

***БАКАЛАВРСЬКА
РОБОТА***

БР.ІІ – 46.00.00.000 ІЗ

Група ІІ-22-2

Семенців Михайло

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Семенців Михайло Дмитрович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

БАКАЛАВРСЬКА РОБОТА

Принципи створення інтерфейсів користувача в програмному забезпеченні

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Семенців М.Д.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Михайлюк Ірина Романівна, доцент, к.п.н.
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доцент.

В.В. Бандура

“ ” 2026 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Семенців Михайлу Дмитровичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) "Принципи створення інтерфейсів користувача в програмному забезпеченні"

керівник проекту (роботи) доц.ент, к.т.н., Михайлюк І.Р.

затверджені наказом вищого навчального закладу від “ 21 ” квітня 2026 р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз вимог до програмного забезпечення

2. Аналіз вимог і постановка задачі

3. Проектування системи

4. Реалізація проекту

5. Тестування та впровадження

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

1 Галузі, що мають значення для дослідження (рис. 2.1, ст. 34)

2 Приклад простої діаграми активності (рис. 2.2, ст. 43)

3 Діаграма варіантів використання в UML 2.0 (рис. 2.4, ст. 45)

4 Розширена діаграма варіантів використання із загальними інтерфейсами (рис. 2.5, ст. 46)

5 Принципи створення інтерфейсів користувача (рис. 3.1, ст. 59)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання _____ 2026 р.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	17.03.2026	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	21.04.2026	виконано
3	Побудова моделі або алгоритму власного рішення	01.05.2026	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	21.05.2026	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	10.06.2026	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 97 сторінок, 28 рисунків, 2 таблиці, список використаних джерел із 40 найменувань.

Метою роботи є дослідження принципів створення інтерфейсів користувача в програмному забезпеченні, а також аналіз методів і підходів до проєктування зручних, ефективних і інтуїтивно зрозумілих інтерфейсів.

Об'єкт дослідження: процеси проєктування та розробки інтерфейсів користувача в програмному забезпеченні.

Предмет дослідження: принципи, методи та інструменти створення інтерфейсів користувача, а також підходи до оцінки їх якості та зручності використання.

Результати дослідження: проведено аналіз основних принципів UX/UI-дизайну, досліджено критерії оцінювання якості інтерфейсів, розглянуто сучасні інструменти та підходи до їх проєктування.

У першому розділі розглянуто теоретичні основи створення інтерфейсів користувача, визначено основні поняття, класифікацію інтерфейсів та критерії їх оцінювання.

У другому розділі досліджено методи та інструменти проєктування інтерфейсів, зокрема принципи взаємодії з користувачем, забезпечення зворотного зв'язку та аналіз сценаріїв використання.

У третьому розділі розглянуто практичну реалізацію інтерфейсу користувача, включаючи застосування стандартів якості, проєктування графічного інтерфейсу та оцінку його відповідності принципам UX/UI.

Висновок: застосування сучасних принципів проєктування інтерфейсів користувача дозволяє підвищити зручність використання програмного забезпечення, покращити взаємодію користувача із системою.

КЛЮЧОВІ СЛОВА: ІНТЕРФЕЙС КОРИСТУВАЧА, UX/UI, ЮЗАБІЛІТІ, ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ, ГРАФІЧНИЙ ІНТЕРФЕЙС, КОРИСТУВАЦЬКИЙ ДОСВІД.

ANNOTATION

The bachelor's thesis contains of 97 pages, 28 figures, 2 tables, and a reference list containing 40 sources.

The aim of the work is to study the principles of user interface design in software and to analyze methods and approaches for creating user-friendly, efficient, and intuitive interfaces.

Research object: processes of designing and developing user interfaces in software systems.

Research subject: principles, methods, and tools for user interface design, as well as approaches to evaluating their quality and usability.

Research results: the main principles of UX/UI design were analyzed, criteria for interface evaluation were studied, modern tools and approaches to interface design were reviewed, and the effectiveness of implemented solutions was evaluated.

The first section describes the theoretical foundations of user interface design, defines key concepts, interface classifications, and evaluation criteria.

The second section examines methods and tools for interface design, including user interaction principles, feedback mechanisms, and analysis of usage scenarios.

The third section presents the practical implementation of a user interface, including the application of quality standards, graphical interface design, and evaluation of compliance with UX/UI principles.

Conclusion: the application of modern user interface design principles improves software usability, enhances user interaction, and increases the efficiency of task execution.

KEY WORDS: USER INTERFACE, UX/UI, USABILITY, INTERFACE DESIGN, GRAPHICAL USER INTERFACE, USER EXPERIENCE.

ЗМІСТ

ВСТУП	8
--------------------	----------

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА	10
---	-----------

1.1 Основні поняття та класифікація інтерфейсів користувача	10
1.2 Критерії оцінювання якості та зручності інтерфейсів (UX/UI)	16
1.3 Методи дослідження та проєктування інтерфейсів користувача	25
1.4 Висновки по розділу	30

РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІНТЕРФЕЙСІВ ..	31
--	-----------

2.1 Основні принципи та підходи до проєктування інтерфейсів	31
2.2 Забезпечення зворотного зв'язку та взаємодії з користувачем	36
2.3 Аналіз сценаріїв використання та поведінки користувачів	42
2.4 Висновки по розділу	48

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЯКОСТІ ІНТЕРФЕЙСІВ	49
--	-----------

3.1 Стандарти якості програмного забезпечення та інтерфейсів	49
3.2 Проєктування та реалізація графічного інтерфейсу користувача	55
3.3 Оцінка відповідності інтерфейсу загальним принципам UX/UI	63
3.4 Висновки по розділу	92

ВИСНОВКИ	93
-----------------------	-----------

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	95
---	-----------

					БР.ІІІ – 46.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Принципи створення інтерфейсів користувача в програмному забезпеченні	Літ.	Арк.	Акрушів
Розроб.		Семенців М.Д.						
Перевір.		Михайлюк І.Р.					7	
Реценз.		Пасека Н.М.				ІФНТУНГ ІІІ-22-2		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В. В.			Пояснювальна записка			

ВСТУП

У сучасному світі інформаційні технології є невід’ємною складовою розвитку суспільства, а програмне забезпечення використовується у всіх сферах діяльності людини. У зв’язку з цим особливої ваги набуває не лише функціональність програмних систем, але й зручність їх використання. Інтерфейс користувача виступає ключовим елементом взаємодії між людиною та програмним продуктом, визначаючи ефективність виконання завдань, рівень задоволеності користувачів і загальну якість програмного забезпечення.

Актуальність теми зумовлена необхідністю створення інтерфейсів користувача, які відповідають сучасним вимогам юзабіліті, доступності та естетичності. Зі зростанням складності програмних систем та різноманітності платформ зростає потреба у застосуванні системного підходу до проєктування інтерфейсів. Недостатня увага до принципів UX/UI може призвести до зниження продуктивності користувачів, збільшення кількості помилок та негативного досвіду взаємодії з системою. Тому дослідження принципів створення інтерфейсів користувача є важливим завданням у галузі розробки програмного забезпечення.

Метою роботи є дослідження принципів створення інтерфейсів користувача в програмному забезпеченні, а також аналіз методів і підходів до проєктування зручних, ефективних і інтуїтивно зрозумілих інтерфейсів.

Завданнями дослідження є аналіз теоретичних основ створення інтерфейсів користувача, визначення їх основних характеристик і класифікації, дослідження критеріїв оцінювання якості інтерфейсів, аналіз методів і інструментів проєктування UX/UI, вивчення принципів взаємодії користувача із системою, реалізація графічного інтерфейсу користувача, а також оцінка його відповідності сучасним стандартам якості.

Об’єктом дослідження є процеси проєктування та розробки інтерфейсів користувача в програмному забезпеченні, що включають аналіз вимог, проєктування, реалізацію та оцінювання якості інтерфейсів.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Предметом дослідження є принципи, методи та інструменти створення інтерфейсів користувача, включаючи забезпечення зручності використання, ефективності взаємодії та відповідності сучасним стандартам UX/UI.

Методи дослідження включають аналіз наукових джерел для вивчення теоретичних основ проєктування інтерфейсів, порівняльний аналіз існуючих підходів і інструментів, моделювання структури інтерфейсу, експериментальний метод для реалізації та тестування інтерфейсу, а також оцінювання результатів із використанням якісних і кількісних показників.

У роботі передбачається дослідження теоретичних основ UX/UI-дизайну, аналіз сучасних підходів до створення інтерфейсів, а також практична реалізація графічного інтерфейсу користувача з урахуванням встановлених стандартів якості. Особлива увага приділяється забезпеченню зручності використання, ефективності взаємодії та адаптивності інтерфейсу до потреб користувачів.

Бакалаврська робота містить 98 сторінок, 28 рисунків, 2 таблиці, 3 розділи, список використаних джерел із 40 найменувань.

					БР.ІІ – 46.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА

1.1 Основні поняття та класифікація інтерфейсів користувача

Дизайн інтерфейсу користувача стає ітеративним, оскільки після завершення початкового дизайну дизайнер зазвичай запитує: чи можна його покращити? І звичайна відповідь — «так». Дизайн програмного забезпечення, особливо дизайн інтерфейсу користувача, є природно ітеративним, оскільки програмне забезпечення дозволяє вносити зміни до тих пір, поки інженери хочуть вносити зміни для покращень або експериментів. Після створення початкового дизайну дизайн програмного забезпечення продовжує розвиватися через ітерації, експерименти з прототипами або поступову розробку. Ми навряд чи багато дізнаємося про дизайн програмного забезпечення з дизайну фізичних систем, таких як будівлі. «Оскільки програмне забезпечення настільки гнучке, дизайн програмного забезпечення — це безперервний процес, який охоплює весь життєвий цикл програмної системи; це відрізняє дизайн програмного забезпечення від дизайну фізичних систем, таких як будівлі, кораблі чи мости» [1]. Дизайн інтерфейсу користувача є однією з найскладніших галузей програмної інженерії. Завдання створення інноваційних інтерфейсів користувача часто вважаються «поза межами досяжності» звичайних розробників програмного забезпечення, особливо порівняно з неодноразовими досягненнями Стіва Джобса та Джонатана Айва з Apple, Inc. Створити чудовий дизайн непросто [2]. Великі дизайнери програмного забезпечення не багато писали про свої інноваційні підходи до дизайну. Це одна з труднощів у розумінні та відтворенні чудових методів проектування [3]. Існують й інші труднощі у вивченні проектування програмного забезпечення, особливо для складних програмних систем [4]. Складність програмного забезпечення є складною, оскільки «неможливо візуалізувати проект великої програмної системи достатньо добре, щоб зрозуміти всі її реалізації, перш ніж щось створювати» [5].

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Початковий проект програмного забезпечення, можливо, доведеться переглядати ітеративно після початкової фази розробки, коли стане очевидним краще розуміння складності системи. Початкові користувацькі інтерфейси системи можуть відігравати дуже важливу конструктивну роль у процесі формування.

Сьогодні ми стикаємося з інтерфейсами користувача майже на кожному електричному пристрої, такому як мобільні телефони, ноутбуки, смартфони та настільні комп'ютери. Але також пральні машини та мікрохвильові печі використовують своєрідний інтерфейс користувача для отримання вхідних даних від користувача. Також у нашому автомобілі ми стикаємося з інтерфейсом користувача, наприклад, використовуючи нашу навігаційну систему або навіть під час використання нашого радіо. Розробка *ефективного* інтерфейсу користувача також важлива з точки зору питань безпеки. Сучасні літаки, такі як, наприклад, новий Airbus A380, мають складне технічне обладнання, яке потребує повної уваги пілота. Тут неправильний дизайн може призвести не лише до розчарування, але й до аварій і навіть до смерті.

Зростання комп'ютерів ще не зупинилося, і вони, здається, є нормальною частиною нашого життя, особливо в галузі мобільного зв'язку та використання мобільного Інтернету. Також з так званими платформами соціальних мереж Інтернет стає все більш важливим для більшої кількості груп людей. У цьому процесі розвитку також слід враховувати людей, які мали менше контактів у минулому з інформаційними технологіями. Місія тут полягає в тому, щоб знайти інтерфейс для нової технології та *користувача*, щоб технологія була *зручною для використання*. Це особливо стосується старших поколінь, які не вирости з інформаційними технологіями. Тут завдання дизайнерів полягає в простому дизайні інтерфейсу користувача, який легко зрозуміти та легко вивчити. Також важливу роль відіграють економічні аспекти. Проблеми, з якими стикається творець користувацького інтерфейсу, починаються на початку теоретичного етапу. Тут необхідно визначити потреби користувача. Також важливо адаптувати користувацький інтерфейс до існуючого апаратного пристрою та

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

його властивостей. Для теоретичної розробки користувацьких інтерфейсів доступні різні підходи. Також стандарти щодо *зручності використання* вже встановлені; однак їх має інтерпретувати розробник.

Щоразу, коли людина використовує цифровий продукт, машину чи інструмент, взаємодія відбувається через межу або інтерфейс між машиною та людиною, навіть коли користувачі мають намір спілкуватися між собою. Якщо інтерфейс правильно структурований, то користувачі, ймовірно, матимуть задовільний досвід, який запрошуватиме їх повертатися знову і знову. Проектування елегантних користувацьких інтерфейсів для складних обчислювальних систем створює складні завдання [6]. Використання аналізу варіантів використання в процесі розробки програмного забезпечення все частіше використовується, оскільки варіанти використання допомагають звести складні системи до керованих аспектів [7]. Питання зручності використання в дизайні привертають більше уваги, ніж будь-які інші, в останні роки [8]. Проектування програмного забезпечення, включаючи проектування користувацького інтерфейсу, базується на сучасних передових практиках, оскільки практикуючі інженери розробили корисні стратегії на основі минулого досвіду [9]. Підтримка контекстно-залежних користувацьких інтерфейсів розвивається до рівня, коли це стає можливим навіть у великих системах [15]. Якість користувацького інтерфейсу важко оцінити, і все ж нова дисципліна намагається це зробити. Гарний інтерфейс користувача по-справжньому цінується лише тоді, коли він інтегрований з інтелектуальною архітектурою загальної системи, включаючи апаратне та програмне забезпечення, яке надає корисну послугу змістовним чином. Інтерфейси користувача не можна розглядати окремо від усієї інтегрованої системи. Розробку програмного забезпечення часто вважають одним із найскладніших процесів сучасних технологій, враховуючи різноманітність знань, ідей, навичок та креативності, необхідних для побудови інтелектуальних систем, щоб досягти успіху на сучасному швидкоплинному та жорсткому конкурентному ринку. Розробку програмного забезпечення можна розглядати як процес створення знань. Деякі

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

підходять до цього з наукової точки зору, а інші — художньо-творчо. Протягом десятиліть було запропоновано безліч підходів до розробки програмного забезпечення. Ці підходи часто описуються вражаючими метафорами.

Значення та роль інженерії користувацького інтерфейсу в дизайні продукту нещодавно були в центрі уваги багатьох дуже успішних інтерактивних систем [10]. Деякі аспекти користувацьких інтерфейсів, включаючи графічні аспекти, не могли бути належним чином розроблені до появи об'єктно-орієнтованого програмування. Дійсно, стало легше розробляти та реалізовувати графічний інтерфейс користувача (GUI) з об'єктно-орієнтованими концепціями та мовами. Уніфікована мова моделювання (UML) зробила значний внесок у представлення дизайну програмного забезпечення, включаючи певні аспекти зручності використання [11]. UML включає моделювання аспектів варіантів використання в різних представленнях, включаючи представлення варіантів використання [12]. Однак UML не включає моделювання та представлення графічного інтерфейсу користувача. У цій роботі критично розглядаються важливі питання розробки та представлення варіантів використання UML, а також пропонується розширене представлення варіантів використання, яке більше підходить для сучасного моделювання користувацького інтерфейсу. У ній пропонується належним чином включати певні елементи інтерфейсу в діаграми варіантів використання. У ній пропонуються деякі елементи моделювання графічного інтерфейсу користувача інтуїтивно зрозумілою мовою, подібною до UML. Крім того, вона представляє набір принципів розробки інноваційних функцій користувацького інтерфейсу відповідно до пропозицій.

У сучасному житті нас оточує технічне обладнання. Також життя без комп'ютерів для деяких людей вже неможливе. Так само використання нових інформаційних технологій також відіграє важливу роль у бізнес-секторі. У цьому відношенні важливо розробляти такі пристрої для користувачів, які мають менше технічних знань. Пристрій також повинен бути корисним для користувача без необхідності тривалого періоду навчання перед його використанням. В економічному секторі ефективність пристрою з категорії інформаційних

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

технологій відіграє важливу роль. Ефективність пристрою також залежить від того, скільки зусиль потрібно для його використання. Цей момент також включає час навчання, необхідний для того, щоб навчитися користуватися пристроєм, а також зручність використання під час активної експлуатації.

Завдання дизайнера інтерфейсу користувача полягає у створенні інтерфейсу між пристроєм та користувачем. В ідеальному випадку цей інтерфейс має бути легким для вивчення користувачем. Також частиною цього завдання є уникнення розчарування користувача під час використання. Розчарування може призвести до зниження ефективності під час активної роботи з економічної точки зору. Ще одним обов'язком дизайнера інтерфейсу користувача є донесення нової технології до недосвідчених користувачів. За допомогою відповідного інтерфейсу користувача можна подолати першу перешкоду, оскільки користувач використовує інтерфейс користувача для керування пристроєм. З огляду на різні виклики та вимоги до дизайнера інтерфейсу користувача, це дослідження має вказати на важливі аспекти та методи, що стосуються процесу проектування інтерфейсу користувача.

Оскільки технологічний розвиток продовжується, з часом з'являтимуться нові функції, а старі стають непотрібними. Щоб зробити ці нові функції корисними для багатьох користувачів і багатьох поколінь, необхідні подальші дослідження, щоб вказати на важливі аспекти щодо розробки відповідного інтерфейсу користувача. Однак з часом змінюється не лише технологія. Це також сам користувач і його поведінка, яка не завжди залишається незмінною. Таким чином, необхідні подальші дослідження в цій галузі.

З моєї власної точки зору, в хорошому інтерфейсі користувача користувач повинен мати можливість виконувати завдання, не витрачаючи забагато часу на сам інтерфейс, або ж інтерфейс користувача повинен бути *непомітним* для користувача. Інтерфейс користувача повинен служити інструментом для користувача, а не додатковим викликом. Інструмент має бути простим у використанні, щоб забезпечувати легке використання програмного забезпечення чи пристрою. Інтерфейс користувача також повинен допомагати поколінням, які

не вирости з інформаційними технологіями. Таким чином, ці покоління також зможуть використовувати нові технології. Цих цілей може досягти дизайнер, враховуючи важливі аспекти та *принципи дизайну*, що стосуються дизайну інтерфейсу користувача. Ці принципи будуть виведені в рамках цього дослідження.

Завдання в процесі проектування користувацького інтерфейсу полягає в задоволенні потреб користувача. Важливо знайти правильний підхід до розробки користувацького інтерфейсу для його належного призначення. Користувацький інтерфейс розроблений для використання пристрою, тому він має бути розроблений, перш за все, для людини. Розробнику часто нелегко побачити ситуацію з позиції недосвідченого користувача. У звичайному випадку розробник знайомий з власноруч створеним програмним забезпеченням. Це дослідження допоможе розробнику зрозуміти потреби недосвідченого користувача. Метою цього дослідження є створення розуміння важливих критеріїв щодо процесу проектування користувацького інтерфейсу, які необхідні для задоволення потреб ефективної комунікації між людиною та машиною.

У рамках цього дослідження я хочу спробувати дати відповіді на таке питання:

- Які принципи дизайну важливо використовувати під час розробки гарного користувацького інтерфейсу?

Основне дослідницьке питання включає такі підпитання:

1. Які доступні опції, щоб допомогти користувачам орієнтуватися в системі?
2. Які важливі принципи дизайну для підвищення зручності використання користувацького інтерфейсу?
3. Які аспекти важливі під час розробки користувацького інтерфейсу для конкретного пристрою?
4. Як можна уникнути створення бар'єрів для користувача під час розробки користувацького інтерфейсу?
5. Як уникнути перевантаження інформацією та, як наслідок,

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

плутанини для користувача?

Інформаційні технології використовуються у все більшій кількості сфер нашого життя. Тому важливо розробляти обладнання та технології, якими легко користуватися кожному. Це дослідження висвітлює важливі питання, які можуть виникнути під час теоретичного процесу розробки користувацького інтерфейсу, та пропонує можливі рішення цих проблем. За допомогою цього дослідження також будуть показані можливості підвищення мотивації користувача щодо використання програмного забезпечення або пристрою. Таким чином, ці можливості підвищення рівня мотивації користувача можна враховувати під час процесу проектування користувацького інтерфейсу. Ще одним очікуваним результатом цього дослідження є можливість використання цього дослідження як керівництва для всього теоретичного процесу розробки користувацького інтерфейсу.

1.2 Критерії оцінювання якості та зручності інтерфейсів (UX/UI)

Оцінюючи минулі та майбутні інструменти, ми визначили деякі теми, які здаються важливими для визначення того, які з них є успішними:

— *Частини інтерфейсу користувача, які розглядаються:* Інструменти, що прийшли на зміну, допомогли (саме) там, де вони були потрібні.

— *Поріг і стеля:* «Поріг» — це те, наскільки складно навчитися користуватися системою, а «стеля» — скільки можна зробити за допомогою системи. Найуспішнішими сучасними системами, здається, є або низький поріг і низька стеля, або високий поріг і висока стеля. Однак залишається важливим завданням знайти способи досягнення дуже бажаного результату систем з низьким порогом і високою стелею одночасно.

— *Шлях найменшого опору:* Інструменти впливають на типи користувацьких інтерфейсів, які можна створювати. Успішні інструменти використовують це на свою користь, спонукаючи розробників робити правильні речі та уникати неправильних.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

— *Передбачуваність*: Інструменти, що використовують автоматичні методи, які іноді є непередбачуваними, були погано сприйняті програмістами.

— *Рухомі цілі*: Важко створювати інструменти, не маючи значного досвіду та розуміння завдань, які вони підтримують. Однак швидкий розвиток нових технологій інтерфейсу та нових методів інтерфейсу може ускладнити для інструментів встигання за змінами. На той час, коли нове завдання реалізації користувацького інтерфейсу буде достатньо добре зрозуміле для створення хороших інструментів, воно може стати менш важливим або навіть застарілим.

Мови подій. За допомогою *мов подій* виникнення кожної значущої події, такої як маніпуляція користувачем із пристроєм введення, поміщається в структуру даних *запису події* (часто її просто називають *подією*).

Ці події потім надсилаються окремим обробникам подій, які містять код, необхідний для належної реакції на цей вхідний сигнал. Дослідники досліджували цей стиль у низці систем, включаючи Систему керування користувацьким інтерфейсом. Це призвело до дуже популярного використання мови подій у багатьох комерційних інструментах, таких як мова HyperTalk від Apple HyperCard, Visual Basic від Microsoft та мова сценаріїв Lingo в Director від Macromedia.

Мови подій були успішними, оскільки вони добре відповідають стилю графічного інтерфейсу користувача з прямим маніпулюванням. Ці системи генерують *події* для кожної дії користувача за допомогою миші та клавіатури, які спрямовуються до відповідної програми, яка потім повинна реагувати. Мови подій також сприяють заохоченню стилю інтерфейсів без режимів, оскільки користувач відповідає за генерування подій, які обробляє програма. Однак , як буде обговорено пізніше, інтерфейси користувача *на основі розпізнавання* , що з'являються для таких модальностей, як жести та мовлення, можуть не дуже добре відповідати цьому стилю, заснованому на подіях, тому нам може знадобитися нова парадигма.

Інтерактивні графічні інструменти. Ще одним важливим внеском у дослідження інтерфейсів користувача стало створення так званих *конструкторів*

інтерфейсів. Це інтерактивні інструменти, які дозволяють розміщувати інтерактивні компоненти за допомогою миші для створення вікон та діалогових вікон. Прикладами є Visual Basic та «редактори ресурсів» або «конструктори», що постачаються з Visual C++ від Microsoft та більшістю інших середовищ. Ранні дослідження цього класу інструментів включають Trillium від Xerox PARC та MenuLay.

Важливою причиною успіху конструкторів інтерфейсів було те, що вони використовують графічні засоби для вираження графічних концепцій (наприклад, макет інтерфейсу). Переміщуючи деякі аспекти реалізації користувацького інтерфейсу зі звичайного коду в інтерактивну систему специфікацій, ці аспекти реалізації інтерфейсу стали доступними для тих, хто не є традиційним програмістом. Це дозволило багатьом експертам у предметній області створювати прототипи та реалізовувати інтерфейси, що точно відповідають їхнім завданням, а також дозволило фахівцям з візуального дизайну більше залучатися до створення зовнішнього вигляду інтерфейсів. Навіть програмісти отримали вигоду, оскільки швидкість побудови різко знизилася. Ці властивості конструкторів інтерфейсів можна розглядати як забезпечення *низького порогу* використання та уникнення крутої кривої навчання (принаймні спочатку). У цих системах прості речі можна робити простими способами.

Компонентні системи. Ідея створення застосунків шляхом динамічного комбінування окремо написаних та скомпільованих *компонентів* була вперше продемонстрована в системі. Кожен компонент керував своїм прямокутником екрана, а інші компоненти могли бути вбудовані всередину. Наприклад, малюнок у текстовому документі керувався б компонентом малювання, який був би незалежним від компонента текстового редактора. Ця ідея була прийнята в OLE та ActiveX від Microsoft, OpenDoc від Apple та Java Beans від Sun. Однією з причин успіху компонентної моделі є те, що вона розглядає важливий та корисний аспект створення застосунків: як належним чином модулювати

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

програмне забезпечення на менші частини, водночас забезпечуючи значні можливості та інтеграцію для користувачів.

Мови сценаріїв. Не випадково перші набори інструментів були розроблені з використанням мов програмування, які інтерпретувалися: Smalltalk, а потім Dlist, розроблені дослідниками Xerox PARC, мали невеликі набори інструментів. Інтерпретована мова дозволяє розробнику швидко створювати прототипи різних ідей інтерфейсу користувача та негайно вносити зміни, що забезпечує швидкий результат. Зі зростанням популярності C та C++ більшість розробок інтерфейсу користувача перейшли на компільовані мови, і ці можливості були втрачені. Дослідники досліджували способи повернення цих переваг, що призвело до появи таких мов сценаріїв, як tcl/tk, Python та Perl.

Ми спостерігаємо, що будь-яка мова сценаріїв, яка досягає широкого використання, здається, вимагає універсальних структур керування, таких як змінні, цикли та умовні оператори. Хоча ці мови призначені для непрофесійних програмістів, вони майже завжди використовуються для того, щоб вийти за рамки того, що розробники мови спочатку мали на увазі, тому повна програмованість неминуче виявляється необхідною. Фактично, коли хтось розробляє *будь-яку* мову, яка має бути «спрощеною» (з мови програмування загального призначення), користувачі завжди хочуть, щоб вона мала більше можливостей. Тому повний спектр структур керування зазвичай знову з'являється. Це здається дилемою компромісу, яку неможливо обійти.

Поєднання можливостей скриптування з компонентами та конструктором інтерфейсів виявилось особливо потужним підходом. Наприклад, існують тисячі компонентів для Visual Basic, доступних від сторонніх постачальників. Використання конструктора інтерфейсів Visual Basic для макета та мови Visual Basic для написання скриптів – «клею», який об'єднує все разом, дозволяє людям, які не є професійними програмістами, створювати складні та корисні інтерактивні програми. Visual Basic показує, що невелике програмування – якщо його правильно упакувати – може дозволити експертам у предметній області

створювати інтерфейси, що відображають їхні знання про предметну область та завдання.

Гіпертекст. Всесвітня павутина (WWW) – це ще один вражаючий успіх досліджень програмного забезпечення та технологій користувацького інтерфейсу. Вона базується на ідеї *гіпертексту*. Тед Нельсон ввів термін «гіпертекст» у 1965 році та працював над однією з перших систем гіпертексту під назвою «Система редагування гіпертексту» в Університеті Брауна. Система NLS також мала функції гіпертексту. Hyperties Університету Меріленду була першою системою, де виділені елементи в тексті можна було клацнути, щоб перейти на інші сторінки. HyperCard від Apple відіграв значну роль у популяризації цієї ідеї серед широкої аудиторії.

Однак гіпертекст не набув широкого поширення до створення системи Всесвітньої павутини Бернерсом-Лі в 1990 році та браузера Mosaic кількома роками пізніше. Деякими з елементів успіху WWW є простота використання Mosaic, простота та доступність мови HTML, що використовується для створення сторінок, легкість створення сторінок доступними в Інтернеті та вбудовування зображень у текст. WWW забезпечувала низький *поріг* використання як для глядачів, так і для авторів. Глядачі мали простий механізм, який забезпечував доступ до багатьох існуючих мережевих ресурсів (наприклад, ftp, telnet, gopher тощо) в рамках гіпертекстової структури, а автори використовували дуже просту мову текстових специфікацій HTML. Це дозволило постачальникам контенту використовувати систему з мінімальним навчанням. По-друге, успіх браузера Mosaic чітко продемонстрував силу та переконливу природу візуальних зображень (і загалом, багатого контенту з високою виробничою цінністю).

Об'єктно-орієнтоване програмування. Об'єктно-орієнтоване програмування та дослідження користувацького інтерфейсу мають довгу та взаємопов'язану історію, починаючи з прагнення Smalltalk спростити створення інтерактивних графічних програм. C++ став популярним, коли програмування графічних користувацьких інтерфейсів стало широко необхідним у Windows

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Об'єктно-орієнтоване програмування особливо природне для програмування користувацького інтерфейсу, оскільки компоненти користувацьких інтерфейсів (кнопки, повзунки тощо) проявляються як видимі об'єкти з власним станом (який відповідає змінним екземпляра) та власними операціями (які відповідають методам).

Перспективні підходи, які не прижилися

Окрім уроків, отриманих з успіхів, багато чого можна також навчитися, розглядаючи ідеї, які спочатку здавалися багатообіцяючими, але зрештою не змогли (або принаймні ще не змогли) виконати цю обіцянку. Багато з них піддалися проблемі рухомої цілі: у міру дослідження цих систем стилі інтерфейсів користувача змінювалися в бік сучасного стандартного графічного інтерфейсу користувача. Крім того, багато з цих інструментів були розроблені для підтримки гнучкої різноманітності стилів, що втратило значення зі стандартизацією. У цьому розділі розглядається кілька таких прикладів, включаючи концепцію системи керування інтерфейсом користувача (UIMS), підходи на основі мови, обмеження та системи на основі моделей.

Системи керування інтерфейсом користувача. На початку 80-х років концепція системи керування інтерфейсом користувача була важливим фокусом для тодішньої спільноти розробників програмного забезпечення для інтерфейсів користувача. Термін «система керування інтерфейсом користувача» був введений, щоб запропонувати аналогію із системами керування базами даних. Системи керування базами даних реалізують набагато вищу та зручнішу абстракцію поверх низькорівневих концепцій, таких як диски та файли [13]. Забезпечуючи цю абстракцію, вони дозволяють отримати незалежність від низькорівневих деталей, автоматизують (або просто приховують) багато раніше дуже виснажливих завдань і загалом роблять завдання створення сховищ даних набагато простішим і надійнішим. Аналогічно, системи керування інтерфейсом користувача повинні були абстрагувати деталі пристроїв введення та виведення, забезпечуючи стандартні або автоматично згенеровані реалізації інтерфейсів і загалом дозволяючи визначати інтерфейси на вищому рівні абстракції.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Однак, на практиці цей поділ не дуже добре спрацював. Для кожного користувацького інтерфейсу важливо контролювати низькорівневу прагматику того, як виглядають та відчуються взаємодії, яку ці UIMS намагалися ізолювати від дизайнера. Крім того, стандартизація елементів користувацького інтерфейсу наприкінці 1980-х років на парадигмі робочого столу зробила потребу в абстракціях від пристроїв введення здебільшого зайвою. Таким чином, UIMS стали жертвами проблеми рухомої цілі.

Інструменти на основі формальних мов. У низці ранніх підходів до побудови UIMS використовувалися методи, запозичені з формальних мов або компіляторів. Наприклад, багато систем базувалися на діаграмах переходів станів та синтаксичних аналізаторах для контекстно-вільних граматик. Спочатку ці підходи виглядали дуже перспективними. Однак ці методи не прижилися з кількох важливих причин, які можуть служити важливими уроками для майбутніх інструментів.

Використання методів формальних мов було частково зумовлене раннім акцентом у роботі UIMS на завданні *управління діалогами*. У той час, коли ці ранні системи управління інтерфейсом користувача задумувалися, домінуючий стиль інтерфейсу користувача базувався на розмовній метафорі (в якій система та користувач розглядаються як такі, що розмовляють про об'єкти інтересу). У цьому контексті управління діалогами відіграє дуже центральну роль, і насправді низка систем на основі формальних мов дуже добре справлялася з підтримкою цього центрального завдання. На жаль, саме в той час, коли розроблялися ці ранні системи, стиль інтерфейсу прямої маніпуляції швидко набував популярності. В інтерфейсах прямої маніпуляції роль управління діалогами значно зменшується, оскільки структурування діалогу системою зазвичай шкодить концепції прямої. В результаті ці ранні інструменти інтерфейсу користувача швидко стали такими, що дуже добре справлялися з вирішенням проблеми, яка більше не мала значення, тим самим ставши жертвою проблеми рухомої цілі.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Існують й інші проблеми з цим класом інструментів. У цих системах дуже легко виразити послідовність (і важко виразити неупорядковані операції). Як результат, вони схильні спонукати програміста створювати інтерфейси з жорсткими послідовностями необхідних дій. Однак, з точки зору прямої маніпуляції, необхідні послідовності майже завжди небажані. Таким чином, шлях *найменшого опору* цих інструментів шкідливий для гарного дизайну користувацького інтерфейсу.

Ще однією причиною, чому деякі системи не прижилися, є *високий поріг* їх використання, оскільки вони вимагали від програмістів вивчення нової спеціальної мови програмування (на додаток до їхньої основної мови реалізації, такої як Pascal, C або C++). Що ще важливіше, системи на основі формальних мов часто вимагали від програмістів використання нових концепцій програмування. Наприклад, деякі з цих систем використовували граматики для вираження послідовності в діалогах. Однак багато програмістів не вміли мислити в термінах граматик. Незважаючи на те, що значно покращена потужність інструментів, здавалося, виправдовувала круту криву навчання, багато потенційних користувачів цих систем не застосовували їх просто тому, що вони так і не подолали початкових труднощів.

За допомогою обмежень розробник може вказати, наприклад, що лінія повинна залишатися приєднаною до прямокутника, або що смуга прокручування повинна залишатися праворуч від вікна. Аналогічно, можна оголосити, що колір, розмір або положення об'єктів у кількох представленнях мають бути отримані зі значення об'єкта в програмі, або що значення одного об'єкта програми має бути пов'язане з іншим на основі формули. Після оголошення цих зв'язків система розв'язання обмежень реагує на зміни в будь-якому місці системи, оновлюючи будь-які значення, необхідні для підтримки оголошених обмежень.

Системи обмежень пропонують прості, декларативні специфікації для можливостей, корисних для реалізації кількох різних аспектів інтерфейсу. Крім того, для їх реалізації було розроблено низку ефективних алгоритмів. Однак, системи обмежень ще не отримали широкого поширення поза межами

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

дослідницьких систем. Одна з головних причин цього полягає в тому, що програмістам не подобається, що розв'язувачі обмежень іноді *непередбачувані*.

Після встановлення набору обмежень завданням розв'язувача є пошук рішення, і якщо існує кілька рішень, розв'язувач може знайти те, якого користувач не очікував. Якщо в методі обмежень є помилка, її може бути важко знайти. Пов'язана проблема, особливо з більш складними розв'язувачами, полягає в тому, що для створення набору обмежень, які може обробити розв'язувач, програмісту може знадобитися переконатися, що набір обмежень має певну форму, наприклад, ациклічну або лише лінійну, або може знадобитися вивчити деталі того, як працює розв'язувач, що створює *високий поріг*. Крім того, декларативний характер обмежень часто важко опанувати людям, які звикли програмувати імперативними мовами — це вимагає від них по-іншому мислити про свої проблеми, що також сприяє наявності *високого порогу*.

Повертаючись до тем, найуспішніші інструменти були зосереджені на певній частині користувацького інтерфейсу, яка була значною проблемою, і яку можна було ретельно та ефективно вирішити. Тривала стабільність парадигми робочого столу та стилю користувацького інтерфейсу з прямим керуванням дозволили вдосконалити інструменти, полегшивши проблему рухомої цілі, яка впливала на найперші інструменти. Однак, як обговорюється далі, ми прогнозуємо нову фазу посиленого експериментування та змін у користувацьких інтерфейсах у майбутньому, що знову зробить проблему рухомої цілі важливою.

Подвійні проблеми ефектів порогу та стелі є поширеними серед інструментів користувацького інтерфейсу. Однією з постійних проблем роботи з інструментами користувацького інтерфейсу є те, як застосовувати дуже потужні підходи та при цьому зробити їх доступними та зрозумілими для типових розробників. Часто існує спокуса вважати, що потужність підходу виправдовує круту криву навчання. У багатьох випадках простий аналіз витрат/вигод може навіть вказати на це. Крім того, деякі методи досягнення потужності, необхідної для систем з високими стелями, здається, мають ефект підвищення порогу. Аналогічно, багато спроб знизити поріг були за рахунок потужних функцій, які

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

дозволяють досягти високої стелі [14]. Такі системи, як SUIT, змогли забезпечити дуже низькі пороги, але надавали доступ до підтримки вищих стелей лише через підхід «пастки», що дозволяє загальні звернення до базової мови програмування. Майбутні тенденції вимагатимуть нових і потужних методів (висока стеля) для вирішення очікуваного різкого зростання різноманітності користувацького інтерфейсу. Уроки минулих інструментів показують, що на практиці системи з високим порогом не були прийняті, оскільки їхня цільова аудиторія ніколи не перевищує початковий поріг продуктивності, тому інструменти повинні забезпечувати легкий вхід і плавний шлях до збільшення потужності. Хоча це дуже складно, створення систем як з низьким порогом, так і з високою стелею є можливим і продовжуватиме залишатися важливою метою для майбутніх інструментів.

Ще одне питання стосується шляху найменшого опору, який пропонує інструмент. Наші інструменти мають глибокий вплив на те, як ми думаємо про проблеми. Залишатиметься вкрай важливим, щоб наші інструменти полегшували «робити правильні речі» (на відміну від «робити неправильні речі»). Це стосується як структури та організації коду, так і користувацького інтерфейсу, який є результатом використання коду. Іншими словами, інструменти повинні полегшувати створення високоякісних користувацьких інтерфейсів, ніж створення низькоякісних.

1.3 Методи дослідження та проєктування інтерфейсів користувача

У цьому підрозділі буде пояснено методологію, яка використовувалася для цього дослідження. Також буде пояснено, чому для цього дослідження використовуються конкретні методології. У цій частині також буде описано, як будуть збиратися та оброблятися дані.

Перед початком дослідження слід чітко визначити, який дослідницький підхід буде обраний для дослідницького проєкту. Для дослідження в рамках цього дослідження зосереджено увагу на двох підходах: *герменевтиці* та

позитивізмi.

Підхід із використанням герменевтики особливо допомагає в галузі соціальних досліджень. За допомогою методу герменевтики поведінку людини та людські дії можна вивчати за допомогою суб'єктивних підходів стосовно дослідницького проекту. Це означає, що результати повинні інтерпретуватися дослідником. Дослідницький підхід за допомогою позитивізму більше зосереджується на об'єктивних підходах. Це включає, наприклад, природничі та математичні науки.

Для підходу за допомогою герменевтики це означає початок дослідження з того, як дослідник розуміє дану проблематику. Однією з причин, чому було обрано підхід з герменевтикою, є мета цього дослідження. Метою цієї роботи є створення розуміння важливих критеріїв для проектування користувацького інтерфейсу [15]. Як згадувалося вище, герменевтика використовується як підхід у галузі соціальних досліджень. Герменевтика – це підхід до інтерпретації соціальної поведінки та людської взаємодії. Особливо для інтерв'ю отриману інформацію необхідно інтерпретувати. Але також дані, які будуть зібрані для всього дослідження, можна розглядати радше як *можливості*, а не як фіксовані факти. Основні дані для цього дослідження в основному зібрані з академічних книг та наукових статей, а також з інтерв'ю. З цієї причини проводиться якісне дослідження, оскільки метою цього підходу є інтерпретація *зібраних* даних. Способи проведення якісного дослідження:

- Натуралізм: спроба *зрозуміти* реальність.
- Етнометодологія: намагається *зрозуміти*, як соціальна поведінка та порядок формуються розмовами та людською поведінкою.
- Емоційність: намагається *зрозуміти* внутрішні установки людей та людську поведінку.

Вищезазначені пункти містять підхід до розуміння конкретної галузі дослідження. Основна відмінність між якісним та кількісним методом полягає у способі збору даних, необхідних для дослідження. Результат кількісного дослідження легше представити, оскільки зібрані дані можна представити у

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

вигляді діаграм та цифр. Цей метод також використовується для статистичних питань, оскільки спосіб збору даних дозволяє створити певну статистику. Тому кількісні дані більше зосереджені на зборі фіксованих даних, тоді як якісний підхід більше зосереджений на поглибленій поведінці людей. Такий вид дослідницької перспективи часто використовується в соціальних науках. У цьому дослідженні основна лінія буде спрямована на якісний підхід. Оскільки я проходжу стажування в Opera Software ASA, це дає гарну можливість дізнатися більше інформації про важливі моменти щодо дослідницьких питань з точки зору досвідченого дизайнера, відповідального за користувацькі інтерфейси.

Стратегія дослідження

Визначення терміну «Дослідницька стратегія» можна підсумувати наступним чином: Дослідницька стратегія – це план дій, який здійснюється для збору відповідної інформації для дослідження. Залежно від дослідницьких питань, це включає план, як дослідник може отримати інформацію щодо теми ефективним способом та у визначений часовий діапазон. Ця стратегія також включає час та фінансові ресурси, доступні досліднику. Якщо дослідження проводиться, рекомендується знати дослідницьку стратегію, яка використовується для дослідження. Однією зі стратегій збору інформації є дедукція. Крім того, в цьому контексті доступні *індукція* та їх поєднання. Якщо дослідження проводиться індуктивним способом, це означає, що дослідник починає безпосередньо з емпіричного дослідження без обґрунтованої теорії. Виходячи з результатів, отриманих за допомогою емпіричного дослідження, дослідник потім формує теорію. Дедуктивна дослідницька стратегія означає початок роботи з існуючими теоріями.

Це дослідження буде проводитися дедуктивним методом. Причиною такого підходу є вже проведені дослідження в цій галузі. Таким чином, вже існує багато існуючих теорій. Однак, як зазначалося у вступі, ця галузь перебуває в русі та все ще надає можливості для подальшого дослідження. Дослідження починається з теоретичної частини; отже, теорія є основою дослідження. Щоб з'ясувати, чи теорія відповідає реальності, буде використано емпіричне

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

дослідження, яке включає інтерв'ю з досвідченим дизайнером користувацького інтерфейсу, щоб спробувати підтвердити теоретичні висновки. Однак, якщо між теорією та реальністю є розбіжності, ці моменти будуть висвітлені. Також буде згадана додаткова інформація, отримана під час інтерв'ю. Таким чином, інтерв'ю пропонує погляд з точки зору реальності.

Процедури збору даних

Під цією темою ми розуміємо спосіб збору даних, необхідних для процесу цього дослідження. Існують різні підходи до збору корисних та необхідних даних для певної теми. У цій роботі основна увага приділяється як первинним, так і вторинним даним. Первинні дані збираються дослідником, наприклад, за допомогою інтерв'ю або анкет. Також спостереження є методом збору первинних даних. У процесі збору первинних даних дослідник також має більше можливостей вибрати, яким чином ці дані будуть зібрані. Таким чином, дослідник може збирати дані таким чином, щоб забезпечити найбільше інформації, необхідної для теми дослідження. Як недолік первинних даних слід згадати зусилля для цього способу дослідження. Наприклад, кроки для підготовки інтерв'ю потребують більше часу, а також аналіз результатів. Ще одним недоліком можуть бути фінансові зусилля, необхідні для збору первинних даних. Ці фінансові зусилля можуть включати транспортні витрати для інтерв'ю або витрати та час на отримання даних з анкети. Вторинні дані складаються з інформації, яка збирається іншими особами. Ці зібрані дані могли бути зібрані з інших причин, залежно від теми, яку досліджує дослідник. Але вони можуть надати корисну інформацію, важливу для дослідника. За допомогою вторинних даних можна створити основу для дослідження. Це також може допомогти зменшити фінансові витрати, пов'язані з самим дослідженням. Ще однією перевагою вторинних даних є можливість отримати уявлення про дослідження, яке проводилося протягом тривалішого періоду. Наприклад, огляд показників компанії може надати інформацію про її минулий успіх. Цю інформацію можна зібрати за дуже короткий час. Недоліком вторинних даних є те, що в певний момент може знадобитися відсортувати інформацію, необхідну для власної теми

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

дослідження, оскільки ці дані могли бути зібрані для іншого питання.

У цій роботі використовується комбінація обох методів. Оскільки існує багато досліджень, які надають інформацію щодо цієї теми, ці вторинні дані будуть використані для отримання відповідної інформації для цього дослідження. Первинні дані будуть використані для того, щоб зосередитися на конкретних питаннях та отримати точку зору розробника користувацького інтерфейсу [16]. Первинні дані також будуть використані для підтвердження висновків, отриманих з вторинних даних. Далі будуть обговорені методи вибірки, використані для теоретичної та емпіричної частини цього дослідження.

Теоретична частина

У теоретичній частині будуть використані вторинні дані. Дані будуть відібрані з наукових статей, які містять інформацію, що відповідає цьому дослідженню. Також увага в цьому дослідженні приділяється науковим книгам. У цій частині підхід полягає в тому, щоб з'ясувати, які дані вже зібрані, а на які питання ще немає відповідей. *Наративний огляд* – це підхід, який відповідає потребам дослідження, виконаного інтерпретаційним способом, оскільки цей підхід є менш фіксованим і таким чином забезпечує ширшу перспективу.

Емпірична частина

Для емпіричної частини було обрано інтерв'ю, щоб отримати більше інформації для відповіді на дослідницькі питання, а також для підтвердження теоретичних висновків або для виявлення відмінностей. Завдяки моєму стажуванню в Opera Software ASA, я маю можливість отримати безпосередню інформацію з точки зору розробника, оскільки ця компанія займається створенням користувацьких інтерфейсів. Тому дизайнери користувацьких інтерфейсів працюють у цій компанії-розробнику програмного забезпечення, і буде зроблена спроба встановити прямий контакт з дизайнером користувацького інтерфейсу з метою отримання інформації з перших вуст :

«Продуктивність людини та взаємодія користувача з комп'ютерними та інформаційними системами залишатимуться темою досліджень та розробок, яка швидко розвиватиметься, у найближчі десятиліття».

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Таким чином, важливо отримати актуальну точку зору досвідченого дизайнера користувацького інтерфейсу [17]. Оскільки на компанію також впливають зміни в інформаційних технологіях, адаптація є необхідною для компанії. Співбесіда має надати актуальну точку зору.

Процедури аналізу даних

Після збору первинних та вторинних даних необхідний їхній аналіз. Це означає виявлення важливої інформації стосовно дослідницьких питань та теми дослідження. Також необхідно довести якість даних. Особливо це стосується вторинних даних, і це важливий момент, який необхідно враховувати. Висновок емпіричного дослідження записується в текстовому форматі та порівнюється з даними, зібраними для теоретичного дослідження. У цій роботі це буде зроблено за допомогою якісного підходу, що означає, що зібрані дані будуть відфільтровані на предмет відповідної інформації щодо теми. Останній крок допоможе класифікувати дані таким чином, щоб на наступному етапі було легше довести теоретичну частину на практиці за допомогою емпіричної частини.

Стратегія перевірки результатів

У цьому дослідженні для перевірки знайдених дослідницьких даних використовуються методи тріангуляції. Описується тріангуляція наступним чином: «Використання більш ніж одного методу або джерела даних у вивченні соціального явища для можливості перехресної перевірки результатів»

Таким чином, тріангуляція допомагає у перевірці результатів шляхом проведення крос-методу дослідження або використання кількох джерел даних. Тріангуляція використовується переважно для кількісних дослідницьких підходів, але також може бути використана в якісних дослідженнях. У цьому дослідженні метод тріангуляції буде використовуватися шляхом порівняння теоретичних висновків з емпіричними результатами. Також тріангуляція даних буде використовуватися для перевірки результатів, оскільки дані для теоретичної частини будуть зібрані з різних джерел. Одним з видів джерел тут є наукові статті, які можуть надати важливу ключову інформацію щодо цієї теми дослідження. Іншими джерелами є академічні книги, які надають більшу

кількість обґрунтованої інформації стосовно теми дослідження.

Метод представлення результатів

Основною формою презентації в цій роботі є письмовий текст. Текст поділено на три частини. Зображення використовуються для кращого розуміння заданих понять. Для порівняння конкретних даних використовуються таблиці, щоб підкреслити відмінності.

1.4 Висновки по розділу

У першому розділі розглянуто теоретичні основи проєктування інтерфейсів користувача. Проаналізовано основні поняття та класифікацію інтерфейсів, визначено ключові критерії оцінювання їх якості та зручності використання. Досліджено методи аналізу потреб користувачів і підходи до проєктування UX/UI-рішень. Встановлено, що ефективний інтерфейс має забезпечувати зрозумілу взаємодію, доступність та комфортне використання програмного забезпечення.

РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІНТЕРФЕЙСІВ

2.1 Основні принципи та підходи до проєктування інтерфейсів

Користувач : особа, яка використовує певний технічний пристрій або певне програмне забезпечення. Це може мати місце як у приватному середовищі, так і в бізнес-середовищі.

					БР.ІП – 46.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс користувача інтерфейс між користувачем-людиною та технічним пристроєм. За допомогою цього інтерфейсу користувач може надсилати вхідні дані на пристрій, щоб мати змогу використовувати його.

Прийнятність для користувача : Прийнятність для користувача означає прийнятність та внутрішнє ставлення користувача до певного програмного забезпечення або певного пристрою. Досягнення високої прийнятності для користувача є частиною роботи *дизайнера інтерфейсу користувача*.

Дизайнер інтерфейсу користувача : людина, яка відповідає за дизайн інтерфейсу користувача. Це можна розглядати як з технічної точки зору, так і з теоретичної точки зору розробника.

Стандарт ISO : ISO – це скорочений термін від *Міжнародної організації зі стандартизації*. Стандарт ISO визначає критерії, що описують зручність використання продуктів різних виробників один з одним.

Зручність використання програмного забезпечення : цей термін означає працездатність певного програмного забезпечення. Чим краще враховується зручність використання під час розробки програмного забезпечення, тим вища прийнятність програмного забезпечення для користувача. За допомогою цього терміна можна отримати пояснення наступного слова.

Якість програмного забезпечення : У цьому контексті головну роль відіграє стандарт ISO/IEC 9126. У цьому стандарті визначено ключові концепції якості програмного забезпечення. Фактори, що стосуються якості, включають, серед іншого, *функціональність*, *надійність* та *зручність використання*. У процесі проектування користувацького інтерфейсу цей момент є дуже важливим.

Ефективність програмного забезпечення : цей термін також можна знайти у стандарті ISO/IEC 9126. Один з аспектів, який описується цим терміном, – це рівень продуктивності програмного забезпечення стосовно (фінансових) ресурсів, що використовуються для певної мети.

Метод введення : описує, як користувач може надсилати вхідні дані на пристрій, або ж сам пристрій використовує.

У цьому дослідженні буде використано підхід, що полягає у відповіді на

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

головне дослідницьке питання за допомогою відповідей на підпитання. На головне дослідницьке питання відповідає переважно теоретичне дослідження. Емпіричне дослідження допоможе підтвердити знання, отримані в результаті теоретичного дослідження. Це дослідження охоплює три основні напрямки дослідження, які будуть описані нижче.

По-перше, увага приділяється *взаємодії людини з комп'ютером*. Тут поведінка користувача та характеристики користувача є важливою областю дослідження для цього дослідження. Для дизайнера важливо знати користувача та те, як він взаємодіє з пристроєм. Також важливо, як користувач реагує на різні сигнали, що надходять від інтерфейсу користувача. Цей розділ в основному стосується підпитань 1 та 4.

Як друга категорія, буде розглянуто *інженерію зручності використання*. Зручність використання є частиною цього дослідження, оскільки важливим моментом для створення користувацького інтерфейсу є зручність використання. Щоб досягти відповідної зручності використання продукту або, в цьому випадку, користувацького інтерфейсу, необхідно враховувати важливі аспекти для досягнення цієї мети. Цей розділ стосується підпитань 1, 2, 4 та 5.

Третя сфера, яка є важливою для цієї теми, – це *дизайн графічного інтерфейсу*. Аспекти дизайну дуже важливі для всього процесу створення користувацького інтерфейсу. Тут розглядаються теоретичні аспекти з вищезазначених галузей під час впровадження елементів зі специфічними методами в користувацький інтерфейс. Але також питання дизайну є частиною цього розділу. Цей розділ має надати відповіді на дослідницькі питання 3 та 4.

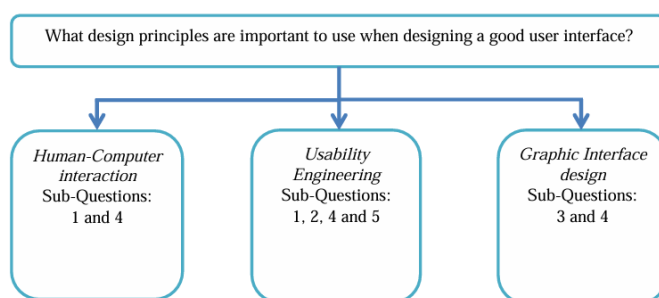


Рисунок 2.1 - Галузі, що мають значення для дослідження

Важливі аспекти взаємодії людини з комп'ютером

Коли людина взаємодіє з комп'ютером, можна знайти певну схожість із взаємодією з іншими речами у світі. Людина запам'ятовує речі та способи досягнення мети, і таким чином адаптується до неї. Люди розвивають специфічне відчуття поведінки комп'ютерних систем і очікують такої ж поведінки в майбутньому. Але людина також є істотою звички. Іноді важко адаптуватися до нових речей і ситуацій, також у сфері інформаційних технологій. Особливо, якщо старе програмне забезпечення або старий пристрій все ще виконують свою роботу. Таким чином, також не потрібно змінювати поточну поведінку користувача - з точки зору користувача [18]. Але загалом, поведінка людини змінюється з часом. Це також означає зміну того, як ми використовуємо пристрої з галузі інформаційних технологій. Слід розуміти, що те, як змінюється поведінка людини, також впливає на дизайн інтерфейсів користувача. Порівняно з часом, аспекти мобільності стають важливим фактором у цьому плані. Пристрої стають меншими, і потреба в доступі до Інтернету під час подорожі зростає. Також змінився спосіб нашої соціальної поведінки. Зараз ми віддаємо перевагу спілкуванню через соціальні мережі, оскільки можемо впливати на цей спосіб комунікації за допомогою зображень та спеціальних програм. Але не всі покоління страждають від цього. Часто це залежить від соціально-економічного стану розвитку країни. Також рівень освіти відіграє важливу роль у використанні інформаційних технологій та нових способів комунікації. Країни, які забезпечують добру комунікаційну інфраструктуру, що включає, серед іншого, широкопasmовий інтернет та високий технологічний рівень, мають, таким чином, вищу освіту щодо нових інформаційних технологій. Це починається ще з дитинства. Діти, які виростили з інформаційними технологіями, мають іншу поведінку під час взаємодії з пристроями порівняно з поколіннями, які не виростили з новими технологіями. Поведінка людини щодо нових технологій також є питанням, яке відношення людина займає до цього виду технологій. Також важливо розуміти аспекти

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

мотивації: чи необхідно розуміти нову технологію, чи я хочу її розуміти? Особливо для старших поколінь це питання може бути важливим з психологічної точки зору. Якщо необхідно зрозуміти складну комп'ютерну систему, мотивація може бути нижчою, якщо є обмеження для розуміння нової технології. Таким чином, для дизайнера пристрою, особливо для дизайнера інтерфейсу користувача, є викликом підвищити рівень мотивації такого користувача. Інша точка зору пропонує погляд на молодше покоління, яке виросло з новими технологіями. Ці користувачі розглядають цю технологію як звичайний засіб для свого життя. Таким чином, рівень мотивації для розуміння нового пристрою чи нового програмного забезпечення може бути вищим [19]. Для молодшого покоління нові технології також часто розглядаються як іграшка. Звичайно, індустрія відеорозваг пропонує продукти спеціально для цієї мети. Але також нові соціальні платформи розглядаються не лише як спосіб спілкування та взаємодії з іншими учасниками. Також мобільні пристрої можна розглядати як іграшку з точки зору молодої людини. З точки зору дитини, іграшку слід відкривати за її особливості та особливості. Для дизайнера цей аспект також може бути важливим критерієм для процесу проектування певної цільової групи.

Визначення користувача

У кожній галузі, доступній в рамках інформаційних технологій, користувачів можна класифікувати по-різному. Ми повинні розділити користувачів з професійної сфери в бізнесі та кінцевих користувачів, які використовують систему для особистих потреб. В обох категоріях користувач хоче досягти певної мети під час роботи з пристроєм або програмним забезпеченням. З одного боку, це може бути виконання завдання, пов'язаного з роботою, з якою стикається користувач. З іншого боку, однією з цілей може бути також виконання завдання в приватній сфері користувача. Користувач - це особа, яка безпосередньо взаємодіє з системою з інформаційних технологій. Таким чином, користувач також відповідає за роботу з пристроєм [20]. Також користувач відповідає за можливий успіх пристрою або певного програмного забезпечення. Інтерфейс користувача розроблений для користувача, і тому він

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

має бути оптимізований для користувача.

Таблиця 2.1

Порівняння продуктивності комп'ютера та людини

Властивість	Комп'ютер	Людина
Швидкість	Можлива швидка реакція	Присутні затримки
Потужність обробки	Велика	Обмежена
Послідовність (Стабільність)	Висока для числових операцій	Ненадійна: залежить від втоми, навчання тощо
Мислення (Аргументація)	Сильні дедуктивні здібності	Хороші індуктивні здібності
Надійність при перевантаженні	Раптовий збій (зупинка)	Поступове зниження ефективності
Інтелект	Обмежений (експертні системи, нейронні мережі)	Значний: може передбачати та адаптуватися

Перш ніж розпочати процес проектування для створення користувацького інтерфейсу, необхідно дослідити самого користувача. Користувацький інтерфейс повинен служити користувачеві, тому необхідні точніші знання про користувача. Спочатку необхідно дослідити сферу діяльності користувача. Дизайнер повинен отримати точку зору користувача, щоб зрозуміти, чого потрібно досягти за допомогою користувацького інтерфейсу. Залежно від робочого середовища користувача, дизайн інтерфейсу необхідно адаптувати. Наприклад, користувацький інтерфейс програми для обробки текстів не може служити для перегляду Інтернету. Під час проектування користувацького інтерфейсу важливо зосередитися лише на тих елементах, які дійсно потрібні. В іншому випадку це може призвести до плутанини, особливо серед початківців. Ще однією важливою частиною, яку слід враховувати, є різниця між людиною та комп'ютерною системою. Це може здатися простим; однак важливо знайти спосіб оптимізувати зв'язок між ними. Зв'язок між людьми та комп'ютерами і таким чином створив пряме порівняння:

Поведінка користувача щодо системи з галузі інформаційних технологій також залежить від того, як часто система використовується. Користувачі працюють із системою більш чи рідше. Таким чином, задається різна поведінка конкретних користувачів. Користувач, який часто працює з табличним

програмним забезпеченням, має іншу точку зору та інші очікування щодо програмного забезпечення. Такий тип користувача також по-іншому реагуватиме на оновлення або зміну інтерфейсу користувача. Такий тип користувача швидше адаптується до зміни програмного забезпечення, якщо основні елементи залишаються незмінними [21]. Користувачі, які рідше взаємодіють з певним програмним забезпеченням або системами, часто намагаються уникнути конфліктів з оновленням або певною зміною. Це є результатом тривалішого часу навчання через низьке використання програми або пристрою.

2.2 Забезпечення зворотного зв'язку та взаємодії з користувачем

Як і в інших сферах роботи, людина хоче отримати винагороду за виконане завдання. Винагорода, розглядається з психологічної точки зору, означає задоволення та заохочує до подальшої діяльності. З точки зору користувацьких інтерфейсів, цього можна досягти, наприклад, шляхом надання невеликих підтверджень після завершення дії, виконаної користувачем. Це можна зробити за допомогою звукових, а також графічних ефектів. Підтвердження кроку також забезпечує відчуття безпеки і таким чином створює більш звичне відчуття при використанні пристрою та його інтерфейсу користувача. Загалом, користувач повинен досягти бажаної мети в програмному забезпеченні якомога швидше [22]. Однією з можливостей тут є скорочення структури меню. Наприклад, просте та зрозуміле меню полегшує користувачеві навігацію. Крім того, це також сприяє кращій зрозумілості. Крім того, мінімізація часу відгуку є ще одним важливим критерієм. Короткий час очікування на відкриття програми або збереженого файлу забезпечує безперебійну роботу користувача. Однак на цей момент дизайнер може вплинути лише обмежено. Для забезпечення безперебійного робочого процесу іноді потрібне також відповідне потужне обладнання. Сучасні твердотільні накопичувачі для комп'ютерів, наприклад, значною мірою сприяють

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

безперебійному робочому процесу. Також важливу роль відіграють інші апаратні аспекти, такі як розмір основної пам'яті або швидкість процесора. Для дизайнера уникнення помилок під час створення користувацького інтерфейсу також слід розглядати без наукових досліджень як питання з високим пріоритетом. Проте слід зазначити, що непередбачені проблеми або кроки можуть дуже заплутати користувача. Особливо ці моменти впливають на користувачів-початківців. Ще одним аспектом підтримки або підвищення рівня мотивації користувача є характер формулювання в діалогових вікнах довідки або вступу. Таким чином, зрозуміле формулювання сприяє благополуччю користувача.

Уникнення розчарування користувача

Як уже зазначалося вище, розробник користувацького інтерфейсу повинен дотримуватися директиви уникати помилок. Якщо трапляється збій програми або пристрою, це має бути пояснено користувачеві у формулюванні, яке користувач може інтерпретувати. Крім того, це повідомлення повинно містити відповідні кроки для можливого вирішення, яке допоможе користувачеві. Користувач не повинен почуватися розгубленим у разі виникнення помилки програми. Повідомлення про помилку, яке занадто зосереджене на технічних деталях, корисне для технічного розробника, але не для користувача, який не знайомий з технічними деталями системи. Також тут важливим аспектом є формулювання: підтверджуюче формулювання повідомлення про помилку дає користувачеві краще відчуття безпеки. Формулювання, яке занадто драматизує виниклу проблему, може призвести до послаблення відчуття безпеки користувача.

Для певних програмних продуктів або обладнання навчання перед використанням відповідного продукту є неминучим. Особливо в бізнес-секторі зі складними програмними застосунками рекомендується вказати користувачеві на важливі робочі функції програми. Неправильне використання програмного забезпечення може завдати компанії фінансових збитків. Навчений користувач швидше ознайомиться з програмним забезпеченням і, таким чином, може швидше зробити внесок у підвищення продуктивності компанії. Навчання

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів повинно проводитися не лише перед першим використанням програми, але й під час її продуктивного використання. Таким чином, користувача можна крок за кроком ознайомити з більш складними областями. Однак, для самого навчання користувачів рекомендується обмежити зміст інформації лише найважливішими моментами. Людський мозок здатний засвоювати та зберігати лише певну кількість інформації. Таким чином, важливо захистити користувача від потоку інформації. Надання лише тієї інформації, яка дійсно потрібна, допомагає користувачеві підтримувати огляд, а отже, сприяє швидшому успіху користувача за допомогою отриманих знань. Останній пункт також важливий для загальних інформаційних діалогів у певному програмному застосунку.

Ще один важливий момент, як уже згадувалося в попередньому розділі, – це короткий час реакції на введення користувача. Якщо довший час реакції все ще неминучий, рекомендується повідомити користувача про довший час очікування. Крім того, слід надати точну інформацію про те, скільки часу користувачеві доведеться чекати. Таким чином, користувач може адаптуватися до довший період очікування та виключити можливість збою.

Щоб підвищити відчуття безпеки під час роботи з інтерфейсом користувача, користувач повинен отримувати зворотний зв'язок за виконані дії. Цей пункт також є частиною згаданих у цій роботі *восьми золотих правил Шнейдермана*. Цей зворотний зв'язок забезпечує підтвердження надходження команди, виконаної користувачем, у систему. Відсутність таких зворотних зв'язків швидко призводить до питання про те, чи виконує пристрій бажану дію. Зворотний зв'язок може бути здійснений шляхом короткочасного запалювання кнопки. Тут користувач може візуально отримати підтвердження після активації кнопки. Але зворотний зв'язок також може бути реалізований у вигляді короткого звуку. У інтерфейсах користувача для смартфонів або мобільних телефонів вібрація використовується для короткого підтвердження користувача. Таким чином, користувач може *відчутти* введення. Це використовується, наприклад, під час використання віртуальної клавіатури. Створюючи реакцію

зворотного зв'язку для інтерфейсу користувача, дизайнер повинен подбати про інтенсивність зворотного зв'язку. Неправильно розроблений зворотний зв'язок може призвести до плутанини.

Певні події можуть вимагати втручання або дій користувача. Ці події потребують особливої уваги. Але слід враховувати, як представляється увага до таких подій. Повідомлення про помилку повинно чітко відрізнятися від діалогового вікна попередження. Користувач повинен бути проінформований про дію, яку необхідно виконати, залежно від важливості. У цій роботі обговорюються можливості цих кроків, щоб ретельно попередити користувачів про відповідну інформацію або діалоги.

Прогнозування поведінки користувача за допомогою блок-схем

Блок-схеми пропонують розробнику користувацьких інтерфейсів огляд одного або всіх можливих видів потоків програми. Блок-схема є важливою для створення складної програми, оскільки вона може значною мірою сприяти уникненню помилок під час процесу проектування. Вона дозволяє побачити можливі кроки, які користувач може виконати під час активного використання програми. Блок-схеми зазвичай складаються до технічної розробки користувацьких інтерфейсів, оскільки технічний розробник може використовувати теоретично розроблену діаграму як орієнтир. Блок-схеми також використовуються для внутрішніх процесів програми. Вони також служать орієнтиром для програміста та допомагають уникнути помилок у коді програми [23]. Діаграма UML2 була розроблена спеціально для представлення складних процесів і тому також підходить для графічного представлення можливих процесів у користувацькому інтерфейсі. UML означає *Уніфіковану Мову Моделювання* і також використовується в технічних аспектах для розробки програмних продуктів.

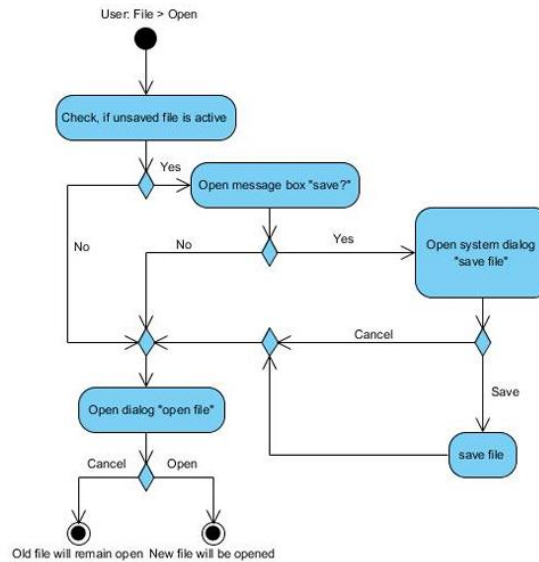


Рисунок 2.2 - Приклад простої діаграми активності

Структурні діаграми, діаграми поведінки та діаграми взаємодії. Далі буде згадано діаграму діяльності, яка належить до сімейства діаграм поведінки. За допомогою діаграми діяльності описуються можливі шляхи в межах одного випадку. Діаграма діяльності складається з таких основних компонентів:

- *Чорна крапка*: використовується для початку робочого процесу (програми)
 - *Заокруглений прямокутник*: У цьому графіці описано дію
 - *Вузли прийняття рішень*: тут можна обрати або один, або інший шлях
 - *Плавлення вузлів*: Об'єднує робочі процеси
- Чорна точка з колом*: використовується для опису кінцевого стану обраного випадку

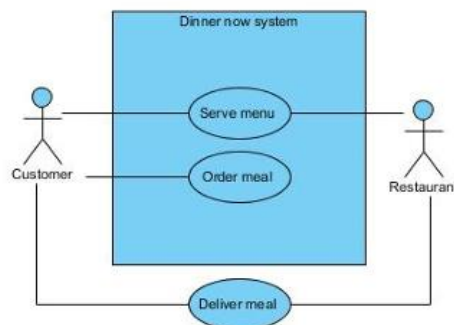


Рисунок 2.3 - Простий приклад діаграми варіантів використання

Іншим типом діаграми, який можна використовувати для прогнозування поведінки користувача, є так звана діаграма *варіантів використання*. Однак цей тип діаграми не такий детальний, як програма дій. Мета цієї діаграми — показати кроки, доступні в системі, загалом. Вона показує, які типи варіантів використання доступні в рамках використання програми. Цей тип діаграми містить такі основні елементи:

- *Система*: Система, в якій беруть участь дії. Цей елемент зображено квадратом.

- *Користувач*: Конкретний користувач, який хоче виконати певне завдання або *варіант використання*.

- *Випадок використання*: Завдання, яке користувач хоче виконати.

Далі буде показано простий приклад діаграми варіантів використання. Цей приклад показує, які кроки робочого процесу необхідні в системі. Зовнішній крок корисний для розробника, щоб побачити, що цей варіант не є частиною системи.

2.3 Аналіз сценаріїв використання та поведінки користувачів

«Розділення інтересів» – це фундаментальна передумова програмної інженерії, призводить до появи кількох поглядів на програмний продукт. Розділення інтересів корисне для розробників програмного забезпечення, якщо взаємодія між елементами системи контролюється [24]. Автори стверджують, що сегментація всієї системи на кілька поглядів, мотивована розділенням інтересів, повинна забезпечити неспотворену загальну картину інтегрованої системи, коли погляди об'єднані. Однак, слід бути обережним, оскільки кілька поглядів можуть надмірно спростити систему без врахування взаємодії елементів системи. Правила композиції необхідно викладати послідовно, оскільки вся картина має стати зрозумілою, коли кілька поглядів об'єднуються в операційну програмну систему. Згідно з UML2.0, існує дев'ять поглядів для опису різних аспектів

програмного забезпечення [25]. Ці погляди: погляд варіантів використання, погляд взаємодії, погляд машини станів, статичний погляд, погляд проектування, погляд діяльності, погляд розгортання, погляд керування моделлю та профіль. Вид зазвичай визначається як підмножина конструкцій моделювання UML, що представляють певні аспекти програмного забезпечення. Кожен вигляд детально пояснюється за допомогою однієї або кількох діаграм, які візуально ілюструють основні особливості вигляду. Вид варіантів використання UML представлений діаграмою варіантів використання для фіксації функцій варіантів використання. Вид варіантів використання добре використовується завдяки ролі, яку варіанти використання відіграють у визначенні аналізу та управління вимогами [26]. Він не використовується належним чином для проектування інтерфейсу користувача в UML, хоча варіанти використання мають великий внесок у інтерфейси користувача. Варіанти використання можуть прояснити багато важливих проблем програмного забезпечення на ранніх етапах процесу розробки, якщо вони належним чином розглянуті в процесі інженерії [27]. Однак в UML робиться спроба дати дуже вузьке визначення вигляду варіантів використання, яке по суті ігнорує природу та значення варіантів використання. «Вигляд варіантів використання моделює функціональність об'єкта (наприклад, системи), як її сприймають зовнішні агенти, які називаються акторами, що взаємодіють з об'єктом з певної точки зору». Сприйняття зовнішніх агентів та взаємодій, згаданих вище, має опосередковуватися через інтерфейс, такий як графічний інтерфейс користувача, особливо коли агенти є людьми. Однак, вигляд варіантів використання UML не враховує інтерфейси користувача або інтерфейси між акторами та варіантами використання. Фактично, не існує представлення UML, яке б належним чином враховувало функції графічного інтерфейсу. Діаграма, що характеризує представлення варіантів використання, – це діаграма варіантів використання, яка представляє основні варіанти використання в рамці з акторами поза рамкою, щоб вказати, що актори є зовнішніми користувачами поточного програмного забезпечення. Одна з центральних проблем діаграми варіантів використання UML полягає в тому, що вона повністю ігнорує інтерфейси з

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

акторами, хоча показано, що кожен актор використовує один або декілька варіантів використання за допомогою лінії або асоціації. Взаємодії між акторами не можуть бути показані на одній діаграмі варіантів використання. Кожен варіант використання представляє послугу, яку можна проілюструвати на діаграмі послідовності UML [28]. Для ілюстрації розглянемо приклад діаграми варіантів використання, показаний на рисунку 2.4.

Наведений нижче опис початкових вимог характеризує початок невеликого програмного проекту: Розробка програмної системи для обчислення площ трьох типів ігрових майданчиків: прямокутних, круглих та трикутних. Підрядник у Лос-Анджелесі будує ігрові майданчики (з таких матеріалів, як дерево, залізо, килимки, пластик тощо) на об'єкті замовника, використовуючи ігрові майданчики різних розмірів. Вартість вказана в доларах, виходячи з площі кожного майданчика в квадратних футах плюс кількість майданчиків. Програмна система необхідна для обчислення вартості, яка базується на площі. Вартість становить: 5,00 доларів США за квадратний фут. Припустимо, що користувачі завжди використовують фути для введення розмірів майданчиків. Для взаємодії з користувачем потрібен графічний інтерфейс користувача (GUI). Щодо цього проекту можна зробити додаткові типові припущення.

Більшість програмних проектів починаються з деяких нечітких вимог. Інженери-програмісти починають свою роботу з початкового аналізу вимог. Після проведення початкового аналізу вимог, інженери-програмісти можуть визначити, що система має бути веб-орієнтованою та доступною цілодобово. Доступ до системи не обов'язково має бути обмежений ідентифікатором для входу. Система має бути легкою в обслуговуванні за допомогою веб-інструментів. Інженери належним чином аналізують функціональні та нефункціональні вимоги. Нарешті, буде підготовлено документ специфікації вимог до програмного забезпечення (SRS); він, як правило, керується варіантами використання. Діаграма варіантів використання для задачі ігрового простору наведена на рисунку 2.4 у стандартних нотаціях UML [29].

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

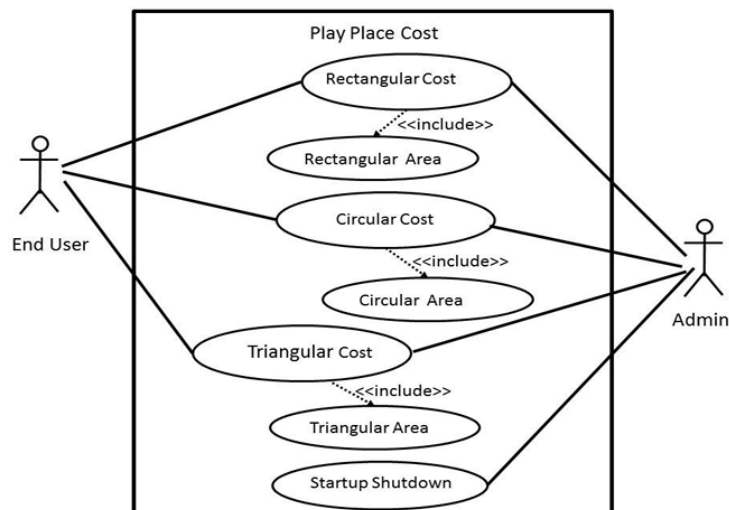


Рисунок 2.4 - Діаграма варіантів використання в UML 2.0

Діаграми варіантів використання UML правильно показують варіанти використання за допомогою овалів у межах системи, представлених прямокутником. Одна з проблем діаграми варіантів використання UML, як-от показана на рисунку 2.4, полягає в тому, що вона ігнорує інтерфейси між акторами та варіантами використання, хоча зображує акторів як фігурки-палички поза межами поточної системи. Наприклад, представляють діаграму варіантів використання для об'єкта під назвою «каса» з чотирма акторами без будь-яких інтерфейсів [30]. Для моделювання функціональності системи, як її сприймають актори, інтерфейси, що відповідають даним акторам, необхідно зобразити на діаграмі варіантів використання. Це дослідження пропонує, щоб відповідні інтерфейси були включені до розширених діаграм варіантів використання. Таким чином, діаграма варіантів використання, наведена на рисунку 2.5, рекомендується для згаданого вище зразка програмного проекту. Важливо зазначити, що інтерфейси показані пунктирними заокругленими прямокутниками на рисунку 2.5.

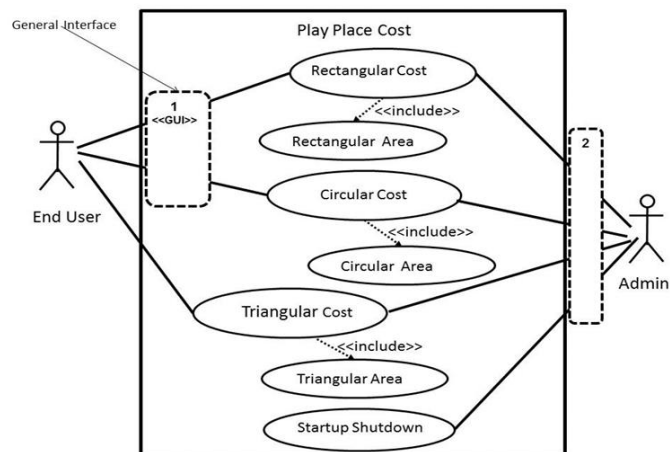


Рисунок 2.5 - Розширена діаграма варіантів використання із загальними інтерфейсами

Щоб посилалися на загальні інтерфейси, вони послідовно нумеруються. Якщо загальний інтерфейс розробляється як частина поточної програмної системи, то він відображається в межах системи; інакше він відображається поза межами системи. Оскільки існує багато різних типів інтерфейсів, деякі з них необхідно позначити за їх важливістю. Якщо інтерфейс є графічним інтерфейсом користувача (GUI), то він позначається терміном <<GUI>> з використанням стереотипів UML [931]. Крім того, коли один загальний інтерфейс включає інший, він може бути позначений відповідним чином. Якщо є третій загальний інтерфейс, який включає перший, то в третьому інтерфейсі можна відобразити «3 D 1».

Наявність загальних інтерфейсів на діаграмі варіантів використання інтуїтивно та логічно підтримує ідею про те, що сприйняття користувача щодо функціональності моделюється належним чином у розширеному поданні варіантів використання. Коли учасником є людина, загальний інтерфейс може бути графічним інтерфейсом для відповідної взаємодії між користувачем та системою. Для інтерактивних систем додавання графічних інтерфейсів до діаграми варіантів використання допомагає зрозуміти сприйняту функціональність системи. Саме роль графічних інтерфейсів недостатньо детально описана в методах моделювання UML, що призводить до високого ступеня плутанини під час розробки сучасних інтерактивних систем.

Окрім діаграм варіантів використання, розширене представлення варіантів використання повинно мати загальні діаграми інтерфейсів. Без такої діаграми проблеми щодо користувацьких інтерфейсів грубо ігноруються, а взаємодія між елементами системи не враховується належним чином. Без діаграм інтерфейсів стандартний UML [32] не містить інформації, життєво важливої для успіху сучасної програмної системи. Він також не дає вичерпного опису програмного забезпечення, яке, як очікується, є композицією стандартних представлень UML. Доцільно бути гнучким щодо позначень загальних діаграм інтерфейсів, особливо якщо це графічні інтерфейси користувача. Двома основними альтернативними позначеннями для загальної діаграми інтерфейсу є (1) знімки екрана з прототипу та (2) абстрактне графічне представлення основних елементів інтерфейсу. Ми показуємо попереднє позначення на загальній діаграмі інтерфейсу, наведений на рисунку 2.4, для загального інтерфейсу рисунка 2.5. Тобто, ми розробили прототип графічного інтерфейсу користувача, використовуючи мову програмування Java, для прикладу задачі ігрових майданчиків та зробили знімок екрана графічного інтерфейсу користувача. Цілком логічно припустити, що з кожною наступною ітерацією графічний інтерфейс користувача рисунка 2.6 розвиватиметься та набуватиме кращих якостей.

Загальні діаграми інтерфейсів, слід вважати важливими для цілей проектування програмного забезпечення. Джейсон Хонг [33] ставить важливе питання: «Як нам ефективно впроваджувати чудовий дизайн у продукти?» Наразі ми не можемо впроваджувати дизайн графічного інтерфейсу у стандартні методи на основі UML. Роль UML у моделюванні може бути посилена шляхом належного врахування сприйнятої функціональності системи шляхом надання розширеного представлення варіантів використання разом із загальними діаграмами інтерфейсів. Це вірно, тому що розширене представлення варіантів використання включає загальні інтерфейси у свою діаграму варіантів використання між акторами та варіантами використання. Сприйнята функціональність очевидно сприймається акторами, коли відповідні асоціативні

зв'язки проходять через загальні інтерфейси на діаграмах варіантів використання.

Баланс між абстракцією та деталями може бути належним чином досягнутий на загальній діаграмі інтерфейсу, оскільки елементи інтерфейсу можна додавати поступово. «Програмісти та розробники програмного забезпечення часто є компетентними користувачами технології... Однак занадто часто вони не використовують цю технологію належним чином і створюють інтерфейси користувача, які є неелегантними, недоречними та складними у використанні» [34]. Розширений вигляд варіантів використання робить додатковий акцент на моделюванні інтерфейсів користувача. Це сприяє зосередженню на багатьох інших аспектах інтерфейсів користувача, таких як підтримка однакових механізмів введення в усьому застосунку. Ніхто не повинен стверджувати, що інтерфейси адекватно обробляються в поданні проектування UML і що розширення подання варіантів використання не є необхідним, оскільки подання проектування просто розміщує надані та необхідні інтерфейси з відповідними компонентами. Додатковий акцент потрібен для відображення деталей інтерфейсів певних типів, таких як графічні інтерфейси користувача (GUI). Моделювання графічних інтерфейсів користувача для інтерактивних систем стало дедалі важливішим за останні два десятиліття [35]. Розділення обов'язків мотивує модульне проектування, де програмна система розкладається на компоненти; однак, серед компонентів необхідно вказати чітко визначені інтерфейси. Для взаємодії людини з компонентами можуть знадобитися графічні інтерфейси користувача (GUI). Основна плутанина з UML полягає в тому, що її представлення аспектів програмного забезпечення повністю ігнорує графічні інтерфейси користувача (GUI). Мова візуального моделювання, така як UML, не може досягти своїх основних цілей без належної уваги до дизайну графічного інтерфейсу користувача (GUI). Крім того, навчання програмній інженерії з використанням UML вимагає керівництва для учнів, щоб різні подання разом могли композиційно визначити повну програмну систему [36]. Через відсутність таких елементів, як графічні інтерфейси користувача (GUI), UML надає

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

фрагментарне уявлення про програмне забезпечення, яке є неадекватним для будь-якого опису інтегрованої цілісної системи. Запропонований розширений погляд на варіанти використання розроблений для заповнення прогалини. Міркування з розширеним поглядом на варіанти використання кращі, ніж з традиційним поглядом на варіанти використання, оскільки функціональність системи, як її сприймають учасники, є більш розумною завдяки включенню загальних інтерфейсів, згаданих вище. Інженерні практики та проектна діяльність із загальними конструкціями інтерфейсів також можуть заохочувати та сприяти вивченню інтерфейсів користувача, що є цінним для студентів в освітніх установах та академічних середовищах.

2.4 Висновок по розділу

У другому розділі досліджено методи та інструменти розробки інтерфейсів користувача. Розглянуто основні принципи проектування, зокрема простоту, послідовність, адаптивність і орієнтацію на потреби користувача. Проаналізовано механізми забезпечення зворотного зв'язку та методи вивчення поведінки користувачів під час роботи із системою. Визначено, що застосування сучасних підходів до проектування інтерфейсів сприяє підвищенню ефективності взаємодії між користувачем і програмним продуктом.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЯКОСТІ ІНТЕРФЕЙСІВ

3.1 Стандарти якості програмного забезпечення та інтерфейсів

Один із способів досягнення важливого підходу до ефективності – це зменшення складності. Користувацький інтерфейс має бути розроблений таким чином, щоб забезпечити користувачеві найпростіший і найкоротший шлях виконання завдання без будь-яких додаткових кроків. «Ефективність описує

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

зручність використання дизайну». Важливим моментом для підвищення ефективності користувацького інтерфейсу є його *простота*. Легкий для розуміння користувацького інтерфейсу зменшує навантаження на користувача, пов'язане з розумінням самого інтерфейсу. Цей крок водночас означає зменшення кількості функцій інтерфейсу користувача, щоб представити лише ті опції, які дійсно необхідні. Якщо потрібні додаткові опції, слід перевірити, чи дійсно вони потрібні в цій ситуації. Оптимізований користувацький інтерфейс повинен мати можливість передбачати *поведінку* користувача та надавати інформацію про кроки, які користувач *може* вибрати. Збереження узгодженості в інтерфейсі користувача є ще одним важливим фактором підвищення ефективності [37]. Якщо користувацький інтерфейс розроблено узгоджено, користувач може використовувати знання з інших завдань, які виконувалися з тим самим інтерфейсом користувача, для виконання нового завдання. Таким чином, процес навчання користувацького інтерфейсу може бути скорочений.

Існує п'ять різних способів ефективного вимірювання зручності використання користувацького інтерфейсу. По-перше, важливий час, який користувачеві потрібно для вивчення функціональності користувацького інтерфейсу. Користувач повинен виконувати спеціальні операції для створення даних для результату. Також швидкість виконання завдань є важливим моментом для створення зворотного зв'язку щодо зручності використання користувацького інтерфейсу. По-третє, вказують на важливість кількості помилок, які користувач допускає під час виконання завдань для вимірювання. Щоб вдатися до деталей, можна задокументувати тип помилок. Крім того, регресію щодо знань користувача щодо використання користувацького інтерфейсу можна спостерігати протягом тривалішого періоду часу. Останній пункт, задоволеність користувача, може бути використана як критерій для вимірювання зручності використання. Останній пункт можна реалізувати за допомогою інтерв'ю або анкетування користувача.

Задані стандарти якості програмного забезпечення

Для забезпечення певного стандарту якості також було встановлено

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

стандарти в цій галузі. Ці стандарти мають забезпечити дотримання принципу якості програмного забезпечення з точки зору зручності використання.

ISO 9241

Цей стандарт був створений для встановлення рекомендацій щодо специфікації ергономіки програмного забезпечення. Його не слід розуміти як суто технічний стандарт, але він надає рекомендації з критеріями, які повинні гарантувати певний рівень зручності використання в рамках користувацького застосунку. Цей стандарт необхідно інтерпретувати, і він включає такі основні характеристики зручності використання програмного забезпечення:

- Придатність для завдання*: це передбачає належну функціональність програмного забезпечення порівняно з відповідними функціями та можливостями, необхідними користувачеві для досягнення цілей, важливих для користувача.

- Придатність для навчання*: Цей показник описує навчальну підтримку самої програми. Він визначає, як програма сприяє розумінню самої програми та її інтерфейсу користувача за допомогою вбудованих функцій.

- Придатність для індивідуалізації*: Тут визначено адаптацію інтерфейсу до індивідуальних потреб користувача.

- Відповідність очікуванням користувача*: Цей пункт має забезпечити відповідність інтерфейсу користувача очікуванням. Це може вплинути, наприклад, на сферу діяльності користувача, а також на його досвід.

- Самоопис*: це забезпечує допомогу та підтримку користувача в рамках програми, наприклад, шляхом надання йому зворотного зв'язку.

- «Керованість»*. У цьому пункті згадується, наскільки користувач може керувати інтерфейсом користувача.

- Допустимість помилок*: Цей пункт відображає працездатність програми у разі непередбачених подій. Серед іншого, ці події можуть включати неправильне введення даних користувачем та помилки програми.

ISO 9126

Ще одним ресурсом для забезпечення зручності використання

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

програмного забезпечення є стандарт ISO 9126. Перш за все, цей стандарт був створений для забезпечення загальної якості програмного забезпечення. Але він також включає критерії зручності використання програми загалом. Це дозволяє застосовувати цей стандарт також для процесу проектування інтерфейсу користувача. Стандарт включає, серед іншого, такі пункти, важливі для проектування інтерфейсу користувача:

•*Відмовостійкість*: Тут відмовостійкість програми розглядається загалом. Для процесу створення користувацького інтерфейсу це означає толерантність до помилкового введення з боку користувача.

•*Зручність використання*: Тут основна увага приділяється зусиллям, необхідним для вивчення програми. Також зрозумілість інтерфейсу користувача є складовою цього пункту. Аналогічно, в цьому пункті згадуються загальні зусилля, необхідні користувачеві щодо працездатності програми.

Так звані вісім золотих правил надають узагальнення інших дуже важливих критеріїв, пов'язаних з проектуванням користувацького інтерфейсу. Ці правила базуються здебільшого на досвіді та не є стандартизованими порівняно з вищезгаданими стандартами ISO. Однак вони забезпечують додаткову основу для орієнтації в процесі проектування. Правила написані для загальних цілей і можуть потребувати подальшого коригування для конкретних областей:

•*Прагнїть до узгодженості*: Досягнення очікуваної узгодженості в інтерфейсі користувача служить користувачеві певним стандартом у різних областях програми. Таким чином, користувач може бути впевнений, що зустріне знайомі функції в програмі. Це забезпечує певне відчуття безпеки для користувача під час роботи з інтерфейсом користувача.

•*Забезпечення універсальної зручності використання* : слід спробувати розробити інтерфейс користувача, який підходить для різних груп користувачів. Це включає, наприклад, наявність комбінацій клавіш для досвідчених користувачів. Останній згаданий пункт дозволяє швидше переміщатися в інтерфейсі користувача. У більшості випадків користувачі-початківці не відчувають потреби в доступних комбінаціях клавіш, тому потрібен інший

спосіб підтримки користувача щодо зручності використання інтерфейсів користувача. Додаткова графічна допомога, що надається інтерфейсом користувача, може бути вирішальним фактором.

•*Пропонуйте інформативний зворотний зв'язок* : для кожного кроку, виконаного користувачем, має бути запропоновано певний вид підтвердження. Для досвідчених користувачів це підтвердження має бути скромним. Тому новачкам або користувачам, які не дуже часто стикаються з майбутнім завданням, потрібне чіткіше підтвердження виконаної дії.

•*Розробка діалогових вікон для завершення угоди*: Інтернет-магазини, такі як Amazon.com, приділяють цьому великому акценту. Користувачеві допомагають детальні діалогові вікна, і таким чином він може бачити поточний етап процесу та крок поточної операції. Це полегшує навігацію для користувача, а також забезпечує певне відчуття безпеки.

•*Запобігання помилкам*: Виникнення помилок через неправильне введення користувачем слід уникати з самого початку. Приховування недоступних наразі пунктів меню є одним із можливостей вирішення цієї проблеми. Якщо користувач виконує неправильне введення, наприклад, за допомогою клавіатури, допоміжні механізми повинні інформувати користувача про помилку у зручний для користувача спосіб.

•*Дозволити легке скасування дій* : Цей пункт дуже важливий для забезпечення безпеки користувача під час роботи з інтерфейсом користувача. Знання користувача про можливість скасування введених даних призводить до подальшого ознайомлення з програмою та до безпечнішого використання.

•*Підтримка внутрішнього локусу контролю*: Більш досвідчені користувачі вітають відчуття контролю над інтерфейсом користувача. Тому слід уникати неочікуваних дій інтерфейсу користувача, оскільки вони можуть створювати відчуття неконтрольованості та, таким чином, призвести до невдоволення.

•*Зменшення навантаження на короткочасну пам'ять* : У цьому пункті згадується базовий простий дизайн утримання. Це також включає надання лише

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

необхідних функцій у інтерфейсі користувача. Дизайн дуже інтуїтивно зрозумілих інтерфейсів користувача може зробити тут суттєвий внесок.

Розробка інтерфейсу користувача для певної цільової групи

Під час створення програмного забезпечення для певної групи користувачів, наприклад, для певних відділів у компанії, необхідно ретельніше дослідити спільні завдання групи. Подібно до вивчення поведінки окремих користувачів, це включає визначення сфери діяльності відділу, а також визначення операційної області програмного забезпечення. На цьому етапі також можна побачити різницю між професійними користувачами, наприклад, користувачами в компанії, та початківцями. Програмне забезпечення для професійних користувачів може мати набагато більше варіантів конфігурації, ніж програмне забезпечення для домашніх користувачів, оскільки корпоративне програмне забезпечення використовується часто і тому швидше вивчається. Чотири найпоширеніші групи користувачів за функціями: кінцеві користувачі, керівні користувачі, користувачі-клієнти, системні користувачі. Кінцеві користувачі безпосередньо взаємодіють з комп'ютерною системою для виконання своїх робочих процесів або вирішення особистих питань. Крім того, кінцевий користувач несе відповідальність за успіх даного програмного забезпечення або пристрою. Таким чином, це також може вплинути на успіх відділу або компанії, оскільки прийняття кінцевими користувачами, якими в цьому випадку є співробітники, також впливає на продуктивність роботи. Підсумовуючи, група кінцевих користувачів потребує особливої уваги під час проектування інтерфейсу користувача. Керівники-користувачі отримують вигоду від успіху певної системи або програмного забезпечення. У компанії менеджери мають уявлення про ефективність роботи, а отже, і про ефективність доступного програмного забезпечення, з яким працюють співробітники. Саме керівники виявляють потребу в новому програмному забезпеченні або новій системі. Користувачі-клієнти менше працюють безпосередньо з системою інформаційних технологій. Вони стикаються з вхідними або вихідними даними системи, отриманими від інформаційних технологій. Користувачі системи

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

належать до частини професійних користувачів. Тому вони керують та адмініструють різні системи, вони звикли працювати з системою та можуть швидше адаптуватися до нового програмного забезпечення або систем. Іноді вони можуть взаємодіяти з програмним забезпеченням або пристроями, як це роблять кінцеві користувачі.

Інший спосіб визначення групи користувачів – це сортування користувачів за рівнем їхньої кваліфікації. Найпростіший спосіб це зробити – поділити користувачів на дві групи: початківці та професіонали. Різниця між цими двома групами користувачів має бути легкою для розуміння. Однак, новачок може досягти рівня професійного користувача шляхом частого використання доступного програмного забезпечення або пристрою. Також користувач, який належить до категорії початківців, може бути професійним користувачем в іншій галузі інформаційних технологій.

Оцінити користувацький інтерфейс

Оцінювання користувацького інтерфейсу є важливим, оскільки сам користувач надає дизайнеру іншу точку зору. Під час оцінювання користувацького інтерфейсу можна використовувати інформацію, надану в цій роботі. Тут дуже важливо, щоб дизайнер об'єктивно виконував процес оцінювання. Загалом, рекомендується використовувати допомогу третьої сторони для огляду. Оцінювачі повинні бути знайомі з процесом оцінювання, щоб досягти кращого результату. Оцінювання також може бути проведене безпосередньо за допомогою відповідного зворотного зв'язку від користувача. Для цієї мети, наприклад, може бути корисним створення анкети. Анкета може бути спеціально зосереджена на ключових операційних критеріях для користувача і таким чином надавати інформацію з точки зору користувача щодо користувацького інтерфейсу. Також оцінювання може бути проведене під час оперативного використання користувацького інтерфейсу. Програмне забезпечення, яке необхідно постійно змінювати, є одним із прикладів використання такого методу. ERP-програмне забезпечення є одним із прикладів цієї категорії, оскільки його необхідно постійно змінювати та адаптувати до

робочих процесів компанії. Оцінка існуючого користувацького інтерфейсу також важлива для дизайнера щодо процесу проектування користувацьких інтерфейсів у майбутньому. Таким чином, творець може бути впевнений, що обирає правильний шлях.

3.2 Проектування та реалізація графічного інтерфейсу користувача

Поведінкою користувача в інтерфейсі користувача можна опосередковано керувати шляхом вибіркового використання певних кольорів. Однак використання кольорів слід проводити економно. Через надмірне використання кольорів користувач може бути заплутаний. Також з точки зору мистецтва, непродумане використання кольорів може призвести до незначного дизайну. За допомогою правильного використання кольорів можна підвищити ефективність інтерфейсів користувача. Тут кольори можуть служити як орієнтиром, так і орієнтацією. Вони також можуть запобігти можливій помилці користувача під час обробки. Сигнал помилки кольором, який відомий як колір попередження, може привернути увагу користувача до проблеми, яка може виникнути через наступний крок користувача. Тут червоний колір відіграє важливу роль. Ми знаємо цей колір з дороги та її дорожніх знаків. Червоний сигналізує нам, що нам потрібно звернути особливу увагу на цей знак або це повідомлення. Іншими прикладами є світлофори. Тут червоний колір сигналізує нам про зупинку. Також жовтий колір можна пояснити на основі цього прикладу. Він використовується для попереджень, наприклад, в операційній системі Windows. Жовтий колір зазвичай попереджає про можливу критичну ситуацію та закликає користувача до дії. Однак зелений колір вказує на нормальний перебіг процесу. Втручання користувача не потрібне. Також тут, порівняно зі світлофором, зелений колір показує нормальний потік руху. Білий колір часто вважається нейтральним і відіграє меншу роль в дизайні інтерфейсів користувача. Так само сірий колір не має особливого статусу, але часто

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

використовується як колір для недоступного пункту меню в різних операційних системах.

Окрім основних моментів, згаданих у розділі вище, культурні аспекти також відіграють важливу роль у використанні кольорів. Залежно від культури, кольори можуть асоціюватися з різними значеннями та переживаннями. Також використання кольорів може викликати різні емоційні аспекти. Але існують не лише відмінності між культурами. Найкращим прикладом тут є запитання різних людей про улюблений колір та асоціації з ним. Результатом будуть різні асоціації для певного кольору. Але також аспекти корпоративної ідентичності можуть бути точкою, як люди сприймають кольори. Одним із прикладів є Apple, який багато людей асоціюють з білим кольором, оскільки цей колір має відношення до чистоти та недоторканності. Крім того, Apple є гарним прикладом використання кольорів у своїй попередній маркетинговій стратегії. Щоб привернути увагу потенційних клієнтів, Apple використовувала кольори веселки у своєму логотипі, оскільки ці кольори на той час були символом гомосексуальності. Apple використовувала цей аспект, щоб рекламувати, що ця компанія мислить інакше.

Щодо використання кольорів, є важливі моменти, які слід враховувати під час розробки користувацького інтерфейсу [38]. Тут головну роль відіграє узгодженість під час використання. Попереднє порівняння використання кольорів з кольорами, доступними в дорожньому русі, також може бути використано для опису узгодженості. Постійне використання однієї й тієї ж колірної схеми може уникнути плутанини та дозволить користувачеві швидко реагувати на заданий колір. Відповідна чутливість під час використання кольорів також може призвести до більшої мотивації користувача. Однак використання кольорів має здійснюватися економним способом. Для досвідчених користувачів занадто велика кількість кольорів не повинна призводити до проблеми.

Завдяки використанню аудіо в процесі проектування інтерфейсу користувача доступно значно менше ресурсів порівняно з керуванням кольором. Крім того, використання аудіо має бути обережним. Не кожен пристрій має

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідний аудіоінтерфейс. Використання аудіо в інтерфейсах користувача, що підпадає під категорію цієї роботи, зазвичай є лише додатковою функцією. Використання аудіо використовується в основному для привернення уваги користувача. Однак, якщо доступний відповідний аудіоінтерфейс, аудіо може бути використано для привернення додаткової уваги користувача. Повідомлення про помилку може бути супроводжено чітким попереджувальним звуком, тоді як просте повідомлення повинно бути надано більш м'яким тоном. Як і у випадку з використанням кольорів, використання аудіо під час проектування інтерфейсу користувача повинно здійснюватися таким чином, щоб уникнути плутанини у користувача. Таким чином, тут також застосовується аспект економічної реалізації.

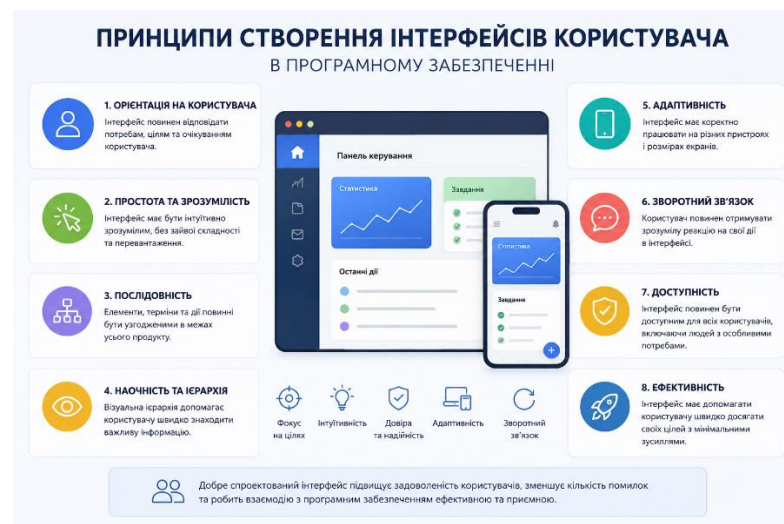


Рисунок 3.1 – Принципи створення інтерфейсів користувача

Зовнішній вигляд та відчуття: Використання вібрацій

З появою мобільних пристроїв, наприклад, смартфонів, з'явилися нові можливості для керування користувачем через інтерфейс користувача. За допомогою невеликої вібрації пристрою користувач може відчути його поточний стан. Вібрація використовується тут здебільшого для підтвердження дії або, більшою мірою, для привернення уваги користувача, наприклад, у разі вхідного дзвінка.

Використання вібрації вже мало місце у світі електронних розваг. Представлена в 1997 році ігрова консоль *Nintendo 64* від однойменної компанії

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Nintendo могла створювати вібрації за допомогою так званого *Rumble Pack*. Цей пристрій був доступний як підключаємий пристрій для контролера консолі та використовувався для підтвердження дій, виконаних гравцем. Використання вібрації дуже обмежене, оскільки її можна змінювати лише в невеликій мірі. Однак і тут слід враховувати економне використання, щоб уникнути плутанини або навіть лякання користувача.

Проектування користувацького інтерфейсу для конкретного пристрою

Неможливо створити універсальний інтерфейс користувача для кожного доступного пристрою. Кожен пристрій має специфічні характеристики, які вимагають спеціального інтерфейсу користувача. Далі представлено різні варіанти пристроїв та їхні важливі аспекти для процесу розробки інтерфейсу користувача для цих пристроїв.

Успішна історія сучасних настільних комп'ютерів розпочалася в 1964 році, коли була розроблена перша операційна система для використання з мишею та клавіатурою. Трохи пізніше були продані перші настільні комп'ютери, серед іншого, Apple I та Apple II. Можливості, які відкривалися завдяки настільним комп'ютерам, з того часу неухильно зростали. Особливо поява Інтернету та загальних глобальних мереж сприяла зростанню успіху настільних комп'ютерів. Сьогодні цей тип комп'ютерів можна знайти в багатьох сферах нашого повсякденного життя. Це включає наш власний дім чи наше робоче місце. Комп'ютер став невід'ємною частиною повсякденного життя. За допомогою комп'ютерів стало можливим писати тексти, коригувати їх та спочатку переносити на папір після досягнення задовільного рівня письмового змісту. Сьогодні комп'ютер пропонує нам більше можливостей, особливо в галузі комунікації. Для використання комп'ютера зазвичай потрібні клавіатура, миша та екран як основне обладнання. Перевагами комп'ютерів, доступних сьогодні, є висока технічна продуктивність, а також можливість відображення контенту на широкому екрані. Останнє стосується порівняння з мобільним ноутбуком, що обговорюється пізніше в цій роботі. Настільний комп'ютер не призначений для мобільного використання. Під час проектування інтерфейсу

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

користувача для настільного комп'ютера доступно багато опцій та можливостей. Завдяки наявності клавіатури та миші для інтерфейсу користувача можна використовувати велику кількість опцій керування. Наприклад, для досвідчених користувачів можливо створювати ярлики для програм, що дозволяє відмовитися від операцій за допомогою миші. Це забезпечує швидшу роботу для досвідчених користувачів в інтерфейсах користувача таким чином, призводить до вищого рівня мотивації. Також сучасні настільні системи забезпечують більшу роздільну здатність екрана, що, наприклад, дозволяє розміщувати більше пунктів меню в інтерфейсі користувача. Розробник інтерфейсу користувача для комп'ютера також може бути впевнений, що він не буде розроблений для мобільного використання. Настільний комп'ютер зазвичай має фіксоване розташування і тому, як згадувалося вище, не призначений для мобільного використання. Загалом, під час створення інтерфейсу користувача для настільного комп'ютера доступно більше можливостей дизайну порівняно з процесом проектування для інших пристроїв. Але й тут економічне та розумне використання наданих можливостей є важливим.

Ноутбук зазвичай не має такої ж продуктивності, як настільний комп'ютер, оскільки мобільність є важливим моментом, а низькопродуктивні компоненти споживають менше енергії. Використання ноутбука подібне до використання настільного комп'ютера. Тільки під час мобільного використання потрібне використання тачпада. Тут можлива навігація, подібна до миші, зі зниженою швидкістю. Оскільки ноутбуки, як уже було описано, мають подібну архітектуру до настільних комп'ютерів, рекомендується аналогічний підхід для проектування інтерфейсу користувача. Також додаток має бути сумісний для використання на настільному комп'ютері. Винятком є Apple MacBook, який за допомогою так званої *функції мультитач від Apple* дозволяє керувати кількома пальцями на тачпаді. Тут методи роботи для Macbook розширені так званими жестами. Залежно від того, яку комбінацію жестів розпізнає користувач, буде виконано певну дію. При проектуванні інтерфейсу користувача це дозволяє розширити можливості щодо використання інтерфейсу користувача.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Планшетний комп'ютер відрізняється від останніх згаданих сімейств пристроїв. Планшетний комп'ютер в першу чергу орієнтований на використання мультимедійного контенту. Планшет оснащений спеціально адаптованою для цього пристрою операційною системою і тому вимагає менш потужної апаратної бази [39]. Тут також важливо враховувати рекомендації, опубліковані виробником, щодо технічних деталей. Інтерфейс користувача для програми для планшетного комп'ютера повністю відрізняється від розробки інтерфейсу користувача для настільного комп'ютера або ноутбука. Планшет безпосередньо керується через сенсорний екран і, таким чином, не має миші чи клавіатури. Оскільки ви можете керувати інтерфейсом користувача лише за допомогою пальців людини шляхом безпосереднього контакту з сенсорними екранами, це також важливий момент, який необхідно враховувати. Планшет розроблений для мобільного використання і тому має обмежені енергетичні ресурси. Тому під час процесу проектування необхідно враховувати обмеження доступних ресурсів пристрою. З технічної точки зору це означає оптимізацію механізмів, що потребують обчислювальної потужності.

Використання смартфона схоже на використання планшета. Смартфон пропонує перевагу об'єднання кількох функцій з багатьох інших пристроїв в одному пристрої. Це включає можливість здійснювати дзвінки, а також можливість отримувати інформацію з Інтернету. Порівняно з планшетом, екран смартфона має менший розмір. Як і у випадку з планшетом, є можливість встановлення програм, якщо основних функцій смартфона недостатньо. Іншими важливими аспектами є мобільність смартфона та можливість використання пристрою для мультимедійних програм. Критерії створення інтерфейсу користувача аналогічні критеріям планшетного ПК. Однак і тут слід враховувати слабке апаратне забезпечення порівняно з ноутбуком або настільним ПК. Як і у випадку з планшетом, смартфон створений для мобільного використання, що також означає створення програм з ефективним управлінням енергоспоживанням. Робота нових смартфонів також здійснюється переважно за допомогою сенсорного екрана. Розмір екрана порівняно з планшетом дуже

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

малий, тому структура меню програми повинна бути компактною. Це допомагає зменшити перевантаження екрана та таким чином уникнути плутанини для користувача.

Ще один пристрій, який буде розглянуто в цій роботі, – це мобільний телефон. Мобільний телефон можна розглядати як попередника смартфона. Мобільний телефон призначений, перш за все, для здійснення дзвінків, але з часом він був оснащений додатковими функціями. Але порівняно зі смартфоном, функції мобільного телефону надають менше можливостей. Як і смартфон, мобільний телефон надає можливість надсилення коротких повідомлень. Однак тут це розглядається як другорядна важлива функція. Однією з таких функцій є можливість надсилення коротких повідомлень. Короткі повідомлення обмежені 160 символами на повідомлення. Це обмеження існує й сьогодні, але його можна обійти, об'єднавши кілька коротких повідомлень в одне велике повідомлення. Після того, як у 1999 році був представлений програмний інтерфейс Java Micro Edition (Java ME), новіші покоління смартфонів змогли встановлювати програми Java. Мобільний телефон має дуже маленький дисплей, і лише деякі пристрої надають можливість використовувати пристрій через сенсорний екран. Тому для введення даних доступна лише цифрова клавіатура. Через малий розмір екрана спектр інформації, яка може відображатися на екрані, також дуже обмежений. Для дизайнера користувацького інтерфейсу це означає відображення лише тієї інформації, яка дійсно потрібна. Крім того, необхідно враховувати можливе обмеження кількості кольорів, які може відображати пристрій.

Найвідомішим пристроєм введення з часів появи настільних комп'ютерів залишається клавіатура. Цей пристрій введення використовується переважно для введення тексту. Але його також можна використовувати для введення команд для дій, які має виконувати комп'ютер. 10-пальцева система письма дозволяє швидше вводити текст. Через особливості спеціальних символів або літер, характерних для різних країн, розташування клавіш може відрізнятися. Але це не впливає на основні функції клавіатури.

Під час розробки інтерфейсу користувача, який в першу чергу

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

призначений для використання з клавіатурою, метод введення можна спростити, реалізувавши комбінації клавіш. Але слід переконатися, що ці комбінації клавіш не можуть бути випадково активовані неспідготовленими користувачами. Гарною можливістю тут є використання комбінацій клавіш у поєднанні зі стандартними функціональними клавішами, такими як клавіші CTRL або ALT, які доступні на більшості сучасних стандартних клавіатур IBM. Додаткові можливості для виконання спеціальних наказів пропонують F-клавіші, які можна знайти у верхній частині клавіатури. Клавіатури ноутбуків часто також мають так звані гарячі клавіші. Вони оснащені функціями, специфічними для виробника ноутбука, і зазвичай недоступні для інших програм. Особливо для досвідчених користувачів клавіатура пропонує можливість швидше вводити дані. Для професійних користувачів так звані *мнемоніки* пропонують швидший метод вибору шляхом вибору пункту меню з доступними комбінаціями клавіш. Це також створює невелику перевагу в швидкості порівняно з мишею як пристроєм введення.

Комп'ютерна миша була розроблена як своєрідна запасна рука для безпосереднього введення даних у графічному інтерфейсі користувача. Таким чином, користувач міг налаштовувати швидкість руху вказівника миші відповідно до власних потреб. Залежно від доступних клавіш миші, розробник міг бути обмежений у їх використанні [40]. У той час як миші, створені для комп'ютерів, призначених для роботи з операційною системою Windows, мають щонайменше дві клавіші миші, офіційна миша для комп'ютерів Apple має лише одну клавішу. На додаток до цих кнопок було розроблено колесо прокручування. За допомогою колеса прокручування можна дозволити прокручування у віконних програмах. Залежно від виробника миші, колесо прокручування можна використовувати як третю кнопку миші.

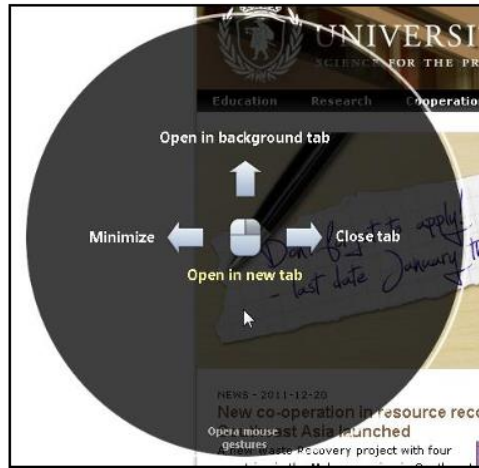


Рисунок 3.2 - Використання жестів у веб-браузері Opera

Ще одним розвитком у сфері керування мишею є використання так званих жестів миші, які були винайдені у 2000 році норвезькою компанією Opera Software ASA. Тут програма ідентифікує жест, що виконується в даний момент, а потім виконує потрібну дію. Однією з переваг жестів миші є швидкість: жест миші виконується швидше, ніж пошук і натискання дії за допомогою звичайної операції миші.

Недоліком жесту є відсутність можливості знайти доступний жест миші видимим способом. Порівняно з доступною кнопкою, жест миші не видно. Таким чином, жести миші забезпечують додатковий спосіб керування інтерфейсом користувача, особливо для досвідчених користувачів.

Від вищезгаданих пристроїв введення сенсорний екран відрізняється своєю роботою для користувацького інтерфейсу. Сенсорний екран здебільшого використовується у смартфонах та планшетних ПК. В інших галузях, наприклад, у медичній галузі, сенсорні екрани також використовуються для керування спеціальними медичними пристроями. Сенсорний екран дає користувачеві відчуття можливості безпосередньо маніпулювати пристроєм, оскільки немає потреби в додатковому управлінні пристроями. Сенсорний екран керується лише людськими пальцями і тому вимагає іншої операційної концепції, ніж, наприклад, пристрій з клавіатурою та мишею. На відміну від роботи за допомогою миші, палець не такий точний, як миша, для керування інтерфейсом

сенсорного екрана. Це означає, що дизайнеру доводиться створювати більші значки для керованих частин користувацького інтерфейсу замість використання маленьких значків, оптимізованих для вказівника миші. Сенсорний екран також дозволяє використовувати жести мультитач. За їх допомогою бажану реакцію програми можна виконати, використовуючи більше одного пальця під час введення. Деякі з відомих функцій, наприклад, - це збільшення тексту або поворот зображень. Недоліком цього типу роботи є обмежена можливість введення тексту. Сенсорний екран справді можна використовувати як віртуальну клавіатуру, але особливо для користувачів, які використовують 10-пальцеву систему, він не забезпечує такої ж швидкості введення тексту, як звичайна клавіатура. Одним із способів протидії цьому є вгадування системою слова, яке користувач хоче ввести. Після введення перших символів слова користувачеві надається кілька варіантів слів. Таким чином, введення всіх символів для завершення слова іноді можна пропустити. Це також є перевагою щодо часу введення.

Подібна концепція керування використовує можливість введення T9, якщо пристрій введення має лише цифрову клавіатуру. Концепція T9 була розроблена для використання з мобільними телефонами. Цифрова клавіатура обмежує можливості введення порівняно з клавіатурою. Таким чином, загалом доступні лише цифри 0-9 та символи * та #. T9 призначає кожній з існуючих цифр набір літер, що дозволяє вводити текст. Недоліком T9 є те, що можна пропонувати лише добре відомі слова. Щоб протидіяти цьому, сучасні системи T9 мають здатність до *навчання*. Це означає, що якщо слово знадобиться знову в певний момент, воно буде запропоновано системою, якщо комбінація цифр збігається. Під час створення інтерфейсу користувача для пристрою з цифровою клавіатурою розробник може покладатися на використання швидких команд, щоб зробити роботу з пристроєм зручнішою. Мобільні телефони зазвичай також мають своєрідну клавіатуру або щось подібне, що дозволяє додаткові опції керування.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

3.3 Оцінка відповідності інтерфейсу загальним принципам UX/UI

Нижче описано цілі проектування при створенні користувацького інтерфейсу. Вони є фундаментальними для проектування та реалізації всіх ефективних інтерфейсів, включаючи графічні інтерфейси користувача та веб-інтерфейси. Ці принципи є загальними характеристиками інтерфейсу та застосовуються до всіх аспектів. Вони походять з різних принципів.

Естетично приємний

Естетика дизайну, або візуально приємна композиція, приваблива для ока. Вона привертає увагу підсвідомо, чітко та швидко передаючи повідомлення [14]. Візуальна привабливість забезпечується шляхом дотримання принципів презентації та графічного дизайну, які будуть обговорюватися, включаючи забезпечення значущого контрасту між елементами екрана, створення просторових груп, вирівнювання елементів екрана, забезпечення тривимірного представлення та ефективне використання кольору та графіки. Гарний дизайн поєднує потужність, функціональність та простоту з приємним зовнішнім виглядом [14]. Графік, показаний на рисунку 3.2, показує відсотки задоволення, яке відчують користувачі.

В інтерфейсах соціальних мереж 26% загальної естетики забезпечується забезпеченням змістовного контрасту між елементами екрана, 24% – правильним групуванням елементів, 24% – правильним вирівнюванням елементів та груп екрана, 3% – тривимірним представленням та 22%



Рисунок 3.3 - Естетично привабливий

ефективне та просте використання кольорів і графіки. Властивість «естетична привабливість» становила 75,2% для всіх досліджених веб-інтерфейсів соціальних мереж. В *інтерфейсах веб-сайтів онлайн-магазинів* 26% загальної естетики забезпечується забезпеченням змістовного контрасту між елементами екрана, 25% – правильним групуванням елементів, 22% – правильним вирівнюванням елементів та груп екрана, 7% – тривимірним представленням та 19% – ефективним та простим використанням кольорів і графіки. Властивість «естетична привабливість» становила 76,2% для всіх досліджених інтерфейсів веб-сайтів магазинів.

Що стосується *інтерфейсів сайтів роботи*, то 27% загальної естетики забезпечується забезпеченням змістовного контрасту між елементами екрана, 28% – правильним групуванням елементів, 24% – правильним вирівнюванням елементів та груп екрана, 1% – тривимірним представленням та 20% – ефективним та простим використанням кольорів та графіки. Властивість «естетична приємність» становила 71,8% для всіх досліджених інтерфейсів сайтів роботи. В *інтерфейсах веб-сайтів біржової торгівлі* 26% загальної естетики забезпечується забезпеченням змістовного контрасту між елементами екрана, 23% – правильним групуванням елементів, 21% – правильним вирівнюванням елементів та груп екрана, 9% – тривимірним представленням та 21% – ефективним та простим використанням кольорів та графіки. Властивість «естетична приємність» становила 71,2% для всіх досліджених інтерфейсів веб-сайтів біржової торгівлі.

Щодо *інтерфейсів електронної пошти*, 26% загальної естетики забезпечується забезпеченням змістовного контрасту між елементами екрана, 24% – правильним групуванням елементів, 24% – правильним вирівнюванням елементів та груп екрана, 2% – тривимірним представленням та 24% – ефективним та простим використанням кольорів та графіки. Естетично привабливий вигляд становив 71,8% для всіх досліджених інтерфейсів веб-сайтів для торгівлі акціями.

Чіткість

Інтерфейс має бути чітким за візуальним виглядом, концепцією та формулюванням. Візуальні елементи повинні бути зрозумілими, стосуватися реальних концепцій та функцій користувача. Метафори, або аналогії, повинні бути реалістичними та простими. Слова та текст інтерфейсу повинні бути простими, однозначними та вільними від комп'ютерного жаргону [14]. Графік на рисунку 3.3 показує відсоток ясності, який відчуває користувач у різних типах веб-інтерфейсів.

Щодо інтерфейсів *сайтів соціальних мереж*, 26% загальної ясності забезпечується візуальним виглядом, концепціями та чіткими формулюваннями, 22% – візуальними елементами, зрозумілими стосовно реальних концепцій та функцій користувача, 20% – простими словами та текстом інтерфейсу, 15% – однозначним текстом та формулюваннями, 17% – словами, вільними від комп'ютерного жаргону. Загальна якість для цієї категорії веб-інтерфейсів становила 74,4%.

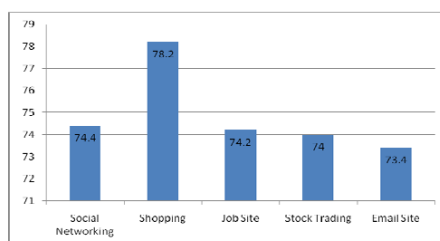


Рисунок 3.4 - Чіткість

Для інтерфейсів веб- *сайтів інтернет-магазинів* 26% загальної ясності забезпечується візуальним виглядом, концепціями та чіткими формулюваннями, 24% – візуальними елементами, зрозумілими стосовно реальних концепцій та функцій користувача, 24% – простими словами та текстом інтерфейсу, 9% – однозначним текстом та формулюваннями, 17% – словами, вільними від комп'ютерного жаргону. Загальна якість для цієї категорії веб-інтерфейсів становила 78,2%.

Для інтерфейсів *сайтів з працевлаштування* 26% загальної ясності забезпечується візуальним виглядом, концепціями та чіткими формулюваннями,

26% – візуальними елементами, зрозумілими стосовно реальних концепцій та функцій користувача, 27% – простими словами та текстом інтерфейсу, 9% – однозначним текстом та формулюваннями, 12% – словами, вільними від комп'ютерного жаргону. Загальна якість для цієї категорії веб-інтерфейсів становила **74,2%**. Для інтерфейсів веб-сайтів біржової торгівлі 25% загальної якості забезпечується візуальним виглядом, концепціями та чіткими формулюваннями, 22% – візуальними елементами, зрозумілими стосовно реальних концепцій та функцій користувача, 22% – простими словами та текстом інтерфейсу, 16% – однозначним текстом та формулюваннями, 14% – словами, вільними від комп'ютерного жаргону. Загальна якість для цієї категорії веб-інтерфейсів становила **74%**.



Рисунок 3.5 - Сумісність

Що стосується веб-сайтів електронної пошти, то 26% загальної якості забезпечується візуальним виглядом, концепціями та чіткими формулюваннями, 21% – візуальними елементами, зрозумілими стосовно реальних концепцій та функцій користувача, 25% – простими словами та текстом інтерфейсу, 13% – однозначним текстом та формулюваннями, 15% – словами, вільними від комп'ютерного жаргону. Загальна якість для цієї категорії веб-інтерфейсів становила **73,4%**.

Сумісність

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Аспекти сумісності, що були досліджені [14]:

- а) Сумісність користувачів,
- б) Сумісність завдань та роботи,
- с) Сумісність продуктів.

Графік на рисунку 3.5 зображує розуміння користувачем сумісності між різними веб-інтерфейсами користувача.

Для користувацьких інтерфейсів *сайтів соціальних мереж* 64% загальної сумісності забезпечується функціями/завданнями, що надаються в інтерфейсі користувача, які пов'язані з роботою, яку користувач хоче виконати, 36% - сприйняттям користувача екрану та способом його структури. Студенти вважають, що сумісність становить лише до **66%** у всіх досліджених користувацьких інтерфейсах цієї категорії. Для користувацьких інтерфейсів *сайтів інтернет-магазинів* 72% загальної сумісності забезпечується функціями/завданнями, що надаються в інтерфейсі користувача, які пов'язані з роботою, яку користувач хоче виконати, 28% - сприйняттям користувача екрану та способом його структури. Студенти вважають, що сумісність становить лише до **60%** у всіх досліджених користувацьких інтерфейсах цієї категорії.

Щодо користувацьких інтерфейсів *сайтів з працевлаштування*, 78% загальної сумісності забезпечується функціями/завданнями, що надаються в інтерфейсі користувача, які пов'язані з роботою, яку користувач хоче виконати, 22% - сприйняттям користувача екрану та способом його структури. Студенти вважають, що сумісність становить лише до **58,5%** у всіх досліджених користувацьких інтерфейсах цієї категорії. Для користувацьких інтерфейсів *сайтів біржової торгівлі* 65% загальної сумісності забезпечується функціями/завданнями, що надаються в інтерфейсі користувача, які пов'язані з роботою, яку користувач хоче виконати, 35% - сприйняттям користувача екрану та способом його структури. Студенти вважають, що сумісність становить лише до **67,5%** у всіх досліджених користувацьких інтерфейсах цієї категорії.

У користувацьких інтерфейсах *веб-сайтів електронної пошти* 71% загальної сумісності забезпечується функціями/завданнями, що надаються в

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

користувацькому інтерфейсі, які пов'язані з роботою, яку користувач хоче виконати, 29% - ставленням користувача до екрана та способом його структури. Студенти вважають, що сумісність становить лише до **60%** у всіх досліджених користувацьких інтерфейсах цієї категорії.

Зрозумілість

Система повинна бути зрозумілою, працювати у зрозумілому та осмисленому порядку. Повинні бути представлені чіткі підказки щодо роботи об'єктів. Кроки для виконання завдання повинні бути очевидними. Читання та засвоєння довгих пояснень ніколи не повинно бути необхідним [14]. Графік на рисунку 3.6 зображує різні рівні зрозумілості, які відчували користувачі у 5 різних досліджених веб-інтерфейсах користувача.



Рисунок 3.6 - Зрозумілість

На *сайтах соціальних мереж*, що стосується користувацьких інтерфейсів, 49% загальної зрозумілості забезпечується екраном та його зрозумілими та значущими елементами, 51% – наданням чітких підказок під час виконання операцій або завдань, 0% – довгими поясненнями. Загальна зрозумілість для веб-інтерфейсів цієї категорії становила 57,3%. На *веб-сайтах інтернет-магазинів*, що стосується користувацьких інтерфейсів, 46% загальної зрозумілості забезпечується екраном та його зрозумілими та значущими елементами, 40% – наданням чітких підказок під час виконання операцій або завдань, 14% – довгими поясненнями. Загальна зрозумілість для веб-інтерфейсів цієї категорії становила 70,3%.

На *сайтах роботи*, для користувацьких інтерфейсів, 48% загальної зрозумілості забезпечується екраном та його зрозумілими та значущими елементами, 36% – наданням чітких підказок під час виконання операцій або завдань, 16% – довгими поясненнями. Загальна зрозумілість для веб-інтерфейсів цієї категорії становила 62,3%. На *веб-сайтах біржової торгівлі*, для користувацьких інтерфейсів, 43% загальної зрозумілості забезпечується екраном та його зрозумілими та значущими елементами, 41% – наданням чітких підказок під час виконання операцій або завдань, 16% – довгими поясненнями. Загальна зрозумілість для веб-інтерфейсів цієї категорії становила 71%.

На *веб-сайтах електронної пошти*, що стосуються користувацьких інтерфейсів, 52% загальної зрозумілості забезпечується екраном та його елементами, які є зрозумілими та значущими, 35% – наданням чітких підказок під час виконання операцій або завдань, а 13% – розгорнутими поясненнями. Загальна зрозумілість для веб-інтерфейсів цієї категорії становила 59%.

Можливість налаштування

Легка персоналізація та налаштування шляхом конфігурації та реконфігурації системи посилює відчуття контролю, заохочує активну роль у розумінні та враховує особисті уподобання та відмінності в рівнях досвіду. Це також призводить до вищої задоволеності користувачів [14]. Деякі люди віддадуть перевагу персоналізації системи, щоб вона краще відповідала їхнім уподобанням. Інші люди цього не зроблять, приймаючи те, що дано. Ще інші експериментуватимуть з реконфігурацією, а потім здаватимуться, втративши терпіння або час. Для цих останніх груп користувачів має бути надана хороша конфігурація за замовчуванням [14]. Графік на рисунку 3.6 показує рівні конфігурації між різними веб-інтерфейсами.

Для *користувацьких інтерфейсів соціальних мереж* 58% загальної конфігурації забезпечується доступністю налаштування, 42% – доступністю персоналізації. Загальна конфігурація для веб-інтерфейсів цієї категорії становила 62%. Для *користувацьких інтерфейсів інтернет-магазинів* 63% загальної конфігурації забезпечується доступністю налаштування, 37% –

доступністю персоналізації. Загальна конфігурація для веб-інтерфейсів цієї категорії в дослідженні становила 35%.



Рисунок 3.7 - Можливість налаштування

Для користувацьких інтерфейсів «Робота» 52% загальної конфігурації забезпечується доступністю налаштування, 48% – доступністю персоналізації. Загальна конфігурація для веб-інтерфейсів цієї категорії становила 48%. Для користувацьких інтерфейсів «Біржова торгівля» 57% загальної конфігурації забезпечується доступністю налаштування, 43% – доступністю персоналізації. Загальна конфігурація для веб-інтерфейсів цієї категорії становила 50%. Для користувацьких інтерфейсів електронної пошти 49% загальної конфігурації забезпечується доступністю налаштування, 51% – доступністю персоналізації. Загальна конфігурованість для веб-інтерфейсів цієї категорії становила 73,5%.

Послідовність

Узгодженість може бути досягнута шляхом:

Система повинна виглядати, діяти та працювати однаково. Подібні компоненти повинні:

- Мати схожий вигляд.
- Мають схоже використання.
- діяти аналогічно.

- Одна й та сама дія завжди повинна давати однаковий результат.
- Функція елементів не повинна змінюватися.
- Положення стандартних елементів не повинно змінюватися.

Узгодженість дизайну – це спільна риса, що проходить через усі ці рекомендації. Це кардинальне правило всієї дизайнерської діяльності. Узгодженість важлива, оскільки вона може зменшити вимоги до навчання людини, дозволяючи переносити навички, отримані в одній ситуації, в іншу, подібну до неї. Графік на рисунку 3.7 зображує рівні узгодженості для різних типів досліджених веб-інтерфейсів користувача.



Рисунок 3.8 - Консистенція

Під час вивчення концепції узгодженості у веб-інтерфейсах соціальних мереж, 32% загальної узгодженості забезпечується подібними компонентами, що мають схожий вигляд, 26% - подібними компонентами, що мають схоже використання, 29% - однаковими діями, що призводять до однакового результату в різний час/на різних сторінках/розділах, 13% - зміною позицій стандартних елементів. Загальна узгодженість для цієї категорії веб-інтерфейсів користувача становила **56,75%**. Під час вивчення концепції узгодженості у веб-інтерфейсах інтернет-магазинів, 29% загальної узгодженості забезпечується подібними компонентами, що мають схожий вигляд, 25% - подібними компонентами, що мають схоже використання, 33% - однаковими діями, що призводять до однакового результату в різний час/на різних сторінках/розділах, 13% - зміною

позицій стандартних елементів. Загальна узгодженість для цієї категорії веб-інтерфейсів користувача становила **58,25%** .

Під час вивчення концепції узгодженості у веб-інтерфейсах *Job*, 35% загальної узгодженості забезпечується подібними компонентами, що мають схожий вигляд, 19% - подібними компонентами, що мають схоже використання, 24% - однаковими діями, що призводять до однакового результату в різний час/на різних сторінках/розділах, 22% - зміною позицій стандартних елементів. Загальна узгодженість для цієї категорії веб-інтерфейсів користувача становила **48,25%** . Під час вивчення концепції узгодженості у веб-інтерфейсах *біржової торгівлі*, 30% загальної узгодженості забезпечується подібними компонентами, що мають схожий вигляд, 24% - подібними компонентами, що мають схоже використання, 34% - однаковими діями, що призводять до однакового результату в різний час/на різних сторінках/розділах, 12% - зміною позицій стандартних елементів. Загальна узгодженість для цієї категорії веб-інтерфейсів користувача становила **54,25%** .

У веб-інтерфейсах *електронної пошти* 27% загальної узгодженості забезпечується схожими компонентами зі схожим зовнішнім виглядом, 24% – схожими компонентами зі схожим використанням, 37% – однаковими діями, що призводять до однакового результату в різний час/на різних сторінках/розділах, та 12% – зміною позицій стандартних елементів. Загальна узгодженість для цієї категорії веб-інтерфейсів користувача становила **59,5%** .

Контроль

Контроль – це відчуття влади, відчуття, що система реагує на ваші дії. Відчуття, що вами керує машина, деморалізує та розчаровує. Інтерфейс повинен мати вигляд інструменту. Контроль досягається, коли людина, працюючи у своєму власному темпі, здатна визначити, що робити, як це робити, а потім легко це зробити [14]. Графік на рисунку 3.9 показує рівні контролю, які відчувають користувачі серед різних веб-інтерфейсів користувача.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.9 - Управління

Щодо користувацького інтерфейсу *веб-сайтів соціальних мереж*, 26% загального контролю забезпечується властивістю виконання дій лише на запити користувачів, 26% – діями, що виконуються швидко, 14% – перервами доступності або припиненням виконуваних дій, 14% – перервами через помилки, 20% – наявністю різних режимів взаємодії. Загальний контроль для цієї категорії веб-інтерфейсів користувача становив **61,8%**. Щодо користувацького інтерфейсу *веб-сайтів інтернет-магазинів*, 28% загального контролю забезпечується властивістю виконання дій лише на запити користувачів, 26% – діями, що виконуються швидко, 22% – перервами доступності або припиненням виконуваних дій, 9% – перервами через помилки, 15% – наявністю різних режимів взаємодії. Загальний контроль для цієї категорії веб-інтерфейсів користувача становив **60,8%**.

Для *веб-сайтів з працевлаштування* 29% загального контролю забезпечується властивістю виконання дій лише за запитами користувачів, 25% – діями, що виконуються швидко, 17% – перервами доступності або припиненням виконуваних дій, 16% – перервами через помилки, 13% – наявністю різних режимів взаємодії. Загальний контроль для веб-інтерфейсів користувача цієї категорії становив 66,8%. Для інтерфейсу користувача *веб-сайтів біржової торгівлі* 27% загального контролю забезпечується властивістю виконання дій лише за запитами користувачів, 27% – діями, що виконуються швидко, 17% – перервами доступності або припиненням виконуваних дій, 14% – перервами через помилки, 15% – наявністю різних режимів взаємодії. Загальний контроль для веб-інтерфейсів користувача цієї категорії становив 67,4%.

Що стосується *веб-сайтів електронної пошти*, 30% загального контролю забезпечується властивістю виконання дій лише на запити користувачів, 27% – діями, які виконуються швидко, 20% – перервами доступності або завершеннями виконуваних дій, 13% – перервами через помилки, 10% – наявністю різних режимів взаємодії. Загальний контроль для веб-інтерфейсів користувача цієї категорії становив 63,4%.

Прямота

Завдання слід виконувати безпосередньо. Доступні альтернативи повинні бути видимими, що зменшує розумове навантаження користувача. Прямоту також найкраще забезпечує послідовність об'єкт-дія систем прямої маніпуляції. Завдання виконуються шляхом безпосереднього вибору об'єкта, потім вибору дії, яку потрібно виконати, а потім спостереження за виконанням дії [14]. Графік на рисунку 3.9 показує рівні прямоті, які відчують користувачі для різних веб-інтерфейсів користувача.

Для користувацьких інтерфейсів *сайтів соціальних мереж* 29% загальної прямої спрямованості досягається шляхом видимості альтернативних дій, 34% – шляхом забезпечення негайного ефекту/результату дій, що виконуються над об'єктами, та 37% – шляхом надання завдань, які можна виконати безпосередньо. Загальна прямота для цієї категорії сайтів становила 82,6%. Для користувацьких інтерфейсів *веб-сайтів інтернет-магазинів* 27% загальної прямої спрямованості досягається шляхом видимості альтернативних дій, 37% – шляхом забезпечення негайного ефекту/результату дій, що виконуються над об'єктами, та 36% – шляхом надання завдань, які можна виконати безпосередньо. Загальна прямота для цієї категорії сайтів становила 86,3%.



Рисунок 3.10 - Прямолінійність

Щодо користувацьких інтерфейсів *сайтів з працевлаштування*, 28% загальної прямої спрямованості досягається шляхом видимості альтернативних дій, 37% – шляхом забезпечення негайного ефекту/результату дій, що виконуються над об'єктами, та 35% – шляхом надання завдань, які можна виконати безпосередньо. Загальна прямота для цієї категорії сайтів становила 73,6%. Щодо користувацьких інтерфейсів *сайтів біржової торгівлі*, 27% загальної прямої досягається шляхом видимості альтернативних дій, 37% – шляхом забезпечення негайного ефекту/результату дій, що виконуються над об'єктами, та 36% – шляхом надання завдань, які можна виконати безпосередньо. Загальна прямота для цієї категорії сайтів становила 81%.

Для *сайтів електронної пошти* 21% загальної прямої досягається шляхом видимості альтернативних дій, 40% – шляхом забезпечення негайного ефекту/результату дій, що виконуються над об'єктами, 39% – шляхом надання завдань, які можна виконати безпосередньо. Загальна прямота для цієї категорії сайтів становила 73%.

Ефективність

Руки очей і рук не повинні витрачатися даремно. Увага користувача має бути прикута до відповідних елементів екрана, коли це необхідно. Послідовні рухи очей між елементами екрана повинні бути передбачуваними, очевидними та короткими. Веб-сторінки повинні бути легкодоступними для сканування. Усі шляхи навігації повинні бути якомога коротшими [14]. Уникайте частих

переходів між пристроями введення, такими як клавіатура та миша [15]. Графік на рисунку 3.11 показує сприйняття користувачами ефективності різних веб-інтерфейсів користувача.

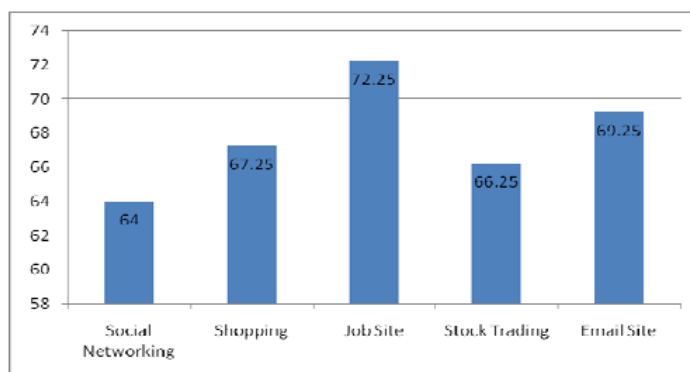


Рисунок 3.11 - Ефективність

В інтерфейсах користувачів *соціальних мереж* 31% загальної ефективності забезпечується спрощенням рухів очей та рук між елементами керування/компонентами, 17% – короткими шляхами навігації, 23% – послідовним рухом очей по екрану та 28% – частим переходом між пристроями введення. Загальна ефективність для цього типу інтерфейсів користувача становила **64%**. В інтерфейсах користувачів *сайтів інтернет-магазинів* 30% загальної ефективності забезпечується спрощенням рухів очей та рук між елементами керування/компонентами, 17% – короткими шляхами навігації, 26% – послідовним рухом очей по екрану та 27% – частим переходом між пристроями введення. Загальна ефективність для цього типу інтерфейсів користувача становила **67,25%**.

Для користувацьких інтерфейсів *сайтів робіт* 32% загальної ефективності забезпечується спрощенням рухів очей та рук між елементами керування/компонентами, 14% – короткими шляхами навігації, 24% – послідовним рухом очей по екрану та 30% – частим переходом між пристроями введення. Загальна ефективність для цього типу користувацьких інтерфейсів становила **72,25%**. В користувацьких інтерфейсах *сайтів біржової торгівлі* 31% загальної ефективності забезпечується спрощенням рухів очей та рук між

елементами керування/компонентами, 18% – короткими шляхами навігації, 23% – послідовним рухом очей по екрану та 28% – частим переходом між пристроями введення. Загальна ефективність для цього типу користувацьких інтерфейсів становила **66,25%** .

Щодо *сайтів електронної пошти*, 31% загальної ефективності забезпечується спрощенням рухів очей і рук між елементами керування/компонентами, 11% – короткими шляхами навігації, 29% – послідовним рухом очей по екрану та 29% – частим переходом між пристроями введення. Загальна ефективність для цього типу користувацьких інтерфейсів склала **69,25%** .

Знайомство

Знайомство може бути досягнуто шляхом:

- Використання знайомих понять та мови, зрозумілої користувачеві.
- Збереження природності інтерфейсу, імітація моделей поведінки користувача.
- Використання метафор реального світу.

Графік на рисунку 3.12 показує рівні знайомства користувачів з різними веб-інтерфейсами.

Для користувацьких інтерфейсів *сайтів соціальних мереж* 55% загальної знайомості забезпечується мовою, відомою користувачеві, 45% – створенням інтерфейсу, що відповідає реальній ситуації. Загальна знайомість для цього типу користувацьких інтерфейсів становила 84%. Для користувацьких інтерфейсів *сайтів інтернет-магазинів* 51% загальної знайомості забезпечується мовою, відомою користувачеві, 49% – створенням інтерфейсу, що відповідає реальній ситуації. Загальна знайомість для цього типу користувацьких інтерфейсів становила 79,5%.



Рисунок 3.12 - Знайомість

Що стосується *сайтів з працевлаштування*, то 56% загальної знайомості забезпечується мовою, відомою користувачеві, 44% – інтерфейсом, що виглядає як реальна ситуація. Загальна знайомість становила 85,5%. Для користувацьких інтерфейсів *сайтів біржової торгівлі* 44% загальної знайомості забезпечується мовою, відомою користувачеві, 56% – інтерфейсом, що виглядає як реальна ситуація. Загальна знайомість для цього типу користувацьких інтерфейсів становила 69,5%.

На *сайтах електронної пошти* 56% загальної знайомості забезпечується мовою, відомою користувачеві, 44% – тим, що інтерфейс виглядає як реальна ситуація. Загальна знайомість для цього типу користувацьких інтерфейсів склала 72,5%.

Гнучкість

Гнучкість – це здатність системи реагувати на індивідуальні відмінності людей. Дозволяє людям вибрати метод взаємодії, який найбільше відповідає їхній ситуації. Це також досягається шляхом налаштування системи [14]. Графік на рисунку 3.13 показує рівні гнучкості, що сприймаються користувачами в різних веб-інтерфейсах.

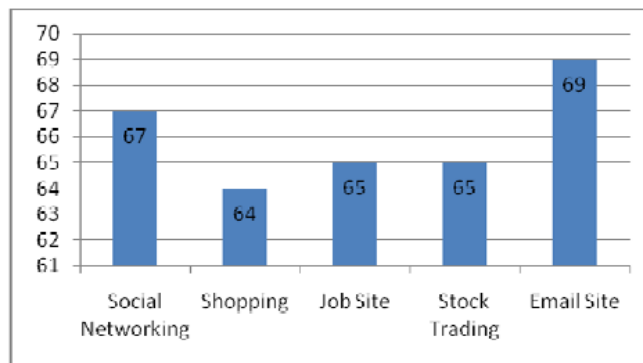


Рисунок 3.13 - Гнучкість

Щодо інтерфейсів користувачів сайтів соціальних мереж, 67% вважали, що система достатньо гнучка для персоналізації відповідно до їхніх уподобань. На сайтах торгівлі акціями гнучкість становила 65%. Щодо інтерфейсів користувачів сайтів онлайн-магазинів, гнучкість оцінювалася приблизно на рівні 64%. Щодо інтерфейсів користувачів сайтів вакансій, гнучкість становила 65%, а для сайту електронної пошти – близько 69%.

Прощення

Люди будуть робити помилки; система повинна толерувати ті, які є поширеними та неминучими. Система, що прощає, оберігає людей від неприємностей [14]. Люди люблять досліджувати та навчатися методом спроб і помилок. Система, надмірно чутлива до помилкових даних, відлякуватиме користувачів від дослідження та спроб нового [14].

Запобігайте виникненню помилок, передбачаючи, де можуть виникнути помилки, та розробляючи заходи для їх запобігання. Дозвольте людям переглядати, змінювати та скасовувати дії за потреби. Зробіть виконання дій, які можуть мати трагічні наслідки, дуже складним. Коли помилки трапляються, надайте чіткі інструкції щодо їх виправлення [17]. Графік на рисунку 3.14 показує різні рівні прощення, які відчувають користувачі.



Рисунок 3.14 - Прощення

В інтерфейсах користувачів *соціальних мереж* 38% загального рівня прощення забезпечується системою, яка допомагає користувачеві виправляти помилки, 22% – запобігає помилкам користувача, 40% – надає конструктивні повідомлення у разі виникнення помилки. Загальний рівень прощення для цього типу інтерфейсів користувача становить 73,3%. В інтерфейсах користувачів *інтернет-магазинів* 37% загального рівня прощення забезпечується системою, яка допомагає користувачеві виправляти помилки, 25% – запобігає помилкам користувача, 38% – надає конструктивні повідомлення у разі виникнення помилки. Загальний рівень прощення для цього типу інтерфейсів користувача становить 64%.

Що стосується *сайтів з працевлаштування*, 38% загального прощення забезпечується системою, яка допомагає користувачеві виправляти помилки, 22% – запобіганням помилкам користувача, 40% – наданням конструктивних повідомлень у разі виникнення помилки. Загальний рівень прощення для цього типу користувацьких інтерфейсів становив 62,66%. В користувацьких інтерфейсах *сайтів біржової торгівлі* 38% загального прощення забезпечується системою, яка допомагає користувачеві виправляти помилки, 22% – запобіганням помилкам користувача, 40% – наданням конструктивних повідомлень у разі виникнення помилки. Загальний рівень прощення для цього типу користувацьких інтерфейсів становив 61%.

У користувацьких інтерфейсах *сайтів електронної пошти* 36%

загального прощення забезпечується системою, яка допомагає користувачеві виправляти помилки, 25% – запобігає помилкам користувача, а 39% – надає конструктивні повідомлення у разі виникнення помилки. Загальний рівень прощення для цього типу користувацьких інтерфейсів становить 75%.

Передбачуваність

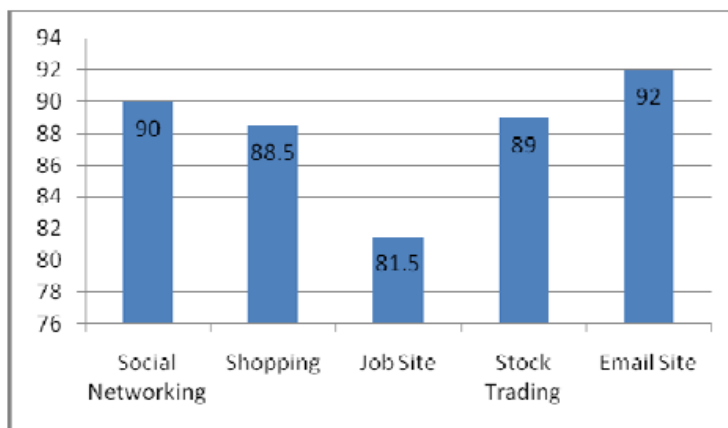


Рисунок 3.15 - Передбачуваність

Передбачуваність, або передбачуваність, зменшує кількість помилок і дозволяє швидше виконувати завдання. Усі очікування користувача повинні виконуватися однаково та повністю [14]. Графік на рисунку 3.15 показує різні рівні передбачуваності, які відчувають користувачі.

Для користувацьких інтерфейсів *веб-сайтів соціальних мереж* 51% загальної передбачуваності забезпечується властивістю, що очікуваний користувачем результат відповідає фактичному результату дій, 49% – шляхом рівномірного та повного надання результатів дій. Загальна передбачуваність для інтерфейсів веб-сайтів цієї категорії становила 90%. Для користувацьких інтерфейсів *веб-сайтів інтернет-магазинів* 46% загальної передбачуваності забезпечується властивістю, що очікуваний користувачем результат відповідає фактичному результату дій, 54% – шляхом рівномірного та повного надання результатів дій. Загальна передбачуваність для інтерфейсів веб-сайтів цієї категорії становила 88,5%.

Що стосується *веб-сайтів з працевлаштування*, 46% загальної

передбачуваності забезпечується властивістю, що очікуваний користувачем результат відповідає фактичному результату дій, 54% – рівномірним та повним наданням результатів дій. Загальна передбачуваність для інтерфейсів веб-сайтів цієї категорії становила 81,5%. Для інтерфейсів користувачів веб-сайтів біржової торгівлі 49% загальної передбачуваності забезпечується властивістю, що очікуваний користувачем результат відповідає фактичному результату дій, 51% – рівномірним та повним наданням результатів дій. Загальна передбачуваність для інтерфейсів веб-сайтів цієї категорії становила 89%.

У користувацьких інтерфейсах веб-сайтів електронної пошти 50% загальної передбачуваності забезпечується властивістю відповідності очікуваного користувачем результату фактичному результату дій, а 50% – рівномірним та повним наданням результатів дій. Загальна передбачуваність для інтерфейсів веб-сайтів цієї категорії становила 92%.

Відновлення

Відновлення має бути очевидним, автоматичним, легким та природним для виконання. Легке відновлення після дії значно полегшує навчання методом спроб і помилок, а також дослідження [14]. Графік на рисунку 3.16 показує різні рівні відновлення, які відчують користувачі.

В інтерфейсах користувачів веб-сайтів соціальних мереж 46% загального відновлення забезпечується «повторенням» щойно виконаних дій, 54% – наданням можливості повернутися до попередньої дії, коли користувач застряг на просуванні вперед або виконанні наступної дії. Загальний показник відновлення для цього типу веб-інтерфейсів користувача становив 65,5%. В інтерфейсах користувачів веб-сайтів покупок 43% загального відновлення забезпечується «повторенням» щойно виконаних дій, 57% – наданням можливості повернутися до попередньої дії, коли користувач застряг на просуванні вперед або виконанні наступної дії. Загальний показник відновлення для цього типу веб-інтерфейсів користувача становив 74%.



Рисунок 3.16 - Відновлення

Що стосується інтерфейсів користувачів *веб-сайтів з пропозиціями роботи*, 46% загального відновлення забезпечується «повторенням» дій, які щойно виконали користувачі, 54% – наданням можливості повернутися до попередньої дії, коли користувач застряг у русі вперед або виконанні наступної дії. Загальний показник відновлення для цього типу веб-інтерфейсів користувача становив 54%. В інтерфейсах користувачів веб- *сайтів біржової торгівлі* 35% загального відновлення забезпечується «повторенням» дій, які щойно виконали користувачі, 65% – наданням можливості повернутися до попередньої дії, коли користувач застряг у русі вперед або виконанні наступної дії. Загальний показник відновлення для цього типу веб-інтерфейсів користувача становив 43,5%.

Для *веб-сайтів електронної пошти* 36% загального відновлення забезпечується «повторенням» дій, які користувачі щойно виконали, а 64% — наданням можливості повернутися до попередньої дії, коли користувач застряг на переміщенні вперед або виконанні наступної дії. Загальний показник відновлення для цього типу веб-інтерфейсів користувача становив **69%** .

Швидкість реагування

На запит користувача необхідно швидко реагувати. Зворотній зв'язок може бути візуальним, зміною форми вказівника миші, або текстовим, у формі повідомлення. Він також може бути слуховим, що складається з унікального звуку чи тону [14].

Для користувацьких інтерфейсів *веб-сайтів соціальних мереж* 35% загальної швидкості реагування забезпечується швидкою реакцією системи на запити/дії користувача, 24% – наданням зворотного зв'язку/підтвердження дій користувача за допомогою **ВІЗУАЛЬНОЇ** підказки, 39% – наданням зворотного зв'язку за допомогою **ТЕКСТОВИХ** підказок і лише 2% – наданням зворотного зв'язку за допомогою **АУДІО** підказки. Загальна швидкість реагування для цієї категорії веб-інтерфейсів користувача становила **54%** . Для користувацьких інтерфейсів *веб -сайтів покупок* 34% загальної швидкості реагування забезпечується швидкою реакцією системи на запити/дії користувача, 20% – наданням зворотного зв'язку/підтвердження дій користувача за допомогою **ВІЗУАЛЬНОЇ** підказки, 37% – наданням зворотного зв'язку за допомогою **ТЕКСТОВИХ** підказок і лише 9% – наданням зворотного зв'язку за допомогою **АУДІО** підказки. Загальна швидкість реагування для цієї категорії веб-інтерфейсів користувача становила **56,75%** .

На *веб-сайтах з пошуку роботи* 38% загальної швидкості реагування забезпечується швидкою реакцією системи на запити/дії користувача, 15% – наданням зворотного зв'язку/підтвердження дій користувача за допомогою **ВІЗУАЛЬНОЇ** підказки, 43% – наданням зворотного зв'язку за допомогою **ТЕКСТОВИХ** підказок і лише 4% – наданням зворотного зв'язку за допомогою **АУДІО** підказки. Загальна швидкість реагування для цієї категорії веб-інтерфейсів користувача становила **48,75%** . Для користувацьких інтерфейсів *веб-сайтів біржової торгівлі* 30% загальної швидкості реагування забезпечується швидкою реакцією системи на запити/дії користувача, 25% – наданням зворотного зв'язку/підтвердження дій користувача за допомогою **ВІЗУАЛЬНОЇ** підказки, 35% – наданням зворотного зв'язку за допомогою **ТЕКСТОВИХ** підказок і лише 10% – наданням зворотного зв'язку за допомогою **АУДІО** підказки. Загальна швидкість реагування для цієї категорії веб-інтерфейсів користувача становила **61,25%** .

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		86



Рисунок 3.17 - Швидкість реагування

На веб-сайтах з електронною поштою 32% загальної швидкості реагування забезпечується швидкою реакцією системи на запити/дії користувача, 24% – наданням зворотного зв'язку/підтвердження дій користувача за допомогою ВІЗУАЛЬНИХ підказок, 38% – наданням зворотного зв'язку за допомогою ТЕКСТОВИХ підказок і лише 6% – наданням зворотного зв'язку за допомогою АУДІО-підказок. Загальний рівень швидкості реагування для цієї категорії веб-інтерфейсів користувача становив **60%**.

Простота

Забезпечте максимально простий інтерфейс. П'ять способів забезпечення простоти [14]:

- Використовуйте поступове розкриття інформації, приховуючи речі, доки вони не знадобляться.
- Спочатку представити загальні та необхідні функції.
- Чітко відображають важливі функції.
- Приховати складніші та менш часто використовувані функції.
- Надайте значення за замовчуванням.
- Мінімізуйте точки вирівнювання екрана.
- Зробіть звичайні дії простими, але ускладніть виконання незвичайних дій.
- Забезпечити однорідність та послідовність.

У досліджених користувацьких інтерфейсах веб-сайтів соціальних мереж 23% загальної простоти було забезпечено наданням загальних та

помітних функцій на перший погляд, 21% – приховуванням складних та менш часто використовуваних функцій або їх доступом шляхом взаємодії, 16% – наданням налаштувань за замовчуванням, 23% – спрощенням поширених дій та 17% – однорідністю та узгодженістю елементів на екрані та взаємодій. Загальна простота для цього типу веб-інтерфейсів користувача становила **70%**.

Для досліджених *веб-сайтів інтернет-магазинів* 23% загальної простоти було забезпечено наданням загальних та помітних функцій/функцій на перший погляд, 19% – приховуванням складних та менш часто використовуваних функцій/функцій або їх отриманням шляхом взаємодії, 16% – наданням налаштувань за замовчуванням, 23% – спрощенням загальних дій та 19% – однорідністю та узгодженістю елементів на екрані та взаємодій. Загальна простота для цього типу веб-інтерфейсів користувача становила **81,2%**.



Рисунок 3.18 - Простота

Що стосується досліджених *веб-сайтів з пропозицій роботи*, то 20% загальної простоти було забезпечено наданням загальних та помітних функцій, помітних на перший погляд, 18% – приховуванням складних та менш часто використовуваних функцій або їх доступом шляхом взаємодії, 20% – наданням налаштувань за замовчуванням, 24% – спрощенням поширених дій, а 18% – однорідністю та узгодженістю елементів на екрані та взаємодій. Загальна простота для цього типу веб-інтерфейсів користувача становила **76,4%**.

Згідно з дослідженням *веб-сайтів для торгівлі акціями*, 20% загальної простоти було забезпечено завдяки наданню загальних та помітних функцій,

помітних на перший погляд, 22% – завдяки приховуванню складних та менш часто використовуваних функцій або їх можливості отримати до них шляхом взаємодії, 14% – завдяки наданню налаштувань за замовчуванням, 21% – спрощенню поширених дій та 23% – однорідності та узгодженості елементів на екрані та взаємодій. Загальна простота для цього типу веб-інтерфейсів користувача становила **78,4%** .

користувацьких інтерфейсах веб-сайтів електронної пошти 24% загальної простоти було забезпечено наданням загальних та помітних функцій, помітних на перший погляд, 17% – приховуванням складних та менш часто використовуваних функцій або їх доступом шляхом взаємодії, 17% – наданням налаштувань за замовчуванням, 23% – спрощенням поширених дій та 19% – однорідністю та узгодженістю елементів на екрані та взаємодій. Загальна простота для цього типу веб-інтерфейсів користувача становила **73,6%** .

Групування

Групування елементів екрана допомагає у встановленні структури, значущих зв'язків та змістовної форми. Окрім забезпечення естетичної привабливості, попередні дослідження показали, що групування сприяє запам'ятовуванню інформації та призводить до швидшого пошуку на екрані. Дослідження Гроуза, Паруша, Надіра та

Штуб виявив, що надання груп елементів екрана, що містять змістовні назви груп, також пов'язане зі скороченням часу пошуку на екрані [14].

Принцип групування становив 74,6% для веб-інтерфейсів соціальних мереж, 77% для веб-інтерфейсів торгівлі акціями, 74% для інтерфейсів електронної пошти, 76% для веб-інтерфейсів онлайн-магазинів та 77% для інтерфейсів сайтів з пошуку роботи.

Групування за допомогою пробілу

Галіц [14] пропонує забезпечити належне розділення між групами шляхом вільного використання білого простору, а також просить ретельно розглянути компроміс між білим простором на екрані та вимогою прокручування сторінки.

Групування з білими пробілами становило 17% в інтерфейсах веб-сайтів соціальних мереж, 16% в інтерфейсах сайтів торгівлі акціями, 16% в інтерфейсах користувачів електронної пошти, 16% в інтерфейсах користувачів онлайн-магазинів та 21% в інтерфейсах сайтів будівництва.

Групування за допомогою меж

Під час групування з рамками, треба

(a) включати лінійні рамки для: (i) Зосередження уваги на групах або пов'язаній інформації. (ii) Спрямовування ока на екран.

(b) Не перевищуйте три товщини ліній або два стилі ліній на екрані.

(c) Створюйте лінії однакової висоти та довжини.

(d) Залишайте достатній відступ між інформацією та навколишніми рамками.

(e) Для суміжних груп з рамками, коли це можливо, вирівнюйте межі ліворуч, праворуч, зверху та знизу.

(f) Використовуйте лінійки та рамки економно.

У дизайні веб-сторінок:

(i) Будьте обережні, використовуючи горизонтальні лінії як роздільники між розділами сторінки.

(ii) Залиште горизонтальні лінії для ситуацій, коли необхідно підкреслити різницю між суміжними областями.

Таблиця 3.1

Різні методи виділення та їхні приклади у різних досліджуваних веб-інтерфейсах.

Техніки фокусування/акцентування	Соціальні мережі	Сайти торгівлі акціями	Електронна пошта	Сайти онлайн-покупок	Сайти з пошуку роботи
Вища яскравість	79	84	80	80	80
Зворотна полярність (інверсія)	42	74	35	53	35
Більший та особливий шрифт	84	89	70	100	100
Підкреслення	32	47	60	50	40
Мерехтіння (блимання)	32	37	15	45	20

Контрастний колір	89	89	80	84	75
-------------------	----	----	----	----	----

Групування з рамками становило 24% у веб-інтерфейсах соціальних мереж, 19% у веб-інтерфейсах торгівлі акціями, 24% у веб-інтерфейсах електронної пошти, 24% в інтерфейсах користувачів онлайн-магазинів та 16% в інтерфейсах сайтів з працевлаштування.

Фокус та акцент

Рекомендується застосовувати техніку візуального акцентування, щоб виділити найважливіші або найвидатніші частини екрана. Виділений елемент повинен контрастувати з рештою екрана, привертаючи до себе увагу користувача.

Для акцентування уваги використовуються різні методи, такі як:

- (a) Вища яскравість
- (b) Зворотна полярність або інверсне відео
- (c) Більший та чіткіший шрифт
- (d) Підкреслення
- (e) Миготіння
- (f) Лінійки та рамки навколо них
- (g) Контрастний колір
- (h) Більший розмір
- (i) Позиціонування,
- (j) Ізоляція
- (k) Характерна або незвичайна форма
- (l) Білий простір можна використовувати систематично та приємно.

Принципи користувацького інтерфейсу залишаються актуальними навіть після багатьох років їх впровадження. Єдина відмінність полягає в тому, що вони мають різні форми, при цьому основні аспекти принципів залишаються сильними. Це зробило досліджувані веб-сайти популярними навіть за низьких значень деяких досліджуваних принципів. Дослідження виявило існування цих принципів у різних формах та важливість їх існування для того, щоб користувацький інтерфейс був привабливим та ефективним. Великий список

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		91

статистичних даних, наведених у цій роботі, допомагає зрозуміти індивідуальну присутність принципів у різних типах веб-інтерфейсів, а також популярність цих користувацьких інтерфейсів серед користувачів.

3.4 Висновок по розділу

У третьому розділі розглянуто практичні аспекти реалізації та оцінки якості інтерфейсів користувача. Проаналізовано стандарти якості програмного забезпечення, виконано проєктування та реалізацію графічного інтерфейсу, а також проведено оцінювання його відповідності принципам UX/UI. Результати дослідження підтвердили, що дотримання сучасних принципів проєктування інтерфейсів забезпечує підвищення зручності, продуктивності та задоволеності користувачів під час роботи з програмним забезпеченням.

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання дипломної роботи було досліджено принципи створення інтерфейсів користувача в програмному забезпеченні, що є важливою складовою розробки сучасних інформаційних систем. У роботі проаналізовано існуючі підходи до проєктування інтерфейсів, визначено ключові принципи UX/UI-дизайну, а також розглянуто методи та інструменти, що забезпечують створення зручних, ефективних і інтуїтивно зрозумілих інтерфейсів. Отримані результати підтверджують, що якісно спроектований інтерфейс суттєво впливає на ефективність взаємодії користувача із системою та загальну якість програмного продукту.

На початковому етапі було проведено аналіз теоретичних основ створення інтерфейсів користувача, визначено основні поняття, класифікацію інтерфейсів та критерії їх оцінювання. Особливу увагу приділено таким характеристикам, як юзабіліті, доступність, ефективність і задоволеність користувачів. Це дозволило сформулювати базу для подальшого дослідження та практичної реалізації.

У процесі роботи було досліджено методи та інструменти проєктування інтерфейсів користувача, включаючи принципи організації взаємодії, забезпечення зворотного зв'язку та аналіз сценаріїв використання. Застосування цих підходів дозволило структурувати процес проєктування та врахувати потреби кінцевих користувачів. Важливим результатом стало розуміння ролі адаптивності інтерфейсу та орієнтації на користувача при розробці програмного забезпечення.

Практична частина роботи охопила реалізацію графічного інтерфейсу користувача з урахуванням сучасних стандартів якості програмного забезпечення. У процесі розробки було застосовано принципи консистентності, простоти, логічної структури та ефективної навігації. Проведене оцінювання

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

інтерфейсу дозволило виявити його сильні сторони та визначити можливі напрями вдосконалення, зокрема підвищення доступності та оптимізацію взаємодії користувача із системою.

Було також проведено тестування інтерфейсу користувача, що включало перевірку зручності використання, швидкості виконання основних операцій і відповідності очікуванням користувачів. Результати тестування підтвердили, що застосовані принципи проєктування позитивно впливають на якість взаємодії та зменшують кількість помилок при роботі з системою.

Загалом, виконана робота досягла поставленої мети: досліджено основні принципи створення інтерфейсів користувача, проаналізовано методи їх проєктування та реалізовано приклад інтерфейсу з подальшою оцінкою його якості. Перевагами запропонованого підходу є орієнтація на користувача, гнучкість у проєктуванні та можливість адаптації до різних типів програмного забезпечення.

Разом з тим, існують певні обмеження, пов'язані з необхідністю більш глибокого врахування індивідуальних особливостей користувачів, а також оптимізації інтерфейсів для різних пристроїв і платформ. У подальшому дослідження можуть бути спрямовані на використання адаптивних інтерфейсів, інтеграцію технологій штучного інтелекту для персоналізації взаємодії, а також розширення можливостей тестування користувацького досвіду.

Таким чином, застосування сучасних принципів створення інтерфейсів користувача є необхідною умовою підвищення якості програмного забезпечення, забезпечення ефективної взаємодії користувача із системою та досягнення високого рівня задоволеності кінцевих користувачів.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Nielsen, J. (1994). *Usability Engineering*. morgan kaufmann.
<https://www.nngroup.com/books/usability-engineering/>
2. Norman, D. (2013). *The Design of Everyday Things*. basic books.
<https://www.basicbooks.com/titles/don-norman/the-design-of-everyday-things/9780465050659/>
3. Krug, S. (2014). *Don't Make Me Think: A Common Sense Approach to Web Usability*. new riders. <https://sensible.com/dont-make-me-think/>
4. Garrett, J. J. (2010). *The Elements of User Experience*. peachpit press.
<https://www.uxdesign.cc/the-elements-of-user-experience-ux-summary-6f3c1c9c7f4b>
5. Tidwell, J. (2019). *Designing Interfaces*. oreilly.
<https://www.oreilly.com/library/view/designing-interfaces-3rd/9781492051954/>
6. Cooper, A., Reimann, R., & Cronin, D. (2014). *About Face: The Essentials of Interaction Design*. wiley. <https://www.wiley.com/en-us/About+Face%3A+The+Essentials+of+Interaction+Design-p-9781118766576>
7. Shneiderman, B., et al. (2016). *Designing the User Interface*. pearson.
<https://www.pearson.com/en-us/subject-catalog/p/designing-the-user-interface/P200000003464>
8. Preece, J., Rogers, Y., & Sharp, H. (2019). *Interaction Design: Beyond Human-Computer Interaction*. wiley. <https://www.wiley.com/en-us/Interaction+Design%3A+Beyond+Human+Computer+Interaction-p-9781119547259>
9. ISO 9241-210. (2019). *Human-centred design for interactive systems*. iso.org. <https://www.iso.org/standard/77520.html>
10. W3C. (2024). *Web Content Accessibility Guidelines (WCAG) 2.2*. w3.org.
<https://www.w3.org/TR/WCAG22/>
11. Microsoft. (2025). *Fluent Design System*. learn.microsoft.com.
<https://learn.microsoft.com/en-us/windows/apps/design/>

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		95

12. Apple. (2025). *Human Interface Guidelines*. developer.apple.com.
<https://developer.apple.com/design/human-interface-guidelines/>
13. Google. (2025). *Material Design*. material.io. <https://material.io/design>
14. Bootstrap. (2025). Documentation. <https://getbootstrap.com/docs/>
15. Figma. (2025). Design System Guide. <https://help.figma.com>
16. Adobe XD. (2024). UI/UX Design Guide.
<https://helpx.adobe.com/xd.html>
17. Nielsen Norman Group. (2025). UX Research Articles.
<https://www.nngroup.com/articles/>
18. UXPin. (2024). UI Design Patterns.
<https://www.uxpin.com/studio/blog/ui-design-patterns/>
19. Interaction Design Foundation. (2025). UX Courses and Articles.
<https://www.interaction-design.org>
20. Smashing Magazine. (2024). UX Design Guidelines.
<https://www.smashingmagazine.com/category/ux>
21. .NET MAUI Documentation. (2025). learn.microsoft.com.
<https://learn.microsoft.com/en-us/dotnet/maui/>
22. WPF Documentation. (2025). learn.microsoft.com.
<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>
23. Blazor Documentation. (2025). learn.microsoft.com.
<https://learn.microsoft.com/en-us/aspnet/core/blazor/>
24. Angular Documentation. (2025). angular.io. <https://angular.io/docs>
25. React Documentation. (2025). react.dev. <https://react.dev/learn>
26. Vue.js Documentation. (2025). vuejs.org.
<https://vuejs.org/guide/introduction.html>
27. Tailwind CSS. (2025). Documentation. <https://tailwindcss.com/docs>
28. SASS. (2024). Guide. <https://sass-lang.com/guide>
29. Material UI. (2025). mui.com. <https://mui.com/material-ui/getting-started/overview/>
30. Chakra UI. (2025). chakra-ui.com. <https://chakra-ui.com/docs>

31. Lazar, J., Feng, J., & Hochheiser, H. (2017). *Research Methods in Human-Computer Interaction*. morgan kaufmann. <https://www.elsevier.com/books/research-methods-in-human-computer-interaction/lazar/978-0-12-805390-4>
32. Rubin, J., & Chisnell, D. (2011). *Handbook of Usability Testing*. wiley. <https://www.wiley.com/en-us/Handbook+of+Usability+Testing-p-9780470185483>
33. Sauro, J., & Lewis, J. (2016). *Quantifying the User Experience*. morgan kaufmann. <https://www.elsevier.com/books/quantifying-the-user-experience/sauro/978-0-12-802308-2>
34. Garrett, J. (2022). UX Strategy Updates. <https://adaptivepath.org>
35. UX Collective. (2024). UX Best Practices. <https://uxdesign.cc>
36. OWASP. (2024). *Top Ten Security Risks*. <https://owasp.org/www-project-top-ten/>
37. Google. (2024). *Mobile UX Design Guidelines*. <https://developers.google.com/web/fundamentals/design-and-ux>
38. Microsoft. (2024). *Accessibility in Software Design*. <https://learn.microsoft.com/en-us/accessibility/>
39. IBM. (2024). *Design Thinking Framework*. <https://www.ibm.com/design/thinking/>
40. Gartner. (2024). *UX Trends in Software Development*. <https://www.gartner.com/en/information-technology/insights>

					БР.ІІІ – 46.00.00.000 ПЗ	Арк.
						97
ЗМН.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: "Принципи створення інтерфейсів користувача в програмному забезпеченні "

Обсяг пояснювальної записки: 97 аркушів

Дата закінчення бакалаврської роботи 10 червня 2026р.

Підпис студента _____