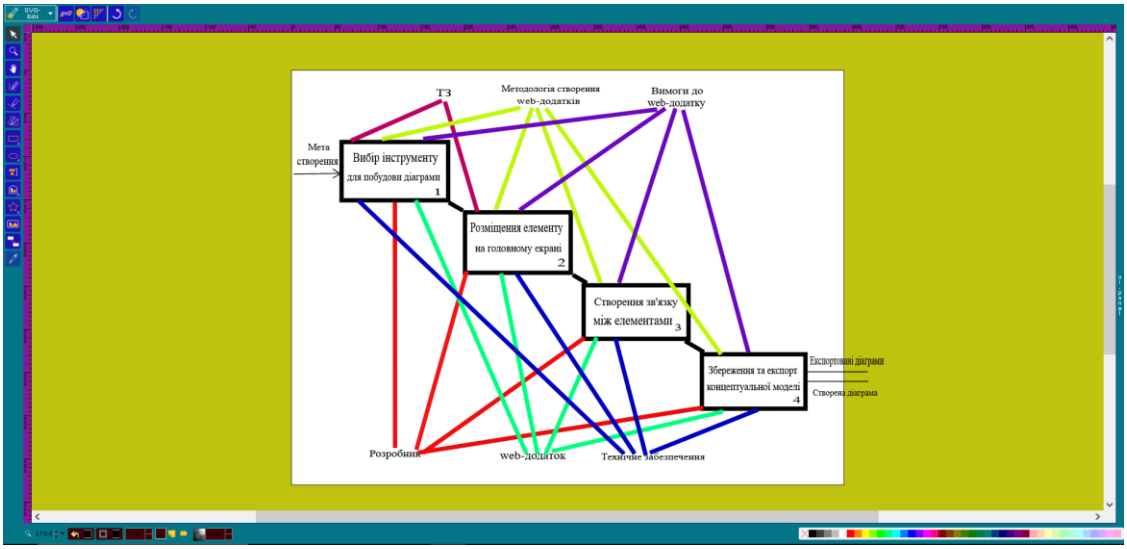


Рисунок 3.30 – Результат створення зв’язків між блоками

Перед тим як перейти до експорту та збереження моделі, необхідно розглянути інші інструменти, які допоможуть скоротити час та ресурси на створення концептуальної моделі.

Піпетка - інструмент, який дозволяє користувачу вибрати колір з моделі або іншого джерела і використовувати його для малювання або заповнення об’єктів на моделі. Використовується для точного відтворення кольору або відтінку на моделі. На рисунку 3.31 представлено вигляд іконки інструменту «Піпетка» та приклад застосування інструменту (рис 3.32).

Рисунок 3.31 – Вигляд іконки інструменту «Піпетка»

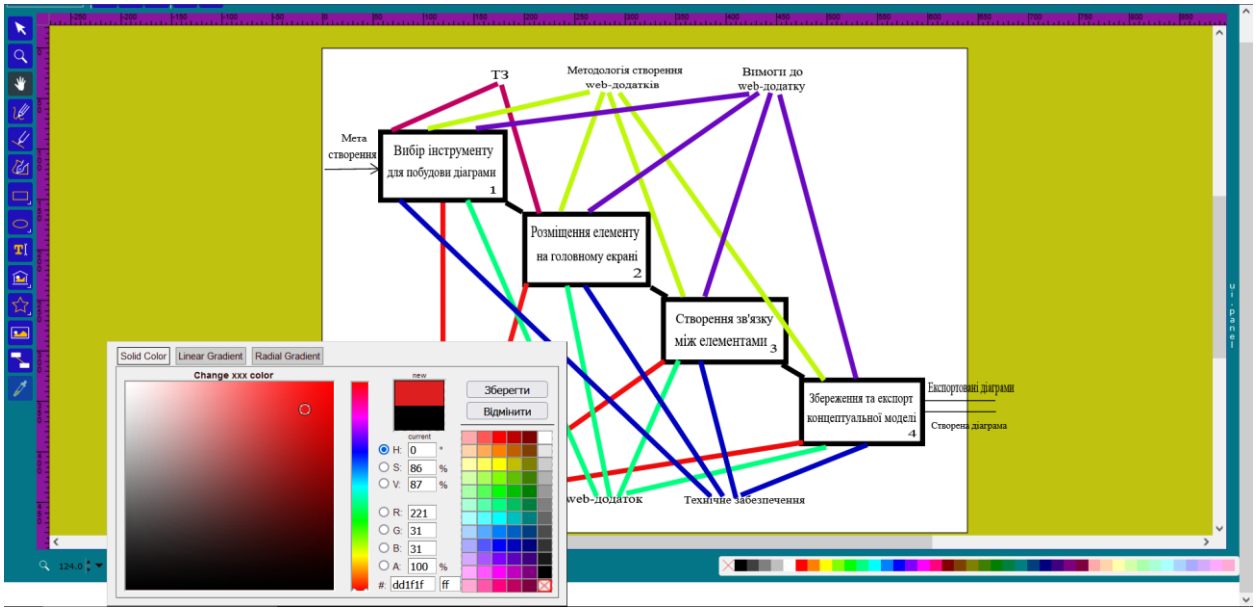


Рисунок 3.32 – Приклад застосування інструменту «Піпетка»

Зображення – інструмент, який дозволяє користувачу додавати зображення або ілюстрації до моделі. Використовується для уточнення або прикраси моделі, а також для візуального пояснення певних аспектів концептуальних моделей. На рисунку 3.33 представлено вигляд іконки інструменту «Зображення» та приклад застосування інструменту (рис 3.34).

Рисунок 3.33 – Вигляд іконки інструменту «Зображення»

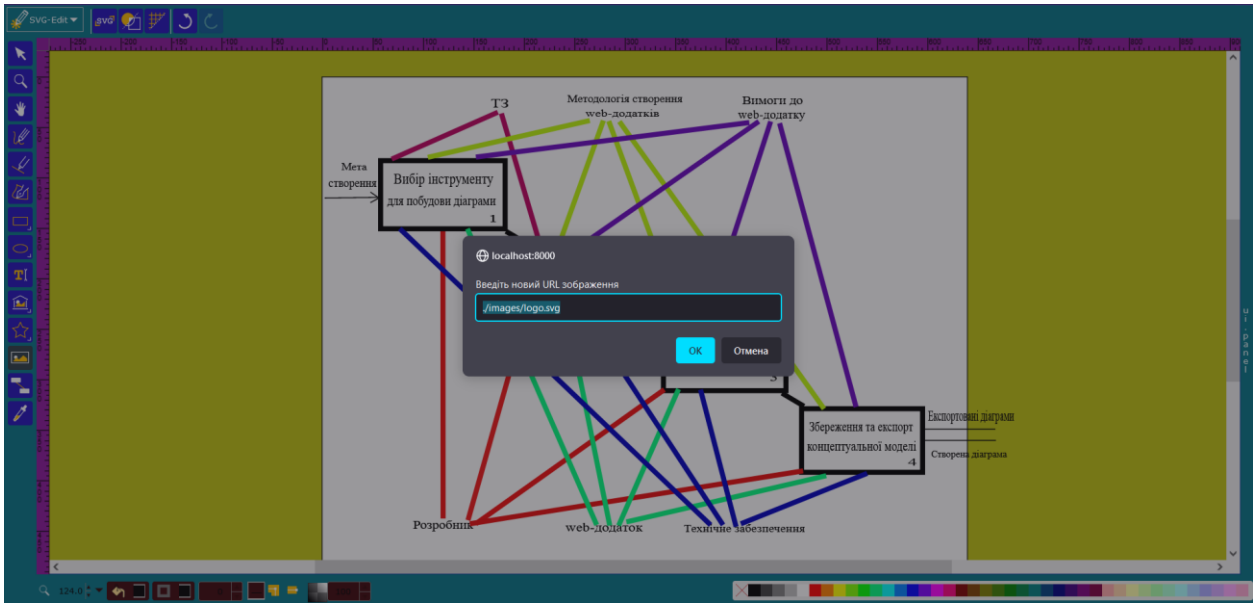


Рисунок 3.34 – Приклад застосування інструменту «Зображення»

Зірка - інструмент, який дозволяє користувачу малювати зіркові форми на моделі. Використовується для виділення, позначення або групування певних об’єктів або областей на моделі. На рисунку 3.35 представлено вигляд іконки інструменту «Зірка».

Рисунок 3.35 – Вигляд іконки інструменту «Зірка»

Бібліотека форм - це колекція попередньо визначених форм або символів, які користувач може вибрати та вставити на модель. Використовується для швидкого та зручного додавання стандартних форм або

об’єктів до моделі. На рисунку 3.36 представлено вигляд іконки інструменту «Бібліотека форм». Також на рисунках 3.37-3.38 представлено доступні символи та форми для використання.

Рисунок 3.36 – Вигляд іконки інструменту «Бібліотека форм»

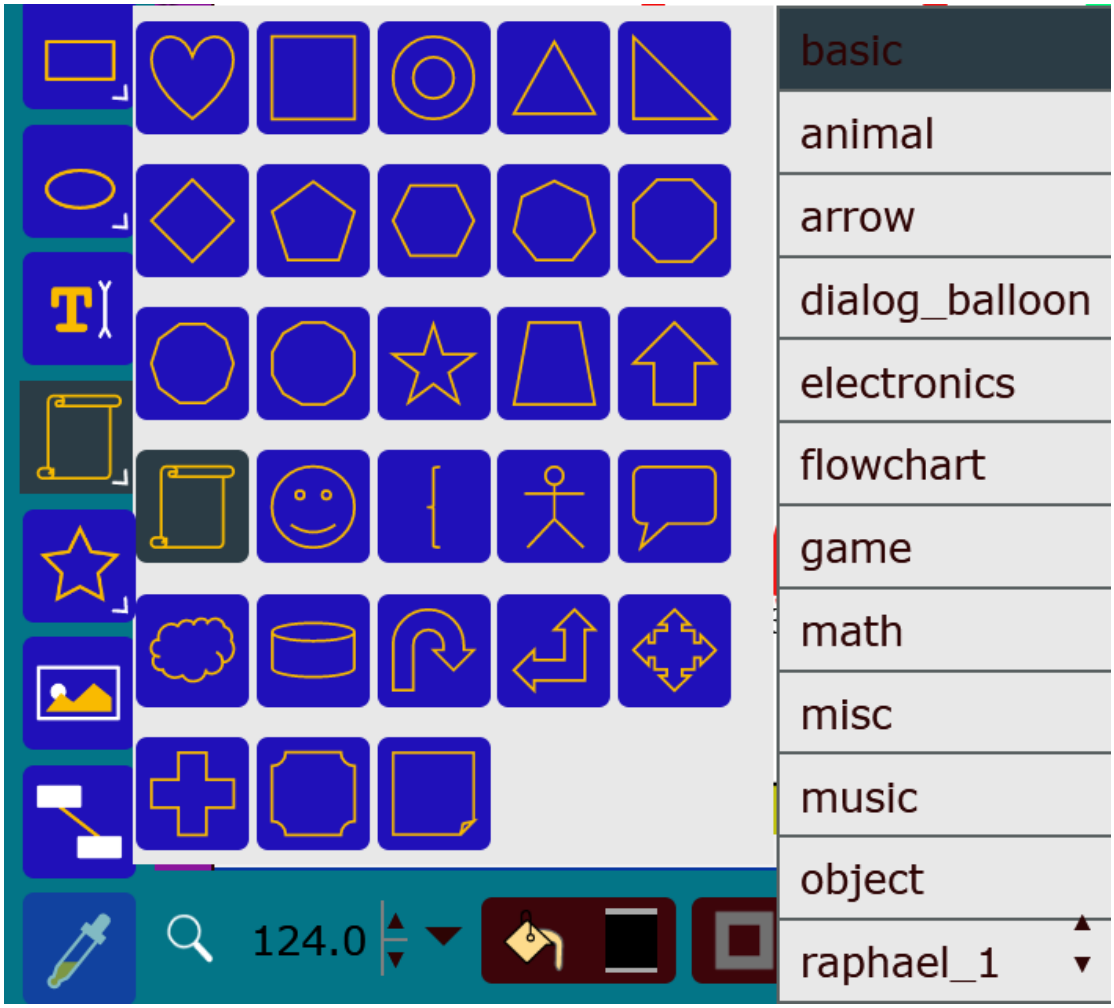


Рисунок 3.37 – Інстумент «Бібліотека форм». Класичні елементи

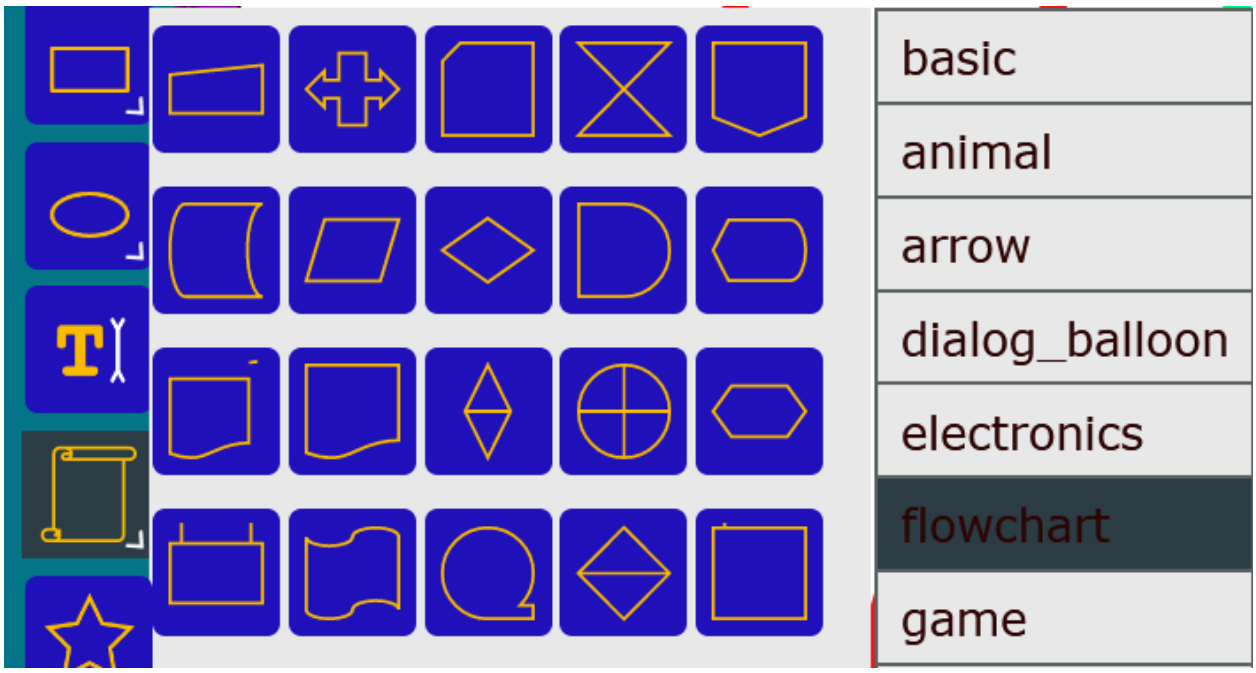


Рисунок 3.38 – Інструмент «Бібліотека форм». Елементи для діаграм

Після того як концептуальну модель було побудовано, користувач може зберегти її в зручному для нього форматі. Для цього на панелі налаштувань робочого середовища необхідно обрати «Save as SVG» або «Save SVG». Після того як користувач обрав збереження моделі в форматі SVG то браузер автоматично збереже модель в папку для завантажень. На рисунку 3.39 представлено результат збереження моделі в форматі SVG.

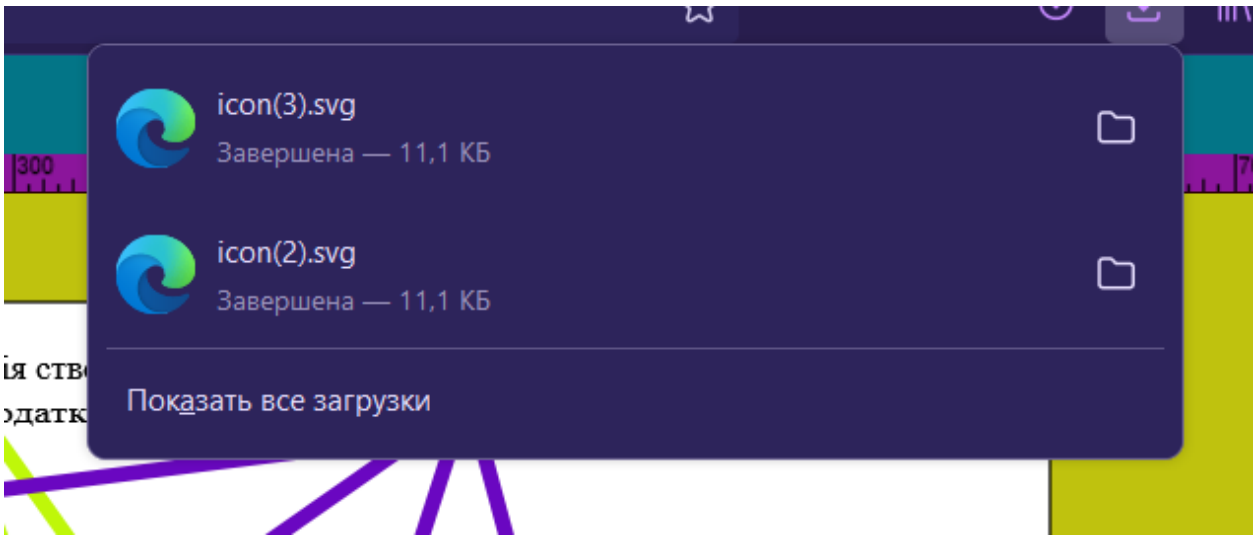


Рисунок 3.39 – Результат збереження моделі в форматі SVG

Також користувач може зберігати модель в різних розширеннях. Для цього доступна функція «Import». Дана функція надає користувачу вибір формату збереження моделі та налаштування якості. На рисунку 3.40 представлено вигляд функції збереження формату моделі.

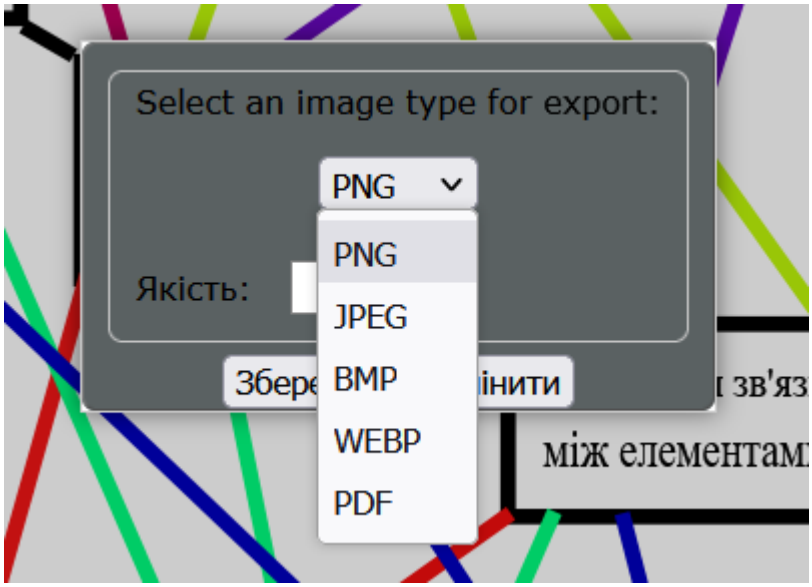


Рисунок 3.40 – Вигляд функції збереження формату моделі

Після того як користувач обрав потрібний формат файлу, браузер автоматично збереже модель в обраному форматі та виконає переадресацію на створений файл (рис.3.41).

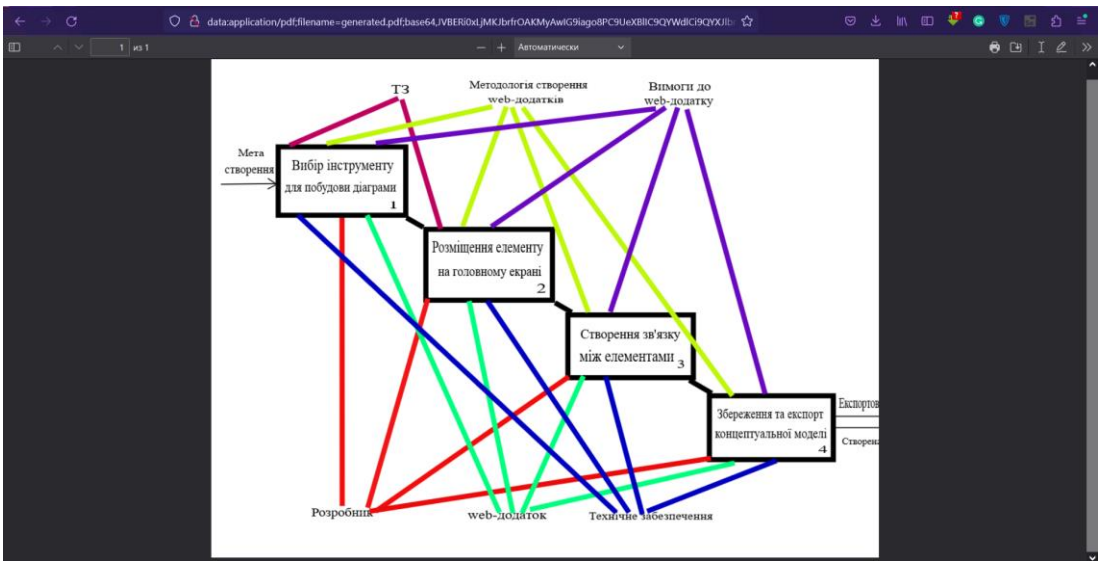


Рисунок 3.41 – Результат збереження моделі в форматі .pdf

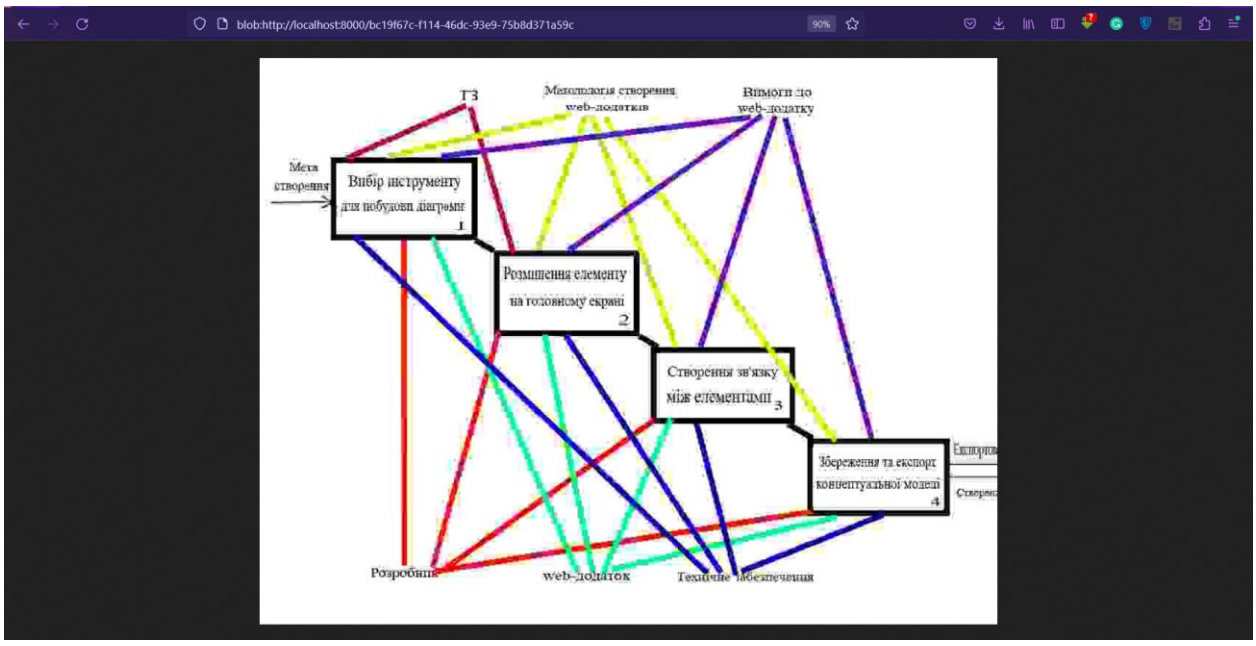


Рисунок 3.42 – Результат збереження моделі в форматі .jpeg

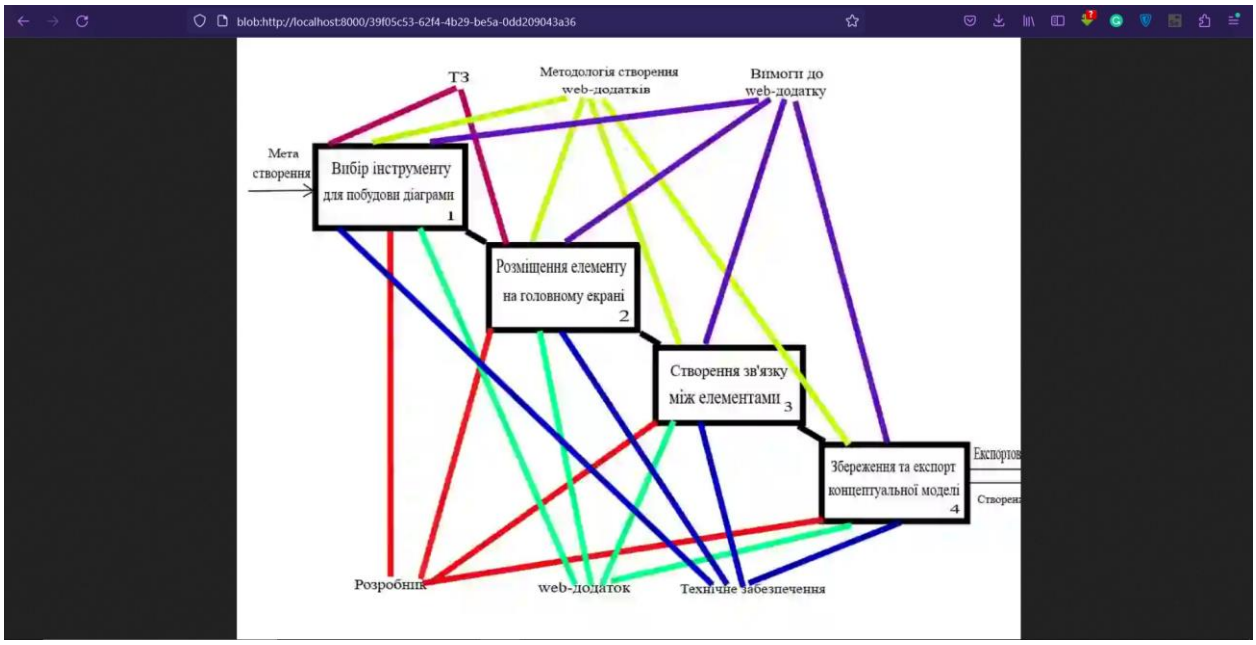


Рисунок 3.43 – Результат збереження моделі в форматі .webp

Також якщо користувач матиме бажання перевірити які сторонні зображення від додавав до файлу, то йому доступна функція «Image library». На рисунку 3.44 представлено вигляд функції перегляду сторонніх зображень.


Рисунок 3.44 – Вигляд функції перегляду сторонніх зображень

Також користувач може переглянути налаштування документа та змінити властивості під власні потреби (рис 3.45).

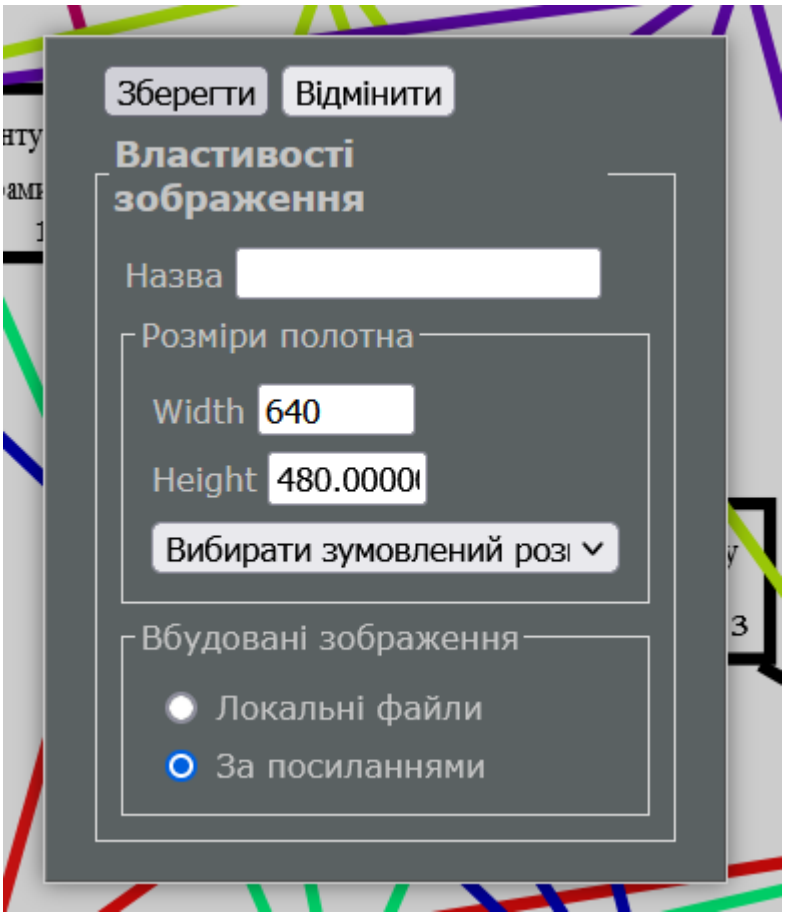


Рисунок 3.45 – Перегляд властивостей налаштування документа

Для модифікації робочого середовища та зміни мови користування, користувач може використати параметри в панелі налаштувань робочого середовища. На рисунку 3.46 представлено вигляд параметрів робочого середовища.

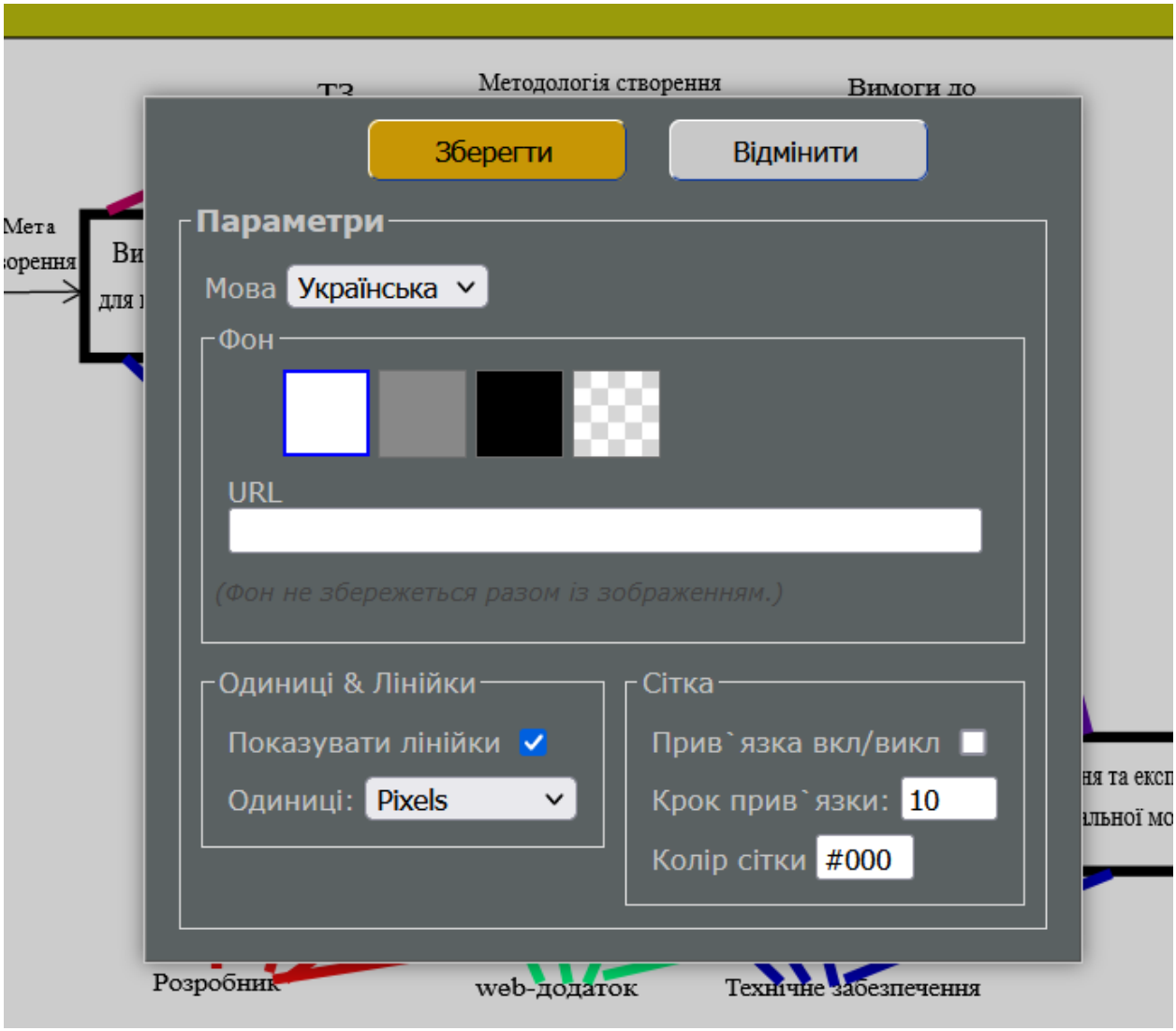


Рисунок 3.46 – Вигляд параметрів робочого середовища

Після було проведено тестування даного web-додатку. Виявлені у ході тестування помилки було виправлено.

# ВИСНОВОК

Під час реалізації web-додатку для візуального контролю концептуальних моделей було проведено аналіз потреб користувачів та визначено основні вимоги до програмного забезпечення. Шляхом використання відкритих джерел було здійснено дослідження предметної області, що дало змогу отримати інформацію про існуючі аналоги та їх переваги та недоліки.

Для реалізації додатку були обрані різноманітні інструменти, що дозволяють користувачу здійснювати різні дії з концептуальними моделями. Інструменти, такі як лупа, панорамування, олівець, лінія, контури, прямокутник, коло, текст, бібліотека форм, зірка, зображення, піпетка та створення зв’язку між об’єктами, надають користувачу можливість детально досліджувати, позначати, показувати зв’язки, малювати та додавати різноманітний контент до моделей.

Підсумковий web-додаток дозволяє виконувати візуальний контроль концептуальних моделей, надаючи зручні та ефективні інструменти для аналізу та маніпулювання моделями. Він може бути використаний для вивчення, редагування, показу та спільної роботи з концептуальними моделями у різних галузях, включаючи наукові дослідження, проектування, освіту та багато іншого.

Такий web-додаток має практичне застосування у різних сферах, де важливо візуалізувати та аналізувати концептуальні моделі. Він може полегшити співпрацю між фахівцями, допомогти в процесі навчання та розвитку нових ідей, а також сприяти кращому розумінню складних концепцій та взаємозв’язків у великих проектах чи системах.

Також було розроблено технчне завдання на розробку даного проєкту (додаток А) і виконано планування його робіт (додаток Б). Лістинг основних модулів представлено в додатку В.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. S. Smith, A. Johnson. "Web-Based Applications for Visualizing Conceptual Models." International Journal of Web Applications, vol. 10, no. 2, 2018, pp. 45-62.

2. K. Brown, J. Wilson. "Web Application Development for Visualizing Conceptual Models Using Virtual Reality." Proceedings of the International Conference on Web Development, 2019, pp. 127-134. https://doi.org/10.1007/978-3-319-98572-5_11

3. M. Thompson, L. Davis. "User Experience Design for Web Applications in Conceptual Model Visualization." Journal of User Interface Design, vol. 15, no. 3, 2020, pp. 78-92.

4. Mendonça, G.V., Silva, C.M., Pimentel, M., Barros, P. (2018). Web-Based Tools for Conceptual Modeling: A Systematic Literature Review. In: Conceptual Modeling. ER 2018. Lecture Notes in Computer Science, vol 11157. Springer, Cham. https://doi.org/10.1007/978-3-030-00847-5_9

5. Luo, L., Hammad, A., Watters, C. (2020). Web-Based Conceptual Modeling: Challenges, Techniques, and Future Directions. Information Systems Frontiers, 22(5), 1065-1086. https://doi.org/10.1007/s10796-019-09964-1

6. Haider, M.A., Younas, M., Sandhu, K. (2017). A Web-Based Conceptual Modeling Framework for Information Systems Development. Journal of Information Systems and Technology Management, 14(1), 45-68. https://doi.org/10.4301/S1807-1775201714013

7. Офіційна сторінка web-додатку «Lucidchart» [Електронний ресурс] - Режим доступу до ресурсу: https://www.lucidchart.com. Дата звертання 12.04.2025

8. Офіційна сторінка web-додатку «draw.io» [Електронний ресурс] - Режим доступу до ресурсу: https://www.drawio.com/. Дата звертання 12.04.2025

9. Офіційна сторінка web-додатку «Creately» [Електронний ресурс] - Режим доступу до ресурсу: https://creately.com.

10. Офіційна сторінка React.js [Електронний ресурс] - Режим доступу до ресурсу: https://legacy.reactjs.org/. Дата звертання 12.04.2025

11. Офіційна сторінка Angular [Електронний ресурс] - Режим доступу до ресурсу: https://angularjs.org/. Дата звертання 12.04.2025

12. Офіційна сторінка Vue.js [Електронний ресурс] - Режим доступу до ресурсу: https://vuejs.org/. Дата звертання 12.04.2025

13. What is a Context Diagram (and How Can You Create One) [Електронний ресурс] - Режим доступу до ресурсу: https://venngage.com/blog/context-diagram/.

14. Functional Decomposition ) [Електронний ресурс] - Режим доступу до ресурсу: https://binaryterms.com/functional-decomposition.html.

15. Офіційна сторінка Visual Studio Code [Електронний ресурс] - Режим доступу до ресурсу: https://code.visualstudio.com/.

16. SVG/Tutorial [Електронний ресурс]. — Режим доступу: URL : https://developer.mozilla.org/ru/docs/Web/SVG/Tutorial — Заголовок з екрану. Дата звертання 12.05.2025.

17. Scalable Vector Graphics (SVG) [Електронний ресурс]. — Режим доступу: URL : http://msdn.microsoft.com/en-us/library/ff971903(v=vs.85).aspx — Заголовок з екрану. Дата звертання 12.05.2025.

18. Coyier Chris. Practical SVG.- A Book Apart, 2016. — 154 p. — ISBN: 978-1-937557-43-0.

19. Larsen Rob. Mastering SVG. Packt Publishing, 2018. — 312 p.

20. Dailey D., Frost J., Strazzullo D. Building Web Applications with SVG. Microsoft Press, 2012. — 293 p. — ISBN10: 0735660123, ISBN13: 978-0-7356-6012-0.

21. Libby Alex. Beginning SVG : A Practical Introduction to SVG Using Real-World Examples.– Apress, 2018. — 305 p. — ISBN: 978-1484237595.

22. Drasner Sarah. SVG Animations. O'Reilly Media, 2017. — 246 p. — ISBN: 978-1491939703.

23. Ben Frain. Responsive Web Design with HTML5 and CSS3. Second Edition. PACKT publishing. 2017. – I312 pp – SBN: 9781784398934 ) [Електронний ресурс]. — Режим доступу: URL : https://www.packtpub.com/web-development/responsive-web-design-html5-and-css3-second-edition?utm_source=github&utm_medium= repository& utm_campaign= 9781784398934 – Заголовок з екрану. Дата звертання 12.05.2025.

## ДОДАТОК А

# ТЕХНІЧНЕ ЗАВДАННЯ

**на «Web-додаток для візуального контролю концептуальних моделей»**

**Суми 2023**

## 1. Призначення й мета створення web-додатку

### 1.1 Призначення web-додатку

Web-додаток для візуального контролю концептуальних моделей призначений для надання користувачам зручного та ефективного інструментарію для аналізу, редагування та візуалізації концептуальних

моделей. Він дозволяє користувачам взаємодіяти з моделями шляхом малювання, додавання тексту, форм та зображень, створення зв’язків між об’єктами, а також здійснювати маніпуляції з областями моделі, такими як збільшення, панорамування та редагування.

### 1.2 Мета створення 3D моделі

Створити web-додаток, який надає можливість візуального контролю концептуальних моделей безпосередньо у web-середовищі. Додаток має забезпечувати зручний та інтуїтивно зрозумілий інтерфейс для користувачів, щоб вони могли створювати, редагувати та аналізувати концептуальні моделі у вигляді графів.

### 1.3 Цільова аудиторія

Цільовою аудиторією для web-додатку візуального контролю концептуальних моделей є широкий спектр користувачів, включаючи фахівців із сфери бізнесу, інформаційних технологій та проектного менеджменту.

## 2 Вимоги до web-додатку

### 2.1 Вимоги web-додатку в цілому

Web-додаток повинен надавати можливість створювати нові концептуальні моделі та редагувати існуючі. Користувачі повинні мати зручний інтерфейс для додавання та видалення об’єктів, зв’язків та атрибутів у моделі. Також додаток повинен забезпечувати можливість відображення концептуальних моделей у вигляді діаграм та мати можливість зручно виконувати навігацію по моделі, збільшувати та зменшувати масштаб, змінювати перегляд моделі для отримання більш детального або загального уявлення про модель. Додаток повинен підтримувати зручний інтерфейс для керування файлами та забезпечувати можливість експортувати моделі в різних форматах (наприклад, PDF, зображення).

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

Web-додаток має забезпечити зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє легко виконувати навігацію та взаємодіяти з додатком. Також необхідно надати можливості створення нових проектів, відкриття та редагування існуючих проектів, а також збереження проектів у форматі, який легко експортувати та імпортувати. Головним елементом є забезпечення інтуїтивно зрозумілого інтерфейсу для створення та редагування концептуальних моделей.

### 2.1.2 Вимоги до персоналу

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації web-додатку, окрім загальних навичок роботи з персональним комп’ютером.

### 2.1.3 Вимоги до збереження інформації

Усі дані будуть зберігатися на персональному комп’ютері користувача.

### 2.1.4 Системні вимоги до побудови та використання web-додатку

Системні вимоги для розробки і використання моделі повинні відповідати таким:

Реалізація webсайту відбувається з використанням:

– JQuery 3.6.4
– JavaScript 1.8.5
– ReactJS 18.2.0

# 3 Структура web-додатку

## 3.1 Загальна інформація про структуру web-додатку

Web-додаток для візуального контролю концептуальних моделей має певну структуру, яка допомагає забезпечити ефективну і зручну роботу з додатком. Основні компоненти структури включають:

Front-end (клієнтська сторона): Це частина додатку, з якою користувач взаємодіє безпосередньо через свій web-браузер. Front-end включає в себе візуальну частину додатку, яка відображається на екрані користувача. Це набір сторінок, форм, кнопок, меню та інших елементів інтерфейсу, які дозволяють користувачу взаємодіяти з функціональністю додатку. Front-end частина web-додатку також включає JavaScript-код, який забезпечує динамічну взаємодію з додатком без потреби перезавантаження сторінки.

Back-end (серверна сторона): Це частина додатку, яка відповідає за обробку запитів користувача і забезпечення необхідних даних і функціональності. Бекенд написаний на мові програмування JavaScript (з використанням фреймворку, наприклад, ReactJS).

### 3.2 Дизайн та структура web-додатку

Додаток буде надавати зручний інтерфейс для створення, редагування та відображення моделей, що допоможе користувачам легко визначати залежності між концептами і візуально аналізувати їх структуру.

## 4. Склад і зміст робіт розробки web-додатку

Докладний опис етапів роботи розробки web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи розробки web-додатку

| № | Склад і зміст робіт | Строк розробки (у робочих днях) |
|---|---|---|
| 1 | Постановка цілей необхідних для досягнення певного результату | 3 дні |
| 2 | Складання технічного завдання | 3 дні |
| 3 | Налаштування середовища розробки web-доадтку | 2 дні |
| 4 | Пошуки розширень для розробки web-додатку | 2 дні |
| 5 | Розробка основних модулів web-додатку | 3 дні |
| 6 | Робота над оптимізацією | 2 дні |
| 7 | Створення компонетів web-додатку | 5 днів |
| 8 | Перевірка працездатності web-додатку | 1 день |
| 9 | Завершення роботи та оформлення документації | 5 днів |
| | Загальна тривалість робіт | 26 днів |

## ДОДАТОК Б

## Планування робіт

Метою даної роботи є створення web-додатку для візуального контролю концептуальних моделей без пунктів. Цей додаток має на меті забезпечити користувачів зручним і ефективним інструментом для візуального аналізу та контролю концептуальних моделей без необхідності використання традиційних пунктів.

Для досягнення цієї мети необхідно виконати наступні задачі:

– Визначення актуальності роботи та проведення дослідження предметної області. Це включає огляд існуючих методів контролю концептуальних моделей та виявлення проблем, які можуть виникнути при використанні традиційних пунктів.

– Аналіз аналогів та існуючих рішень у сфері візуального контролю концептуальних моделей. Цей аналіз дозволить виявити сильні та слабкі сторони існуючих додатків та використати ці знання для розробки ефективного та інноваційного рішення.

– Моделювання ігрового додатку. На цьому етапі будуть розроблені концептуальні моделі та архітектура додатку, які відповідатимуть потребам користувачів і забезпечуватимуть зручний та ефективний візуальний контроль.

– Розробка та реалізація структури додатку для створення системи візуального контролю концептуальних моделей без пунктів. Цей етап включає створення відповідних інтерфейсів, функцій та інструментів, які дозволять користувачам легко та ефективно працювати з концептуальними моделями.

В результаті виконання цих задач очікується створення web-додатку, який забезпечить користувачів зручним та ефективним інструментом для візуального контролю концептуальних моделей без необхідності використання пунктів. Цей додаток може стати цінним ресурсом для професіоналів у галузі концептуального моделювання та сприяти покращенню якості роботи та ефективності процесу прийняття рішень.

## Деталізація мети проекту методом SMART

SMART (Specific, Measurable, Achievable, Relevant, Time-bound) - це абревіатура, яка представляє собою мнемонічний або мнемонічно-акронімічний підхід до формулювання та досягнення метей. Кожна літера у складі SMART відповідає певному критерію, який допомагає зрозуміти та конкретизувати поставлену мету:

Specific (Специфічна): Мета повинна бути чітко сформульована та конкретизована, щоб уникнути неоднозначності та розмитості.

Measurable (Вимірювана): Мета повинна бути вимірювана або кількісно, або якісно, для того, щоб мати засоби оцінки прогресу та досягнення цілі.

Achievable (Досяжна): Мета повинна бути реалістичною та досяжною у контексті ресурсів, знань та навичок, які доступні для досягнення цілі.

Relevant (Зв’язана): Мета повинна бути пов’язана з контекстом, стратегічними цілями та потребами організації або індивіда. Вона повинна мати значення та відповідати потребам та цілям.

Time-bound (Обмежена в часі): Мета повинна бути обмежена в часі шляхом встановлення конкретного терміну або крайнього строку для досягнення.

Результати формування мети проєкту за методом SMART наведено у таблиці Б.1.

Таблиця Б.1 – Формалізація мети за технологією SMART

| | |
|---|---|
| Specific | Розробка web-додатку для візуального контролю концептуальних моделей |

| | |
|---|---|
| Measurable | Результат допоможе користувачам створювати, редагувати та переглядати концептуальні моделі відповідно до їх потреб. |
| Achievable | Проєкт реалізовується у відповідності до рівня досвіду та на основі затвердженого ТЗ. |
| Revelant | Web-додаток забезпечить зручну та ефективну можливість візуального контролю концептуальних моделей. |
| Time-bound | Проєкт виконується враховуючи встановлені на ранньому етапі обмеження в часі (червень 2023). |

## WBS

WBS (Work Breakdown Structure), або структура розбиття робіт, є методологічним інструментом управління проектами, який використовується для розбиття комплексних проектів на менші, більш керовані та керовані робочі пакети. WBS дозволяє подробиць розкласти проект на окремі фрагменти, елементи та завдання, що утворюють ієрархічну структуру робіт.

WBS визначає ієрархічну структуру проекту, де кожен робочий пакет визначається і описується як окрема задача або елемент проекту. Цей підхід дозволяє розкрити проект на більш зрозумілі та керовані елементи, забезпечуючи керівникам проектів можливість більш ефективного планування, оцінки ресурсів, управління та контролю прогресу проекту.

За допомогою WBS можна визначити послідовність та залежності між робочими пакетами, розподілити відповідальності між учасниками проекту, встановити часові та вартісні оцінки для кожного елементу робіт. Крім того, WBS дозволяє уникнути дублювання робіт та незрозумілих завдань, забезпечуючи чітку структуру та сприяючи ефективному управлінню проектом [].

## OBS

Організаційна структура виконавців (OBS) - це концептуальна модель, що використовується в проектному управлінні для організації та структурування робочих груп або команд, які виконують завдання проекту. OBS визначає ієрархічну структуру, в якій вказується, які робочі одиниці або виконавці відповідають за виконання конкретних робіт або етапів проекту.

Організаційна структура виконавців допомагає управляти проектом, забезпечуючи чіткість і відповідальність виконавців. Вона може бути представлена у вигляді деревоподібної структури, де батьківські вузли представляють вищі рівні керівництва, а дочірні вузли - робочі групи або індивідуальних виконавців. Кожен виконавець або група в OBS має чітко визначені обов’язки, функції та ролі, що дозволяє ефективно координувати та контролювати виконання завдань проекту.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, що функціонують в проекті описано в таблиці Б.2.

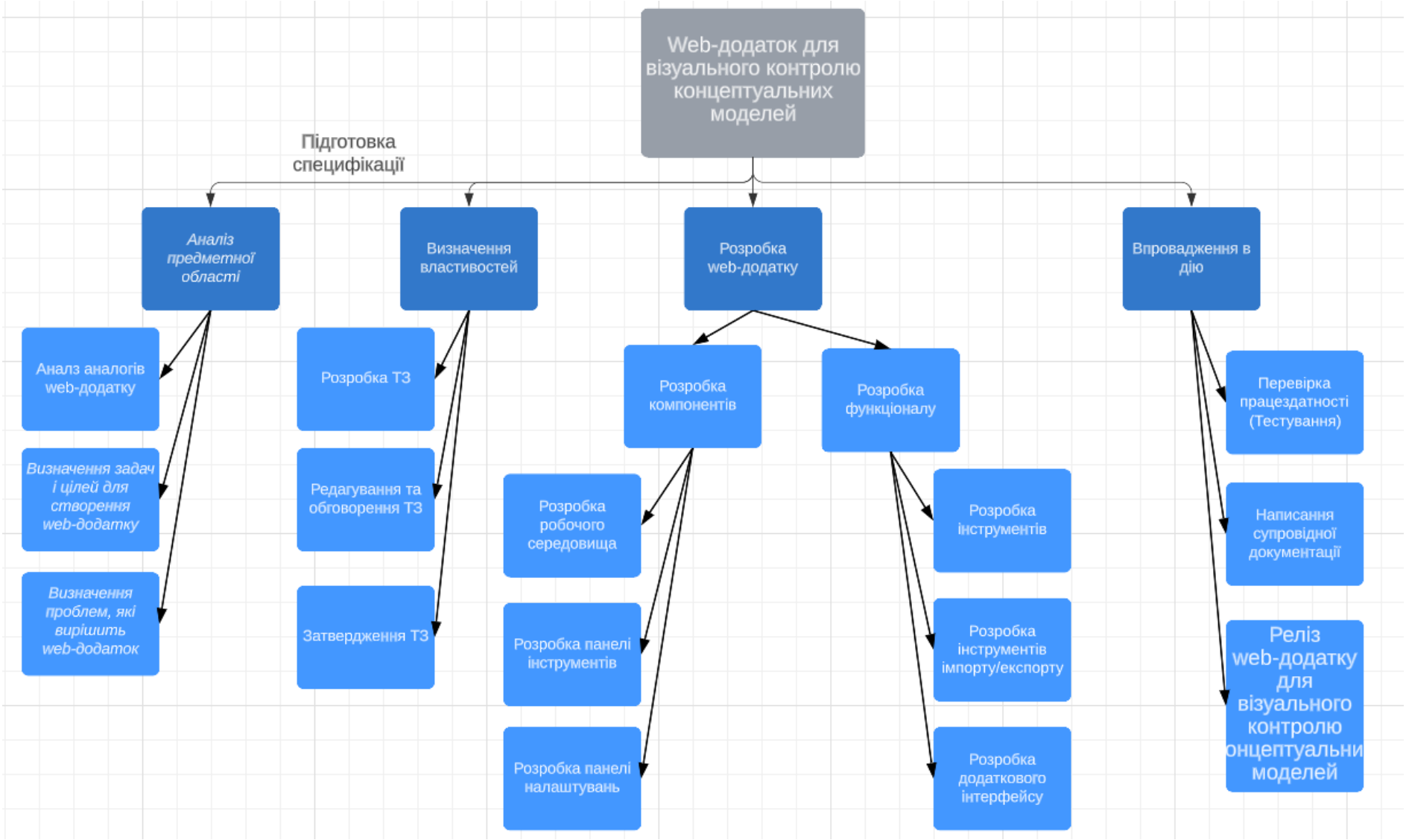


Рисунок Б.1 – WBS-структура робіт проекту

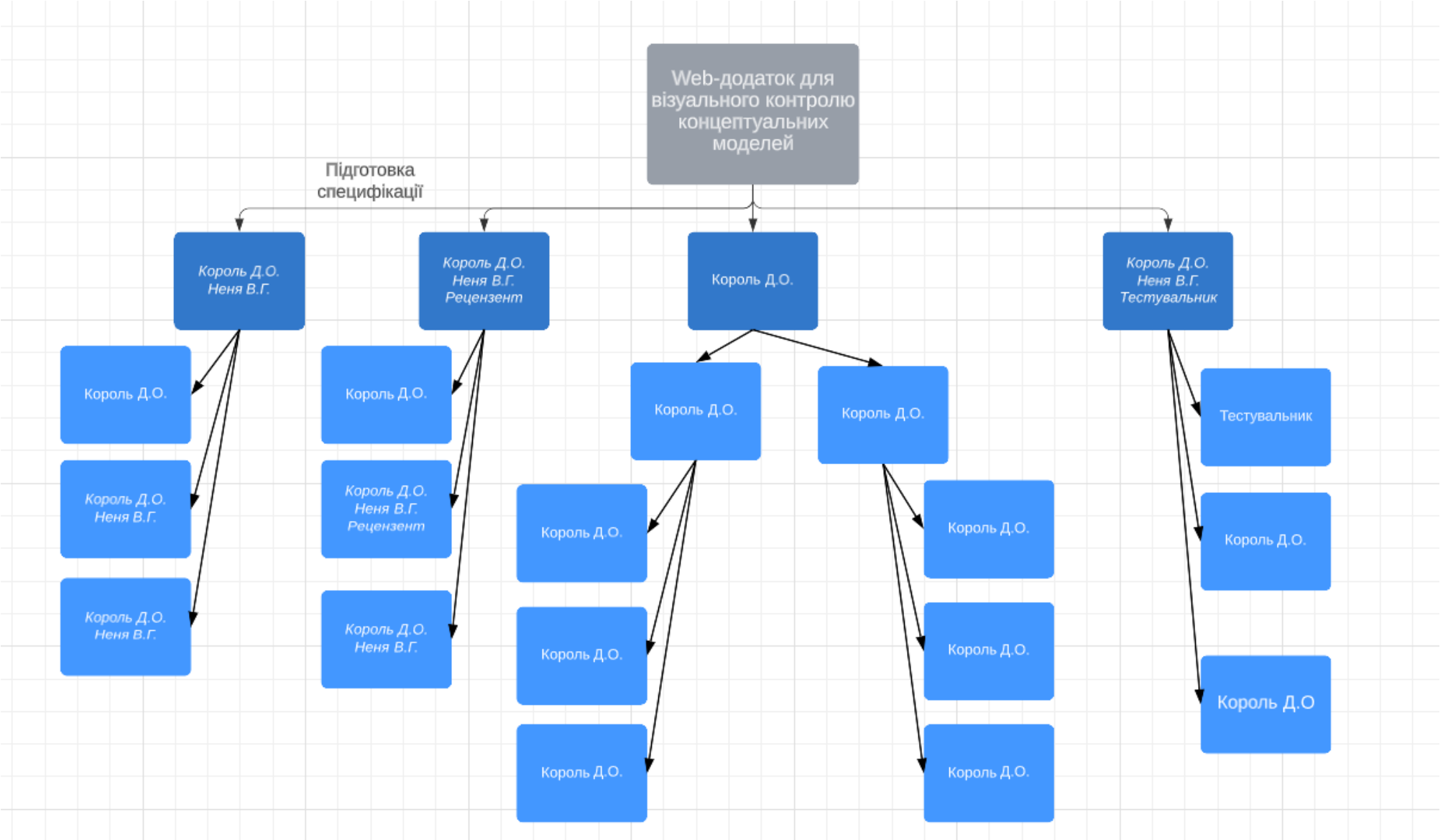


Рисунок Б.2 – OBS-структура робіт проекту

Таблиця Б.2 – Виконавці проєкту

| Роль | Ім’я | Проєктна роль |
|---|---|---|
| Керівник проєкту | Неня В.Г. | Відповідає за виконання термінів, розподіл ресурсів, та завдань між учасниками. Виконує збір та аналіз даних. |
| Розробник | Король Д.О. | Виконує розробку модулів |
| Тестувальник | Король Д.О. Тестувальник | Знаходження помилок до релізу web-додатку |

## Діаграма Ганта

Діаграма Ганта представляє собою горизонтальну смугу, що представляє загальний час проекту, розбиту на дискретні відрізки, які відповідають окремим завданням або фазам проекту. Завдання зображаються у вигляді прямокутників на діаграмі, причому їхня довжина відповідає тривалості завдання. Додатково, діаграма може містити інформацію про ресурси, які призначені для виконання кожного завдання, та мілістоуни, які є ключовими точками у проекті.

Діаграма Ганта дозволяє проектним менеджерам та командам проекту відстежувати та керувати ходом робіт, встановлювати пріоритети, визначати залежності між завданнями та аналізувати ресурси, необхідні для успішного виконання проекту. Вона також допомагає уникнути перекриття завдань, затримок та планових змін у проекті.

Календарний графік проєкту представлено на рисунках Б.3-Б.5.

| | Режим задачи | Название задачи | Длительн | Начало | Окончание | Предшественник | Названия ресурсов | Витрат |
|---|---|---|---|---|---|---|---|---|
| 1 | | **Розробка web-додатку для візуального контролю концептуальних моделей** | 93 дней | Ср 16.11.22 | Пт 24.03.23 | | | |
| 2 | | Ініціалізація проекту | 4 дней | Ср 16.11.22 | Пн 21.11.22 | | | |
| 3 | | Опрашювання ТЗ | 1 день | Ср 16.11.22 | Ср 16.11.22 | | Король Д.О. | |
| 4 | | Аналіз аналогів | 1 день | Ср 16.11.22 | Чт 17.11.22 | 3 | Король Д.О. | $1000 |
| 5 | | Пошук аналогів-додатків | 1 день | Чт 17.11.22 | Пт 18.11.22 | 4 | Король Д.О. | $1,116 |
| 6 | | Ідентифікація проекту | 1 день | Пт 18.11.22 | Пн 21.11.22 | 5 | Неня В.Г. | $72.00 |
| 7 | | **Планування проекту** | **15 дней** | Пн 21.11.22 | Пн 12.12.22 | 6 | | $72.00 |
| 8 | | Структура проекту | 2 дней | Пн 12.12.22 | Ср 14.12.22 | 7 | Король Д.О. | $72.00 |
| 9 | | Постановка мети | 2 дней | Ср 14.12.22 | Пт 16.12.22 | 8 | Король Д.О. | $900.0 |
| 10 | | Розробка web-додатку для візуального контролю концептуальних моделей» | 5 дней | Пт 16.12.22 | Пт 23.12.22 | 9 | Король Д.О. | $1,432 |
| 11 | | Управління ресурсами | 1 день | Пт 23.12.22 | Пн 26.12.22 | 10 | Король Д.О. | $72.00 |
| 12 | | Управління фінансами | 1 день | Пн 26.12.22 | Вт 27.12.22 | 11 | Король Д.О. | $72.00 |
| 13 | | Управління якістю | 1 день | Вт 27.12.22 | Ср 28.12.22 | 12 | Король Д.О. | $100.0 |
| 14 | | Оптимізація плану | 1 день | Ср 28.12.22 | Чт 29.12.22 | 13 | Король Д.О. | $72.00 |
| 15 | | Затвердження плану | 1 день | Чт 29.12.22 | Пт 30.12.22 | 14 | Неня В.Г. | $72.00 |
| 16 | | **Реалізація проекту** | 47 дней | Пн 02.01.23 | Вт 07.03.23 | | Король Д.О. | $7,020 |
| 17 | | **Розробка web-додатку** | 21 дней | Пн 02.01.23 | Пн 30.01.23 | | Король Д.О. | $3,250 |
| 18 | | Реалізація компонентів та налаштування серверу | 9 дней | Вт 31.01.23 | Пт 10.02.23 | | Король Д.О. | $1,300 |
| 19 | | Верстка Front-end | 8 дней | Пт 10.02.23 | Ср 22.02.23 | 18 | Король Д.О. | $1,500 |
| 20 | | Розробка функцій | 5 дней | Ср 22.02.23 | Ср 01.03.23 | 19 | Король Д.О. | $350.0 |
| 21 | | Розробка системи налаштувань | 3 дней | Ср 01.03.23 | Пн 06.03.23 | 20 | Король Д.О. | $100.0 |
| 22 | | Створення локального | 1 день | Пн 06.03.23 | Вт 07.03.23 | 21 | Король Д.О. | $100.0 |

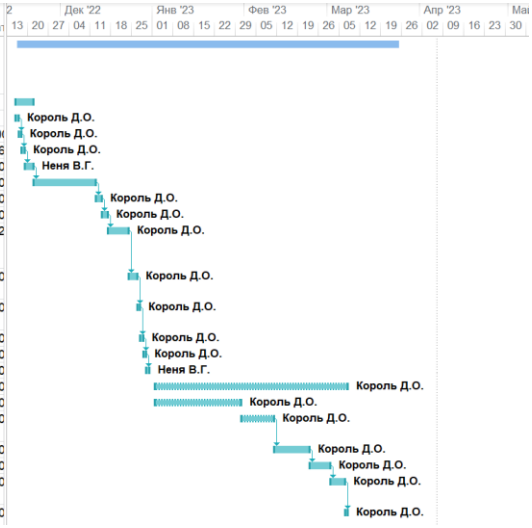


Рисунок Б.3 – Діаграма Ганта. Частина 1

| | Режим задачи | Название задачи | Длительн | Начало | Окончание | Предшественник | Названия ресурсов | Витрат |
|---|---|---|---|---|---|---|---|---|
| 22 | | Створення локального серверу | 1 день | Пн 06.03.23 | Вт 07.03.23 | 21 | Король Д.О. | $100.0 |
| 23 | | **Розробка модулів створення концептуальних моделей** | **21 дней** | Ср 08.03.23 | Ср 05.04.23 | | Король Д.О. | $3,270 |
| 24 | | Розробка модулю для вибору інструментів та застосування на робочому середовищі | 1 день | Ср 08.03.23 | Ср 08.03.23 | | Король Д.О. | $270.0 |
| 25 | | Розробка модулю налаштувань додатку | 1 день | Ср 08.03.23 | Чт 09.03.23 | 24 | Король Д.О. | $1,500 |
| 26 | | Розробка модулю експорту/імпорту моделей | 2 дней | Чт 09.03.23 | Пн 13.03.23 | 25 | Король Д.О. | $1,500 |
| 27 | | Реалізація модулю зміни кольору | 4 дней | Пн 13.03.23 | Пт 17.03.23 | 26 | Король Д.О. | $500.0 |
| 28 | | Тестування | 2 дней | Пт 17.03.23 | Вт 21.03.23 | 27 | Король Д.О. | $500.0 |
| 29 | | **Завершення проекту** | 6 дней | Вт 21.03.23 | Ср 29.03.23 | 28 | | $432.0 |
| 30 | | Створення документації | 1 день | Ср 29.03.23 | Чт 30.03.23 | 29 | Король Д.О. | $72.00 |
| 31 | | Створення інструкції користувача | 2 дней | Пт 31.03.23 | Пн 03.04.23 | 30 | Король Д.О. | $72.00 |
| 32 | | Запуск і підтримка | 2 дней | Вт 04.04.23 | Ср 05.04.23 | 31 | Король Д.О. | $72.00 |
| 33 | | Архівація проекту | 1 день | Чт 06.04.23 | Чт 06.04.23 | 32 | Король Д.О. | $72.00 |

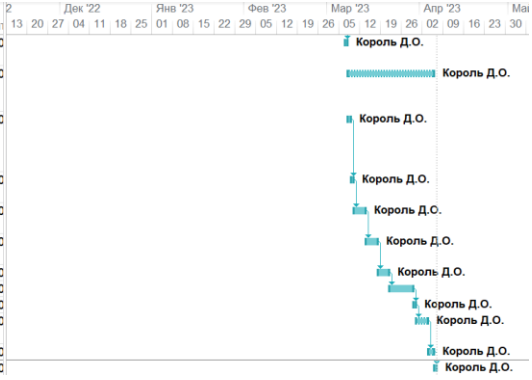


Рисунок Б.3 – Діаграма Ганта. Частина 2

## Управління ризиками проєкту

Під час виконання якісної оцінки ризиків треба визначити ризики, які мають бути усунені якнайшвидше. У залежності від ступеня важливості ризику – реагування буде відповідне. Наступним етапом є виконання кількісного оцінювання ризиків. Кількісне та якісне оцінювання можуть виконувати одночасно або окремо, що залежить від ступеня забезпечення проекту. У таблиці Б.3 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

| **Оцінка** | **Ймовірність виникнення** | **Вплив ризику** | **Тип ризику** |
|---|---|---|---|
| 1 | Низька | Низький | Прийнятні |
| 2 | Середня | Середній | Виправдані |
| 3 | Висока | Високий | Недопустимі |

Для того, щоб знизити негативний вплив ризиків на проект треба виконати планування реагування на них. До нього входить визначення ефективності розробки та оцінка наслідків впливу на проект. Оцінювання виконується за показниками, що описані в таблиці Б.3. У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку Б.4. Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

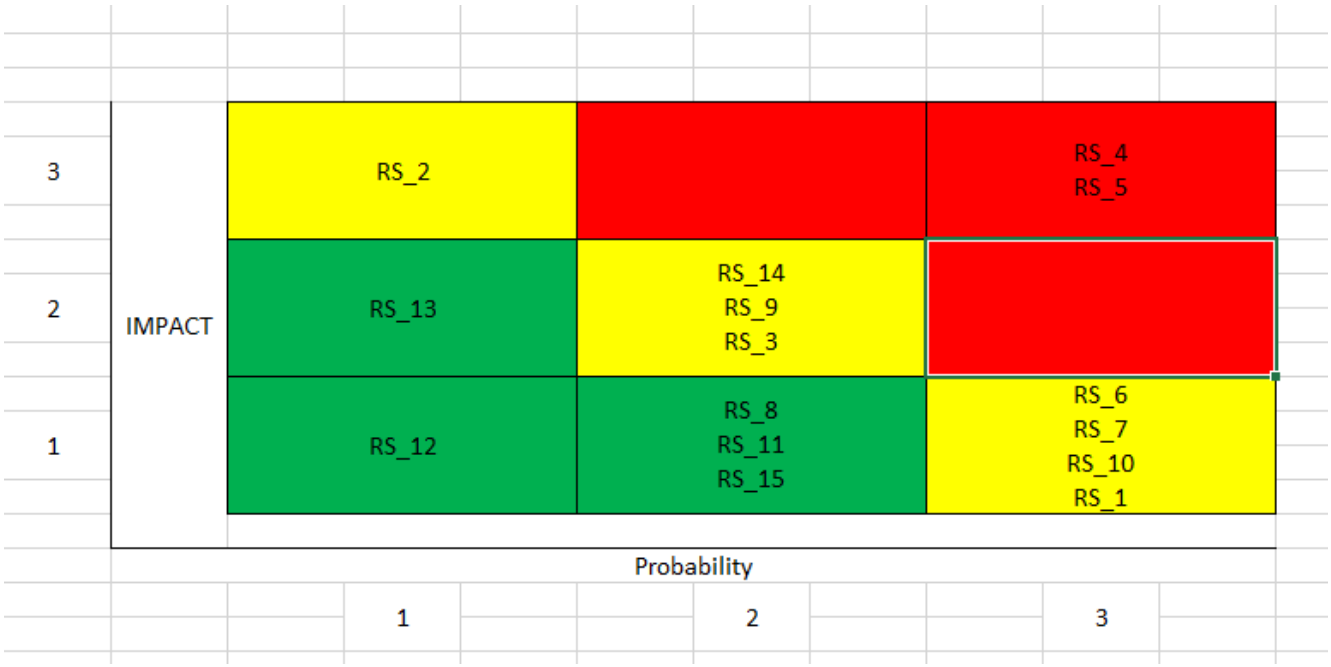


Рисунок Б.4. – Матриця ймовірності

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.4. У таблиці Б.5 описано ризики та стратегії реагування на кожен з них.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику.

| № | Назва | Межі | Ризики, які входять |
|---|---|---|---|
| 1 | Прийнятні | 1<R<2 | 8,11,12,13,15 |
| 2 | Виправдані | 3<R<4 | 1,2,3,6,7,9,10,14 |
| 3 | Недопустимі | 6<R<9 | 4,5 |

Таблиця Б.5 – Ризики та стратегії реагування

| ID | Статус ризику | Опис ризику | Ймовірність виникнення | Вплив ризику | Ранг ризику | План А | Тип стратегії реагування | План Б |
|---|---|---|---|---|---|---|---|---|
| RS_1 | Відкритий | Непорозуміння між розробником та замовником | Низька | Середній | 3 | 1.Налагодити гарні відносини між розробником та керівником.<br>2.Дотримуватися ділового етикету спілкування.<br>3.Створити комфортні умови для співпраці. | Попередження | При виявлення непорозуміння потрібно вияснити, що саме стало причиною непорозуміння обсудити її та створити здорову атмосферу в колективі. |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_2 | Відкритий | Поява альтернативного додатку | Низька | Середній | 4 | 1.Провести попереднє дослідження.<br>2.Вибрати унікальну стратегію створення додатку. | Прийняття | Змінити ідею створюваного додатку |
| RS_3 | Відкритий | Низька кваліфікація розробників | Середня | Середній | 4 | 1.Підвищити кваліфікацію персоналу.<br>2.Використати онлайн-ресурси для підвищення рівня знань | Пом’якшення | Врахувати час на підготовку працівників. Видати літературу, переглянути онлайн-уроки. |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_4 | Відкритий | Нечіткий тест план | Середня | Високий | 6 | 1.Ясно і однозначно обговорити із замовником усі види вимог.<br><br>2.Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів.<br><br>3.Періодичний контроль замовником етапів роботи. | Попередження | При виявленні невідповідностей деяких характеристик продукту заявленим вимогам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_5 | Відкритий | Неоптимальний розподіл часу | Висока | Високий | 9 | Провести аналіз актуальності найважливіших процесів та робіт. Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Чітко дотримуватися календарного плану | Пом’якшення | Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи з вже існуючою розстановкою. Обговорити варіанти внесення поправок до термінів реалізації із замовником. |

Продовження табл. Б.5.

| ID | Статус ризику | Опис ризику | Ймовірність виникнення | Вплив ризику | Ранг ризику | План А | Тип стратегії реагування | План Б |
|---|---|---|---|---|---|---|---|---|
| RS_7 | Відкритий | Вибір неефективної технології розробки и тестування | Середня | Середній | 4 | 1.Проаналізувати методи та засоби, для виконання додатку.<br>2.Обрати зрозумілу та легку в використанні технологію тестування | Пом’якшення | Виділити час та ресурсі на пошуки покращення обраної технології.<br>Застосувати допоміжні ресурси. |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_8 | Відкритий | Неправильна оцінка в масштабі додатку | Низька | Середній | 2 | 1.Провести детальний аналіз проекту. 2.Визначити основні етапу проекту, розподілити час на їх виконання. 3.Проаналізувати масштаби проекту на основі додаткових джерел. | Пом'якшення | Переоцінка масштабів проекту.<br><br>Перебудова стратегії реалізації проекту |
| RS_9 | Відкритий | Помилки наповнення контенту | Середня | Середній | 4 | На етапі наповнення контентом додатку тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати | Пом'якшення | Здійснювати проміжний контроль результатів в ході виконання проекту. |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_10 | Відкритий | Збої в роботі програмного забезпечення | Низька | Середній | 3 | 1.Підготувати резерв програмних засобів. 2.Залучити спеціаліста для усунення збоїв. | Попередження | Замінити програмне забезпечення |
| RS_11 | Відкритий | Відсутність резервних копій даних | Низька | Середній | 2 | 1.Налаштувати автоматичне збереження даних. 2.Зберігати дані на різних носіях інформації. | Попередження | Робити копію даних після кожного виконаного етапу. |
| RS_12 | Відкритий | Тестування непотрібних компонентів web-додатку | Низька | Низький | 1 | Попередити замовника про можливість додаткових тестувань web-додатку. | Використання | Обговорити вимоги і збитки від можливих змін проекту. |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_13 | Відкритий | Невикористання моніторингу проекту | Середня | Низький | 2 | Здійснювати проміжний контроль результатів в ході виконання проекту. Здійснювати моніторинг проекту працівниками. | Перенесення | Здійснювати моніторинг проекту замовником. Надання проміжних результатів виконання проекту після кожного етапу. |
| RS_14 | Відкритий | Виникнення проблем із програмним забезпеченням користувачів | Середня | Середня | 4 | Розробка проекту з урахуванням вимог до програмного забезпечення користувачів проекту. | Прийняття | |

Продовження табл. Б.5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RS_15 | Відкритий | Зміна вимог замовника в процесі розробки додатку | Низька | Середній | 3 | Узгодити всі питання на початкових етапах, щоб мінімізувати кількість змін під час розробки | Пом’якшення | Переоцінка проекту, кожного разу, коли вимоги змінюються |

# ДОДАТОК В

## Лістинг основних модулів

Лістинг коду MainMenu.js

```
import SvgCanvas from "../svgcanvas/svgcanvas.js";
import { convertUnit, isValidUnit } from '../common/units.js';
import { isChrome } from '../common/browser.js';

const { $id } = SvgCanvas;
const homePage = '';

/**
 *
 */
class MainMenu {
  /**
   * @param {PlainObject} editor svgedit handler
   */
  constructor(editor) {
    this.editor = editor;
    /**
     * @type {Integer}
     */
    this.exportWindowCt = 0;
  }

  /**
   * @fires module:svgcanvas.SvgCanvas#event:ext_onNewDocument
   * @returns {void}
   */
```

```
  async clickClear() {
    const [ x, y ] = this.editor.configObj.curConfig.dimensions;
    const ok = await
seConfirm(this.editor.i18next.t('notification.QwantToClear'));
    if (ok === "Cancel") {
      return;
    }
    this.editor.leftPanel.clickSelect();
    this.editor.svgCanvas.clear();
    this.editor.svgCanvas.setResolution(x, y);
    this.editor.updateCanvas(true);
    this.editor.zoomImage();
    this.editor.layersPanel.populateLayers();
    this.editor.topPanel.updateContextPanel();
    this.editor.svgCanvas.runExtensions("onNewDocument");
  }
  /**
   *
   * @returns {void}
   */
  hideDocProperties() {
    const $imgDialog = $id("se-img-prop");
    $imgDialog.setAttribute("dialog", "close");
    $imgDialog.setAttribute("save",
this.editor.configObj.pref("img_save"));
    this.editor.docprops = false;
  }

  /**
   *
   * @returns {void}
   */
  hidePreferences() {
```

```
    const $editDialog = $id("se-edit-prefs");
    $editDialog.setAttribute("dialog", "close");
    this.editor.configObj.preferences = false;
  }

  /**
   * @param {Event} e
   * @returns {boolean} Whether there were problems saving the
document properties
   */
  saveDocProperties(e) {
    // set title
    const { title, w, h, save } = e.detail;
    // set document title
    this.editor.svgCanvas.setDocumentTitle(title);

    if (w !== "fit" && !isValidUnit("width", w)) {

seAlert(this.editor.i18next.t('notification.invalidAttrValGiven'
));
      return false;
    }
    if (h !== "fit" && !isValidUnit("height", h)) {

seAlert(this.editor.i18next.t('notification.invalidAttrValGiven'
));
      return false;
    }
    if (!this.editor.svgCanvas.setResolution(w, h)) {

seAlert(this.editor.i18next.t('notification.noContentToFitTo'));
      return false;
    }
    // Set image save option
```

```
  this.editor.configObj.pref("img_save", save);
  this.editor.updateCanvas();
  this.hideDocProperties();
  return true;
}
/**
 * Save user preferences based on current values in the UI.
 * @param {Event} e
 * @function module:SVGthis.savePreferences
 * @returns {Promise<void>}
 */
async savePreferences(e) {
  const {
    lang,
    bgcolor,
    bgurl,
    gridsnappingon,
    gridsnappingstep,
    gridcolor,
    showrulers,
    baseunit
  } = e.detail;
  // Set background
  this.editor.setBackground(bgcolor, bgurl);

  // set language
  if (lang && lang !== this.editor.configObj.pref("lang")) {
    this.editor.configObj.pref("lang", lang);
    seAlert('Changing the language needs reload');
  }

  // set grid setting
```

```
    this.editor.configObj.curConfig.gridSnapping =
gridsnappingon;
    this.editor.configObj.curConfig.snappingStep =
gridsnappingstep;
    this.editor.configObj.curConfig.gridColor = gridcolor;
    this.editor.configObj.curConfig.showRulers = showrulers;
    if (this.editor.configObj.curConfig.showRulers) {
      this.editor.rulers.updateRulers();
    }
    this.editor.configObj.curConfig.baseUnit = baseunit;

this.editor.svgCanvas.setConfig(this.editor.configObj.curConfig)
;
    this.editor.updateCanvas();
    this.hidePreferences();
  }
  /**
   *
   * @param e
   * @returns {Promise<void>} Resolves to `undefined`
   */
  async clickExport(e) {
    if (e?.detail?.trigger !== "ok" || e?.detail?.imgType ===
undefined) {
      return;
    }
    const imgType = e?.detail?.imgType;
    const quality = e?.detail?.quality ? e?.detail?.quality /
100 : 1;
    // Open placeholder window (prevents popup)
    let exportWindowName;

    /**
     *
```

```
     * @returns {void}
     */
    const openExportWindow = () => {
      const loadingImage  =
this.editor.i18next.t('notification.loadingImage');
      if (this.editor.configObj.curConfig.exportWindowType ===
"new") {
        this.editor.exportWindowCt++;
      }
      this.editor.exportWindowName =
        this.editor.configObj.curConfig.canvasName +
this.editor.exportWindowCt;
      let popHTML; let popURL;
      if (this.editor.loadingURL) {
        popURL = this.editor.loadingURL;
      } else {
        popHTML = `<!DOCTYPE html><html>
          <head>
            <meta charset="utf-8">
            <title>${loadingImage}</title>
          </head>
          <body><h1>${loadingImage}</h1></body>
        <html>`;
        if (typeof URL !== "undefined" && URL.createObjectURL) {
          const blob = new Blob([ popHTML ], { type: "text/html"
});
          popURL = URL.createObjectURL(blob);
        } else {
          popURL = "data:text/html;base64;charset=utf-8," +
popHTML;
        }
        this.editor.loadingURL = popURL;
      }
      this.editor.exportWindow = window.open(
```

```
        popURL,
        this.editor.exportWindowName
      );
    };
    const chrome = isChrome();
    if (imgType === "PDF") {
      if (!this.editor.customExportPDF && !chrome) {
        openExportWindow();
      }
      this.editor.svgCanvas.exportPDF(exportWindowName);
    } else {
      if (!this.editor.customExportImage) {
        openExportWindow();
      }
      /* const results = */ await
this.editor.svgCanvas.rasterExport(
        imgType,
        quality,
        this.editor.exportWindowName
      );
    }
  }

  /**
   *
   * @returns {void}
   */
  // eslint-disable-next-line class-methods-use-this
  clickImport() {
    /* empty fn */
  }
```

```
  /**
   *
   * @returns {void}
   */
  showDocProperties() {
    if (this.editor.docprops) {
      return;
    }
    this.editor.docprops = true;
    const $imgDialog = $id("se-img-prop");

    // update resolution option with actual resolution
    const resolution = this.editor.svgCanvas.getResolution();
    if (this.editor.configObj.curConfig.baseUnit !== "px") {
      resolution.w =
        convertUnit(resolution.w) +
this.editor.configObj.curConfig.baseUnit;
      resolution.h =
        convertUnit(resolution.h) +
this.editor.configObj.curConfig.baseUnit;
    }
    $imgDialog.setAttribute("save",
this.editor.configObj.pref("img_save"));
    $imgDialog.setAttribute("width", resolution.w);
    $imgDialog.setAttribute("height", resolution.h);
    $imgDialog.setAttribute("title",
this.editor.svgCanvas.getDocumentTitle());
    $imgDialog.setAttribute("dialog", "open");
  }

  /**
   *
   * @returns {void}
```

```
 */
showPreferences() {
  if (this.editor.configObj.preferences) {
    return;
  }
  this.editor.configObj.preferences = true;
  const $editDialog = $id("se-edit-prefs");
  // Update background color with current one
  const canvasBg = this.editor.configObj.curPrefs.bkgd_color;
  const url = this.editor.configObj.pref("bkgd_url");
  if (url) {
    $editDialog.setAttribute("bgurl", url);
  }
  $editDialog.setAttribute(
    "gridsnappingon",
    this.editor.configObj.curConfig.gridSnapping
  );
  $editDialog.setAttribute(
    "gridsnappingstep",
    this.editor.configObj.curConfig.snappingStep
  );
  $editDialog.setAttribute(
    "gridcolor",
    this.editor.configObj.curConfig.gridColor
  );
  $editDialog.setAttribute("canvasbg", canvasBg);
  $editDialog.setAttribute("dialog", "open");
}

/**
 *
 * @returns {void}
```

```
   */
  // eslint-disable-next-line class-methods-use-this
  openHomePage() {
    window.open(homePage, "_blank");
  }

  /**
   * @type {module}
   */
  init() {
    // add Top panel
    const template = document.createElement("template");
    // eslint-disable-next-line no-unsanitized/property
    template.innerHTML = `
    <se-menu id="main_button" label="SVG-Edit" src="logo.svg"
alt="logo">
        <se-menu-item id="tool_import" label="tools.import_doc"
src="importImg.svg"></se-menu-item>
        <se-menu-item id="tool_export" label="tools.export_img"
src="export.svg"></se-menu-item>
        <se-menu-item id="tool_docprops" label="tools.docprops"
shortcut="D" src="docprop.svg"></se-menu-item>
        <se-menu-item id="tool_editor_prefs"
label="config.editor_prefs" src="editPref.svg"></se-menu-item>
        <se-menu-item id="tool_editor_homepage"
label="tools.editor_homepage" src="logo.svg"></se-menu-item>
    </se-menu>`;

this.editor.$svgEditor.append(template.content.cloneNode(true));

    // register action to main menu entries
    /**
     * Associate all button actions as well as non-button
keyboard shortcuts.
     */
```

```
$id("tool_import").addEventListener("click", () => {
  this.clickImport();
  window.dispatchEvent(new CustomEvent("importImages"));
});
$id("tool_export").addEventListener("click", function() {
  document
    .getElementById("se-export-dialog")
    .setAttribute("dialog", "open");
});
$id("se-export-dialog").addEventListener(
  "change",
  this.clickExport.bind(this)
);
$id("tool_docprops").addEventListener(
  "click",
  this.showDocProperties.bind(this)
);
$id("tool_editor_prefs").addEventListener(
  "click",
  this.showPreferences.bind(this)
);

$id("se-img-prop").addEventListener(
  "change",
  function(e) {
    if (e.detail.dialog === "closed") {
      this.hideDocProperties();
    } else {
      this.saveDocProperties(e);
    }
  }.bind(this)
);
```

```
    $id("se-edit-prefs").addEventListener(
      "change",
      function(e) {
        if (e.detail.dialog === "closed") {
          this.hidePreferences();
        } else {
          this.savePreferences(e);
        }
      }.bind(this)
    );
  }
}

export default MainMenu;
```

Лістинг коду TopPanel.js

```
import SvgCanvas from "../../svgcanvas/svgcanvas.js";
import { isValidUnit, getTypeMap, convertUnit } from
"../../common/units.js";

const { $qa, $id } = SvgCanvas;

/*
 * register actions for left panel
 */
/**
 *
 */
class TopPanel {
  /**
   * @param {PlainObject} editor svgedit handler
   */
  constructor(editor) {
    this.editor = editor;
```

```
  }
  /**
   * @type {module}
   */
  displayTool(className) {
    // default display is 'none' so removing the property will
make the panel visible
    $qa(`.${className}`).map( (el) =>
el.style.removeProperty('display'));
  }
  /**
   * @type {module}
   */
  hideTool(className) {
    $qa(`.${className}`).map( (el) => el.style.display =
'none');
  }
  /**
   * @type {module}
   */
  get selectedElement() {
    return this.editor.selectedElement;
  }
  /**
   * @type {module}
   */
  get multiselected() {
    return this.editor.multiselected;
  }
  /**
   * @type {module}
   */
  get path() {
    return this.editor.svgCanvas.pathActions;
  }
```

```
  /**
   *
   * @param {Element} opt
   * @param {boolean} changeElem
   * @returns {void}
   */
  setStrokeOpt(opt, changeElem) {
    const { id } = opt;
    const bits = id.split('_');
    const [ pre, val ] = bits;

    if (changeElem) {
      this.svgCanvas.setStrokeAttr('stroke-' + pre, val);
    }
    opt.classList.add('current');
    const elements =
Array.prototype.filter.call(opt.parentNode.children, function
(child) {
      return child !== opt;
    });
    Array.from(elements).forEach(function (element) {
      element.classList.remove('current');
    });
  }

  /**
   * Updates the toolbar (colors, opacity, etc) based on the
selected element.
   * This function also updates the opacity and id elements that
are in the
   * context panel.
   * @returns {void}
   */
  update() {
    let i; let len;
    if (this.selectedElement) {
```

```
      switch (this.selectedElement.tagName) {
      case "use":
      case "image":
      case "foreignObject":
        break;
      case "g":
      case "a": {
        // Look for common styles
        const childs =
this.selectedElement.getElementsByTagName("*");
        let gWidth = null;
        for (i = 0, len = childs.length; i < len; i++) {
          const swidth = childs[i].getAttribute("stroke-width");

          if (i === 0) {
            gWidth = swidth;
          } else if (gWidth !== swidth) {
            gWidth = null;
          }
        }

        $id("stroke_width").value = (gWidth === null ? "" :
gWidth);
        this.editor.bottomPanel.updateColorpickers(true);
        break;
      }
      default: {
        this.editor.bottomPanel.updateColorpickers(true);

        $id("stroke_width").value =
this.selectedElement.getAttribute("stroke-width") || 1;
        $id("stroke_style").value =
this.selectedElement.getAttribute("stroke-dasharray") || "none";
        $id("stroke_style").setAttribute("value",
$id("stroke_style").value);
```

```
        let attr =
            this.selectedElement.getAttribute("stroke-linejoin")
|| "miter";

        if ($id("linejoin_" + attr)) {
          this.setStrokeOpt($id("linejoin_" + attr));
          $id("stroke_linejoin").setAttribute("value", attr);
        }

        attr = this.selectedElement.getAttribute("stroke-
linecap") || "butt";
        if ($id("linecap_" + attr)) {
          this.setStrokeOpt($id("linecap_" + attr));
          $id("stroke_linecap").setAttribute("value", attr);
        }
      }
      }
    }

    // All elements including image and group have opacity
    if (this.selectedElement) {
      const opacPerc =
        (this.selectedElement.getAttribute("opacity") || 1.0) *
100;
      $id("opacity").value = opacPerc;
      $id("elem_id").value = this.selectedElement.id;
      $id("elem_class").value =
this.selectedElement.getAttribute("class") ?? "";
    }

    this.editor.bottomPanel.updateToolButtonState();
  }
  /**
   * @param {PlainObject} [opts={}]
   * @param {boolean} [opts.cancelDeletes=false]
   * @returns {void} Resolves to `undefined`
```

```
   */
  promptImgURL({ cancelDeletes = false } = {}) {
    let curhref =
this.editor.svgCanvas.getHref(this.editor.selectedElement);
    curhref = curhref.startsWith("data:") ? "" : curhref;
    // eslint-disable-next-line no-alert
    const url = prompt(
      this.editor.i18next.t('notification.enterNewImgURL'),
      curhref
    );
    if (url) {
      this.setImageURL(url);
    } else if (cancelDeletes) {
      this.editor.svgCanvas.deleteSelectedElements();
    }
  }
  /**
   * Updates the context panel tools based on the selected
element.
   * @returns {void}
   */
  updateContextPanel() {
    let elem = this.editor.selectedElement;
    // If element has just been deleted, consider it null
    if (!elem?.parentNode) {
      elem = null;
    }
    const currentLayerName = this.editor.svgCanvas
      .getCurrentDrawing()
      .getCurrentLayerName();
    const currentMode = this.editor.svgCanvas.getMode();
    const unit =
      this.editor.configObj.curConfig.baseUnit !== "px"
        ? this.editor.configObj.curConfig.baseUnit
        : null;
```

```
    const isNode = currentMode === "pathedit"; // elem ?
(elem.id && elem.id.startsWith('pathpointgrip')) : false;
    const menuItems = document.getElementById("se-
cmenu_canvas");
    this.hideTool("selected_panel");
    this.hideTool("multiselected_panel");
    this.hideTool("g_panel");
    this.hideTool("rect_panel");
    this.hideTool("circle_panel");
    this.hideTool("ellipse_panel");
    this.hideTool("line_panel");
    this.hideTool("text_panel");
    this.hideTool("image_panel");
    this.hideTool("container_panel");
    this.hideTool("use_panel");
    this.hideTool("a_panel");
    this.hideTool("xy_panel");
    if (elem) {
      const elname = elem.nodeName;

      const angle =
this.editor.svgCanvas.getRotationAngle(elem);
      $id("angle").value = angle;

      const blurval = this.editor.svgCanvas.getBlur(elem) * 10;
      $id("blur").value = blurval;

      if (
        this.editor.svgCanvas.addedNew &&
        elname === "image" &&
        this.editor.svgCanvas.getMode() === "image" &&
        !this.editor.svgCanvas.getHref(elem).startsWith("data:")
      ) {
        /* await */ this.promptImgURL({ cancelDeletes: true });
      }
```

```
      if (!isNode && currentMode !== "pathedit") {
        this.displayTool("selected_panel");
        // Elements in this array already have coord fields
        if ([ "line", "circle", "ellipse" ].includes(elname)) {
          this.hideTool("xy_panel");
        } else {
          let x; let y;

          // Get BBox vals for g, polyline and path
          if ([ "g", "polyline", "path" ].includes(elname)) {
            const bb = this.editor.svgCanvas.getStrokedBBox([
elem ]);
            if (bb) {
              ({ x, y } = bb);
            }
          } else {
            x = elem.getAttribute("x");
            y = elem.getAttribute("y");
          }

          if (unit) {
            x = convertUnit(x);
            y = convertUnit(y);
          }

          $id("selected_x").value = (x || 0);
          $id("selected_y").value = (y || 0);
          this.displayTool("xy_panel");
        }

        // Elements in this array cannot be converted to a path
        if ([
          "image",
          "text",
          "path",
          "g",
```

```
          "use"
        ].includes(elname)) {
          this.hideTool("tool_topath");
        } else {
          this.displayTool("tool_topath");
        }
        if (elname === "path") {
          this.displayTool("tool_reorient");
        } else {
          this.hideTool("tool_reorient");
        }
        $id("tool_reorient").disabled = (angle === 0);
      } else {
        const point = this.path.getNodePoint();
        $id("tool_add_subpath").pressed = false;
        // eslint-disable-next-line max-len
        (!this.path.canDeleteNodes) ?
$id("tool_node_delete").classList.add("disabled") :
$id("tool_node_delete").classList.remove("disabled");

        // Show open/close button based on selected point
        // setIcon('#tool_openclose_path', path.closed_subpath ?
'open_path' : 'close_path');

        if (point) {
          const segType = $id("seg_type");
          if (unit) {
            point.x = convertUnit(point.x);
            point.y = convertUnit(point.y);
          }
          $id("path_node_x").value = (point.x);
          $id("path_node_y").value = (point.y);
          if (point.type) {
            segType.value = (point.type);
            segType.removeAttribute("disabled");
          } else {
```

```
            segType.value = 4;
            segType.setAttribute("disabled", "disabled");
          }
        }
        return;
      }

      // update contextual tools here
      const panels = {
        g: [],
        a: [],
        rect: [ "rx", "width", "height" ],
        image: [ "width", "height" ],
        circle: [ "cx", "cy", "r" ],
        ellipse: [ "cx", "cy", "rx", "ry" ],
        line: [ "x1", "y1", "x2", "y2" ],
        text: [],
        use: []
      };

      const { tagName } = elem;

      let linkHref = null;
      if (tagName === "a") {
        linkHref = this.editor.svgCanvas.getHref(elem);
        this.displayTool("g_panel");
      }
      // siblings
      if (elem.parentNode) {
        const selements =
Array.prototype.filter.call(elem.parentNode.children, function
(child) {
          return child !== elem;
        });
        if (elem.parentNode.tagName === "a" &&
!selements.length) {
```

```
          this.displayTool("a_panel");
          linkHref =
this.editor.svgCanvas.getHref(elem.parentNode);
        }
      }

      // Hide/show the make_link buttons
      if (linkHref) {
        this.displayTool('tool_make_link');
        this.displayTool('tool_make_link_multi');
        $id("link_url").value = linkHref;
      } else {
        this.hideTool('tool_make_link');
        this.hideTool('tool_make_link_multi');
      }

      if (panels[tagName]) {
        const curPanel = panels[tagName];
        this.displayTool(tagName + "_panel");

        curPanel.forEach((item) => {
          let attrVal = elem.getAttribute(item);
          if (this.editor.configObj.curConfig.baseUnit !== "px"
&& elem[item]) {
            const bv = elem[item].baseVal.value;
            attrVal = convertUnit(bv);
          }
          $id(`${tagName}_${item}`).value = attrVal || 0;
        });

        if (tagName === "text") {
          this.displayTool("text_panel");
          $id("tool_italic").pressed =
this.editor.svgCanvas.getItalic();
          $id("tool_bold").pressed =
this.editor.svgCanvas.getBold();
```

```
          $id("tool_font_family").setAttribute("value",
elem.getAttribute("font-family"));
          $id("font_size").value = elem.getAttribute("font-
size");
          $id("text").value = elem.textContent;
          const textAnchorStart = $id("tool_text_anchor_start");
          const textAnchorMiddle =
$id("tool_text_anchor_middle");
          const textAnchorEnd = $id("tool_text_anchor_end");
          switch (elem.getAttribute("text-anchor")) {
          case "start":
            textAnchorStart.pressed = true;
            textAnchorMiddle.pressed = false;
            textAnchorEnd.pressed = false;
            break;
          case "middle":
            textAnchorStart.pressed = false;
            textAnchorMiddle.pressed = true;
            textAnchorEnd.pressed = false;
            break;
          case "end":
            textAnchorStart.pressed = false;
            textAnchorMiddle.pressed = false;
            textAnchorEnd.pressed = true;
            break;
          }
          if (this.editor.svgCanvas.addedNew) {
            // Timeout needed for IE9
            setTimeout(() => {
              $id("text").focus();
              $id("text").select();
            }, 100);
          }
          // text
        } else if (
          tagName === "image" &&
```

```
          this.editor.svgCanvas.getMode() === "image"
        ) {
          this.editor.svgCanvas.setImageURL(
            this.editor.svgCanvas.getHref(elem)
          );
          // image
        } else if (tagName === "g" || tagName === "use") {
          this.displayTool("container_panel");
          const title = this.editor.svgCanvas.getTitle();
          const label = $id("g_title");
          label.value = title;
          $id("g_title").disabled = (tagName === "use");
        }
      }
      menuItems.setAttribute(
        (tagName === "g" ? "en" : "dis") + "ablemenuitems",
        "#ungroup"
      );
      menuItems.setAttribute(
        (tagName === "g" || !this.multiselected ? "dis" : "en")
+
        "ablemenuitems",
        "#group"
      );

      // if (!isNullish(elem))
    } else if (this.multiselected) {
      this.displayTool("multiselected_panel");
      menuItems.setAttribute("enablemenuitems", "#group");
      menuItems.setAttribute("disablemenuitems", "#ungroup");
    } else {
      menuItems.setAttribute(
        "disablemenuitems",

"#delete,#cut,#copy,#group,#ungroup,#move_front,#move_up,#move_d
own,#move_back"
```

```
      );
    }

    // update history buttons
    $id("tool_undo").disabled =
      this.editor.svgCanvas.undoMgr.getUndoStackSize() === 0;
    $id("tool_redo").disabled =
      this.editor.svgCanvas.undoMgr.getRedoStackSize() === 0;

    this.editor.svgCanvas.addedNew = false;

    if ((elem && !isNode) || this.multiselected) {
      // update the selected elements' layer
      $id("selLayerNames").removeAttribute("disabled");
      $id("selLayerNames").value = currentLayerName;
      $id("selLayerNames").setAttribute("value",
currentLayerName);

      // Enable regular menu options
      const canCMenu = document.getElementById("se-
cmenu_canvas");
      canCMenu.setAttribute(
        "enablemenuitems",

"#delete,#cut,#copy,#move_front,#move_up,#move_down,#move_back"
      );
    } else {
      $id("selLayerNames").setAttribute("disabled", "disabled");
    }
  }
  /**
   * @param {Event} [e] Not used.
   * @param {boolean} forSaving
   * @returns {void}
   */
  showSourceEditor(e, forSaving) {
```

```
    const $editorDialog = document.getElementById("se-svg-
editor-dialog");
    if ($editorDialog.getAttribute("dialog") === "open") return;
    const origSource = this.editor.svgCanvas.getSvgString();
    $editorDialog.setAttribute("dialog", "open");
    $editorDialog.setAttribute("value", origSource);
    $editorDialog.setAttribute("copysec", Boolean(forSaving));
    $editorDialog.setAttribute("applysec", !forSaving);
  }
  /**
   *
   * @returns {void}
   */
  clickWireframe() {
    $id("tool_wireframe").pressed =
!$id("tool_wireframe").pressed;
    this.editor.workarea.classList.toggle("wireframe");

    const wfRules = $id("wireframe_rules");
    if (!wfRules) {
      const fcRules = document.createElement('style');
      fcRules.setAttribute('id', 'wireframe_rules');

document.getElementsByTagName("head")[0].appendChild(fcRules);
    } else {
      while (wfRules.firstChild)
        wfRules.removeChild(wfRules.firstChild);
    }
    this.editor.updateWireFrame();
  }
  /**
   *
   * @returns {void}
   */
  clickUndo() {
    const { undoMgr, textActions } = this.editor.svgCanvas;
```

```
  if (undoMgr.getUndoStackSize() > 0) {
    undoMgr.undo();
    this.editor.layersPanel.populateLayers();
    if (this.editor.svgCanvas.getMode() === 'textedit') {
      textActions.clear();
    }
  }
}

/**
 *
 * @returns {void}
 */
clickRedo() {
  const { undoMgr } = this.editor.svgCanvas;
  if (undoMgr.getRedoStackSize() > 0) {
    undoMgr.redo();
    this.editor.layersPanel.populateLayers();
  }
}
/**
 * @type {module}
 */
changeRectRadius(e) {
  this.editor.svgCanvas.setRectRadius(e.target.value);
}

/**
 * @type {module}
 */
changeFontSize(e) {
  this.editor.svgCanvas.setFontSize(e.target.value);
}

/**
 * @type {module}
```

```
   */
  changeRotationAngle(e) {
    this.editor.svgCanvas.setRotationAngle(e.target.value);
    // eslint-disable-next-line max-len
    (Number.parseInt(e.target.value) === 0) ?
$id("tool_reorient").classList.add("disabled") :
$id("tool_reorient").classList.remove("disabled");
  }

  /**
   * @param {PlainObject} e
   * @returns {void}
   */
  changeBlur(e) {
    this.editor.svgCanvas.setBlur(e.target.value / 10, true);
  }
  /**
   *
   * @returns {void}
   */
  clickGroup() {
    // group
    if (this.editor.multiselected) {
      this.editor.svgCanvas.groupSelectedElements();
      // ungroup
    } else if (this.editor.selectedElement) {
      this.editor.svgCanvas.ungroupSelectedElement();
    }
  }

  /**
   *
   * @returns {void}
   */
  clickClone() {
    this.editor.svgCanvas.cloneSelectedElements(20, 20);
```

```
  }

  /**
   * @param {PlainObject} evt
   * @returns {void}
   */
  clickAlignEle(evt) {

this.editor.svgCanvas.alignSelectedElements(evt.detail.value,
"page");
  }

  /**
   * @param {string} pos indicate the alignment relative to top,
bottom, middle etc..
   * @returns {void}
   */
  clickAlign(pos) {
    let value = $id("tool_align_relative").value;
    if (value === "") {
      value = "selected";
    }
    this.editor.svgCanvas.alignSelectedElements(pos, value);
  }
  /**
   *
   * @type {module}
   */
  attrChanger(e) {
    const attr = e.target.getAttribute("data-attr");
    let val = e.target.value;
    const valid = isValidUnit(attr, val, this.selectedElement);

    if (!valid) {
      e.target.value = this.selectedElement.getAttribute(attr);
      // eslint-disable-next-line no-alert
```

```
alert(this.editor.i18next.t('notification.invalidAttrValGiven'))
;
      return false;
    }

    if (attr !== "id" && attr !== "class") {
      if (isNaN(val)) {
        val = this.editor.svgCanvas.convertToNum(attr, val);
      } else if (this.editor.configObj.curConfig.baseUnit !==
"px") {
        // Convert unitless value to one with given unit

        const unitData = getTypeMap();

        if (
          this.editor.selectedElement[attr] ||
          this.editor.svgCanvas.getMode() === "pathedit" ||
          attr === "x" ||
          attr === "y"
        ) {
          val *=
unitData[this.editor.configObj.curConfig.baseUnit];
        }
      }
    }

    // if the user is changing the id, then de-select the
element first
    // change the ID, then re-select it with the new ID
    if (attr === "id") {
      const elem = this.editor.selectedElement;
      this.editor.svgCanvas.clearSelection();
      elem.id = val;
      this.editor.svgCanvas.addToSelection([ elem ], true);
    } else {
```

```
    this.editor.svgCanvas.changeSelectedAttribute(attr, val);
  }
  return true;
}
/**
 *
 * @returns {void}
 */
convertToPath() {
  if (this.editor.selectedElement) {
    this.editor.svgCanvas.convertToPath();
  }
}
/**
 *
 * @returns {void}
 */
reorientPath() {
  if (this.editor.selectedElement) {
    this.path.reorient();
  }
}
/**
 *
 * @returns {void} Resolves to `undefined`
 */
makeHyperlink() {
  if (this.editor.selectedElement || this.multiselected) {
    // eslint-disable-next-line no-alert
    const url = prompt(
      this.editor.i18next.t('notification.enterNewLinkURL'),
      "http://"
    );
    if (url) {
      this.editor.svgCanvas.makeHyperlink(url);
    }
```

```
    }
  }
  /**
   *
   * @returns {void}
   */
  linkControlPoints() {
    $id("tool_node_link").pressed =
($id("tool_node_link").pressed) ? false : true;
    const linked = ($id("tool_node_link").pressed) ? true :
false;
    this.path.linkControlPoints(linked);
  }

  /**
   *
   * @returns {void}
   */
  clonePathNode() {
    if (this.path.getNodePoint()) {
      this.path.clonePathNode();
    }
  }

  /**
   *
   * @returns {void}
   */
  deletePathNode() {
    if (this.path.getNodePoint()) {
      this.path.deletePathNode();
    }
  }

  /**
   *
```

```
   * @returns {void}
   */
  addSubPath() {
    const button = $id("tool_add_subpath");
    const sp = !button.classList.contains("pressed");
    button.pressed = sp;
    // button.toggleClass('push_button_pressed tool_button');
    this.path.addSubPath(sp);
  }

  /**
   *
   * @returns {void}
   */
  opencloseSubPath() {
    this.path.opencloseSubPath();
  }
  /**
   * Delete is a contextual tool that only appears in the ribbon
if
   * an element has been selected.
   * @returns {void}
   */
  deleteSelected() {
    if (this.editor.selectedElement ||
this.editor.multiselected) {
      this.editor.svgCanvas.deleteSelectedElements();
    }
  }
  /**
   *
   * @returns {void}
   */
  moveToTopSelected() {
    if (this.editor.selectedElement) {
      this.editor.svgCanvas.moveToTopSelectedElement();
```

```
    }
  }

  /**
   *
   * @returns {void}
   */
  moveToBottomSelected() {
    if (this.editor.selectedElement) {
      this.editor.svgCanvas.moveToBottomSelectedElement();
    }
  }
  /**
   *
   * @returns {false}
   */
  clickBold() {

this.editor.svgCanvas.setBold(!this.editor.svgCanvas.getBold());
    this.updateContextPanel();
    return false;
  }

  /**
   *
   * @returns {false}
   */
  clickItalic() {

this.editor.svgCanvas.setItalic(!this.editor.svgCanvas.getItalic
());
    this.updateContextPanel();
    return false;
  }

  /**
```

```
   *
   * @param {string} value "start","end" or "middle"
   * @returns {false}
   */
  clickTextAnchor(value) {
    this.editor.svgCanvas.setTextAnchor(value);
    this.updateContextPanel();
    return false;
  }
  /**
  * Set a selected image's URL.
  * @function module:SVGthis.setImageURL
  * @param {string} url
  * @returns {void}
  */
  setImageURL(url) {
    const { editor } = this;
    if (!url) {
      url = editor.defaultImageURL;
    }
    editor.svgCanvas.setImageURL(url);
    $id("image_url").value = url;

    if (url.startsWith('data:')) {
      // data URI found
      this.hideTool("image_url");
    } else {
      // regular URL
      const promised = editor.svgCanvas.embedImage(url);
      // eslint-disable-next-line promise/catch-or-return
      promised
        // eslint-disable-next-line promise/always-return
        .then(() => {
          // switch into "select" mode if we've clicked on an
element
          editor.svgCanvas.setMode('select');
```

```
editor.svgCanvas.selectOnly(editor.svgCanvas.getSelectedElems(),
true);
        }, (error) => {
          console.error("error =", error);
          seAlert(editor.i18next.t('tools.no_embed'));
          editor.svgCanvas.deleteSelectedElements();
        });
      this.displayTool("image_url");
    }
  }
  /**
  * @param {boolean} editmode
  * @param {module:svgcanvas.SvgCanvas#event:selected} elems
  * @returns {void}
  */
  togglePathEditMode(editMode, elems) {
    if (editMode) {
      this.displayTool('path_node_panel');
    } else {
      this.hideTool('path_node_panel');
    }
    if (editMode) {
      // Change select icon
      $id('tool_path').pressed = false;
      $id('tool_select').pressed = true;
      $id('tool_select').setAttribute('src', `select_node.svg`);
      this.editor.multiselected = false;
      if (elems.length) {
        this.editor.selectedElement = elems[0];
      }
    } else {
      setTimeout(() => {
        $id('tool_select').setAttribute('src', `select.svg`);
      }, 1000);
    }
```

```
  }

  /**
   * @type {module}
   */
  init() {
    // add Top panel
    const template = document.createElement("template");
    const { i18next } = this.editor;
    // eslint-disable-next-line no-unsanitized/property
    template.innerHTML = `
      <div id="tools_top">
        <div id="editor_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_source" title="tools.tool_source"
shortcut="U" src="source.svg"></se-button>
          <se-button id="tool_wireframe"
title="tools.wireframe_mode" shortcut="F"
src="wireframe.svg"></se-button>
        </div> <!-- editor_panel -->
        <div id="history_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_undo" title="tools.undo"
shortcut="Z" src="undo.svg" disabled></se-button>
          <se-button id="tool_redo" title="tools.redo"
shortcut="Y" src="redo.svg" disabled></se-button>
        </div> <!-- history_panel -->
        <!-- Buttons when a single element is selected -->
        <div class="selected_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_clone" title="tools.clone"
shortcut="D" src="clone.svg"></se-button>
          <se-button id="tool_delete" title="tools.del"
shortcut="Delete/Backspace" src="delete.svg"></se-button>
        </div>
        <div class="selected_panel">
```

```
          <div class="tool_sep"></div>
          <se-button id="tool_move_top" title="tools.move_top"
shortcut="Ctrl+Shift+]" src="move_top.svg"></se-button>
          <se-button id="tool_move_bottom"
title="tools.move_bottom" shortcut="Ctrl+Shift+["
src="move_bottom.svg"></se-button>
        </div>
        <div class="selected_panel">
          <se-button id="tool_topath" title="tools.to_path"
src="to_path.svg"></se-button>
          <se-button id="tool_reorient"
title="tools.reorient_path" src="reorient.svg"></se-button>
          <se-button id="tool_make_link" title="tools.make_link"
src="globe_link.svg"></se-button>
        </div>
        <div class="selected_panel">
          <div class="tool_sep"></div>
          <se-input id="elem_id" data-attr="id" size="10"
label="properties.id_label" title="properties.id"></se-input>
        </div>
        <div class="selected_panel">
          <se-input id="elem_class" data-attr="class" size="10"
label="properties.class_label" title="properties.class"></se-
input>
          <se-spin-input size="3" id="angle" min=-180 max=180
step=5 src="angle.svg" title="properties.angle"></se-spin-input>
          <se-spin-input size="2" id="blur" min=0 max=100 step=5
src="blur.svg" title="properties.blur"></se-spin-input>
          <se-list id="tool_position"
title="tools.align_to_page" label="" width="22px" height="22px">
            <se-list-item id="tool_posleft" value="l"
title="tools.align_left" src="align_left.svg" img-
height="22px"></se-list-item>
            <se-list-item id="tool_poscenter" value="c"
title="tools.align_center" src="align_center.svg" img-
height="22px"></se-list-item>
```

```
            <se-list-item id="tool_posright" value="r"
title="tools.align_right" src="align_right.svg" img-
height="22px"></se-list-item>
            <se-list-item id="tool_postop" value="t"
title="tools.align_top" src="align_top.svg" img-
height="22px"></se-list-item>
            <se-list-item id="tool_posmiddle" value="m"
title="tools.align_middle" src="align_middle.svg" img-
height="22px"></se-list-item>
            <se-list-item id="tool_posbottom" value="b"
src="align_bottom.svg" title="tools.align_bottom" img-
height="22px"></se-list-item>
          </se-list>
        </div>
        <div class="xy_panel">
          <se-spin-input id="selected_x" data-attr="x" size="4"
type="text" label="properties.x_label" title="properties.pos_x">
          </se-spin-input>
          <se-spin-input id="selected_y" data-attr="y" size="4"
type="text" label="properties.y_label" title="properties.pos_y">
          </se-spin-input>
        </div>
        <!-- Buttons when multiple elements are selected -->
        <div class="multiselected_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_clone_multi" title="tools.clone"
shortcut="C" src="clone.svg"></se-button>
          <se-button id="tool_delete_multi" title="tools.del"
shortcut="Delete/Backspace" src="delete.svg"></se-button>
        </div>
        <div class="multiselected_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_group_elements"
title="tools.group_elements" shortcut="G"
src="group_elements.svg">
          </se-button>
```

```
          <se-button id="tool_make_link_multi"
title="tools.make_link" src="globe_link.svg"></se-button>
          <se-button id="tool_align_left"
title="tools.align_left" src="align_left.svg"></se-button>
          <se-button id="tool_align_center"
title="tools.align_center" src="align_center.svg"></se-button>
          <se-button id="tool_align_right"
title="tools.align_right" src="align_right.svg"></se-button>
          <se-button id="tool_align_top" title="tools.align_top"
src="align_top.svg"></se-button>
          <se-button id="tool_align_middle"
title="tools.align_middle" src="align_middle.svg"></se-button>
          <se-button id="tool_align_bottom"
title="tools.align_bottom" src="align_bottom.svg"></se-button>
          <se-select id="tool_align_relative"
label="tools.relativeTo"

options="tools.selected_objects,tools.largest_object,tools.small
est_object,tools.page"
            values="selected::largest::smallest::page"></se-
list-item>
          </se-select>
        </div> <!-- multiselected_panel -->
        <div class="rect_panel">
          <se-spin-input id="rect_width" data-attr="width"
size="4" label="properties.w_label"
title="properties.rect_width"></se-spin-input>
          <se-spin-input id="rect_height" data-attr="height"
size="4" label="properties.h_label"
title="properties.rect_height"></se-spin-input>
          <se-spin-input id="rect_rx" min=0 max=1000 step=1
size="3" title="properties.corner_radius" data-attr="Corner
Radius" src="c_radius.svg"></se-spin-input>
        </div> <!-- rect_panel -->
        <div class="image_panel">
```

```
          <se-spin-input id="image_width" data-attr="width"
size="4" type="text" label="properties.w_label"
title="properties.image_width"></se-spin-input>
          <se-spin-input id="image_height" data-attr="height"
size="4" type="text" label="properties.h_label"
title="properties.image_height"></se-spin-input>
        </div>
        <div class="image_panel">
          <se-input id="image_url" data-attr="image_url"
size="15" label="properties.image_url"></se-input>
        </div>
        <div class="circle_panel">
          <se-spin-input id="circle_cx" data-attr="cx" size="4"
label="properties.cx_label"></se-spin-input>
          <se-spin-input id="circle_cy" data-attr="cy" size="4"
label="properties.cy_label"></se-spin-input>
        </div>
        <div class="circle_panel">
          <se-spin-input id="circle_r" data-attr="r" size="4"
label="properties.r_label"></se-spin-input>
        </div>
        <div class="ellipse_panel">
          <se-spin-input id="ellipse_cx" data-attr="cx" size="4"
title="properties.ellipse_cx" label="properties.cx_label"></se-
spin-input>
          <se-spin-input id="ellipse_cy" data-attr="cy" size="4"
title="properties.ellipse_cy" label="properties.cy_label"></se-
spin-input>
        </div>
        <div class="ellipse_panel">
          <se-spin-input id="ellipse_rx" data-attr="rx" size="4"
title="properties.ellipse_rx" label="properties.rx_label"></se-
spin-input>
          <se-spin-input id="ellipse_ry" data-attr="ry" size="4"
title="properties.ellipse_ry" label="properties.ry_label"></se-
spin-input>
```

```
        </div>
        <div class="line_panel">
          <se-spin-input id="line_x1" data-attr="x1" size="4"
title="properties.line_x1" label="properties.x1_label"></se-
spin-input>
          <se-spin-input id="line_y1" data-attr="y1" size="4"
title="properties.line_y1" label="properties.y1_label"></se-
spin-input>
          <se-spin-input id="line_x2" data-attr="x2" size="4"
title="properties.line_x2" label="properties.x2_label"></se-
spin-input>
          <se-spin-input id="line_y2" data-attr="y2" size="4"
title="properties.line_y2" label="properties.y2_label"></se-
spin-input>
        </div>
        <div class="text_panel">
          <se-button id="tool_bold" title="properties.bold"
src="bold.svg" shortcut="B"></se-button>
          <se-button id="tool_italic" title="properties.italic"
src="italic.svg" shortcut="I"></se-button>
          <se-select id="tool_font_family"
label="properties.font_family_label"
options="properties.serif,properties.sans_serif,properties.cursi
ve,properties.fantasy,properties.monospace,properties.courier,pr
operties.helvetica,properties.times" values="Serif::Sans-
serif::Cursive::Fantasy::Monospace::Courier::Helvetica::Times"><
/select>
          <se-spin-input size="2" id="font_size" min=1 max=1000
step=1 title="properties.font_size" src="fontsize.svg"></se-
spin-input>
        </div>
        <div class="text_panel">
          <se-button id="tool_text_anchor_start"
title="properties.text_anchor_start"
src="anchor_start.svg"></se-button>
```

```
          <se-button id="tool_text_anchor_middle"
title="properties.text_anchor_middle"
src="anchor_middle.svg"></se-button>
          <se-button id="tool_text_anchor_end"
title="properties.text_anchor_end" src="anchor_end.svg"></se-
button>
        </div>
        <!-- Not visible, but still used -->
        <input id="text" type="text" size="35" />
        <div class="container_panel">
          <div class="tool_sep"></div>
          <se-input id="g_title" data-attr="title" size="8"
label="properties.label"></se-input>
        </div> <!-- container_panel -->
        <div class="use_panel">
          <se-button id="tool_unlink_use"
title="tools.tool_unlink_use" src="unlink_use.svg"></se-button>
        </div> <!-- use_panel -->
        <div class="g_panel">
          <se-button id="tool_ungroup" title="tools.ungroup"
src="ungroup.svg"></se-button>
        </div> <!-- g_panel -->
        <!-- For anchor elements -->
        <div class="a_panel">
          <label id="tool_link_url">
            <span id="linkLabel" class="icon_label"></span>
            <input id="link_url" type="text" size="35" />
          </label>
        </div> <!-- a_panel -->
        <div class="path_node_panel">
          <div class="tool_sep"></div>
          <se-button id="tool_node_link" title="tools.node_link"
src="tool_node_link.svg" pressed></se-button>
          <div class="tool_sep"></div>
```

```
          <se-spin-input id="path_node_x" data-attr="x" size="4"
title="properties.node_x" label="properties.x_label"></se-spin-
input>
          <se-spin-input id="path_node_y" data-attr="y" size="4"
title="properties.node_y" label="properties.y_label"></se-spin-
input>
          <se-select id="seg_type" title="properties.seg_type"
label=""
options="properties.straight_segments,properties.curve_segments"
values="4::6"></se-select>
          <se-button id="tool_node_clone"
title="tools.node_clone" src="tool_node_clone.svg"></se-button>
          <se-button id="tool_node_delete"
title="tools.node_delete" src="tool_node_delete.svg"></se-
button>
          <se-button id="tool_openclose_path"
title="tools.openclose_path" src="tool_openclose_path.svg"></se-
button>
          <se-button id="tool_add_subpath"
title="tools.add_subpath" src="tool_add_subpath.svg"></se-
button>
        </div> <!-- path_node_panel -->
        <div id="cur_context_panel"></div>
     </div>
       `;

this.editor.$svgEditor.append(template.content.cloneNode(true));
    // svg editor source dialoag added to DOM
    const newSeEditorDialog = document.createElement(
      "se-svg-source-editor-dialog"
    );
    newSeEditorDialog.setAttribute("id", "se-svg-editor-
dialog");
    this.editor.$container.append(newSeEditorDialog);
    newSeEditorDialog.init(i18next);
```

```
    $id("tool_link_url").setAttribute("title",
i18next.t('tools.set_link_url'));
    // register action to top panel buttons
    $id("tool_source").addEventListener("click",
this.showSourceEditor.bind(this));
    $id("tool_wireframe").addEventListener("click",
this.clickWireframe.bind(this));
    $id("tool_undo").addEventListener("click",
this.clickUndo.bind(this));
    $id("tool_redo").addEventListener("click",
this.clickRedo.bind(this));
    $id("tool_clone").addEventListener("click",
this.clickClone.bind(this));
    $id("tool_clone_multi").addEventListener("click",
this.clickClone.bind(this));
    $id("tool_delete").addEventListener("click",
this.deleteSelected.bind(this));
    $id("tool_delete_multi").addEventListener("click",
this.deleteSelected.bind(this));
    $id("tool_move_top").addEventListener("click",
this.moveToTopSelected.bind(this));
    $id("tool_move_bottom").addEventListener("click",
this.moveToBottomSelected.bind(this));
    $id("tool_topath").addEventListener("click",
this.convertToPath.bind(this));
    $id("tool_make_link").addEventListener("click",
this.makeHyperlink.bind(this));
    $id("tool_make_link_multi").addEventListener("click",
this.makeHyperlink.bind(this));
    $id("tool_reorient").addEventListener("click",
this.reorientPath.bind(this));
    $id("tool_group_elements").addEventListener("click",
this.clickGroup.bind(this));
    $id("tool_position").addEventListener("change", (evt) =>
this.clickAlignEle.bind(this)(evt));
```

```
    $id("tool_align_left").addEventListener("click", () =>
this.clickAlign.bind(this)("left"));
    $id("tool_align_right").addEventListener("click", () =>
this.clickAlign.bind(this)("right"));
    $id("tool_align_center").addEventListener("click", () =>
this.clickAlign.bind(this)("center"));
    $id("tool_align_top").addEventListener("click", () =>
this.clickAlign.bind(this)("top"));
    $id("tool_align_bottom").addEventListener("click", () =>
this.clickAlign.bind(this)("bottom"));
    $id("tool_align_middle").addEventListener("click", () =>
this.clickAlign.bind(this)("middle"));
    $id("tool_node_clone").addEventListener("click",
this.clonePathNode.bind(this));
    $id("tool_node_delete").addEventListener("click",
this.deletePathNode.bind(this));
    $id("tool_openclose_path").addEventListener("click",
this.opencloseSubPath.bind(this));
    $id("tool_add_subpath").addEventListener("click",
this.addSubPath.bind(this));
    $id("tool_node_link").addEventListener("click",
this.linkControlPoints.bind(this));
    $id("angle").addEventListener("change",
this.changeRotationAngle.bind(this));
    $id("blur").addEventListener("change",
this.changeBlur.bind(this));
    $id("rect_rx").addEventListener("change",
this.changeRectRadius.bind(this));
    $id("font_size").addEventListener("change",
this.changeFontSize.bind(this));
    $id("tool_ungroup").addEventListener("click",
this.clickGroup.bind(this));
    $id("tool_bold").addEventListener("click",
this.clickBold.bind(this));
    $id("tool_italic").addEventListener("click",
this.clickItalic.bind(this));
```

```
    $id("tool_text_anchor_start").addEventListener("click", ()
=> this.clickTextAnchor.bind(this)("start"));
    $id("tool_text_anchor_middle").addEventListener("click", ()
=> this.clickTextAnchor.bind(this)("middle"));
    $id("tool_text_anchor_end").addEventListener("click", () =>
this.clickTextAnchor.bind(this)("end"));
    $id("tool_unlink_use").addEventListener("click",
this.clickGroup.bind(this));
    $id('image_url').addEventListener('change', (evt) => {
this.setImageURL(evt.currentTarget.value);});

    // all top panel attributes
    [
      "elem_id",
      "elem_class",
      "circle_cx",
      "circle_cy",
      "circle_r",
      "ellipse_cx",
      "ellipse_cy",
      "ellipse_rx",
      "ellipse_ry",
      "selected_x",
      "selected_y",
      "rect_width",
      "rect_height",
      "line_x1",
      "line_x2",
      "line_y2",
      "image_width",
      "image_height",
      "path_node_x",
      "path_node_y"
    ].forEach((attrId) =>
      $id(attrId).addEventListener("change",
this.attrChanger.bind(this))
```

```
    );
  }
}

export default TopPanel;
```