

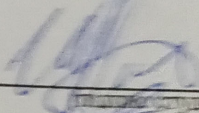
## Пояснювальна записка до кваліфікаційної роботи

магістр  
(ступінь вищої освіти)

на тему: Аналіз графічних інтерфейсів за допомогою трекінга очей

за освітньою програмою Інженерія програмного забезпечення  
зі спеціальності: 121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

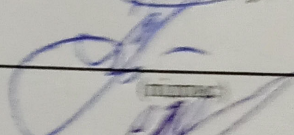
Виконав: студент групи: ПЗ2027

  
(підпис студента)

Ілля ФРОЛОВ

(підпис)

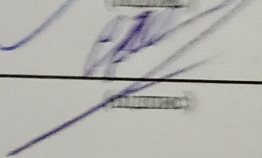
Керівник:

  
(підпис)

д-р. Олександра ГИРЬОВА

(підпис)

Нормоконтролер:

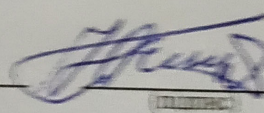
  
(підпис)

д-р. Світлана ВОЛКОВА

(підпис)

Консультанти:

Техніко-економічні розрахунки  
(завда розділу)

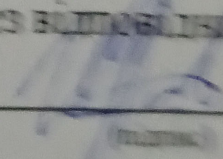
  
(підпис)

д-р. Микола ГЕННИЙ

(підпис)

Засвідчую, що у цій роботі немає заплігачень  
праць інших авторів без відповідних посилань.

Студент

  
(підпис)

Дніпро - 2022\_ рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Український державний університет науки і технологій  
Кафедра Комп'ютерні інформаційні технології

«ДО ЗАХИСТУ»

Завідувач кафедри

\_\_\_\_\_ Вадим ГОРЯЧКІН

«\_\_\_» \_\_\_\_\_ 20\_\_р.

**ДИПЛОМНА РОБОТА**

на здобуття освітнього ступеня «магістр»

Галузь знань **12 Інформаційні технології**

Спеціальність **121 Інженерія програмного забезпечення**

Тема **Аналіз графічних інтерфейсів за допомогою трекінга очей**

Theme **Analysis of graphical interfaces using eye tracking**

Керівник дипломної роботи доц. \_\_\_\_\_ Олександра ГОРБОВА

Нормоконтролер доц. \_\_\_\_\_ Світлана ВОЛКОВА

Студент групи ПЗ2027 \_\_\_\_\_ Ілля ФРОЛОВ

Student Illia FROLOV



Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем

Кафедра: Комп'ютерні інформаційні технології

Рівень вищої освіти: Магістр

Освітня програма: Інженерія програмного забезпечення

Спеціальність: 121 Інженерія програмного забезпечення

(шифр та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри \_\_\_\_\_

\_\_\_\_\_ Вадим ГОРЯЧКІН  
(підпис) (Ім'я ПРІЗВИЩЕ)

Дата \_\_\_\_\_

**ЗАВДАННЯ**

на кваліфікаційну роботу \_\_\_\_\_ магістр  
(ступінь вищої освіти))

студенту \_\_\_\_\_ Фролов Ілля Валерійович  
(Прізвище, Ім'я По батькові)

1. Тема роботи: \_\_\_\_\_ Аналіз графічних інтерфейсів за допомогою трекінга очей

Керівник роботи: \_\_\_\_\_ Горбова Олександра Вікторівна  
(Прізвище, Ім'я По батькові)

затверджені наказом від \_\_\_\_\_ «\_\_» \_\_\_\_\_ 2022 р. № \_\_\_\_\_

2. Строк подання студентом роботи: \_\_\_\_\_ .\_\_\_.202\_\_р.

3. Вихідні дані до роботи: \_\_\_\_\_ аналіз графічних інтерфейсів за допомогою трекінга очей

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Аналітична частина: \_\_\_\_\_ аналіз сучасних методів розробки і досліджень зручності використання веб-сайтів

4.2 Основна частина: \_\_\_\_\_ огляд існуючих рішень, проектування та розробка програмного забезпечення

4.3 Економічна частина: \_\_\_\_\_ техніко-економічні розрахунки

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):  
\_\_\_\_\_ доповідь, презентація, демонстраційне відео

## 6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)
Техніко-економічні розрахунки	доц. Микола ГНЕНИЙ		

## КАЛЕНДАРНИЙ ПЛАН

№ пор.	Назва розділів дипломної роботи	Термін виконання розділів роботи	Примітка
1	Вступ		
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	03.03.22 - 15.03.22	від 70 джерел
3	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	15.03.22 - 01.04.22	
4	Постановка задачі, технічне завдання	01.04.22 – 31.03.22	30%
5	Техніко-економічні показники	31.03.22 - 16.05.22	
6	Розробка інструментальних засобів дослідження	16.05.22 - 25.08.22	
7	Виконання досліджень	25.08.22 - 08.10.22	60%
8	Оформлення тез доповідей	08.10.22 - 10.10.22	
9	Оформлення статті у фаховий журнал	10.10.22 - 14.10.22	
10	Оформлення пояснювальної записки	14.10.22 - 12.12.22	
11	Розробка демонстраційних матеріалів	12.12.22 - 19.12.22	100%

Студент

(підпис)

Ілля ФРОЛОВ

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Олександра ГОРБОВА

(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту містить 76 сторінок, 21 рис., 1 таблиця, 3 додатки, 70 джерел.

Об'єкт дослідження – аналіз графічних інтерфейсів за допомогою технології трекінгу очей.

Метою роботи є дослідження графічних інтерфейсів за допомогою технології відстеження відповідності фокусу зору зі структурою інформації на моніторі.

Методи дослідження здійснені з використанням методів обробки відеокадрів, методів розпізнавання рис обличчя на відеокадрі, методів розпізнавання зіниці ока, методів підвищення точності розпізнавання фокусу зору, методів підрахунку координат фокусу зору, методів відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера.

Наукова новизна. Запропоновано систему відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера з використанням доступних апаратних засобів: звичайної відеокамери, вбудованої в ноутбук та зовнішньої веб-камери. Запропонована система забезпечує більшу точність відстеження фокусу зору порівняно з іншими системами.

Практичне значення одержаних результатів. Розроблена система відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера для аналізу графічних інтерфейсів дозволяє зробити висновки щодо необхідності редагування інформаційного контенту, для покращення концентрації уваги.

## Зміст

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ .....	9
ВСТУП.....	10
РОЗДІЛ 1 Трекінг очей та сфера його застосування.....	12
1.1 Призначення та сфера застосування трекінгу очей.....	12
1.2 Метрики айтрекінгу .....	14
1.3 Дослідження веб сторінок за допомогою айтрекінгу .....	16
1.4 Види та характеристики айтрекерів .....	17
1.5 Аргументація застосування айтрекінгу .....	18
1.5.1 Огляд сучасних айтрекерів .....	21
1.5.2 Огляд бібліотек комп'ютерного зору для дослідження розробки засобів відстеження фокусу зору .....	23
Висновок до розділу 1 .....	25
РОЗДІЛ 2 Дослідження та оцінка графічного інтерфейсу за допомогою айтрекінгу .....	27
2.1 Методи досліджень з використанням айтрекінгу, їх переваги та недоліки. ....	27
2.1.1 Мобільний айтрекінг .....	33
2.1.2 Стаціонарний айтрекінг .....	34
2.1.3 Моніторний айтрекінг .....	35
2.2 Основні показники та критерії оцінки ефективності аналізу графічних інтерфейсів за допомогою айтрекінгу .....	35
Висновок до розділу 2 .....	42
Розділ 3 Розробка інструментальних засобів для дослідження графічного інтерфейсу за допомогою айтрекінгу .....	43
3.1 Зовнішнє проектування.....	43
3.1.1 Формалізація задачі.....	43
3.1.2 Базова архітектура системи .....	45
3.2 Внутрішнє проектування .....	46
3.2.1 Технологічна платформа.....	46
3.2.2 Використані принципи проектування .....	51
3.2.3 Розробка інтерфейсу користувача .....	51
3.3 Тестування та налагоджування програми.....	55
Висновок до розділу 3 .....	56

Розділ 4 Дослідження відповідності фокусу зору із структурою інформації на моніторі.....	58
4.1 Дослідження на основі рекламного банеру на веб-сторінці .....	59
4.2 Дослідження абсолютної тривалості погляду на веб-сторінці 1 .....	59
4.3 Дослідження абсолютної тривалості погляду на веб сторінці 2.....	63
4.4 Дослідження абсолютної тривалості погляду на веб сторінці 3.....	66
Висновки до розділу 4 .....	67
ВИСНОВКИ .....	69
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	70
ДОДАТКИ .....	79



## **СПИСОК УМОВНИХ ПОЗНАЧЕНЬ**

ПК – персональний комп'ютер;

ПЗ – програмне забезпечення;

ІС – інформаційна система;

ІК – інформативна компетентність;

GUI – графічний інтерфейс користувача;

## ВСТУП

Актуальність роботи.

Айтрекінг — це метод спостереження за рухами очей людини за допомогою спеціального пристрою, який називається окуляром. Відстеження очей використовується для виявлення недоліків у взаємодії між людиною та машиною та додатково покращується шляхом аналізу людських емоцій, тобто рухів очей людини.

Людино-машинна взаємодія як окремий інженерний напрям виникла в 50-60-х роках 20 століття. Створення надійних і безпечних інформаційних систем не є достатньою умовою успішного використання. Зручність використання є одним із найважливіших факторів, що визначають успіх компаній, які розробляють і продають інформаційні системи та програмне забезпечення. Те наскільки зручно користуватись продуктом буде безпосереднє впливати на те чи прийде користувач повторно. Це досягається шляхом розробки простих, зрозумілих і зручних інтерфейсів. Наразі майже кожна ІТ-компанія має конкретну назву посади, наприклад, розробник інтегральних комунікацій або дизайнер, основною роботою якого є UX/UI (користувацький досвід/взаємодія з користувачем) — дослідження, дизайн і розробка інтерфейсу користувача. Основним завданням у розробці ІС є оцінка якості. Сучасні технології, зокрема оптичні технології, допомагають розвивати мікросхеми в цій галузі [1].

В останні десятиліття зростає потреба в дослідженні нового типу суспільства, відомого як інформаційне суспільство. Саме розвиток сучасних інформаційно-комунікаційних технологій та збільшення обсягів інформації все більше визначає зміст нашого часу. Потенційне використання обміну технологіями сьогодні полегшує та пришвидшує створення та продаж пакетів обслуговування клієнтів, вирішення фінансових та операційних завдань управління, маркетингових планів, а також збільшення конкуренції та продажів. Тому ІТ-внесок з кожним роком збільшується і сприяє зміцненню позицій країни на світовому ринку ІТ-технологій.

Предмет дослідження.

Предметом дослідження є засоби реалізації аналізу графічних інтерфейсів за допомогою відстеження погляду.

Метою роботи є дослідження графічних інтерфейсів за допомогою технології відстеження відповідності фокусу зору зі структурою інформації на моніторі.

Основним завданням є дослідження та реалізація засобів відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера з використанням доступних апаратних засобів для здійснення аналізу.

Методи дослідження.

Дослідження здійснювалося за допомогою опосередкованого спостереження за діями та рухами зіниць ока респондента із застосуванням розробленої системи. Отриманні данні у вигляді масиву координат піддавалися математичній обробці за допомогою програмного забезпечення MS Excel. Також масив координат використовувався для побудови теплових карт досліджуваних інтерфейсів.

Наукова новизна.

Запропоновано систему відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера з використанням доступних апаратних засобів: звичайної веб-камери. В порівнянні до інших досліджень були використанні одразу 2 методи аналізу: аналіз теплових карт та аналіз довжини сакад, що дає змогу більш комплексно оцінити досліджуваний інтерфейс

Практичне значення одержаних результатів.

Розроблена система відстеження відповідності фокусу зору зі структурою інформації на моніторі комп'ютера для аналізу графічних інтерфейсів дозволяє зробити висновки щодо необхідності редагування інформаційного контенту, для покращення концентрації уваги.

Апробація результатів дослідження та публікації.

На основні положень магістерської роботи було створено тези доповіді, котрі були схвалені на XVI Міжнародної науково-практичної конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті», яка відбулася 14-15 грудня 2022 року в Українському державному університеті науки та технологій в онлайн форматі, див додаток В.

## РОЗДІЛ 1 Трекінг очей та сфера його застосування

### 1.1 Призначення та сфера застосування трекінгу очей

Відстеження очей – це процес вимірювання властивостей і рухів очей. Він використовується в дослідженнях для виявлення та покращення взаємодії людини та машини, а також для вимірювання характеристик руху очей і фізичних характеристик ока, таких як розмір зіниці.

Айтрекер — це пристрій, який фіксує рухи очей людини, дивлячись на екран, фізичний об'єкт або навколишнє середовище в цілому. Було розроблено декілька типів оптичних зображень, деякі з яких розміщуються на об'єкті або поза окулярами, інші розміщуються перед об'єктом і прикріплюються до нижньої частини окулярів, а деякі використовують веб-камери або камери телефону [2].

Давайте розглянемо елементи процесу відстеження джерела. Трекер освітлює око об'єкта інфрачервоним світлом і записує видиме інфрачервоне світло сітківки об'єкта. Ця система дозволяє окоміру визначати центр зіниці респондента та аналізувати видиме інфрачервоне світло, відбивається від ока. Точку, на яку спрямована увага людини, можна визначити за положенням центру зіниці щодо зору рогівки. Коли людина повертає голову, щоб подивитися на ту саму точку, співвідношення між центром зіниці та зображенням рогівки залишається незмінним. Навіть якщо людина рухається, пристрій відстеження знає, що вона дивиться в ту саму точку.

Сучасні айтрекери мають дві основні частини. По-перше, це ближнє інфрачервоне світло, яке впливає на людське око. Друга частина – відеокамера. Камера фокусується на очах респондентів і записує їхні думки. Позиція джерела розраховується програмним забезпеченням і розміщується на веб-сайті як приклад. Очі білки сприймають довжини хвиль, невидимі для людини. Іншими словами, він використовує довжини хвиль, які відбиваються в оці, не заважаючи людині.

Крім того, сучасні технології дозволяють контролювати рухи очей за допомогою нейронних мереж і стандартних веб-камер, що значно збільшує кількість користувачів, оскільки не потрібно спеціального обладнання.

Одним із таких методів є бібліотека оптичного пошуку WebGazer.js. Він використовує звичайну веб-камеру для визначення погляду відвідувача сайту в реальному часі. Включена модель спостереження спостерігає за тим, як відвідувачі взаємодіють із веб-сайтом, коригуючи та автоматично тренуючи кореляцію між функцією очей і положенням на екрані. Повністю написаний на JavaScript, WebGazer.js можна інтегрувати в будь-який веб-сайт, який дозволяє краще розуміти своїх відвідувачів і трансформувати взаємодію користувачів лише за допомогою кількох рядків коду. WebGazer.js повністю працює в браузері клієнта, тому немає необхідності надсилати відео на сервер і не потрібні дозволи користувача для доступу до веб-камери.

Ця бібліотека може бути використана будь-ким через її відкритий код, але її переваги не є єдиними перевагами. Він також прогнозує погляд у режимі реального часу для загального перегляду, оскільки існують різні типи передбачення очей і різні компоненти виявлення очей, які забезпечують краще розпізнавання. Він також автоматично обробляє клацання та рух курсору та легко інтегрується лише з кількома рядками JavaScript, що робить його доступним для багатьох розробників.

Рішучість і зосередженість часто пов'язані. Коли наші очі дивляться на щось, наші органи чуття допомагають нам обробляти й оцінювати те, що ми бачимо. Отримавши інформацію про напрямок людської уваги в будь-який момент, дослідники можуть визначити та покращити обмеження дизайну та взаємодію людини та машини. Рух очей абсолютно незалежний. Це означає, що одна й та сама людина може бачити те саме по-різному, якщо у нього різні мотиви.

Відстеження очей має два основних способи використання:

1. Використання методів дослідження.
2. Такі інструменти, як інструменти введення інформації та контролю.

Як пристрій введення та керування рухи очей використовуються як керуючі сигнали в комп'ютерних системах, замінюючи або доповнюючи комп'ютерні миші та клавіатури [3]. Люди з обмеженими можливостями, наприклад люди з синдромом Альцгеймера або церебральним паралічем, використовують візуальні засоби для спілкування. Відстеження очей можна використовувати в таких дослідженнях, як

інженерна психологія, ІК-дизайн, дослідження користувачів, ІК-дизайн, відеоігри та оцінка мобільних додатків

Використання відстеження очей у дизайні ІК має такі цілі:

1. Дізнатися більше про потреби потенційних користувачів, уподобання, цілі та процеси. Те, що дослідник має окуляр і знає, як ним користуватися, не означає, що він повинен використовувати його в усіх дослідженнях.
2. Дослідники повинні знати та вміти застосовувати айтрекінг для отримання надійної та необхідної інформації. Такий процес має включати планування, підготовку та проведення дослідження, а також аналіз та інтерпретацію зібраних даних стеження за очима, і його слід розглядати як повну методологію.

## 1.2 Метрики айтрекінгу

Метрики використовуються як залежні змінні в дослідженнях. Це дозволяє досліднику обчислити значення метрики для одного дослідження та порівняти його з тим самим значенням метрики, розрахованим для іншого набору досліджень. Відстеження джерела допомагає проводити детальні вимірювання не лише цілих завдань, але й окремих частин завдання. Важливо підібрати номенклатуру, необхідну для дослідження за допомогою айтрекінгу, враховуючи особливості досліджуваного. Існує понад 100 параметрів оптичного відстеження. Деякі з них дуже легко розрахувати та інтерпретувати отримані результати. До них відносяться:

1. Загальний час, витрачений респондентами на розгляд теми;
2. Кількість респондентів, які спостерігали та розглядали тему дослідження;
3. Моменти фіксації респондента на окремому елементі;
4. Швидкість виконання поставленої задачі;
5. Швидкість знаходження потрібного елементу;
6. Зручність користування веб-сторінкою;
7. Загальний час витрачений респондентами на виконання певної дії;
8. Загальний час витрачений респондентами на знаходження певного елементу, та багато інших.

Загальна класифікація метрик айтрекінгу представлена на Рисунку 1.1.

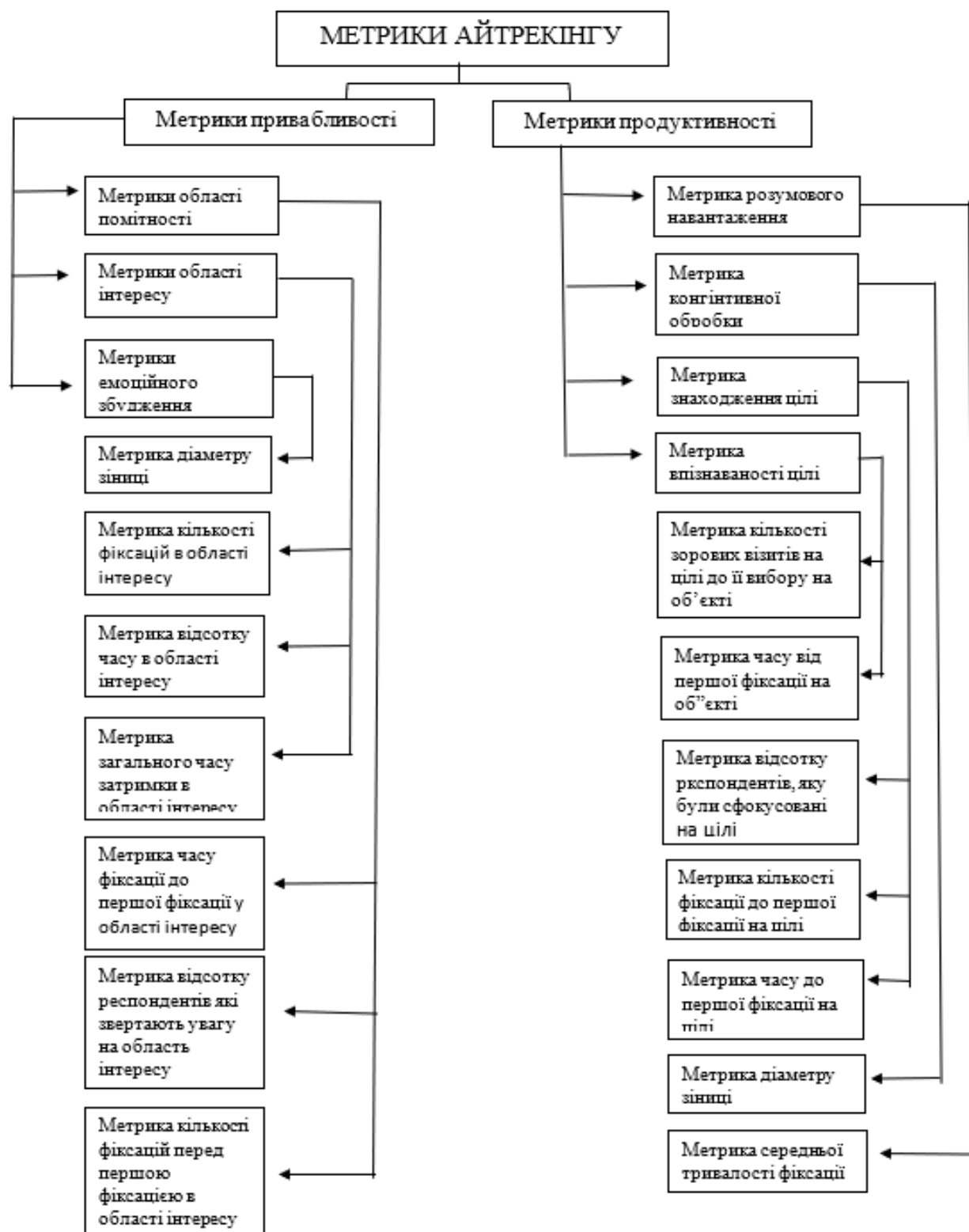


Рисунок 1.1 Загальна класифікація метрик айтрекінгу

Один і той самий параметр може виявляти різні явища в контексті різних мотивацій і цілей. Наприклад, він може повідомити досліднику, які ситуації можуть виникнути, коли респондент дивиться на об'єкт дослідження протягом тривалого

періоду часу. Це означає, з одного боку, що суб'єкт захопив і привернув увагу респондента, а з іншого боку, це може виникнути, коли респонденту важко зрозуміти або вирішити певну проблему [5].

Теплові карти широко використовуються у візуальній візуалізації даних. Він забезпечує значення змінної, такої як концентрація уваги, як колір, де кількість «теплоти» відповідає рівню наданої інформації [4].

Індекс привабливості відповідає на такі запитання: які місця на веб-сторінках роблять вашу рекламу найбільш запам'ятовуваною? Вимірювання продуктивності дають відповіді на наступні запитання: чи полегшує перероблений макет веб-сторінки пошук і порівняння інформації?

### 1.3 Дослідження веб сторінок за допомогою айтрекінгу

Розроблено веб-додаток який може відстежувати та аналізувати веб-сторінки. Цей вид роботи може мати багато можливостей для активного застосування в реальному житті, допомагаючи покращити сайт, зробити його більш зручним для користувачів, збільшити продажі та інші переваги, досягнуті за допомогою технологій. А також ефективно показувати рекламу на основі видимих даних. Люди з обмеженими можливостями також можуть отримати значну користь.

На цьому етапі додаток може генерувати дані, які мають велику кількість коефіцієнтів кореляції, вони передбачають сприйняття користувачем графічного інтерфейсу, яке можна використовувати в майбутніх дослідженнях. Цей додаток також може створити ілюзію того, що користувачі зробили клік на сторінці. Також можна знайти інформацію про те, на що користувач звертав увагу, де зупинявся, а що взагалі не помічав, це допоможе створювати зручніші та зрозуміліші інтерфейси.

Ці можливості не тільки роблять використання більш захоплюючим і простим, але також дозволяють нам дізнатися більше від взаємодії користувачів на сайті[6].



## 1.4 Види та характеристики айтрекерів

При виборі оптичного пошуку основним завданням є визначення мети пошуку. Розрізняють два види лінз: поверхневі та дистанційні. Основною відмінністю між ними є розташування обладнання.

Дистанційне відстеження очей використовується в опитуваннях, де тестовий елемент представлений у фіксованому місці, наприклад опитуваннях із стимулами на екрані, такими як веб-сторінки, зображення та відео, у той час як респондент сидить або стоїть у тому самому місці. Пристрої для спостереження, встановлені на голові, використовуються в опитуваннях, які вимагають від респондентів руху та взаємодії з візуальним об'єктом або людиною, наприклад тест-об'єктом, їжею чи іншим тестовим предметом. Дистанційне відстеження очей існує у двох формах: відстеження очей, вбудоване в монітори, відоме як інтегроване відстеження очей, і відстеження очей, яке є незалежною комп'ютерною програмою [7].

Оскільки об'єкт і транспондер є нерухомими, для відстеження віртуальних об'єктів можна використовувати автономне віддалене відстеження джерела. Наприклад, коли респондент грає в гру, він відводить погляд від телевізора, взаємодіє з екранними меню або переглядає відео. Деякі з цих пристроїв можна встановлювати в автомобілях для контролю уваги водія. Автономне відстеження джерела можна використовувати для визначення того, як респонденти взаємодіють із мобільними пристроями. З огляду на те, що це можна використовувати для аналізу, тип обраного методу забезпечує гнучкість, оптимізацію даних, з одного боку, компроміс між фактичними умовами тестування та середовищем між науковим контролем і моделюванням [8].

Технічні особливості оптичного трекінгу:

WebGazer відносно простий з погляду комп'ютерного зору — у ньому немає явних тривимірних міркувань, як у складніших трекарах. Ця простота дозволяє працювати в режимі реального часу через браузерний JavaScript. Відсутність тривимірних міркувань зазвичай призводить до ненадійних прогнозів, але головна новинка WebGazer — можливість постійного самокалібрування на основі взаємозв'язків між курсором та поглядом. Це не тільки усуває необхідність у

початкових сеансах калібрування, а й означає, що користувачі можуть вільно переміщатися, а WebGazer вивчить нові зіставлення між положенням зіниці, особливостями ока та координатами на екрані.

WebGazer був розроблений за допомогою трьох різних бібліотек розпізнавання рис обличчя: clmtrackr, jsobjectdetect та tracking.js.

Усі три реалізують різні алгоритми зору JavaScript. Jsobjectdetect і tracking.js виявляють обличчя і очі і повертають навколишні прямокутники. Замість того, щоб використовувати весь відеокадр, розробники спочатку виконуємо розпізнавання облич для більш точного виявлення очей у його верхній половині, що прискорює прогноз погляду і пригнічує помилкові спрацьовування. Clmtrackr виконує більш реалістичне припасування контуру обличчя та очей. Щоб забезпечити узгоджене введення для WebGazer, розробники використовували найменший прямокутник, який відповідає контуру ока [9].

### 1.5 Аргументація застосування айтрекінгу

Одним із визначальних факторів успіху сучасних комерційних, технологічних, соціальних і наукових проектів є графічний інтерфейс, який визначає ефективність взаємодії людини і машини та сприйняття програмного забезпечення, забезпечує доступ до функціональних компонентів програми та забезпечує задоволення. Користувачі програмного забезпечення або виконувани файли. Основним завданням, яке стоїть сьогодні перед розробниками ІС, є розробка гнучких, простих у використанні та інтуїтивно зрозумілих орієнтованих на користувача інтерфейсів, які забезпечують повний доступ до функціональних можливостей програмного забезпечення.

Оскільки зручність використання є найважливішим критерієм для користувачів програмного забезпечення, це один із ключових факторів, що визначають успіх компаній, які розробляють і продають інформаційні системи та програмне забезпечення. Простота використання ІР – показник, який визначає кількість зусиль, необхідних для вивчення того, як працює програмне забезпечення за допомогою запропонованого інтерфейсу. Ефективність, прибутковість і

задоволення, з якими користувач може виконувати певне завдання в даному середовищі [10].

Розробники інтерфейсу повинні чітко розуміти, що дизайн повинен відповідати потребам користувача. Перш ніж розпочати проектування ІС, розробники можуть полегшити собі роботу, зосередившись на спільності вимог до інтерфейсу, а потім досліджуючи різні вимоги на основі конкретних завдань.

Інтерфейс користувача слід розробляти з урахуванням моделей мислення та поведінки людини. Тому основою проектування взаємодії людини і машини є базові принципи когнітивної психології (наприклад, гештальттеорія), а основним критерієм успіху є задоволеність мети користувача та емоційне задоволення користувача від роботи з програмним забезпеченням.

Для використання сучасних методів проектування програмного забезпечення необхідні належним чином розроблені методи та інструменти проектування ІС. Технологія відстеження очей — це сучасна техніка, яка використовується для оцінки зручності використання дизайну, зокрема ІЧ, процес відстеження рухів очей людини за допомогою спеціального пристрою (itracker).

Відстеження очей використовується для виявлення недоліків у взаємодії людини з машиною та вдосконалюється шляхом аналізу уваги людини (аналіз рухів очей людини [11]). Тому актуальним завданням на даний час є оцінка простоти використання ІК, а також відповідності принципу Гештальт за рахунок використання технології відстеження очей у процесі проектування ІК.

Технологія стеження за очима використовується в дослідженнях, щоб знайти та покращити взаємодію людини з машиною, а також для вимірювання характеристик руху очей і фізичних властивостей ока, таких як розмір зіниці. Метою використання технології відстеження очей в ІЧ-проекуванні є: Теплові карти широко використовуються під час візуалізації даних відстеження очей.

Останнім часом, оскільки використання технології відстеження очей стає все більш популярним, дослідники приділяють все більше уваги вивченню цієї технології в дизайні графічного інтерфейсу.

Так, завдання полягає в розробці інтерфейсу на основі візуального спостереження. Оцінка дані за допомогою технології відстеження очей це мультимодальний метод взаємодії, який використовує поєднання технології відстеження очей і розпізнавання жестів за допомогою контролера жестів і значно покращує час орієнтації для маленьких точок.

Також використовують технологію стеження за очима для взаємодії користувача з комп'ютером, щоб підвищити стабільність, мобільність і зручність.

Існує також дослідницький метод спостереження, який використовує методи стеження за очима для оцінки двох загальноприйнятих методів візуалізації онтології та представляє собою результати, отримані в результаті експериментального дослідження з використанням технології відстеження очей у контексті досвіду користувача під час виконання пошукового завдання в Інтернеті, що свідчить про те, що відображення результатів у пошуковій системі може впливати на прийняття рішень учасниками та поведінку виробничого процесу та відповідні посилення опцій [12].

У роботі був використаний метод оцінки за допомогою абсолютної тривалості та терлових карт поглядів користувачів.

Багато практиків-розробників ПЗ діляться на два протилежні табори: тих, хто виступає за використання технології відстеження очей у дослідженнях, і тих, хто виступає проти цього [10]. Прихильники хочуть, щоб відстеження очей використовувалося в усіх дослідженнях, незалежно від мети. Опоненти стверджують, що стеження за очима не має великого значення. Техніка стеження за очима може бути цінним доповненням до деяких досліджень ІР (наприклад, для оцінки корисності ІР, а також дотримання принципів гештальту), тоді як в інших вона може бути неефективною або дорогою.

Варто зазначити, що недоцільно виконувати айтрекінг, поки не визначені чіткі цілі дослідження. Крім того, айтрекінг є досить дорогим (дороге обладнання та час, щоб його обслуговували професіонали). Для проведення айтрекінгу необхідно чітко сформулювати цілі та завдання, пов'язані з предметом навчання. Рішення про те, чи

потрібне відстеження очей для конкретного дослідження, залежить від відповіді на три ключові запитання на рисунку 1.2 [11].

Перше і найважливіше питання: «чи дасть спостереження інформацію, необхідну для досягнення цілей дослідження?» Друге і третє питання спонукають дослідників розглянути реальну цінність обраного ними методу. Перше питання справді потребує відповіді. Якщо відповідь на друге та/або третє питання ствердна, екзаменатор матиме можливість і право оглянути око для огляду.

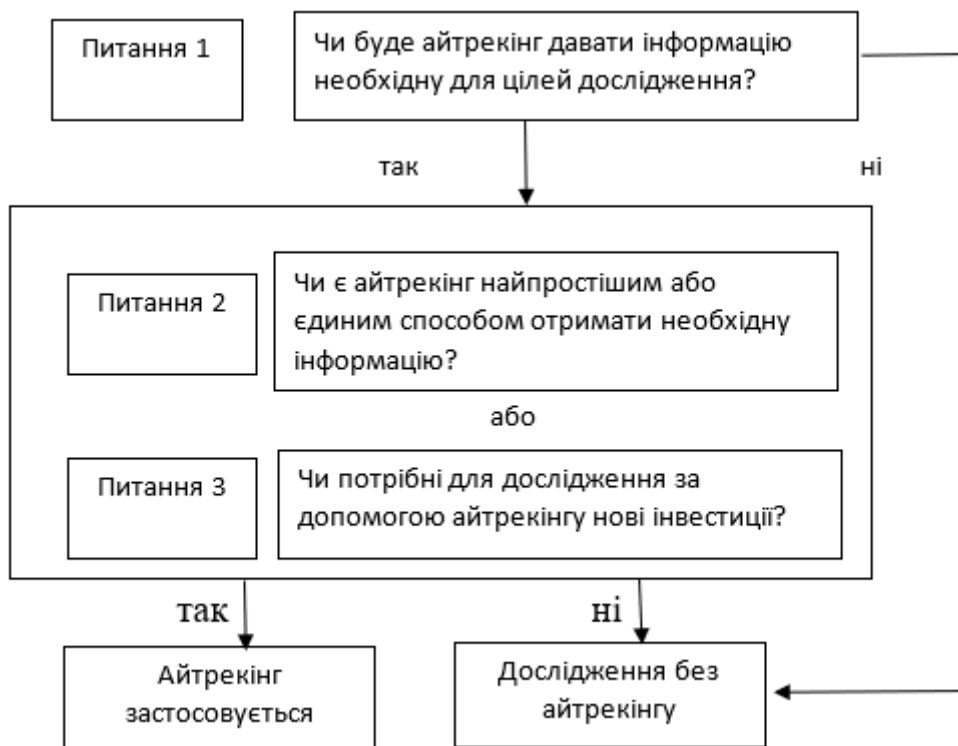


Рисунок 1.2 Набір питань для визначення необхідності застосування айтрекінгу

### 1.5.1 Огляд сучасних айтрекерів

Окулографія - це метод реєстрації рухів очей. Ця система складається з двох частин. Перша частина - система, яку потрібно носити - окуляри або спеціальні оптичні системи. На голову людини встановлюється система відеокамер для фіксації того, що відбувається перед людиною та відображення джерела інфрачервоного випромінювання та «композитних» сигналів в оптоволокні ока людини. відеокамера. Цей вид перевірки широко використовується для аналізу складних видів професійної діяльності ( водіння, роботи з технікою тощо). На сьогоднішній день

лідером у розробці таких систем є шведська компанія Tobii. Компанія виробляє пристрій, який за допомогою очей відстежує напрямок руху людини. Прикладом такого продукту є мобільна версія монітора Tobii Glass. Точність вимірювання цієї моделі дуже висока 0,5°, а частота зображення становить 30 Гц, що достатньо для використання під час перевірки. Ціна цього продукту становить 20 000 доларів [12].

Основним недоліком окулярів, який можна носити, є те, що голова людини не звернена до камери, тому камера не може виявити рухи очей за допомогою оптичного програмного забезпечення. До другої категорії відносяться дистанційні оптичні треки. Зовні це виглядає як окремий блок, розміщений перед робочим столом на екрані монітора. Пристрій стеження складається з двох камер з інфрачервоними джерелами, які вловлюють світлові промені, відбиті від очей людини.

Є ще один вид айтрекеру з інфрачервоним портом для декількох відеокамер від 2 до 8 камер. Для збільшення можливості запису зорової активності перед людиною розміщують камеру [13]. Цей сканер широко використовується для дослідження двовимірних структур.

Порівняно з айтрекерами першої категорії, він повинен враховувати допустиму кількість рухів голови, тобто наскільки вона може рухатися вгору та вниз з боку в бік. Прикладом дистанційного продукту є Tobii X2. Точність вимірювання даної моделі 0,4°, частота кадрів 30Hz, допустима відстань об'єкта 40-90cm, допустимий обхват голови 50x36cm. Ціна цього продукту становить 10 000 доларів.

Дистанційні треки повинні бути точнішими, ніж окуляри, але це не так. Термін дії залежить від типу дослідження, яке використовується як спостерігач. Головний убір підходить для досліджень, де люди можуть вільно пересуватися. Джерело дистанційних трекерів має обмежену область, в якій головка перетворювача може рухатися без шкоди для точності отриманих даних. Тому при виборі оптичного приладу необхідно враховувати вимоги об'єкта дослідження, оскільки від цього залежать попередні результати. Наприклад, використання віддаленого перегляду ідеально підходить для вивчення зображень або тексту, що відображаються на екрані комп'ютера. Використання окулярів доцільно, коли

респондентам для проведення дослідження необхідно побачити об'єкти навколишнього середовища [14].

### 1.5.2 Огляд бібліотек комп'ютерного зору для дослідження розробки засобів відстеження фокусу зору

Комп'ютерний зір – програмно-технічний засіб, що забезпечує зчитування зображень, їх обробку та отримання результатів у цифровому форматі, придатному для практичного використання в режимі реального часу [15]. Для розробки систем комп'ютерного зору використовуються спеціалізовані бібліотеки. Для визначення найбільш підходящої бібліотеки для вирішення поставлених завдань проводиться її аналіз. OpenCV — це бібліотека з відкритим вихідним кодом алгоритмів комп'ютерного зору, обробки зображень і багатоцільових алгоритмів. OpenCV - одна з найпопулярніших бібліотек для вирішення проблем комп'ютерного зору. Реалізовано на мовах програмування C/C++, розроблено для Python, Java, Ruby, Matlab, Lua та інших мов. Він підтримує різні платформи: Microsoft Windows, Linux, Mac OS, Android, iOS. Є документація в Інтернеті. Бібліотека містить більше 500 різних функцій і алгоритмів. Зокрема, є багато вдосконалених алгоритмів, пов'язаних з обробкою та аналізом відеосигналів. Реалізовано фільтрацію, дослідження контурів, трансформацію геометрії, алгоритми аналізу руху, виявлення об'єктів, фокусування та багато інших функцій. Підтримується робота з файлами XML [16]. Незважаючи на всі можливості, переваги та переваги бібліотеки OpenCV, слід відзначити її недоліки. Бібліотека OpenCV повна розширених додаткових функцій, що робить її непотрібною у використанні та дуже важкою для вивчення. VXL є частиною бібліотеки, можливості якої використовуються в комп'ютерній теорії та наукових дослідженнях. Він написаний на мові програмування C++ і може використовуватися в багатьох операційних системах. VXL забезпечує високу продуктивність при обробці матриці. VXL також працює з зображеннями високої роздільної здатності. Однак необхідно знати проблеми, які виникають в процесі його збору та монтажу. LTI-lib – це бібліотека об'єктно-орієнтованих алгоритмів і структур даних, які використовуються для обробки зображень у сфері

комп'ютерного зору [17]. Основною метою розробки цієї бібліотеки було створення об'єктно-орієнтованих алгоритмів на мові C++, що в більшості випадків спрощує використання та обслуговування коду. Тому при аналізі бібліотек комп'ютерного зору слід звернути увагу на швидкість, оскільки від цього залежить якість і швидкість роботи всієї підсистеми розпізнавання об'єктів в цілому. Порівняльний аналіз OpenCV, OpenCV з використанням IPP, VXL і розділу LTI для розрахунку швидкості виконання різних завдань: - Дискретне перетворення Фур'є; - розмір зображення; - рух очей; - нервова система.

Для кожної операції передбачено чотири різні індикатори: чотири стовпці кожного параметра показують оцінку, що відповідає часу роботи кожної з даних бібліотек. У всіх випадках OpenCV перевершує інші бібліотеки [18].

Підводячи підсумок, можемо побачити на Рисунку 1.3 практичні області, для яких можна застосувати відстеження очей у дослідженнях.

#### Практичне застосування айтрекінгу

<b>Якісне розуміння (оцінювання)</b> (Айтрекінг підтримує проведення досліджень, надаючи інформацію, на основі якої можуть бути сформовані рекомендації для покращення дизайну)	
<b>Виявлення проблем щодо зручності використання (usability)</b>	<b>Пояснення проблем щодо зручності використання (usability)</b>
<b>Кількісне розуміння (оцінювання)</b> (Айтрекінг підтримує проведення досліджень, надаючи можливості для UX аналізу та порівняння дизайнів)	
<b>Оцінювання привабливості</b>	<b>Оцінювання ефективності</b>

Рисинок 1.3 Практичне застосування айтрекінгу



Питання, на які допомагає відповісти айтрекінг: 1) Чому респонденти роблять неадекватні дії? – опитування сайту показало, що лише невелика частина респондентів знайшла потрібне посилання. Це може бути тому, що важко знайти потрібне посилання, або тому, що посилання погано представлено і його важко пов'язати з метою завдання. Для більш детального вивчення варто застосувати айтрекінг, яким може бути неправильне розташування посилання, його низький зміст, захаращеність елементів на сторінці, погані мітки або відсутність орієнтирів, які співпраця по посиланню виявляє; 2) Чому респонденту знадобилося більше часу, ніж очікувалося, щоб вжити належних дій? - це питання виявляє успішне, але неефективне виконання дій; 3) Чому респонденти не отримують необхідної інформації? - айтрекінг не обмежується поясненням фізичних дій респондентів [19]. Цей процес також включає коментарі учасників. Відстеження очей може вказувати на причини випадків, коли інтерфейси не повністю передають необхідну інформацію респондентам.

### Висновок до розділу 1

Під час проведення спостережень зазвичай рекомендується протокол усного огляду, оскільки коментарі респондентів, отримані після виконання завдання, можуть не вплинути на виконання самого завдання порівняно з використанням протоколу. Паралельний усний протокол також використовується в обсерваційних дослідженнях, переважно дослідженнях, спрямованих на виявлення проблем зручності використання. У цих дослідженнях відстеження очей використовується як інструмент для спостереження, а не з метою аналізу та узагальнення даних руху очей [20].

Візуальний пошук можна використовувати як доповнення до традиційних методів дослідження. Крім того, це може допомогти пояснити поведінку респондентів, які купують або відмовляються. Спостереження також може побачити проблеми в дослідженні поведінки респондентів, які мало говорять (наприклад, діти). Крім того, спостереження можна використовувати для інтерпретації результатів неспостережних досліджень [21].

Підсумовуючи, можна побачити, що аналіз графічних інтерфейсів за допомогою айтрекінгу має як переваги, так і недоліки як з технічної, так і соціальної точки зору.

## **РОЗДІЛ 2 Дослідження та оцінка графічного інтерфейсу за допомогою айтрекінгу**

Графічний інтерфейс користувача — це тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями за допомогою графічних зображень і візуальних інструкцій, на відміну від текстового інтерфейсу, який базується на використанні тексту, текстових команд і текстових інструкцій. навігація. зробити їх компетентними. Виконання будь-якої дії в ньому є прямим маніпулюванням графічними елементами. Окрім комп'ютерів, GUI використовуються в мобільних пристроях, таких як мобільні телефони, планшети, електронні книги, портативні медіаплеєри тощо [22].

З іншого боку, графічний інтерфейс є «обличчям» системи, і від його мислення залежить ефективність роботи з системою. Фактори, що впливають на ефективність роботи, залежать формально у вигляді конкретних вимог до окремих об'єктів, але повинні бути враховані у вигляді загальних рекомендацій і принципів побудови інтерфейсу. Щоб визначити це, доцільно скористатися джерельним аналізом. Він легко може визначити, що цей інтерфейс є дружнім до користувача, тому що користувач є «клеєм», який з'єднує всі частини, які об'єднуються в програмний продукт.

Крім того, за допомогою спостережень ми можемо оцінити рівень задоволеності користувачів; аналізувати поведінку, пов'язану із зовнішнім виглядом, поведінкою та взаємодією користувача; Визначте та розв'яжіть проблеми.

### **2.1 Методи досліджень з використанням айтрекінгу, їх переваги та недоліки.**

Відстеження очей насправді можна розглядати як інструмент для оцінки сенсорних процесів користувача, його спостережень за навколишнім середовищем, а також поведінкових реакцій, що цікавлять, за допомогою спеціальних методів, які фіксують напрямок, тривалість та інші аспекти людського зору. Зрештою, ні для кого не секрет, що спостереження та аналіз графічних взаємодій будуть використовуватися в більшості цих досліджень [23].

Дослідження рухів очей респондентів відносяться до початку 19 століття. Через просте спостереження, час, витрачений на збір і обробку інформації дуже великий. Однак в останні роки процес запису точок для поліпшення погляду і рухів очей став більш автоматизованим. Прикладні спостерігачі довели свою надійність і ефективність для допомоги дослідникам у вивченні процесів уваги, дозволяючи їм отримувати до 80% необхідної інформації.

Спостережні дослідження мають такі переваги:

- висока швидкість тестування в разі моніторингу стаціонарного стану;
- При відстеженні мобільного джерела багато часу витрачається на оптимізацію відповідей;
- збільшення даних, оскільки дані людини та пристрою не проходять через свідомість, що дозволяє перевірити все, що людина може побачити в реальності: це представлено на моніторі Сайтів, до яких можна отримати доступ або показати користувачеві;
  - результати доказів, які дослідники бачать відразу, використовуючи мінімальний метод;
  - залучення невеликої кількості респондентів до опитувань і психометричних тестів;
  - отримання унікальної інформації на етапі тестування створення функції продукту або оцінки нового дизайну.

Онлайн-фокус-групи є одним із найпопулярніших методів якісного дослідження, вони мають максимально широке охоплення різних аудиторій. Розглянемо ключові моменти традиційних та онлайн фокус-груп.

Спільними рисами фокус-груп є:

- можна помацати і навіть вживати той чи інший товар;
- можливе спостереження за невербальною поведінкою учасників групи;
- респонденти частіше використовують змішані ефекти.

З іншого боку, онлайн-фокус-групи мають можливість:

- залучення респондентів із важкодоступних аудиторій;
- дистанційне дослідження;

- розподіл ролей для кожного респондента;
- Отримання чітких відповідей і пряме спілкування;
- тестування інтернет-сервісів у їх природному середовищі;
- швидке отримання результату;
- зниження витрат на дослідження.

Онлайн-фокус-групи не мають географічних обмежень і набагато дешевші, ніж звичайні (офлайн-фокус-групи).

Серед найбільш важливих типів онлайн-фокус-груп можна назвати фокус-групи у формі чатів, а також форуми [24].

Фокус-групи в стилі чату – це фокус-групи в реальному часі, коли учасники реєструються на певному веб-сайті та обговорюють певну тему з модератором. Список запитань складено заздалегідь, щоб не витрачати час на роздрукування з комп'ютера. Під час обговорення можуть виникати додаткові запитання, які відповідають переліку питань. Зазвичай фокус-групи онлайн тривають від одного до трьох днів. Цей метод дослідження дуже корисний, коли потрібно швидко приймати рішення.

Фокус-група як форум – це мережеве дослідження, яке не має нічого схожого на стандартний формат. Тривалість такого типу групи велика, а кількість учасників велика. Учасники реєструються на окремій платформі. Потім учасники відповідають на запитання модератора, обмінюються думками та роблять коментарі. За часом опитування такого типу займають близько трьох-п'яти днів і мають можливість залучити в найкоротші терміни глядачів, які не можуть висидіти на 2 години розмови.

На форумах глядачі можуть увійти в будь-який зручний для них час. Переваги включають те, що ви можете помститися, а також той факт, що тривалість обговорення дозволяє учасникам детально проаналізувати обговорення та відповісти після розгляду своїх відповідей. У результаті відповіді зважуються та розглядаються. Також є можливість актуалізувати запитання та спрямувати обговорення в потрібне для модератора русло.

Використання фокус-груп онлайн можна знайти в таких місцях:

- тестування концепції реклами;
- оцінка іміджу компанії;
- задоволеність клієнтів і співробітників компанії;
- дослідження динаміки ринку;
- сегментація ринку, аналіз окремих сегментів ринку;
- тестування та розробка назв, брендів, торгових марок, логотипів;
- тестування дизайну та функціональності продукту;
- аналіз потреб аудиторії;
- тестування нового продукту або варіанта розробки;
- оцінка політики компанії;
- знання про товар, ставлення до нього;
- використання продукту, аналіз поведінки клієнтів;
- оцінка сайту клієнта, вивчення роботи сайту компанії;
- оцінка упаковки товару.

Серед недоліків онлайн-груп можна відзначити складний набір учасників, оскільки відносна анонімність респондентів негативно позначається на результатах. Звичайно, частково це вирішується через спеціальні онлайн-розділи, де всі учасники фокус-груп зареєстровані в Інтернеті, проте не завжди можна гарантувати максимальну точність експерименту. [25].

Флеш-онлайн-опитування – ще один ефективний метод дослідження поведінки споживачів в Інтернеті, інструмент, який має багато переваг.

Одним із найцікавіших видів прямих онлайн-опитувань є опитування електронною поштою, коли сумнівні речі надсилаються респонденту листом, а респонденту достатньо лише відповісти на лист або натиснути відповідну кнопку внизу ліхтарика. Крім того, респондент може побачити результат через секунду після своєї відповіді.

Основним типом Flash-опитування є Web Flash. У цьому випадку база даних розміщується на веб-сайті, і відвідувач може брати участь у дослідженні в будь-який

момент під час відвідування веб-ресурсу. Результати також доступні через секунду після відповіді на запитання [26].

Ось основні переваги онлайн-опитувань:

1. Більша швидкість. Але на першому місці – швидкість онлайн-голосування. Проект можна розпочати за кілька годин. Відповіді аналізуються миттєво, а результати можна переглядати у відео в реальному часі. Також є можливість швидко відреагувати на ситуацію і внести зміни в опитування на вимогу модератора.

2. Доступ. Прямі онлайн-опитування є найбільш економічно ефективним методом аналізу поведінки. Вартість одного респондента низька, оскільки найдорожча частина – польове дослідження – виключена, опитування не залежить від географії чи місцезнаходження респондента. Причому все обробляється автоматично, і модератор бачить лише кінцевий візуальний результат потрібної йому моделі.

3. Якість результатів опитування. Великий розмір вибірки Flash-Опросы Online дозволяє зменшити похибку результатів. Отримання унікального опитування дає змогу досліджувати навіть найскладніші цілі. Конфіденційність середовища дозволяє отримати чесні відповіді на кожне запитання.

4. Простота створення онлайн-опитувань і керування ними. Модератор може автоматично змінювати запит, завантаження даних і багато інших областей. Все це підвищує корисність і ефективність опитування.

5. Мета відповідей. Конфіденційність опитування та відсутність впливу на опитування з боку модератора дають змогу отримати об'єктивні результати. Крім того, відповідач завжди має можливість розглянути питання, тобто немає тиску з боку арбітра [27].

Серед критики можна повторно ідентифікувати неправильну вибірку та помилки представлення. Будь-хто може взяти участь в опитуванні, навіть не представлені користувачі. Відсутність особистого контакту погіршує ситуацію. Але навіть тут флеш-технології пропонують спеціальні трекери, які можуть видалити ці помилки. Однак переваг, звичайно, набагато більше, ніж недоліків.

Для компаній, які хочуть отримати швидкі результати, цей онлайн-метод дослідження є найкращим.

Інший метод - тестування юзабіліті. Його можна описати як експеримент, який проводиться для визначення того, наскільки добре люди можуть використовувати певний віртуальний об'єкт, наприклад веб-сторінку, інтерфейс користувача чи інструмент, за призначенням, тобто. Тест зручності використання. Він вимірює легкість використання об'єкта. Тестування зручності використання зосереджується на конкретному об'єкті або невеликій групі об'єктів, тоді як дослідження взаємодії людини з комп'ютером зазвичай моделюють універсальні принципи. Юзабіліті-тестування – це метод оцінки зручності використання продукту, який ґрунтується на участі користувачів і тестувальників та узагальненні зроблених ними висновків [28].

Процедура тестування така. Респонденту пропонується спробувати вирішити завдання, створені для тестових пунктів. Під час виконання завдань респондент повідомляє про свої відчуття, думки та спостереження. Мета медіатора — зосередити респондента на діяльності, направити респондента в потрібне русло, але не впливати на нього. Окрім медіаторів, у ПК-тесту беруть участь супервізори, метою яких є моніторинг ситуації, а також моніторинг певних моментів, що сприяють порушенню тестового дозволу. Все фіксується в протоколі за допомогою відео чи аудіотехніки і навіть детального аналізу. Комп'ютерне тестування добре допомагає визначити потенційні проблеми, з якими може зіткнутися реальний клієнт продукту чи послуги. Спостереження за користувачами дозволяє зібрати багато корисних даних для покращення продукту чи послуги. Як правило, отримана інформація служить хорошою основою для генерації нових ідей.

Є ще один спосіб - сканування ПК. Для вирішення заданої проблеми користувачеві пропонується пройти відповідний сценарій її вирішення. Під час виконання відповідач фіксує відхилення від ідеального сценарію. На основі даних респондента розробники вносять необхідні корективи [29].

Переваги тестування ПК включають:



- можливість підтвердити / спростувати заяви про будь-який продукт або послугу;
- можливість порівняти ідеальну ситуацію з реальною;
- мета необхідного вдосконалення за результатами випробування.
- До недоліків скринінгу можна віднести:
- відсутність рекомендацій щодо вирішення проблеми за результатами тестування;
- дослідження є дорогими та тривалими;
- можливість відтворення реальних умов.

Всі ці методи можливо і зручно використовувати для дослідницьких спостережень, це дозволить не тільки отримати більш детальну інформацію, а й прочитати нерозбірливі дії та думки користувачів, які вони не згадали або не надали під час опитування. мають значення, що допоможе вам отримати реальні результати.

Обсерваційні дослідження поділяються на такі види: пересувні, стаціонарні і обсерваційні.

#### 2.1.1 Мобільний айтрекінг

Мобільне спостереження дозволяє поєднати відеозображення з координатами зору людини, яка рухається (за допомогою визначення кута повороту ока). Таким чином можна знайти не тільки зовнішній вигляд зображення, яке бачить користувач, але й напрямок його появи, який є привабливим. Спеціальні окуляри з невеликою відеокамерою і записуючим пристроєм, які кріпляться на респондента, дозволяють зафіксувати всі деталі, які знаходяться в полі його зору.

Для дослідження покупець набирається заздалегідь або безпосередньо в магазині, одягає окуляри і купує, як зазвичай. Вбудована камера знімає все, що бачить респондент [30].

Оскільки інтерв'юер не слідкує за клієнтом під час польових робіт, коротка бесіда проводиться з ним, коли він відвідує магазин в окулярах. Усі отримані записи були проаналізовані за допомогою якісних методів аналізу кодування відео. За результатами дослідження показано поведінкові моделі пошуку та вибору торгових

марок, маршрути пересування покупця по магазину та його відділах. Крім того, дослідник отримує інформацію про те, що відбувається в очах покупця, що він читає або дізнається в торговому залі, що особливо приваблює магазин.

Але такі дослідження можливі не тільки в магазинах, а й в інтернеті. За допомогою тих же окулярів або без них, адже технологія дозволяє використовувати звичайну веб-камеру як дослідницький інструмент.

Технологія таких досліджень мало чим відрізняється від попередніх. У цьому випадку замість відвідування звичайного магазину покупець відвідує веб-сайт магазину для досліджень, блогів або програм.

У випадку з магазином чи блогом ми можемо отримати інформацію про те, наскільки підходить сайт чи блог, на що звернути увагу в першу чергу, чи є зони, які не потрапляють у поле зору. Де людина натискає, що вона дивиться, переглядає інформацію чи вивчає. Після отримання даних ми зможемо провести повний аналіз, на основі якого можна буде вдосконалити продукт, зробити сторінки більш релевантними, розмістити рекламу в більш корисних місцях.

### 2.1.2 Стаціонарний айтрекінг

Під статичним оптичним відстеженням розуміється ситуація, коли зображення об'єкта (або моделі), пристрою та об'єкта знаходяться на відстані 1-3 метрів. Для цього типу тесту використовуються насадки, які легко від'єднуються від пристрою [31].

З іншого боку, протизавага не потребує спостереження, що дуже зручно, і підвищується легкість регулювання. З іншого боку, виникає проблема помітити зображення досліджуваного продукту, його розмір може бути набагато більше розміру монітора, але переваги переважають цей недолік. Практичною перевагою даного типу є можливість тестування багатьох респондентів за короткий час порівняно з повним тестом мобільного рівня джерела за рахунок проектування зображення на екран за допомогою проектора, що не викликає сприйняття невеликих проблем з, а при аналізі отриманих даних достатньо врахувати можливу помилку.

Ці типи досліджень дуже всеосяжні та коштують менше як з точки зору часу, так і грошей. Крім того, через можливість протестувати велику кількість людей за один і той же період цей метод дуже поширений.

### 2.1.3 Моніторний айтрекінг

Моніторити відстеження очей у формі певної діяльності, яку виконує респондент (наприклад, відвідування певних веб-сайтів і пошук там необхідної інформації) за допомогою стаціонарного комп'ютера з інфрачервоною камерою та програмним забезпеченням, яке ідентифікує особливості очей «карта уваги». При цьому респондент, який виконує завдання, знаходиться в місці, наближеному до природи, його рух ніщо не обмежує. Це один із найпопулярніших видів дослідження, оскільки найдешевший і найпростіший у виконанні. Це дає можливість протестувати кількох користувачів однієї служби або декілька служб одночасно за допомогою кількох користувачів і домашніх комп'ютерів.

Для проведення досліджень не обов'язково мати спеціальні окуляри чи інфрачервону камеру. Сучасні технології дозволяють проводити ці дослідження за допомогою камер на ноутбуках або настільних комп'ютерах, і незабаром цей тип дослідження можна буде використовувати на будь-якому смартфоні чи планшеті. та встановлення відповідного програмного забезпечення [32].

## 2.2 Основні показники та критерії оцінки ефективності аналізу графічних інтерфейсів за допомогою айтрекінгу

При тестуванні веб-сторінок необхідно довести релевантність коментарів, визначити найважливіші показники та показати їх зв'язок з критеріями оцінки ефективності Інтернет-комунікації.

Інформаційною базою даної роботи стали теоретичні напрацювання вчених, які проводили дослідження з питань інтеграції графіки та коментарів, тематичних напрямів, результати досліджень, матеріали закордонних конференцій.

Під час дослідження використовувалися такі загальнонаукові методи: аналіз і синтез, системний аналіз та загальний аналіз.

Поки що доведено, що людська свідомість є первинним ефектом свідомості. Завдяки технології, яка дозволяє визначити, як патоген обробляється в мозку, ви можете отримати найточніше уявлення про процеси та пов'язані з ними когнітивні та емоційні здібності.

Оскільки ринок стає все більш конкурентним, нам потрібні точні, прості та інформативні методи аналізу та дослідження, одним з яких є перспективна технологія айтрекінгу. В результаті фахівці намагаються використовувати в своїй роботі цифрові технології. Технологічні компанії застосовують нові технології в боротьбі за частку ринку [33].

Загальна мета цього дослідження — відстежити рухи очей користувача та з'ясувати, на чому він фокусується. Технологія дозволяє переглянути свій підхід до вивчення графічної взаємодії і зосередитися на вивченні людського сприйняття, дослідити свідомість і свідомість людини.

Якщо говорити про дослідження очей як про один із напрямів дослідження, то основним об'єктом його практики є активність мозку у відповідь, його емоційна залученість, увага та увага.

Етапи такого дослідження відрізняються від традиційного дослідження. Різниця полягає в тому, що в обсерваційних дослідженнях підготовка даних і використання спеціального обладнання займає менше часу.

У дослідженні наявність того чи іншого інструменту визначає хід дослідження. Ми вже знаємо, що айтрекер записує рухи очей і відстежує координати погляду. Використовується для вивчення почуттів і інтересу респондентів. Ідеально підходить для вимірювання ефективності веб-сайтів, реклами, фільмів, відеоігор, рекламних матеріалів, дизайну логотипів, дизайну продукту, розміщення продукту та багато іншого. Переглядаючи джерело, ми можемо підібрати відповіді людини, перш ніж вони будуть відфільтровані та представлені вам у викривленому чи об'єктивному вигляді. Цей інструмент дає змогу знати, що люди насправді бачать, а що ні. Камера високої роздільної здатності допомагає аналізувати дрібні рухи, щоб ідентифікувати такі емоції, як здивування, страх, гнів, щастя та інші емоції. Веб-

камера знімає 6 кадрів за секунду, дозволяючи зафіксувати найменші зміни емоцій учасника, який сидить перед комп'ютером [30].

Але такі дослідження передбачають глибокий аналіз даних. Тому необхідно визначити ті показники, які будуть інтегровані в інструмент аналізу, який повністю відповідатиме меті дослідження. Наведемо показники для оцінки результатів дослідження:

- сенсорні сигнали (реакції, пов'язані з вибіркоким сприйняттям об'єктів
- предмети, призначені для візуалізації інформації та акцентування уваги на важливих деталях;
- інтерес (пізнавальна потреба знати, що відбувається, безпосередній інтерес до візуальної інформації);
- запам'ятовування (можливість стимуляції пам'яті);
- емоційна залученість (короткочасна і довготривала активність, що демонструє емоційну реакцію у відповідь на пред'явлення стимулу).

У ході дослідження було вирішено детально проаналізувати можливості та особливості використання такої аналітичної методики, як айтрекінг. Метод заснований на реєстрації руху зіниці ока. Око реєструє фіксацію очей (визначає координати затримки ока в певному місці, тривалість такої фіксації), рух очей і його характер, напрямки очей респондента [34] .

Технологія айтрекінгу є необхідною для компаній, які тестують веб-сайти. Одним з найважливіших показників Інтернет-комунікації є ROI (Return on Investment), який впливає на дохід від реклами, кількість продажів, вартість реклами та розміщення. Розробники сайтів прагнуть удосконалити ці рекомендації і витрачають багато грошей на дизайн сайту, роблять гарні фотографії товарів, створюють унікальні торгові пропозиції, але продажі не збільшуються. Експерти починають шукати проблему та аналізувати її за допомогою методів веб-аналізу та виявляють, що користувачі проводять час на головній сторінці та йдуть. Потім відчувається потреба в таких технологіях, як відстеження очей. Щоб це виправити, вам потрібно відстежувати користувача, точніше, що він дивиться. Для цього була

розроблена технологія спостереження. Це дозволяє контролювати елементи сайту, на які користувач звертає увагу, і не так просто звертає увагу [16].

Для цього використовуються нові чутливі пристрої: пристрої, оснащені камерами та інфрачервоними датчиками для фіксації сприйняття людиною об'єкта, а також є прості опції, такі як веб-камера на пристрої. В результаті замовник дослідження отримує звіт, що складається з двох частин: графічної та аналітичної.

Графічний розділ звіту складається з трьох форматів:

1. Sight Pass - карта напрямку руху очей і точок фіксації погляду респондента. Перегляд шляхів дозволяє досліднику побачити порядок, у якому учасник дивиться на цікаві об'єкти, а також побачити точки фіксації (місця, де учасники дивляться від 100 до 500 мс) і сакади (як рухи). - Області, де респондент здається ледачим. Дороги на карті позначені лініями, точки покращення – кружечками, чим вони більші, тим більше часу респондент витратив на них, зосереджуючись.

2. Теплові карти - теплові карти чутливості, що показують рівень концентрації в тій чи іншій частині ділянки. На карті точки оптимізації кожного респондента перекриваються і утворюють окремі регіони, де червоний колір позначає високий рівень оптимізації, жовтий - середній, зелений - низький, а відсутність нагріву області - відсутність оптимізації у вигляді відповіді - провайдер . га ,

3. Області інтересу (AOI) - регіони інтересу з відсотковим значенням коефіцієнта оптимізації в конкретному регіоні веб-сторінки та середньою тривалістю кожного регіону. Таким чином, клієнт може чітко бачити, що якщо його заклик до дії або повідомлення знаходиться в зоні інтересу лише для 10 відсотків відвідувачів сайту, переміщення елемента в більш цікаве для відвідувачів місце може збільшити ймовірність відповіді на запланований пункт [ 4].

Розділ аналітики містить детальний аналіз поведінки користувачів на сайті, а також експертні поради щодо того, що потрібно змінити на сайті для покращення взаємодії з користувачем та збільшення конверсії. Слід зазначити, що для якісної інтерпретації результатів обсерваційних досліджень потрібен фахівець високого

рівня кваліфікації не тільки в галузі маркетингу та ІТ-технологій, а й у психології процесів. зовнішній вигляд. , пропонується використовувати спеціальні програми для обробки даних, наприклад, IBM Watson Analytics і Tobii Technology, які дозволять легко знаходити зв'язок між різними змінними [35].

Айтрекінг може видавати наступні показники:

- час до першої фіксації (time to first fixation) - час в секундах від того моменту, коли стимул був показаний до моменту, коли він потрапить у зону інтересів респондента;
- кількість фіксацій не в рамках зони, яка досліджується (fixations before) - кількість фіксацій до того, як респондент опиниться у рамках АОІ;
- тривалість фіксації (fixation length) - тривалість фіксації в секундах в межах АОІ;
- кількість фіксацій (fixation count) - кількість фіксацій за весь період перегляду чи за одну секунду в рамках АОІ;
- тривалість спостереження (observation length) - загальний час в секундах, коли респондент перебував у зоні інтересів, починаючи з фіксації всередині зони і закінчуючи фіксацією за її межами;
- кількість спостережень (observation count) - кількість відвідувань однакових АОІ;
- відсоток респондентів, що фіксували свій погляд принаймні один раз на об'єкті дослідження і він входив у їх зону інтересів (participant %);
- швидкість переміщення погляду (path velocity) - тривалість всіх відрізків між послідовними фіксаціями, що розділена на загальний час в секундах [3];
- довжина сакади – відстані між фіксаціями, характеризує зацікавленість респондента у тій чи іншій області інтересів.

Показник вимірюється в пікселях в секунду. Слід зауважити, що значення швидкості, звичайно, пов'язано з розміром монітора, відстанню між ним і очима, частотою айтрекера та ін.

Для наочності, зобразимо деякі аналітичні показники на Рисунку 2.1.

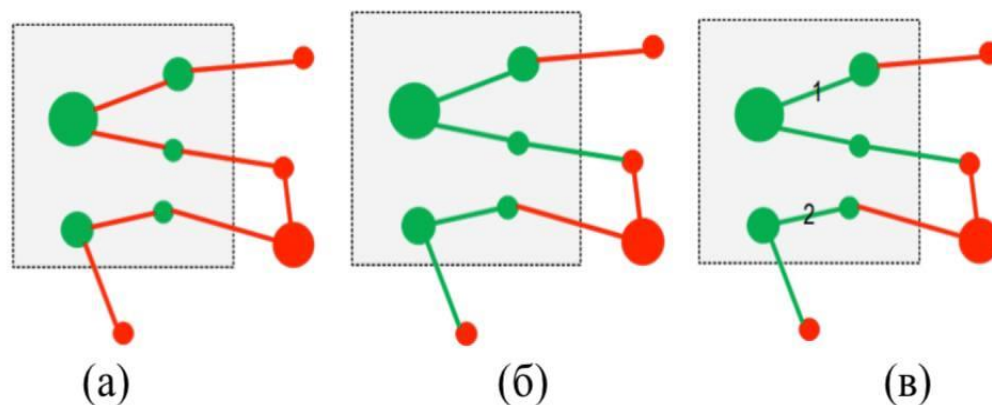


Рисунок 2.1 - Схематичне зображення аналітичних показників: (а) – кількість фіксацій; (б) - тривалість спостереження; (в) - кількість спостережень. Побудовано авторами на основі [3]

Варто зазначити, що порівнюючи критерії оцінки інтернет-спілкування, слід враховувати показники при тестуванні веб-сайтів за допомогою зорових спостережень. Вчені виділяють три критерії, за якими слід оцінювати ефективність комунікації: target (цільова аудиторія), speed (швидкість), value (цінність) [36].

У рамках критеріїв «цінності» розглядаються ключові показники ефективності (KPI). У таблиці спробуємо порівняти спостережувані показники та показники, які можна оцінити в процесі визначення KPI Інтернет-комунікації.

Зазначені в таблиці показники ефективності Інтернет-комунікації не є повним переліком змінних, які впливають на рекомендації та висновки коментарів. За допомогою комплексного звіту про дослідження можна покращити не лише показник глибини перегляду, але й конверсію, яка є одним із найважливіших показників Інтернет-спілкування.

Критерії оцінки ефективності Інтернет-комунікації:

«Цільову аудиторію» можна дослідити за допомогою спостережних досліджень. Так, можливий відбір респондентів на основі заздалегідь сформованих гіпотез і забезпечення їх участі в дослідницькому процесі. Скаутинг допомагає в моніторингу та оцінці якості цільової та цільової аудиторії.

За критерієм «швидкість» цей рівень піддається ретельній оцінці за допомогою методу спостереження, оскільки він здатний виміряти не тільки час і, відповідно, швидкість руху, яку ми бачимо як сайт, але й ідентифікувати місця які



можна покращити, що сприятиме підвищенню показників умов. Ці характеристики наведені в таблиці 2.1

Показники ефективності Інтернет-комунікацій	Показники айтрекінгу	Пояснення
Кількість замовлень за вибраний період	<ul style="list-style-type: none"> <li>- кількість фіксацій</li> <li>- тривалість фіксацій</li> <li>- тривалість спостереження</li> <li>- кількість спостережень</li> <li>- відсоток респондентів, що фіксували погляд</li> </ul>	Отримані результати після айтрекінгу можуть слугувати прогнозом для визначення потенційного обсягу замовлень
Кількість повторних замовлень	<ul style="list-style-type: none"> <li>- кількість спостережень</li> </ul>	Дозволить оцінити наскільки сайт цікавий для цільової аудиторії
Кількість активних дій та кліків	<ul style="list-style-type: none"> <li>- тривалість фіксації</li> <li>- тривалість спостереження</li> <li>- швидкість перегляду</li> <li>- відсоток респондентів</li> </ul>	Отримані показники не можуть оцінити «кількість», але можуть продемонструвати зони, на які з достатньою ймовірністю були б здійсненні переходи
Час перебування на сайті	<ul style="list-style-type: none"> <li>- час до першої фіксації</li> <li>- тривалість фіксації</li> <li>- тривалість спостереження</li> </ul>	Визначення за допомогою айтрекера «слабких місць» може продемонструвати причини відповідного значення показника часу перебування на сайті
Показник відмов	<ul style="list-style-type: none"> <li>- кількість фіксацій не в рамках зони, яка досліджується</li> <li>- час до першої фіксації</li> <li>- швидкість перегляду</li> </ul>	Отримані показники здатні доповнити комплексний показник відмов і виявити зони, які потребують покращення

Таблиця 2.1 - Зіставлення показників ефективності Інтернет-комунікацій та показників айтрекінгу (побудовано авторами на основі [2])

## Висновок до розділу 2

У процесі дослідження всебічно розглядається метод дослідження із застосуванням айтрекінгу та його інструментарій. Під час аналізу було встановлено, що результати досліджень за допомогою айтрекінгу містять як індикатори, так і візуальні карти, що допомагає чітко відповідати на дослідницькі запитання та пропозиції [37].

Важливою частиною дослідження стало порівняння аналітичних показників, отриманих з джерела, відповідно до критеріїв оцінки ефективності маркетингу в Інтернет-комунікації, що доводить важливість використання спостережень при тестуванні веб-сторінок. Подальше дослідження та поглиблений аналіз взаємодії ігор за допомогою айтрекінгу має багато можливостей, а головне, практичне застосування, яке вже на початку свого розвитку дає дивовижні результати.

## Розділ 3 Розробка інструментальних засобів для дослідження графічного інтерфейсу за допомогою айтрекінгу

### 3.1 Зовнішнє проектування

#### 3.1.1 Формалізація задачі

Формалізація задачі на рівні зовнішнього проектування наведена у вигляді діаграми варіантів використання (Рисунок 3.1). Діаграма варіантів використання – це графік спеціального вигляду, який є графічною нотацією для представлення конкретних варіантів використання, акторів, можливо деяких інтерфейсів, і відносин між цими елементами [26].



Рисунок 3.1 – Діаграма варіантів використання

В якості акторів системи відстеження відповідності фокусу зору зі структурою інформації на моніторі виступають декілька суб'єктів: користувач, дослідник, програмне забезпечення для відстеження відповідності фокусу зору зі

структурою інформації на моніторі та відеокамера. Кожен з цих акторів має свою роль у системі [38].

Адміністратор займається наступними задачами:

- налаштування інформаційного контенту за яким спостерігатиме користувач;
- вибір файлів з інформаційним контентом та даними про розміщення елементів інформаційного контенту;
- аналіз результуючих даних системи відстеження відповідності фокусу зору зі структурою інформації на моніторі, задля дослідження поведінки користувача під час роботи з інформаційного контентом на моніторі.

Основним завданням користувача є спостереження за позицією калібрувальної точки під час процесу калібрування та під час розрахунку похибки відстеження координат фокусу зору, а також робота з інформаційним контентом, зображеним на моніторі [39].

Відеокамера займається безпосередньо записом зображення користувача в прямій трансляції.

Основним та найважливішим актором системи є програмне забезпечення для відстеження відповідності фокусу зору зі структурою інформації на моніторі.

Програмне забезпечення займається наступними задачами:

- отримання кадрів з відеокамери під час запису користувача при роботі з ПК;
- обробка отриманих відеокadrів;
- розпізнавання рис обличчя на кадрі;
- знаходження антропометричних точок розташування очей на обличчі;
- знаходження координат центру ока;
- знаходження координат центру зіниці ока;
- проведення процесу калібрування;
- відстеження процесу блимання очей;
- пошук залежності координати зіниці ока від позиції калібрувальної точки;
- підрахунок координати фокусу зору;

- оцінка похибки знаходження координати фокусу зору;
- пошук відповідності фокусу зору зі структурою інформації на моніторі.

### 3.1.2 Базова архітектура системи

У процесі проектування було виконано декомпозицію модулів системи відстеження відповідності фокусу зору зі структурою інформації на моніторі на три рівні згідно з моделлю «Модель-вигляд-контролер» (MVC) [40], результати якої наведені на Рисунку 3.2.

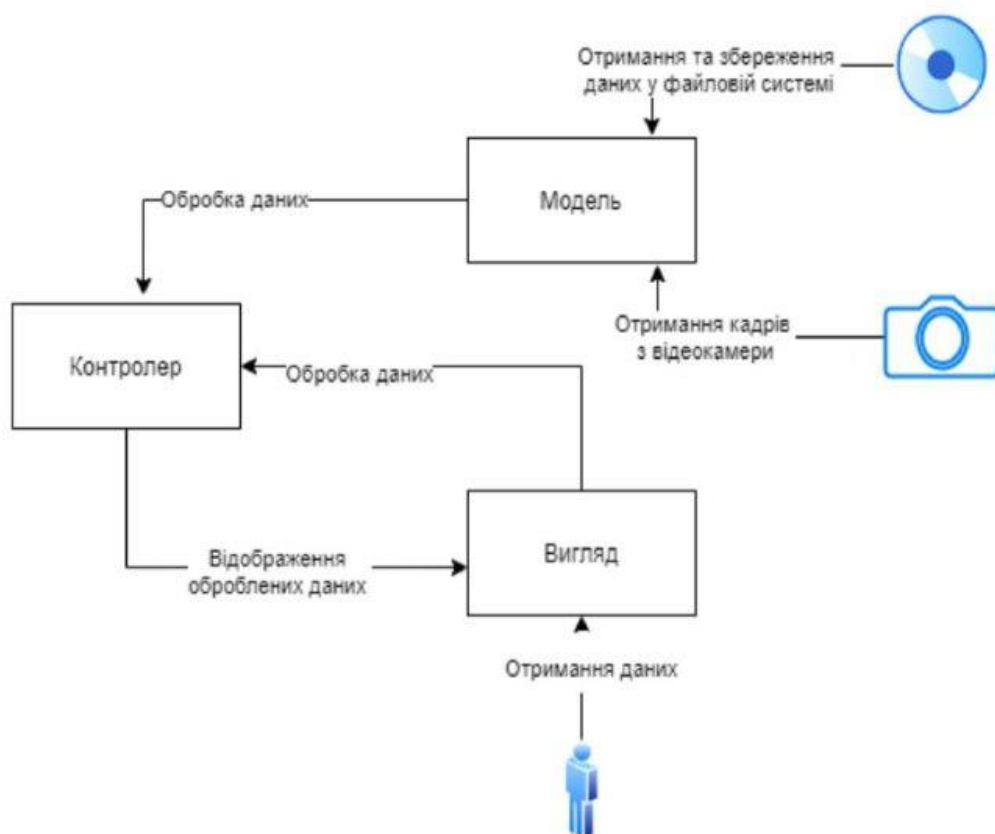


Рисунок 3.2 – Модель «Model-View-Controller» системи відстеження відповідності фокусу зору зі структурою інформації на моніторі

Основна мета застосування цієї концепції полягає в відділенні бізнес-логік (моделі) від її візуалізації (вигляду). За рахунок такого поділу підвищується можливість повторного використання коду.

У рамках архітектурного шаблону MVC програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами:

- модель (Model) – відповідає за збереження та отримання даних з файлової системи та відеокамери. За допомогою цього компоненту система отримує кадри з відеокамери, зберігає результат калібрування у файл, зчитує дані за файлу для відображення інформаційного контенту, тощо;

- вигляд (View) – відповідає за представлення даних користувачеві (наприклад, відображення процесу калібрування, інформаційного контенту, тощо). Компонент представлений у вигляді графічного інтерфейсу;

- контролер (Controller) – відповідає за бізнес-логіку. Контролер виконує основну частину роботи системи відстеження відповідності фокусу зору зі структурою інформації на моніторі. Він отримує дані від компоненту Model, обробляє їх та відправляє у компонент View для відображення користувачеві. Також контролер отримує сигнали у вигляді реакції на дії користувача (натискання кнопки, введення даних у текстове поле) та обробляє їх.

Для реалізації системи відстеження відповідності фокусу зору зі структурою інформації на моніторі обрана об'єктно-орієнтована та функціональна парадигми програмування [30].

## 3.2 Внутрішнє проектування

### 3.2.1 Технологічна платформа

Для розробки системи відстеження відповідності фокусу зору зі структурою інформації на моніторі було обрано середовище розробки React. Для відстеження фокусу зору обрана бібліотека комп'ютерного зору WebGazer.js. Бібліотека WebGazer.js є найпопулярнішою і розвиненою в області обробки зображень і комп'ютерного зору, має базові методи обробки зображення, методи геометричного перетворення, перетворення колірних просторів, машинного навчання, виявлення об'єктів на зображенні. Вихідний код бібліотеки написаний на JavaScript. Зберігання даних було здійснено за допомогою Firebase.

React - відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Він спрощує створення інтерактивних інтерфейсів, потрібно лише описати, як різні частини інтерфейсу виглядають у кожному стані вашого додатку і React ефективно оновить та відрендерить лише потрібні компоненти, коли дані зміняться. Декларативні інтерфейси роблять код більш передбачуваним і його набагато легше налагоджувати [41].

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Він обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків.

React має ряд особливостей:

- Одностороння передача даних. Властивості передаються в рендерер компоненту, як властивості html тегу. Компонент не може напряму змінювати властивості, що йому передані, але може їх змінювати через callback функції. Такий механізм називають «властивості донизу, події нагору»;
- Віртуальний DOM. Оскільки логіка компонентів написана на JavaScript, замість шаблонів, ви з легкістю можете передавати складні дані у вашому додатку і зберігати стан окремо від DOM. React підтримує віртуальний DOM, а не покладається виключно на DOM браузера. Це дозволяє бібліотеці визначити, які частини DOM змінилися, порівняно (diff) зі збереженою версією віртуального DOM, і таким чином визначити, як найефективніше оновити DOM браузера. Таким чином програміст працює зі сторінкою, вважаючи що вона оновлюється вся, але бібліотека самостійно вирішує які компоненти сторінки треба оновити;

- JSX. Компоненти React зазвичай написані на JSX. Код написаний на JSX компілюється у виклики методів бібліотеки React. Розробники можуть так само писати на чистому JavaScript;
- React використовують не лише для рендерингу HTML в браузері;
- Вкладені елементи. Кілька елементів на одному рівні повинні бути загорнутими в один елемент контейнера, наприклад елемент `<div>`, або повернутий як масив;
- Атрибути. JSX надає ряд атрибутів елементів, призначених для відображення тих, що надаються у форматі HTML. Користувацькі атрибути також можуть бути передані компоненту. Всі атрибути будуть отримані компонентом як реквізит.

Методи життєвого циклу — це різні методи, які вбудовуються за допомогою ReactJS. Вони дозволяють розробнику обробляти дані в різних точках життєвого циклу програми React. Наприклад:

- `shouldComponentUpdate` — це метод життєвого циклу, який каже Javascript оновити компонент, використовуючи логічні змінні.
- `componentWillMount` — це метод життєвого циклу, який каже Javascript налаштувати певні дані перед монтуванням компонентів (вставлення у віртуальний DOM).
- `componentDidMount` — це метод життєвого циклу, подібний до компонента `WillMount`, за винятком того, що він працює після методу `render`, і може використовуватися для додавання JSON-даних, а також для визначення властивостей та станів.
- `render` є найважливішим методом життєвого циклу, необхідним у будь-якому компоненті. Метод `render` — це те, що з'єднується з JSX і відображати власний JSX.

React також може рендеритись на сервері, використовуючи Node, і приводити в дію мобільні додатки, які використовують React Native. Принципи роботи React Native в основному такі ж, як ReactJS, за винятком того, що він не маніпулює DOM через VirtualDom. Він працює у фоновому процесі (який



інтерпретує Javascript код написаний розробниками) безпосередньо на кінцевому пристрої і спілкується з нативною платформою. React Native взагалі не покладається на HTML, все написано на Javascript і залежить від нативних SDK [42].

React не намагається надати повну «схему додатків». Він безпосередньо спрямований на побудову користувацьких інтерфейсів, і тому не включає в себе безліч інструментів, які деякі розробники вважають необхідними для створення програми. Це дозволяє вибрати будь-які бібліотеки, які розробник вважає за краще виконувати, щоб виконати певних завдань, таких як здійснення доступу до мережі або локальне зберігання даних.

WebGazer.js — це бібліотека відстеження погляду, яка використовує звичайні веб-камери для визначення розташування відвідувачів веб-сторінки на сторінці в режимі реального часу. Модель відстеження погляду, що міститься в ньому, автоматично калібрується, спостерігаючи, як відвідувачі взаємодіють з веб-сторінкою, і тренує відображення між особливостями ока і положенням на екрані. WebGazer.js повністю написаний на JavaScript, і за допомогою всього декількох рядків коду його можна інтегрувати в будь-який веб-сайт, який хоче краще зрозуміти своїх відвідувачів і змінити їх досвід користувача. Бібліотека повністю працює в клієнтському браузері, тому відео не потрібно надсилати на сервер, а для доступу до веб-камери потрібна згода користувача [38].

Відстеження погляду здійснюється з опорою на побудовану модель обличчя у поєднанні з показниками переміщення курсору та клацання миші. Передбачається, що користувач дивиться місце бажаного чи виконаного натискання. Калібрування моніторингу триває близько трьох клацань миші, після чого система здатна локалізувати погляд у режимі реального часу з похибкою 100–200 пікселів.

В основі WebGazer.js лежать модулі відстеження та регресії. Модуль трекара управляє тим, як виявляються очі, а модуль регресії визначає, як вивчається регресійна модель і як робляться прогнози на основі пов'язок на очах, витягнутих із модуля трекара. Для WebGazer.js потрібна обмежувальна рамка, яка включає пікселі відеопотоку з веб-камери, що відповідають виявленим очам користувача. В даний час ми включаємо одну зовнішню бібліотеку для виявлення обличчя та очей.

```
webgazer.setTracker("TFFacemesh"); //set a tracker module
```

```
webgazer.addTrackerModule("newTracker", NewTrackerConstructor); //add a new  
tracker module
```

Для того щоб можна було за необхідності призупинити збір даних та прогнози WebGazer.js з міркувань продуктивності було застосовано:

```
webgazer.pause(); //WebGazer.js is now paused, no data will be collected and the  
gaze callback will not be executed
```

```
webgazer.resume(); //data collection resumes, gaze callback will be called again
```

Firebase була застосована для зберігання даних. Це платформи розробки мобільних та веб застосунків.

Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js. REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері [42].

Firebase Storage забезпечує надійне завантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firebase підтримується Google Cloud Storage [43].

Firebase Hosting являється статичним та динамічним вебхостингом. Він підтримує хостинг статичних файлів, таких як CSS, HTML, JavaScript та інші файли, а також динамічну підтримку Node.js через Cloud Functions [Архівовано 21 квітня 2020 у Wayback Machine.]. Служба передає файли через мережу доставки контенту (CDN) за допомогою протоколу HTTPS та шифрування SSL. Firebase підтримує Fastly, CDN, щоб забезпечити підтримку CDN Firebase Hosting. Компанія стверджує,

що хостинг Firebase виріс із запитів клієнтів; розробники використовували Firebase для своєї бази даних в режимі реального часу, але вони потребували місця для розміщення їхнього вмісту [44].

Також було розроблено оновлення тестів та виправлення помилок. Додаток є повністю функціональним та виконує усі поставлені перед початком розробки задачі.

### 3.2.2 Використані принципи проектування

Для проектування внутрішньої структури системи відстеження відповідності фокусу зору зі структурою інформації на моніторі використовувалися такі принципи об'єктно-орієнтованого проектування:

- принцип інкапсуляції: прикладом є компоненти додатка, стан яких може бути змінений тільки за допомогою пропсів;
- принцип успадкування: прикладом успадкування властивостей користувацького інтерфесу, наприклад "кнопка", який розширює свої базові властивості при доданні властивості "type" [45].

### 3.2.3 Розробка інтерфейсу користувача

Програмне забезпечення відстеження відповідності фокусу зору зі структурою інформації на моніторі має назву Eye Tracking. Інтерфейс користувача програмного забезпечення було поділено на декілька вікон графічного інтерфейсу, задля покращення розуміння його використання. «Головне вікно» – вікно, яке з'являється після запуску програми Рисунок 3.3. та Рисунок 3.4.

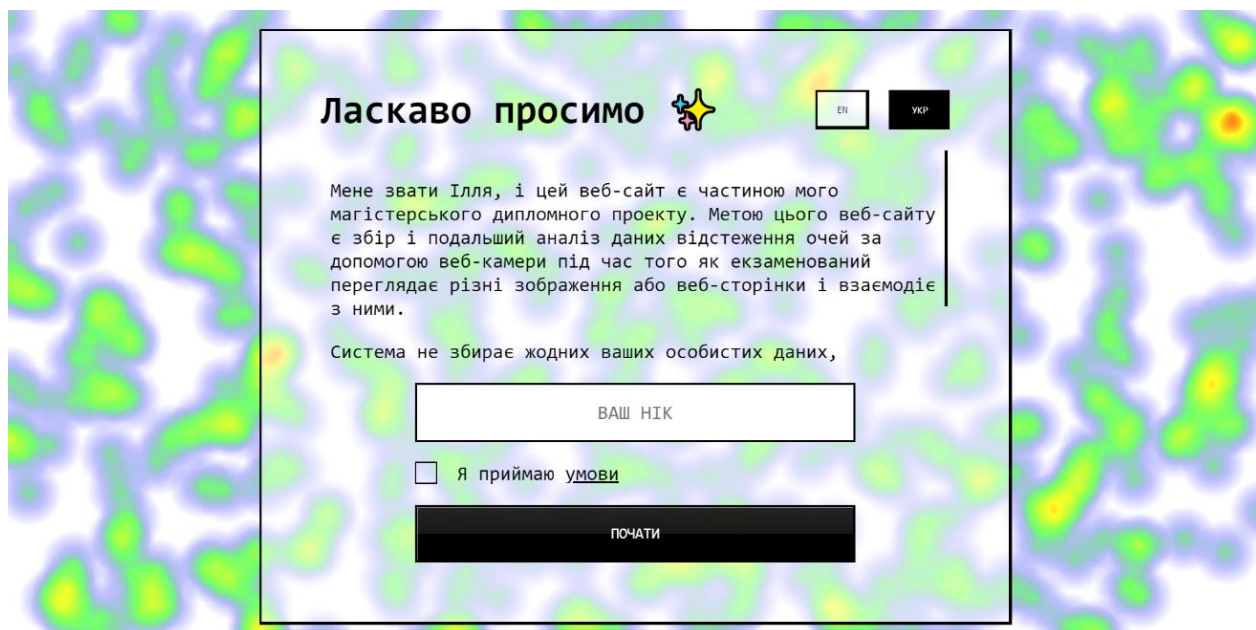


Рисунок 3.3 Головне вікно додатку

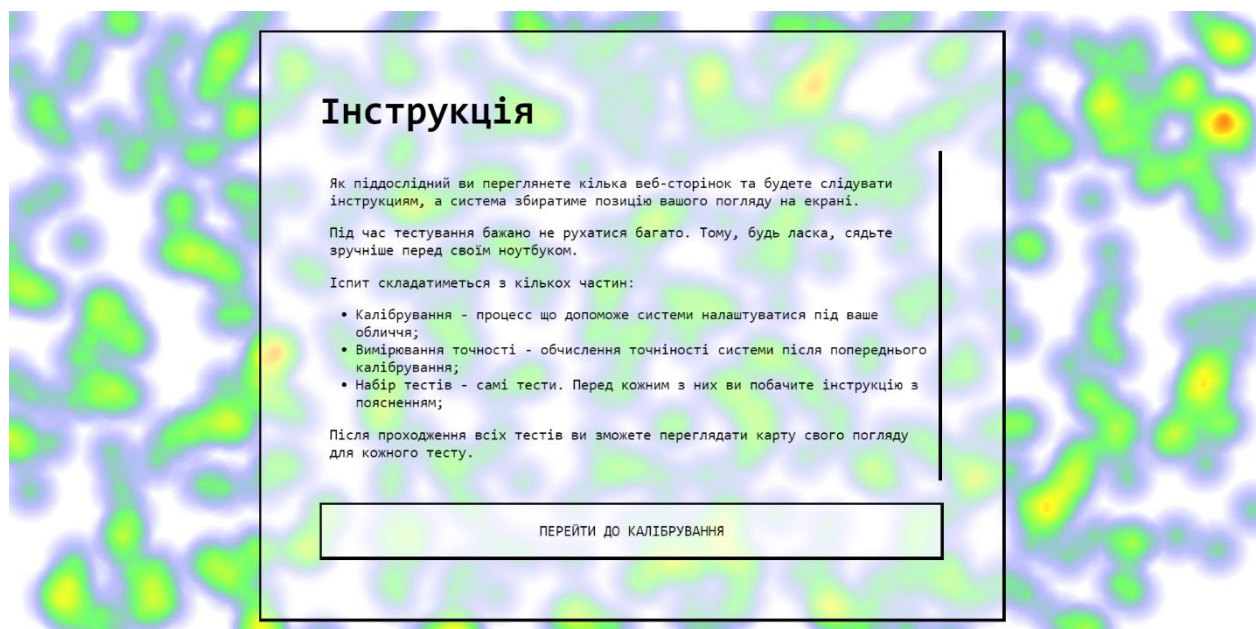


Рисунок 3.4 Інструкція перед початком роботи з додатком

Наступним і дуже важливим етапом є калібрування. Воно забезпечить там точність тестів та досліджень. Також у вікні ми бачимо інструкцію у якій пояснюється що додаток буде використовувати вашу камеру. Також учасник повинен розмістити камеру так, щоб його обличчя могло поміститися у зелений квадрат прев'ю камери, це забезпечить точність калібрування. Вікно «Калібрування» (Рисунок 3.5) дозволяє налаштовувати та проводити процес калібрування. Також прокрутивши до низу ми бачимо інструкцію щодо калібрування, окремо знизу також є кнопка «почати калібрування». Процес

калібрування зазначений на Рисунку 3.6. після калібрування ми отримуємо міру точності яка повинна становити не менше 70%, а також бачимо дві кнопки «перекалібрувати» на випадок низької точності та «перейти до тестів» Рисунок 3.7.



Рисунок 3.5 Вікно початку калібрування

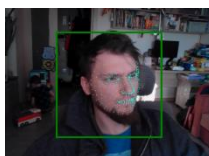


Рисунок 3.6 Вікно процесу калібрування

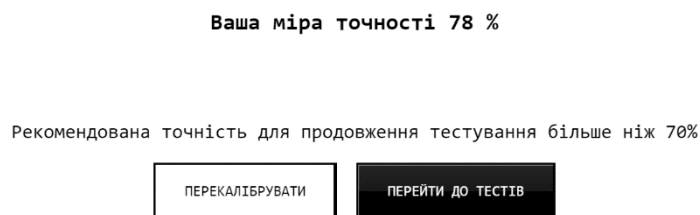


Рисунок 3.7 Вікно міри точності

Далі ми побачимо два вікно з тестами які будуть іти одне після другого (Рисунок 3.8), перед кожним тестом ми маємо інструкцію щодо його проходження. Після тестів з'явиться вікно з можливістю або отримати результати тестів (рис. 3.9), що являють собою теплові карти (Рисунок 3.10, 3.11), або завершити роботу без результатів.

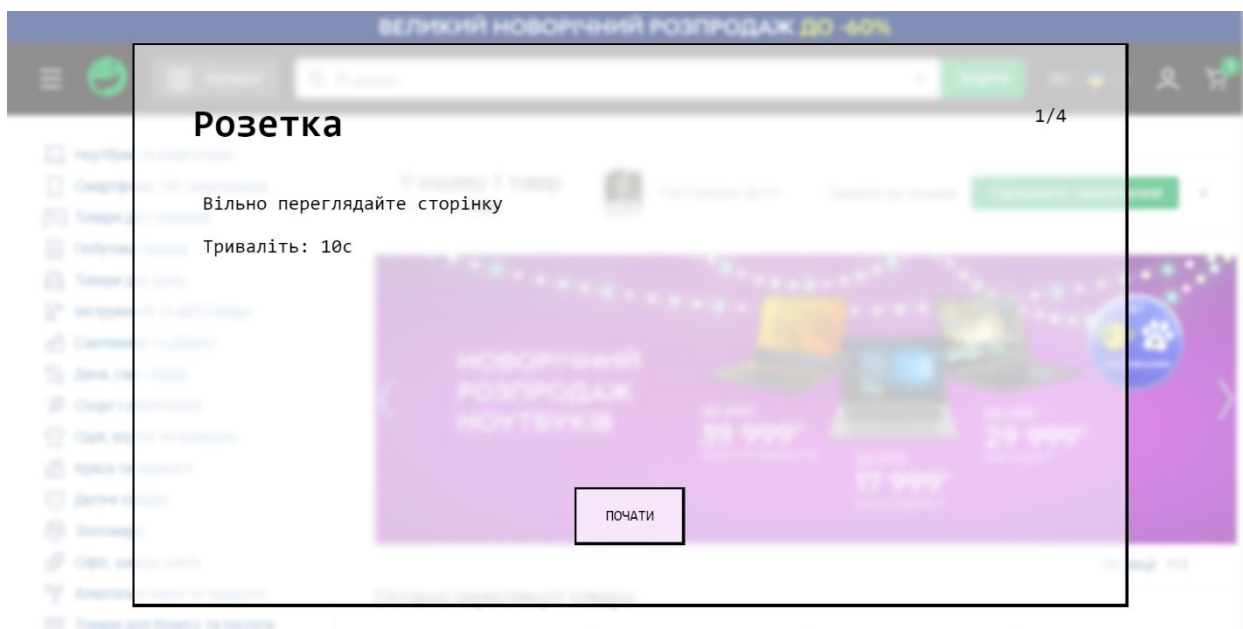


Рисунок 3.8 Вікно тесту.



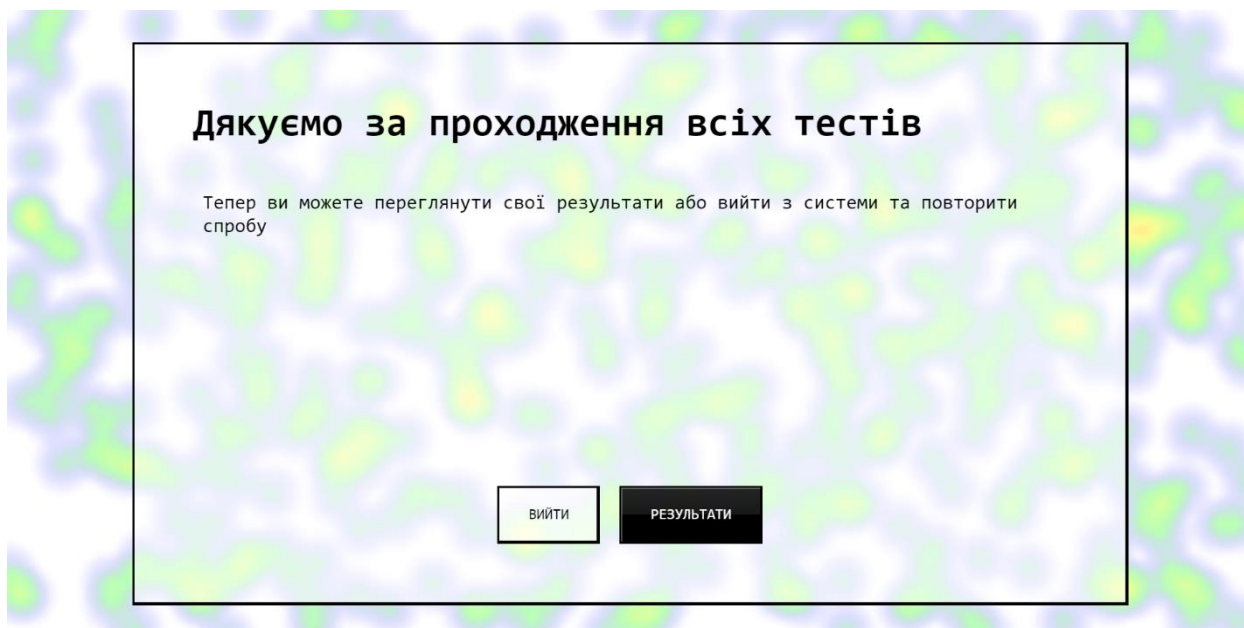


Рисунок 3.9 Вікно переходу до результатів або завершення роботи.

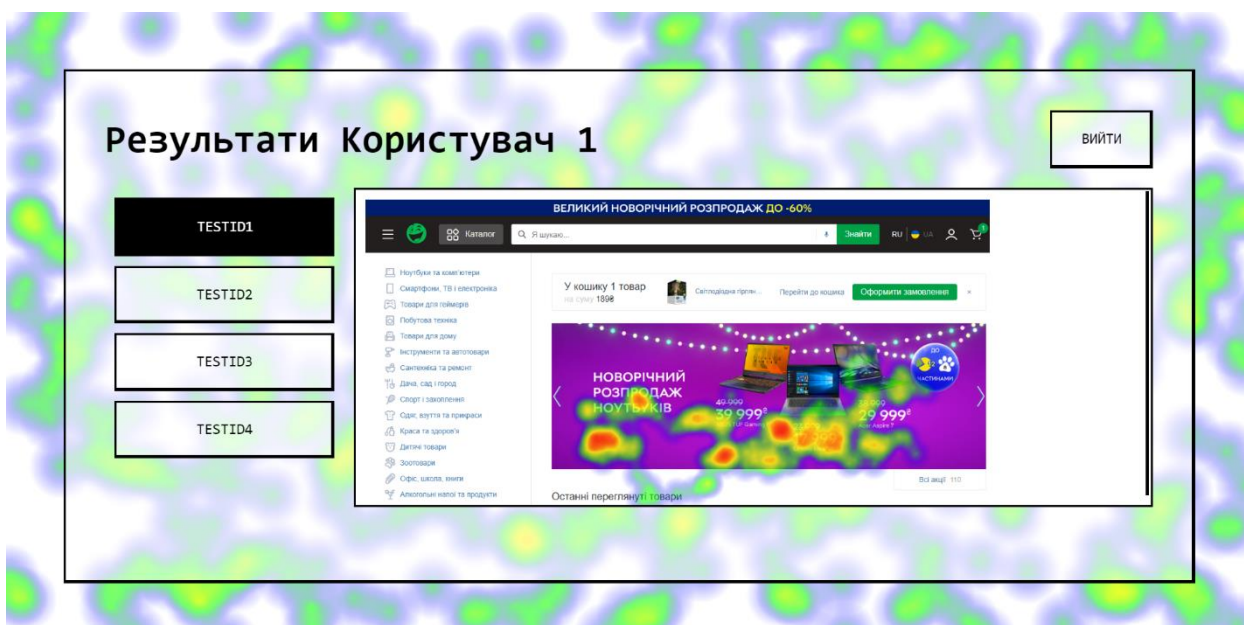


Рисунок 3.10 Результати з тепловими картами досліджуваних інтерфейсів.

### 3.3 Тестування та налагоджування програми

Тестування розробки програмного забезпечення є завершальним етапом розробки веб-сайту та програмного забезпечення. Тестування – це процес оцінки того, чи відповідає готова програма вимогам. Існує два види тестування – ручне та автоматизоване. Ручне тестування виконується людьми, а автоматизація виконує перевірку за допомогою програмного коду. Тестування веб-додатків перевіряє його сторінки різними способами: перевірка продуктивності сторінки в різних браузерах,

перевірка якості сторінок, пошук помилок коду, тестування швидкості сторінки тощо. [46] Вибране. Вибір інструменту є однією з найбільших проблем, яку необхідно вирішити перед переходом до автоматизації. Необхідно визначити вимоги, щоб з'ясувати різні пристрої та їх можливості. Метою роботи є дослідження та порівняння автоматизованого випробувального обладнання, а також написання програмного коду для його практичного застосування. Основні етапи автоматизованого тестування включають: визначення вимог до автоматизованої системи функціонального тестування; автоматичний вибір інструменту; Виконуються автоматизовані тести; Створіть остаточний документ із результатами тестування, включаючи виявлені дефекти, відхилення від стандартів і пропозиції щодо вдосконалення системи. У роботі були розглянуті існуючі засоби автоматизованого тестування програмного забезпечення, зокрема: Ranorex Studio, Selenium WebDriver, QTP (MicroFocus UFT), Logical Functional Testing. Розробка додатків може здійснюватися за допомогою автоматизованих інструментів або за допомогою інструментів керування тестуванням, які будуть називатися автоматизованими інструментами [47]. Для порівняння інструментів розглядалися такі критерії: екологічна підтримка; простота використання; тестування бази даних; розпізнавання об'єктів; тест зображення; перевірка помилок; викладка товарів; використовується письмова мова; підтримка різних типів тестів, у тому числі функціональних, тестових, мобільних; Підтримка кількох тестових сайтів; Спрощення налагодження автоматизованих програмних сценаріїв; Вартість використання обраного обладнання. При аналізі процесу за цими критеріями порівнюються переваги та недоліки програм. Кожен з них має свій предмет перевірки. Правильний вибір інструменту для автоматизованого процесу тестування є важливим фактором успішної автоматизації, тому від його вибору безпосередньо залежить швидкість і якість тестування програмного забезпечення [40].

### Висновок до розділу 3

Конструкція приладу була розроблена для паперового дослідження теоретичного спрямування з інформаційною структурою наукового керівника. Під



час проектування структури системи існує можливість розширення системи шляхом її поділу на гнучкі та зв'язані частини. Це дозволить повторно використовувати та розширювати функціональні можливості розроблених модулів. У цьому розділі описано дизайн інтерфейсу користувача. Досвід користувача використовується для покращення інтерфейсу; Весь інтерфейс системи умовно розділений на сторінки, завдяки яким користувач може легко знаходити та змінювати потрібні йому параметри та проводити дослідження.

## **Розділ 4 Дослідження відповідності фокусу зору із структурою інформації на моніторі**

Під час дослідження відстеження відповідності фокусу зору зі структурою інформації на моніторі було проведено декілька експериментів. Для проведення експериментів використовувалось розроблене ПЗ Eye Tracking. З апаратного середовища використовувалися наступні прилади:

- ноутбук;
- відеокамера з роздільною здатністю 1280x720 (в даному випадку використовувалася відеокамера, вбудована в ноутбук);

Кожний експеримент проводився у рівномірно освітленому приміщенні, з використанням однієї або двох камер. Користувач знаходився навпроти ноутбуку на відстані 50 см від камери. Місце розташування камери було змінним.

Карта поглядів показує фіксацію респондентів, зберігаючи інформацію про тривалість і послідовність фіксацій. Основний недолік карти поглядів - швидке засмічення. Карту поглядів можна використовувати для коротких проміжків часу та для невеликої кількості респондентів.

Теплова карта дозволяє підсумовувати велику кількість даних. На ній втрачається інформація про тривалість та послідовність окремих фіксацій. Теплові карти можуть будуватися за кількістю фіксацій або їх тривалості.

Переваги використання eye-tracker:

- Адекватність: тестування мало відрізняється від нормального перегляду
- Унікальність: отримання інформації, яка недоступна іншими способами
- Об'єктивність: немає необхідності покладатися на думку експертів
- Наочність: можливість подивитися на свій сайт чужими очима.
- Загальна оцінка дизайну: що відвідувачі помічають на сторінці, які тексти читають, в якій послідовності і т.п.
- Де шукають інформацію: де на сторінці відвідувачі намагаються знайти потрібну інформацію Підбір місця для посилань: де відвідувачі очікують побачити потрібне посилання.

Наші дослідження проведені з урахуванням тривалості абсолютної фіксації.

#### 4.1 Дослідження на основі рекламного банеру на веб-сторінці

На сайтах часто розміщують рекламу, та саме вона є одним з важливих елементів дослідження веб-сторінок з використанням айтрекінгу.

У дослідженні приймали участь декілька користувачів що проходили тестування окремо один від одного з використанням ноутбуку та веб-камери. Дослідження проводилось з використанням рекламного банеру на сторонньому сайті Рисунок 4.1. Результат дослідження рекламного банеру зазначений на рисунку 4.1.



Рисунок 4.1 Досліджуваний рекламний банер

На перший погляд здається що банер досить вдалий, адже одразу потрапляють у поле зору потрібні для реклами елементи. Результати дослідження тільки підтверджують цю здогадку.

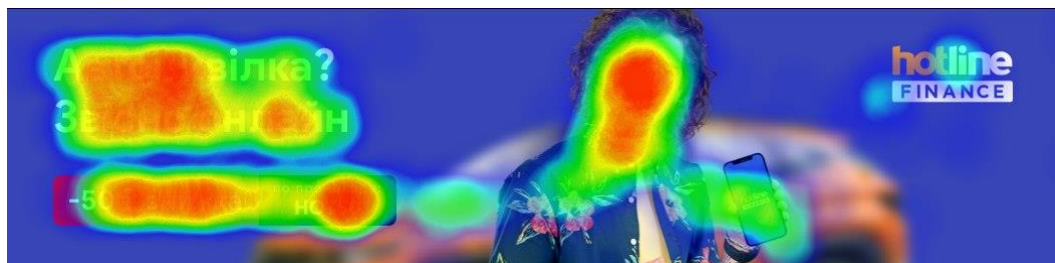


Рисунок 4.2 Теплова карта рекламного банеру

Результат дуже однозначний. Завдяки такому дослідженню ми можемо бачити що найбільше привертає увагу користувачів. Аналізуючи даний банер ми бачимо що не вся увага прикута до обличчя і користувач бачить об'єкт реклами. Такий банер можна вважати досить вдалим, він не потребує правок.

#### 4.2 Дослідження абсолютної тривалості погляду на веб-сторінці 1

Візуальну фіксацію можна умовно визначити як відносно стаціонарне положення очей, сфокусованих на певному місці подразника. Фіксації — це важливі

події, які дають зрозуміти людське пізнання оскільки під час кожної фіксації ми витягуємо візуальну інформацію, яку обробляємо [32].

Дослідження було проведено на декількох веб-сторінках. На першій з них ми бачимо той самий банер, але уже в контексті власного сайту, Рисунок 4.3.

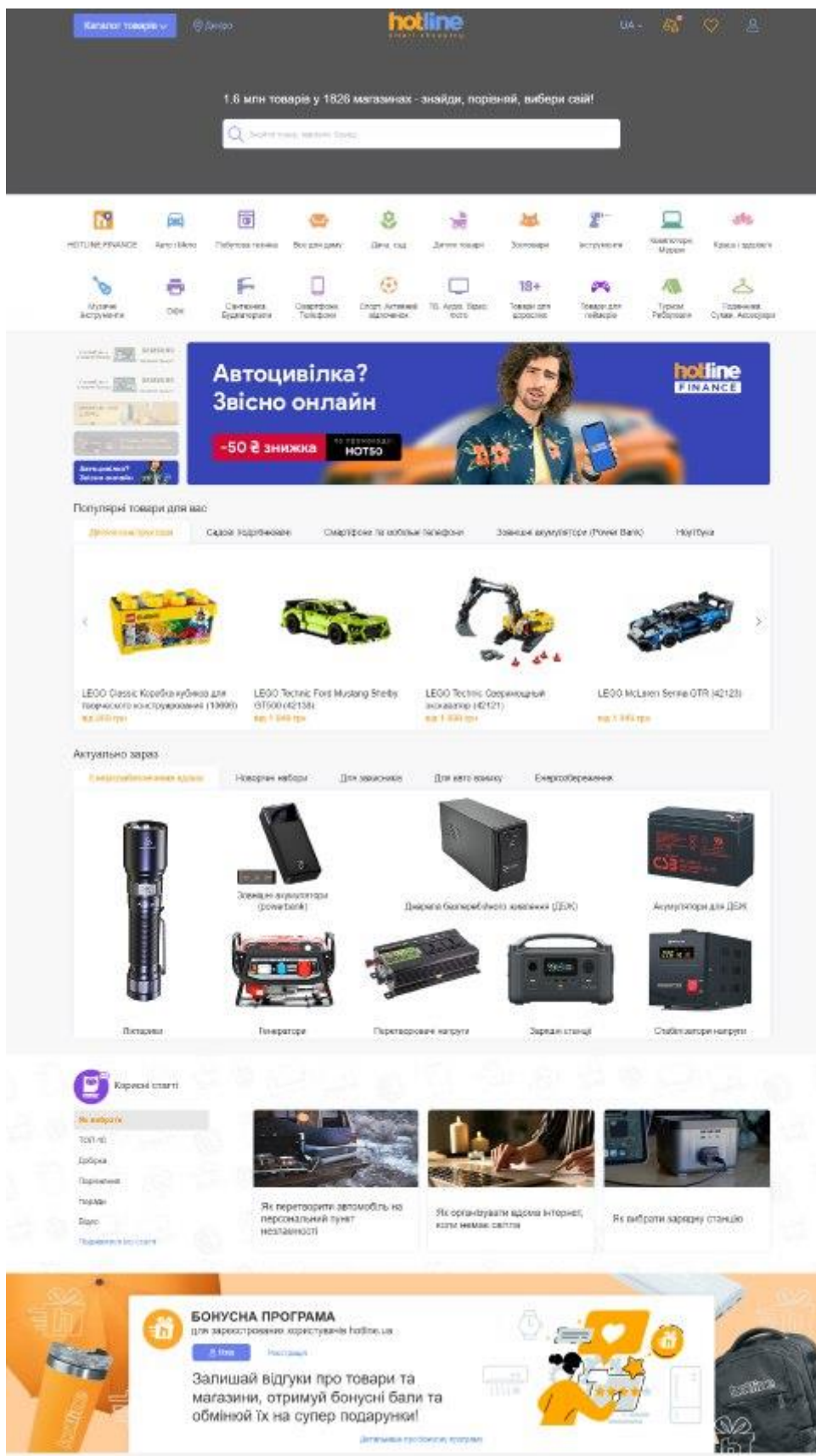


Рисунок 4.3 Досліджувана веб-сторінка

На цей раз ми вже бачимо безліч товарів та структуру сайту, що неминуче нас відволікає від самого банеру, але тут він не грає важливої ролі, тому не повинен

прикути до себе всю увагу користувачів. Визначити що більш за все привертає увагу користувачів і на чому вони затримують всій погляд ми можемо з допомогою теплових карт. Результат аналізу веб-сторінки зазначений на Рисунку 4.4



Рисунок 4.4 Результат аналізу веб-сторінки

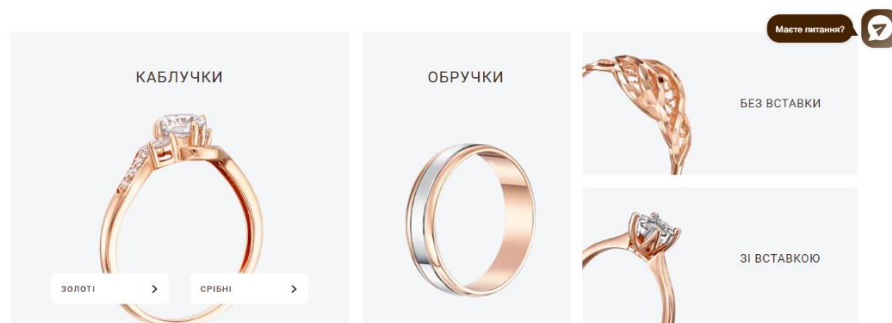
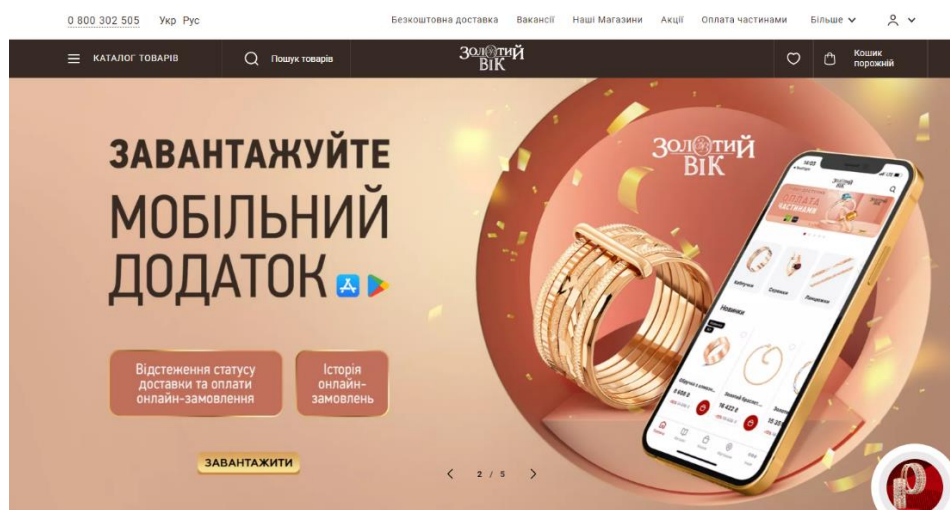
На даній карті ми бачимо що найбільша концентрація була на товарах сайту але дуже незначна. Також бачимо що банер майже не привернув уваги, найменше уваги тримала назва самого сайту а також меню та у нижній частині.

#### 4.3 Дослідження абсолютної тривалості погляду на веб сторінці 2

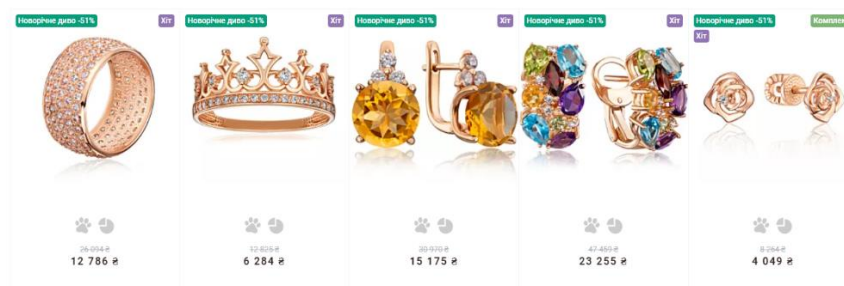
Для експерименту була взята сторінка ювелірного магазину, Рисунок 4.5.

Теплова карта абсолютної тривалості погляду (Рисунок 4.6) показує накопичений час приведений на різних областях сторінки учасниками. Кожна зроблена фіксація кожного учасника додає яскравість на кольоровій карті, яка пропорційна її тривалості [13].

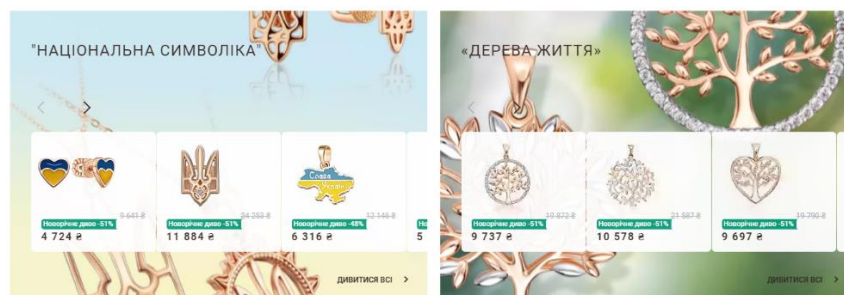




#### ТОП ПРОДАЖІВ



#### НОВІ КОЛЕКЦІЇ



#### КАТЕГОРІЇ



Рисунок 4.5 Досліджувана веб-сторінка



Оскільки тривалість фіксації є показником когнітивної обробки [24], теплова карта не тільки показує які області були присутні, але також являє собою рівень когнітивної обробки.

Подібно до теплових карт підрахунку фіксації, абсолютна тривалість погляду теплові карти можуть бути упередженими щодо людей, які витрачали більше часу на перегляд елементів веб-сторінки.

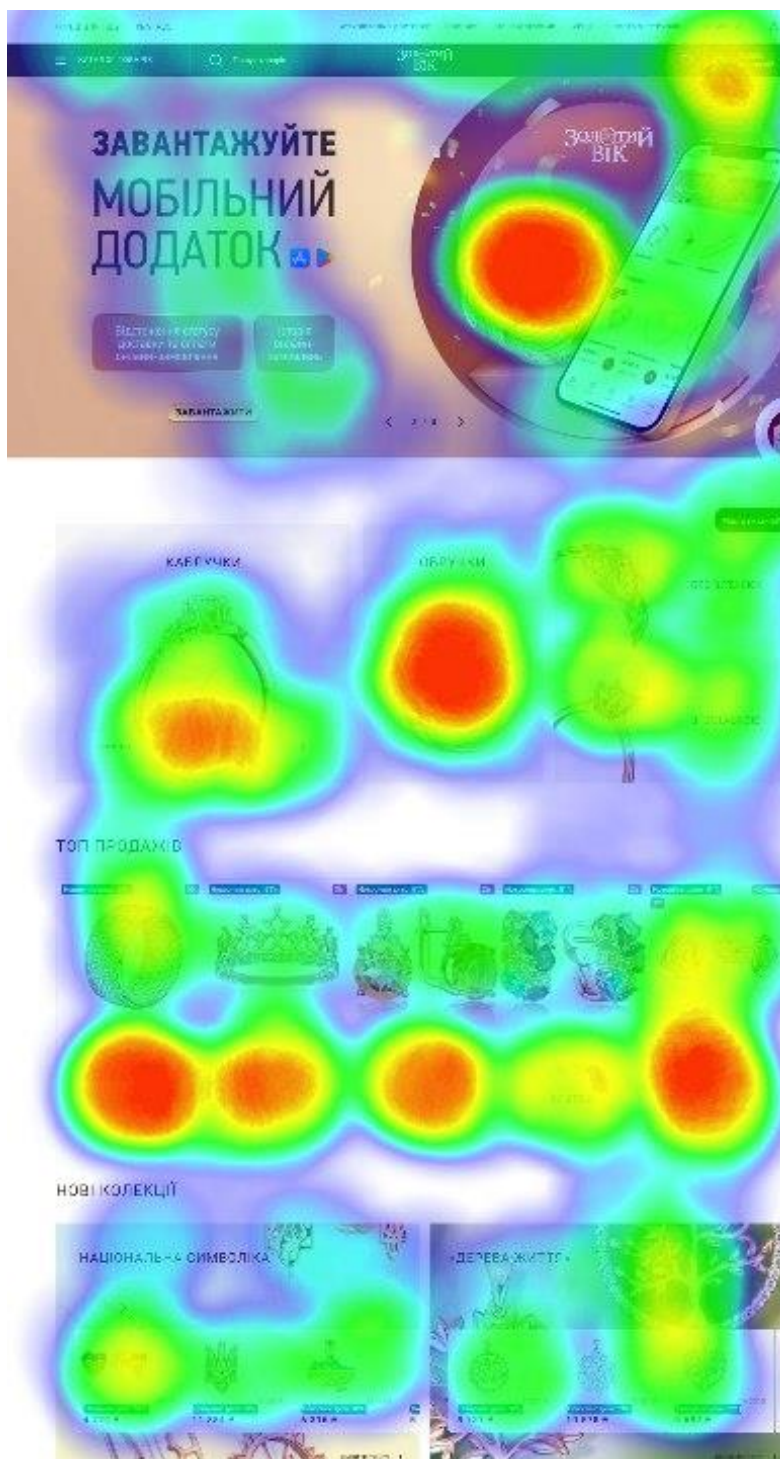


Рисунок 4.6 Теплова карта абсолютної тривалості погляду

Аналізуючи теплову карту абсолютної тривалості погляду ми можемо бачити що як і в попередньому дослідженні більше уваги користувачі приділяли товарам на сайті та інформації на початку сторінки. А ось кінці затримувались вже менше, що свідчить про нецікавість до викладеної інформації. Також як і в попередньому випадку більше уваги прикуто до зображень аніж до інформації.

#### 4.4 Дослідження абсолютної тривалості погляду на веб сторінці 3

Дослідження проводились на базі сайту з продажу техніки, Рисунок 4.7.

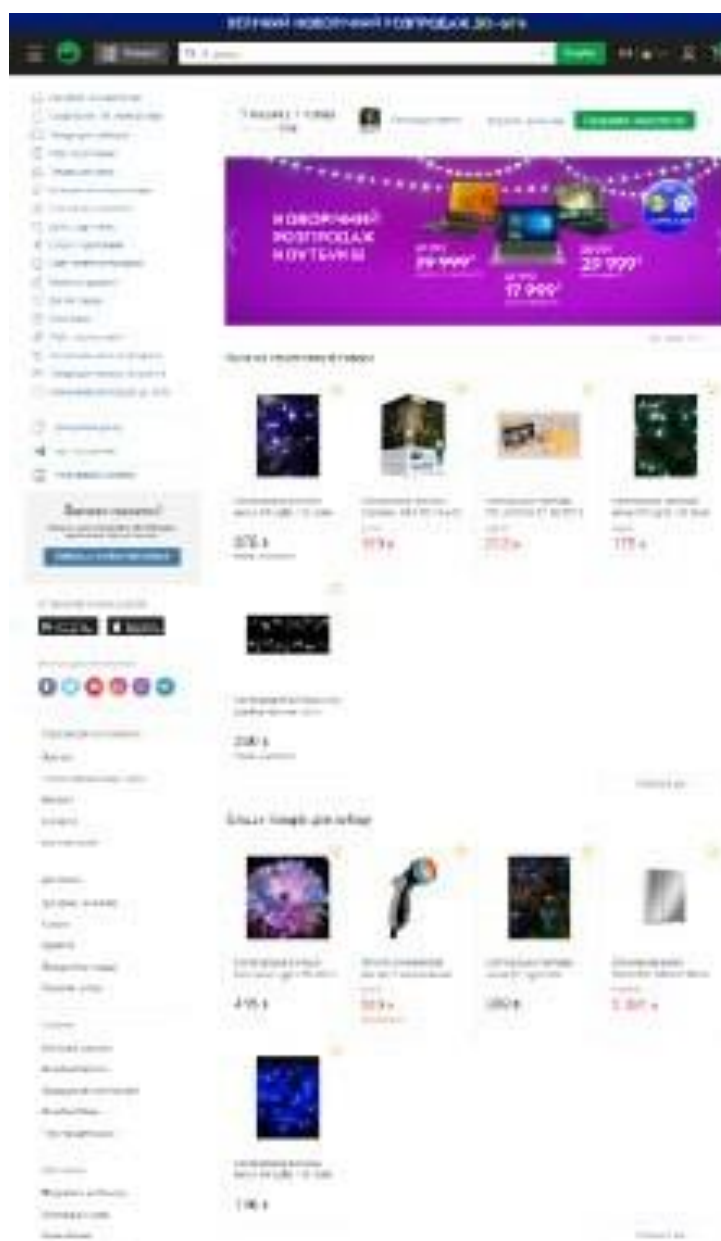


Рисунок 4.7 Досліджувана веб-сторінка

Результати аналізу сторінки зазначені на Рисунку 4.8, та є досить передбачуваними.

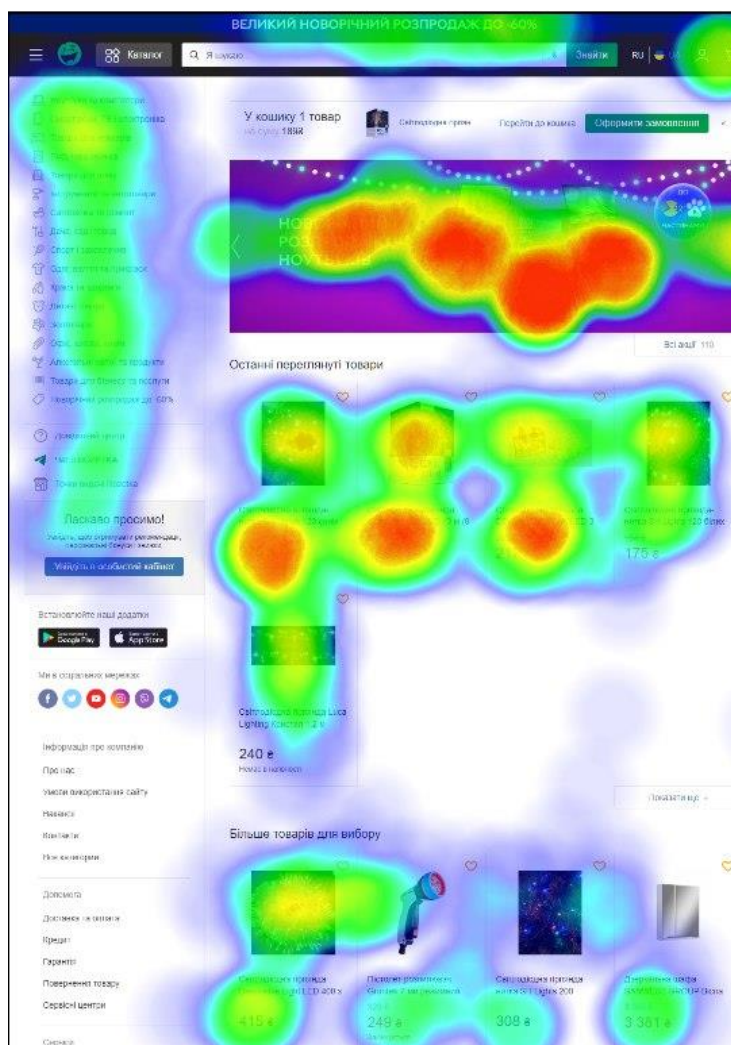


Рисунок 4.8 Теплова карта абсолютної тривалості погляду

Виходячи з результатів експерименту ми можемо зробити висновок стосовно цієї веб-сторінки. Ми бачимо що хоч концентрація уваги на окремих елементах велика на початку сторінки. Багато уваги користувачів прикуто до банеру а не до товару. Концентрація уваги на товарах значно менша ніж на банері, тому слід або його прибрати, або зробити акцент на товарах. Меню сайту хоч і має невеликий показник, але його гарно видно та легко орієнтуватись.

#### Висновки до розділу 4

Для дослідження відстеження відповідності фокусу зору зі структурою інформації на моніторі було проведено декілька експериментів за різних сайтів.

Результати експериментів показали, що багато сайтів є зручними для користувачів але мають багато відволікаючої інформації. Також, було встановлено, що картинки розміщені на веб-сторінках забирають багато уваги користувачів, що

добре для продажів. Порівнюючи результати експериментів, було встановлено, що результат відстеження фокусу зору при аналізі графічних інтерфейсів є надзвичайно важливим оскільки дає багато інформації про поведінку користувачів, що дозволить оптимізувати інтерфейс користувача та тим самим збільшити ефективність продаж.

## ВИСНОВКИ

У роботі було виконано дослідження графічних інтерфейсів з допомогою айтрекінгу з використанням апаратних засобів, які доступні кожному.

Реалізовано програмне середовище, яке здатне дослідити відстеження відповідності фокусу зору зі структурою інформації на моніторі з використанням веб-камери пристрою.

У ході розробки проекту використовувалися новітні технології. Завдяки дотриманню принципів об'єктно-орієнтованого дизайну, систему буде легко супроводжувати та вдосконалювати.

Під час аналізу предметної сфери було висвітлено основні існуючі проблеми ринку програмних та апаратних засобів окулографії, виконано огляд бібліотек комп'ютерного зору та їх порівняння.

Проаналізовано бібліотеки комп'ютерного зору, наведено метод відстеження відповідності фокусу зору зі структурою інформації на моніторі, а також розроблено комплексну модель користувацького інтерфейсу додатка.

Інтерфейс програми було спроектовано з урахуванням потреб цільового користувача. Було досягнуто основної мети – зробити інтерфейс зручним та зрозумілим для використання.

Було проведено декілька експериментів за різних умов використання веб-камери, використання різних налаштувань процесу калібрування, зміна положення стану голови користувача під час експерименту.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вудвуд В. В. Нейромаркетинг – новітній інструмент впливу на поведінку споживачів у недосконалих умовах ринкової економіки України / В. В. Вудвуд, А. Я. Білоус //Інноваційна економіка. - 2013. - № 7. - С. 210-212.
2. Зозульов О. В. Підходи до оцінювання ефективності маркетингових Інтернеткомунікацій / Є. А. Домашева, О. В. Зозульов. // Економічний вісник НТУУ "КПІ". – 2017.– №14.
3. 7 Most Used Eye Tracking Metrics and Terms [Electronic resource] // Imotions. – 2015. – Resource access mode: <https://imotions.com/blog/7-terms-metrics-eye-tracking/>.
4. Ай-трекинг для оценки пользовательского опыта [Електронний ресурс] // GetGoodRank – Режим доступу до ресурсу: <http://blog.getgoodrank.ru/eye-tracking-dlya-ocenkipolzovatelskogo-opyta/>.
5. Кан Е.Я. Нейромаркетинг – механизм манипулирования потребителями [Електронний ресурс] / Е.Я. Кан. – Режим доступу до ресурсу: <https://scholar.google.com/citations?user=hf5v5uUAAAAJ&hl=ru>.
6. Следи за взглядом, или технология айтрекинга увеличивает продажи [Електронний ресурс] // Demis group – Режим доступу до ресурсу: <http://www.demis.ru/articles/eyetracking>.
7. NeuroLab Hardware [Електронний ресурс] // CoolTool. – 2017. – Режим доступу до ресурсу: <https://cooltool.com/neurolab>.
8. <http://projects.dune-hd.com/bitstream/handle/2010/24414/2936.pdf?sequence=1&isAllowed=y>
9. Alexandra Papoutsaki WebGazer: Scalable Webcam Eye Tracking Using User Interactions/ WebGazer: Scalable Webcam Eye Tracking Using User Interactions, Patsorn Sangkloy, James Laskey, Nediya Daskalova, Jeff Huang, James Hays// Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence
10. Т.О. Говорущенко, В.С. Гончарук, К.О. Кобель Метод використання технології eye-tracking у проектуванні інтерфейсів користувача / Т.О. Говорущенко // Вісник Хмельницького національного університету. – 2018. – №2. – С. 259.

11. Kailie Chang DETECTING USER BEHAVIOR WITH WEBGAZER DATA/ California State Polytechnic University 2021
12. Айтрекинг в психологической науке и практике / Ред. В. А. Барабанщиков. – М.: Когито-Центр, 2015.– 410 с.
13. Pavlenko V., Salata D., Dombrovskiy M. and Maksymenko Yu. Estimation of the Multidimensional Transient Functions Oculo-Motor System of Human // Mathematical Methods and Computational Techniques in Science and Engineering: AIP Conf. Proc. MMCTSE'2017, Cambridge, UK. – 2017. – Vol. 1872. – Melville, New York. Published by AIP Publishing. – P.110-117.
14. Pavlenko V. D., Salata D. V. and Chaikovskiy H. P. Identification of a Oculo-Motor System Human Based on Volterra Kernels // Inter. national Journal of Biology and Biomedical Engineering. – 2017. – Vol. 11. – P. 121-126.
15. Салата Д. В., Чори В. В., Mishchenko I., Павленко В. Д., Milosz M., Dzienkowski M. Экспериментальные исследования глазодвигательной системы с помощью айтрекера TOBII PRO TX300 // Матеріали VIII Міжнар. конф. студентів і молодих науковців «Сучасні інформаційні технології 2018», МІТ-2018. – Одеса: ОНПУ ІКС, 2018. – С.105-106.
16. Айтрекинг в психологической науке и практике / Отв. ред. В. А. Барабанщиков. – М.: Когито-Центр, 2015. – 410 с. ISBN 978-5-89353-477-1
17. Барабанщиков В.А. Регистрация и анализ направленности взора человека / В. А. Барабанщиков, А. В. Жегалло – М.: ИП РАН, 2013.
18. Барабанщиков В. А. Методы регистрации движений глаз в психологии: основы учебно-методического комплекса / В. А. Барабанщиков, А. В. Жегалло // Экспериментальная психология. – 2014. – № 1. – С. 132–137.
19. Galdia Chiara. Eye movement analysis for human authentication: Critical survey. / Chiara Galdia, Michele Nappib, Daniel Riccioc and Harry Wechsler // Pattern Recognition Letters – November 2016. – P. 1-13. DOI: 10.1016/j.patrec.2016. 11.002
20. Guestrin E. D. General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections / E. D. Guestrin, M. Eizenman // IEEE Transactions on



biomedical engineering. – 2006. – Vol. 53, Issue – P. 1124–1133. DOI: 10.1109/tbme.2005.863952

21. Pavlenko Vitaliy. Estimation of the Multidimensional Transient Functions Oculo-Motor System of Human / Vitaliy Pavlenko, Dmytro Salata, Mykola Dombrovskiy and Yuri Maksymenko // Mathematical Methods and Computational Techniques in Science and Engineering: AIP Conf. Proc. MMCTSE 2017, Cambridge, UK, 24-26 February 2017. Vol. 1872. Melville, New York, 2017. – 110-117. 020014-1–020014-8; DOI: 10.1063/1.4996671. Published by AIP Publishing. 978-0- 7354-1552-2.

22. Pavlenko V. D. Identification of a Oculo-Motor System Human Based on Volterra Kernels / V. D. Pavlenko, D. V. Salata, H. P. Chaikovskiy // Int. J. of Biology and Biomedical Engineering. – 2017. – Vol. 11. – P. 121-126.

23. Jansson D. Parametric and Nonparametric Analysis of Eye-Tracking Data by Anomaly Detection / D. Jansson, O. Rosén, O., A. Medvedev // IEEE Trans. Control Syst. Technol. 2015, 23, 1578–1586.

24. Rigas I. Biometric Recognition via the Complex Eye Movement Behavior and the Incorporation of Saccadic Vigor and Acceleration Cues / I. Rigas, O. V. Komogortsev and R. Shadmehr // ACM Transactions on Applied Perception, 2016, 13 (2), 1-21.

25. Pavlenko V. Estimation of the Multidimensional Transient Functions Oculo-Motor System of Human / Vitaliy Pavlenko, Dmytro Salata, Mykola Dombrovskiy and Yuri Maksymenko // Mathematical Methods and Computational Techniques in Science and Engineering: AIP Conf. Proc. MMCTSE 2017, Cambridge, UK, 24-26 February 2017. Vol. 1872. Melville, New York, 2017, 110-117. 020014-1–020014-8; doi: 10.1063/1.4996671. Published by AIP Publishing. 978- 0-7354-1552-2.

26. Pavlenko V. D. Identification of a Oculo-Motor System Human Based on Volterra Kernels / V. D. Pavlenko, D. V. Salata, H. P. Chaikovskiy // Int. J. of Biology and Biomedical Engineering, 2017, Vol. 11, 121-126.

27. Снігур Ю.О., Разносілін В.В. Отслеживание фокуса зрения пользователя при работе с компьютером: тез. XIII Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості



та освіти» (11.12-12.12.2019) / Дніпровський національний університет залізничного транспорту академіка В. Лазаряна, 2019.

28. Снігур Ю.О., Розробка засобів відстеження відповідності фокусу зору зі структурою інформації на моніторі ПК: тез. . XIV Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіти» / Дніпровський національний університет залізничного транспорту академіка В. Лазаряна, 2020.

29. Стаття з Web-сайту Строеие и функции глаза [Електронний ресурс]. – Режим доступу: <https://excimerclinic.ru/press/stroenieglaza/>.

30. Аналіз сучасного стану дослідження окулографії [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BA%D0%BA%D0%B0%D0%B4%D0%B0>.

31. Матеріал з Вікіпедії — доступна енциклопедія. Окулографія [Електронний ресурс]. – Режим доступу :<https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%83%D0%BB%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F>.

32. История айтрекинга в лицах и картинках. [Електронний ресурс]. – Режим доступу: <https://usabilityin.ru/eye-tracking-history/>.

33. Стаття з Web-сайту. Yarbus, eye movements, and vision. В W Tatler, N J Wade, H Kwan, J M Findlay, B M Velichkovsky Published under a Creative Commons Licence a Pion publication. Copyright 2010 – Режим доступу: <https://journals.sagepub.com/doi/pdf/10.1068/i0382>.

34. Барабанщиков, В. А. Методы регистрации движений глаз: теория и практика [Текст] / В. А. Барабанщиков, А. В. Жегалло // Электронный журнал «Психологическая наука и образование». – 2010. – № 5. – С. 240–252.

35. Костин, А. Современные ай-трекеры и их возможности для юзабилити-тестирования. Часть II [Електронний ресурс]. – Режим доступу: <https://usabilitylab.ru/blog/obzormodelej-aj-trekerov/>.

36. Айтрекинг. Методическое пособие по применению [Текст] / [сост. Е.Ю.Шелепин, К.Ю.Шелепин, К.А.Скуратова]. – СКИФИЯ-ПРИНТ, 2019. – 54 с..

37. Understanding different aspects of learning [Электронный ресурс]. – Режим доступа: <https://www.tobiiipro.com/applications/scientific-research/education/>.

38. Halszka Jarodzka, Кеннет Холмквист. Айттрекинг в образовательной науке: теоретические основы и исследовательские программы. [Электронный ресурс]. – Режим доступа: [https://www.researchgate.net/publication/316633908\\_Eye\\_tracking\\_in\\_Educational\\_Science\\_Theoretical\\_frameworks\\_and\\_research\\_agendas](https://www.researchgate.net/publication/316633908_Eye_tracking_in_Educational_Science_Theoretical_frameworks_and_research_agendas).

39. Милках Луханю. Устройства слежения за глазами могут улучшить системы образования. [Электронный ресурс]. – Режим доступа: <https://techmoran.com/2019/11/12/eye-tracking-devices-can-improve-education-systems/>.

40. David Rosengrant, Doug Hearnington, Kerriann Alvarado and Danielle Keeble. Following Student Gaze Patterns in Physical Science Lectures. [Электронный ресурс]. – Режим доступа: [https://www.tobiiipro.com/siteassets/tobii-pro/customercases/tobii\\_customercasepaper\\_followingstudentgazepatterns\\_in\\_physicalsciencelectures.pdf/?v=1.0](https://www.tobiiipro.com/siteassets/tobii-pro/customercases/tobii_customercasepaper_followingstudentgazepatterns_in_physicalsciencelectures.pdf/?v=1.0).

41. Jonathan Oglesby, project leader, Department of Geography and Geology. Eye Tracking and Water Education in West Africa. [Электронный ресурс]. – Режим доступа: <https://www.tobiiipro.com/applications/scientific-research/education/customer-cases/westernkentucky-university/>.

42. Шахин Гадир Сравнительный анализ библиотек компьютерного зрения. – Access Mode : DOI: 10.24411/2520-6990-2019-10812/ «Colloquium-journal» № 24(48), 2019 / ARCHITECTURE.

43. Стаття з Web-сайту Аннотации лицевых точек [Электронный ресурс]. – Режим доступа: <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>.

44. Стаття з Web-сайту Tracking your eyes with Python [Электронный ресурс]. – Режим доступа: <https://medium.com/@stepanfilonov/tracking-your-eyes-with-python3952e66194a6>.

45. Стаття з Web-сайту Обработка растровых изображений [Электронный ресурс]. – Режим доступа: <http://pzs.dstu.dp.ua/ComputerGraphics/raster/index.html>.

46. SATOSHI SUZUKI, Topological Structural Analysis of Digitized Binary Images by Border Following. / SATOSHI SUZUKI, KEIICHI ABE // COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING N,32-46 (1985) – Access Mode : <http://pdfs3.xuebalib.com:1262/xuebalib.com.17233.pdf>.

47. Айттрекинг в психологической науке и практике [Текст] / под ред. В.А. Барабанщикова. – Москва : Когито-Центр, – 2016. – 408 с.

48. Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague: Real-Time Eye Blink Detection using Facial Landmarks / Tereza Soukupova and Jan ´ Cech ~ Center for Machine Perception: 21st Computer Vision Winter Workshop Luka Cehovin, Rok Mandeljic, Vitomir ~ Struc (eds.) . – Rimske Toplice, Slovenia, February 3–5, 2016. – Access Mode : URL : <http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf> – Title from Screen.

49. Матеріал з Вікіпедії — доступна енциклопедія. Метод найменших квадратів [Електронний ресурс]. – Режим доступу [https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\\_%D0%BD%D0%B0%D0%B9%D0%BC%D0%B5%D0%BD%D1%88%D0%B8%D1%85\\_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D1%96%D0%B2](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BD%D0%B0%D0%B9%D0%BC%D0%B5%D0%BD%D1%88%D0%B8%D1%85_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D1%96%D0%B2).

50. Стаття з Web-сайту Розробка uml діаграми варіантів використання [Електронний ресурс].– Режим доступу: <https://studfile.net/preview/5200239/page:6/#:~:text=%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0%20%D0%B2%D0%B0%D1%80%D1%96%D0%B0%D0%BD%D1%82%D1%96%D0%B2%20%D0%B2%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F%20E2%80%93%20%D1%86%D0%B5%20%D0%B3%D1%80%D0%B0%D1%84,%D1%96%20%D0%B2%D1%96%D0%B4%D0%BD%D0%BE%D1%81%D0%B8%D0%BD%20%D0%BC%D1%96%D0%B6%20%D1%86%D0%B8%D0%BC%D0%B8%20%D0%B5%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D0%BC%D0%B8>.

51. Стаття з Web-сайту Диаграммы классов [Електронний ресурс].– Режим доступу: <https://it.wikireading.ru/27850>.

52. ДСанПІН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», затвердженими Постановою головного державного санітарного лікаря України 10 грудня 1998 р. № 7.

53. ДСТУ 7299:2013 Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки, затверджено та введено в дію наказом міністерства економічного розвитку і торгівлі України 14.10.2013 № 1231.

54. ДСТУ ISO 9241-1:2003 Національний стандарт України. Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 1. Загальні положення.

55. ДСТУ ISO 9241-6:2004 Національний стандарт України. Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 6. Вимоги до робочого середовища, введено в дію з 01.01.2006.

56. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затвердженими Наказом Міністерства соціальної політики України 14.02.2018 № 207.

57. ДБН В.2.5-28:2018 Державні будівельні норми України «Природне і штучне освітлення», затверджені наказом Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України 03.10.2018 № 264, введено в дію з 01.03.2019.

58. ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень», затверджені Постановою головного санітарного лікаря України № 42 від 1 грудня 1999 року.

59. ДСН 3.3.6.037-99 «Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку», затверджені Постановою головного санітарного лікаря України № 37 від 1 грудня 1999 року.

60. Закон України «Про охорону праці» зі змінами і доповненнями, введений в дію Постановою ВР № 2695-XII від 14.10.92, ВВР, 1992, № 49, ст.669.

61. «Керівництво з визначення і оцінки важкості та напруженості трудового процесу» НАМН України від 02.05.2019р.

62. ДСанНіП «Гігієнічна класифікація праці», затверджені Наказом Міністерства охорони здоров'я України №248 від 08.04.14р.

63. НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні», затверджені Наказом Міністерства внутрішніх справ України 30 грудня 2014 року N 1417, зареєстровані в Міністерстві юстиції України 05 березня 2015 р. за N 252/26697.

64. Кодекс цивільного захисту України зі змінами та доповненнями, Відомості Верховної Ради (ВВР), 2013, № 34-35, ст.458.

65. Блок схема веб-проекту, створення основних блоків[Електронний ресурс]. Режим доступу до джерела: <http://pro-prof.com/archives/1462>

66. Поняття веб простору та веб ресурсу. [Електронний ресурс]. – Джерело: <http://usw.com.ua/profiles/blogs/ponyattya-vebprostoru-ta>

67. Ай-трекінг для оцінки пользовательского опыта [Електронний ресурс] // GetGoodRank –

а. Режим доступу до ресурсу: <http://blog.getgoodrank.ru/eye-tracking-dlya-ocenki-polzovatelskogo-opyta/>.

68. Most Used Eye Tracking Metrics and Terms [Electronic resource] // Imotions. – 2015. – Resource access mode: <https://imotions.com/blog/7-terms-metrics-eye-tracking/>.

69. Вучетич Є.В. Стандарт тестування персоналу / Вучетич Є.В., Костромина С.Н., Кукаркин Б.А. // .// Науковий журнал "Теория и практика современной науки". – 2013. – №38. – С. 9-15.

70. Іванова С. В. Оценка компетенций методом интервью: Универсальное руководство / С. В. Иванова. – Москва: Альпіна Паблішерз, 2011. – 155 с.

71. Бурдюгова О. В. Оценка персонала организации: цели, этапы, методы, критерии./ Бурдюгова О. В., Коркешко О. Н., Притков Р. М., Ахтямова А. А.// Науковий журнал "Теория и практика современной науки". – 2010. – №18. – С. 6.

72. Зленко А. М. Компетенції в системі оцінки персоналу / А. М. Зленко // Економічний вісник університету. - 2013. - № 20(1). - С. 99-102. 11. Стенли Р. Перечислительная комбинаторика / Пер. с англ. – М.: Мир , 1990. – 440 с.



## ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Проректор

Українського державного університету  
науки і технології

Анатолій РАДКЕВИЧ

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Технічне завдання  
ЛИСТ ЗАТВЕРДЖЕННЯ  
1116130.01285-01-ЛЗ

Завідувач кафедри КІТ

\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки

\_\_\_\_\_Олександра ГОРБОВА

Виконавець

\_\_\_\_\_Ілля ФРОЛОВ

Нормоконтролер

\_\_\_\_\_Світлана ВОЛКОВА



ЗАТВЕРДЖЕНО  
1116130.01285-01

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Технічне завдання

Аркушів 22

### **АНОТАЦІЯ**

Документ 1116130.01285-13 «Аналіз графічних інтерфейсів за допомогою трекінга очей» є керівництвом для проектування та розробки програмного додатку «Аналіз графічних інтерфейсів за допомогою трекінга очей», та входить до складу документації на проект. Розробка додатку для трекінгу очей необхідна для отримання відповідної статистики, для подальшої оптимізації графічних інтерфейсів, та для обґрунтування щодо доцільності використання та поліпшення такого методу аналізу в реальному житті. В документі містяться основні вимоги до розробки програмного додатку та його функціональні можливості.

## ЗМІСТ

ВСТУП.....	Error! Bookmark not defined.
1 ПІДСТАВА ДЛЯ РОЗРОБКИ.....	Error! Bookmark not defined.
2 ПРИЗНАЧЕННЯ РОЗРОБКИ.....	Error! Bookmark not defined.
2.1 Функціональне призначення .....	Error! Bookmark not defined.
2.2 Експлуатаційне призначення.....	Error! Bookmark not defined.
3 ВИМОГИ ДО ПРОГРАМИ .....	Error! Bookmark not defined.
3.1 Вимоги до функціональних характеристик .....	Error! Bookmark not defined.
3.2 Вимоги до надійності.....	Error! Bookmark not defined.
3.3 Умови експлуатації .....	Error! Bookmark not defined.
3.4 Вимоги до складу і параметрів технічних засобів	Error! Bookmark not defined.
3.5 Вимоги до інформаційної та програмної сумісності.....	Error! Bookmark not defined.
3.6 Вимоги до маркування і упаковки .....	Error! Bookmark not defined.
3.7 Вимоги до транспортування і зберігання .....	Error! Bookmark not defined.
4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....	Error! Bookmark not defined.
5 ВИЗНАЧЕННЯ ВИТРАТ НА ПРОЕКТУВАННЯ ПРОГРАМИ	Error! Bookmark not defined.
6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ .....	Error! Bookmark not defined.
7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ .....	Error! Bookmark not defined.
БІБЛІОГРАФІЧНИЙ СПИСОК.....	Error! Bookmark not defined.

## **ВСТУП**

Все більше і більше людей з кожним роком проводять свій вільний час онлайн. Вони так спілкуються, розважаються, роблять покупки, бронюють білети та готелі, навчаються. Тому надзвичайно важливо вміти розробити зручний графічний інтерфейс. Іноді буває складно передбачити наскільки розроблений інтерфейс буде зручним та де саме краще розмістити конкретні елементи, у цьому може допомогти айтрекінг.

Додаток для аналізу за допомогою трекінга очей допоможе більш точно проводити дослідження та аналізувати отримані дані. Це посприятиме створенню більш зручних інтерфейсів у додатках, на сайтах, у іграх а також буде мати великий вплив у маркетингу та медицині.

Основна термінологія: КАЛІБРУВАННЯ, АЙТРЕКІНГ, ФОКУС ЗОРУ, РОЗПІЗНАВАННЯ ОБРАЗІВ.

Область застосування: ДОСЛІДЖЕННЯ ПРИ РОЗРОБЦІ ГРАФІЧНИХ ІНТЕРФЕЙСІВ ТА МАРКЕТИНГОВІ ДОСЛІДЖЕННЯ.

## **1 ПІДСТАВА ДЛЯ РОЗРОБКИ**

Основою для розробки є наказ ректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проектів» №01206 ст. від 10.12.2022 р.

Тема проекту: «Аналіз графічних інтерфейсів за допомогою трекінга очей».

Керівник дипломного проекту: доцент кафедри «КІТ» О.В. Горбова.

## **2 ПРИЗНАЧЕННЯ РОЗРОБКИ**

### **2.1 Функціональне призначення**

Функціональним призначенням розробки є аналіз графічних інтерфейсів за допомогою трекінга очей. Програма не призначена для відстеження фокусу зору кількох користувачів одночасно.

### **2.2 Експлуатаційне призначення**

Експлуатаційне призначення полягає у відстеженні відповідності фокусу зору, задля дослідження поведінки користувача під час роботи з графічним інтерфейсом. Результат дослідження дозволить зрозуміти, на які елементи графічного інтерфейсу зосереджено більше уваги, а які залишилися поза увагою, скільки часу знадобилося для засвоєння матеріалу, тощо. Дану інформацію можна використовувати для оптимізації інтерфейсів та можливості адаптувати, налаштовувати графічні інтерфейси програм та додатків.

### **3 ВИМОГИ ДО ПРОГРАМИ**

#### **3.1 Вимоги до функціональних характеристик**

Вимоги до функціональних характеристик програмного забезпечення:

- відстеження координат фокусу зору користувача в області екрану монітора під час роботи за комп'ютером. Відстеження відбувається за рахунок знаходження зміщення центру зіниці ока відносно центру ока;
- використання веб-камери ноутбуку для розпізнавання руху зіниць під різним кутом;
- забезпечення процесу калібрування задля підвищення точності розпізнавання фокусу зору;
- встановлення інформаційного контенту на моніторі та відстеження відповідності отриманих координат фокусу зору зі структурою інформації на екрані. Знайдена відповідність зберігається у файл для подальшого аналізу поведінки користувача під час роботи з інформаційним контентом.

Вхідними даними програми є:

- параметри налаштування алгоритму розпізнавання зіниці ока: розмір апертури медіанного фільтр, порогове значення;
- параметри налаштування процесу калібрування: форма калібрувальної точки, колір, таймер затримки позиції калібрувальної точки у секундах, розмір калібрувальної точки, кількість позицій калібрувальної точки по осі ординат та абсцис;
- потік відеокadrів зі зображенням користувача, відзнятих відеокамерою під час роботи користувача з комп'ютером. Результатом роботи програми є наступні вихідні дані:

- дані, зібрані під час процесу калібровки. Файл повинен містити наступні дані: координати позиції калібрувальної точки, координати контуру лівого та правого ока, координати центру зіниці лівого та правого ока;
- дані зібрані під час співвідношення координат фокусу зору з інформаційним контентом на моніторі: точка фокусу зору, ім'я елементу інформаційного контенту, область розташування елементу інформаційного контенту на моніторі;
- зображення інформаційного контенту на моніторі з відображенням координати фокусу зору у вигляді червоної точки.

### 3.2 Вимоги до надійності

Надійність системи повинна бути забезпечена за наступними напрямками:

- забезпечення працездатності компонентів програмно-технічної платформи;
- збереження даних.

При цьому повинна вимагатися мінімальна увага з боку системного адміністратора щодо реакції на усунення наслідків відмов компонентів, а також програмно-апаратними засобами повинно бути забезпечене збереження даних. Збереженість інформації на випадок аварій повинна бути забезпечена у повному обсязі. Резервне копіювання має відбуватися з періодичністю, що забезпечує повне збереження та відновлення даних.

Збереження даних має забезпечуватися у випадках:

- вимкнення живлення;
- відмови технічних засобів обробки інформації;
- помилки, збоїв або руйнування програмного забезпечення.

Вимоги щодо надійності системи можуть бути уточнені виконавцем та повинні бути зазначені в наступних версіях технічного завдання.



### 3.3 Умови експлуатації

Даний програмний продукт може використовуватись в умовах, які відповідають вимогам документу [1]. Для нормального функціонування програмного продукту необхідно виконання наступних вимог:

- ЕОМ повинна відповідати вимогам чинних в Україні стандартів, нормативних актів з охорони праці [2];
- програма повинна експлуатуватись у приміщенні, призначеному для роботи з ЕОМ, з відповідними кліматичними умовами: температура 21° – 25°С та вологість 40 - 60%;
- стан технічних засобів повинен задовольняти відповідним нормам та вимогам;

Для забезпечення надійного функціонування програмного продукту необхідно дотримуватися таких умов:

- програма призначена для роботи на ЕВМ з операційною системою Windows 10;
- працювати з програмою може людина, яка володіє навичками роботи з ОС у графічному режимі та ознайомилась з керівництвом користувача.

### 3.4 Вимоги до складу і параметрів технічних засобів

Пристрій, що має щонайменше 128Мб вільного місця на вбудованому носії інформації та 1024Мб оперативної пам'яті для коректної роботи. Вбудована відеокамера з роздільною здатністю 1280x720. Для зручної роботи з програмою необхідно мати монітор з роздільною здатністю 1366x768 або вище, веб-камера, маніпулятор «миша» та клавіатуру. Для використання ПЗ необхідна підключення до мережі Інтернет.

### 3.5 Вимоги до інформаційної та програмної сумісності

Вимоги до інформаційної та програмної сумісності: будь-яка система яка підтримує сучасні веб браузері з їх останніми оновленнями.

### 3.6 Вимоги до маркування і упаковки

Програмний продукт може маркуватися як зображено на рис. 3.1:

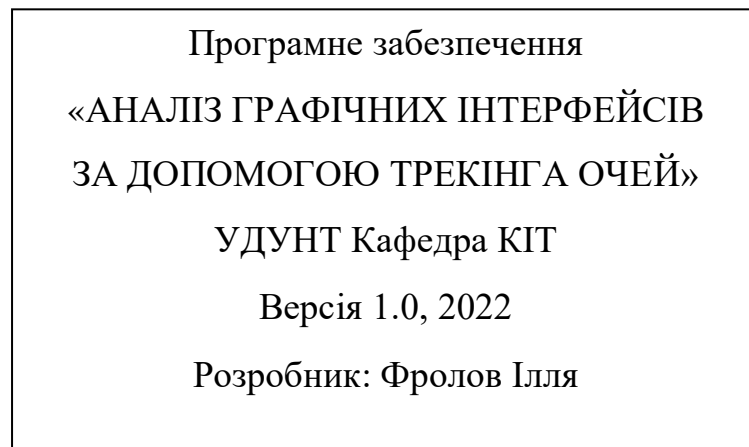


Рисунок 3.1 – Маркувальний штамп

### 3.7 Вимоги до транспортування і зберігання

Транспортувати програмний продукт можна через всесвітню систему взаємоз'єднаних комп'ютерних мереж, що базуються на комплекті Інтернет-протоколів

Термін збереження обумовлений збереженням інформації на носії віддаленого серверного сховища.

#### **4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

До складу програмної документації має входити технічне завдання та робочий проект.

До складу робочого проекту мають входити:

- специфікація;
- текст програми;
- опис програми;
- керівництво користувача.

Керівництво відстеження фокусу зору користувача. Вся документація до програми повинна задовольняти вимогам державного стандарту до оформлення програмних документів [3].

## 5 ВИЗНАЧЕННЯ ВИТРАТ НА ПРОЕКТУВАННЯ ПРОГРАМИ

Основна мета розробки техніко-економічного обґрунтування (ТЕО) – дати фінансову оцінку передбачуваних витрат та одержуваного корисного результату, а також оцінити прибутковість проекту і, в кінцевому підсумку, економічну доцільність його розробки та впровадження.

Початковим етапом розрахунку величини трудових витрат розробників є оцінка розміру програмного забезпечення. Основні відмінності методик, що застосовуються в оцінці трудовитрат, полягають у використуваному типі критерію оцінки якості [4].

Згідно моделі COCOMO, розмір проекту  $S$  вимірюється в рядках коду LOC (KLOC), а трудовитрати в людино-місяцях.

$$E = a \cdot S^b \cdot EAF, \quad (5.1)$$

де  $E$  – витрати праці на проект (в людино-місяцях);

$S^b$  – розмір коду (в KLOC);

$EAF$  – фактор уточнення витрат (effort adjustment factor).

Для простих систем,  $a = 2,4$ ;  $b = 1,05$

Отже розмір програмного коду становить 5059 рядки:

$$E = 2,4 \cdot 5,059^{1,05} \cdot 1 = 13,15$$

Отже, згідно моделі COCOMO, орієнтовні трудовитрати на проект складуть приблизно 13,15 людино-місяці.

Нижче наведені розрахунки вартості розробки «Автоматизована система оцінки схожості програм». Основними статтями витрат прийняті:

- основна заробітна плата;
- відрахування на соціальні потреби;
- накладні витрати;
- витрати на персональний комп'ютер і ліцензійні базові програмні засоби.

Основна заробітна плата (ОЗП) оцінює працю інженера-програміста зі

створення програмного продукту і визначається виходячи з кількості розробників, часу виконання розробки (годин), а також заробітної плати в розрахунку на одну годину. Розрахунок заробітної платні проводиться по формі табл. 5.1.

Таблиця 5.1 – Фонд місячної заробітної плати

№ п/п	Посада виконавця	Оклад, грн/міс	Кількість		Сума зарплати грн
			чол	місяців	
1	інженер- програміст	15750 [2]	1	13,15	25830

Описаний в проекті програмний продукт був розроблений одним програмістом в період з 25.07.21 до 25.10.22, що складає 93 дні або приблизно 13,28 робочих тижнів. Витрати робочого часу прийняті за 40 годин у тиждень. Погодинна ставка кваліфікованого інженера–програміста складає 93,75 грн/год. Таким чином, витрачено робочого часу:

$$t_{\text{розробки}} = N_{\text{чол}} \times N_{\text{тиж}} \times N_{\text{год}}, \quad (5.2)$$

де  $N_{\text{чол}}$  – кількість виконавців, чол;

$N_{\text{тиж}}$  – тривалість розробки;

$N_{\text{год}}$  – витрати робочого часу, год;

$$t_{\text{розробки}} = 1 \cdot 13,28 \cdot 40 = 531,2 \text{ чол/год.}$$

ОЗП визначається за формулою:

$$\text{ОЗП} = t_{\text{розробки}} \cdot N \cdot K_{KB}, \quad (5.3)$$

де  $t_{\text{розробки}}$  – витрати праці у чол/год;

$N$  – погодинна ставка;

$K_{KB}$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. Коефіцієнт кваліфікації розробника ( $k$ ) - ступінь підготовленості виконавця до дорученої йому роботи (він визначається залежність

від стажу праці становить:

- для працюючих до 2 років- 0,75;
- від 2 до 3 років 1,0;
- від 3 до 5 років - 1,1-1,2;
- від 5 до 7 років - 1,3-1,4;
- понад 7 років - 1,5-1,6.

В даному випадку *ККВ* приймається 0,75. ОЗП складає:

$$\text{ОЗП} = 531,2 \cdot 93,75 \cdot 0,75 = 37350 \text{ грн.}$$

Відрахування на соціальні потреби встановлюються у відсотках від суми заробітної плати (22% [4]):

$$C_{\text{соц}} = \frac{\text{ОЗП} \cdot 22\%}{100\%} \quad (5.4)$$

$$C_{\text{соц}} = \frac{37350 \cdot 22\%}{100\%} = 8217 \text{ грн.}$$

Отримані результати за (5.3) та (5.4) підсумовуються. Вони складають 45567грн. та визначають основні прямі витрати.

Накладні витрати враховують загальногосподарчі витрати по забезпеченню проведення роботи: витрати на опалення, електроенергію, амортизація будівель, зарплату адміністративного персоналу та інше. Вони визначаються в процентах (30 – 40%) від суми прямих витрат:

$$C_{\text{накл}} = \frac{(\text{ОЗП} + C_{\text{соц}}) \cdot 35\%}{100\%}; \quad (5.5)$$

$$C_{\text{накл}} = \frac{(37350 + 8217) \cdot 35\%}{100\%} = 15948,45 \text{ грн.}$$

Протягом усього терміну використання нової техніки підприємство щорічно витрачає певні кошти, пов'язані з її експлуатацією.

Експлуатаційні витрати на персональний комп'ютер визначаються протягом

терміну розробки програмного засобу в залежності від вартості комп'ютеру. В експлуатаційні витрати входять:

- вартість витратних матеріалів;
- витрати на ремонт;
- заробітна плата ремонтника;
- оренда приміщення;
- додаткові витрати – прибирання приміщення, охорона, оренда, комунальні послуги;
- амортизаційні витрати на персональний комп'ютер і програмне забезпечення;
- витрати на електроенергію ( $C_{\text{ел}}$ ), які визначаються за формулою:

$$C_{\text{ел}} = P \cdot B \cdot T_{\text{розр}}, \quad (5.6)$$

- де  $P$  – потужність комп'ютера та допоміжних споживачів електричної енергії, приймається 0,45 кВт/год;
- $B$  – вартість 1 кВт/годин для непобутових споживачів,
- складає 1,8 грн [3];
- $T_{\text{розр}}$  – час роботи з ЕВМ, приймається рівним робочому часу.

Витрати на електроенергію визначаються так:

$$C_{\text{ел}} = 0,45 \cdot 1,8 \cdot 531,2 = 430,27 \text{ грн.}$$

Витрати на витратні матеріали ( $C_{\text{вм}}$ ) протягом всього терміну експлуатації приблизно 10% від вартості комп'ютеру. Вартість робочої

станції приймається 18 000 грн. [6], термін експлуатації – 5 років. Отже, можна визначити ці витрати за період створення програмного засобу:

$$C_{\text{вм}} = B_{\text{ком}} \cdot \frac{N_{\text{д}}}{N_{\text{експ}} \cdot 365} \cdot \frac{10\%}{100\%}, \quad (5.7)$$

де  $B_{\text{ком}}$  – вартість персонального комп'ютеру;

$N_{\text{д}}$  – кількість днів розробки програмного продукту;

$N_{\text{експ}}$  – термін експлуатації персонального комп'ютеру.

Витрати на витратні матеріали визначаються так:

Витрати на витратні матеріали визначаються так:

$$C_{BM} = 18000 \cdot \frac{93}{5 \cdot 365} \cdot \frac{10}{100} = 90 \text{ грн.}$$

Заробітна плата ремонтника ( $C_{\text{рем}}$ ) визначена наступним чином: на ремонт 50 комп'ютерів потрібен один інженер-системотехнік. Його середньомісячна заробітна плата приймається 9000 грн. Тоді в перерахунку на один комп'ютер його заробітна плата за період розробки програмного продукту складає:

$$C_{\text{рем}} = \frac{C'_{\text{рем}}}{N_{\text{КОМ}}} \cdot T_{\text{міс}}, \quad (5.8)$$

де  $C'_{\text{рем}}$  – середньомісячна заробітна плата;

$N_{\text{КОМ}}$  – кількість комп'ютерів на одного ремонтника.

$T_{\text{мес}}$  – час розробки програмного продукту, міс.

Заробітна плата ремонтника ( $C_{\text{рем}}$ ) буде складати:

$$C_{\text{рем}} = \frac{9000}{50} \cdot 13,15 = 2367 \text{ грн.}$$

За статистикою витрати на комплектуючі вироби ( $C_{\text{КОМ}}$ ) для ремонту персонального комп'ютера складає 10% від його вартості за термін його експлуатації, тобто рівні витратам на витратні матеріали:

$$C_{\text{КОМ}} = C_{\text{BM}} = 90 \text{ грн.} \quad (5.9)$$

Амортизаційні відрахування на персональний комп'ютер (АПК) визначені з положення, що амортизаційний період в даний час дорівнює терміну морального старіння обчислювальної техніки і складає 3 роки. Отже, за 3 роки амортизаційні відрахування на персональний комп'ютер дорівнюють вартості комп'ютера. За період проектування амортизаційні відрахування складуть:

$$\begin{aligned} \text{АПК} &= B_{\text{КОМ}} \cdot \frac{N_{\text{Д}}}{N_{\text{експ}} \cdot 365}; \\ \text{АПК} &= 18000 \cdot \frac{13,15}{3 \cdot 12} = 6575 \end{aligned} \quad (5.10)$$



Амортизаційні відрахування на програмне забезпечення (АПЗ) залежать від його циклу заміни. Якщо прийняти термін морального старіння для Windows 5 років то амортизаційні відрахування на програмне забезпечення дорівнюють його вартості.

Для функціонування персонального комп'ютера використовувалася операційна система Windows 10.

$$АКП_w = 13800 \cdot \frac{13,15}{5 \cdot 12} = 3024,5$$

Розрахунок амортизаційних відрахувань на програмне забезпечення зведений в табл. 5.2. Додаткові витрати ( $C_{дод}$ ): прибирання приміщень, охорона, комунальні послуги важко оцінити точно і прийняти рівними 50% заробітної плати інженера- програміст, тобто 7875 гривень на місяць.

Оренду приміщень для однієї людини приймемо рівною 2500 гривень на місяць [5]. Тобто за весь період розробки – 32875 грн. Сумарні експлуатаційні витрати на один персональний комп'ютер складають:

$$C_{експ} = C_{ел} + C_{ВМ} + C_{рем} + АПК + АПО + C_{ор} + C_{дод}; \quad (5.11)$$

$$C_{експ} = 430,27 + 90 + 2367 + 6575 + 3024,5 + 32875 + 7875 = 53236,77 \text{ грн}$$

Результати розрахунків зведено у табл. 5.3.

Таблиця 5.2 – Використовуване програмне забезпечення

Найменування програмного забезпечення	Вартість програмного забезпечення, грн	Джерело придбання	Амортизаційні відрахування, грн
Windows 10	13800	<a href="http://mtsoft.kiev.ua/product/windows-10-professional">http://mtsoft.kiev.ua/product/windows-10-professional</a>	3024,5
Всього:	13800		377,2

Таблиця 5.3 – Експлуатаційні витрати на ПК і ПЗ.

Найменування витрат	Витрати, грн
Витрати на електроенергію	430,27
Вартість витратних матеріалів	90
Витрати на ремонт	2367
Амортизація персонального комп'ютера	6575
Амортизація програмного забезпечення	3024,5
Оренда приміщення	32875
Додаткові витрати	7875
Всього	53236,77

Таким чином, витрати на створення програмного продукту складають:

$$C_{\text{розробки}} = OЗП + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}; \quad (5.12)$$

$$C_{\text{розробки}} = 37350 + 8217 + 15948,45 + 53236,77 = 114752,22 \text{ грн.}$$

Розрахунок витрат зведено у табл. 5.4.

Таблиця 5.4 – Кошторис витрат на розробку програмного засобу

Найменування витрат	Витрати, грн
Основна заробітна плата	37350
Відрахування на соціальні потреби	8217
Накладні витрати	15948,45
Експлуатаційні витрати	53236,77
Всього	114752,22

За отриманими значеннями техніко-економічних показників проекту складено кошторис витрат на розробку сучасного програмного забезпечення для оцінки схожості програм. За результатами розрахунків, приблизна вартість розробки складає 114752,22 грн.

## 6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки приводяться в табл. 6.1.

Таблиця 6.1 – Стадії та етапи розробки

№	Стадії розробки	Етап розробки	Термін
1	Технічне завдання	Постановка задачі	01.10.21 – 30.12.21
		Розробка структур вхідних і вихідних даних	14.01.22 – 30.01.22
		Розробка вимог до програми	03.02.22 – 28.02.22
		Затвердження технічного завдання	02.03.22 - 31.03.22
2	Робочий проект	Розробка і програмування логіки програми	01.04.22 – 30.06.22
		Розробка і програмування користувацького інтерфейсу	01.07.22 – 7.08.22
		Відлагодження програми	10.08.22 – 31.08.22
		Розробка програмної документації	01.09.22 – 30.10.22
3	Впровадження	Підготовка і передача програми і програмної документації для супроводу	01.11.22 – 12.12.22

## **7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ**

Контроль здійснюється за допомогою виконання набору тестів з метою знаходження помилок в програмному продукті та його специфікації. Контроль виконання роботи забезпечується головним керівником розробки. Прийом програмного продукту здійснюється уповноваженою комісією.

## БІБЛІОГРАФІЧНИЙ СПИСОК

1. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».
2. Закон Міністерства охорони здоров'я України від 09.10.2000 № 247 (у редакції наказу МОЗ від 14.03.2006 № 120) «Про затвердження Тимчасового порядку проведення державної санітарно-гігієнічної експертизи».
3. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. – 38 с.
4. Методики оценки трудозатрат по разработке программного обеспечения информационных систем [Текст]: Є. Борисенков; ТарГУ ім. М.Х.Дулати.
5. Огляд статистики зарплатні професії "Інженер-програміст в Україні" [Електронний ресурс]. – Режим доступу: <https://ua.trud.com/ua/salary/2/67643.html> (дата звернення: 10.10.2020).
6. Закон України «Про збір та облік єдиного внеску на загальнообов'язкове державне соціальне страхування», стаття 8, пункт 5. Режим доступу: <https://zakon.rada.gov.ua/laws/show/2464-17#Text>.
7. Стаття з Web-сайту Середня потужність комп'ютера [Електронний ресурс]. – Режим доступу: <https://beasthackerz.ru/uk/windows-7/srednyaya-moshchnost-kompyutera-vtskolko-elektroenergii-potreblyaet.html> (дата звернення: 10.10.2020).
8. Офіційний сайт Національної комісії, що здійснює державне регулювання у сферах енергетики та комунальних послуг. Тарифи на електроенергію для непобутових споживачів. Режим доступу: <http://www.nerc.gov.ua/>.
9. Ноутбук Asus ROG Strix G15 G512LI-HN094. Режим доступу: [https://rozetka.com.ua/asus\\_90nr0381\\_m01620/p231822349/](https://rozetka.com.ua/asus_90nr0381_m01620/p231822349/). 01116130.01192-01 25
10. Системний адміністратор, сервісний інженер. Пошук робочих місць. Режим доступу: <https://www.work.ua/ru/resumes/425938/>

11. Об'ява «Аренда офисов -21, 9, 11, 31, 19 м.кв. центр города, собственник».  
Режим доступа: <https://www.olx.ua/uk/obyavlenie/arenda-ofisov-21-9-11-31-19-m-kv-tsentrgoroda-sobstvennik-IDm1cOs.html#fd7f062b72>;

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Проректор

Українського державного університету  
науки і технології

Анатолій РАДКЕВИЧ

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Робочий проект

ЛИСТ ЗАТВЕРДЖЕННЯ

1116130.01285-01-ЛЗ

Завідувач кафедри КІТ

\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки

\_\_\_\_\_Олександра ГОРБОВА

Виконавець

\_\_\_\_\_Ілля ФРОЛОВ

Нормоконтролер

\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО  
1116130.01285-01

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Специфікація  
Аркушів 2



2  
1116130.01285-01

Позначення	Найменування	Примітка
1116130.01285-01-ЛЗ	Лист затвердження	
1116130.01285-01 12 01-ЛЗ	Лист затвердження	
1116130.01285-01 12 01	Текст програми	
1116130.01285-01 13 01-ЛЗ	Лист затвердження	
1116130.01285-01 13 01	Опис програми	
1116130.01285-01 ІЗ 01-ЛЗ	Лист затвердження	
1116130.01285-01 ІЗ 01	Керівництво користувача	
	Керівництво експерта з моделювання та прогнозування	

ЗАТВЕРДЖЕНО  
1116130.01285-01

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Опис програми

1116130.01285-01  
Аркушів 21

## **АНОТАЦІЯ**

Документ 01116130.01285-01 13 01 «Аналіз графічних інтерфейсів за допомогою трекінгу очей. Опис програми» входить до складу програмної документації до програми. Програмний продукт дозволяє розпізнавати фокус зору користувача, шляхом обробки кадрів, отриманих з веб-камери та шукає відповідність отриманих координат фокусу зору зі структурою інформації на моніторі. В документі міститься опис програми та її функціональних можливостей. Програма реалізована на мові React за допомогою бібліотеки WebGazer.js.

## ЗМІСТ

1 ЗАГАЛЬНІ ВІДОМОСТІ .....	Error! Bookmark not defined.
2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ.....	Error! Bookmark not defined.
3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ.....	Error! Bookmark not defined.
3.1 Алгоритм програми.....	Error! Bookmark not defined.
3.2 Задачі програмного забезпечення .....	Error! Bookmark not defined.
3.3 Базова архітектура системи .....	Error! Bookmark not defined.
3.4 Технологічна платформа .....	Error! Bookmark not defined.
4 ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ.....	Error! Bookmark not defined.
5 ВИКЛИК ТА ЗАВАНТАЖЕННЯ.....	Error! Bookmark not defined.
6 ВХІДНІ ДАНІ.....	Error! Bookmark not defined.
7 ВИХІДНІ ДАНІ .....	Error! Bookmark not defined.
9 ПОРЯДОК РОБОТИ З ПРОГРАМОЮ .....	Error! Bookmark not defined.
10 ПОВІДОМЛЕННЯ .....	Error! Bookmark not defined.

## **1 ЗАГАЛЬНІ ВІДОМОСТІ**

Додаток для аналізу графічних інтерфейсів за допомогою трекінга очей дозволяє зробити висновки щодо необхідності редагування та оптимізації графічних інтерфейсів для покращення сприйняття контенту. Для експлуатації програмного продукту необхідний будь-який пристрій з параметрами, які задовольняють нормальному функціонуванню операційної системи Windows. Програма реалізована на мові React за допомогою бібліотеки WebGazer.js.

## **2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Функціональним призначенням розробки аналіз графічних інтерфейсів за допомогою трекінгу очей та побудова теплової карти досліджуваних інтерфейсів. Програма не призначена для відстеження фокусу зору кількох користувачів одночасно.

## **3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ**

### **3.1 Алгоритм програми**

Алгоритм додатку для аналізу графічних інтерфейсів за допомогою трекінгу очей складається з наступних кроків:

Крок 1: отримання потоку відеокадрів з камери.

Крок 2: обробка відеокадрів. Для кожного кадру розпізнається образ обличчя для пошуку області розташування очей на кадрі.

Крок 3: для кожного ока знаходиться координата його центру та зіниці. Розраховується відстань зміщення зіниці ока відносно його центру.

Крок 4: калібрування. Відбувається збір даних співвідношення координат позиції калібрувальної точки зі обчисленим зміщенням зіниці ока відносно його центру. Дані зберігаються та аналізуються для знаходження залежності між екранними координатами та переміщення зіниці.

Крок 5: налаштування інформаційного контенту на екрані монітора.

Крок 6: підрахунок координат фокусу зору.

Крок 7: пошук відповідності між фокусом зору користувача на інформаційним контентом зображеним на моніторі екрану, шляхом побудови теплової карти на накладання її на дослідж

### **3.2 Задачі програмного забезпечення**

Програмне забезпечення займається наступними задачами:

- отримання кадрів з відеокамери під час запису користувача при роботі з ПК;
- обробка отриманих відеокадрів;
- розпізнавання рис обличчя на кадрі;
- знаходження антропометричних точок розташування очей на обличчі;
- знаходження координат центру ока;
- знаходження координат центру зіниці ока;

- проведення процесу калібрування;
- відстеження процесу блимання очей;
- пошук залежності координати зіниці ока від позиції калібрувальної точки;
- підрахунок координати фокусу зору;
- оцінка похибки знаходження координати фокусу зору;
- пошук відповідності фокусу зору зі структурою інформації на моніторі.

### 3.3 Базова архітектура системи

У процесі проектування було виконано декомпозицію модулів системи відстеження відповідності фокусу зору зі структурою інформації на моніторі на три рівні згідно з моделлю «Модель-вигляд-контролер» (MVC), результати якої наведені на Рисунку 3.1.

Рисунок 3.1 – Модель «Model-View-Controller» системи відстеження відповідностіфокусу зору зі структурою інформації на моніторі

Основна мета застосування цієї концепції полягає в відділенні бізнес-логік (моделі) від її візуалізації (вигляду). За рахунок такого поділу підвищується можливість повторного використання коду.

У рамках архітектурного шаблону MVC програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами:

- модель (Model) – відповідає за збереження та отримання даних з файлової системи та відеокамери. За допомогою цього компоненту система отримує кадри з відеокамери, зберігає результат калібрування у файл, зчитує дані за файлу для відображення інформаційного контенту, тощо;
- вигляд (View) – відповідає за представлення даних користувачеві (наприклад, відображення процесу калібрування, інформаційного контенту, тощо). Компонент представлений у вигляді графічного інтерфейсу;
- контролер (Controller) – відповідає за бізнес-логіку. Контролер виконує основну частину роботи системи відстеження відповідності фокусу зору зі



структурою інформації на моніторі. Він отримує дані від компоненту Model, обробляє їх та відправляє у компонент View для відображення користувачеві. Також контролер отримує сигнали у вигляді реакції на дії користувача (натискання кнопки, введення даних у текстове поле) та обробляє їх.

Для реалізації системи відстеження відповідності фокусу зору зі структурою інформації на моніторі обрана об'єктно-орієнтована та функціональна парадигми програмування.

### 3.4 Технологічна платформа

Для розробки системи відстеження відповідності фокусу зору зі структурою інформації на моніторі було обрано середовище розробки React. Для відстеження фокусу зору обрана бібліотека комп'ютерного зору WebGazer.js. Бібліотека WebGazer.js є найпопулярнішою і розвиненою в області обробки зображень і комп'ютерного зору, має базові методи обробки зображення, методи геометричного перетворення, перетворення кольірних просторів, машинного навчання, виявлення об'єктів на зображенні. Вихідний код бібліотеки написаний на JavaScript. Зберігання даних було здійснено за допомогою Firebase.

React - відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Він спрощує створення інтерактивних інтерфейсів, потрібно лише описати, як різні частини інтерфейсу виглядають у кожному стані вашого додатку і React ефективно оновить та відрендерить лише потрібні компоненти, коли дані зміняться. Декларативні інтерфейси роблять код більш передбачуваним і його набагато легше налагоджувати.

React дозволяє розробникам створювати великі вебзастосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Він обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з

іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови вебзастосунків.

React має ряд особливостей:

- одностороння передача даних. Властивості передаються в рендерер компоненту, як властивості html тегу. Компонент не може напряму змінювати властивості, що йому передані, але може їх змінювати через callback функції. Такий механізм називають «властивості донизу, події нагору»;
- віртуальний DOM. Оскільки логіка компонентів написана на JavaScript, замість шаблонів, ви з легкістю можете передавати складні дані у вашому додатку і зберігати стан окремо від DOM. React підтримує;
- JSX. Компоненти React зазвичай написані на JSX. Код написаний на JSX компілюється у виклики методів бібліотеки React. Розробники можуть так само писати на чистому JavaScript;
- атрибути. JSX надає ряд атрибутів елементів, призначених для відображення тих, що надаються у форматі HTML. Користувацькі атрибути також можуть бути передані компоненту. Всі атрибути будуть отримані компонентом як реквізит.

Методи життєвого циклу — це різні методи, які вбудовуються за допомогою ReactJS. Вони дозволяють розробнику обробляти дані в різних точках життєвого циклу програми React. Наприклад:

- `shouldComponentUpdate` — це метод життєвого циклу, який каже Javascript оновити компонент, використовуючи логічні змінні.
- `componentWillMount` — це метод життєвого циклу, який каже Javascript налаштувати певні дані перед монтуванням компонентів (вставлення у віртуальний DOM).
- `componentDidMount` — це метод життєвого циклу, подібний до компонента `WillMount`, за винятком того, що він працює після методу `render`, і може

використовуватися для додавання JSON-даних, а також для визначення властивостей та станів.

— `render` є найважливішим методом життєвого циклу, необхідним у будь-якому компоненті. Метод `render` — це те, що з'єднується з JSX і відображати власний JSX.

React також може рендеритись на сервері, використовуючи Node, і приводити в дію мобільні додатки, які використовують React Native. Принципи роботи React Native в основному такі ж, як ReactJS, за винятком того, що він не маніпулює DOM через VirtualDom. Він працює у фоновому процесі (який інтерпретує Javascript код написаний розробниками) безпосередньо на кінцевому пристрої і спілкується з нативною платформою. React Native взагалі не покладається на HTML, все написано на Javascript і залежить від нативних SDK.

React не намагається надати повну «схему додатків». Він безпосередньо спрямований на побудову користувацьких інтерфейсів, і тому не включає в себе безліч інструментів, які деякі розробники вважають необхідними для створення програми. Це дозволяє вибрати будь-які бібліотеки, які розробник вважає за краще виконувати, щоб виконати певних завдань, таких як здійснення доступу до мережі або локальне зберігання даних.

WebGazer.js — це бібліотека відстеження погляду, яка використовує звичайні веб-камери для визначення розташування відвідувачів веб-сторінки на сторінці в режимі реального часу. Модель відстеження погляду, що міститься в ньому, автоматично калібрується, спостерігаючи, як відвідувачі взаємодіють з веб-сторінкою, і тренує відображення між особливостями ока і положенням на екрані. WebGazer.js повністю написаний на JavaScript, і за допомогою всього декількох рядків коду його можна інтегрувати в будь-який веб-сайт, який хоче краще зрозуміти своїх відвідувачів і змінити їх досвід користувача. Бібліотека повністю працює в клієнтському браузері, тому відео не потрібно надсилати на сервер, а для доступу до веб-камери потрібна згода користувача.

#### **4 ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ**

Для експлуатації програмного продукту необхідний будь-який пристрій з параметрами, які задовольняють нормальному функціонуванню сучасного веб-браузера. Мінімальна конфігурація пристрою, що забезпечить нормальну роботу програмного продукту:

- щонайменше 128 Мб вільного місця на вбудованому носії інформації;
- щонайменше 1024 Мб оперативної пам'яті;
- монітор з роздільною здатністю 1360x780;
- відеокамера з роздільною здатністю 1280x720;
- центральний процесор з тактовою частотою 1 ГГц та більше;
- маніпулятор «миша»;
- стандартна клавіатура;
- веб камера;
- для користування ПЗ необхідна наявність підключення до мережі

Інтернет.

## **5 ВИКЛИК ТА ЗАВАНТАЖЕННЯ**

Для запуску програмного продукту «аналіз графічних інтерфейсів за допомогою трекінгу очей» необхідно виконати запуск додатку засобами передбаченими встановленою на пристрої ОС. Додатком можна користуватися онлайн, для цього потрібен тільки доступ до мережі Інтернет.

## **6 ВХІДНІ ДАНІ**

Вхідними даними програми є:

- список досліджуваних інтерфейсів у вигляді JSX модулів або електронних зображень;
- потік відеокадрів зі зображенням користувача, відзнятих відеокамерою під час роботи користувача з комп'ютером.



## **7 ВИХІДНІ ДАНІ**

Результатом роботи програми є наступні вихідні дані:

- дані, зібрані під час процесу калібрування. Файл містить наступні дані:  
координати позиції калібрувальної точки (значення у вигляді натуральних чисел),  
координати контуру лівого та правого ока (значення у вигляді натуральних чисел),  
координати центру зіниці лівого та правого ока (значення у вигляді цілих чисел);
- дані зібрані під час співвідношення координат фокусу зору з інформаційним контентом на моніторі: точка фокусу зору (значення представлене у вигляді двох координат по осі X та Y);
- зображення інформаційного контенту на моніторі з відображенням координати фокусу зору у вигляді температурної карти.



## 8 ОПИС ПРИЗНАЧЕНОГО ДЛЯ КОРИСТУВАЧА ІНТЕРФЕЙСУ

### 8.1 Опис станів програми

Стани, в яких може знаходитись програма наведено у табл. 8.1.

Таблиця 8.1 – Повідомлення користувачу

№ стану	Стан	Опис стану	Рекомендовані дії
1	Завантаження додатку	Додаток запускається з браузеру	Почекайте, доки додаток завантажиться
2	Відкрита програма з незаповненими параметрами	Щойно запущений додаток з усіма незаповненими полями	Починайте працювати з додатком
3	Програма у процесі отримання параметрів	Користувач поступово вводить всі параметри	Введіть усі параметри у зручному порядку
4	Програма з заповненими параметрами	Система отримала всі дані, необхідні для роботи	Перейдіть до головного діалогового вікна, та натисніть кнопку «почати»
5	Програма у процесі обробки відеокадрів зі зображенням користувача	Програма обробляє відеокадри та розпізнає образ обличчя, знаходить координати центру правого та лівого ока та їх зіниць. Розраховується відстань зміщення зіниці ока відносно його центру.	Перейдіть до вікна калібрування та натисніть кнопку «почати калібрування». Слідкуйте за переміщенням позиції калібрувальної точки. Намагайтесь не відволікатись, бо це вплине на точність розпізнавання фокусу зору

Продовження табл. 8.1

6	Програма обробляє результати калібрування	Програма зчитує дані з csv файлу та обробляє їх. Програма аналізує отримані дані та знаходить залежність між екранними координатами та переміщенням зіниці ока	Почекайте, доки завершиться цей етап. Далі натисніть кнопку «перейти до тестів»
7	Програма відображає інформаційний контент	Програма розраховує координати фокусу зору користувача та шукає відповідності між інформаційним контентом на екрані.	Пройдіть запропоновані тести

## 8.2 Опис керування діалогом

Додаток «Аналіз графічних інтерфейсів за допомогою трекінгу очей» має декілька діалогових вікон. Головне вікно з'являється після запуску програми (рис. 8.1)., натиснувши на кнопку «почати» ми перейдемо до калібрування та побачимо інструкцію керування додатком. Вікно «Калібрування» (рис. 8.2) дозволяє налаштовувати та проводити процес калібрування.

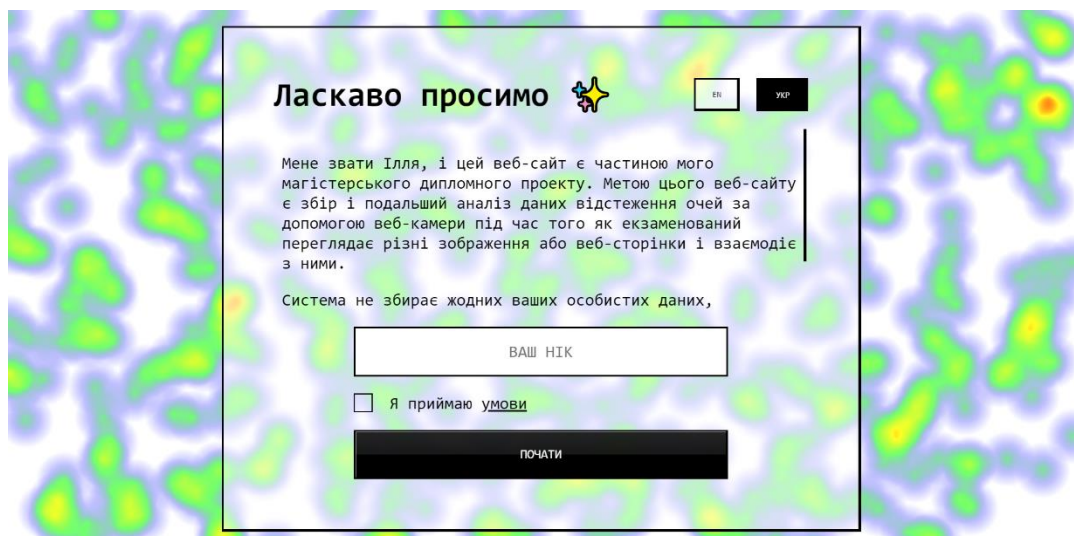


Рисунок 8.1 Головне вікно додатку

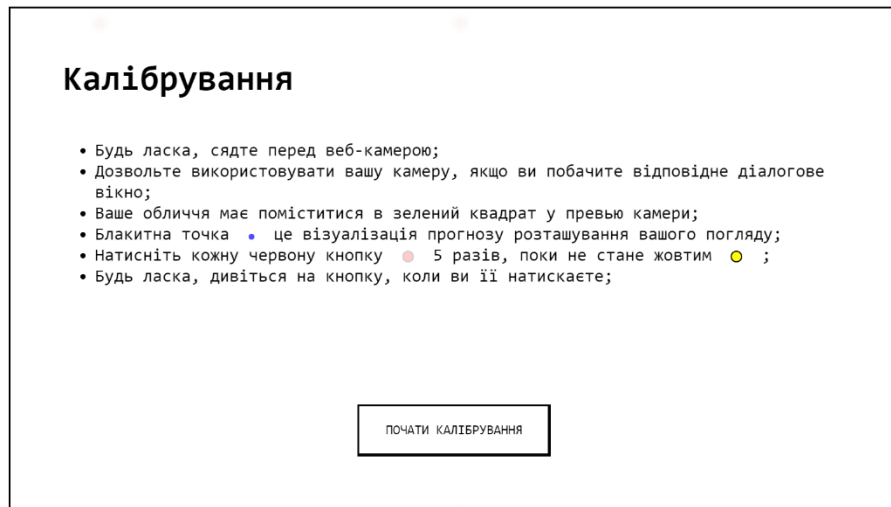


Рисунок 8.2 Вікно початку калібрування

Далі проходить сам процес калібрування який складається з декількох етапів. Після чого ми отримуємо свою міру точності (рис. 8.3). Якщо ця міра задовольняє умови додатку то ми можемо перейти до тестів. Після їх проходження ми отримаємо результати тестування у вигляді теплових карт погляду (рис. 8.4).

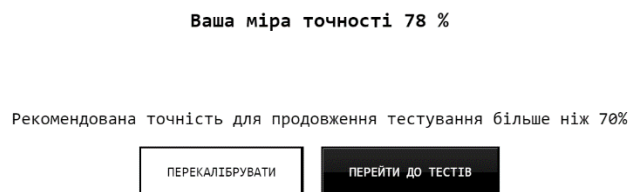


Рисунок 8.3 Вікно міри точності за результатами калібрування

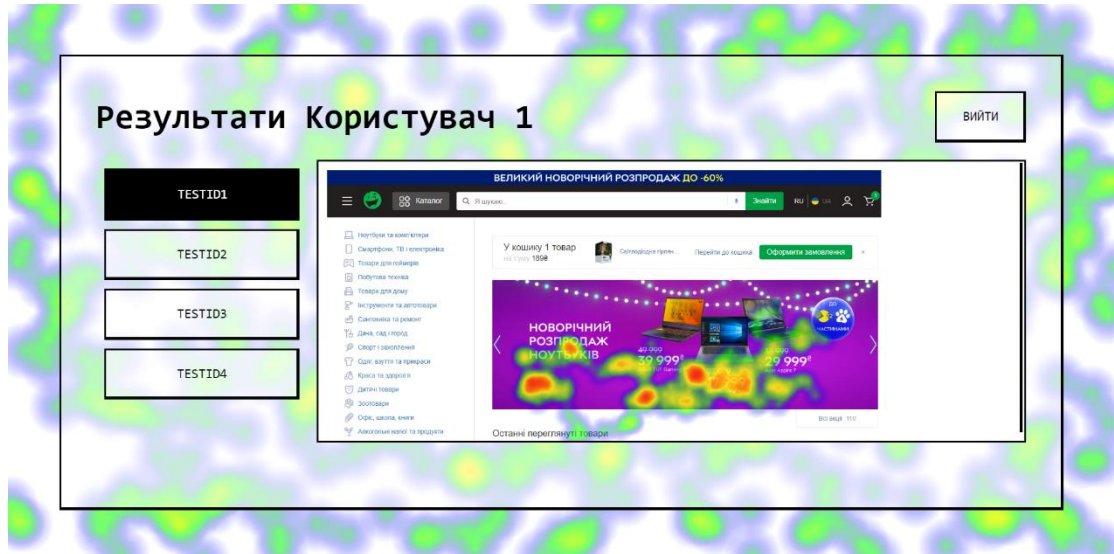


Рисунок 8.4 Теплова карта результату тестування

## **9 ПОРЯДОК РОБОТИ З ПРОГРАМОЮ**

1. Відкрити сторінку за посиланням <https://gaze-analytic.web.app/>
2. Ввести псевдонім, підтвердити погодження з умовами
3. Запустити калібрування системи за інструкціями на екрані
4. Пройти запропоновані тести за інструкціями на екрані
5. Ознайомитись з результатами

## 10 ПОВІДОМЛЕННЯ

У табл. 10.1 наведені повідомлення користувачу, що можуть з'явитись у процесі роботи з програмою.

Таблиця 10.1 – Повідомлення користувачу

Текст повідомлення	Опис ситуації	Рекомендовані дії
Будь ласка увімкніть камеру	Веб-камера не підключена	Перевірте підключення веб-камери
Некоректний формат псевдоніму або його відсутність	Не заповнене поле псевдоніма або заповнене у хибному форматі	Введіть/відкоригуйте текст у полі псевдоніма
Відсутність підтвердження умовами користування	Не активовано чекбокс погодження з умовами користування	Активуйте чекбокс погодження з умовами користування
Тест успішно пройдено	Користувач успішно пройшов тест	Перейти до наступного
Ви пройшли останній тест	Користувач успішно пройшов останній тест	Перейти до результатів або повернуть до головної сторінки

ЗАТВЕРДЖЕНО  
1116130.01285-01

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Керівництво користувача

1116130.01285-01  
Аркушів 13

## **АНОТАЦІЯ**

Документ 01116130.01285-01 ІЗ 01 «Аналіз графічних інтерфейсів за допомогою трекінга очей» входить до складу програмної документації до програми. Програмний продукт дозволяє розпізнавати фокус зору користувача, шляхом обробки даних, отриманих з веб-камери та шукає відповідність отриманих координат фокусу зору з інформації на моніторі.



## ЗМІСТ

ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ .....	5
2 ПІДГОТОВКА ДО РОБОТИ .....	6
3 ОПИС ОПЕРАЦІЙ.....	7
3.1 Реєстрація респондента.....	7
3.2 Калібрування .....	8
3.3 Вимірювання точності .....	9
3.4 Проходження тестів .....	10
3.5 Перегляд результатів.....	11
4 АВАРІЙНІ СИТУАЦІЇ.....	12
5 РЕКОМЕНДАЦІЇ ЩОДО ЗАСВОЄННЯ.....	13

## **ВСТУП**

Аналіз графічних інтерфейсів за допомогою трекінгу очей дозволяє зробити висновки щодо необхідності редагування інформаційного контенту розміщеного на сайтах, для покращення концентрації уваги користувача та легкості сприйняття інформації розміщеної на сторінці .

Користувачі додатку аналізу графічних інтерфейсів за допомогою трекінгу очей дослідники, які займаються вивченням поведінки користувачів під час відвідування веб-сторінок та взаємодії з ними. Користувачі системи повинні мати досвід роботи з ПК.

## **1 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ**

Додаток для аналізу графічних інтерфейсів за допомогою трекінгу очей призначений для відстеження відповідності фокусу зору, для дослідження поведінки користувача під час відвідування веб-сторінок.

Результат дослідження дозволить зрозуміти, на які елементи графічного інтерфейсу зосереджено більше уваги, а які залишилися поза увагою, скільки часу користувач затримався на окремому елементі, тощо.

Дану інформацію можна використовувати для налаштування та оптимізації веб-сторінок, сайтів, додатків, реклами, тощо. Умови застосування додатку:

- наявність веб-камери
- доступ до Інтернету
- ПК або ноутбук
- монітор з роздільною здатністю 1366x768;
- маніпулятор «миша» та клавіатуру;

## **2 ПІДГОТОВКА ДО РОБОТИ**

Для запуску додатку «Аналіз графічних інтерфейсів за допомогою трекінгу очей» необхідно перейти по посиланню <https://gaze-analytic.web.app/>.

### 3 ОПИС ОПЕРАЦІЙ

Керівництво додатку складається з наступних операцій:

- реєстрація респондента;
- процес калібрування;
- вимірювання точності калібрування;
- проходження тестів;
- перегляд результатів.

#### 3.1 Реєстрація респондента

Робота з додатком для аналізу графічних інтерфейсів за допомогою трекінгу очей прочитається з головного вікна у якому необхідно заповнити дані для подальшого користування Рисунок 3.1. Також тут можна ознайомитися з переліком умов користування та змінити мову додатка.

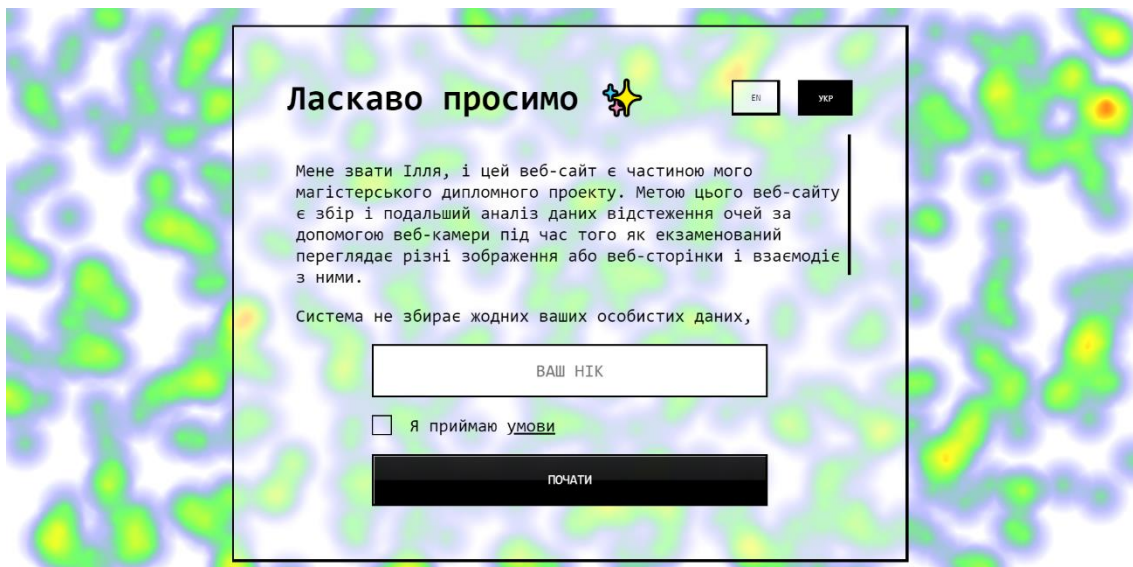


Рисунок 3.1 Головне вікно додатку

Далі необхідно ознайомитися з інструкцією з користування додатком Рисунок 3.2.



Рисунок 3.2 Інструкція перед початком роботи з додатком

### 3.2 Калібрування

Наступним етапом є калібрування. Воно забезпечить точність тестів та досліджень. Також у вікні ми бачимо інструкцію у якій пояснюється що додаток буде використовувати вашу камеру. Учасник повинен розмістити камеру так, щоб його обличчя могло поміститися у зелений квадрат прев'ю камери, це забезпечить точність калібрування. Вікно «Калібрування» (Рисунок 3.3) дозволяє налаштовувати та проводити процес калібрування.

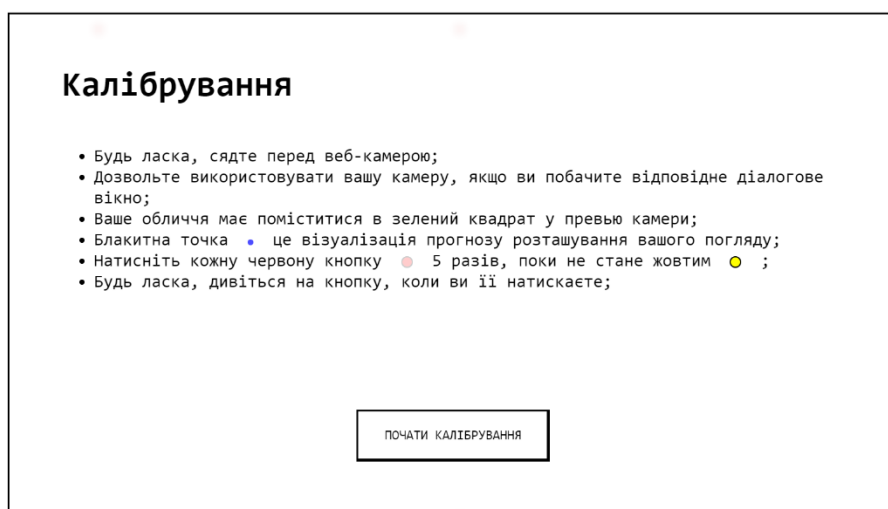


Рисунок 3.3 Вікно початку калібрування

Також прокрутивши до низу ми бачимо інструкцію щодо калібрування, окремо знизу також є кнопка «почати калібрування». Процес калібрування зазначений на Рисунку 3.4. Його суть ці співпадіння погляду респондента та положення його миші протягом короткого проміжку часу, що допомагає системі підлаштуватись під кожний окремий випадок тестування.

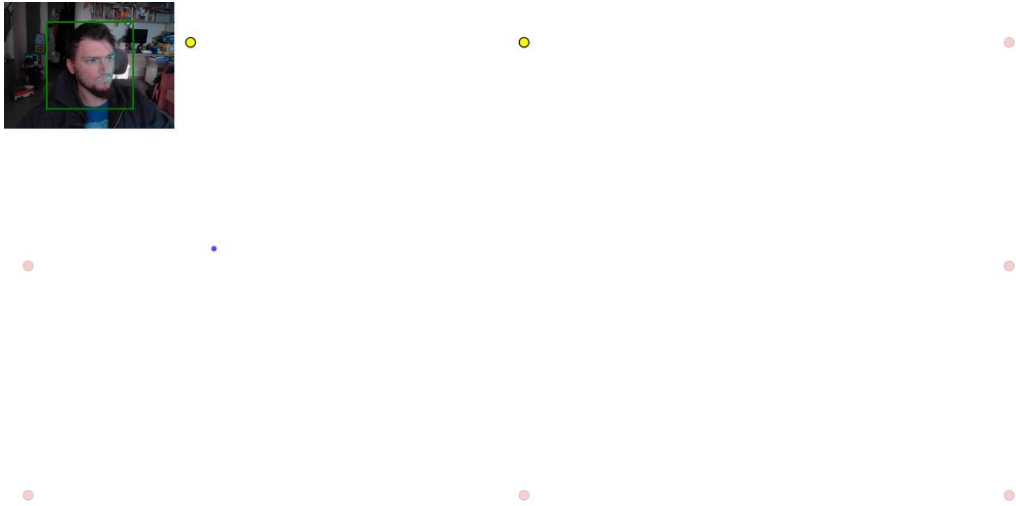


Рисунок 3.4 Вікно процесу калібрування

### 3.3 Вимірювання точності

Після калібрування ми отримуємо міру точності яка повинна становити не менше 70%, а також бачимо дві кнопки «перекалібрувати» на випадок низької точності та «перейти до тестів» Рисунок 3.5.

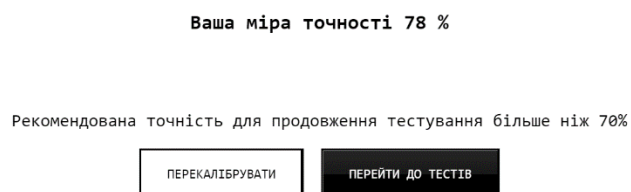


Рисунок 3.5 Вікно міри точності

### 3.4 Проходження тестів

Далі ми побачимо два вікно з тестами які будуть іти одне після другого (Рисунок 3.6), перед кожним тестом ми маємо інструкцію щодо його проходження. Після тестів з'явиться вікно з можливістю або отримати результати тестів (рис. 3.7), що являють собою теплові карти (Рисунок 3.8, 3.9), або завершити роботу без результатів.

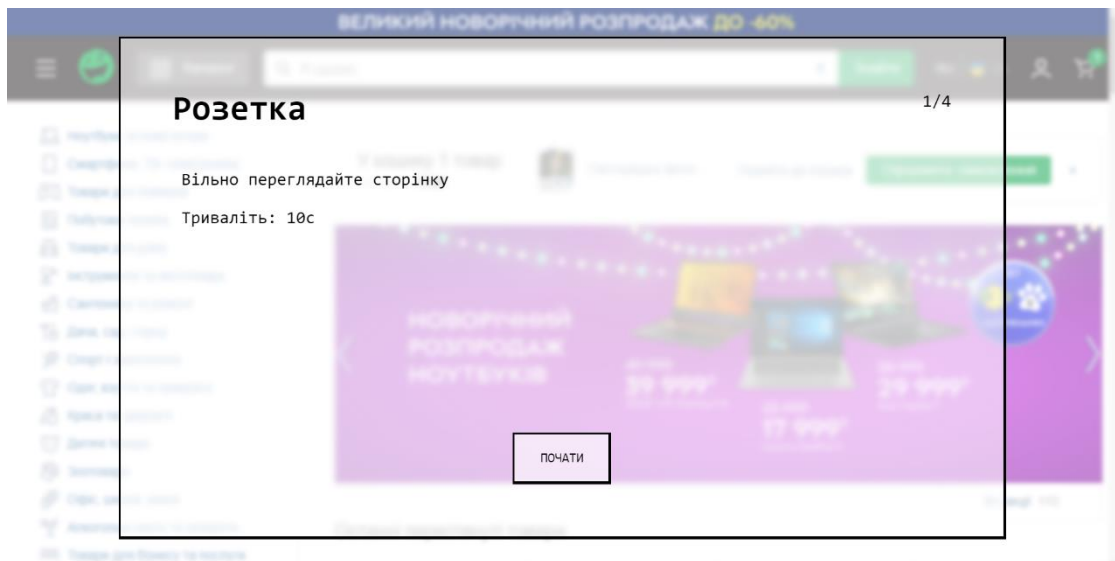


Рисунок 3.6 Вікно тесту.

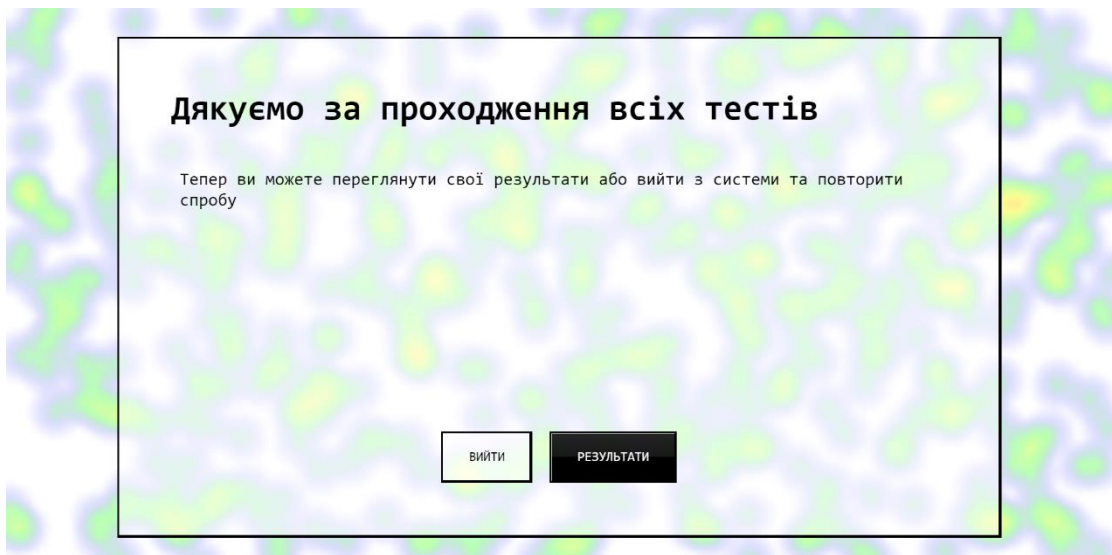


Рисунок 3.7 Вікно переходу до результатів або завершення роботи.



### 3.5 Перегляд результатів

Респондент при бажанні, може подивитись на результати пройдених їм тестів.

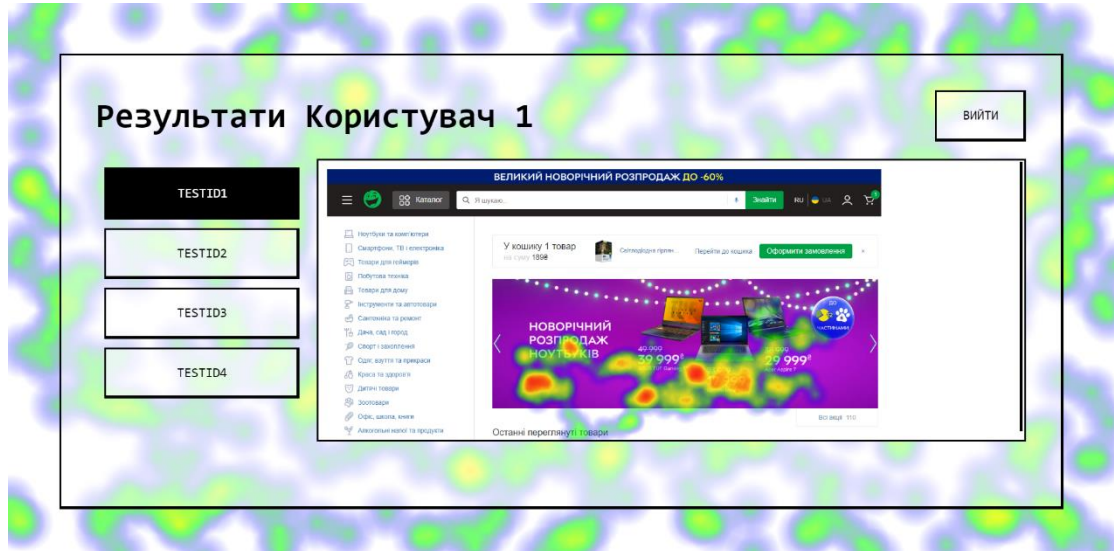


Рисунок 3.8 Теплові карти тестів.

#### **4 АВАРІЙНІ СИТУАЦІЇ**

Під час експлуатації додатку для аналізу графічних інтерфейсів за допомогою трекінгу очей можуть виникати аварійні ситуації. Якщо під час роботи системи буде відсутній зв'язок з інтернетом, спробуйте перезавантажити сторінку, усунути всі можливі варіанти які можете перевірити самостійно, якщо це не допоможе, зверніться до технічної підтримки.

## **5 РЕКОМЕНДАЦІЇ ЩОДО ЗАСВОЄННЯ**

Під час проведення процесу калібрування, намагайтеся не відводити свій погляд від позиції калібрувальної точки, адже це може вплинути на результат розпізнавання координат фокусу зору. Щоб впевнитися, що система працює коректно, переконайтесь, що система відображає прев'ю вашої камери та відображає мітку знаходження координат фокусу зору. Мітка представлена у вигляді синьої крапки (рис. 3.4).

ЗАТВЕРДЖЕНО  
1116130.01285-01

АНАЛІЗ ГРАФІЧНИХ ІНТЕРФЕЙСІВ  
ЗА ДОПОМОГОЮ ТРЕКІНГА ОЧЕЙ

Текст програми

1116130.01285-01  
Аркушів 48

## **АНОТАЦІЯ**

Документ 01116130.01285-01 12 01 «Аналіз графічних інтерфейсів за допомогою трекінгу очей. Текст програми» входить до складу програмної документації до програми, яка розпізнає фокус зору користувача, шляхом обробки кадрів, отриманих з відеокамери та шукає відповідність отриманих координат фокусу зору зі структурою інформації у графічному інтерфейсі. У даному документі представлений текст програм. Додаток написана на мові React. Програма реалізована у програмному середовищі Github. Додатком можна користуватись у браузері. Конфігурація комп'ютера: процесор Intel Core i5- 3230M, RAM – 4Gb, відеокарта – NVIDIA GeForce GT 740M, монітор з роздільною здатністю 1366x768. Комплекс функціонує в середовищі WINDOWS 10.

## Зміст

1. Схема взаємодії модулів .....	4
2. Текст програми .....	5
2.1 Модуль index.js .....	5
2.2 Модуль constants.js .....	6
2.3 Модуль App.js .....	6
2.4 Модуль utils/precisionUtils.js .....	8
2.5 Модуль utils/resizeUtils.js .....	10
2.6 Модуль hooks/useModal.js .....	11
2.7 Модуль firebase/api.js .....	12
2.8 Модуль pages/calibrationPage.js .....	14
2.9 Модуль pages/instructionPage.js .....	18
2.10 Модуль pages/resultsPage.js .....	19
2.11 Модуль pages/signInPage.js .....	21
2.12 Модуль pages/testsPage.js .....	22
2.13 Модуль pages/welcomePage.js .....	24
2.14 Модуль components/logout.js .....	26
2.15 Модуль components/heatmapViewer.js .....	27
2.16 Модуль components/webGazerLoader.js .....	28
2.17 Модуль components/testSwitcher.js .....	31
2.18 Модуль components/testItem.js .....	32
2.19 Модуль components/signinForm.js .....	34
2.20 Модуль components/modal.js .....	37
2.21 Модуль components/langSwitcher.js .....	39
2.22 Модуль components/button.js .....	40
2.23 Модуль components/checkbox.js .....	41
2.24 Модуль components/scrolbar.js .....	42
2.25 Модуль testScreens/testsCopm.js .....	43
2.26 Модуль locales/en.js .....	45
2.27 Модуль locales /ua.js .....	47

## 1. Схема взаємодії модулів

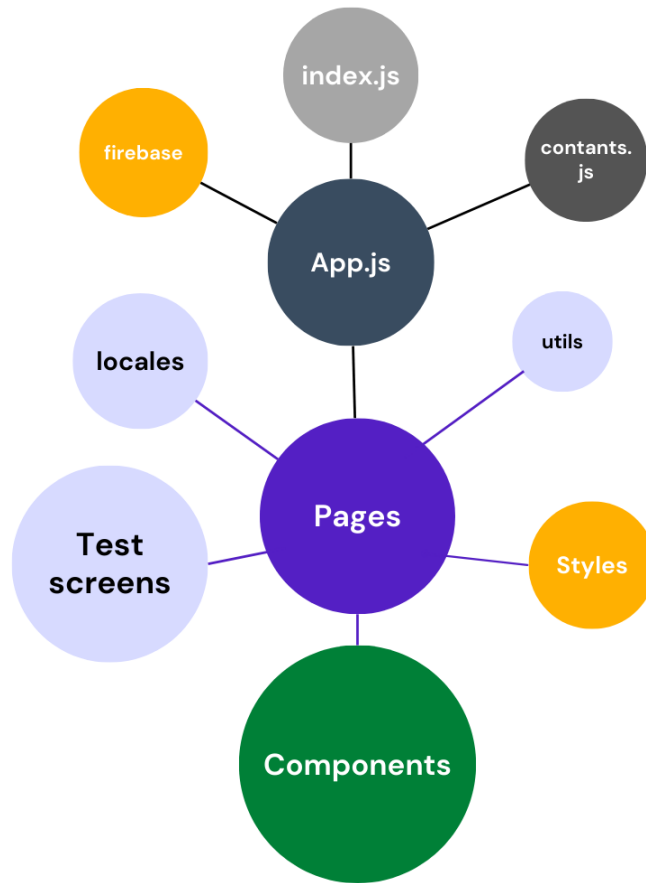


Рисунок 1.1 – Схема взаємодії модулів

## 2. Текст програми

### 2.1 Модуль index.js

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import * as firebase from 'firebase/app';  
import 'firebase/database';  
import config from './firebaseConfig';  
import App from './App';  
import './index.css';  
import './i18n.js';  
  
}
```

```
firebase.initializeApp(config);  
  
const root =  
ReactDOM.createRoot(document.getElementById('root'));  
  
root.render(  
  <App />,  
);
```



## 2.2 Модуль constants.js

```
export const SAVED_USER_NAME =  
'userName';  
export const SAVED_USER_KEY = 'userKey';  
export const SAVED_USER_ACCEPTED_TAC  
= 'userTac';  
export const USER_LANG = 'userLang';  
export const SAVED_USER_IS_CALIBRATED  
= 'isCalibrated';  
export const SAVED_USER_HAS_RESULT =  
'userHasResult';
```

```
export const FIREBASE_USERS_PATH =  
'/users';  
export const FIREBASE_RESULTS_PATH =  
'/results';  
export const PATH_HOME = '/';  
export const PATH_INSTRUCTIONS_PAGE =  
'/instruction';  
export const PATH_TESTS_PAGE = '/tests';  
export const PATH_RESULTS_PAGE =  
'/results';  
export const PATH_SING_IN_PAGE = '/signin';
```

## 2.3 Модуль App.js

```
import React, { useEffect, useState } from
'react';

import { BrowserRouter, Routes, Route,
Navigate, useLocation } from 'react-router-dom';

import SignInPage from
'./pages/signinPage/signInPage';

import ResultsPage from './pages/resultsPage';

import TestPage from
'./pages/testsPage/testsPage';

import {
  PATH_HOME,
  PATH_INSTRUCTIONS_PAGE,
  PATH_RESULTS_PAGE,
  PATH_SING_IN_PAGE,
  PATH_TESTS_PAGE,
  SAVED_USER_KEY,
  SAVED_USER_NAME,
} from './constants';

import HeatmapViewer from
'./components/heatmapViewer';

import InstructionPage from
'./pages/instructionPage';

import WelcomePage from
'./pages/welcomePage';

import generateSampleHeatMap from
'./utils/generateBg';

const App = () => {
  const [randomHeatmap, setRandomHeatmap]
= useState([]);

  useEffect(() => {
    setRandomHeatmap(generateSampleHeatMap())
    ;

    }, []);

  return (
    <div className="App">
      <BrowserRouter>
        <Routes>
          <Route
path={PATH_SING_IN_PAGE}
element={<SignInPage />} />

          <Route path={PATH_HOME}
element={<WelcomePage />} />

          <Route

path={PATH_INSTRUCTIONS_PAGE}

element={
          <RequireAuth>
            <InstructionPage />
          </RequireAuth>
        }
      />

          <Route

path={PATH_TESTS_PAGE}

element={
          <RequireAuth>
            <TestPage />
          </RequireAuth>
        }
      />
        </Routes>
      </BrowserRouter>
    </div>
  );
}
```

<pre> &lt;Route path={PATH_RESULTS_PAGE}   element={     &lt;RequireAuth&gt;       &lt;ResultsPage /&gt;     &lt;/RequireAuth&gt;   } /&gt; &lt;/Routes&gt; &lt;/BrowserRouter&gt; &lt;HeatmapViewer   config={{     radius: 50,     maxOpacity: 0.9,     minOpacity: 0,     blur: 1,   }}   style={{     width: '100vw',     height: '100vh',     position: 'absolute',     top: '0',     left: '0',     zIndex: '-1', </pre>	<pre>     }}     id="backgroundHeatmap"     result={randomHeatmap}   /&gt; &lt;/div&gt; ); };  function RequireAuth({ children }) {   const savedName =     JSON.parse(localStorage.getItem(SAVED_USE R_NAME));    const savedKey =     JSON.parse(localStorage.getItem(SAVED_USE R_KEY));    let location = useLocation();    if (!savedName    !savedKey) {     return &lt;Navigate to={PATH_HOME} state={{ from: location }} replace /&gt;;   }    return children; }  export default App; </pre>
--	--

## 2.4 Модуль utils/precisionUtils.js

```
/* * This function calculates a measurement for
how precise
* the eye tracker currently is which is displayed
to the user
*/

export function calculatePrecision(past50Array)
{
    var windowHeight = window.innerHeight;
    var windowWidth = window.innerWidth;

    // Retrieve the last 50 gaze prediction points
    var x50 = past50Array[0];
    var y50 = past50Array[1];

    // Calculate the position of the point the user is
    staring at
    var staringPointX = windowWidth / 2;
    var staringPointY = windowHeight / 2;

    var precisionPercentages = new Array(50);

    calculatePrecisionPercentages(precisionPercenta
ges, windowHeight, x50, y50, staringPointX,
staringPointY);

    var precision =
calculateAverage(precisionPercentages);

    // Return the precision measurement as a
rounded percentage
    return Math.round(precision);
};
```

```
/*
* Calculate percentage accuracy for each
prediction based on distance of
* the prediction point from the centre point
(uses the window height as
* lower threshold 0%)
*/

export function
calculatePrecisionPercentages(precisionPercenta
ges, windowHeight, x50, y50, staringPointX,
staringPointY) {
    for (let x = 0; x < 50; x++) {
        // Calculate distance between each prediction
and staring point
        var xDiff = staringPointX - x50[x];
        var yDiff = staringPointY - y50[x];
        var distance = Math.sqrt((xDiff * xDiff) +
(yDiff * yDiff));

        // Calculate precision percentage
        var halfWindowHeight = windowHeight / 2;
        var precision = 0;

        if (distance <= halfWindowHeight &&
distance > -1) {
            precision = 100 - (distance /
halfWindowHeight * 100);
        } else if (distance > halfWindowHeight) {
            precision = 0;
        } else if (distance > -1) {
            precision = 100;
        }
    }
}
```

```
// Store the precision
precisionPercentages[x] = precision;
}
}

/*
 * Calculates the average of all precision
 * percentages calculated
 */
}
```

```
export function
calculateAverage(precisionPercentages) {
  var precision = 0;
  for (let x = 0; x < 50; x++) {
    precision += precisionPercentages[x];
  }
  precision = precision / 50;
  return precision;
}
```

## 2.5 Модуль utils/resizeUtils.js

```
function resize() {  
  console.log('resize works');  
  var canvas =  
document.getElementById('plotting_canvas');  
  var context = canvas.getContext('2d');  
  context.clearRect(0, 0, canvas.width,  
canvas.height);  
  canvas.width = window.innerWidth;  
  canvas.height = window.innerHeight;  
  };  
window.addEventListener('resize', resize, false);
```

## 2.6 Модуль hooks/useModal.js

<pre>import { useState } from 'react';  const useModal = (default) =&gt; {   const [isShowing, setIsShowing] = useState(default    false);    const toggle = (value) =&gt; {     value ? setIsShowing(value) : setIsShowing(!isShowing);</pre>	<pre>};    return {     isShowing,     toggle,   }; };  export default useModal;</pre>
--	--

## 2.7 Модуль firebase/api.js

```
import { getDatabase, ref, set, push, update,
child, get } from 'firebase/database';
import { FIREBASE_RESULTS_PATH,
FIREBASE_USERS_PATH } from '../constants';
```

```
const createUser = async (name, callback) => {
  const db = getDatabase();
  try {
    const newUserKey = await
push(child(ref(db),
FIREBASE_USERS_PATH)).key;
    const valueToSubmit = { name: name,
results: " " };
    set(ref(db,
`${FIREBASE_USERS_PATH}/${newUserKey
}`), valueToSubmit).then(
      callback(newUserKey),
    );
  } catch (e) {
    console.log('firebase api error', e);
  }
};
```

```
const updateUserData = async (id, data,
callback) => {
  const db = getDatabase();

  try {
    const newResultKey = push(child(ref(db),
'results')).key;
    const valueToSubmit = {

[`${FIREBASE_USERS_PATH}/${id}/results`]
: data,

[`${FIREBASE_RESULTS_PATH}/${newResu
ltKey}`]: data,
    };
    await update(ref(db),
valueToSubmit).then(callback());
  } catch (e) {
    console.log('firebase api error', e);
  }
};
```

```
const pushUserResult = async (userId, data,
callback) => {
  const db = getDatabase();

  try {
    const newResultKey = push(child(ref(db),
'results')).key;
    const valueToSubmit = {

[`${FIREBASE_USERS_PATH}/${userId}/resu
lts/${newResultKey}`]: data,

[`${FIREBASE_RESULTS_PATH}/${newResu
ltKey}`]: data,
    };
    await update(ref(db),
valueToSubmit).then(callback());
  } catch (e) {
    console.log('firebase api error', e);
  }
};
```

```
const getUserData = async (id, callback) => {
  const dbRef = ref(getDatabase());
  let res;
  try {
    get(child(dbRef,
`${FIREBASE_USERS_PATH}/${id}${FIREB
ASE_RESULTS_PATH}')).then(
      async (snapshot) => {
        if (snapshot.exists()) {
          res = await snapshot.val();
          callback(res);
        } else {
          console.log('No data available');
        }
      },
    );
  } catch (e) {
    console.log('firebase api error', e);
  }
  return res;
};
```



```
export { createUser, updateUserData,  
getUserData, pushUserResult };
```

## 2.8 Модуль pages/calibrationPage.js

```
import React, { useEffect, useState } from
'react';
import cn from 'classnames';
import { calculatePrecision } from
'../../utils/precisionUtils';
import Modal from
'../../components/modal/modal';
import useModal from ' ../../hooks/useModal';
import Button from
'../../components/button/button';
import gs from ' ../../styles/global.module.css';
import s from './calibrationPage.module.css';
import './style.css';
import { useTranslation } from 'react-i18next';

const Calibration = ({ onSuccess, start, stop })
=> {
  const defaultCalibrationButtons = {
    id1: { pressedTimes: 0, passed: false },
    id2: { pressedTimes: 0, passed: false },
    id3: { pressedTimes: 0, passed: false },
    id4: { pressedTimes: 0, passed: false },
    id5: { pressedTimes: 0, passed: false },
    id6: { pressedTimes: 0, passed: false },
    id7: { pressedTimes: 0, passed: false },
    id8: { pressedTimes: 0, passed: false },
    id9: { pressedTimes: 0, passed: false },
  };
  const [step, setStep] = useState('1');
  const [calibrationPoints, setCalibrationPoints]
= useState(defaultCalibrationButtons);
  const [pointCalibrate, setPointCalibrate] =
useState(0);
  const [accuracy, setAccuracy] = useState(0);
  const idReadyToMeasure = step === '2';
  const { t } = useTranslation();

  const { isShowing: isShowing1, toggle:
toggle1 } = useModal(true);
  const { isShowing: isShowing2, toggle:
toggle2 } = useModal(true);

  const store_points_variable = () => {
```

```
    window.webgazer.params.storingPoints =
true;
  };
  const stop_storing_points_variable = () => {
    window.webgazer.params.storingPoints =
false;
  };

  function sleep(time) {
    return new Promise((resolve) =>
setTimeout(resolve, time));
  }

  const incrementPressedTimes = (id, prev) =>
{
    return { ...prev, [id]: { ...prev[id],
pressedTimes: prev[id].pressedTimes + 1 } };
  };

  const setPassed = (id, prev) => ({ ...prev, [id]:
{ ...prev[id], passed: true } });

  useEffect(() => {
    for (const btnInx in calibrationPoints) {
      const btn = calibrationPoints[btnInx];
      if (btn.pressedTimes === 5 &&
!btn.passed) {
        setCalibrationPoints((prev) =>
setPassed(btnInx, prev));
        setPointCalibrate((prev) => ++prev);
      }
    }
    if (pointCalibrate === 8) {
      setStep('2');
    }
  }, [calibrationPoints, pointCalibrate]);

  useEffect(() => {
    if (idReadyToMeasure) {
      window.webgazer.showVideo(false);

      stop();
    }
  })
```

```
// eslint-disable-next-line react-
hooks/exhaustive-deps
}, [step, idReadyToMeasure]);

const precisionMeasurement = () => {
  start();
  //
window.webgazer.removeMouseEventListeners(
);
  store_points_variable();
  sleep(5000).then(() => {
    stop_storing_points_variable(); // stop
    storing the prediction points
    const past50 =
window.webgazer.getStoredPoints(); // retrieve
the stored points
    const precision_measurement =
calculatePrecision(past50);
    console.log({ precision_measurement });
    stop();
    setAccuracy(precision_measurement);
    setStep('3');

window.webgazer.hidePredictionPoints();
  });
};

const onCalibrationButtonClick = (e) => {
  const id = e.target.id;
  setCalibrationPoints((prev) =>
incrementPressedTimes(id, prev));
};

const recalibrate = () => {
  toggle1();
  toggle2();

setCalibrationPoints(defaultCalibrationButtons);
  setPointCalibrate(0);
  // window.webgazer.clearData();
  setAccuracy(0);
  setStep('1');
};

const renderButtons = () => {
```

```
return
Object.keys(calibrationPoints).map((btnInx) =>
{
  const btn = calibrationPoints[btnInx];
  const opacity = (btn.pressedTimes + 1) *
0.2;

  const is5thVisible =
    (btnInx !== 'id5' &&
!idReadyToMeasure) || (btnInx === 'id5' &&
idReadyToMeasure);
  return (
    <button
      id={ btnInx }
      key={ btnInx }
      className={ cn(s.calibrationBtn,
btn.passed && s.passed, s[`btn-${btnInx}`]) }
      style={{ opacity: opacity, visibility:
is5thVisible ? 'visible' : 'hidden' }}

onClick={ onCalibrationButtonClick }
      disabled={ btn.passed }
    ></button>
  );
});

const calibrationDescription = (
  <ul>
    <li>{t('calibration-instruction-1')}</li>
    <li>{t('calibration-instruction-2')}</li>
    <li>{t('calibration-instruction-3')}</li>
    <li>
      {t('calibration-instruction-4-1')}
      &nbsp;
      <div className={ s.blueDot } />
      &nbsp;
      {t('calibration-instruction-4-
2')}
    </li>
    <li>
      {t('calibration-instruction-5-1')}
      &nbsp;
      <button
        className={ cn(s.calibrationBtn,
s.btnText)}
        style={{ opacity: '0.2' }}
        disabled
```

bodyContent={ measurementDescription }

label') }

id5`]) }></b>

$$\}} > \{ \backslash \$ \{ t ($$

<div

```
className={gs.flexWrapperRowCenter}>
```

onClick={recalibr}

 $);$ 

```
switch (step) {
```

case '2':

```
return secondStep();
```

case '3':

```
return thirdStep();
```

default:

return firstStep();

}

};

|

export default Calibration;

## 2.9 Модуль pages/instructionPage.js

```

import classNames from 'classnames';
import React from 'react';
import { useTranslation } from 'react-i18next';
import { useNavigate } from 'react-router-dom';
import Button from '../components/button';
import Scrollbar from
'../components/scrollbar';
import { PATH_TESTS_PAGE,
SAVED_USER_IS_CALIBRATED } from
'../constants';
import gs from '../styles/global.module.css';
import s from './instructionPage.module.css';

const InstructionPage = () => {
  const navigate = useNavigate();
  const { t } = useTranslation();
  const isCalibrated =
JSON.parse(localStorage.getItem(SAVED_USE
R_IS_CALIBRATED));
  return (
    <div
      className={gs.flexFullScreenCenterCenter}>
      <div className={classNames(gs.box,
s.box)}>
        { /* <div className={s.boxInner}> */}

```

```

<h1>{t('instruction-title')}</h1>
<Scrollbar className={s.text}>
  <p>{t('instruction-p1')}</p>
  <p>{t('instruction-p2')}</p>
  <p>{t('instruction-p3')}</p>
  <ul>
    <li>{t('instruction-li1')}</li>
    <li>{t('instruction-li2')}</li>
    <li>{t('instruction-li3')}</li>
  </ul>
  <p>{t('instruction-p4')}</p>
</Scrollbar>
  <Button onClick={() =>
navigate(PATH_TESTS_PAGE)}>
    {isCalibrated
      ? t('instruction-to-tests-label')
      : t('instruction-calibration-label')}
  </Button>
</div>
</div>
  );
};

export default InstructionPage;

```

## 2.10 Модуль pages/resultsPage.js

```
import React, { useEffect, useState } from
'react';
import HeatmapViewer from
'./../components/heatmapViewer';
import Spinner from './../components/spinner';
import { SAVED_USER_KEY,
SAVED_USER_NAME } from './../constants';
import { getUserData } from './../firebase/api';
import genTests from
'./../testScreens/testsCopr';
import Logout from './../components/logout';
import { useNavigate } from 'react-router-dom';
import s from './resultsPage.module.css';
import gs from './../styles/global.module.css';
import classNames from 'classnames';
import Button from './../components/button';
import { useTranslation } from 'react-i18next';
import Scrollbar from
'./../components/scrollbar';

const ResultsPage = () => {
  const [isLoading, setLoading] =
useState(true);
  const savedName =
JSON.parse(localStorage.getItem(SAVED_USE
R_NAME));
  const savedKey =
JSON.parse(localStorage.getItem(SAVED_USE
R_KEY));
  const [result, setResult] = useState(null);
  const [currentResultId, setCurrentResultId] =
useState(null);
  const navigate = useNavigate();
  const { t } = useTranslation();

  useEffect(() => {
    getUserData(savedKey, (res) => {
      setResult(res);
      setLoading(false);
    });
    // eslint-disable-next-line react-
hooks/exhaustive-deps
  }, []);
```

```
console.log('firebase result', result);

const idArr = result ? Object.keys(result) : [];

useEffect(() => {
  if (result)
setCurrentResultId(Object.keys(result)[0]);
}, [result]);

useEffect(() => {
  if (result && currentResultId) {
    // setTestedComponent(
    //   tests.find((t) => t.id ===
result[currentResultId].testName).testComponent
    ,
    // );
  }
}, [currentResultId, result]);

const tests = genTests();

if (isLoading) return <Spinner />;
return (
  <div
className={classNames(s.pageWrapper)}>
    <div
className={classNames(s.pageContent,
gs.box)}>
      <div className={s.header}>
        <h1>{`${t('results-title')}
${savedName}`}</h1>
        <Button
onClick={() => {
  Logout(navigate);
}}
>
        {t('results-logout-label')}
      </Button>
    </div>
    {result && (
      <div className={s.accordion}>
        <div
className={s.buttonsWrapper}>
          <Scrollbar>
```

```

      <div
className={s.buttons}>
      {idArr.map((id) => (
        <Button
puched={currentResultId === id}
        key={id}
        onClick={() =>
setCurrentResultId(id)}
        >
        {result[id].testName}
        </Button>
      )}}
    </div>
  </Scrollbar>
</div>
{currentResultId && (
  <Scrollbar
className={s.content}>
    <div
      style={{
        transformOrigin: 'top
left',
        transform: 'scale(.5)',
        width:
result[currentResultId].screenSize.width,
        height:
result[currentResultId].screenSize.height,
        position: 'absolute',
        padding: '0',
      }}
    >
      <div
className={s.testetUi}>
        {
          tests.find(
            (t) => t.id ===
result[currentResultId].testName,
          ).testComponent
        }
      </div>
    </div>
  )}
</div>

```

```

    </div>

    <HeatmapViewer
className={s.heatmapViewer}
      key={currentResultId +
'key'}
result={result[currentResultId].result}
      config={{
        radius: 50,
        maxOpacity: 0.9,
        minOpacity: 0,
        blur: 0.95,
      }}
    />
  </div>
</Scrollbar>
)}
</div>
</div>
{ /* <HeatmapViewer
id="backgroundHeatmap" result={} /> */}
{ /* {result && currentResultId && (
  <HeatmapViewer
    id="backgroundHeatmap"

className={s.backgroundHeatmap}
      key={currentResultId + 'key'}

result={result[currentResultId].result}
    />
  )} */}
</div>
);
};

export default ResultsPage;

```



## 2.11 Модуль pages/signInPage.js

```
import classNames from 'classnames';
import React, { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import SignInForm from
'../../components/signinForm';
import { PATH_HOME, SAVED_USER_KEY,
SAVED_USER_NAME } from '../../constants';
import gs from '../../styles/global.module.css';
// import s from './signInPage.module.css';

const SignInPage = () => {
  const savedName =
JSON.parse(localStorage.getItem(SAVED_USE
R_NAME));
  const savedKey =
JSON.parse(localStorage.getItem(SAVED_USE
R_KEY));

  const navigate = useNavigate();
```

```
  useEffect(() => {
    if (savedName && savedKey) {
      navigate(PATH_HOME, { replace: true
    });
    }
    // eslint-disable-next-line react-
hooks/exhaustive-deps
    }, [savedName, savedKey]);

    return (
      <div
        className={gs.flexFullScreenCenterCenter}>
        <SignInForm
          className={classNames(gs.flexWrapperRowCe
            nter, gs.box)} />
        </div>
      );
    );
  };

  export default SignInPage;
```

## 2.12 Модуль pages/testsPage.js

```
import React, { useEffect, useMemo } from
'react';
import { useNavigate } from 'react-router-dom';
import Modal from '../components/modal';
import TestItem from '../components/testItem';
import TestSwitcher from
'../components/testSwitcher';
import { WebGazerLoader } from
'../components/webGazerLoader';
import {
  PATH_RESULTS_PAGE,
  SAVED_USER_IS_CALIBRATED,
  SAVED_USER_KEY,
  SAVED_USER_HAS_RESULT,
} from '../constants';
import { pushUserResult } from
'../firebase/api';
import useModal from '../hooks/useModal';
import Logout from '../components/logout';
import genTests from
'../testScreens/testsCopr';
import { useTranslation } from 'react-i18next';
// require('../utils/resizeUtils');

const TestPage = () => {
  const savedKey =
JSON.parse(localStorage.getItem(SAVED_USE
R_KEY));
  const hasResult =
JSON.parse(localStorage.getItem(SAVED_USE
R_HAS_RESULT));
  const navigate = useNavigate();
  const { isShowing, toggle } =
useModal(false);
  const { t } = useTranslation();

  useEffect(() => {
    if (hasResult) {
      toggle(true);
    }
    // eslint-disable-next-line react-
hooks/exhaustive-deps
  }, [hasResult]);
```

```
const pushResult = async (testItemResult) =>
{
  console.log({ testItemResult });
  //
  localStorage.setItem(SAVED_USER_RESULT,
JSON.stringify({ ...savedResult, testItemResult
}));
  await pushUserResult(savedKey,
testItemResult, () => console.log('Saved on
firebase'));
};

const setUserIsCalibrated = () => {

  localStorage.setItem(SAVED_USER_IS_CALI
BRATED, true);
};

const finishTesting = () => {

  localStorage.setItem(SAVED_USER_HAS_RES
ULT, JSON.stringify(true));
  toggle(true);
};

const tests = genTests();

return (
  <>
    <WebGazerLoader
      setUserIsCalibrated={setUserIsCalibrated}
      competent={TestItem}>
      {!hasResult
        ? (sessionResult, additionalProps) =>
          (
            <TestSwitcher
              tests={tests}
              {...additionalProps}
              finishTesting={finishTesting}
              processResult={
                (id,
                  pushResult({ testName: id,
                    result: sessionResult, screenSize })
                )
              }
            </TestSwitcher>
          )
        : sessionResult
      }
    </WebGazerLoader>
  </>
)
```

<pre>        /&gt;       )       : null}     &lt;/WebGazerLoader&gt;     &lt;Modal       isShowing={isShowing}       action={[() =&gt; Logout(navigate), () =&gt; navigate(PATH_RESULTS_PAGE)]}       buttonLabel={[t('thanks-logout-label'), t('thanks-results-label')]}     &lt;/Modal&gt;</pre>	<pre>      hide={toggle}       header={t('thanks-title')}       bodyText={t('thanks-text')}     &lt;/Modal&gt;   ); };  export default TestPage;</pre>
--	--

## 2.13 Модуль pages/welcomePage.js

```
import classNames from 'classnames';
import React, { useEffect, useState } from
'react';
import { useNavigate } from 'react-router-dom';
import SignInForm from
'../../components/signinForm';
import { PATH_INSTRUCTIONS_PAGE,
SAVED_USER_KEY, SAVED_USER_NAME
} from '../../constants';
import gs from '../../styles/global.module.css';
import s from './welcomePage.module.css';
import { useTranslation } from 'react-i18next';
import LangSwitcher from
'../../components/langSwitcher';
import Button from '../../components/button';
import Scrollbar from
'../../components/scrollbar';
import Modal from '../../components/modal';
```

```
const WelcomePage = () => {
  const savedName =
JSON.parse(localStorage.getItem(SAVED_USE
R_NAME));
  const savedKey =
JSON.parse(localStorage.getItem(SAVED_USE
R_KEY));
  const [isShowing, toggle] = useState(false);
  const navigate = useNavigate();
  const { t } = useTranslation();

  useEffect(() => {
    if (savedName && savedKey) {

navigate(PATH_INSTRUCTIONS_PAGE, {
replace: true });
    }
    // eslint-disable-next-line react-
hooks/exhaustive-deps
  }, [savedName, savedKey]);
  const tacContent = (
    <div>
      <p>{t('toc-body-p1')}</p>
      <p>{t('toc-body-p2')}</p>
      <p>
```

```
      <a target="_blank" rel="noreferrer"
href="https://webgazer.cs.brown.edu/">
        {t('toc-lib-web-link')}
      </a>
    </p>
    <p>
      <a target="_blank" rel="noreferrer"
href="https://github.com/brownhci/WebGazer">
        {t('toc-lib-git-link')}
      </a>
    </p>
    <p>
      <a target="_blank" rel="noreferrer"
href="github.com">
        {t('toc-project-link')}
      </a>
    </p>
  </div>
);
  return (
    <div
      className={gs.flexFullScreenCenterCenter}>
      <div className={classNames(gs.box,
s.box)}>
        <div className={s.header}>
          <h1>{t('welcome-
header')} ✨</h1>
          <LangSwitcher
            className={s.lang} />
        </div>
        <Scrollbar>
          <p>{t('welcome-body-p1')}</p>
          <p>
            {t('welcome-body-p2')}
            <Button onClick={() =>
toggle(true)} type="link">
              {t('welcome-body-btn-
label')}
            </Button>
          </p>
        </Scrollbar>
```

```
      <SignInForm
nextPath={PATH_INSTRUCTIONS_PAGE}
className={s.form} />
    </div>
  </div>
  <Modal
    header={t('toc-lib-header')}
    bodyContent={tacContent}
    isShowing={isShowing}
```

```
      hide={toggle}
      action={() => {}}
      buttonLabel={t('ok-label')}
    />
  </>
);
};

export default WelcomePage;
```

## 2.14 Модуль components/logout.js

```
import {
  SAVED_USER_IS_CALIBRATED,
  SAVED_USER_KEY,
  SAVED_USER_NAME,
  SAVED_USER_HAS_RESULT,
  PATH_HOME,
} from '../constants';

const logout = (navigate) => {

  localStorage.removeItem(SAVED_USER_NAME);

  localStorage.removeItem(SAVED_USER_KEY);

  localStorage.removeItem(SAVED_USER_HAS_RESULT);

  localStorage.removeItem(SAVED_USER_IS_CALIBRATED);
  navigate(PATH_HOME);
};

export default logout;
```

## 2.15 Модуль components/heatmapViewer.js

```
import React, { useEffect } from 'react';
import * as heatmap from 'heatmap.js';

const HeatmapViewer = ({ result = [],
  className, style, id = 'heatmapContainer',
  config }) => {
  useEffect(() => {
    const heatmapInstance = heatmap.create({
      container: document.getElementById(id),
      ...config,
    });
    if (result?.length) {
      const dataPoint = result.map((e) => ({
        x: e.x > 0 ? Math.round(e.x) : 0,
        y: e.y > 0 ? Math.round(e.y) : 0,
        value: 1,
      }));
      const data = {
        max: 2,
        min: 0,
        data: dataPoint,
      };
      heatmapInstance.setData(data);
    }
  });

  return (
    <div style={{ width: '100%', height: '100%',
    ...style }} className={className}>
      <div style={{ width: '100%', height:
        '100%' }} id={id || 'heatmapContainer'}></div>
    </div>
  );
};

export default HeatmapViewer;
```

## 2.16 Модуль components/webGazerLoader.js

```
import React, { useEffect, useState } from
'react';

import { PATH_HOME,
SAVED_USER_IS_CALIBRATED } from
'../constants';

import Script from 'react-load-script';

import useModal from '../hooks/useModal';

import Calibration from
'../pages/calibrationPage';

import Modal from './modal/modal';

import Spinner from './spinner';

import { useTranslation } from 'react-i18next';

import logout from './logout';

export const WebGazerLoader = ({ children,
setUserIsCalibrated }) => {

  const { t } = useTranslation();

  const [isLoading, setLoading] =
useState(true);

  const [x, setX] = useState(-1);

  const [y, setY] = useState(-1);

  const [sessionResult, setSessionResult] =
useState([]);

  const [calibrated, setCalibrated] = useState(

JSON.parse(localStorage.getItem(SAVED_USE
R_IS_CALIBRATED)),

);

  const [webgazerInstance,
setWebgazerInstance] = useState(null);

  const [error, setError] = useState(null);

  const { isShowing, toggle } = useModal();
```

```
useEffect(() => {

  if (x !== -1 && y !== x && calibrated) {

    setSessionResult((prev) => [...prev, { x, y
}]);

  }

}, [x, y, calibrated]);

const handleScriptLoad = () => {

  setLoading(false);

  try {

    window.webgazer.setRegression('ridge');

    window.webgazer.showVideo(false);

    setWebgazerInstance(

window.webgazer.setGazeListener(function
(data, elapsedTime) {

  if (data === null) {

    return;

  }

  console.log({ data });

  const scrollTopDistance =
document.getElementsByClassName('App')[0]?.
scrollTop;

  console.log({ scrollTopDistance });

  setX(data.x);

  setY(scrollTopDistance ? data.y +
scrollTopDistance : data.y);
```



```

    }),
  );
} catch (e) {
  console.log('handleScriptLoad', e);
}
};

```

```

const handleScriptError = () => {
  setLoading(false);
  setError('Webgazer.js loading Error!');
};

```

```

const start = () => {
  console.log(window.webgazer);
  setSessionResult([]);
  if (window.webgazer.isReady()) {
    window.webgazer.resume();
  } else {
    try {
      webgazerInstance.begin();
    } catch (e) {
      setError(e);
      console.log('start-error', e);
    }
  }
};

```

```

const stop = (end) => {
  if (end) {
    window.webgazer.end();
  } else {

```

```

    window.webgazer.pause();
  }
};

const onSuccessCalibration = () => {
  setCalibrated(true);
  setUserIsCalibrated(true);
};

return (
  <>
  <Script
    url="js/Webgazer.js"
    onLoad={handleScriptLoad}
    onError={handleScriptError}
  />

  {error && (
    <Modal
      isShowing={isShowing}
      hide={toggle}
      header={t('error-title')}
      bodyText={error?.message || error}
      action={() =>
        logout(PATH_HOME)}
      buttonLabel={t('error-btn-label')}
    />
  )}

  {isLoading ? (
    <Spinner />
  ) : !calibrated ? (

```

```
      <Calibration start={start} stop={stop}  
onSuccess={onSuccessCalibration} />
```

```
    ): children ? (
```

```
      children(sessionResult, { stop, start })
```

```
    ): null}
```

```
<canvas
```

```
  id="plotting_canvas"
```

```
  width="500"
```

```
    height="500"
```

```
    style={{ display: 'none', cursor:  
'crosshair' }}
```

```
></canvas>
```

```
</>
```

```
);
```

```
};
```

## 2.17 Модуль components/testSwitcher.js

```
import React, { useState } from 'react';
import TestItem from '../testItem';

const TestSwitcher = ({ tests = [], start, stop,
processResult, finishTesting }) => {
  const testAmount = tests.length;

  const [currentIdx, setCurrentIdx] =
useState(0);

  const [testsIsFinished, setFinished] =
useState(false);

  const number = currentIdx + 1;

  const goNext = () => {
    if (number === testAmount) {
      setFinished(true);
      finishTesting();
    } else {
      setCurrentIdx((prev) => ++prev);
    }
  };

  if (testsIsFinished) return null;

  return (
    <TestItem
      key={currentIdx}
      {...tests[currentIdx]}
      goNext={goNext}
      stop={stop}
      start={start}
      processResult={processResult}
      number={number}
      total={testAmount}
    />
  );
};

export default TestSwitcher;
```

## 2.18 Модуль components/testItem.js

```
import React, { useEffect, useRef, useState }
from 'react';

import { useTranslation } from 'react-i18next';
import useModal from '../hooks/useModal';
import Modal from '../modal/modal';
```

```
const TestItem = ({
  id,
  title,
  startDescription,
  endDescription,
  time,
  stop,
  start,
  testComponent,
  processResult,
  goNext,
  number,
  total,
}) => {
  const { isShowing: startModalShow, toggle:
toggleStartModal } = useModal(true);

  const { isShowing: endModalShow, toggle:
toggleEndModal } = useModal(false);

  const [started, setStarted] = useState(false);
  const testTimer = useRef(null);

  const [isTimerFinished, setIsTimerFinished] =
useState(false);

  const [isProcessed, setIsProcessed] =
useState(false);

  const { t } = useTranslation();
```

```
const startTest = () => {
  setStarted(true);
  start();
};

useEffect(() => {
  if (isTimerFinished && !isProcessed) {
    const screenSize = { height:
window.innerHeight, width: window.innerWidth
};

    setIsProcessed(true);
    processResult(id, screenSize);
    stop(number === total);
  }

  // eslint-disable-next-line react-
hooks/exhaustive-deps
}, [isTimerFinished, processResult,
isProcessed]);

useEffect(() => {
  if (time > 0 && started &&
!testTimer.current) {
    console.log('Created timer');
    testTimer.current = setTimeout(() => {
      setIsTimerFinished(true);
      toggleEndModal();
      console.log('Time is over');
    }, time);
  }
}
```

```

    }, [started, time, processResult, id,
toggleEndModal]);

useEffect(() => {
  return () => {
    if (time) {
      clearTimeout(testTimer.current);
    }
  };
  // eslint-disable-next-line react-
hooks/exhaustive-deps
}, []);

```

```

const headerInfo =
<span>{`${number}/${total}`}</span>;

const startModalBody = (
  <div>
    <p>{startDescription}</p>
    {time > 0 && <p>{`${t('test-duration')}:
${time / 1000}${t('test-sec')}`}</p>}
  </div>
);

return (
  <
    <Modal

```

```

    isShowing={ startModalShow }
    action={ startTest }
    hide={ toggleStartModal }
    header={ title }
    bodyContent={ startModalBody }
    buttonLabel={ t('test-start-label') }
    headerInfo={ headerInfo }
  />
{testComponent}
<Modal
  isShowing={ endModalShow }
  action={ goNext }
  hide={ toggleEndModal }
  header={ title }
  bodyText={ endDescription }
  buttonLabel={ number === total ?
t('test-last-label') : t('test-next-label') }
  headerInfo={ headerInfo }
/>
</>
);

};

export default TestItem;

```

## 2.19 Модуль components/signinForm.js

```

/* eslint-disable jsx-a11y/anchor-is-valid */

import classNames from 'classnames';

import React, { useEffect, useState } from
'react';

import { useNavigate } from 'react-router-dom';

import Button from '../button';

import { SAVED_USER_KEY,
SAVED_USER_NAME } from '../constants';

import { createUser } from '../firebase/api';
// import gs from '../styles/global.module.css';
import s from './signinForm.module.css';
import Checkbox from '../checkbox/checkbox';
import useModal from '../hooks/useModal';
import Modal from '../modal';

import { useTranslation } from 'react-i18next';

const SignInForm = ({ className, nextPath })
=> {
  const [name, setName] = useState("");
  const [isValid, setValid] = useState(false);
  const [isAccepted, setIsAccepted] =
useState(false);

  const [isNameBlured, setNameIsBlured] =
useState(false);

  const [isTacBlured, setTacIsBlured] =
useState(false);

  const { isShowing, toggle } = useModal();
  const { t } = useTranslation();

  const [error, setError] = useState({ });

  const navigate = useNavigate();

  useEffect(() => {
    if (isNameBlured) handleValidationName();
    // eslint-disable-next-line react-
hooks/exhaustive-deps
  }, [name, isNameBlured]);

  useEffect(() => {
    if (isTacBlured) handleValidationTac();
    // eslint-disable-next-line react-
hooks/exhaustive-deps
  }, [isAccepted, isTacBlured]);

  const handleValidationName = () => {
    setError((prev) => ({ ...prev, name: null }));
    let nameError = null;
    let formIsValid = true;
    setValid(formIsValid);
    if (name.length < 3) {
      nameError = t('signin-error-short');
      formIsValid = false;
    }
    if (name.length > 20) {
      nameError = t('signin-error-long');
      formIsValid = false;
    }
    setError((prev) => ({ ...prev, name:
nameError }));
    setValid(formIsValid);
  }

```

```

    return formIsValid;
};

const handleValidationTac = () => {
    setError((prev) => ({ ...prev, tac: null }));
    let tocError = null;
    let formIsValid = true;
    setValid(formIsValid);
    if (!isAccepted) {
        tocError = t('signin-error-toc');
        formIsValid = false;
    }
    setError((prev) => ({ ...prev, tac: tocError }));
    setValid(formIsValid);
    return formIsValid;
};

const onSubmit = async (e) => {
    e.preventDefault();
    if (handleValidationTac() &&
    handleValidationName()) {

        localStorage.setItem(SAVED_USER_NAME,
        JSON.stringify(name));

        await createUser(name, (key) => {

            localStorage.setItem(SAVED_USER_KEY,
            JSON.stringify(key));

            nextPath && navigate(nextPath);

        });
    }

```

```

    };

    const tacContent = (
        <div>
            <p>{t('toc-body-p1')}</p>
            <p>{t('toc-body-p2')}</p>
            <p>
                <a target="_blank" rel="noreferrer"
                href="https://webgazer.cs.brown.edu/">
                    {t('toc-lib-web-link')}
                </a>
            </p>
            <p>
                <a target="_blank" rel="noreferrer"
                href="https://github.com/brownhci/WebGazer">
                    {t('toc-lib-git-link')}
                </a>
            </p>
            <p>
                <a target="_blank" rel="noreferrer"
                href="github.com">
                    {t('toc-project-link')}
                </a>
            </p>
        </div>
    );

    return (
        <
            <form className={classNames(s.form,
            className)}>
                <div className={s.inputLine}>
                    <input
                        className={s.input }

```

<pre> type="text" value={ name} onChange={ (e) =&gt; {     setName(e.target.value);     // handleValidationName(); }} onBlur={ () =&gt; setNameIsBlured(true)} placeholder={t('signin- nickname')} aria-label="Nick name" /&gt; {error?.name &amp;&amp; &lt;p className={s.error}&gt;{error.name}&lt;/p&gt;} &lt;/div&gt; &lt;div className={s.inputLine}&gt; &lt;Checkbox checked={isAccepted} onBlur={ () =&gt; setTacIsBlured(true)} onChange={ () =&gt; {     setIsAccepted((prev) =&gt; !prev);     // handleValidationTac(); }} &gt; &lt;span&gt; {t('signin-checkbox')} &lt;Button onClick={ (e) =&gt; {     e.preventDefault();     toggle(); }} </pre>	<pre> type="link" &gt; {t('signin-toc-link')} &lt;/Button&gt; &lt;/span&gt; &lt;/Checkbox&gt; {error?.tac &amp;&amp; &lt;p className={s.error}&gt;{error.tac}&lt;/p&gt;} &lt;/div&gt; &lt;div className={s.inputLine}&gt; &lt;Button type="primary" onClick={onSubmit}&gt;     {t('signin-submit-btn-label')} &lt;/Button&gt; &lt;/div&gt; &lt;/form&gt; &lt;Modal header={t('toc-lib-header')} bodyContent={tacContent} isShowing={isShowing} hide={toggle} action={ () =&gt; setIsAccepted(true)} buttonLabel={t('toc-lib-submit-btn- label')} /&gt; &lt;/&gt; ); };  export default SignInForm; </pre>
---	---



## 2.20 Модуль components/modal.js

```
import classNames from 'classnames';
import React from 'react';
import ReactDOM from 'react-dom';
import { useTranslation } from 'react-i18next';
import Button from '../button/button';
import Scrollbar from '../scrollbar';
import gs from '../../styles/global.module.css';
import s from './modal.module.css';

const Modal = ({
  isShowing,
  hide,
  action,
  header,
  bodyText,
  bodyContent,
  buttonLabel,
  headerInfo,
}) => {
  const { t } = useTranslation();
  const onAction = (action) => {
    hide();
    action();
  };

  const renderButton = () => {
    if (Array.isArray(action) &&
    Array.isArray(buttonLabel)) {
      return buttonLabel.map((lbl, idx, arr) =>
(
```

```
<Button
  key={lbl + idx}
  type={idx === arr.length - 1 &&
'primary'}
  data-dismiss="modal"
  onClick={() =>
onAction(action[idx])}
  >
    {lbl}
  </Button>
));
}
return (
  <Button data-dismiss="modal"
onClick={() => onAction(action)}>
    {buttonLabel || t('ok-label')}
  </Button>
);
};

return isShowing
  ? ReactDOM.createPortal(
    <React.Fragment>
      <div
className={s.modalOverlay}></div>
      <div className={s.wrapper}>
        <div
className={classNames(s.modal, gs.box)}
          aria-modal
          aria-hidden
```

```
      tabIndex={-1}
      role="dialog"
    >
      <div className={s.header}>
        <h1>{header}</h1>
        {headerInfo}
      </div>
      <Scrollbar
        className={s.body}>
        {bodyText &&
        <p>{bodyText}</p>}
        {bodyContent}
      </Scrollbar>
```

```
      <div
        className={classNames(s.buttonWrapper,
          gs.flexWrapperRowCenter)}>
        {renderButton()}
      </div>
    </div>
  </div>
  </React.Fragment>,
  document.body,
)
: null;
};
export default Modal;
```

## 2.21 Модуль components/langSwitcher.js

```
import classNames from 'classnames';
import React from 'react';
import { useTranslation } from 'react-i18next';
import { USER_LANG } from '../constants';
import Button from '../button';
import gs from '../styles/global.module.css';

const LangSwitcher = ({ className }) => {
  const { i18n } = useTranslation();
  const currentLang = i18n.language;
  const changeLanguageHandler = (lang) => {
    const languageValue = lang;
    i18n.changeLanguage(languageValue);
    localStorage.setItem(USER_LANG,
      JSON.stringify(lang));
  };
  return (
    <div
      className={classNames(gs.flexWrapperRowCe
        nter, className)}>
      <Button
        size="sm"
```

```
        puched={currentLang === 'en'}
        onClick={() =>
          changeLanguageHandler('en')}
        tabIndex={1}
      >
        EN
      </Button>
      <Button
        tabIndex={2}
        size="sm"
        puched={currentLang === 'ua'}
        onClick={() =>
          changeLanguageHandler('ua')}
      >
        YKP
      </Button>
    </div>
  );
};

export default LangSwitcher;
```

## 2.22 Модуль components/button.js

```
import classNames from 'classnames';
import React from 'react';
import s from './button.module.css';

const Button = ({ children, type, pushed,
disabled, size, ...restProps }) => {
  return (
    <button
      {...restProps}
      disabled={disabled}
      className={classNames(
        s.button,
        s[size],
        disabled && s.buttonDisabled,
        pushed && s.buttonPushed,
        s[type],
      )}
    >
      {children}
    </button>
  );
};

export default Button;
```

## 2.23 Модуль components/checkbox.js

```
import React from 'react';
import s from './checkbox.module.css';

const Checkbox = ({ children, checked,
onChange, ...restProps }) => {
  return (
    <div className={s.checkboxWrapper}>
      <label
className={s.checkboxWrapper}>
        <input
          {...restProps}
          type="checkbox"
          checked={checked}
          onChange={onChange}
          className={s.input}
        />
      </label>
      {children}
    </div>
  );
};

export default Checkbox;
```

## 2.24 Модуль components/scrollbar.js

```
import React, { useState } from 'react';
import { Scrollbars } from 'react-custom-scrollbars-2';
// import s from './scrollbar.module';

const Scrollbar = (props) => {
  const [top, setTop] = useState(0);

  const handleUpdate = (values) => {
    const { top } = values;
    setTop(top);
  };

  const renderView = ({ style, ...props }) => {
    const viewStyle = {
      padding: 15,
    };

    return <div className="box" style={{
...style, ...viewStyle }} {...props} />;
  };

  const renderThumb = ({ style, ...props }) => {
    const thumbStyle = {
      width: '5px',
      backgroundColor: 'black',
    };

    return <div style={{ ...style, ...thumbStyle }} {...props} />;
  };

  return (
    <Scrollbars
      renderView={renderView}
      renderThumbHorizontal={renderThumb}
      renderThumbVertical={renderThumb}
      onUpdate={handleUpdate}
      {...props}
    />
  );
};

export default Scrollbar;
```

## 2.25 Модуль testScreens/testsCopr.js

```
import React from 'react';
import { useTranslation } from 'react-i18next';
import s from './testScreen.module.css';
const rozetkaImg = require('./rozetka.png');
const zolotoyvekImg =
require('./zolotoyvek.png');
const hotlineFullImg =
require('./hotlineFull.png');
const hotlineFinanceImg =
require('./hotlineFinance.png');

const Tests = () => {
  const { t } = useTranslation();

  return [
    {
      id: 'testId1',
      title: t('1test-title'),
      startDescription: t('1test-start-
description'),
      endDescription: t('1test-end-description'),
      time: 10000,
      testComponent: (
        <div className={s.iframe}>
          <img src={rozetkaImg}
alt="rozetka" />
        </div>
      ),
    },
    {
      id: 'testId2',
```

```
      title: t('2test-title'),
      startDescription: t('2test-start-
description'),
      endDescription: t('2test-end-description'),
      time: 10000,
      testComponent: (
        <div className={s.iframe}>
          <img src={zolotoyvekImg}
alt="zolotoyvek" />
        </div>
      ),
    },
    {
      id: 'testId3',
      title: t('3test-title'),
      startDescription: t('3test-start-
description'),
      endDescription: t('3test-end-description'),
      time: 10000,
      testComponent: (
        <div className={s.iframe}>
          <img src={hotlineFullImg}
alt="hotlineFull" />
        </div>
      ),
    },
    {
      id: 'testId4',
      title: t('4test-title'),
      startDescription: t('4test-start-
description'),
```

endDescription: t('4test-end-description'),	),
time: 10000,	},
testComponent: (	];
<div className={s.iframe}>	};
<img src={hotlineFinanceImg}	
alt="hotlineFinance" />	export default Tests;
</div>	



## 2.26 Модуль locales/en.js

```
{
  "1test-title": "Colors",
  "1test-start-description": "Look on the screen",
  "1test-end-description": "Thanks!",
  "2test-title": "Colors and icons",
  "2test-start-description": "Look on the screen",
  "2test-end-description": "Thanks!",
  "welcome-header": "Welcome ",
  "welcome-body-p1": "My name is Ilia and this
website is a part of my master's diploma project.
The purpose of this web site is to collect and
further analysis eye tracking data with the help
of a web camera while examinee look at
different pictures or web pages and interact with
them.",
  "welcome-body-p2": "The system is not
collecting any of your personal data, more about
it and principles of system operation in ",
  "welcome-body-btn-label": "terms and
condition",
  "signin-error-short": "The name is too short",
  "signin-error-long": "The name is too long",
  "signin-error-toc": "Needs accept terms and
conditions",
  "signin-nickname": "YOUR NICK",
  "signin-checkbox": "I'm accept ",
  "signin-toc-link": "terms and conditions",
  "signin-submit-btn-label": "Start",
  "toc-body-p1": "This web application is not
collecting any of your personal data. You can
use your nick name instead real name. Web
camera is not save or sent video or audio but
only using for recognition of eyes and prediction
of gaze location on screen surface.",
  "toc-body-p2": "For implementing this
functionality was used Webgazer library.",
  "toc-project-link": "Link on project source",
  "toc-lib-git-link": "Link on Webgazer source
repository",
  "toc-lib-web-link": "Link on Webgazer
website",
  "toc-lib-header": "Terms and conditions",
  "toc-lib-submit-btn-label": "Accept",
  "instruction-title": "Instruction",
  "instruction-p1": "As examinee you will
viewed and interactive with several web pages
and system will collect position of your gaze on
the screen.",
  "instruction-p2": "During testing is preferably
to do not move a lot. So please seat comfortable
in front of your laptop.",
  "instruction-p3": "The examination will
contain several parts:",
  "instruction-li1": "Calibration - process what
help setup the system for your face;",
  "instruction-li2": "Precision measurement -
will calculate precision of system after previous
calibration;",
  "instruction-li3": "Set of tests - tests itself.
Before each of them you will see instruction
with explanation;",
  "instruction-p4": "After passing all test you be
able to view map of your gaze for each test.",
  "instruction-to-tests-label": "Start testing",
  "instruction-calibration-label": "Go to
calibration",
  "calibration-title": "Calibration",
  "calibration-instruction-1": "Please seet in
from of your webcamera;"
```

"calibration-instruction-2": "Allow use your camera if you will see corresponding dialog;",

"calibration-instruction-3": "Your face must fit into green square of camera preview;",

"calibration-instruction-4-1": "The blue dot ",

"calibration-instruction-4-2": "is a visulisation of your gaze location prediction;",

"calibration-instruction-5-1": "Please click each red button ",

"calibration-instruction-5-2": " 5 times until it beacome yellow ",

"calibration-instruction-6": "Please look on the button when you click it;",

"calibration-start-label": "Start calibraton",

"measurement-title": "Precision measurement",

"measurement-instruction-1": "Please do not move your mouse;",

"measurement-instruction-2-1": "To measure accuracy of calibration look on the button ",

"measurement-instruction-2-2": " during 5 seconds;",

"measurement-result-title": "Your accuracy measure is ",

"measurement-result-text": "Recommended accuracy to continue testing is more then 70%",

"measurement-start-label": "Start",

"measurement-recalibrate-label": "Recalibrate",

"measurement-continue-label": "Go to test",

"results-title": "Results of ",

"results-logout-label": "Logout",

"thanks-title": "Thank you for passing all tests",

"thanks-text": "Now you can view your results or logout and try again",

"thanks-logout-label": "Logout",

"thanks-results-label": "Results",

"test-duration": "Duration",

"test-sec": "sec",

"test-start-label": "Start",

"test-next-label": "Next",

"test-last-label": "Ok",

"error-title": "Error is happened",

"error-btn-label": "Start from scratch",

"ok-label": "ok"

}

## 2.27 Модуль locales /ua.js

```
{
  "1test-title": "Кольори",
  "1test-start-description": "Подивіться на
екран",
  "1test-end-description": "Дякую!",
  "2test-title": "Кольори та іконки",
  "2test-start-description": "Подивіться на
екран",
  "2test-end-description": "Дякую!",
  "welcome-header": "Ласкаво просимо ",
  "welcome-body-p1": "Мене звати Ілля, і цей
веб-сайт є частиною мого магістерського
дипломного проекту. Метою цього веб-сайту
є збір і подальший аналіз даних відстеження
очей за допомогою веб-камери під час того як
екзаменований переглядає різні зображення
або веб-сторінки і взаємодіє з ними.",
  "welcome-body-p2": "Система не збирає
жодних ваших особистих даних, докладніше
про це та принципи роботи системи в розділі
",
  "welcome-body-btn-label": "умови",
  "signin-error-short": "Ім'я занадто коротке",
  "signin-error-long": "Ім'я занадто довге",
  "signin-error-toc": "Потрібно прийняти
положення та умови",
  "signin-nickname": "ВАШ НІК",
  "signin-checkbox": "Я приймаю ",
  "signin-toc-link": "умови",
  "signin-submit-btn-label": "Почати",
  "toc-body-p1": "Ця веб-програма не збирає
жодних ваших особистих даних. Ви можете
використовувати свій псевдонім замість
```

справжнього імені. Веб-камера не зберігає та не надсилає відео чи аудіо, а використовує лише для розпізнавання очей і передбачення розташування погляду на поверхні екрана.",

"toc-body-p2": "Для реалізації цієї функції була використана бібліотека Webgazer.",

"toc-project-link": "Посилання на джерело проекту",

"toc-lib-git-link": "Посилання на репозиторій джерел Webgazer",

"toc-lib-web-link": "Посилання на веб-сайт Webgazer",

"toc-lib-header": "Умови",

"toc-lib-submit-btn-label": "Прийняти",

"instruction-title": "Інструкція",

"instruction-p1": "Як піддослідний ви переглянете кілька веб-сторінок та будете слідувати інструкціям, а система збиратиме позицію вашого погляду на екрані.",

"instruction-p2": "Під час тестування бажано не рухатися багато. Тому, будь ласка, сядьте зручніше перед своїм ноутбуком.",

"instruction-p3": "Іспит складатиметься з кількох частин:",

"instruction-li1": "Калібрування - процес що допоможе системи налаштуватися під ваше обличчя;",

"instruction-li2": "Вимірювання точності - обчислення точності системи після попереднього калібрування;",

"instruction-li3": "Набір тестів - самі тести. Перед кожним з них ви побачите інструкцію з поясненням;",

"instruction-p4": "Після проходження всіх тестів ви зможете переглядати карту свого погляду для кожного тесту.",

"instruction-to-tests-label": "Почати тестування",

"instruction-calibration-label": "Перейти до калібрування",

"calibration-title": "Калібрування",

"calibration-instruction-1": "Будь ласка, сядте перед веб-камерою;",

"calibration-instruction-2": "Дозвольте використовувати вашу камеру, якщо ви побачите відповідне діалогове вікно;",

"calibration-instruction-3": "Ваше обличчя має поміститися в зелений квадрат у превью камери;",

"calibration-instruction-4-1": "Блакитна точка ",

"calibration-instruction-4-2": "це візуалізація прогнозу розташування вашого погляду;",

"calibration-instruction-5-1": "Натисніть кожну червону кнопку ",

"calibration-instruction-5-2": " 5 разів, поки не стане жовтим ",

"calibration-instruction-6": "Будь ласка, дивіться на кнопку, коли ви її натискаєте;",

"calibration-start-label": "Почати калібрування",

"measurement-title": "Вимірювання точності",

"measurement-instruction-1": "Будь ласка, не рухайте мишею;",

"measurement-instruction-2-1": "Щоб виміряти точність калібрування, подивіться на кнопку ",

"measurement-instruction-2-2": "протягом 5 секунд;",

"measurement-result-title": "Ваша міра точності ",

"measurement-result-text": "Рекомендована точність для продовження тестування більше ніж 70%",

"measurement-start-label": "Початок",

"measurement-recalibrate-label": "Перекалібрувати",

"measurement-continue-label": "Перейти до тестів",

"results-title": "Результати ",

"results-logout-label": "Вийти",

"thanks-title": "Дякуємо за проходження всіх тестів",

"thanks-text": "Тепер ви можете переглянути свої результати або вийти з системи та повторити спробу",

"thanks-logout-label": "Вийти",

"thanks-results-label": "Результати",

"test-duration": "Тривалість",

"test-sec": "с",

"test-start-label": "Почати",

"test-next-label": "Далі",

"test-last-label": "Добре",

"error-title": "Сталася помилка",

"error-btn-label": "Почати з нуля",

"ok-label": "добре"

}

Міністерство освіти і науки України  
Міністерство інфраструктури України  
Український державний університет науки та технологій  
Східний науковий центр транспортної академії наук

ДОДАТОК В



**TEMPUS: CITISET & SEREIN & CRENG**

### **ТЕЗИ**

**XVI Міжнародної науково-практичної конференції  
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ  
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ ТА ОСВІТІ»**

**ABSTRACTS  
of the XVI International Conference  
«MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES  
ON A TRANSPORT, IN INDUSTRY AND EDUCATION»**

**14.12.2022 – 15.12.2022**

**Дніпро  
2022**

Аналіз напрямків розвитку систем військового радіозв'язку .....	108
Скалько В.В., Панченко Є.О., Український державний університету науки і технологій, Україна	
Розроблення алгоритму масштабування растрових зображень за допомогою сплайнів .....	109
Товстоп'ят М.В., Бабенко М.В., Дніпровський державний технічний університет, Україна	
Аналіз графічних інтерфейсів за допомогою трекінга очей.....	110
Фролов І.В., Горбова О.В., Український державний університет науки і технологій, Україна	
Дослідження часових характеристик в системах реального часу .....	111
Чокан К. Ю., Горячкін В.М., Український державний університет науки і технологій, Україна	
Методи перед обробки даних для алгоритмів сортування .....	112
Шинкаренко В. І., Макаров О. В., Український державний університет науки і технологій, Україна	
Підвищення пропускної спроможності сервера на основі дослідження паралельних алгоритмів обробки великих масивів сейсмологічних даних.....	113
Ющенко В.В., Жульковський О.О., Жульковська І.І., Дніпровський державний технічний університет, Україна	
Statistical Analysis of Medical Data About Women's Health: Case Study.....	114
Alexey Dubinsky, Dnipro State Medical University, Ukraine, University of Seville, Spain Aurea Simon-Soro, University of Seville, Spain	
Ontologies in the transportation context.....	115
Larysa Zhuchyi, Ukrainian State University of Science and Technologies, railML.org, Germany	
Using code refactoring to optimize and improve cross-platform applications maintain.....	116
Horiachkin V. M., Syrota O. A., Ukrainian State University of Science and Technologies, Ukraine	
Дослідження методів сегментації зображень земної поверхні високого просторового розрізнення.....	117
Гнатушенко Вік.В., Китова К.В., Калинина Н.Ю., Український державний університет науки і технологій, Україна	
Інтелектуальна комп'ютерна підтримка надбання вмінь побудови частотних характеристик.....	119
Двінських Д.Г., Чухрай А.Г., Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут», Україна	
Екосистема React та майбутнє фреймворків.....	121
Трипутін В. С., Український державний університет науки та технологій, Україна	

## **ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА ДІДЖИТАЛІЗАЦІЯ СФЕРИ ОСВІТИ ..... 123**

Інформаційні технології в управлінні економічною ефективністю інфраструктурних проєктів.....	124
Башмаков М., Український державний університет науки і технологій, Україна	

## Аналіз графічних інтерфейсів за допомогою трекінга очей

Фролов І.В., Горбова О.В., Український державний університет науки і технологій, Україна

Технологію eye-tracking ще іноді називають «окулографія», але термін не прижився. Ай-трекінг — технологія відстеження положення очей, також звана технологією відстеження ліній погляду або точок погляду. Технологія ця відносно нова і в Україні вона ще не отримала належного поширення, хоча в останні роки ситуація змінюється на краще. Пристрої для відстеження погляду або айтрекери, складаються як правило з апаратної частини, яка підключається до комп'ютера і програмного забезпечення. Хоча існують і самодостатні пристрої «все в одному».

За допомогою технології айтрекінгу маємо можливість по новому досліджувати графічний інтерфейс та отримувати більш точну інформацію, не приховану свідомо чи підсвідомо учасником дослідження. Завдяки такому підходу ми зможемо бачити де саме фокусується погляд людини, чи може вона знайти те, що треба та куди саме вона дивиться, ми зможемо вдосконалювати інтерфейси та робити їх більш зручними для користування, в також прибирати те, що заважає та правильно розміщувати рекламу.

Методом який є доцільний у нашому випадку є Usability-тестування. Його можна охарактеризувати як експеримент, що виконується з метою визначення, наскільки добре люди можуть використовувати певний штучний об'єкт, такий як веб-сторінка, призначений для користувача інтерфейс або пристрій, для його передбачуваного застосування, тобто usability-тестування вимірює зручність користування об'єктом. Usability-тестування зосереджено на певному об'єкті або невеликому наборі об'єктів, в той час як дослідження взаємодії людина - комп'ютер в цілому формулюють універсальні принципи. Usability-тестування - метод оцінки зручності продукту у використанні, заснований на залученні користувачів в якості тестувальників, випробувачів і підсумовуванні отриманих від них висновків. Процес тестування проходить наступним чином. Респонденту пропонують спробувати вирішити завдання, для яких розроблявся тестований продукт. Все фіксується в протоколі для ще більш детального аналізу. ПК-тестування добре виявляє можливі труднощі, з якими може зіткнутися реальний споживач товару або послуги. Спостереження за користувачем дозволяє зібрати велику кількість корисної інформації для поліпшення продукту або сервісу.

Для проведення Usability-тестування доцільно буде використати Моніторний айтрекінг. Він виконується респондентом у вигляді спеціального завдання за допомогою настільного комп'ютера з інфрачервоною камерою і програмним забезпеченням, яке ідентифікує особливості ока, дозволяє аналізувати його поведінку, будувати "карти уваги". При цьому респондент, який виконує завдання, знаходиться в умовах, близьких до природних, його руху нічого не обмежує. Це один з найпоширеніших видів дослідження, тому, що він являється одним з найдешевших, та дозволяє простим у своїй реалізації. Також він дає можливість протестувати одразу декілька користувачів на одному сервісі, або декілька сервісів одночасно за допомогою декількох користувачів та комп'ютерів.

Аналіз за допомогою айтрекінгу тільки набуває своєї популярності, але вже зараз можна сказати, що ця технологія через деякий час стане невіддільною у дослідженнях різних сфер, серед яких маркетингові дослідження, застосування у медицині, розробці веб-сторінок, веб-додатків, програм інтерфейсів. Наразі сфера досліджень та аналізу за допомогою айтрекінгу є однією з найперспективніших.