

Київський національний торговельно-економічний університет

Кафедра цифрової економіки та системного аналізу

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**“Розробка оптимізованого графічного інтерфейсу для веб-системи
поліграфічного підприємства”**

Студентки 2 курсу, 1м групи,

спеціальності
051 “Економіка”,

спеціалізації
“Цифрова економіка”

Науковий керівник
кандидат економічних наук,
доцент

Гарант освітньої програми
доктор фізико-математичних
наук, професор

Новик Альони
Сергіївни

Кулаженко Володимир
Валерійович

Гамалій Володимир
Федорович

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра цифрової економіки та системного аналізу

Освітній рівень магістр

Спеціальність 051 “Економіка”

Спеціалізація “Цифрова економіка”

Затверджую

Зав. кафедри _____ Роскладка А.А.

“15” листопада 2021

Завдання

на випускню кваліфікаційну роботу (проект) студентці

Новик Альоні Сергіївни

(прізвище, ім'я, по-батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

“Розробка оптимізованого графічного інтерфейсу для веб-системи поліграфічного підприємства”

Затверджена наказом ректора від “22” жовтня 2020р. №3066

2. Строк здачі студентом закінченої роботи “05” листопада 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: створення функціонального інтерфейсу веб-системи для поліграфічного підприємства із зручним юзабіліті для автоматизації та пришвидшення роботи персоналу.

Об'єкт дослідження: поліграфічне підприємство.

Предмет дослідження: процеси, пов'язані з роботою менеджера з продажу на поліграфічному підприємстві.

4. Консультанти по роботі (проекту) із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Кулаженко В.В.	15.11.2020 р.	15.11.2020 р.
2	Кулаженко В.В.	15.11.2020 р.	15.11.2020 р.
3	Кулаженко В.В.	15.11.2020 р.	15.11.2020 р.

5. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ІНТЕРФЕЙСІВ ДЛЯ ВЕБ-СИСТЕМ

ПІДПРИЄМСТВ

1.1. Поняття та особливості інтерфейсу веб-системи

1.2. Види інтерфейсів веб-системи

1.3. Існуючі інтерфейси для веб-систем поліграфічного підприємства

Висновки до розділу 1

РОЗДІЛ 2. ПОБУДОВА БІЗНЕС-ПРОЦЕСІВ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА, ПОВ'ЯЗАНИХ З ПРОДАЖЕМ ТОВАРІВ ТА ПОСЛУГ

2.1. Виділення основних бізнес-процесів поліграфічного підприємства

2.2. Побудова моделі бізнес-процесів поліграфічного підприємства

2.3. Співставлення бізнес-процесів з функціоналом інтерфейсу веб-системи

Висновки до розділу 2

РОЗДІЛ 3. СТВОРЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ ДЛЯ ВЕБ-СИСТЕМИ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА

3.1. Верстка елементів інтерфейсу веб-системи

3.2. Стилізація інтерфейсу веб-системи

3.3. Анімація інтерфейсу веб-системи

3.4. Зв'язок з бекендом веб-системи

Висновки до розділу 3

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи (проекту)

№ з/п	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	20.10.2020	20.10.2020
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.11.2020	15.11.2020
3	<i>Вступ</i>	01.03.2021	
4	<i>Розділ 1. Аналіз існуючих інтерфейсів для веб-систем підприємств</i>	25.06.2021	
5	<i>Розділ 2. Побудова бізнес-процесів поліграфічного підприємства, пов'язаних з продажем товарів і послуг</i>	01.09.2021	
6	<i>Підготовка статті у збірник наукових статей магістрів</i>	15.09.2021	
7	<i>Розділ 3. Створення графічного інтерфейсу для веб-системи поліграфічного підприємства</i>	18.10.2021	
8	<i>Висновки</i>	01.11.2021	
9	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	05.11.2021	
10	<i>Попередній захист випускної кваліфікаційної роботи</i>	25.11.2021	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	28.11.2021	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	30.11.2021	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом ЕК	

7. Дата видачі завдання “15” листопада 2021р.

8. Науковий керівник випускної кваліфікаційної роботи (проекту)

Кулаженко В.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

Гамалій В.Ф.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

Новик А.С.

(прізвище, ініціали, підпис)

11. Відгук керівника випускної кваліфікаційної роботи (проекту)

Науковий керівник випускної кваліфікаційної роботи (проекту)

05.11.2021 р.

(підпис, дата)

Відмітка про попередній захист Кулаженко В.В.

25.11.2021 р.

(ПІБ наукового керівника, підпис, дата)

12. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студентки Новик А.С.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми

Гамалій В.Ф.

(підпис, прізвище, ініціали)

Завідувач кафедри

Роскладка А.А.

(підпис, прізвище, ініціали)

“ ” 2021р.

Анотація

У цій роботі досліджено поняття користувацького інтерфейсу, його особливості та види, досліджено гештальт-принципи розробки графічного інтерфейсу веб-системи, що базуються на психології сприйняття людського мозку. Обґрунтовано застосування гештальт-принципів для розробки графічного інтерфейсу. Досліджено інтерфейси додатків для управління поліграфічним підприємством. Досліджено організаційну структуру поліграфічного підприємства Starter та важливість побудови моделі бізнес-процесів підприємства.

У практичній частині роботи побудовано модель організаційної структури поліграфічного підприємства Starter, модель його бізнес-процесів, відмальовано макет інтерфейсу веб-системи за допомогою графічного редактору, а також реалізовано веб-систему за допомогою таких технологій, як React, Typescript, Redux, Webpack, SCSS.

Ключові слова: гештальт-принципи, моделювання бізнес-процесів, інтерфейс, веб-система, веб-додаток, макет, GUI.

Anotation

This research explores the concept of user interface, its features and types, and the Gestalt principles of developing a graphical interface of a web system based on the psychology of perception of the human brain. The research includes the application of Gestalt principles for the development of a graphical interface. The interfaces of applications for management of the polygraphic enterprise are investigated. The organizational structure of the enterprise Starter and the importance of building a model of business processes of the enterprise are studied.

In the practical part of work the model of organizational structure of the graphic enterprise Starter, model of its business processes is created, the model of the interface of web system by means of the graphic editor is drawn, and also the web system is realized by means of such technologies as React, Typescript, Redux, Webpack, SCSS.

Keywords: gestalt principles, business process modeling, interface, web system, web application, layout, GUI.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ІНТЕРФЕЙСІВ ДЛЯ ВЕБ-СИСТЕМ ПІДПРИЄМСТВ	6
1.1. Поняття та особливості інтерфейсу веб-системи	6
1.2. Види інтерфейсів веб-системи	13
1.3. Існуючі інтерфейси для веб-систем поліграфічного підприємства	18
<i>Висновки до розділу 1</i>	21
РОЗДІЛ 2. ПОБУДОВА БІЗНЕС-ПРОЦЕСІВ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА, ПОВ'ЯЗАНИХ З ПРОДАЖЕМ ТОВАРІВ ТА ПОСЛУГ	23
2.1. Виділення основних бізнес-процесів поліграфічного підприємства	23
2.2. Побудова моделі бізнес-процесів поліграфічного підприємства	31
2.3. Співставлення бізнес-процесів з функціоналом інтерфейсу веб-системи	34
<i>Висновки до розділу 2</i>	35
РОЗДІЛ 3. СТВОРЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ ДЛЯ ВЕБ-СИСТЕМИ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА	37
3.1. Верстка елементів інтерфейсу веб-системи	37
3.2. Стилзація інтерфейсу веб-системи	44
3.3. Анімація інтерфейсу веб-системи	48
3.4. Зв'язок з бекендом веб-системи	52
<i>Висновки до розділу 3</i>	55
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТКИ	63

ВСТУП

У сучасному світі стрімкого розвитку цифрових технологій вимоги до застосунків, веб-систем, веб-додатків та інших інтернет-ресурсів дуже високі. Ресурс має привабливо виглядати, бути зручним у використанні, мати відмінні показники завантаження сторінок. Користувач, відвідуючи додаток, має чітко розуміти, куди саме йому потрібно перейти, щоб отримати результати, за якими він прийшов на ресурс.

Така криза, як пандемія Covid19, показала, що компанії можуть працювати віддалено. Поліграфічне підприємство не є виключенням. Тому для ефективної роботи його працівників необхідно розробити веб-систему, що дозволить в реальному часі працювати співробітникам.

Для досягнення усього вище зазначеного необхідно побудувати чітку архітектуру як фронтенду, так і бекенду. Добре розроблена архітектура дозволяє не лише краще орієнтуватися користувачу на сторінках веб-системи, а й підтримувати веб-систему з меншими витратами часу та коштів.

Аналіз останніх досліджень та публікацій. На сьогоднішній день проводиться доволі багато наукових досліджень проблеми створення зручного та привабливого інтерфейсу веб-системи. Серед наукових праць вітчизняних науковців присвячених дослідженню та створенню інтерфейсів веб-систем можна виділити роботи Мосіюк О.О., Компанець М.О., Матвейко І. І., Купрунець Т.Я[1-4].

Серед зарубіжних вчених значний вплив на розвиток теоретичних і практичних знань у розробленні інтерфейсу веб-системи зробили Letizia Bollini, Liqiong Deng, Marshall Scott Poole, Melius Weideman[5-7].

Актуальність цього дослідження полягає в необхідності оптимізувати роботу персоналу поліграфічного підприємства STARTER, а також у потребі дослідити принципи створення зручного і зрозумілого інтерфейсу веб-системи.

Об'єктом дослідження є поліграфічне підприємство.

Предметом дослідження є процеси, пов'язані з роботою менеджера з продажу на поліграфічному підприємстві.

Метою роботи є створення функціонального інтерфейсу веб-системи для поліграфічного підприємства із зручним юзабіліті для автоматизації та пришвидшення роботи персоналу.

Для досягнення поставленої мети необхідно вирішити наступні **завдання:**

- Дослідити поняття, особливості та види інтерфейсів сучасних веб-систем;
- Проаналізувати існуючі інтерфейси веб-систем поліграфічного підприємства;
- Розглянути основні бізнес-процеси поліграфічного підприємства;
- Побудувати модель бізнес-процесів поліграфічного підприємства;
- Розробити функціонал інтерфейсу веб-системи поліграфічного підприємства на базі моделі його бізнес-процесів;
- Зверстати компоненти інтерфейсу веб-системи поліграфічного підприємства;
- Стилізувати компоненти інтерфейсу веб-системи поліграфічного підприємства;
- Додати анімацію до інтерфейсу веб-системи поліграфічного підприємства;
- Інтегрувати інтерфейс веб-системи з її бекендом.

Для виконання дослідження було використано наступні **методи:** моделювання, порівняння, аналіз і синтез, програмування.

Теоретична значущість випускної кваліфікаційної роботи полягає в систематизації існуючих видів користувацьких інтерфейсів, теоретичних принципів розробки графічного інтерфейсу, що стали основою для визначення практичних розробок.

Практичне значення отриманих в рамках виконання випускної кваліфікаційної роботи результатів полягає в розробці моделі бізнес-процесів

поліграфічного підприємства STARTER та розробці графічного інтерфейсу для веб-системи поліграфічного підприємства STARTER на основі розробленої моделі бізнес-процесів.

Апробація результатів досліджень. За результатами проведеного дослідження опубліковано наукову статтю на тему “The Role of Javascript Frameworks in Web System UI Development”, яка увійшла до збірника наукових статей студентів КНТЕУ “Цифрова економіка”, м. Київ 2021 рік.

Структура роботи. Робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи - 62 сторінок. Вона містить 6 таблиць, 43 рисунки та 6 додатків. Кількість використаних джерел - 30, їх список наведений на 4 сторінках.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ІНТЕРФЕЙСІВ ДЛЯ ВЕБ-СИСТЕМ ПІДПРИЄМСТВ

1.1. Поняття та особливості інтерфейсу веб-системи

Користувацький інтерфейс (User interface, UI) - це точки доступу, які користувач використовує для взаємодії з дизайном [8]. Розробка користувацького інтерфейсу передбачає створення дизайнером за допомогою програмного забезпечення зовнішнього вигляду веб-сайту чи додатку, приділяючи особливу увагу стилю та зручності використання.

Елементами користувацького інтерфейсу веб-додатку є кнопки, випадаючі меню, випадаючі списки, форми (зворотнього зв'язку, замовлення тощо), контекстне меню, сайдбар, сповіщення, модальні вікна, підказки (tooltip), слайдери, таби, акордеон, пагінація, прогрес бар, лодер, "хлібні крошки" тощо.

Щоб створити інтерфейс користувача у найкращому вигляді, слід врахувати те, що:

Користувачі швидко оцінюють дизайн і їх хвилює зручність використання та привабливість.

- користувачам важливо виконувати свої завдання легко і з мінімальними зусиллями; користувачі бажають заповнювати зрозумілу, розбиту на кроки чи блоки форму замовлення, а не величезний лист з полями, в якому важко віднайти логіку;
- дизайн має бути «непомітним»: користувачі повинні зосереджуватися не на ньому, а на виконанні завдань. Дизайн є важливим для деяких типів сайтів, наприклад, портфоліо веб-дизайнера;
- тому важливо розуміти контексти та потоки завдань ваших користувачів (які ви можете знайти, наприклад, на картах подорожей клієнтів), щоб точно налаштувати найкращі, найінтуїтивніші інтерфейси, які забезпечують безперебійний користувацький досвід.

Користувальницькі інтерфейси також повинні бути приємними (або принаймні задовольняти користувача і не викликати розчарувань).

- дизайн має бути максимально персоналізованим, тоді юзери будуть відвідувати саме цей сайт.
- де це доречно, елементи гейміфікації можуть зробити дизайн більш привабливим для юзерів.

Інтерфейси повинні передавати цінності бренду та зміцнювати довіру користувачів.

- хороший дизайн - це емоційний дизайн. Користувачі пов'язують хороші почуття з брендами, які взаємодіють з ними на всіх рівнях і зберігають приємний, бездоганний досвід у пам'яті.

Для побудови зручного користувацького інтерфейсу необхідно притримуватись певних правил. Їх існує досить велика кількість, але, на нашу думку, найбільш доречними є ті правила, що мають в основі психологію людини. Прикладом таких правил є **гештальт-принципи**(рис.1.1).



Рисунок 1.1. Складові гештальт-принципів у UI-дизайні

Джерело: [12]

Гештальт-принципи дозволяють зрозуміти, як працює людський мозок і відповідно спрощують створення UI.

Гештальт - це група принципів візуального сприйняття, розроблена в 1920-их німецькими психологами. Гештальт-принципи будуються на теорії про те, що “організоване ціле сприймається, як щось більше, ніж просто сума його частин”. Німецько-американський психолог, один з розробників гештальт-психології Курт Коффка сказав: “Ціле - це не просто сума його складових частин”. Далі розглянемо кожен з принципів зображених на Рисунку 1.1 більш детально.

Близькість (Proximity). Елементи, що розміщені близько один до одного, сприймаються більш зв’язаними, ніж ті, що розміщені на деякій відстані один від одного. Таким чином, ми часто сприймаємо окремі елементи групою, а не окремо. На рисунку 1.2 можна побачити візуалізацію цього принципу.

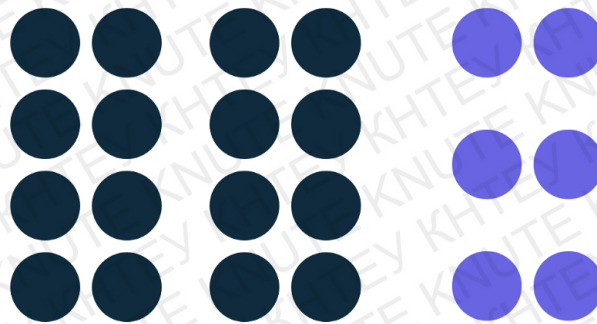


Рисунок 1.2. Візуалізація гештальт-принципу Близькість

Джерело: [12]

Цей принцип у UI інтерфейсі можна застосувати до навігації, карток, галереї, списків тощо. У Додатку А можна побачити приклад використання розглянутого вище гештальт-принципу. Принцип Близькості говорить про те, що близькі за змістом елементи повинні розміщуватись близько один до одного, а всі інші на відстані. Наприклад, пункти меню в навігації мають бути близько один до одного на відміну від логотипу, який повинен розташовуватись на більшій відстані. Важливу роль для утримання та

правильного перемикання уваги відіграє білий простір (місця без контенту). Саме правильна структура наповнення та розміщення вільного простору дозволить зберегти увагу відвідувача і ймовірність того, що користувач доскролить сторінку до кінця зросте.

Спільна область (Common Region). Цей принцип дуже схожий на принцип Близькості. Він описує те, що елементи розміщені в межах однієї області, сприймаються як група (рис.1.3).

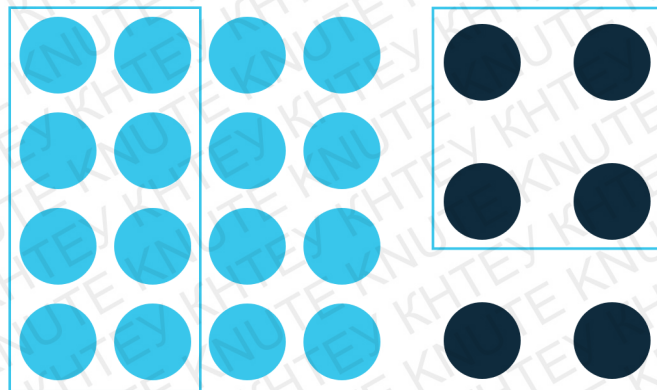


Рисунок 1.3. Візуалізація гештальт-принципу Спільна область

Джерело: [12]

У Додатку Б можна розглянути приклад використання принципу Спільної області при побудові UI. На зображенням окремі блоки розміщуються на деякій відстані один від одного. Таким чином, користувач розуміє, що у наступному блоці розміщено іншу інформацію. Наприклад, на односторінкових сайтах часто виділяються окремі блоки такі, як вітальний блок (інформація про товар, послугу тощо; цей блок має максимально зацікавити відвідувача), блок про компанію, галерея, кроки, які потрібно зробити для замовлення, контактна інформація. Для групування контенту по областях можна використовувати лінії, колір, форми й тіні. Використовуючи ці засоби, можна вивести значимий контент на передній план. Зазвичай кнопка “Замовити” найбільше притягує до себе увагу і “змушує” на себе натиснути.

Схожість (Similarity). Елементи зі схожими візуальними характеристиками мозок сприймає більш зв'язаними, ніж ті, що виглядають по-різному (рис. 1.4).

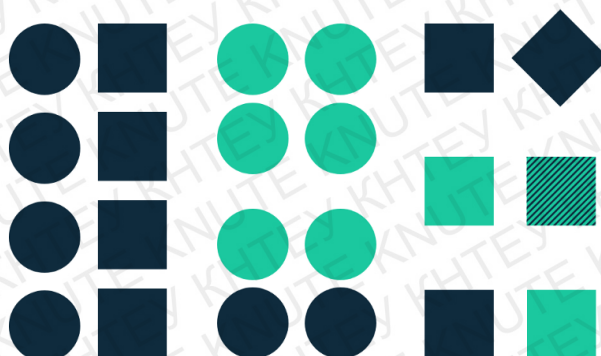


Рисунок 1.4. Візуалізація гештальт-принципу Схожість

Джерело: [12]

Людині властиво сприймати схожі елементи як групу чи патерн. Крім того, мозок припускає, що ці елементи мають однакове призначення. Є декілька способів зробити елементи схожими: за кольором, розміром, текстурою, формою і орієнтацією. При цьому, колір є більш визначальним фактором, ніж розмір, а розмір - більш важливим, ніж форма. Для виділення більш значимого елементу можна використати метод Аномалії, тобто виділити потрібний елемент серед інших за допомогою розміру, кольору, текстури, форми чи орієнтації. У додатку В бачимо, що Аномалія використовується для виділення найбільш вигідного тарифу та найбільш релевантного відгуку.

Замкнутість (Closure). Принцип Замкнутості говорить про те, що навіть коли об'єкт неповний або якісь частини не з'єднані між собою, мозок все одно сприймає цей елемент як єдине ціле. Тобто при меншій кількості деталей можна передати той самий сенс (рис. 1.5).



Рисунок 1.5. Візуалізація гештальт-принципу Замкнутість

Джерело: [12]

При розробці UI можна використати принцип Замкнутості для відображення іконографіки (іконок).

За допомогою невеликої кількості простих фігур можна передати важливу інформацію (рис. 1.6).

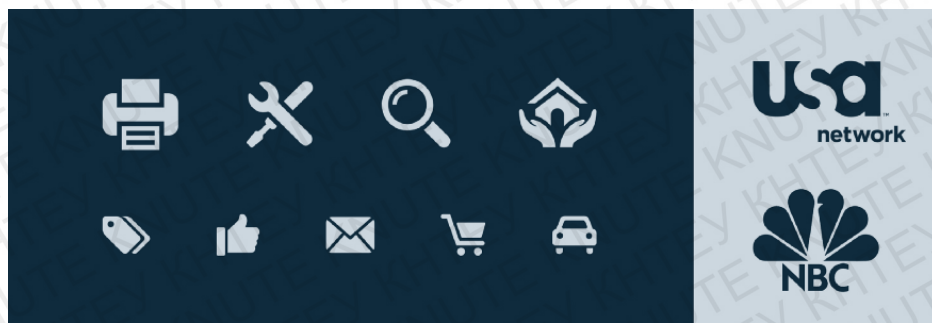


Рисунок 1.6. Приклад використання гештальт-принципу Замкнутість

Джерело: [12]

Симетрія (Symmetry). Симетричні елементи, навіть якщо вони знаходяться на відстані, зазвичай сприймаються взаємопов'язаними і створюють відчуття цілості й порядку (рис. 1.7).

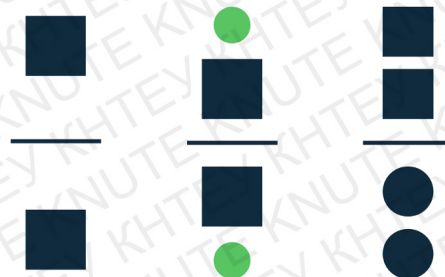


Рисунок 1.7. Візуалізація гештальт-принципу Симетрія

Джерело: [12]

Симетрія дозволяє користувачу відчувати комфорт, не відволікатися і сприймати лише на важливому. Симетрію варто використовувати, щоб

ефективно донести інформацію до відвідувача. Проте повністю симетричний дизайн створює відчуття нудьги, тому варто виділяти важливі елементи асиметрією. У Додатку Г бачимо, що невелика кількість асиметрії додає дизайну свіжості та приваблює око.

Симетрію варто використовувати для галерей, продуктових каталогів, навігації та іншого.

Продовження (Continuation). Елементи, що вибудовані по прямій або плавно зігнутій лінії, здаються більш взаємопов'язаними, ніж ті, що розміщені хаотично або по ламаній лінії (рис. 1.8).



Рисунок 1.8. Візуалізація гештальт-принципу Продовження

Джерело: [12]

Оскільки мозок людини переважно йде по шляху найменшого спротиву, тому елементи розміщені по прямій сприймаються ним, як згруповані.

При розробці UI принцип Продовження можна реалізувати за допомогою вирівнювання. Коли ефект Продовження переривається, користувач розуміє, що далі розміщено інший логічний блок (Додаток Д).

Спільна доля (Common Fate). Елементи, що рухаються в одному напрямку, сприймаються більш взаємопов'язаними, ніж ті, що рухаються в іншу сторону або стоять на місці (рис.1.9).



Рисунок 1.9. Візуалізація гештальт-принципу Спільна доля

Джерело: [12]

Цей принцип працює навіть коли елементи не рухаються явно, а лише мають вказівку на рух (мають форму стрілки чи вказівника). Принцип спільної долі можна використовувати для випадаючих меню, акордеонів, спливаючих підказок, слайдерів з продуктами, сторінок з ефектом паралаксу і елементів, що керуються прокруткою (Додаток Е).

Дизайн - це не лише гарні картинки та привабливі елементи інтерфейсу, це й ефективна взаємодія з користувачем.

1.2. Види інтерфейсів веб-системи

Виділяють велику кількість видів користувацьких інтерфейсів, проте розглянемо 6 з них[8, 9]:

1. Інтерфейс командного рядка (Command Line Interface, CLI);
2. Інтерфейс, керований за допомогою меню (Menu-driven Interface);
3. Графічний інтерфейс (Graphical User Interface, GUI);
4. Графічний інтерфейс користувача з сенсорним екраном (Touchscreen Graphical User Interface);
5. Інтерфейс, керований голосом (Voice-controlled interfaces, VUI);
6. Інтерфейс на основі жестів (Gesture-based interface).

Інтерфейс командного рядка не поширений так, як форма базового інтерфейсу користувача у повсякденних споживчих продуктах, але він все ще використовується за певних обставин. Інтерфейс командного рядка вимагає від користувачів введення відповідних інструкцій у командний рядок.

Комп'ютер спочатку отримує команду перейти до необхідного файлу або каталогу. Звідти стає доступним цілий ряд команд, починаючи з вилучення файлів і закінчуючи програмами.

Переваги інтерфейсу командного рядка:

- Проста структура;
- Мінімальне використання пам'яті;
- Відмінно підходить для повільно працюючих комп'ютерів або для тих, яким не вистачає пам'яті;
- Досвідчений користувач CLI може реалізовувати команди та виконувати завдання набагато швидше, ніж при використанні альтернативного типу інтерфейсу.

Недоліки інтерфейсу командного рядка:

- Важко вивчити командну мову;
- Складно для користувачів-початківців;
- Повідомлення про помилку містять мінімальну кількість інформації.

Інтерфейс користувача, що керується за допомогою меню, надає ряд команд або опцій у вигляді списку або меню, що відображається у повноекранному, спливаючому або спадному меню. Банкомат - це приклад інтерфейсу, керованого меню.

Переваги інтерфейсу, керованого за допомогою меню:

- Не обов'язково запам'ятовувати довгий список команд вручну;
- Простий інтерфейс для новачків;
- Пояснення знаходяться безпосередньо в меню.

Недоліки інтерфейсу, керованого за допомогою меню:

- Для досвідчених користувачів такий інтерфейс є повільним;
- Обмежені можливості меню;
- Часто для виконання простих функцій потрібен доступ до декількох екранів меню.

Графічний інтерфейс користувача або графічний інтерфейс - це тип інтерфейсу, з яким більшість людей знайомі. Взаємодія з цими інтерфейсами

відбувається з використанням миші, тачпаду або іншого периферійного пристрою, щоб вказувати та натискати на графіку або значки.

Користувачі взаємодіють із візуальними зображеннями на цифрових панелях управління. Робочий стіл комп'ютера - це графічний інтерфейс.

Переваги графічного інтерфейсу:

- Пояснюється само собою (користувач інтуїтивно розуміє, як виконати ту чи іншу команду, наприклад, хрестик на модальному вікні його закриття);
- Простий у використанні;
- Запам'ятовувати списки команд не потрібно;
- Дозволяє одночасно запускати декілька додатків, програм та завдань;
- Міцні засоби підтримки;
- Подібний формат серед різних програм додає знайомства;
- WYSIWYG (What You See Is What You Get - з англійської перекладається, як "Що ти бачиш, те й отримаєш") спрощує дизайн та форматування. Існують спеціальні графічні редактори для редагування веб-сторінок, які дозволяють навіть без використання програмного коду створювати сторінки. При цьому можна створити сторінку за допомогою графічного інтерфейсу і потім побачити код створеного контенту. Прикладами таких редакторів є TinyMCE Editor, Draft.js, Froala, CKEditor 4, Editor.js, Slate, tiptap, Dante Editor, Toast Editor.

Недоліки графічного інтерфейсу:

- Використовує велику кількість пам'яті. Проте, оскільки комп'ютери постійно розвиваються і стають більш потужними, то цей недолік не дуже викликає хвилювання.

Графічний інтерфейс сенсорного екрану дуже схожий на звичайний графічний інтерфейс, за винятком того, що для взаємодії з ним використовуються пальці або стилус для вибору піктограм та виконання завдань, а не миші або тачпад. Графічні інтерфейси сенсорного екрану зазвичай зустрічаються на планшетах, смартфонах та медичних пристроях,

таких як інсуліновий насос t: slim. Графічний інтерфейс сенсорного екрану має ті ж переваги та недоліки, що і стандартний графічний інтерфейс, але також пропонує більш тісний метод взаємодії. Відсутність периферійних пристроїв робить графічний інтерфейс з сенсорним екраном дуже зручним.

Інтерфейс, керований голосом дозволяє користувачеві взаємодіяти з системою за допомогою голосових або мовних команд. Віртуальні помічники, такі як Siri, Google Assistant та Alexa, є прикладами VUI. Основна перевага VUI полягає в тому, що він дає змогу користуватися інтерфейсом без використання рук та очей, зосереджуючи свою увагу в іншому місці[10].

Застосування до VUI тих же керівних принципів дизайну, що і до графічних інтерфейсів користувача, неможливе. У VUI немає візуальних можливостей. Тому, дивлячись на VUI, користувачі не мають чітких ознак того, що може робити інтерфейс або які є варіанти його використання. При розробці дій VUI важливо, щоб система чітко вказувала можливі варіанти взаємодії, повідомляла користувачеві, які функції він/вона може виконувати, і обмежувати кількість інформації, яку вона видає, до такої кількості, яку користувач може запам'ятати.

Оскільки люди зазвичай асоціюють голос із міжособистісним спілкуванням, а не з взаємодією між людиною та технологією, вони іноді не впевнені, запити якого рівня складності є допустимими для інтерфейсу. Таким чином, для успішної взаємодії з VUI він не тільки повинен мати чудову здатність розуміти розмовну мову, але й повинен навчити користувачів розуміти, які типи голосових команд вони можуть використовувати та які типи взаємодій вони можуть виконувати.

Користувальницький інтерфейс на основі жестів - це інновація, яка використовує жести як вхідні дані.

Інтерфейс жестів - це використання певних жестів, таких як прокрутка, стискання та торкання для управління інтерфейсом. Також до жестів управління відносяться нахили, рух очей і тряска. Інтерфейс користувача та технологія розпізнавання жестів еволюціонували від найпростіших рухів до

складних, і тепер вони є частиною повсякденного життя величезної кількості людей. Оскільки ця технологія продовжує розвиватися, майбутні можливості також неймовірно захоплюючі[11].

В даний час смартфони та планшети - це найпоширеніше місце, де повсякденні споживачі можуть знайти жестовий інтерфейс. Переважна більшість сучасних телефонів, починаючи від айфонів Apple і закінчуючи Samsung Galaxy, містять деякі елементи користувацького інтерфейсу жестів - від прокрутки, що є загальною для більшості телефонів та планшетів, до розпізнавання орієнтації.

Samsung Galaxy S4 - приклад смартфона, який зробив величезний крок у розвитку інтерфейсу жестів. Телефон відстежує і розпізнає рух ваших очей і автоматично прокручує вниз, коли ваші очі досягають нижньої частини сторінки.

Сучасні дослідження та експерименти свідчать про те, що в майбутньому жестовий інтерфейс у телефонах і планшетах буде використовувати більше камер та датчиків, що збільшить чутливість розпізнавання. Експерти стверджують, що телефони та планшети не тільки зможуть розуміти жести та вирази обличчя без дотику, але й будуть враховувати контекст, тобто ці пристрої зможуть передбачати те, що хочуть користувачі зробити, з більшою точністю, ніж зараз.

Ігри - це ще одна сфера, де користувацький інтерфейс жестів вже звичний. Наприклад, консолі Xbox 360 та Xbox One використовують систему Kinect для відстеження рухів гравців за допомогою камер та датчиків. Консолі PlayStation 3 і 4 використовують PlayStation Play подібним чином, тоді як Nintendo Wii та Wii U використовують датчики руху та пульти дистанційного керування для відстеження жестів та рухів гравців.

Крім того, за словами Microsoft та Sony, їх бачення майбутнього ігор включає в себе поєднане використання відстеження погляду, розпізнавання жестів та мозкових хвиль або технологічного телекінезу для набагато більш захоплюючого ігрового досвіду, що межує з віртуальною реальністю.

Саме інтерфейс жестів використовується у іграх у віртуальній реальності.

У світі медицини майбутнє користувацького інтерфейсу жестів дуже перспективне. Одним з найбільш помітних нововведень є впровадження жестового інтерфейсу в хірургії. Плани включають систему розпізнавання жестів руками, яка дозволяє хірургам переглядати зображення та записи пацієнтів під час операції. Можливість керувати інтерфейсом за допомогою жестів без дотику скорочує час операції, оскільки хірурги не будуть змушені виходити з операційної, щоб отримати доступ до традиційних комп'ютерних терміналів.

Найбільшою перевагою є потенціал зниження ризику зараження. Якщо хірурги можуть маніпулювати інтерфейсами без фізичного контакту, ризик передачі пацієнту шкідливих бактерій значно зменшується.

З шести типів користувацького інтерфейсу графічний інтерфейс, безумовно, є найпоширенішим, за яким слідує варіація сенсорного екрану. Незважаючи на альтернативні технології, які вже існують і продовжують з'являтися, графічний інтерфейс залишається кращим стандартом. Багато в чому це пояснюється простотою та зручністю використання.

Більшість кінцевих користувачів вважають, що графічні інтерфейси легше зрозуміти, оскільки піктограми та меню зазвичай зрозумілі, а також графічний інтерфейс не вимагає від користувача запам'ятовування або введення складних команд.

1.3. Існуючі інтерфейси для веб-систем поліграфічного підприємства

Ринок систем управління поліграфічним підприємством надає готові рішення для автоматизації роботи. Найвідомішими є ASystem (рис. 1.10), Sevit Print Shop Manager (рис. 1.11), PrintEffect (рис. 1.12), 1С Поліграфія 8 (рис. 1.13). Усі ці системи є настільними застосунками, тобто встановлюються на комп'ютер і займають на ньому місце. На нашу думку,

ефективніше використовувати веб-системи, що відкриваються безпосередньо в браузері. Звичайно у такої реалізації є недоліки (наприклад, така система не працюватиме за відсутності Інтернету), проте ми вважаємо, що переваг при такому підході більше. Уся інформація зберігається в хмарі, тобто втратити дані досить важко. Також при такій організації системи у додатку можуть працювати без перешкод одразу декілька працівників.

Створення веб-системи дозволяє також доступ не лише з комп'ютера, а й з телефону чи планшета. Якщо менеджер знаходиться не на робочому місці, а його негайно потрібно виконати якусь функцію (переглянути, змінити, видалити, додати замовлення тощо), він/вона може з телефону чи планшета це зробити.

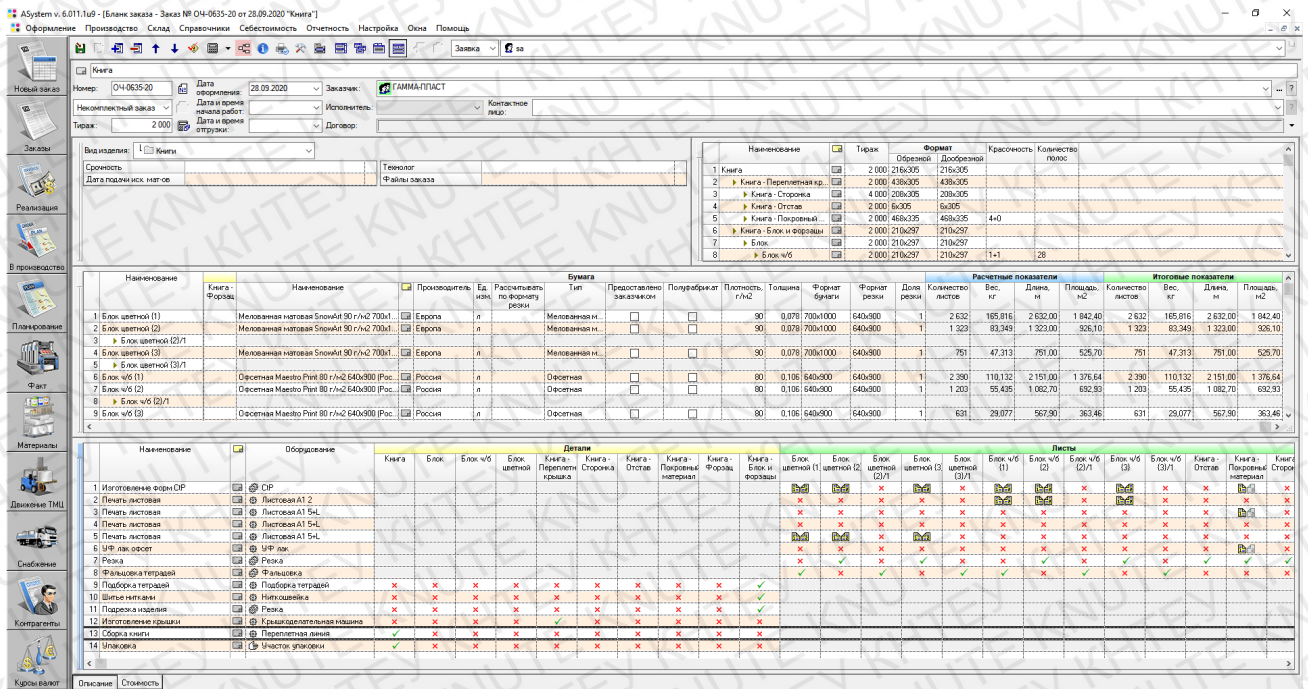


Рисунок 1.10. Интерфейс додатку ASystem

Джерело: Авторська розробка

Sevit Print Shop Manager : Заказы

Офис: Главный офис | Услуги: Все | Статус: Все | Создан: Все
 Тип: Все | Материал: Все | Оплата: Не оплачен | Месяц: Все | 00/00/0000
 Клиент: Все | Доп. услуги: Все | Способ: Все | Дата: Все | 00/00/0000
 Почетить/сбросить все | Точный | F11: Сброс поиска | F12: Сброс фильтров | Дата оплаты: Все | Сорт: Порядок создания ↑

Принят	Заказ	Услуги	Клиент	Описание	Доп. услуги	Статус	Сроки	Стоимость	Оплата
<input type="checkbox"/>	02/11/11 15:42:22 Палеолог	842235 VUTEK 360 dpi	ИЧП Д.Черномор	банер золушка Баннер литой DLS (Россия) 2940 мм x 1640 мм, 1 экз.	Поля	Готов : Ожидает заказчика	07/11/11 09:00:00	1,525.05 руб	Нет Безнал Счет
<input type="checkbox"/>	07/11/11 15:23:29 Складовская-1	846604 ROLAND 720x720 dpi	Стандартный	фото Пленка гляцевая Neschen (Германия) 5800 мм x 1600 мм, 1 экз.	Shutterstock - Большой размер	Препресс		5,738.34 руб 5,000.00 руб	Предоплата Касса Заказ-наряд : 2438
<input type="checkbox"/>	07/11/11 19:42:17 Складовская-1	847042 VUTEK 360 dpi	Стандартный	БАНЕР ТУРНИР ПО ВОЛ_1 Баннер литой DLS (Россия) 2000 мм x 9000 мм, 1 экз.	Люверсы с произвольным интеро Карманы по периметру Укрепления	Ожидает оплаты	10/11/11 09:00:00	5,811.60 руб	Нет Безнал Счет : 2454
<input type="checkbox"/>	08/11/11 14:05:19 Складовская-1	847779 OCE 1440x1440 dpi	Стандартный	осминог 2x0.35 Пластик 3 мм 2000 мм x 350 мм, 1 экз.	Макет с векторной графикой Вырезать по контуру	В печати		1,559.36 руб 800.00 руб	Предоплата Касса Заказ-наряд : 2446
<input type="checkbox"/>	08/11/11 14:24:03 Палеолог	847820 ROLAND 720x720 dpi	ИЧП Д.Черномор	МАМА МИА 3 Пленка матовая Neschen (Германия) 1419 мм x 121 мм, 1 экз.	Ламинация матовой прозрачной	В печати		244.13 руб	Нет Безнал Счет
<input type="checkbox"/>	15/01/12 17:43:19 Администрат	848234 Хегох 6060 4+0	НИИ ЧЛВО	Объявления о найме Бумага мелованная матовая 300 г/м2 A4 (297 мм x 210 мм), 300 экз.		Драфт		4,407.00 руб	Нет Безнал Счет
<input type="checkbox"/>	19:27:10 23/01/12 Администрат	848254 ROLAND 360x720 dpi	ИЧП Д.Черномор	Заказ № 848253, часть 2 (№ 847820) Пленка с синим клеевым слоем SOLO (Корея) 1200 мм x 2100 мм, 2 экз.		Драфт	24/01/12 19:27:10	2,733.70 руб	Нет Безнал Счет
<input type="checkbox"/>	07/04/13 15:36:18 Администрат	848271 Хегох 6060 4+0	Без привязки к прайс-листу	Заказ для клиента без привязки к пра Бумага мелованная 170 г/м2 A4 (297 мм x 210 мм), 1 экз.		Драфт	08/04/13 15:36:18	0.00 руб	Нет Безнал Счет
<input type="checkbox"/>	07/04/13 15:43:10 Администрат	848274 Товары Стандартный	Стандартный	Продажа CD-R болванок Чистый CD-R (болванка) 20 экз.		Драфт	08/04/13 15:43:10	0.00 руб	Нет Безнал Счет

Рисунок 1.11. Интерфейс добавки Sevit Print Shop Manager

Джерело: Авторська розробка

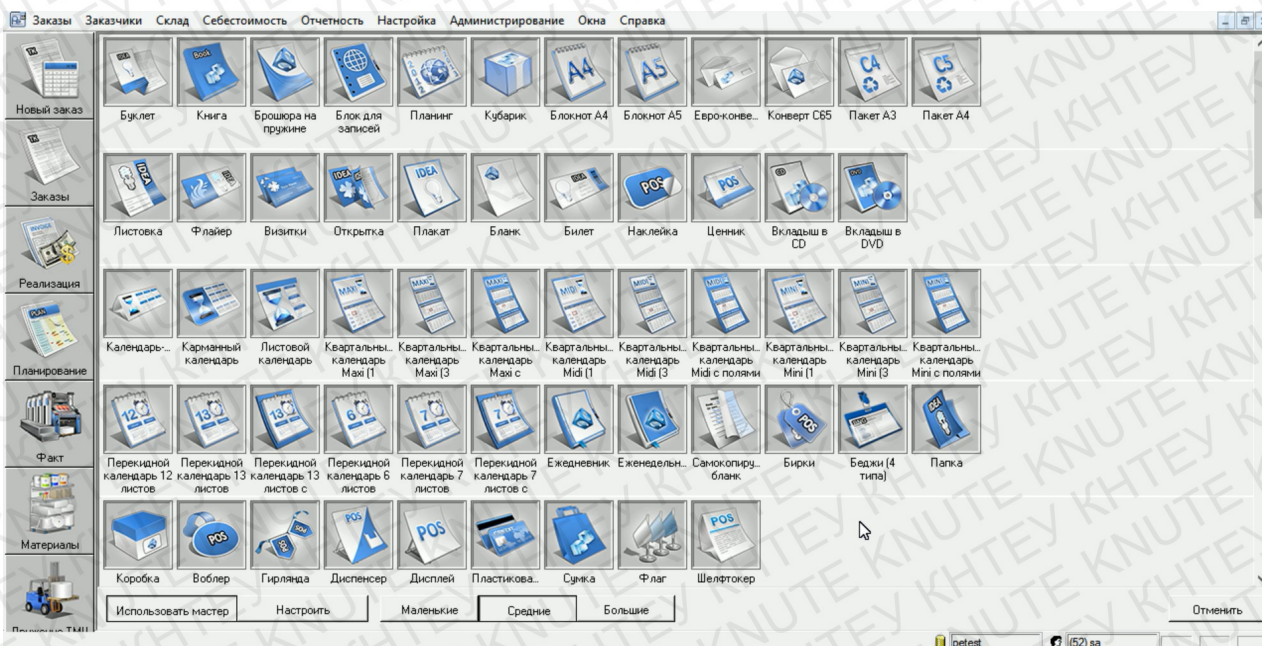


Рисунок 1.12. Интерфейс добавки PrintEffect

Джерело: Авторська розробка

Дата начала	Дата окончания	Номер	Контрагент	Продукция	Характеристика	Тираж	Формат продукции (м)	
							Ширина	Выс...
04.07.2018 0:00:00	07.08.2018 0:00:00	ОПНФ-000013	Призма ООО	Конверт		120 000	110,000	220,0
01.08.2019 0:00:00	30.09.2017 0:00:00	П0000000004	ИП Колосков	Конверт		100 000	110,000	220,0
14.07.2019 0:00:00	16.07.2019 0:00:00	ОПНФ-000004	Новый покупатель	Листовка		10 000	148,000	200,0
18.06.2018 0:00:00	29.06.2018 0:00:00	ОПНФ-000005	Новый покупатель	Листовка		10 000	148,000	200,0
03.07.2018 0:00:00	31.07.2018 0:00:00	ОПНФ-000011	Новый покупатель	Листовка		50 000	148,000	200,0
14.08.2017 0:00:00	31.10.2017 0:00:00	П0000000006	Магистраль ООО	Открытка		20 000	210,000	297,0
01.08.2017 0:00:00	02.08.2017 18:00:00	П0000000009	Магистраль ООО	Плакат		500	458,000	330,0
11.10.2019 16:24:11	11.10.2019 16:00:00	ОПНФ-000005	Призма ООО	Плакат		2 000	300,000	500,0
11.10.2019 16:42:33	11.10.2019 16:00:00	ОПНФ-000007	Призма ООО	Плакат		1 000	300,000	500,0
14.07.2019 9:00:00	17.07.2019 0:00:00	ОПНФ-000002	Призма ООО	Плакат		6 000	458,000	330,0
14.07.2019 13:15:42	17.07.2019 15:00:00	ОПНФ-000003	Магистраль ООО	Плакат		4 000	458,000	330,0
01.08.2017 0:00:00	11.08.2017 0:00:00	П0000000005	Призма ООО	Плакат		1 000	300,000	500,0
01.09.2017 0:00:00	31.10.2017 0:00:00	ОПНФ-000001	ИП Колосков	Упаковка для ж...	Кефир Николь...	50 000	200,000	290,0
18.06.2018 0:00:00	29.06.2018 0:00:00	ОПНФ-000003	Новый покупатель	ЧИП - Пластико...		5 000	54,000	86,0

Рисунок 1.13. Интерфейс додатку 1С Поліграфія 8

Джерело: Авторська розробка

Зовнішній вигляд представлених вище систем автоматизації поліграфічного підприємства є застарілим та незрозумілим. Таким чином, новому співробітнику знадобиться час на навчання, щоб реалізувати свої функції. Інтерфейс системи має бути максимально зрозумілим, щоб скоротити час на навчання і зменшити кількість помилкових дій.

Висновки до розділу 1

Проаналізувавши інформацію у першому розділі, можна дійти висновку, що дуже важливо продумати сценарій дій користувача відповідно до психології людини, максимально його спростити і розмістити кроки по прямій лінії. Для розробки ефективного, зручного та зрозумілого інтерфейсу можна використати гештальт-принципи. Важливо візуалізувати інтерфейс у графічному редакторі перед тим, як накладати на нього дизайн.

Для того, щоб кінцевий користувач був задоволений роботою з веб-системою, потрібно прискіпливо та уважно пропрацювати усі важливі компоненти та крок за кроком побудувати зручний та привабливий на вигляд додаток.

Виділяють такі основні види інтерфейсів: Інтерфейс командного рядка (Command Line Interface, CLI); Інтерфейс, керований за допомогою меню (Menu-driven Interface); Графічний інтерфейс (Graphical User Interface, GUI); Графічний інтерфейс користувача з сенсорним екраном (Touchscreen Graphical User Interface); Інтерфейс, керований голосом (Voice-controlled interfaces, VUI); Інтерфейс на основі жестів (Gesture-based interface).

Ринок пропонує готові рішення для автоматизації роботи поліграфічного підприємства. Наприклад, ASystem, Sevit Print Shop Manager, PrintEffect, 1С Поліграфія 8. Але вони мають застарілий дизайн інтерфейсу, не відповідають принципам побудови інтерфейсів та реалізовані у формі настільного додатку.

РОЗДІЛ 2. ПОБУДОВА БІЗНЕС-ПРОЦЕСІВ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА, ПОВ'ЯЗАНИХ З ПРОДАЖЕМ ТОВАРІВ ТА ПОСЛУГ

2.1. Виділення основних бізнес-процесів поліграфічного підприємства

Бізнес-процес - це послідовність кроків, що реалізується групою стейкхолдерів для досягнення певної, конкретної мети[13]. Важливо будувати модель бізнес-процесів підприємства, оскільки таким методом можна відслідковувати ефективність усіх процесів, що відбуваються на підприємстві.

Спосіб керування компанією за допомогою побудови моделі бізнес-процесів, на нашу думку, має такі переваги, як всеохопність, тобто відображаються усі аспекти діяльності компанії, гнучкість (у модель легко вносити доповнення, виправлення чи видаляти процеси, що вже не мають місця в роботі підприємства). А також такий спосіб управління дозволяє легко відслідковувати показники ефективності роботи компанії.

Виділяють 3 основні типи бізнес-процесів[14]:

1. **Операційні.** Це ті бізнес-процеси, що пов'язані безпосередньо з основним видом діяльності компанії. Наприклад, якщо розглянути магазин еко-товарів, то прийом замовлення, збір товарів, що замовив клієнт і видача замовлення - це саме операційні бізнес-процеси.
2. **Управлінські.** Бізнес-процеси, що передбачають контроль за виконанням щоденних завдань, планування цілей компанії на майбутнє, навчання персоналу або запуск нового продукту чи послуги, є управлінськими. Ці процеси реалізуються вищим керівництвом компанії.
3. **Допоміжні.** Для того, щоб створити середовище для реалізації операційних та управлінських процесів, потрібні допоміжні. У магазині

еко-товарів допоміжними бізнес-процесами є найм персоналу HR-відділом, охорона приміщень, робота бухгалтерського відділу тощо.

У цій роботі виділимо операційні та управлінські бізнес-процеси поліграфічного підприємства STARTER, що пов'язані з роботою менеджера.

Для того, щоб побудувати модель бізнес-процесів підприємства, потрібно відтворити такі кроки:

- побудувати схему організаційної структури підприємства;
- побудувати матрицю відповідальності;
- розписати паспорти бізнес-процесів;
- побудувати модель бізнес-процесів підприємства.

Організаційна структура поліграфічного підприємства STARTER зображена на рисунку 2.1.

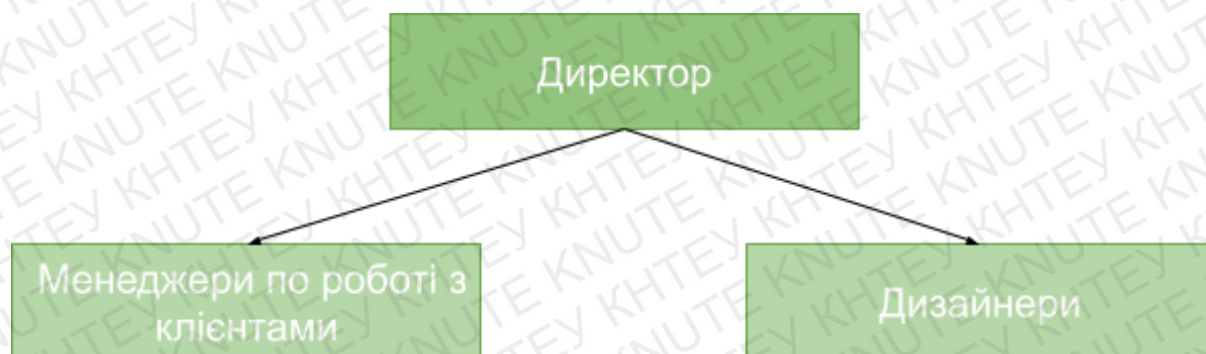


Рисунок 2.1. Схема організаційної структури поліграфічного підприємства STARTER

Джерело: Авторська розробка

До обов'язків менеджерів входить прийом замовлень від клієнтів, написання технічного завдання для дизайнера, передача замовлення на виробництво підрядникам, прийом замовлення з виробництва, видача замовлення клієнту, збір відгуків від клієнтів.

Обов'язками дизайнерів є прийняти технічне завдання від менеджера, виконати якісно та зазначені терміни завдання, передати готову роботу менеджеру.

Директор, у свою чергу, зобов'язаний проводити переговори з новими клієнтами для їх залучення до клієнтської бази, ініціювати появу нових

товарів чи послуг, слідкувати за показниками ефективності роботи компанії, оплачувати роботу аутсорсингових компаній, виплачувати заробітну плату працівникам.

Виходячи з вищенаведеної інформації, можна побудувати матрицю відповідальності (таблиця. 2.1).

Таблиця 2.1

Матриця відповідальності поліграфічного підприємства STARTER

Назва бізнес-процесу	Менеджери	Дизайнери
Процес прийому замовлення	Вик	-
Процес створення макету	Відп	У
Процес передачі замовлення на друк підрядникам	Відп	-
Процес передачі замовлення клієнту	Вик	-
Процес збору відгуків від клієнтів	Вик	-

Джерело: Авторська розробка

Умовні позначення:

Відп - відповідальний за бізнес-процес (несе відповідальність за результат бізнес-процесу, але може його не виконувати)

Вик - виконавець (і виконує бізнес-процес, і несе відповідальність за його результат)

У - учасник (бере участь у бізнес-процесі)

Після побудови матриці відповідальності можна переходити до описання паспортів бізнес-процесів.

Процес прийому замовлення (таблиця 2.2) є першим у циклі роботи підприємства. Клієнт виявляє бажання замовити продукції в поліграфії,

передає менеджеру вимоги щодо вигляду продукції чи макет, якщо він в нього є, і надає передплату.

Таблиця 2.2

Паспорт бізнес-процесу Процес прийому замовлення

Параметр		Значення
Найменування процесу		Процес прийому замовлення
Результати		Прийняте замовлення
Виходи БП	матеріальні	-
	інформаційні	Технічне завдання для дизайнера (якщо це потрібно)
Входи БП	матеріальні	Гроші (передплата)
	інформаційні	Запит на виконання робіт, вимоги до готового продукту
Власник БП		Менеджер
Управлінські (нормативні) документи		Правила спілкування з клієнтами, Правила оформлення технічного завдання
Учасники	персонал	Менеджер
	засоби реалізації	веб-сайти, інтернет, телефон, соціальні мережі, електронна пошта, комп'ютер
Споживач		Дизайнер (якщо потрібно відмальовувати макет)
Попередній процес		-
Наступний процес		Процес створення макету або Процес передачі замовлення на друк підрядникам

Джерело: Авторська розробка

Менеджер обробляє замовлення клієнта і складає технічне завдання для дизайнера за правилами написання технічного завдання або передає замовлення на виробництво друкарні, якщо клієнт має макет. Клієнт може зробити замовлення через декілька каналів зв'язку: зателефонувати,

залишити заявку на сайті компанії, в соціальних мережах (Instagram, Facebook, LinkedIn), надіслати повідомлення на електронну пошту чи особисто переговорити з менеджером в офісі компанії. Відповідальність за виконання цього бізнес-процесу несе менеджер.

Наступний процес відбувається, якщо клієнт не має власного макету або до його макету треба вносити виправлення (таблиця 2.3).

Таблиця 2.3

Паспорт бізнес-процесу Процес створення макету

Параметр		Значення
Найменування процесу		Процес створення макету
Результати		Готовий макет у графічному редакторі
Виходи БП	матеріальні	-
	інформаційні	Файл з макетом
Входи БП	матеріальні	-
	інформаційні	Технічне завдання
Власник БП		Менеджер
Управлінські (нормативні) документи		-
Учасники	персонал	Дизайнер
	засоби реалізації	Графічні редактори (Adobe Photoshop, Adobe Illustrator, Figma тощо), Інтернет
Споживач		Менеджер
Попередній процес		Процес прийому замовлення
Наступний процес		Процес передачі замовлення на друк підрядникам

Джерело: Авторська розробка

Менеджер передає створене технічне завдання дизайнеру. Далі дизайнер за певний час виконує завдання, використовуючи графічні

редактори Adobe Photoshop, Adobe Illustrator, Figma та інші. Після закінчення виконання роботи дизайнер передає файл з макетом менеджеру.

Наступним кроком є передача макету на виробництво підрядникам (таблиця 2.4). Оскільки підприємство не має власної друкарні, то ці послуги надаються аутсорсинговою компанією.

Таблиця 2.4

Паспорт бізнес-процесу Процес передачі замовлення на друк
підрядникам

Параметр		Значення
Найменування процесу		Процес передачі замовлення на друк підрядникам
Результати		Готовий результат
Виходи БП	матеріальні	Акт виконаних робіт, готовий продукт
	інформаційні	-
Входи БП	матеріальні	Гроші
	інформаційні	Файл з макетом
Власник БП		Менеджер
Управлінські (нормативні) документи		-
Учасники	персонал	Менеджер
	засоби реалізації	-
Споживач		Менеджер
Попередній процес		Процес прийому замовлення або Процес створення макету
Наступний процес		Процес передачі замовлення клієнту

Джерело: Авторська розробка

Для цього менеджер залишає заявку на сайті компанії-друкарні, прикріплюючи файли макету та після того, як менеджери компанії-друкарні

розрахують вартість надання послуги, повідомляє директора, що потрібно оплатити замовлення.

Після того, як друкарня виконає свою роботу, менеджер забирає замовлення і перевіряє його на якість і кількість.

Після перевірки продукції менеджер може переходити до наступного кроку - Процес передачі замовлення клієнту (таблиця 2.5).

Таблиця 2.5

Паспорт бізнес-процесу Процес передачі замовлення клієнту

Параметр		Значення
Найменування процесу		Процес передачі замовлення клієнту
Результати		Виконане замовлення
Виходи БП	матеріальні	Акт виконаних робіт
	інформаційні	-
Входи БП	матеріальні	Готовий продукт
	інформаційні	-
Власник БП		Менеджер
Управлінські (нормативні) документи		-
Учасники	персонал	Менеджер
	засоби реалізації	телефон, соціальні мережі, месенджери, Інтернет
Споживач		-
Попередній процес		Процес передачі замовлення на друк підрядникам
Наступний процес		Процес збору відгуків від клієнта

Джерело: Авторська розробка

Менеджер зв'язується з клієнтом, зателефонувавши, написавши в соціальних мережах чи месенджерах йому/їй, і повідомляє про готовність замовлення. Клієнт обирає спосіб отримання замовлення (таксі, пошта чи

особисте отримання в офісі). У разі вибору клієнтом способу доставки таксі чи пошта, менеджер організовує доставку продукції.

Після отримання клієнтом замовлення менеджер робить запит клієнту на написання відгуку (таблиця 2.6). Цей процес є дуже важливим для розвитку компанії, оскільки саме клієнти помічають прогалини у сервісі чи продукції.

Таблиця 2.6

Паспорт бізнес-процесу Процес збору відгуків від клієнта

Параметр		Значення
Найменування процесу		Процес збору відгуків від клієнта
Результати		Заповнена форма відгуку
Виходи БП	матеріальні	-
	інформаційні	Заповнена форма відгуку
Входи БП	матеріальні	-
	інформаційні	Форма відгуку
Власник БП		Менеджер
Управлінські (нормативні) документи		-
Учасники	персонал	-
	засоби реалізації	телефон, соціальні мережі, месенджери, Інтернет, електронна пошта
Споживач		Директор
Попередній процес		Процес передачі замовлення клієнту
Наступний процес		-

Джерело: Авторська розробка

Менеджер надсилає клієнту лист з проханням заповнити форму відгуку. Усі відгуки переглядає директор і в разі виникнення негативного досвіду

взаємодії з компанією, ініціює дії для покращення сервісу чи виготовлення продукції.

2.2. Побудова моделі бізнес-процесів поліграфічного підприємства

Для побудови моделі бізнес-процесів ми підготували паспорти бізнес-процесів, що значно спростить відмальовування моделі. Для малювання моделі використаємо методологію BPMN та інструмент bpmn.io[15].

BPMN (Business Process Model and Notation) - це нотація або методологія, яка надає підприємствам можливість розуміти свої внутрішні бізнес-процедури у графічній нотації та дає організаціям можливість оголошувати ці процедури у стандартний спосіб. Крім того, графічне позначення полегшить розуміння співпраці щодо ефективності та ділових операцій між організаціями. Це гарантує, що підприємства зрозуміють себе та учасників свого бізнесу, і дасть можливість організаціям швидко пристосуватися до нових внутрішніх та B2B бізнес-обставин[16].

Взагалі нотація бізнес-моделювання стандартизує спосіб відображення елементів бізнес-процесу. BPMN, наприклад, передбачає, яким чином варто позначати початкову та кінцеву подію, завдання, шлюз та інше.

Почнемо малювати модель бізнес-процесів з першого кроку - Прийом замовлення у клієнта.

Як можна побачити на рисунку 2.2 модель прийому замовлення має шлюз, тобто розгалуження за умови чи потрібно дизайнеру робити макет.

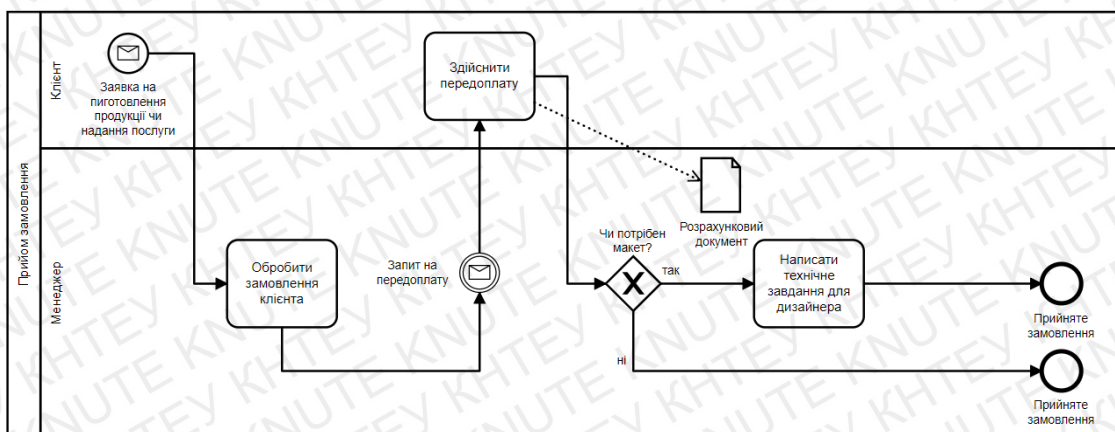


Рисунок 2.2. Модель бізнес-процесу Прийм замовлення у клієнта

Джерело: розроблено автором на основі [16]

Модель створення макету також передбачає розгалуження (рис. 2.3), оскільки клієнт може виявити бажання внести виправлення в макет або менеджер може побачити певні помилки чи невідповідності технічному завданню.

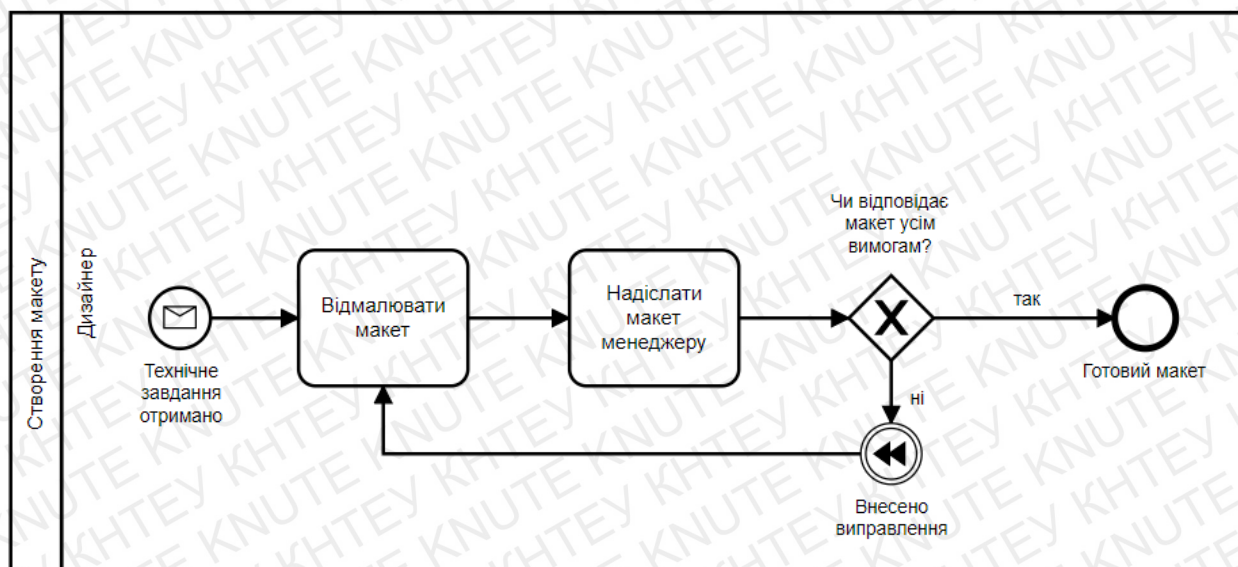


Рисунок 2.3. Модель бізнес-процесу Створення макету

Джерело: розроблено автором на основі [16]

Модель передачі замовлення на друк підрядникам (рис. 2.4) є досить простою, оскільки немає жодних розгалужень та умов.

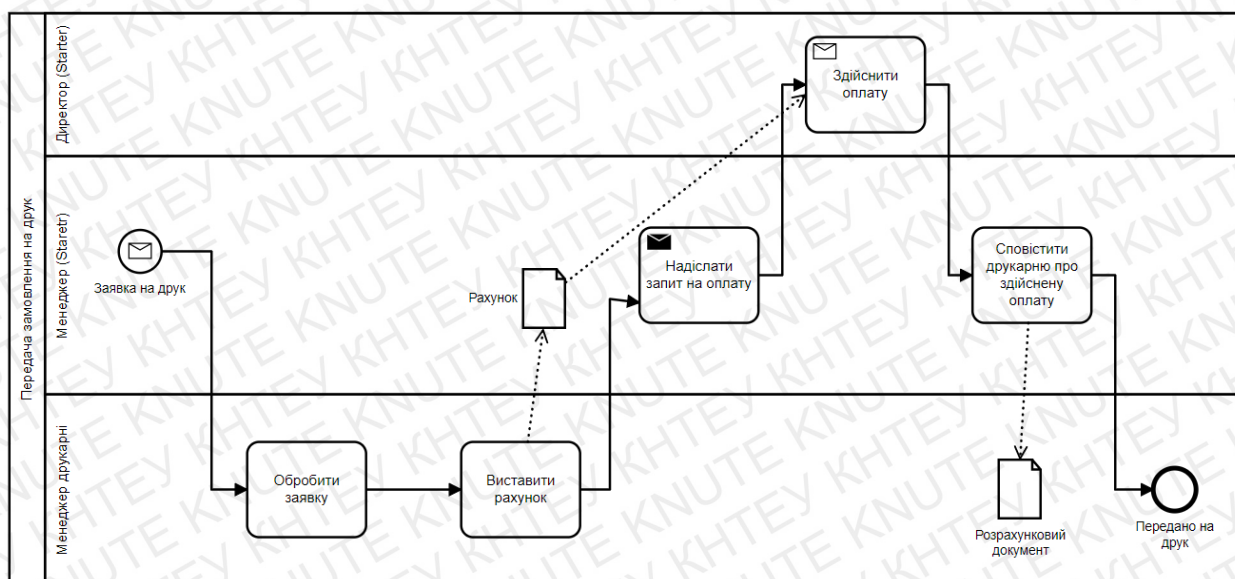


Рисунок 2.4. Модель бізнес-процесу Передача замовлення на друк підрядникам
Джерело: розроблено автором на основі [16]

Менеджер поліграфії заповнює заявку на сайті друкарні і далі робочі зобов'язання переходять до друкарні.

Модель передачі замовлення клієнту передбачає перевірку готової продукції менеджером. Якщо друкарня виконала замовлення з браком чи кількість продукції не відповідає заявленій, то менеджер скаржитися друкарні.

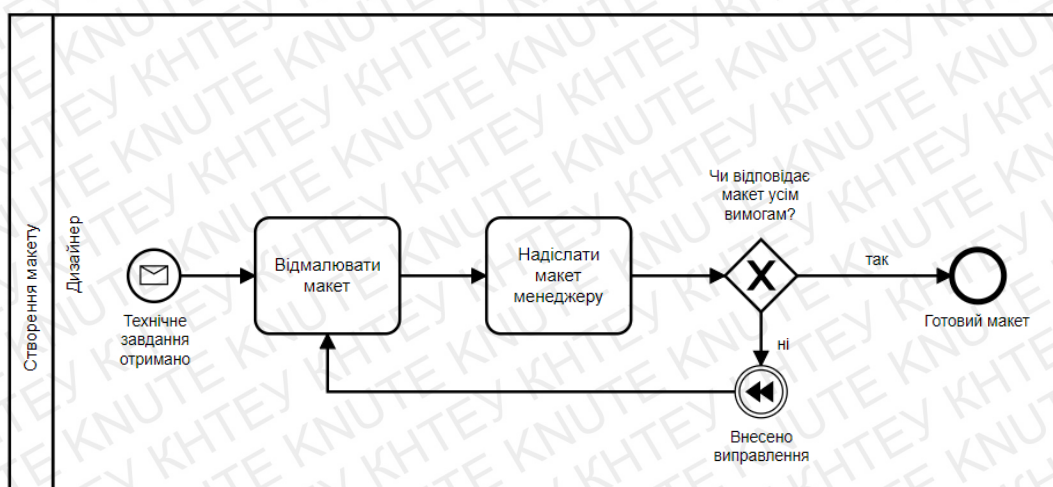


Рисунок 2.5. Модель бізнес-процесу Передача замовлення клієнту
Джерело: розроблено автором на основі [16]

Також було виділено процес збору відгуків, проте його ми не візуалізували, оскільки цей процес має лише просте завдання отримати форму відгуку від клієнта і внести ці дані у веб-систему.

2.3. Співставлення бізнес-процесів з функціоналом інтерфейсу веб-системи

Додаток має 4 основних модулі: модуль логіну, модуль директора, модуль дизайнера та модуль менеджера. Кожен користувач має свою роль, директор, дизайнер чи менеджер, і відповідно до своєї ролі після логіну він потрапляє на відповідний модуль.

Відповідно до моделі бізнес-процесів модуль менеджера має функціонал для додавання нового замовлення, перегляду існуючих замовлень, редагування замовлень, видалення замовлень, формування технічного завдання та відправка його менеджеру.

Замовлення має такі поля, як ім'я, прізвище, по-батькові клієнта, його контактні дані (електронна пошта та номер телефону), назва компанії, яку представляє клієнт (опціональне поле), дата створення замовлення, товари, що входять в замовлення, собівартість замовлення, націнка, вартість замовлення (сума, яку повинен сплатити клієнт), статус замовлення, чекбокс “Чи потрібен дизайн”, технічне завдання та дата готовності.

Після того, як менеджер додає замовлення із поміткою Потрібен дизайн і прикріплює технічне завдання, воно з'являється в модулі дизайнера. Далі вільні дизайнери вибирають для роботи замовлення і ставлять статус “Розробка дизайну”. Після цієї дії виконання дизайну закріплюється за певним дизайнером. Далі дизайнер виконує свою роботу і прикріплює макет до замовлення.

У модулі дизайнера є 3 вкладки: Потребують дизайн, В роботі та Виконані. Вкладка Потребують дизайн відображає ті замовлення, що вимагають розробку макета і ще жодним дизайнером не “заброньовані”. У вкладці В роботі відображаються ті замовлення, на яких дизайнер поставив

статус “Розробка дизайну”. При цьому в цій вкладці дизайнер бачить лише свої замовлення. А у вкладці Виконані дизайнер бачить ті замовлення, до яких він прикріпив макет і поставив статус “Готове до виробництва”.

Після того, як дизайнер виконав свою частину роботи, дизайнер бачить зміну статусу замовлення і перевіряє якість макету. Далі він робить заявку в друкарні. Цей етап відбувається поза межами веб-системи, тому цей бізнес-процес відображається в додатку лише зміною статусу товару.

Коли друкарня виконає свої зобов’язання, менеджер перевіряє замовлення на кількість і якість і, якщо все відповідає заявці, то бере контакти клієнта з веб-системи і сповіщає клієнта. Цей етап жодним чином не відображається в додатку. А от після прийняття замовлення клієнтом, менеджер змінює статус замовлення на Виконано.

Менеджер також посилає клієнту прохання про заповнення форми відгуку і заносить відгук до системи.

Директор, у свою чергу, бачить агреговану інформацію по виконаних та відмінених замовленнях, а також по виконаних та відмінених дизайнах. Є можливість фільтрувати інформацію по конкретному менеджеру чи дизайнеру, по даті, сортувати по даті, а також побачити суму вартості та собівартості всіх виконаних замовлень.

Висновки до розділу 2

Отже, поліграфічне підприємство STARTER має 3 основні позиції в організаційній структурі: менеджер, дизайнер та директор. Кожен з суб’єктів має свої обов’язки, які відображаються в бізнес-процесах та є операційною та управлінською діяльністю.

Можна виділити такі основні бізнес-процеси:

- Процес прийому замовлення;
- Процес створення макету;
- Процес передачі замовлення на друк підрядникам;

- Процес передачі замовлення клієнту;
- Процес збору відгуків від клієнтів.

Веб-система будується на основі бізнес-процесів, розроблених в другому розділі. Веб-система дозволить відслідковувати статуси товарів та замовлення загалом, тобто вся інформація зберігатиметься в одному місці і її буде легко знайти.

Для того, щоб кожен суб'єкт мав доступ лише до свого функціоналу, додаток поділено на окремі модулі: модуль логіну, модуль дизайнера, модуль менеджера та модуль директора. Суб'єкт може відвідувати сторінки лише того модулю, який відповідає його ролі.

РОЗДІЛ 3. СТВОРЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ ДЛЯ ВЕБ-СИСТЕМИ ПОЛІГРАФІЧНОГО ПІДПРИЄМСТВА

3.1. Верстка елементів інтерфейсу веб-системи

Для того, щоб спростити створення елементів інтерфейсу та чітко виділити компоненти, що повторюються, потрібно відмалювати бажаний дизайн в графічному редакторі. Ми використали зручний онлайн редактор Figma [17].

На макеті розміщено підказки та пояснення для зручності та швидкого розуміння користувачами інтерфейсу. Наприклад, у дизайнера є спеціальний компонент, який пояснює, яка норма часу виділена на створення дизайну відповідно до складності (рис.3.1).

Товари, що потребують створення макету			
НАЗВА	ЗАМОВНИК	ДАТА	СТАТУС
★ Простий дизайн - виконання займає 1-3 години	Блокнот з логотипом	Поліна Теренчук NoName Brand	19.10.2021 Створення дизайну ★
★★ Середня складність дизайну - виконання займає 4-8 годин	Візитки з тисненням	Іван Доронько Some Company	19.10.2021 Створення дизайну ★
★★★ Складний дизайн - виконання займає більше 9 годин	Баннер для квіткового магазину	Поліна Теренчук NoName Brand	19.10.2021 Створення дизайну ★★
	Блокнот з логотипом	Поліна Теренчук NoName Brand	19.10.2021 Створення дизайну ★★★
	Блокнот з логотипом	Поліна Теренчук NoName Brand	19.10.2021 Створення дизайну ★
	Блокнот з логотипом	Поліна Теренчук NoName Brand	19.10.2021 Створення дизайну ★

Рисунок. 3.1 Рівні складності дизайну на макеті

Джерело: Авторська розробка

Бізнес-процес Прийом замовлення від клієнта можна прослідкувати на макеті. Для створення нового замовлення менеджер натискає на кнопку Додати замовлення (рис. 3.2).

Замовлення Додати замовлення

НАЗВА	ЗАМОВНИК ↑	ДАТА ↓	Вартість	СТАТУС
Блокнот (13486785)	Поліна Теренчук NoName Brand	19.10.2021	1590 грн	Створення дизайну
Візитки (145986495)	Іван Доронько Some Company	19.10.2021	350 грн	Нове
Баннер, візитки, блокнот (89345670)	Поліна Теренчук NoName Brand	19.10.2021	1590 грн	Відмінено
Блокнот (4586394850)	Поліна Теренчук NoName Brand	19.10.2021	1590 грн	Виконано
Блокнот (3874753094)	Поліна Теренчук NoName Brand	19.10.2021	1590 грн	Виконано

Усі замовлення

Статус

- Нове
- Створення дизайну
- Готово до виконання
- У процесі виконання
- Готово
- Відмінено

Рисунок. 3.2. Сторінка замовлень у менеджера

Джерело: Авторська розробка

Після натискання на кнопку менеджера буде перенаправлено на сторінку створення нового замовлення (рис. 3.3).

Створення замовлення

Інформація про замовника

Ім'я* Прізвище* По-батькові

Компанія Яку представляє замовник Телефон Email

Товари Додати товар

Товарів поки в замовленні немає. Натисніть на кнопку Додати товар, щоб створити та прикріпити товар до замовлення

Вартість замовлення: 0

Статус замовлення: Нове Зберегти

Рисунок. 3.3. Сторінка створення нового замовлення у клієнта

Джерело: Авторська розробка

Менеджер вносить загальну інформацію про клієнта: його прізвище, ім'я, по-батькові, назва компанії, яку представляє клієнт та контактну інформацію таку, як електронна пошта та номер телефону.

Наступним кроком є додавання товарів до замовлення. Для цього потрібно натиснути на кнопку Додати товар (рис. 3.3).

Після натискання на кнопку з'явиться модальне вікно створення товару (рис. 3.4).

The image shows two versions of a 'Створити товар' (Create Item) modal form. Both forms have a title bar with a close button (X) and a text input field for 'Назва товару' (Item name). Below this is a 'Матеріали' (Materials) section with a note: 'Матеріали до товару поки не додано. Натисніть кнопку Додати матеріал, щоб додати матеріал та його кількість.' (Materials for the item have not yet been added. Click the Add material button to add material and its quantity.) and a 'Додати матеріал' (Add material) button. To the right of this button is a 'Вартість товару: 0' (Item price: 0) label. The 'Дизайн' (Design) section has a checkbox 'потрібен макет' (design template required). In (a), it is unchecked; in (b), it is checked. Below the checkbox is a text input field for 'Макет' (Design template) with a note: 'Посилання на файл з макетом, розміщений на Google Drive' (Link to the design template file on Google Drive) and the value 'https://someurl.com'. At the bottom, there is a 'Статус товару' (Item status) dropdown menu. In (a), it is set to 'Готовий до виконання' (Ready for execution); in (b), it is set to 'Новий' (New). Both forms have a red 'Зберегти' (Save) button at the bottom right.

а)

б)

Рисунок. 3.4. Форма створення товару в замовленні:

а) коли макет не потрібен; б) коли макет потрібен

Джерело: Авторська розробка

Шлюз із бізнес-процесу (Чи потрібен макет?) реалізовано за допомогою чекбоксу Потрібен макет. При натисканні на нього з'являється поле для вводу технічного завдання для дизайнера.

Після збереження замовлення, в якому стоїть галочка Потрібен макет, дизайнер побачить відповідні товари у своєму профілі.

Як можна побачити на загальному макеті, є можливість виділити такі компоненти:

- кнопка;
- кнопка з рамкою;

- кнопка другорядного значення;
- хедер;
- навігаційне меню (з перевіркою ролі користувача);
- пагінація;
- список замовлень для менеджера;
- елемент списку замовлень для менеджера;
- список замовлень для директора;
- елемент списку замовлень для директора;
- список продуктів для дизайнера;
- елемент списку продуктів для дизайнера;
- рівні складності дизайну;
- модальне вікно;
- картка відгуку (з перевіркою ролі користувача);
- фільтрація замовлень у директора;
- фільтрація замовлень у менеджера;
- фільтрація замовлень у менеджера;
- картка товару в замовленні у менеджера;
- панель з кнопками для редагування та видалення;
- форми:
 - редагування товару в замовленні;
 - створення товару в замовленні;
 - редагування відгуку;
 - додавання відгуку;
 - редагування замовлення;
 - створення замовлення;
 - редагування продукту у дизайнера;
- графік.

Для початку роботи над створенням програмної реалізації проекту, потрібно налаштувати середовище:

1. Створити папку для проекту;

2. Створити репозиторій на github[18];
3. Створюємо структуру файлів та папок;
4. Встановлюємо пакети за допомогою npm (пакетний менеджер для мови програмування JavaScript). Необхідно встановити react, react-dom, react-router, react-router-dom, react-icons, react-hook-form, sass, webpack, webpack-cli, webpack-dev-server, typescript, clean-webpack-plugin, copy-webpack-plugin, terser-webpack-plugin, html-webpack-plugin, mini-css-extract-plugin, optimize-css-assets-webpack-plugin, css-loader, postcss, postcss-loader, prettier, sass-loader, style-loader, ts-loader, а також пакети, що належать materialUI, redux, redux-saga, babel та типи з усіх основних модулів для Typescript;
5. Прописуємо конфігураційні файли для Webpack (пакувальник, з відкритим кодом, статичних модулів для сучасних додатків JavaScript)[25], Typescript[26], PostCSS[27], Babel[28].

На рис. 3.5 можна побачити структуру папок та файлів. Папка dist містить вихідні файли, згенеровані пакувальником, тобто готові до деплою файли.

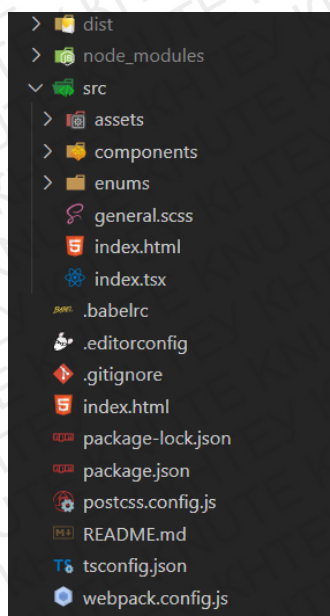


Рисунок. 3.5. Структура файлів та папок в проекті

Джерело: Авторська розробка

У папці `node_modules` зберігаються пакети, встановлені за допомогою пакетного менеджера `npm`.

Папка `src` містить усі ресурсні файли, тобто код компонентів, сторінок, їх стилізаційні файли, а також файли роботи з `redux` (бібліотека для роботи зі станом програм JavaScript) тощо.

Для створення сторінок потрібно спочатку створити компоненти, необхідні для відображення сторінок.

Приклад програмної реалізації одного з компонентів представлений на рисунку 3.6. Як можна побачити, кнопка є модулем, який експортується та до якого імпортуються інші модулі. В основі такого модулю лежить звичайна функція, яка приймає параметри, які в React називаються пропси. В залежності від пропсів, локального та глобального (`redux`) станів відбувається перемальовування компонентів.

```
1 import React from 'react'; 7.2K (gzipped: 2.9K)
2 import styles from './Button.module.scss';
3
4 interface IProps {
5   classes?: string,
6   btnText: string | JSX.Element,
7   onBtnClick: () => void
8 };
9
10 export const Button: React.FC<IProps> = ({ classes, btnText, onBtnClick }: IProps) => {
11   return (
12     <button
13       onClick={{(e) => {e.preventDefault(); onBtnClick()}}
14       className={` ${classes} ${styles.btn} `}
15     >
16       {btnText}
17     </button>
18   );
19 };
20
```

Рисунок.3.6. Код компоненту кнопка

Джерело: Авторська розробка

Команда розробників на чолі з Hai Nguyen [23], створити бібліотеку MaterialUI, яка дозволяє спростити адаптацію сторінок під планшети та мобільні пристрої, а також надає готові компоненти, наприклад, кнопка, роздільник, поле вводу, поле вибору, акордеон, картка, меню, посилання та

багато іншого. У нашому проєкті ми використали сітку, представлену MaterialUI.

Після того, як створені всі компоненти, можна переходити до створення сторінок. Сторінки розподіляються за ролями, тобто є сторінки доступні для всіх користувачів, лише для менеджерів, лише для дизайнерів та лише для директора.

Сторінки для всіх користувачів:

- Сторінка логіну;
- Сторінка з помилкою 403 (якщо користувач намагається перейти на сторінку, прав для перегляду якої не має);
- Сторінка з помилкою 404 (якщо користувач намагається перейти на сторінку, якої не існує).

Сторінки для менеджерів:

- Список замовлення;
- Створення замовлення;
- Редагування замовлення;
- Список відгуків.

Сторінки для дизайнерів:

- Сторінка із замовленнями, що потребують створення дизайну;
- Сторінка із замовленнями, які знаходяться в процесі виконання авторизованим дизайнером;
- Сторінка із виконаними замовленнями авторизованим дизайнером.

Сторінки директора:

- Статистика (з діаграмами);
- Список усіх замовлень;
- Список усіх дизайнерів;
- Список усіх відгуків.

Сторінки в реакті - це по суті ті ж самі компоненти, які відображаються при певному значенні `window.location.pathname`, тобто яка URL-адреса

знаходиться в адресному рядку. Перевірку адреси та відображення певного компоненту здійснює пакет react-router та react-router-dom.

Створення та редагування товару в замовленні менеджером, а також створення та редагування відгуку менеджером буде відбуватися не на окремій сторінці, а в модальному вікні.

Використання або виклик одних компонентів в інших виконується за допомогою синтаксису `<Component prop="prop_value" />` (рис. 3.7). Таким чином, за допомогою реакт компонентів можна швидко створювати сторінки.

```

24   return (
25     <div className={styles.app}>
26       <Header />
27     </div>
28   );

```

*Рисунок. 3.7. Використання одних компонентів в інших
Джерело: Авторська розробка*

3.2. Стилізація інтерфейсу веб-системи

Стилізація веб-сайту відбувається за допомогою спеціальної мови стилів сторінок CSS, що з'явилася в 1994 р[19]. Розробником мови CSS є World Wide Web Consortium. CSS дозволяє змінювати розміри, зовнішній вигляд елементів, їх розміщення, додавати анімацію та ефекти при наведенні курсору та багато іншого.

На рисунку 3.8 можна побачити CSS в дії. На першій частині рисунку відображено кнопку з дефолтними стилями браузера, а в другій частині зі стилями, прописаними в коді.



Рисунок. 3.8. Компонент кнопка:

а) до стилізації; б) після стилізації

Джерело: Авторська розробка

Також існують спеціальні програми, що мають власний синтаксис, який перетворюють на CSS. Такі програми називають CSS препроцесорами. Прикладами препроцесорів є SCSS, SASS, LESS, Stylus, PostCSS.

До особливостей та зручностей препроцесорів належать:

- синтаксис оголошення змінних коротше, ніж у чистого CSS;
- написання міксинів (mixin), тобто функцій, що дозволяють використовувати однаковий код у різних частинах додатку. Міксини також можуть приймати аргументи. Таким чином можна дотримуватися принципу написання коду DRY (don't repeat yourself - уникати повторення коду);
- препроцесори дозволяють писати наслідування селекторів зручно та швидко;
- імпорт файлів один в інший. Ця особливість дозволяє розділити стилі на декілька файлів, які значно легше підтримувати, ніж один величезний файл.

Для розробки веб-системи поліграфічного підприємства ми використали препроцесор SCSS[29].

Як можна побачити на рис. 3.9 для стилізації кнопки використано імпорт, наслідування селекторів та використання змінних.

```
1  @import './variables.scss';
2
3
4  .btn {
5      display: inline-block;
6      text-align: center;
7      vertical-align: middle;
8      background-color: $main_color;
9      color: #fff;
10     border: none;
11     border-radius: 33px;
12     padding: 13px 33px;
13     transition: all .5s;
14
15     &:hover {
16         cursor: pointer;
17         background-color: #700306;
18     }
19 }
```

Рисунок. 3.9. Стилї кнопки, написані за допомогою препроцесора SCSS

Джерело: Авторська розробка

Для стилізації компоненту Рівні складності дизайну використано міксини (рис.3.10).

```

1  @import '@mixins.scss';
2
3  .levels {
4  @include contentBlock;
5  }
6

```

Рисунок. 3.10. Стили компоненту Рівні складності дизайну

Джерело: Авторська розробка

На рис. 3.11 зображено написання змінних за допомогою SCSS. У свою чергу на рис. 3.9 в рядку 7 можна побачити використання змінної, із імпортованого в першому рядку файлу.

```

1  $main_color: #c40109;
2
3  $secondary_color: #4b4b4b;
4
5  $page_bg_color: #f1ecec;
6
7  $gradient: linear-gradient(135deg, #c40109 0%, #4b4b4b 100%);
8
9  $shadow: 4px 4px 10px 1px rgba(0, 0, 0, 0.25);
10

```

Рисунок. 3.11. Написання змінних SCSS

Джерело: Авторська розробка

За допомогою конструкції `@mixin mixin_name {}` оголошується міксин (рис. 3.11), а потім використовуються на допомогу конструкції `@include mixin_name` (рис. 3.10, рядок 4).

```

1  @mixin input {
2    border: 1px solid #a4a4a4;
3    border-radius: 5px;
4  }
5
6  @mixin contentBlock {
7    background: #fff;
8    box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.25);
9    border-radius: 10px;
10   padding: 10px;
11 }
12

```

Рисунок. 3.12. Написання міксинів SCSS

Джерело: Авторська розробка

Відповідно до психології кольору важливі елементи, що мають привертати увагу користувачів, повинні мати достатньо яскравий зовнішній

вигляд. А другорядні елементи повинні мати звичайний дизайн. Так наприклад кнопку видалення зазвичай роблять в сірих тонах і з дизайном, що не виділяється.

Крім зміни зовнішнього вигляду елементів, CSS має можливість правильно розміщувати елементи. Цю властивість мови стилізації можна продемонструвати на компоненті Хедер (рис. 3.13).

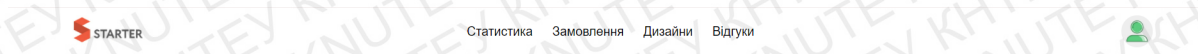


Рисунок. 3.13. Розміщення елементів за допомогою CSS

Джерело: Авторська розробка

При написанні коду компоненту Хедер було використано компонент Grid, наданий бібліотекою MaterialUI (рис. 3.14). Grid дозволяє розміщувати відповідну кількість елементів у рядку при певній ширині екрану.

```
8 export const Header: React.FC = () => {
9   return (
10    <header className={styles.header}>
11      <div className={styles.headerInner}>
12        <Grid container spacing={2} columns={{ xs: 3 }} alignItems='center'>
13          <Grid item xs={1}>
14            <img src={Logo} alt="Logo" className={styles.headerLogo} />
15          </Grid>
16          <Grid item xs={1}>
17            <NavMenu />
18          </Grid>
19          <Grid item xs={1}>
20            <img src={UserImg} alt="User" className={styles.headerUser} />
21          </Grid>
22        </Grid>
23      </div>
24    </header>
25  );
26};
```

Рисунок. 3.14. Код компоненту Хедер

Джерело: Авторська розробка

Використання сітки готової бібліотеки значно зменшує код стилів, що дозволяє легше підтримувати файл CSS (рис. 3.15). Якби довелося прописувати стилі для розміщення елементів, то кількість коду збільшилась більше ніж в 2 рази, оскільки крім правил розміщення за замовчуванням (для розташування на екрані шириною більше 1440 px), потрібно написати медіа запити зі стилями для екранів планшету та мобільних пристроїв.

```

1  .header {
2  background-color: #fff;
3  padding: 10px 0;
4
5  &Inner {
6  width: 90%;
7  margin: 0 auto;
8  }
9
10 &Logo {
11 width: 100px;
12 }
13
14 &User {
15 display: block;
16 width: 40px;
17 height: 40px;
18 object-fit: cover;
19 border-radius: 50%;
20 margin-left: auto;
21 }
22 }

```

Рисунок. 3.15. Стилі для компоненту Хедер

Джерело: Авторська розробка

3.3. Анімація інтерфейсу веб-системи

Анімація є незамінною частиною сучасних веб-сайтів. Вони дозволяють показати, що дія відбулася успішно чи з помилкою; що дія в процесі виконання (наприклад, ладер); скільки часу займає дія; впорядкувати дані та різні смислові блоки; акцентувати увагу користувача на ключових речах; надають відчуття “живої” взаємодії; дозволяють “пожвавити” нудний дизайн [24]. Усі ці переваги анімування інтерфейсу викликають емоції у користувача, що, як наслідок, сприяє приверненню уваги та підвищенню у веб-сайту здатності продавати.

Анімацію на веб-сайтах поділяють на [24]:

- мікроанімацію;
- великі анімації;
- анімація логотипу.

Мікроанімація - це зміна зовнішнього вигляду кнопок, посилань, іконок та інших клікабельних елементів. Прикладом мікроанімації є рух іконок, клік по яких призводить до певної дії, при наведенні курсору, кліку та після завершення дії.

Великі анімації - це анімована глобальна взаємодія усіх елементів сторінки. Такий тип анімації обирають для створення портфоліо, сайтів архітекторів, агентств з нерухомості, розробки дизайну (меблів, сайтів, одягу і тд.) та інших творчих компаній, осіб, агентств. Цей підхід дозволяє показати креативність, здатність мислити оригінально, не шаблонно.

Анімація логотипу приваблює увагу користувача та може розповісти історію, викликати емоцію за 1-3 секунди. Проте з використанням анімації логотипу треба бути обережними, оскільки такий тип анімації може відволікати користувача від виконання основних функцій на сайті.

Відповідно до опису типів анімації, можна дійти висновку, що для веб-системи поліграфічного підприємства потрібно використовувати лише мікроанімацію, оскільки працівники поліграфії мають бути максимально сконцентрованими над виконанням своїх професійних функцій. Але користувач нашої веб-системи повинен розуміти, що дія знаходиться в процесі виконання, що дія виконалася, а також анімація повинна викликати позитивні емоції у робітника, а не дратувати його.

У веб-дизайні існує поняття мікрвзаємодії - це короткий момент, коли юзер виконує одну дію через інтерфейс користувача інтерактивної системи, і при цьому система виконує одну конкретну функцію [20].

Прикладом мікрвзаємодії є анімація при наведенні курсору на елемент. Так у нашій системі ми змінюємо колір фону кнопки при наведенні курсору (рис. 3.16). Елемент, на який можна натиснути для реалізації певного функціоналу, повинен мати також зміну курсору-стрілки на курсор-руку, щоб користувач одразу розумів, що кнопка активна.

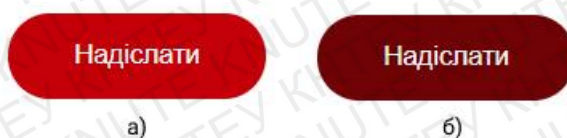


Рисунок. 3.16. Анімація кнопки при наведенні курсору

Джерело: Авторська розробка

При користуванні веб-системою робітник повинні розуміти, на якій сторінці він знаходиться у певний момент часу. Для цього використовується стилізація активного пункту меню (рис. 3.17). Пункт меню активної сторінки має зовнішній вигляд, як Статистика на рисунку 3.17. А зовнішній вигляд пункту меню при наведенні курсору виглядає, як Дизайни.

Статистика Замовлення Дизайни Відгуки

Рисунок. 3.17. Зовнішній вигляд активного пункту меню та пункту меню при наведенні курсору

Джерело: Авторська розробка

Під час виконання запитів користувач повинен розуміти, що дія знаходиться в процесі роботи і ще не завершилась. Функцію сповіщення користувача про виконання дії реалізує лоадер. Приклади візуалізації лоадерів представлено на рисунку 3.18.



Рисунок. 3.18. Приклади зовнішнього вигляду лоадерів

Джерело: Авторська розробка

Для веб-системи поліграфічного підприємства ми обрали простий лоадер, що рухається по колу (рис. 3.19). Такий лоадер показує, що дія відбувається і не відволікає користувача занадто сильно. Тобто користувач не забуває про свою мету, доки чекає завантаження. Головне забезпечити швидке виконання дії, щоб користувач не встиг отримати негативні емоції від користування додатком.



Рисунок. 3.19. Зовнішній вигляд лоадера, використаного для веб-системи

Джерело: Авторська розробка

Ще одним прикладом анімації в інтерфейсі є поява та зникнення модального вікна. Модальне вікно - це дуже ефективний елемент інтерфейсу, оскільки він максимально приваблює до себе увагу і концентрує її на собі. Поява модального вікна може здійснюватися по кліку на певний елемент (кнопка, іконка тощо), при скролі до певного рівня сторінки чи при закінченню певного проміжку часу. На нашу думку, для веб-системи поліграфічного підприємства варто використовувати лише появу модального вікна по кліку, оскільки інші типи збивають увагу робітника на виконанні своїх функцій, а також інші типи підходять лише для сайтів, які щось продають.

При появі модального вікна простір навколо нього затемнюється, розмивається чи висвітлюється.

Як можна побачити на рисунку 3.20 у веб-системі поліграфічного підприємства простір навколо модального вікна затемнюється. На модальному вікні за замовчуванням присутній хрестик, який закриває вікно. Контент у модальному вікні може бути зміненим відповідно до потреб дизайну. У випадку з веб-системою в модальних вікнах знаходяться форми створення та редагування товару в замовленні та створення і редагування відгуку.

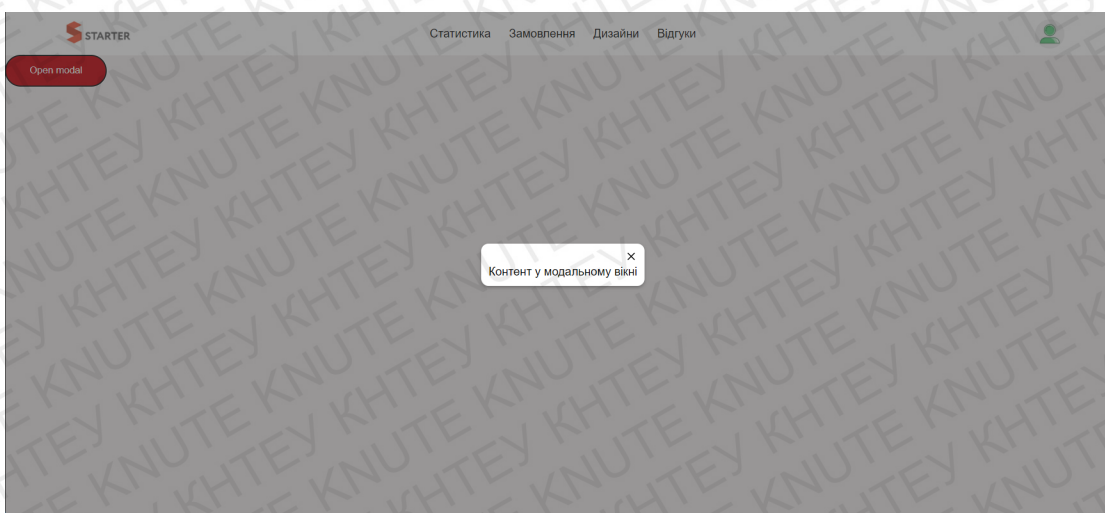


Рисунок. 3.20. Зовнішній вигляд модального вікна у веб-системі

Джерело: Авторська розробка

Для того, щоб користувач зрозумів, що дія завершилась можна використати перетворення кнопки на зелену галочку (при успішному завершенні) та на червоний хрестик (при завершенні з помилкою), появу тостеру та ін.

Для веб-системи було вирішено використати тостери для сповіщення користувача. Їх зовнішній вигляд можна побачити на рисунку 3.21. Для реалізації тостеру не потрібно писати кастомний код, оскільки вже створено спеціальну бібліотеку React Hot Toast[30]. Достатньо лише в потрібному місці розмістити компонент тостеру та функцію, що його викликає. А також потрібно правильно сконфігурувати компонент.

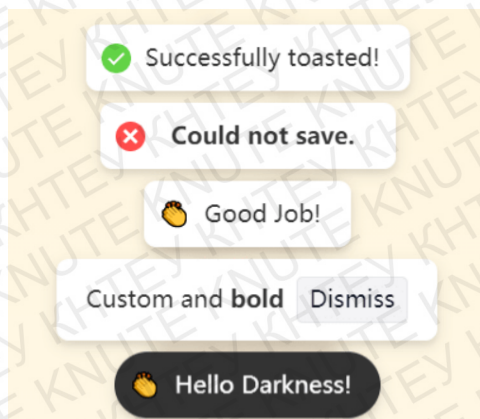


Рисунок. 3.21. Приклади зовнішнього вигляду тостерів, створених за допомогою модулю react-hot-toast
Джерело: Авторська розробка

3.4. Зв'язок з бекендом веб-системи

У додатку для поліграфічного підприємства бекенд реалізований у вигляді Rest API. REST API описує набір ресурсів і набір операцій, які можна викликати на цих ресурсах. Операції в REST API можна викликати з будь-якого клієнта HTTP, включаючи код JavaScript на стороні клієнта, який працює у веб-браузері[21].

Для отримання даних з бекенду потрібно використовувати поєднання таких технологій, як fetch API (JavaScript), redux та redux saga.

Щоб повністю зрозуміти процес зв'язку фронтенду з бекендом потрібно розуміти поняття зазначені нижче [22].

Redux State - це стан додатку, який глобально доступний для всіх компонентів додатку.

Dispatch Action - виклик дії redux. Цей метод співставляє тип дії з ред'юсером та мідлваром і виконує відповідних код.

Middleware - це технологія, що забезпечує сторонню точку розширення між відправленням дії та моментом, коли вона досягає ред'юсера.

Reducer - це функція, що приймає поточний стан та дію в якості аргументів, а повертає оновлений стан.

Для виконання запиту на бекенд необхідно виконати наступні кроки:

1. Діспатч певної дії (викликається або з компоненту, або з мідлвару.);
2. Мідлвар перехоплює дію, тому в ньому потрібно прописати запит на серверну частину, а потім виконання дії на збереження отриманих даних у redux store.

Після того, як дані с серверу отримані, їх можна зчитувати в компоненті і відмальовувати компонент відповідно до отриманих даних.

Розгляд теоретичної інформації дозволяє перейти до практичної реалізації.

Є декілька способів здійснити запит на сервер: fetch API, axios, XMLHttpRequest та інші. Для нашої веб-системи ми обрали fetch API (рис. 3.22).

Запити на сервер можна робити безпосередньо з компоненту, а можна використовувати Redux для збереження інформації з сервера. Саме другий варіант було обрано для веб-системи поліграфічного підприємства.

```

1 import { ILoginRes } from '@/interfaces';
2 import { ALL_URL } from './apiConfig';
3
4 export const auth = async (loginInfo: ILoginRes) => {
5   const { login, password } = loginInfo;
6   console.log(loginInfo)
7
8   const response = await fetch(`${ALL_URL}/auth`, {
9     method: 'POST',
10    headers: {
11      'Content-Type': 'application/json;charset=utf-8',
12    },
13    body: JSON.stringify({
14      login,
15      password,
16    })
17  });
18  const result = await response.json();
19
20  return result;
21 };
22
23 export const getUser = async (userId: string) => {
24   const response = await fetch(`${ALL_URL}/user/${userId}`);
25   const result = await response.json();
26
27   return result;
28 };

```

Рисунок. 3.22. Приклади запитів з клієнта на сервер (логін та отримання даних про клієнта по ID)

Джерело: Авторська розробка

Розглянемо код Reducer та Saga на прикладі логіну користувача. На рисунку 3.23 можна розглянути Reducer.

```

6 interface IInitialState {
7   user: IUser | null,
8   isLogin: boolean,
9   loginStatus: IStatus,
10  getUserStatus: IStatus,
11 };
12
13 const initialState: IInitialState = {
14   user: null,
15   isLogin: false,
16   loginStatus: helpers.getDefaultState(),
17   getUserStatus: helpers.getDefaultState(),
18 };
19
20 export const userReducer: Reducer = (state = initialState, action: AnyAction) => {
21   switch(action.type) {
22     case types.LOGIN:
23       return {
24         ...state,
25         loginStatus: helpers.getRequestState()
26       };
27
28     case types.LOGIN_SUCCESS:
29       return {
30         ...state,
31         loginStatus: helpers.getSuccessState('You are successfully login'),
32         isLogin: true
33       };
34
35     case types.LOGIN_FAILURE:
36       return {
37         ...state,
38         loginStatus: helpers.getErrorState('Something went wrong. Try again.')
39       };
40
41     case types.CLEAR_LOGIN_STATUS:
42       return {
43         ...state,
44         loginStatus: helpers.getDefaultState()
45       };
46

```

Рисунок. 3.23. Reducer для аутентифікації користувача

Джерело: Авторська розробка

На рисунку 3.24 представлено Saga для аутентифікації користувача. Як можна побачити на цьому рисунку, після успішної аутентифікації токен та id користувача зберігаються в Localstorage браузера для того, щоб при перезавантаженні сторінки користувач мав доступ до дозволених йому сторінок.

```

22 export function* login({ type, loginInfo }: ILogin) {
23   try {
24     const response: ILoginReq = yield call(api.auth, loginInfo);
25     console.log(response)
26     yield put(actions.loginSuccess());
27
28     const { token, id } = response;
29
30     token && localStorage.setItem('token', token);
31     id && localStorage.setItem('userId', id);
32     yield put(actions.getUser(id));
33   } catch (error) {
34     yield put(actions.loginFailure());
35   };
36 };

```

Рисунок. 3.24. Saga для аутентифікації користувача

Джерело: Авторська розробка

У компоненті логіну необхідно отримати інформацію про користувача, що аутентифікувався, щоб здійснити редірект на сторінку відповідно до ролі юзера. Код, що необхідно написати для отримання інформації з Redux Store представлено на рисунку 3.25.

```

11 export const Login: React.FC = () => {
12   const dispatch = useDispatch();
13   const user = useSelector((state: RootState) => state.userReducer.user);

```

Рисунок. 3.25. Отримання інформації про користувача із Redux Store

Джерело: Авторська розробка

Таким чином, після здійснення логіну компонент перевіряє роль користувача і рендерить сторінку замовлень для менеджера; товарів, що потребують розробку макета для дизайнера; статистику для директора.

Висновки до розділу 3

Отже, аналізуючи вище зазначену інформацію, можна дійти висновку, що розробка графічного інтерфейсу складається з наступних етапів:

- відмальовування макету в графічному редакторі;

- верстка компонентів макету;
- стилізація компонентів макету;
- додавання анімації до компонентів;
- написання redux flow та запитів на бекенд.

Реалізувавши кожен з кроків розробки графічного інтерфейсу, вказаних вище, було розроблено веб-додаток для поліграфічного підприємства зі зручним та зрозумілим управлінням. Поля, що можуть викликати сумніви у користувача, мають опис. Тобто користувач має пояснення, що йому треба ввести в те чи інше поле.

Інтерфейс розроблено без зайвих дизайнерських деталей, які могли б відволікати користувачів веб-системи. А також важливі елементи (кнопки, активні пункти меню, фільтрація, активні сторінки пагінації тощо) виділені кольором, підкресленням чи товщиною тексту.

ВИСНОВКИ

Отже, високий рівень конкуренції на ринку та розвиток інформаційних технологій провокують розробку зручного і зрозумілого інтерфейсу для додатку. Необхідно велику увагу приділяти психології сприйняття інформації, або гештальт-принципам в дизайні, а також психології кольору (поєднання кольорів між собою та сприйняття кольору людським мозком).

Оскільки поліграфічне підприємство STARTER має брендбук, у якому зазначені корпоративні кольори, то основними кольорами для додатку були обрані саме корпоративні.

У роботі досліджено поняття, особливості та види інтерфейсів сучасних веб-систем, а також проаналізовано існуючі інтерфейси веб-систем поліграфічного підприємства.

Для того, щоб побудувати ефективний інтерфейс без надлишкового функціоналу, потрібно розуміти специфіку роботи підприємства та бізнес-процеси, що лежать в основі його діяльності.

Одним з найбільш ефективних розділів у менеджменті є управління бізнес-процесів підприємства. Цей спосіб управління має вагомі переваги: можливість у короткі терміни масштабувати підприємство, вносити зміни у бізнес-процеси, аналізувати ефективність роботи підприємства, на основі аналізу покращувати економічні показники підприємства.

Поліграфічне підприємство має 5 основних бізнес-процесів: процес прийому замовлення, процес створення макету, процес передачі замовлення на друк підрядникам, процес передачі замовлення клієнту, процес збору відгуків від клієнта.

Організаційна структура підприємства складається з директора на верхівці структури, дизайнерів та менеджерів. Кожен з робітників має свою роль у системі бізнес-процесів.

Для наглядності схеми бізнес-процесів відмальовано за допомогою спеціалізованих додатків. У цій роботі було використано сервіс bpmn.io.

Досить ефективно роздруковувати схему і розміщувати її біля робочого місця директора. Таким чином, якщо на підприємстві виникла проблема, то директор одразу зможе зрозуміти, у якому саме місці виник пролом, і зможе оперативно відреагувати на проблему.

Базуючись на гештальт-принципах, було розроблено макет графічного інтерфейсу, використовуючи графічний редактор Figma. На основі макету зверстано, стилізовано та анімовано компоненти. А також здійснено інтеграцію графічного інтерфейсу з бекендом цієї веб-системи.

Оскільки бекенд побудовано за допомогою архітектури Rest API, то написано запити, використовуючи CRUD-операції. Інформацію з бази даних, що надходить з бекенду, збережено в Redux Store веб-системи. Компоненти веб-системи отримують з Redux Store дані і перемальовуються, якщо в Store оновлюється.

Підсумуємо технології, що були використані для розробки інтерфейсу.

- візуалізація макету: Figma;
- верстка компонентів: React, TypeScript, MaterialUI;
- стилізація компонентів: SCSS, MaterialUI;
- анімація компонентів: SCSS;
- зв'язок з бекендом: Redux, Redux Saga;
- допоміжні технології: Webpack, Babel, React Icons, React Hook Form, React Hot Toast.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мосіюк О.О. UX проектування сайтів як основа підготовки спеціалістів Web-дизайну. *Педагогічні науки*. 2017. № 3. URL: <https://pedagogy.bdpu.org/wp-content/uploads/2018/01/33-2.pdf> (дата звернення: 11.02.2021)
2. Компанеєтс М.О. Принципи проектування ефективних веб-сайтів. *Молодий вчений*. 2015. № 9 (24) частина 2. URL: <http://molodyvcheny.in.ua/files/journal/2015/9/66.pdf> (дата звернення: 11.02.2021)
3. Матвейко І. І. Веб-сайт бібліотеки університету - важливий інструмент в системі обслуговування користувачі. *Матеріали шостої міжнародної науково-практичної конференції. Національний університет «Львівська політехніка»*. 2015. URL: <http://ena.lp.edu.ua:8080/bitstream/ntb/33690/1/42-221-226.pdf> (дата звернення: 12.02.2021)
4. Купрунець Т.Я. Тенденції та перспективи розвитку сайтобудування, їх вплив на архівне копіювання. *Статті та повідомлення*. 2013. URL: <https://archives.gov.ua/wp-content/uploads/09-18.pdf> (дата звернення: 12.02.2021)
5. Letizia Bollini. Beautiful interfaces. From user experience to user interface design. *The Design Journal. An International Journal for All Aspects of Design*. 2017. Vol. 20 P. S89-S101. DOI: 10.1080/14606925.2017.1352649. URL: <https://www.tandfonline.com/doi/pdf/10.1080/14606925.2017.1352649?needAccess=true&> (дата звернення: 17.02.2021)
6. Liqiong Deng, Marshall Scott Poole. Poole Affect in Web Interfaces: A Study of the Impacts of Web Page Visual Complexity and Order. *MIS Quarterly*. 2010. Vol. 34, No. 4. P. 711-730. DOI:10.2307/25750702
7. Oyanda T Kona. Website interface design: a study on the status quo of South African e-Commerce website interfaces. *Working Paper, Cape Peninsula*

- University of Technology, Cape Town.* 2014. URL: https://www.researchgate.net/publication/322499765_Website_interface_design_a_study_on_the_status_quo_of_South_African_e-Commerce_website_interfaces_0069 (дата звернення: 17.02.2021)
8. What is User Interface Design? *Interaction Design Foundation*: веб-сайт. URL: <https://www.interaction-design.org/literature/topics/ui-design> (дата звернення: 03.03.2021)
9. Exploring the UI Universe: Different Types of UI. *Altia: Concept to Production UIs*: веб-сайт. URL: <https://www.altia.com/2014/09/22/different-types-of-ui/> (дата звернення: 15.03.2021)
10. What are Voice User Interfaces? *Interaction Design Foundation*: веб-сайт. URL: <https://www.interaction-design.org/literature/topics/voice-user-interfaces> (дата звернення: 16.03.2021)
11. The Future of Gesture-Based UI. *Altia: Concept to Production UIs*: веб-сайт. URL: <https://www.altia.com/2014/10/16/the-future-of-gesture-based-ui/> (дата звернення: 16.03.2021)
12. Gestalt principles in UI design. *Medium*: веб-сайт. URL: <https://medium.muz.li/gestalt-principles-in-ui-design-6b75a41e9965> (дата звернення: 01.04.2021)
13. The Extensive Guide to Business Processes. *Kissflow Process*: веб-сайт. URL: <https://kissflow.com/workflow/bpm/business-process/> (дата звернення: 15.06.2021)
14. What is Business Process? – Types, Benefits, and Examples. *Fasproc*: веб-сайт. URL: <https://fasproc.com/blog/business-process/> (дата звернення: 12.07.2021)
15. Web-based tooling for BPMN, DMN and Forms. *BPMN*: веб-сайт. URL: <https://bpmn.io/>
16. BPMN specification. *BPMN*: веб-сайт. URL: <https://www.bpmn.org/>

17. <https://www.figma.com/file/v3MolRhsz9Y5gr9uxoCsQ7/Polygraphy-PB?no-de-id=0%3A1>
18. <https://github.com/AlionaN/Polygraphy>
19. A brief history of CSS until 2016. *W3*: веб-сайт. URL: <https://www.w3.org/Style/CSS20/history.html> (дата звернення: 16.09.2021)
20. Omar Sosa-Tzec, Erik Stolterman Bergqvist. Delight by Motion: Investigating the Role of Animation in Microinteractions. *MODE: Motion Design Education Summit*. 2021. URL: http://www.modesummit.com/2021/wp-content/uploads/2021/06/Sosa-Tzec_Stolterman_MODE2021-1.pdf (дата звернення: 02.10.2021)
21. Vijay Surwase. REST API Modeling Languages - A Developer's Perspective. *IJSTE - International Journal of Science Technology & Engineering*. 2016. Vol. 2. №10. P. 634-637. URL: <http://www.ijste.org/articles/IJSTEV2I10199.pdf> (дата звернення: 04.10.2021)
22. Redux Documentation. *Redux*: веб-сайт. URL: <https://redux.js.org/> (дата звернення: 03.10.2021)
23. MaterialUI Developing Team. *MaterialUI*: веб-сайт. URL: <https://v1.mui.com/discover-more/team/> (дата звернення: 05.10.2021)
24. Анімації як засіб UI дизайну. *Evergreen*: веб-сайт. URL: <https://evergreens.com.ua/ua/articles/animations.html> (дата звернення: 05.10.2021)
25. Webpack Documentation. *Webpack*: веб-сайт. URL: <https://webpack.js.org/> (дата звернення: 03.10.2021)
26. Typescript Documentation. *Typescript*: веб-сайт. URL: <https://www.typescriptlang.org/> (дата звернення: 03.10.2021)
27. PostCSS Documentation. *PostCSS*: веб-сайт. URL: <https://postcss.org/> (дата звернення: 03.10.2021)
28. Babel Documentation. *Babel*: веб-сайт. URL: <https://babeljs.io/> (дата звернення: 03.10.2021)

29. SCSS Documentation. SCSS: веб-сайт. URL: <https://sass-lang.com/> (дата звернення: 03.10.2021)
30. React Hot Toast Documentation. *React Hot Toast*: веб-сайт. URL: <https://react-hot-toast.com/> (дата звернення: 03.10.2021)

ДОДАТКИ

Додаток А

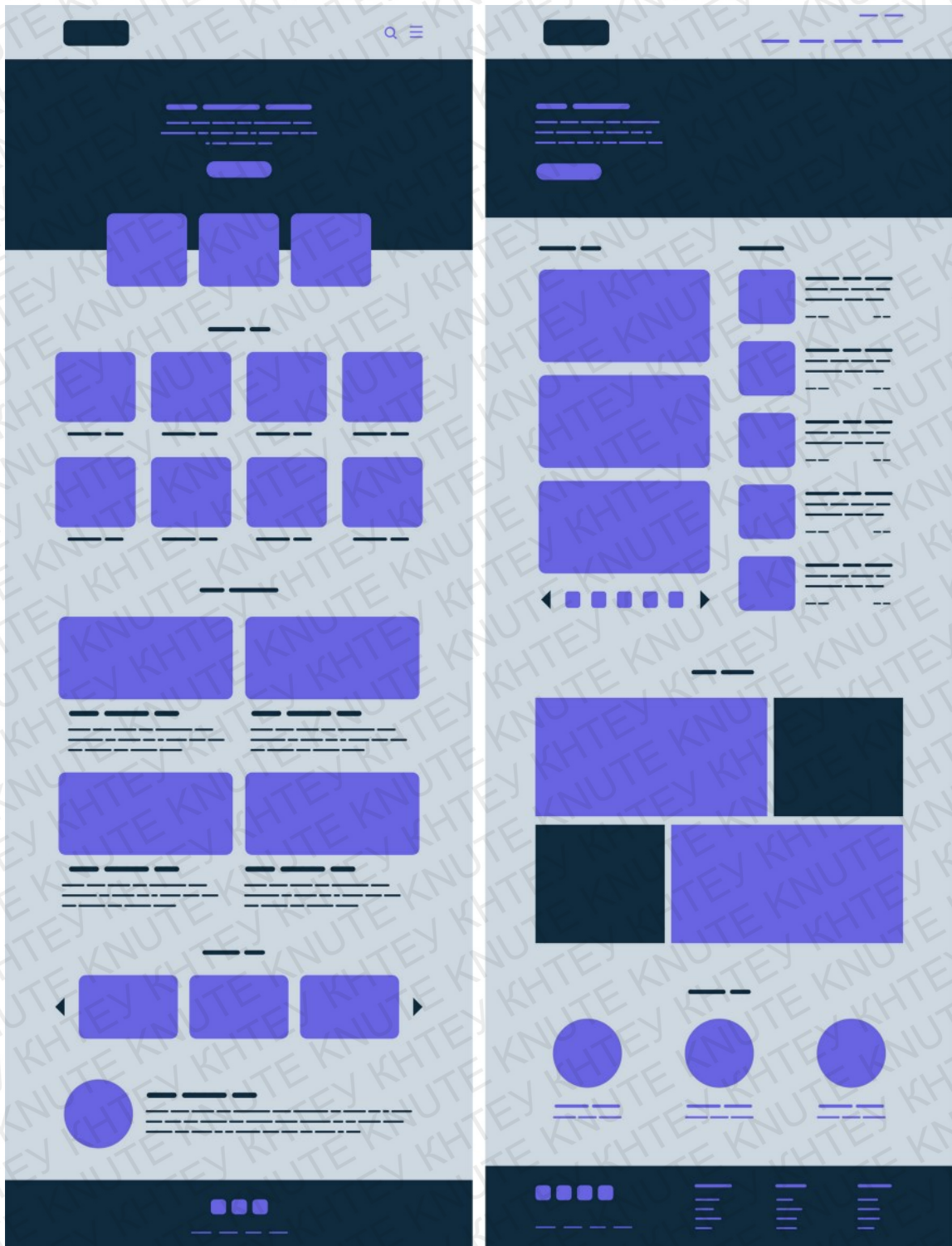


Рис. А.1. Приклад використання гештальт-принципу Близькість при створенні UI [12]

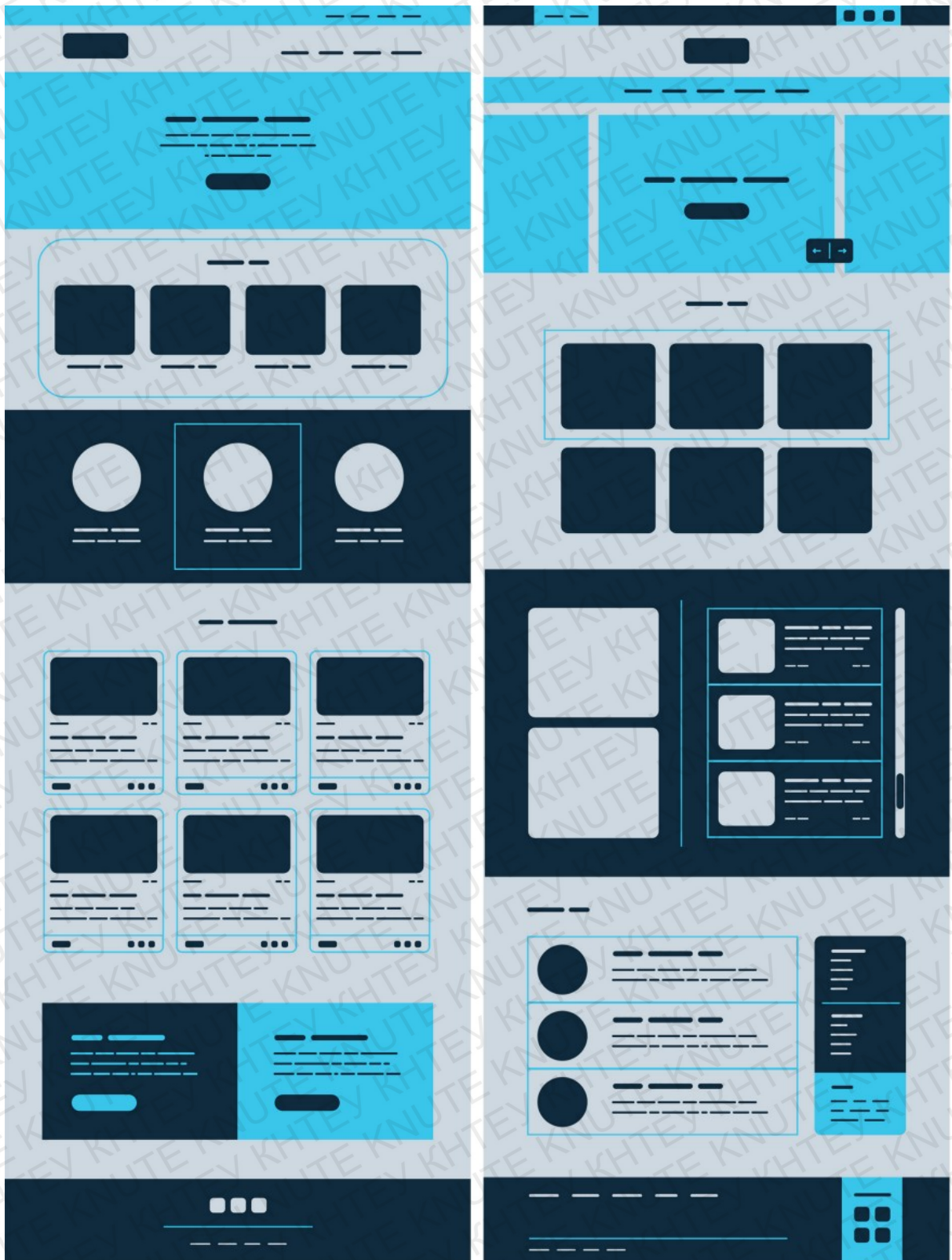


Рис. Б.1. Приклад використання геішталт-принципу Спільна область при побудові UI [12]

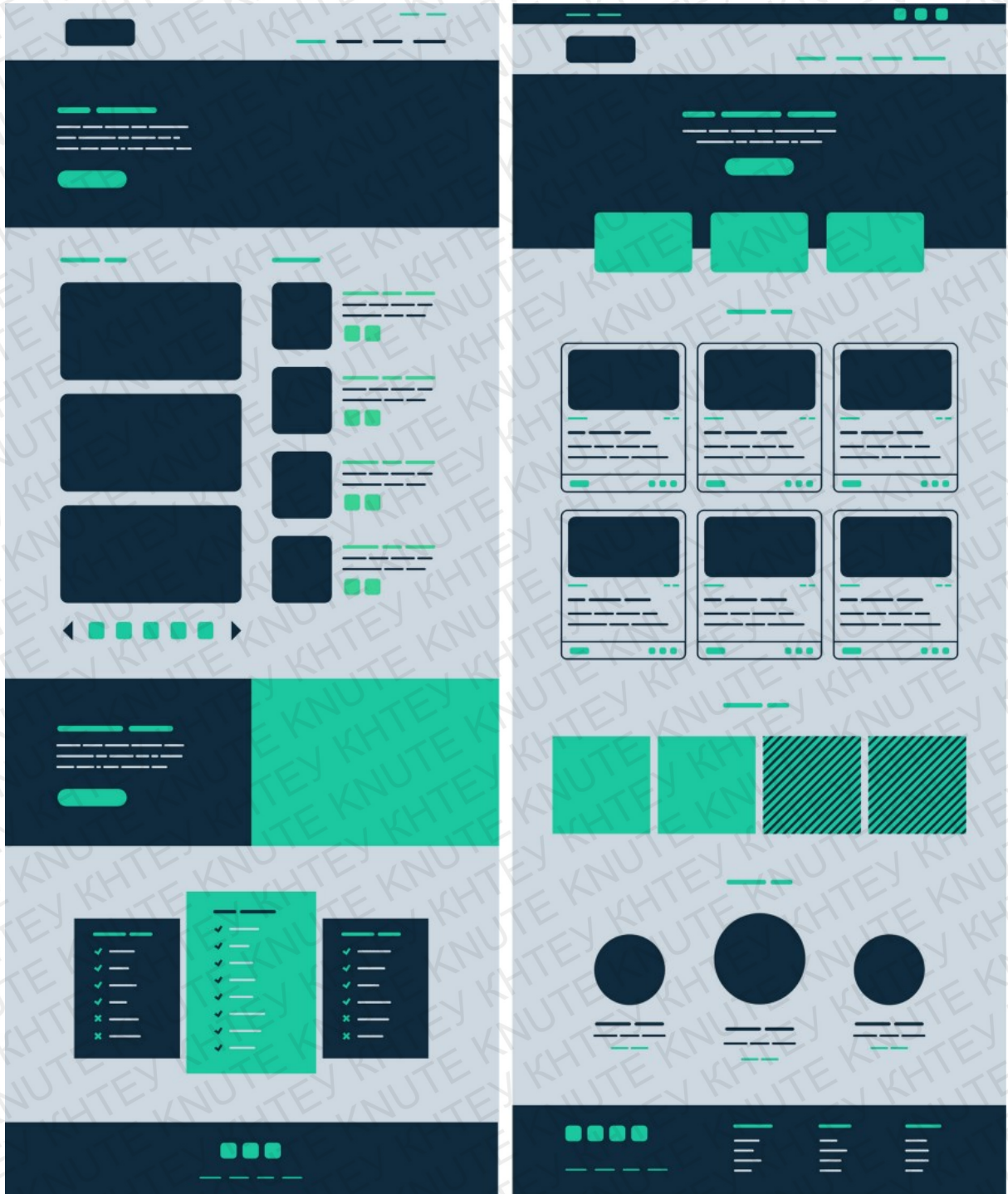


Рис. В.1. Приклад використання гештальт-принципу Схожість при побудові UI [12]



Рис. Г.1. Приклад використання гештальт-принципу Замкнутість при побудові UI [12]

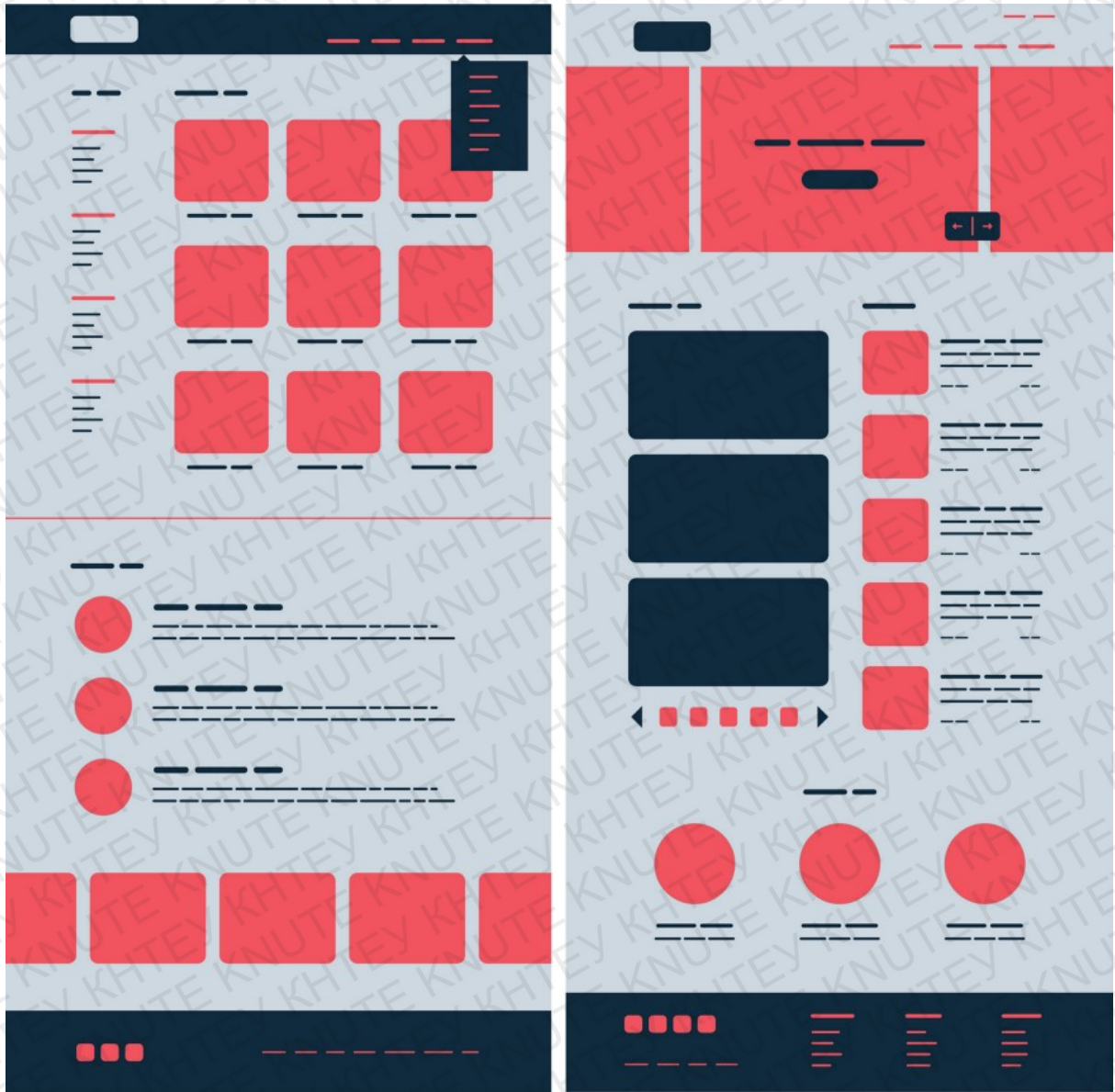


Рис. Д.1. Приклад використання гештальт-принципу Продовження при побудові UI [12]



Рис. Е.1. Приклад використання гештальт-принципу Спільної долі при побудові UI [12]