

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ **Ю. П. Кондратенко**
«____» _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СИСТЕМА ДІАГНОСТИКИ ЗОРУ ДЛЯ
ПРОФІЛАКТИКИ ЗАХВОРЮВАНЬ ОЧЕЙ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810114

Виконав студент 4-го курсу, групи 401
_____ *А. В. Клішевська*
«20» червня 2022 р.

Керівник: зав. кафедри ІС, д.т.н.,
професор
_____ *Ю. П. Кондратенко*
«20» червня 2022 р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«___» _____ 2021 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Клішевській Альоні Володимирівні.

1. Тема кваліфікаційної роботи «Система діагностики зору для профілактики захворювань очей».

Керівник роботи Кондратенко Юрій Пантелійович, д-р техн. наук, професор.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» _____ 20__ р. № _____

2. Строк представлення кваліфікаційної роботи студентом «___» _____ 2022 р.

3. Вхідні (початкові) дані до роботи: дані згідно з предметною областю та постановленою задачею.

Очікуваний результат: система діагностики зору для профілактики захворювань очей.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз сучасного стану задачі діагностування зору;
- огляд існуючих методів та технологій для вирішення поставленої задачі;

– реалізація системи для діагностування зору для профілактики захворювань очей з використанням популярних фреймворків та бібліотек.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Аналіз забезпечення вимог охорони праці у приміщенні серверної кімнати».

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	ст. викладач Макарова О.В.	

Керівник роботи д-р техн. наук, проф. Кондратенко Ю. П.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Клішевська А. В.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 23 » _____ листопада _____ 2021 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Система діагностики зору для профілактики захворювань очей

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	26.10.2021	29.10.2021	Виконано
2	Отримання завдання на виконання БКР	20.11.2021	20.11.2021	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	05.12.2021	08.12.2021	Виконано
4	Отримання завдання на переддипломну практику	20.05.2022	20.05.2022	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	23.05.2022	04.06.2022	Виконано
6	Розробка звіту з переддипломної практики	03.06.2022	06.06.2022	Виконано
7	Виконання БКР: аналіз сучасного стану задачі відслідковування очей, огляд існуючих технологій, розробка ПЗ	28.02.2022 та 06.06.2022	27.03.2022 та 19.06.2022	Виконано
8	Попередній захист БКР на засіданні комісії кафедри	30.05.2022	31.05.2022	Виконано
9	Доробка та остаточне оформлення БКР	02.06.2022	20.06.2022	Виконано
10	Подання БКР рецензенту	16.06.2022	18.06.2022	Виконано
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	20.06.2022	22.06.2022	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	27.06.2022	29.06.2022	Виконано

Розробив студент Клішевська А.В _____
(прізвище та ініціали) (підпис)

Керівник роботи д.т.н., проф. Кондратенко Ю. П. _____
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

« 11 » _____ 12 _____ 2021 р.

АНОТАЦІЯ

**бакалаврської кваліфікаційної роботи студентки групи 401 ЧНУ ім. Петра
Могили**

Клішевської Альони Володимирівни

Тема: «Система діагностики зору для профілактики захворювань очей»

Об'єкт роботи – процеси та інформаційні технології діагностики захворювань.

Предмет роботи – методи діагностики зору для запобігання захворювань очей.

Метою бакалаврської кваліфікаційної роботи є діагностика зору у різних напрямках (гострота, кольоросприйняття, ступінь контрасту, тощо) для профілактики захворювань очей на основі розробленої системи.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, трьох розділів та висновків.

У першому розділі розглядається актуальність та проблеми сталого розвитку діагностики зору, проводиться аналіз наявних аналогів та розглядаються види проведення діагностики зору.

У другому розділі досліджено методи, алгоритми та інформаційні технології для вирішення поставленої задачі. Обґрунтовується вибір мови програмування та СКБД.

У третьому розділі описано програмну реалізацію системи діагностики зору для профілактики захворювань очей. Додатково інтегровано базу даних для зберігання результатів користувача і відображення статистики у особистому кабінеті.

В результаті було досягнуто поставлені цілі та реалізовано систему, яка коректно відпрацьовує та повертає досить точний та зрозумілий результат користувачу.

Бакалаврська кваліфікаційна робота містить 67 сторінок, 57 рисунків, 4 таблиці, 30 використаних джерел.

Ключові слова: C#, користувач, додаток, форми, Windows Forms, наївний байєсівський класифікатор, Data Mining.

ABSTRACT

**of the bachelor's qualification work of the student of group 401 of Petro Mohyla
Black Sea National University**

Klishevska Alona Volodymyrivna

Topic "Visual diagnostics system for the prevention of eye diseases"

The object of work - processes and information technologies for disease diagnosis.

The subject of work - methods of vision diagnosis to prevent eye diseases.

The purpose of the bachelor's thesis is the diagnosis of vision in different areas (sharpness, color perception, degree of contrast, etc.) for the prevention of eye diseases based on the developed system.

The work consists of a professional section and a special section on labor protection. The explanatory note consists of an introduction, three chapters and conclusions.

The first section considers the relevance and problems of sustainable development of vision diagnostics, analyzes the existing analogues and considers the types of vision diagnostics.

The second section examines the methods, algorithms and information technologies for solving the problem. The choice of programming language and DBMS is substantiated.

The third section describes the software implementation of the visual diagnostics system for the prevention of eye diseases. Additionally, a database is integrated to store user results and display statistics in the personal account.

As a result, the set goals were achieved and a system was implemented that correctly works out and returns a fairly accurate and clear result to the user.

The bachelor's thesis contains 67 pages, 57 figures, 4 tables, 30 sources used.

Key words: C#, user, application, forms, Windows Forms, Naive Bayes classifier, Data Mining.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП	5
1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ СИСТЕМ ДІАГНОСТУВАННЯ ЗОРУ ДЛЯ ПРОФІЛАКТИКИ ЗАХВОРЮВАНЬ ОЧЕЙ. ПОСТАНОВКА ЗАДАЧІ	7
1.1 Актуальність та проблеми сталого розвитку	7
1.2 Види і методи діагностики зору	9
1.3 Опис та аналіз наявних аналогів	16
1.4 Постановка задачі	21
Висновки до розділу 1	22
2 МОДЕЛІ, МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	24
2.1 Використання наївного байєсівського класифікатора в розробці системи	24
2.2 Аналіз інструментальних засобів розробки системи	26
2.3 Аналіз інструментальних засобів розробки графічного інтерфейсу системи	31
2.4 Обґрунтування вибору СКБД для системи	33
Висновки до розділу 2	34
3 МОДЕЛЮВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІС	36
3.1 Структура бази даних	36
3.2 Структура системи діагностики зору для профілактики захворювань очей	37
Висновки до розділу 3	59
4 ОХОРОНА ПРАЦІ	62
ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	73
ДОДАТОК А Код програмної реалізації авторизації	76
ДОДАТОК Б Код програмної реалізації реєстрації	79
ДОДАТОК В Код програмної реалізації відновлення паролю користувача	83
ДОДАТОК Г Код програмної реалізації головного меню системи	87
ДОДАТОК Д Код програмної реалізації діагностики наявності патології макули	89
ДОДАТОК Е Код програмної реалізації діагностики на наявність далекозорості або короткозорості	92
ДОДАТОК Ж Код програмної реалізації діагностики контрастної чутливості	96

ПЕРЕЛІК СКОРОЧЕНЬ

БД	– база даних
ВМД	– вікова макулодистрофія
ГК	– графічний інтерфейс користувача
ІС	– інформаційна система
ПК	– персональний комп'ютер
ПЗ	– програмне забезпечення
СКБД	– системи керування базами даних

Пояснювальна записка

до кваліфікаційної роботи

на тему:

«СИСТЕМА ДІАГНОСТИКИ ЗОРУ ДЛЯ ПРОФІЛАКТИКИ ЗАХВОРЮВАНЬ ОЧЕЙ»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810114

Виконав студент 4-го курсу, групи 401

А.В. Клішевська

(підпис, ініціали та прізвище)

«___» _____ 2022 р.

Керівник: зав. кафедри ІС, д.т.н.,
професор

(наук. ступінь, вчене звання)

Ю.П. Кондратенко

(підпис, ініціали та прізвище)

«___» _____ 2022 р.

Миколаїв – 2022

ВСТУП

Приблизно 80% інформації з оточуючого світу, яка досягає людського мозку, походить від очей. Зорова кора розташована в задній частині мозку і є основним відділом, який отримує, інтегрує та обробляє візуальну інформацію, яку передає сітківка ока. Дослідження показали, що існує великий зв'язок між мигдалеподібним тілом та зоровою корою [1].

Мигдалеподібне тіло грає ключову роль в обробці емоцій, спогадів та мотивації, отримуючи інформацію від органів чуття. Потім мигдалеподібне тіло обробить це та скаже тілу відреагувати відповідним чином [1].

У випадку коли ми бачимо павука, ми зазвичай можемо відчувати страх і підстрибувати. Отримуючи подарунок ви можете відреагувати, висловивши захоплення, любов та прихильність до людини. За відсутності очей, людському мозку було б дуже складно автоматично обробляти інформацію та правильно реагувати на навколишні чинники.

На превеликий жаль, у житті настає переломний момент, коли за читанням улюбленої книги або стоячи біля прилавка магазину, приходить розуміння того, що ви не можете прочитати необхідний вам рядок або роздивитися ціну на ціннику.

Останніми роками лікарі констатують загальний рівень погіршення гостроти зору не тільки серед дорослих і літніх людей, а й серед молодого покоління та підлітків. Варто не нехтувати першими ознаками погіршення стану зору (підвищену стомлюваність очей, почервоніння та подразнення повік) та своєчасно проконсультуватися у лікаря-офтальмолога.

Перевірка зору з використанням online-сервісів або застосунків допоможе комфортно та досить просто слідкувати за станом зору, а у випадку його погіршення користувач зможе своєчасно звернутися до лікаря-офтальмолога за більш точною консультацією.

Тому згідно з актуальністю теми діагностики зору **об'єктом дослідження** процеси та інформаційні технології діагностики захворювань, а саме найвний байєсівський класифікатор, який є широко застосовуваним у медицині. Згідно з цим, **предметом дослідження** є методи діагностики зору для запобігання захворювань очей.

Мета: діагностика зору у різних напрямках (гострота, кольоросприйняття, ступінь контрасту, тощо) для профілактики захворювань очей на основі розробленої системи.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати сучасний стан задачі діагностування зору;
- окреслити існуючі методи діагностування зору;
- проаналізувати існуючі аналоги;
- реалізувати систему діагностування зору для профілактики захворювань очей з використанням популярних фреймворків та бібліотек.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ СИСТЕМ ДІАГНОСТУВАННЯ ЗОРУ ДЛЯ ПРОФІЛАКТИКИ ЗАХВОРЮВАНЬ ОЧЕЙ. ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність та проблеми сталого розвитку

Останніми роками лікарі констатують загальний рівень погіршення гостроти зору не тільки серед дорослих і літніх людей, а й серед молодого покоління та підлітків. Поширеність та доступність різноманітних девайсів, всеосяжна комп'ютеризація, погіршення екології, стабілізатори та консерванти – усе це стало частиною глобальних причин погіршення зору не тільки в Україні, а й в усьому світі. Для того, щоб привернути загальну увагу суспільності до проблем очей, ВООЗ заснувало свято, котре відзначається у другий четвер жовтня і називається «Всесвітній день зору».

На сьогоднішній день в світі налічується близько 134 млн людей, які страждають поганим зором і 34 млн сліпих. За підрахунками ВООЗ, кожні 5 секунд сліпне одна доросла людина і щохвилини - дитина. При цьому вважається, що 75% випадків можна було б уникнути [2]. Тому, щоб запобігти поповнюванню сумного списку, профілактичний огляд обов'язковий для всіх: для дітей - починаючи з 6 місячного віку, для дорослих - не рідше одного разу на рік.

Варто не нехтувати першими ознаками погіршення стану зору (підвищену стомлюваність очей, почервоніння та подразнення повік) та своєчасно проконсультуватися у лікаря-офтальмолога. Перевірити зір, як правило, можна у найближчому салоні оптики. Якщо така можливість відсутня, на допомогу придуть спеціальні онлайн-тести для самостійної перевірки зору та матеріали, доступні в інтернеті для кожного.

Відмінний зір значно підвищує рівень якості життя, оскільки очі є нашим найважливішим органом почуттів. 80% вражень ми отримуємо саме за допомогою очей. Не дарма очі ще називають «вікном у навколишній світ».

Регулярна діагностика стану очей у кваліфікованого спеціаліста має бути невід'ємною частиною профілактики захворюваності. Це важливо не тільки для вас, але і для безпеки оточуючих людей, наприклад, якщо ви – автоводій. Якщо у вас відсутні проблеми з зором, вам необхідно перевіряти зір один раз на два роки.

Сучасні технології надають можливість будь-кому бажаючому перевірити зір, не виходячи з дому. Така діагностика необхідна у таких випадках:

- людям, які працюють за комп'ютером, тривалий час або діяльність яких пов'язана з дрібними деталями;
- при швидкій втомі очей;
- у разі почервоніння, слъозогінності, печіння або сухості органів зору;
- при частих головних болях під час читання;
- у разі появи «мушок» або завіси перед очима.

Інтерактивна перевірка зору online надає можливість комфортно та легко стежити за власним зором, а у разі його погіршення ви зможете вчасно звернутися до лікаря-офтальмолога та отримати додаткову консультацію.

За останні десятиліття покращилася точність ранньої діагностики катаракти. Органи зору обстежуються не лише за допомогою офтальмоскопії, але й із застосуванням сканування та бінокулярної мікроскопії. Дані дослідження дозволяють встановити правильний діагноз і розрахувати параметри для інтраокулярних лінз.

Діагностика із застосуванням сучасних технічних засобів є точною та інформативною. Комп'ютерне обстеження зоре надає можливість підібрати оптимальну тактику лікування, у тому числі лазерними методами.

Комп'ютерне обстеження зору допомагає виявити майже будь-які патології та захворювання очей. Для комплексного обстеження всіх структур ока використовуються прилади, які збільшують зображення в 60-90 разів, при цьому всі дослідження безконтактні та безболісні.

Діагностування стану зору – найважливіша частина турботи про здоров'я очей, адже вчасне виявлення патологій та відхилень зорового апарату надасть можливість ефективно усунути виявлені порушення і повернути пацієнта до повноцінного життя.

1.2 Види і методи діагностики зору

1.2.1 Гострота зору

На початку 19-го століття для проведення діагностики гостроти зору використовувалися таблиці, які містять спеціально підібрані значки – оптотипи. У ролі оптотипів можуть виступати як цифри, смужки, літери, гачки, так і малюнки, які зазвичай використовуються для дошкільнят. Пацієнтам необхідно розпізнавати ці позначення та відрізнити окремі елементи.

У наш час найпопулярнішими таблицями для перевірки зору є Головіна-Сівцева, яку використовують для старшої вікової категорії і таблиця Орлової, котру використовують для проведення діагностики у дітей раннього віку. Кожна з таблиць складається з 12-ти рядів оптотипів, які зменшуються зверху вниз [3].

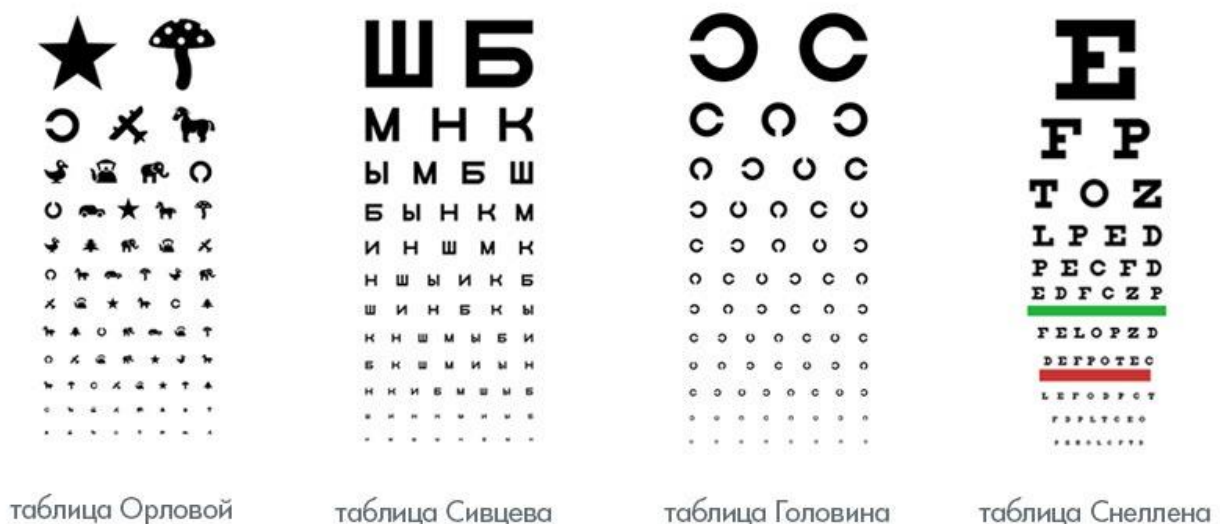


Рисунок 1.1 – Найбільш популярні таблиці для перевірки зору

1.2.2 Кольоровідчуття

Під терміном «кольоровідчуття» йдеться про здатність людини розрізняти кольори. При патологічній зміні зорової функції спостерігається неправильне сприймання кольорів. Це порушення можуть бути як вродженими, так і придбаними.

Для визначення, чи є у людини порушення кольоровідчуття, офтальмологи проводять ряд досліджень. Найчастіше використовуються поліхроматичні таблиці Флетчера-Гамблінга, Ішихари, Штиллінга та інші [4]. Також широко застосовуються тести Рабкіна [5], які проходять усі водії транспортних засобів.

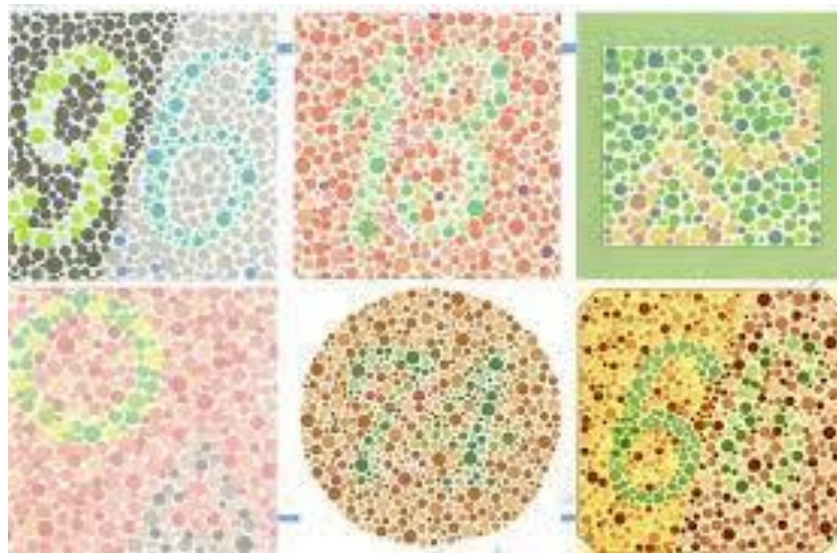


Рисунок 1.2 – Приклади тестів Рабкіна

Всі перелічені методи є однаковими за принципом дії та представлені у вигляді малюнків з точок або кіл різного діаметру і відтінків, візуально можуть нагадувати мозаїку. Якщо уважно подивитись на картинку, то крізь основний фон буде проглядатись певна картинка, котра виконана в інших тонах. Люди, у яких присутня патологія, яка стосується кольоровідчуття, не розглянуть, що зображено на малюнку.

За ступенем сприйняття кольору розрізняють:

- трихромати (норма);
- протоаномалів (людей з порушеннями кольоросприйняття в червоному спектрі);

– дейтераномалія (людей з порушенням сприйняття кольорів зеленого сектору).

Патології кольоровідчуття поділяються на дві основні групи: вроджені патології та набуті. Вродженні порушення, які, як правило, поширюються на обидва ока. Діагностувати їх можна тільки за допомогою проведення спеціального дослідження. Дальтонізм не впливає на зниження якості або втрату інших зорових функцій. Переважно вроджені аномалії передаються у спадок. Такі люди сприймають лише два кольори, але в іншій пропорції, ніж трихромати.

Види вродженої патології:

- дейтераномалія (слабо сприймається зелений відтінок);
- протаномалія (червоний колір майже непомітний);
- тританомалія (непомітний синій відтінок);
- дихромазія (зорові рецептори зовсім не сприймають один з трьох відтінків);
- монохромазія («колірна сліпота», людина бачить все тільки в чорно-білому кольорі).

Дальтонізм – патологія, яка названа на честь вченого Джона Дальтона [6], який особисто страждав порушенням кольоровідчуття з дитинства. Набуті патології кольоровідчуття зазвичай є наслідком захворювань зорового нерву, центральної нервової частини або сітківки ока. Патологія може поширитись тільки на один або на обидва ока одночасно.

Види набутих порушень:

- ксантопсія (все сприймається в жовтому кольорі);
- еротропсія (все сприймається в червоному кольорі);
- цианопсія (все сприймається у синьому кольорі).

На відміну від вродженої патології, яка не піддається корекції, від придбаних аномалій можна позбутися, якщо усунути причину захворювання.

1.2.3 Контрастна чутливість

Контрастна чутливість характеризує здатність людського зору бачити об'єкти, які мало відрізняються по світлості або яскравості від фону. Контрастна чутливість є одним з найважливіших показників функцій зору, особливо при слабкому освітленні, у тумані або при відблисках, коли контраст між об'єктами та їх фоном зменшується. Водіння автомобіля в нічний час є прикладом діяльності, для безпечного здійснення якої необхідна хороша контрастна чутливість.

Перевірка контрастної чутливості – це визначення здатності розрізнити світлі об'єкти при поступовому зниженні їхньої яскравості на темному тлі (контраст) [7].

Пеллі-Робсон – найпопулярніший та один з широко використовуваних тестів на контрастну чутливість, у якому використовуються чорні оптотипи (літери) на білому тлі. Розмір букв залишається незмінним, змінюється лише кількість чорного кольору, стаючи сірішим, через що букви стає складніше розібрати. Тест займає приблизно 1 хвилину на кожне око.



Рисунок 1.3 – Тест Пеллі-Робсон

Іноді для діагностування та оцінювання контрастної чутливості зору застосовуються більш складні пристрої. Зокрема, застосовують так звані «мішені» або синусоїдальні ґрати, що складаються з розташованих паралельно

розмитих смужок світлого та темного кольорів. У кожній цілі ці грати відрізняються по ширині (просторовій частоті) і контрасту, що надає можливість максимально точно оцінити, наскільки очі пацієнта розрізняють зміну контрасту.

На деяких пристроях із синусоїдальною решіткою в очі спрямовується джерело яскравого світла, що імітує відблиски, подібні, наприклад, до світла фар зустрічного транспорту під час водіння в нічний час.

Діагностика контрастної чутливості зазвичай проводиться після стандартної оцінки гостроти зору перед розширенням зіниць.

1.2.4 Дуохромний тест

Дуохромний тест – найбільш об'єктивний та ефективний метод діагностики рефракції ока. Тест засновано на явищі, при якому промені з різною довжиною хвилі (різного кольору) по-різному заломлюються оптичною системою ока. Сенс дослідження полягає у тому, щоб виявити, на якому тлі (червоному або зеленому) представлені символи будуть більш чіткими [8].

При відсутності патологій (еметропії) око розрізнятиме однаково чітко оптотици на червоному та зеленому фоні.

У разі, коли символи здаються більш чіткими на червоному фоні, є велика ймовірність наявності короткозорості у пацієнта. В протилежному разі, коли символи здаються чіткішими на зеленому фоні, є велика ймовірність наявності далекозорості.

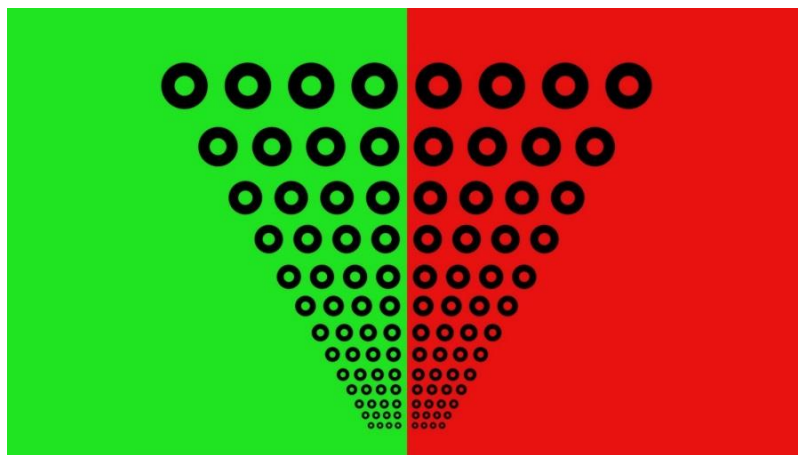


Рисунок 1.4 – Дуохромний тест

1.2.5 Перевірка зору на наявність астигматизму

Астигматизм – захворювання очей, при якому фізичні зміни у очному яблуці призводять до порушення рефракції (заломлення світла). При відсутності даного захворювання світловий промінь, проходячи оптичну систему ока, чітко фокусується в одній точці на сітківці, даючи чітке зображення.

Головною причиною виникнення астигматизму є неправильна (не сферична) форма рогівки (рідше кришталика). Рогівка та кришталик здорового ока мають рівну сферичну поверхню. При наявності астигматизму сферичність порушена. Вона має різну кривизну за різними напрямками. Відповідним чином, при наявності астигматизму пацієнт замість нормального зображення бачить спотворене, у якому одні лінії чіткі, а інші розмиті [9].

Одним з найпопулярніших методів самодіагностування є «Зірка Сіменса». Цей метод діагностування являє собою тест-картинку, на якому зображено коло, що складається зі смужок (променів), які звужуються всередину до центра кола. Діаметр кола становить 10 сантиметрів. 54 промені чорного кольору на білому фоні утворюють різноманітні ілюзії сприйняття, завдяки розшифровки яких можна в домашніх умовах припустити наявність патології, після чого необхідно обов'язково звернутися за більш точною консультацією та вірною інтерпретацією результатів до відповідного фахівця.

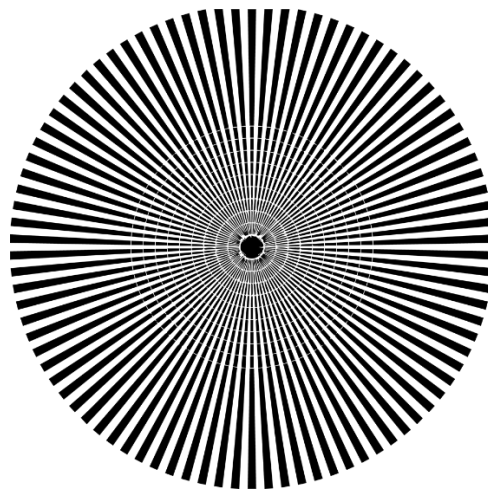


Рисунок 1.5 – Зірка Сіменса

Зірка Сіменса надає чудову можливість відстежити, що гострота зору постійно змінюється, причому ці зміни частково підкоряються вольовому контролю [10].

1.2.6 Тест Амслера (патологія макули)

Вікова макулодистрофія (ВМД) - найпоширеніша і найнебезпечніша патологія центральної області сітківки (жовтої плями, макули). Вона є головною причиною втрати центрального зору і соціальної сліпоти у людей старше 50 років у всьому світі, незалежно від рівня медицини і рівня фінансового благополуччя конкретного пацієнта [11].

Тест (решітка) Амслера вигадав офтальмолог Марк Амслер [12]. Основною метою даного тесту є швидка перевірка наявності дистрофічних змін у центральному зору пацієнта. Тест Амслера являє собою решітку, розділену на маленькі квадратики однакового розміру. У центрі сітки розташована чорна цятка.

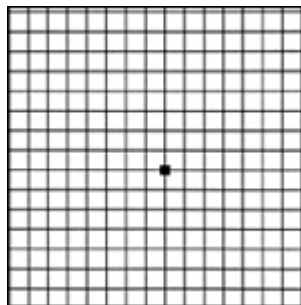


Рисунок 1.6 – Сітка Амслера у нормі

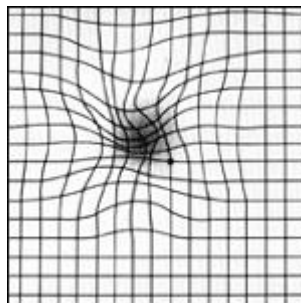


Рисунок 1.7 – Сітка Амслера при патології центральної зони сітківки

Тест Амслера – найстаріший спосіб, яким на сьогоднішній день користуються для самостійного діагностування та спостереженням за перебігом макулодистрофії.

1.3 Опис та аналіз наявних аналогів

Інтерактивна перевірка зору online надає можливість комфортно та легко стежити за власним зором, а у разі його погіршення ви зможете вчасно звернутися до лікаря-офтальмолога та отримати додаткову консультацію.

Фактори, які впливають на оцінку якості системи для діагностики зору:

– функціональність. Система має надавати користувачеві змогу здійснювати перевірку не лише гостроти зору, а й перевірку на сприйняття кольорів, тест Амслера і т.п.;

– точність. Система має надавати максимально точний результат. У випадку перевірки гостроти зору може бути використана методика таблиці Сівцева;

– доступність. Це зручний і зрозумілий користувачеві інтерфейс, швидка обробка результатів тестування.

– візуальна привабливість. Система має бути виконана у єдиному, лаконічному дизайні з використанням спокійних кольорів, для проведення комфортної діагностики зору.

1. ESSILOR (рис. 1.8)

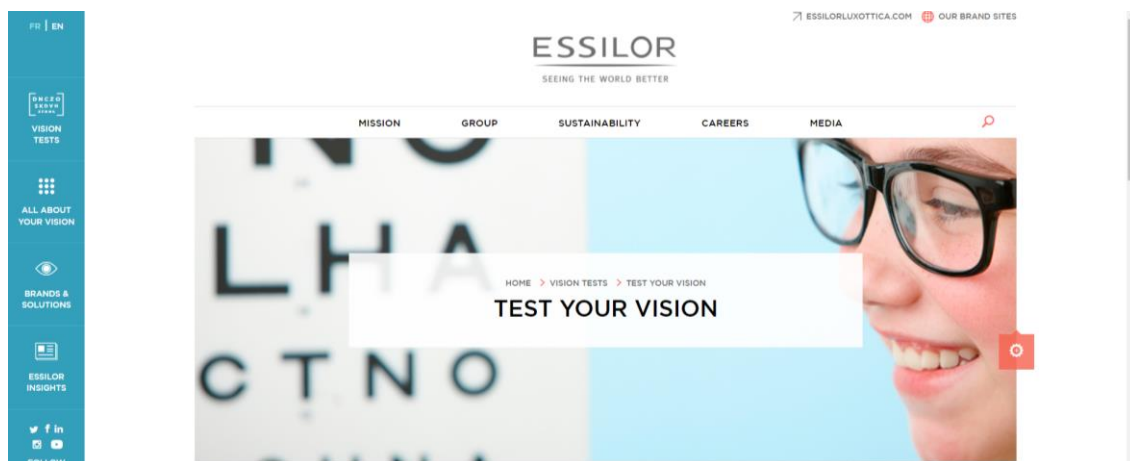


Рисунок 1.8 – Зовнішній вигляд сайту ESSILOR

Переваги:

- лаконічний дизайн у нейтральних, спокійних кольорах;
- можливість комплексної перевірки зору;
- досить зрозумілий користувацький інтерфейс.

Недоліки:

- після проходження комплексної перевірки відсутній загальний висновок.

2. ZEISS (рис. 1.9)

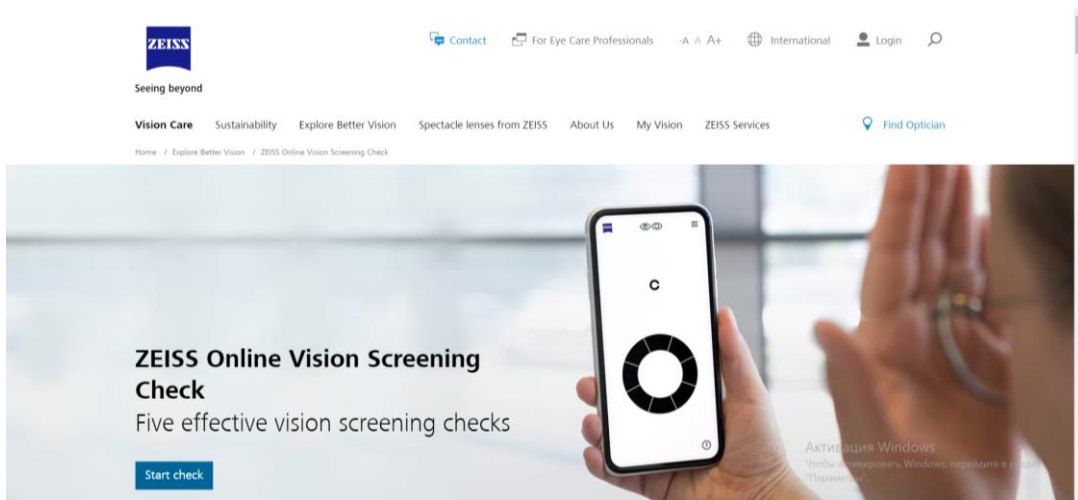


Рисунок 1.9 – Зовнішній вигляд сайту ZEISS

Переваги:

- лаконічний дизайн у нейтральних, спокійних кольорах;
- анімований інтерфейс інструкції проведення діагностики, який покращує її зрозумілість;
- відсутність яскравих, відволікаючих елементів на сторінці проходження самодіагностики.

Недоліки:

- перевірка загального стану зору, без коефіцієнту гостроти зору та детального результату.

3. Глазка (рис. 1.10)



Новосибирск
ул. Нарымская, д. 19 • ул. Сибиряков-Гвардейцев, д. 23
Ежедневно 8:00-20:00 без обеда и выходных

8 (383) 247-9471

ЗАКАЗАТЬ ЗВОНОК

ЗАПИСАТЬСЯ НА ПРИЕМ

О КОМПАНИИ УСЛУГИ И ЦЕНЫ НАШИ СПЕЦИАЛИСТЫ АКЦИИ ТЕСТ ЗРЕНИЯ ЛЕЧЕНИЕ ЗАБОЛЕВАНИЙ КОНТАКТЫ

ПОЛНАЯ ДИАГНОСТИКА СТАБИЛИЗАЦИЯ БЛИЗУРОКУСТИ НОЧНЫЕ ЛИНЗЫ ДИАГНОСТИКА ДЕТЕЙ

Главная - Тест зрения

Проверка зрения online

ПОЛЕЗНЫЕ СТАТЬИ

СОВЕТЫ И ПАМЯТКИ

ГЛАЗКА НА YOUTUBE

ПРОВЕРКА ЗРЕНИЯ
ОНЛАЙН

ЗАПИСКИ ВРАЧА
ОФТАЛЬМОЛОГА



Тест на проверку остроты зрения

Таблица Сивцева представляет собой стандартный набор печатных знаков для определения остроты зрения человека. Таблица Сивцева состоит из комбинации семи разных букв разного размера, но равных по своей ширине и высоте. В соответствии с условно принятой нормой, глаз со 100%-ным зрением (V=1.0) способен различать две удаленные точки с угловым разрешением в 1 минуту (или 1/60 градуса).

ПОДРОБНЕЕ



Тест на цветовосприятие

После того, как в интернете вызвало бурные споры цвет платья, которое одним казалось сине-чёрным, а другим – бело-золотым, стало понятно, что не все мы видим мир

Активация Windows

Чтобы активировать Windows, перейдите к...

Рисунок 1.10 – Зовнішній вигляд сайту Глазка

Преваги:

- можливість повного діагностування стану очей;
- різноманітні тести для виконання самодіагностики.

Недоліки:

- тести не інтерактивні;
- результат проходження тесту не індивідуальні – з загального переліку варіантів результатів необхідно самостійно обрати більш підходящий вам результат.

4. НЬЮ ВІЖН (рис. 1.11)

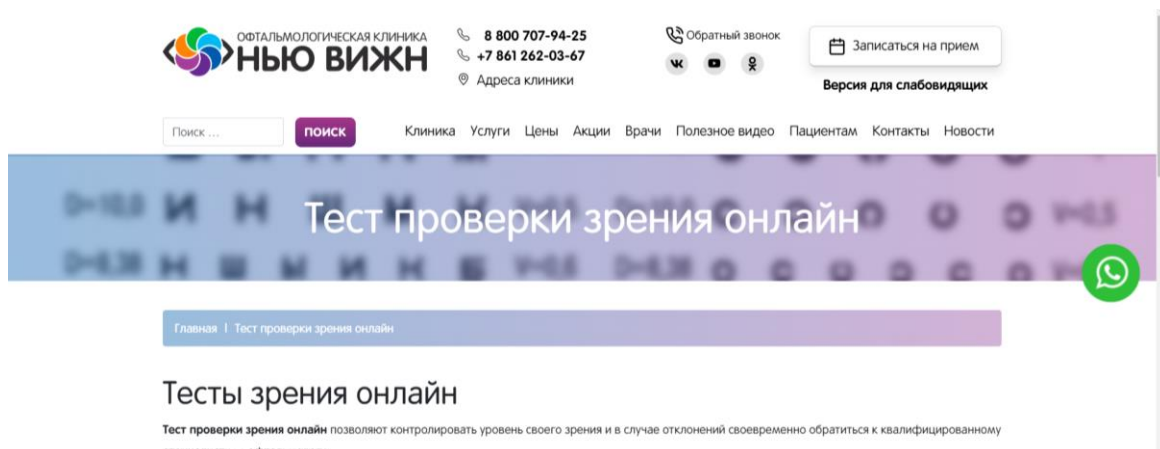


Рисунок 1.11 – Зовнішній вигляд сайту НЬЮ ВІЖН

Преваги:

2022 р.

Клішевська А.В.

122-БКР-401.21810114

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

- комплексне діагностування;
- можливість перегляду допущених помилок;
- індивідуальна оцінка стану зору.

Недоліки:

- відсутність перевірки гостроти зору;
- не досить зручний процес проведення самодіагностики (спочатку необхідно приблизитися до монітору, щоб перечитати можливі варіанти відповідей, а потім повертатися до картинки, яка призначена для діагностування).

5. ЛАЗЕР ПЛЮС (рис. 1.12)

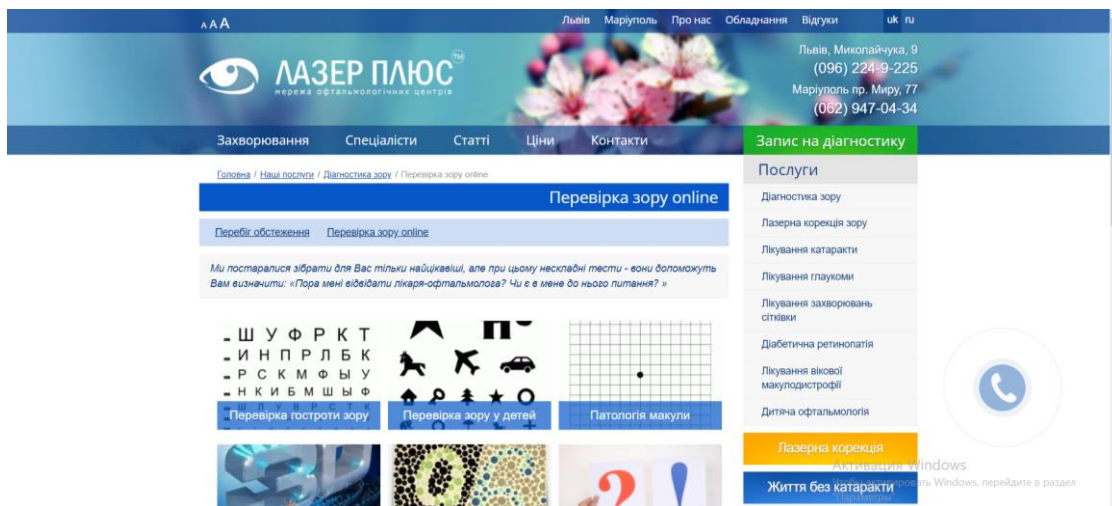


Рисунок 1.12 – Зовнішній вигляд сайту ЛАЗЕР ПЛЮС

Переваги:

- комплексне діагностування;
- можливість проведення діагностики зору у дітей;
- можливість перевірки уважності;
- лаконічний інтерфейс.

Недоліки:

- сторінка з тестом «А ви бачите у 3D?» не існує;
- відсутня інструкція проведення діагностики зору у дітей;
- не інтерактивні тести;

– результат проходження тесту не індивідуальні – з загального переліку варіантів результатів необхідно самостійно обрати більш підходящий вам результат.

6. Santen (рис. 1.13)

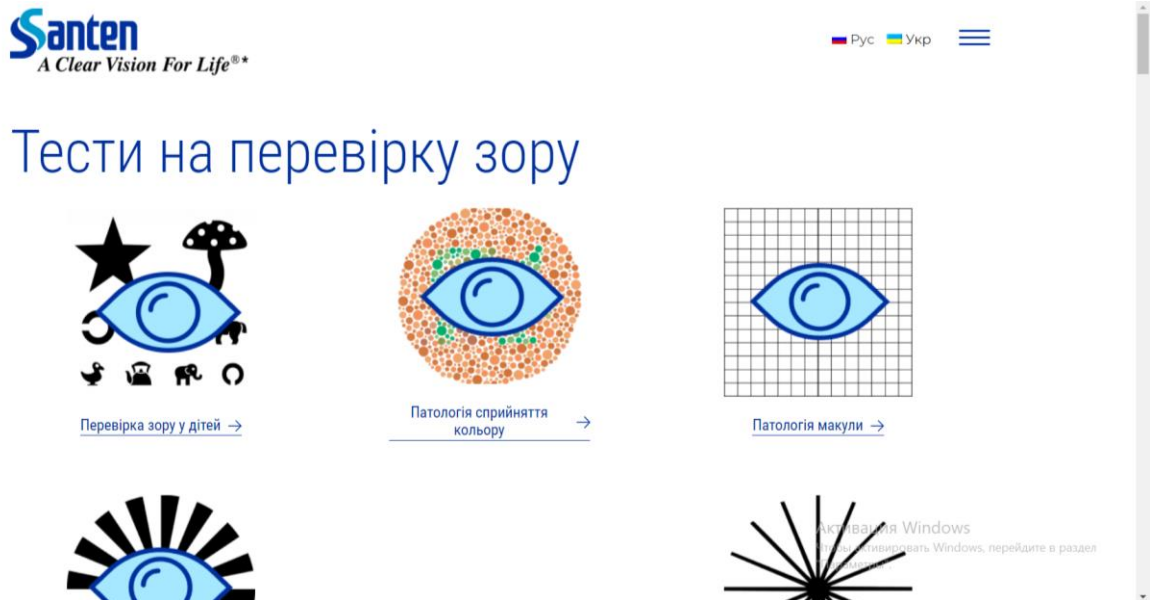


Рисунок 1.13 – Зовнішній вигляд сайту Santen

Переваги:

- різноманітність тестів для проведення діагностування;
- можливість проведення діагностики зору у дітей;
- наявність обширної, корисної інформації про усі тестування;
- лаконічний інтерфейс.

Недоліки:

- не інтерактивні тести;
- результат проходження тесту не індивідуальні – з загального переліку варіантів результатів необхідно самостійно обрати більш підходящий вам результат.

7. Ваш Зір (рис. 1.14)

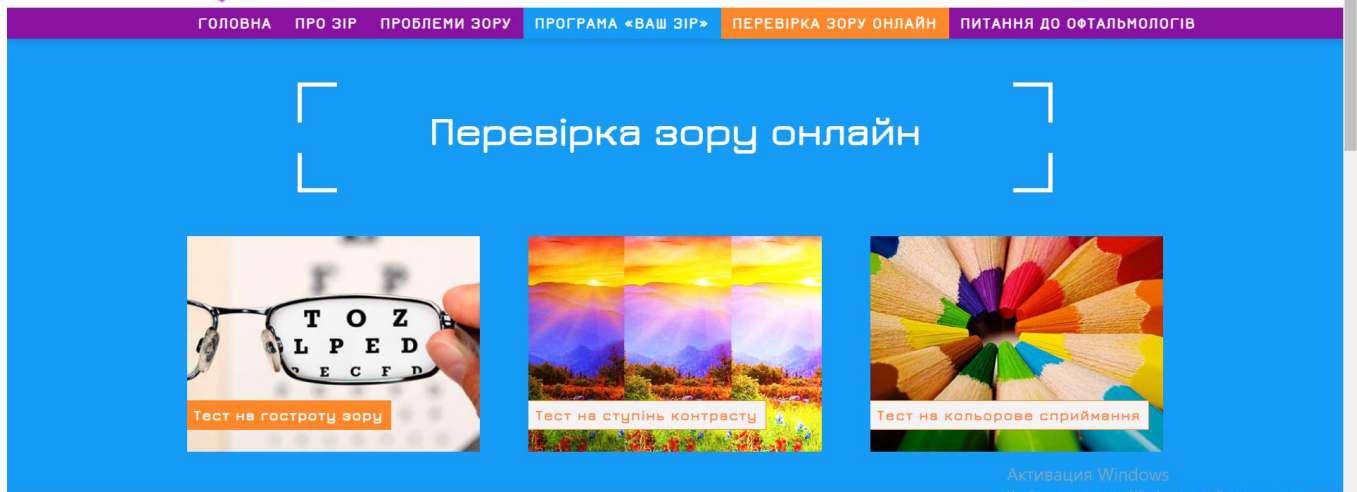


Рисунок 1.14 – Зовнішній вигляд сайту Ваш Зір

Переваги:

- наявність вибору тестів для проведення діагностування;
- зрозумілість принципу проведення тестування;
- індивідуальний результат у відсотковій шкалі;
- якісний тест на визначення ступеню контрасту.

Недоліки:

- досить тривала затримка зміни картинок у тесті на колірне сприйняття;
- у тесті з перевіркою на гостроту зору розміщено дуохромний тест для визначення короткозорості(-) і далекозорості(+);
- відволікаючий інтерфейс.

1.4 Постановка задачі

Тому згідно з актуальністю теми діагностики зору об'єктом дослідження процеси та інформаційні технології діагностики захворювань, а саме найвний байєсівський класифікатор, який є широко застосовуваним у медицині. Згідно з цим, предметом дослідження є методи діагностики зору для запобігання захворювань очей.

Мета: діагностика зору у різних напрямках (гострота, кольоросприйняття, ступінь контрасту, тощо) для профілактики захворювань очей на основі розробленої системи.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати сучасний стан задачі діагностування зору;
- окреслити існуючі методи діагностування зору;
- проаналізувати існуючі аналоги;
- реалізувати систему діагностування зору для профілактики захворювань очей з використанням популярних фреймворків та бібліотек.

Висновки до розділу 1

Останніми роками лікарі констатують загальний рівень погіршення гостроти зору не тільки серед дорослих і літніх людей, а й серед молодого покоління та підлітків. На сьогоднішній день в світі налічується близько 134 млн людей, які страждають поганим зором і 34 млн сліпих. За підрахунками ВООЗ, кожні 5 секунд сліпне одна доросла людина і щохвилини - дитина. При цьому вважається, що 75% випадків можна було б уникнути [2]. Тому, щоб запобігти поповнюванню сумного списку, профілактичний огляд обов'язковий для всіх: для дітей - починаючи з 6 місячного віку, для дорослих - не рідше одного разу на рік. Варто не нехтувати першими ознаками погіршення стану зору (підвищену стомлюваність очей, почервоніння та подразнення повік) та своєчасно проконсультуватися у лікаря-офтальмолога. Перевірити зір, як правило, можна у найближчому салоні оптики. Якщо така можливість відсутня, на допомогу

прийдуть спеціальні онлайн-тести для самостійної перевірки зору та матеріали, доступні в інтернеті для кожного.

Тести для самостійної перевірки зору в домашніх умовах нічим не відрізняються від тих, якими користуються спеціалісти в салонах оптики, поліклініках та медичних центрах. Дві основні умови проходження тестів - досить інтенсивне освітлення кімнати (в ідеалі природне денне світло) і гарне самопочуття. Перевірка зору при головному болю, підвищеній температурі або надмірній втомі може показати спотворені результати. Тьмяне штучне світло в темний час доби додатково напружує очі - результати такого тесту можуть бути необ'єктивними.

Проведене дистантне обстеження зору дозволяє зробити орієнтовний висновок про стан органів зору. Проте така діагностика не може стати повноцінною заміною консультації лікаря та медичного обстеження. За результатами тесту людина отримує узагальнені відомості щодо показників зорових функцій. Будь-які відхилення – привід звернутися до фахівця.

2 МОДЕЛІ, МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Використання наївного байєсівського класифікатора в розробці системи

Наївний байєсівський класифікатор – алгоритм, який для виконання класифікації об'єктів застосовує теорему Байєса. Naive Bayes класифікатор завбачує сильну або наївну незалежність між атрибутами точок даних. Наївний класифікатор є досить популярним при створенні спам-фільтрів, аналізах тексту, медичній діагностиці та для машинного навчання, через те, що його досить легко реалізувати [13].

Naive Bayes класифікатор вважається наївним саме через його припущення. Під припущенням розуміється, що всі атрибути розглянутої точки даних є незалежними один від одного. Класифікатор, який буде виконувати сортування яблук та апельсинів знає, що яблука мають червоний колір, круглу форму та певний розмір, але не припускатиме всі ці факти одночасно. Врешті-решт, апельсини мають теж круглу форму.

Байєсівський класифікатор – являється не єдиним алгоритмом, а цілим сімейством алгоритмів машинного навчання, які застосовують статичну незалежність. Ці алгоритми є досить простими для запису та виконуються більш ефективно, ніж складніші алгоритми Байєса.

Властивості класифікації Naive Bayes:

1. Використання всіх змінних і визначення всіх залежностей між ними.
2. Наявність припущень щодо змінних:
 - 2.2 усі змінні є однаково важливими;
 - 2.3 усі змінні є статистично незалежними, тобто значення однієї змінної нічого не говорить про значення іншої.

Загальна теорема Байєса має наступний вигляд:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.1)$$

Використовуючи теорему Байєса, ми можемо знайти ймовірність того, що подія **A** станеться, враховуючи, що сталася подія **B**. У цьому випадку **B** — це доказ, а **A** — гіпотеза. Припущення, зроблене тут, полягає в тому, що предиктори/характеристики є незалежними. Тобто наявність однієї особливої ознаки не впливає на іншу. Тому його називають наївним [14].

Теорема Байєса стверджує наступне співвідношення, задане змінною класу та залежним вектором ознак x_1 через x_n :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (2.2)$$

Використовуючи наївне припущення умовної незалежності, що

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y) \quad (2.3)$$

для всіх i це співвідношення спрощується до:

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)} \quad (2.4)$$

Оскільки $P(x_1, \dots, x_n)$ – константа з урахуванням вхідних даних, ми можемо використовувати таке правило класифікації:

$$P(y | x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i | y) \quad (2.5)$$

У нашому випадку змінна класу (y) має лише два результати – так чи ні. Можуть бути випадки, коли класифікація може бути багатоваріантною. Отже, потрібно знайти клас y з максимальною ймовірністю.

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y) \quad (2.6)$$

і ми можемо використовувати максимальну апостеріорну оцінку (Maximum A Posteriori (MAP)) для оцінки $P(y)$ та $P(x_i | y)$.

Різні наївні байєсівські класифікатори відрізняються головним чином за припущеннями, які вони роблять щодо розподілу $P(x_i | y)$.

Не дивлячись на його припущення, яке, на перший погляд, можуть здатися занадто спрощеними, наївний байєсівський класифікатор досить добре справляється в багатьох реальних ситуаціях, як відомо, класифікація документів і фільтрація спаму. Вони вимагають невеликої кількості навчальних даних для оцінки необхідних параметрів.

Наївний байєсівський класифікатор поділяється на декілька типів: мультиномінальний, бернуллі та гаусівський наївний Байєс.

Мультиномінальний наївний Байєс зазвичай використовується для вирішення проблем класифікації документів, тобто чи належить документ до категорії спорту, політики, технологій, тощо. Ознаками/прогнозами, які застосовує даний класифікатор є частота слів, присутніх у документі.

Бернуллі наївний Байєс – є подібним до мультиномінального наївного Байєса, але ознаками/прогнозами є булеві змінні. Параметри, які ми застосовуємо для прогнозування змінної класу, приймають лише значення так чи ні, наприклад, якщо слово зустрічається в тексті чи ні.

Гаусівський наївний Байєс – випадок, коли предиктори беруть безперервне значення і не є дискретними. Припускається, що ці значення відбираються з гаусівського розподілу.

Оскільки спосіб представлення значень у наборі даних змінюється, формула умовної ймовірності змінюється на:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (2.7)$$

Наївні байєсівські учні та класифікатори можуть бути надзвичайно швидкими в порівнянні з складнішими методами. Відокремлення розподілів умовних ознак класу означає, що кожен розподіл може бути незалежно оцінений як одновимірний розподіл. Це, у свою чергу, допомагає полегшити проблеми, пов'язані з різницею у розмірності.

2.2 Аналіз інструментальних засобів розробки системи

Для створення системи було використано: мову програмування C#. Графічний користувальницький інтерфейс реалізовано з використанням інтерфейсу програмування додатків (API) Windows Forms.

У якості середовища розробки було обрано Microsoft Visual Studio. Реалізацію наївного байєсівського класифікатора цілком реально виконати без

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
використання додаткових бібліотек та пакетів, тому при створенні системи обмежимося використанням лише однієї мови програмування.

Мова програмування C# є однією з найпотужніших у наш час та найпопулярніших мов, яка досить швидко та енергійно розвивається і є затребуваною в ІТ-галузі.

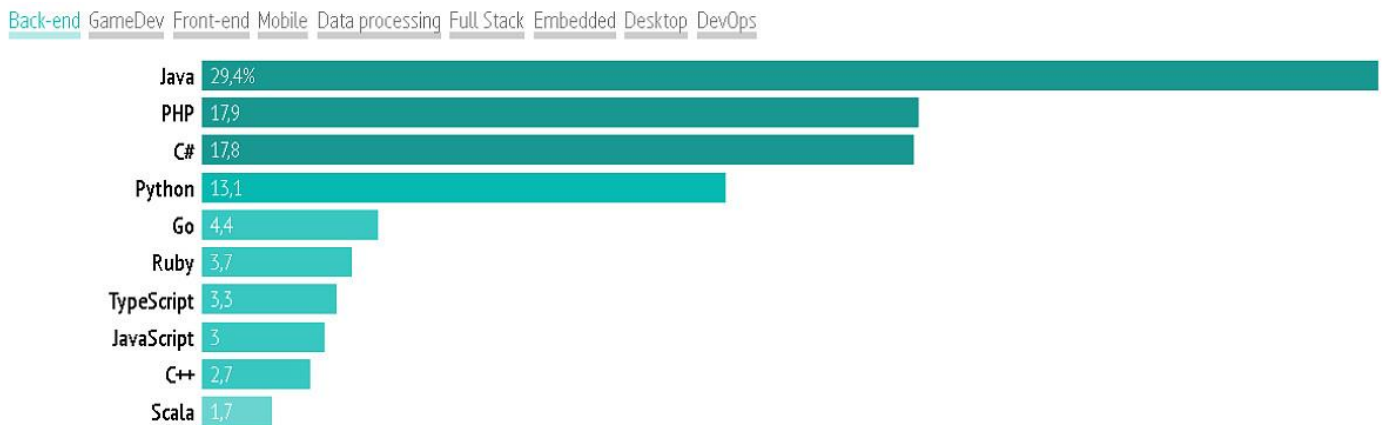


Рисунок 2.1 – Рейтинг мов програмування 2022 року, якими найчастіше користуються українські розробники

На сьогоднішній день з використанням цієї мови реалізуються різні програми: від невеликих десктопних додатків до великих веб-сервісів, які кожного дня обслуговують близько мільйону користувачів.

C#, як і платформа .NET – є досить старою мовою програмування, яка пройшла достатньо великий та довгий шлях. Перша версія C# з'явилася разом із випуском середовища розробки Microsoft Visual Studio.NET у лютому 2002 року.

Актуальною версією мови є версія C# 10.0, яка була випущена у реліз 8 листопада 2021 разом із релізом .NET 6 [15]. C# - мова програмування з Cі-подібним синтаксисом і тому є досить подібною та близькою у цьому відношенні до таких популярних та поширених мов, як C++ та Java.

Затим, якщо вам знайомий синтаксис однієї з вищенаведених мов, то вивчити та зрозуміти мову C# буде досить просто. Також C# є об'єктно-орієнтованою і в цьому сенсі було багато запозичено та перейнято у таких мов, як Java та C++.

Для прикладу, С# піддержує поліморфізм, успадкування, перевантаження операторів, статичну типізацію.

Об'єктно-орієнтований підхід програмування дозволяє вирішувати такі завдання, як побудови великих, але в той же час гнучких, масштабованих та легко розширюваних додатків. С# продовжує активно розвиватися і з кожною новою версією з'являється все більше цікавих функціональностей.

Коли згадується «С#», часто припускають саме технології платформи .NET (найвідоміші з яких: Windows Forms, WPF, ASP.NET, Xamarin). І у протилежному випадку, коли говорять .NET, досить часто мають на увазі С#. Хоч ці поняття і пов'язані, ототожнювати їх так не можна, адже це зовсім різні речі. Мова програмування С# була розроблена спеціально для взаємодії з фреймворком .NET, проте саме поняття .NET є дещо ширшим.

Одного разу Білл Гейтс констатував, що фреймворк .NET – це найкраще, що розробила компанія Microsoft. Здається, він мав рацію. Платформа .NET є дуже потужним фреймворком для написання та розробки додатків. Можна виділити такі основні характеристики, які є досить важливими і мають бути врахованими:

– підтримка кількох мов. Джерелом та основою фреймворку є загальномовне середовище виконання Common Language Runtime (CLR), дякуючи якому .NET може підтримувати одночасно декілька мов: поряд з С# це також VB.NET, С++, F#, а також різні діалекти інших мов, які є прив'язаними до .NET, наприклад, Delphi.NET. При виконанні та компіляції код будь-якою з цих мов виконується у складання загальною мовою CIL (Common Intermediate Language), яка свого роду є асемблером платформи .NET. Тому за певних умов ми можемо зробити окремі модулі однієї програми окремими мовами;

– кросплатформність. .NET є фреймворком, який переноситься (але з деякими обмеженнями). Для прикладу, останній реліз фреймворку на даний момент – .NET 6 підтримується на більшій частині сучасних операційних систем Windows, MacOS, Linux. Застосовуючи різноманітні технології на фреймворці

.NET, можна створювати застосунки мовою C# для різноманітних платформ - Windows, MacOS, Linux, Android, iOS, Tizen;

– досить могутня та потужна бібліотека класів. .NET представляє всім мовам єдину підтримувану бібліотеку класів. Пишучи будь-яку програму мовою C#, такі як: текстовий редактор, чат або складний веб-сайт - так чи інакше буде застосовуватися бібліотека класів .NET;

– багатоманітність технологій. Загальномовне середовище виконання CLR та базова бібліотека класів є основою цілого стеку технологій, які розробники можуть задіяти при побудові тих чи інших додатків. Для прикладу, для того, щоб працювати з базами даних у цьому стеку технологій призначено технологію ADO.NET та Entity Framework Core. Для того, щоб побудувати графічні програми з насиченим інтерфейсом використовується технологія WPF і WinUI, для розробки простіших графічних програм - Windows Forms. Для розробки кросплатформних мобільних та десктопних застосунків зазвичай використовують такі технології, як Xamarin/MAUI. Для написання веб-сайтів та веб-додатків застосовують широковідомий ASP.NET. До вище згаданих технологій слід додати ще активний Blazor - фреймворк, який стрімко розвивається і набирає популярність, працюючи поверх .NET і дозволяючи розробляти веб-додатки як на стороні сервера, так і на стороні клієнта. А в майбутньому зможе підтримувати розробку мобільних програм і, можливо, навіть десктоп-застосунків;

– продуктивність. Згідно з рядом тестів веб-програми на .NET 6 у ряді категорій динамічно переганяють веб-програми, які побудовано з застосуванням інших технологій. Застосунки та програми на .NET 6 відрізняються високою продуктивністю.

Так само слід згадати таку особливість мови C# і платформи .NET, як машинальне збирання сміття. Це має на увазі, що здебільшого не доведеться, на відміну від C++, піклуватися про звільнення пам'яті програми. Вищезгадане

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
загальнономвне середовище CLR самостійно викличе збирання сміття та очистить пам'ять.

Варто відмітити, що платформа .NET досить довгий час розвивалася переважно як платформа для Windows під назвою .NET Framework. У 2019 році вийшла фінальна версія цієї платформи - .NET Framework 4.8. На цьому розвиток платформи було завершено.

З 2014 року Microsoft почав динамічно виробляти нову альтернативну платформу - .NET Core, яка вже мала застосовуватися для різних платформ і повинна була містити у собі всі можливості застарілого .NET Framework і приєднати нову функціональність.

Потім Microsoft послідовно випустив ряд версій цієї платформи: .NET Core 1, .NET Core 2, .NET Core 3, .NET 5. І поточною версією є платформа .NET 6. Тому слід розуміти різницю між .NET Framework, який призначений переважно для Windows, і кросплатформений .NET 6.

Для написання та розробки системи з застосуванням мови C# використано середовище Microsoft Visual Studio. Програмне забезпечення Microsoft Visual Studio – вбудоване середовище розробки (IDE).

Microsoft Visual Studio охоплює в собі набір інструментів, хмарних сервісів та розширень, які потрібні програмістам та розробникам для розробки програм та ігор. Можна застосовувати продукти Visual Studio з інструментами, що вже існують. Visual Studio допомагає редагувати, налагоджувати та створювати код, а потім публікувати програму.

Microsoft Visual Studio включає в собі такі компоненти, як: компілятори, засоби автодоповнення коду, візуальні редактори макетів і багато інших функцій, що значно полегшують процес написання програмного забезпечення.

Microsoft Visual Studio — серія продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення та інші інструменти для розробників.

Ці продукти надають можливість створювати як консольні додатки, так і ігри та програми з графічним інтерфейсом, у тому числі з підтримкою такої відомої технології, як Windows Forms, UWP а також веб-сайти, веб-програми, веб-служби як в рідному, так і в керованому кодах всіх платформ, які підтримують Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework та Silverlight. Після того, як корпорація Microsoft придбала Xamarin з'явилася можливість створювати OS та Android програми.

Visual Studio вміщує в собі редактор вихідного коду з підтримкою технології IntelliSense і надає можливість достатньо простого процесу рефакторингу коду.

Вбудований налагоджувач може працювати як налагоджувач рівня вихідного коду, так і відладчик машинного рівня. У деяких випадках вбудовані інструменти вміщують редактор форм для більш простого створення графічного інтерфейсу застосунку, веб-редактор, дизайнер класів і дизайнер схеми бази даних.

Visual Studio надає можливість розробляти та підключати сторонні доповнення (плагіни) для того, щоб розширювати функціональність майже на будь-якому рівні, охоплюючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion та Visual SourceSafe), додавання нових наборів інструментів (для прикладу, інструменти для редагування та візуального проектування коду предметно-орієнтованими мовами програмування) або інструментами для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

2.3 Аналіз інструментальних засобів розробки графічного інтерфейсу системи

Графічний інтерфейс користувача (ГІК, англ. Graphical user interface) – це вид інтерфейсу, який надає можливість користувачам взаємодіяти з

електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації [16].

Виконання дій у ГІК — є безпосередньою взаємодією з графічними елементами. GUI використовується не тільки на комп'ютерах, а і у мобільних пристроях, таких, як мобільні телефони, планшети, електронні книги, портативні медіапрогравачі тощо.

Термін ГІК переважно не використовують стосовно інтерфейсів з низькою роздільною здатністю. Як приклад, у відеоіграх використовують інтерфейс HUD [16].

Вперше концепцію ГІК була запропонованою вченими з дослідницької лабораторії Xerox PARC у 1970-х, але комерційне втілення було отримано лише в продуктах корпорації Apple Computer.

В операційній системі AMIGAOS ГІК з багатозадачністю був використаний у 1985 році. Нині цей інтерфейс - стандартна складова частина більшості доступних на ринку операційних систем і застосунків. [16]

Для розробки графічних інтерфейсів з застосуванням платформи .NET використовуються різноманітні технології: Windows Forms, WPF, UWP. Однак найбільш простою і зручною платформою досі залишається Window Forms або ще скорочено називають WinForms.

Windows Forms – це платформа інтерфейсу користувача для створення класичних додатків Windows. Вона забезпечує один з найефективніших способів створення класичних програм за допомогою візуального конструктора в Visual Studio. Такі функції, як розміщення візуальних елементів керування шляхом перетягування, полегшують створення класичних додатків.

У Windows Forms можна створювати програми з достатньо складним графічним інтерфейсом, які легко розгортати, додавати нові функціонали, і з якими зручно працювати як в автономному режимі, так і в мережі. Програми

Windows Forms також мають змогу отримувати доступ до локального обладнання та файлової системи комп'ютера, на якому працює програма.

2.4 Обґрунтування вибору СКБД для системи

При реалізації системи для діагностики зору для профілактики захворювань зору було вирішено використати СКБД для зберігання результатів тестувань користувача, які будуть використовуватися для подальшого аналізування та відображення статистики. Це зроблено для того, щоб користувач міг відслідковувати те, як змінюється стан його зору.

При виборі СКБД виявлено, що найважливішими критеріями, які необхідно врахувати є:

- тип аналізованих даних;
- кількість даних;
- фокус інженерної команди;
- швидкість отримання інформації.

При врахуванні усіх вище перелічених критеріїв було виявлено, що найбільш підходящою СКБД є SQL Server.

SQL Server – є простою в установці та працює без особливих налаштувань. При підходящому підході вона здатна гнучко налаштовуватись під усі необхідні потреби розробника.

Також, інтеграція з SQL Server є у будь-якій мові програмування та фреймворку, що однозначно є великою перевагою. Якщо необхідно зберігати дані розміром до 1 ТБ, то саме з SQL Server можливо отримати ефективну та швидку продуктивність обробки даних.

Оскільки дані у таблиці є структурованими та невеликими за розміром SQL Server прекрасно впорається з усіма необхідними та поставленими задачами.

SQL Server є реляційною базою даних. Реляційні бази даних мають ще одну значну перевагу: для запиту до таблиць такої БД можна використовувати мову SQL.

SQL є добре відомою мовою не тільки для аналітиків, але і для інженерів, цю мову значно легше вивчати, ніж більшість мов програмування. SQL Server Management Studio (SSMS) – є основним інструментом, який використовують для адміністрування ядра СКБД та написання коду Transact-SQL. Він розміщений в оболонці Visual Studio.

Entity Framework надає спеціальну об'єктно-орієнтовану технологію на базі фреймворку .NET для роботи з даними. Якщо традиційні засоби ADO.NET дозволяють створювати підключення до бази даних, запити, команди та інші об'єкти для взаємодії з базами даних, то Entity Framework є більш високим рівнем абстракції, який надає можливість інженеру абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища.

Якщо фізично ми оперуємо таблицями, індексами, первинними та зовнішніми ключами, тільки на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

Найперша версія Entity Framework – 1.0 з'явилася ще у 2008 році та являла собою досить обмежену функціональність, базову підтримку ORM (object-relational mapping – відображення даних на реальні об'єкти) та один єдиний підхід до взаємодії з БД – Database First.

З появою версії 4.0 в 2010 році прийшло багато змін - з цього часу Entity Framework став рекомендованою технологією для доступу до даних, а в сам фреймворк були введені нові можливості взаємодії з БД - підходи Model First та Code First [17].

Висновки до розділу 2

Отже, проаналізувавши інструментальні засоби створення системи, для реалізації системи діагностики зору для профілактики захворювань очей було вирішено використати: мову програмування C# та платформу .NET.

Графічний користувальницький інтерфейс реалізовано з використанням інтерфейсу програмування додатків (API) Windows Forms. У якості середовища розробки було обрано Microsoft Visual Studio.

При виборі СКБД та врахуванні усіх необхідних критеріїв було виявлено, що найбільш підходящою СКБД є MySQL. Адже MySQL – є простою в установці та працює без особливих налаштувань.

При належному підході вона здатна гнучко налаштовуватись під потреби розробника. Також, інтеграція з MySQL є у будь-якій мові програмування та фреймворку, що однозначно є великою перевагою. Якщо у вас менше 1 ТБ даних, з MySQL ви отримаєте хорошу продуктивність. Оскільки дані у таблиці є структурними та невеликими (до 1-2 гігабайтів) MySQL чудово впорається з поставленими задачами.

Для роботи з СКБД використано середовище Microsoft SQL Server Management Studio. SQL Server Management Studio (SSMS) – це основний інструмент для адміністрування ядра СКБД та написання коду Transact-SQL. Він розміщується в оболонці Visual Studio.

3 МОДЕЛЮВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІС**3.1 Структура бази даних**

При реалізації системи для діагностики зору для профілактики захворювань зору було вирішено створити БД vision_diagnostics_db для зберігання результатів тестувань користувача, які будуть використовуватися для подальшого аналізування та відображення статистики.

База даних vision_diagnostics_db має дві таблиці: dbo.customers та dbo.diagnostic. У таблиці dbo.customers зберігаються дані користувача, отримані при реєстрації та які необхідні для здійснення авторизації. У таблиці dbo.diagnostic зберігаються такі дані: ім'я користувача, дата проведення діагностики, назва виду діагностики, результат лівого ока, результат правого ока, загальний результат та загальний висновок про стан зору.

Загальна структура таблиць виглядає наступним чином (див. табл. 3.1).

Таблиця 3.1 – Структура таблиць БД

Назва таблиці	Ім'я поля	Тип даних	Розмір поля	Ключ
dbo.customers	id	int	від 0 до 4294967295	Первинний
	login	varchar(max)	від 0 до 65535	Зовнішній
	password	varchar(max)	від 0 до 65535	
	registerCode	varchar(max)	від 0 до 65535	
dbo.diagnostic	id	int	від 0 до 4294967295	Первинний
	username	varchar(max)	від 0 до 65535	Зовнішній
	date	date	Формат: YYYY-MM-DD. Підтримуваний діапазон від '1000-01-01' до '9999-12-31'	
	test	varchar(max)	від 0 до 65535	
	left_eye	varchar(max)	від 0 до 65535	
	right_eye	varchar(max)	від 0 до 65535	
	total_result	float	Число з плаваючою комою. Від 0 до 24 знаків після коми	
	conclusion	varchar(max)	від 0 до 65535	

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
Зв'язки таблиць зображено на діаграмі бази даних (див. рис. 3.1).

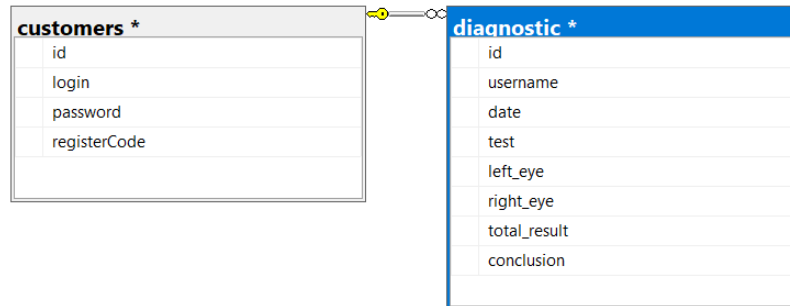


Рисунок 3.1 – Діаграма бази даних vision_diagnostics_db

3.2 Структура системи діагностики зору для профілактики захворювань очей

Авторизація, реєстрація, головне меню системи та особистий кабінет користувача

При запуску системи з'являється головне меню авторизації користувача. Якщо користувач вже має обліковий запис, то необхідно натиснути кнопку «Login», щоб перейти до вікна входу в обліковий запис. У випадку, коли обліковий запис відсутній необхідно натиснути на кнопку «Sign up», щоб перейти до вікна реєстрації.

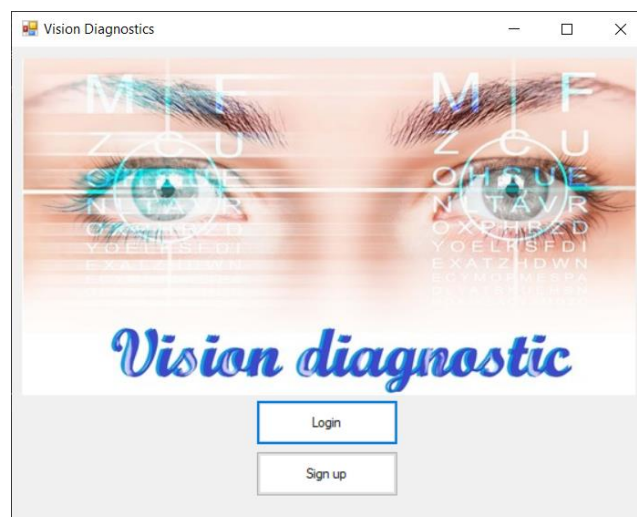


Рисунок 3.2 – Вікно авторизації

При переході до вікна входу в обліковий запис з'являється наступне вікно (рис. 3.3), у якому необхідно ввести логін та пароль користувача і натиснути на

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
кнопку «Login». Якщо введені дані не відповідають даним з БД з'явиться повідомлення про помилку (рис. 3.3).

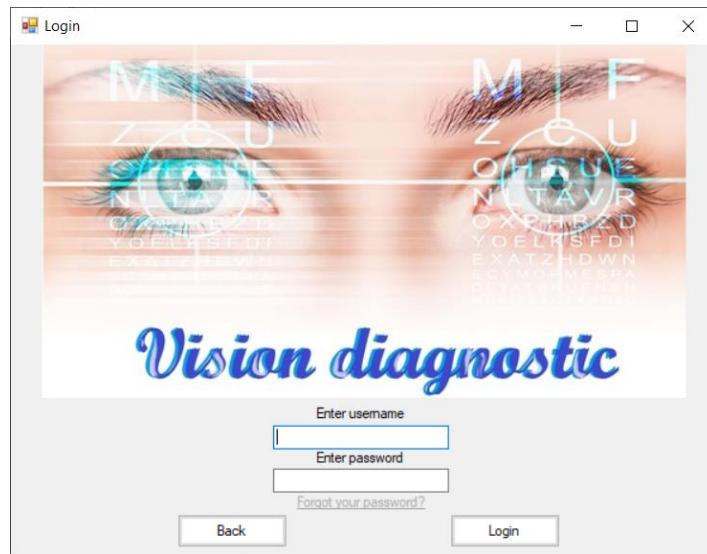


Рисунок 3.3 – Вікно авторизації

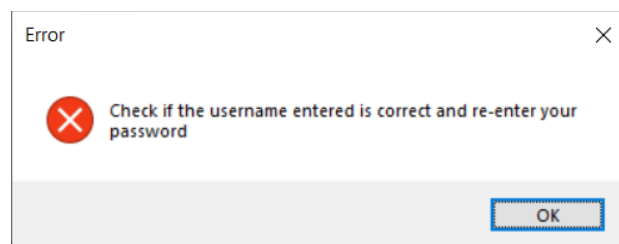


Рисунок 3.4 – Повідомлення при помилку при введенні логіну та паролю

Якщо користувач не пам'ятає пароль, то є можливість відновлення паролю. Для відновлення паролю необхідно натиснути на «Forgot your password?» після чого з'явиться вікно відновлення облікового запису (рис. 3.5).

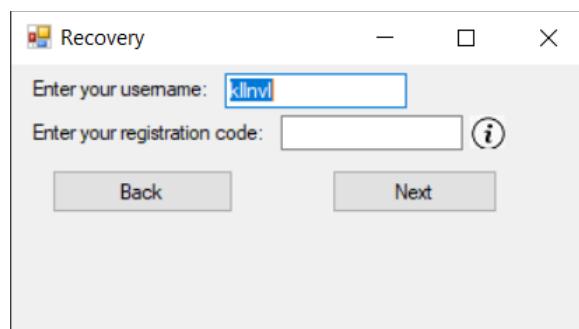


Рисунок 3.5 – Вікно відновлення паролю

Для того, щоб відновити пароль користувачеві необхідно ввести код, отриманий при реєстрації. При реєстрації додатково створюється файл RegisterCode.txt на ПК користувача, для того щоб користувач міг легко його

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
знайти. Ввівши свій код та натиснувши на кнопку «Next» користувач отримує можливість змінити свій пароль, ввівши новий та підтвердивши його (рис. 3.6). Якщо новий пароль не відповідає підтвердженому, то виникне відповідне повідомлення (рис. 3.7).

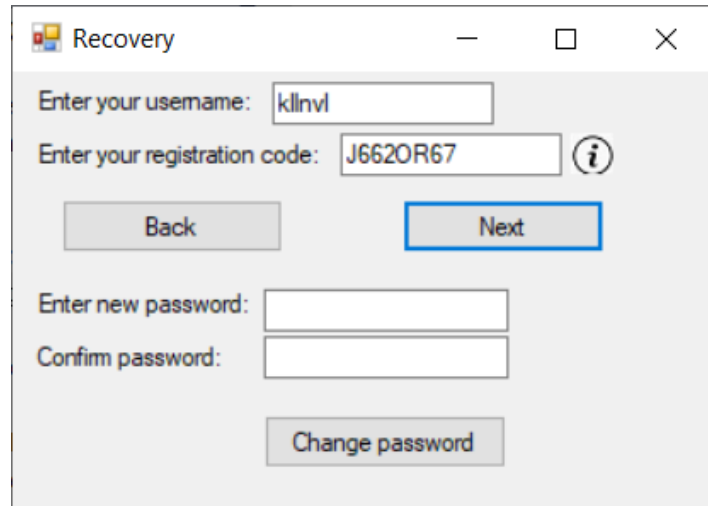


Рисунок 3.6 – Вікно відновлення паролю

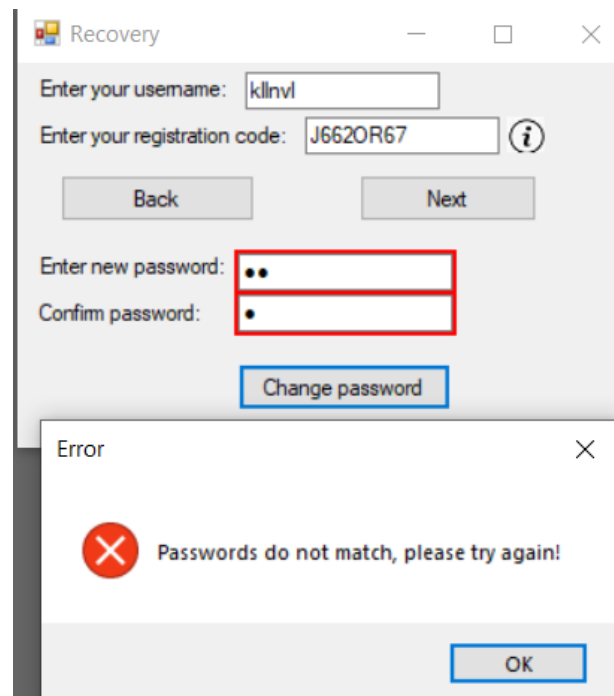


Рисунок 3.7 – Повідомлення про невідповідність введених паролів

Якщо новий пароль відповідає підтвердженому, то виникне відповідне повідомлення про успішну зміну паролю облікового запису (рис. 3.8).

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

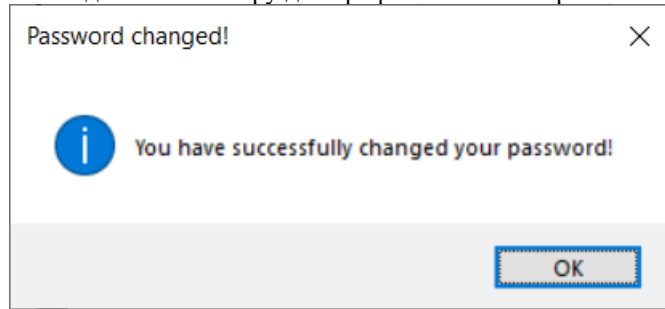


Рисунок 3.8 – Повідомлення про зміну паролю

У випадку, коли обліковий запис відсутній спочатку необхідно виконати реєстрацію у відповідному вікні (рис. 3.9). Для реєстрації необхідно ввести логін та пароль. Якщо у БД вже існують користувач з введеним логіном, то з'явиться повідомлення про помилку (рис. 3.10).

Для забезпечення безпеки даних, пароль зберігається у БД зашифрованим за допомогою 128-бітного алгоритму хешування MD5.

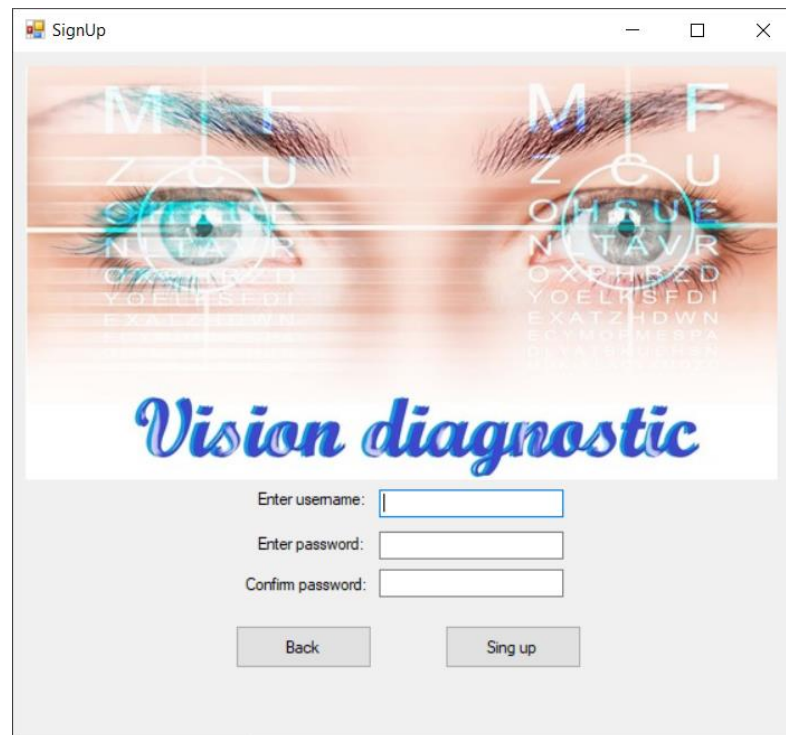


Рисунок 3.9 – Вікно реєстрації користувача

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

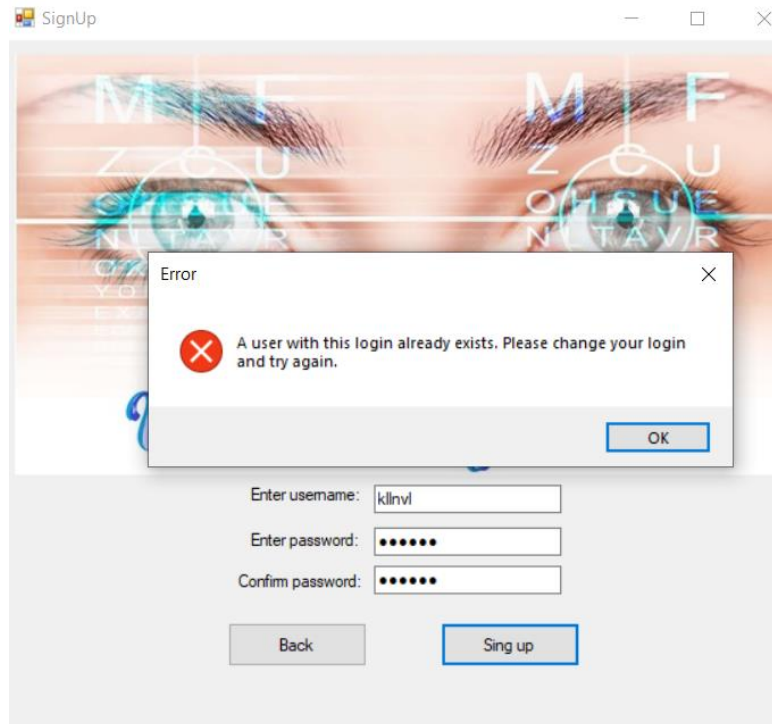


Рисунок 3.10 – Повідомлення про помилку при реєстрації (користувач з таким логіном вже існує у БД)

Якщо при авторизації усі дані вказано правильно, то по натисканню на кнопку «Login» з'явиться повідомлення про успішну авторизацію користувача та відбудеться перехід до головного меню системи (див. рис. 3.12).

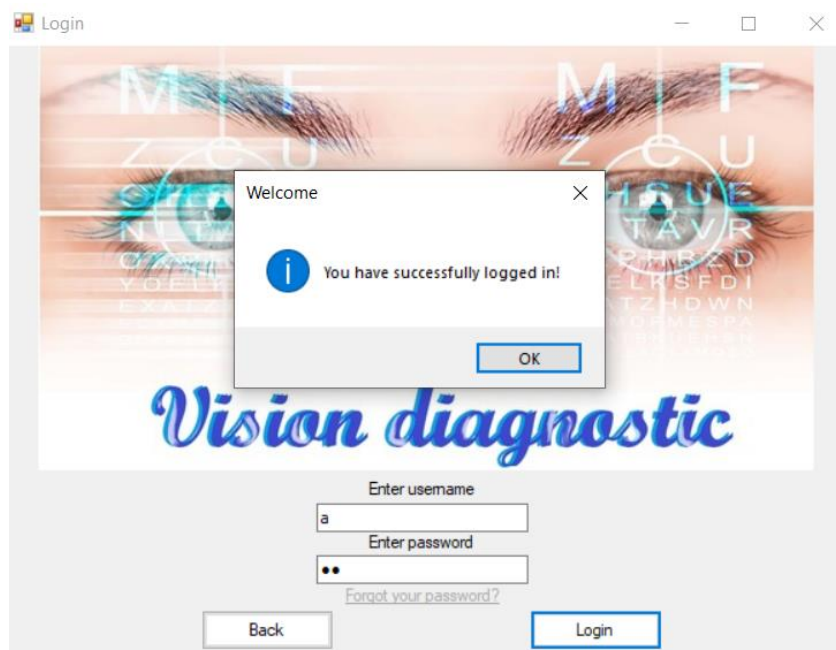


Рисунок 3.11 – Успішна авторизація користувача

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

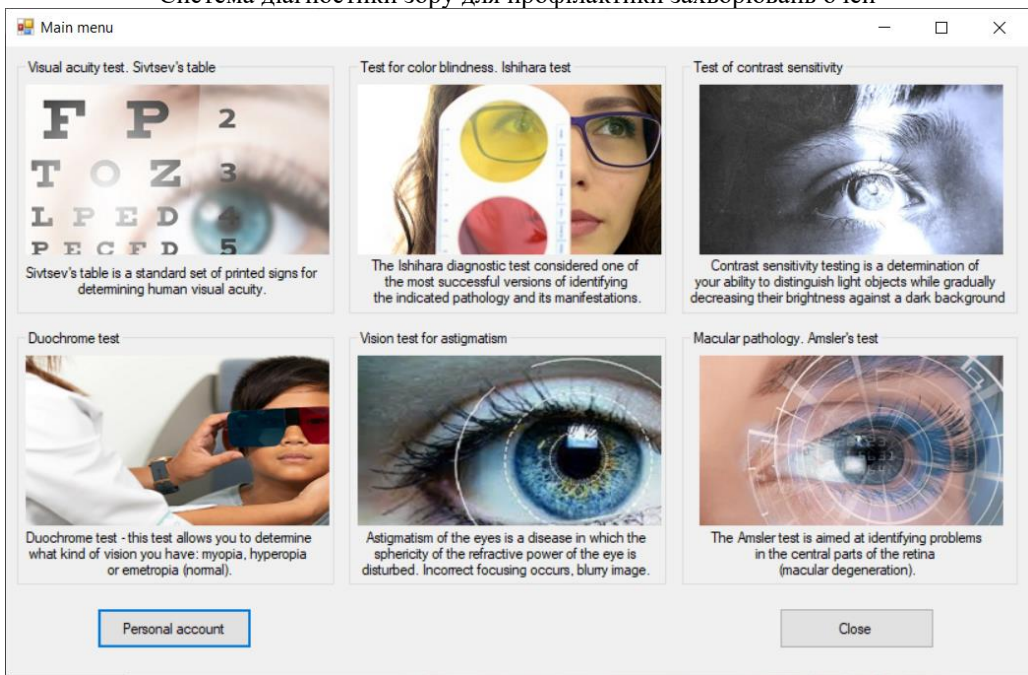


Рисунок 3.12 – Головне меню системи

На головному меню системи розташовано усі тести, які використовуються для діагностики зору з їх коротким описом. По натисканню на кнопку «Personal account» здійснюється перехід до особистого кабінету, у якому можна відстежити результати діагностування зору за необхідний період часу.

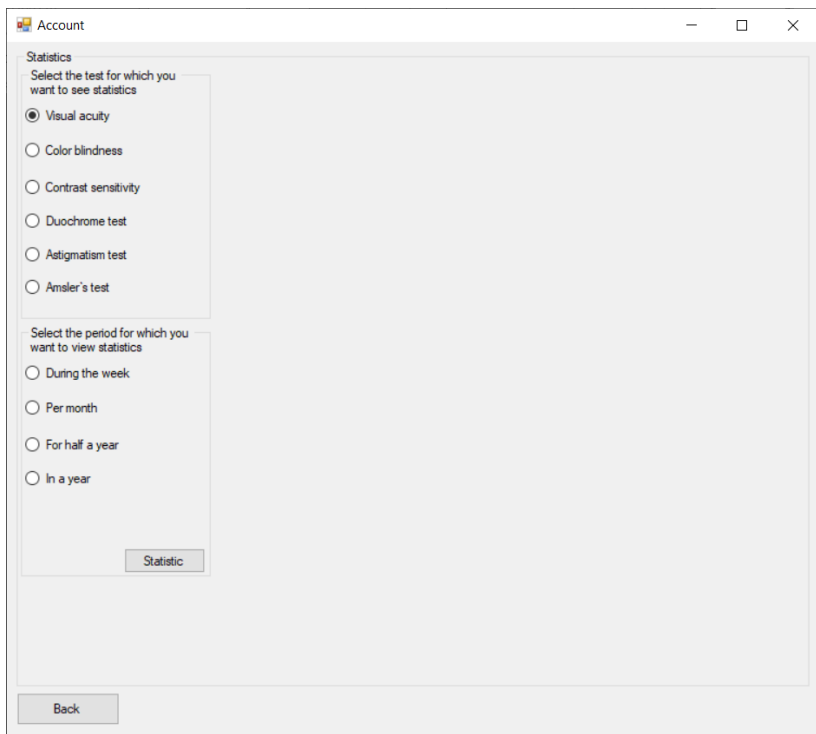


Рисунок 3.13 – Загальний вигляд особистого кабінету

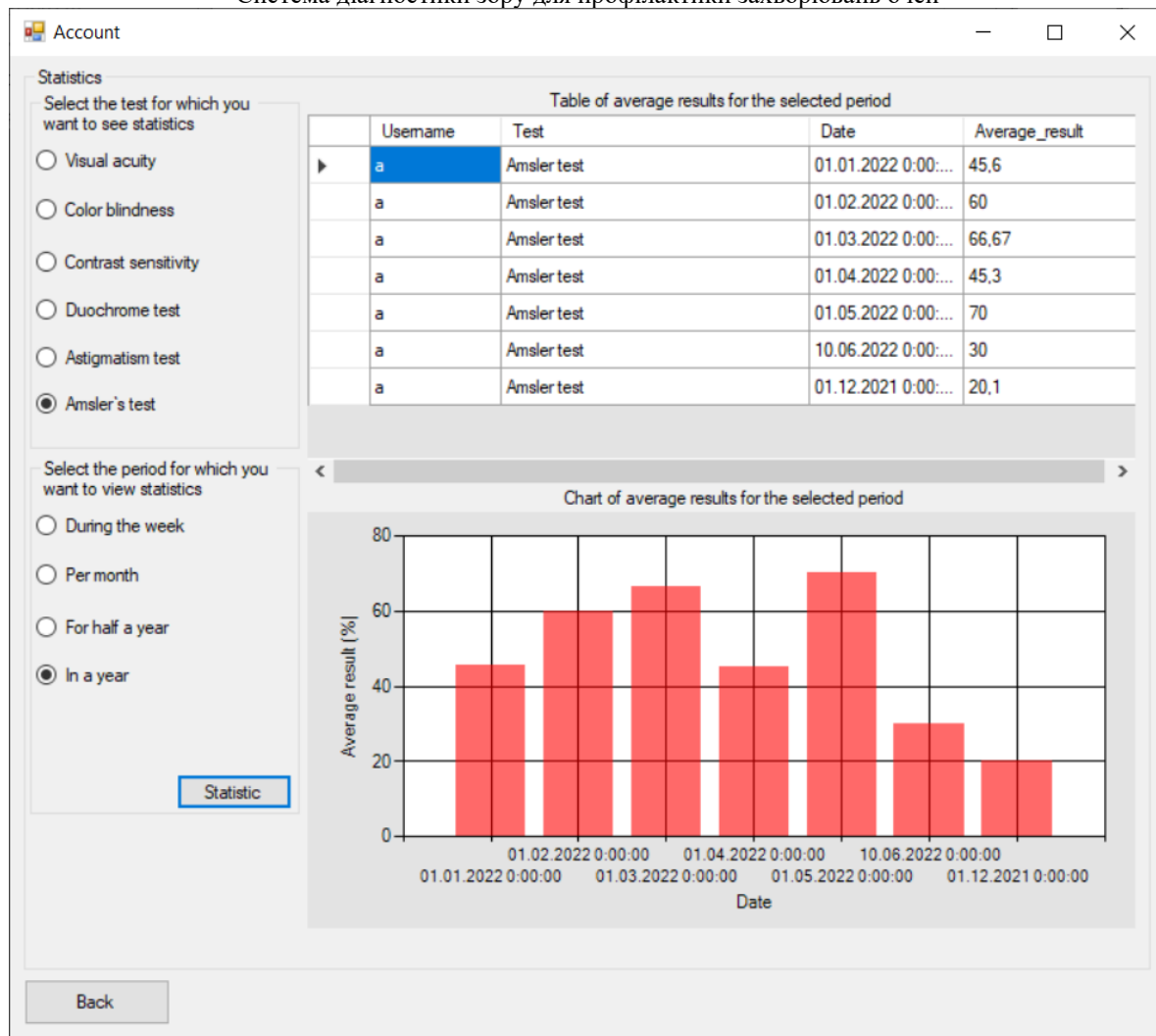


Рисунок 3.14 – Приклад відображення результатів проходження тесту Амслера за останній рік

Діагностика гостроти зору

Для того, щоб розпочати діагностику гостроти зору необхідно на головному меню систему натиснути на зображення у відповідній області. По натисканню відбудеться перехід до вікна діагностування (рис. 3.15).

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

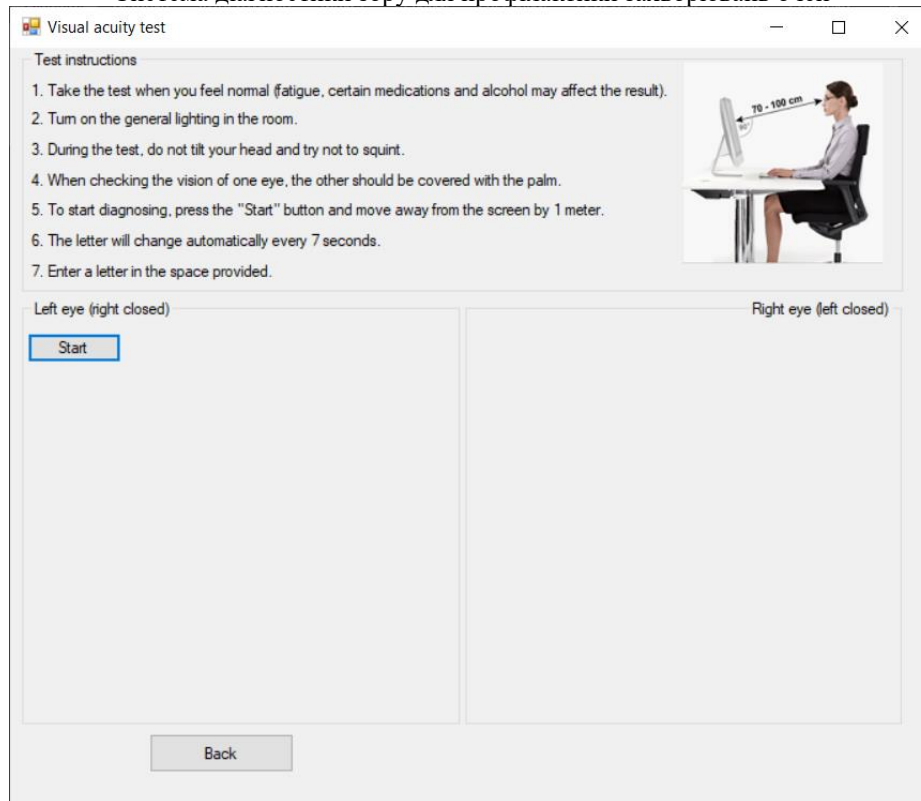


Рисунок 3.15 – Меню проведення діагностики гостроти зору

У верхньому вікні розміщено інструкція та правила проведення тестування. Для того, щоб розпочати діагностику необхідно відсісти від монітору на відстань 70-100 см та натиснути на кнопку «Start». Діагностування розпочнеться з дослідження гостроти зору лівого ока, тому праве має бути прикрите рукою. Після натиску на кнопку «Start» у області «Left eye» з'явиться перше поле для вводу та перший оптотип для діагностування. Кожні 7 секунд змінюється оптотип та додається нове поле для вводу.

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

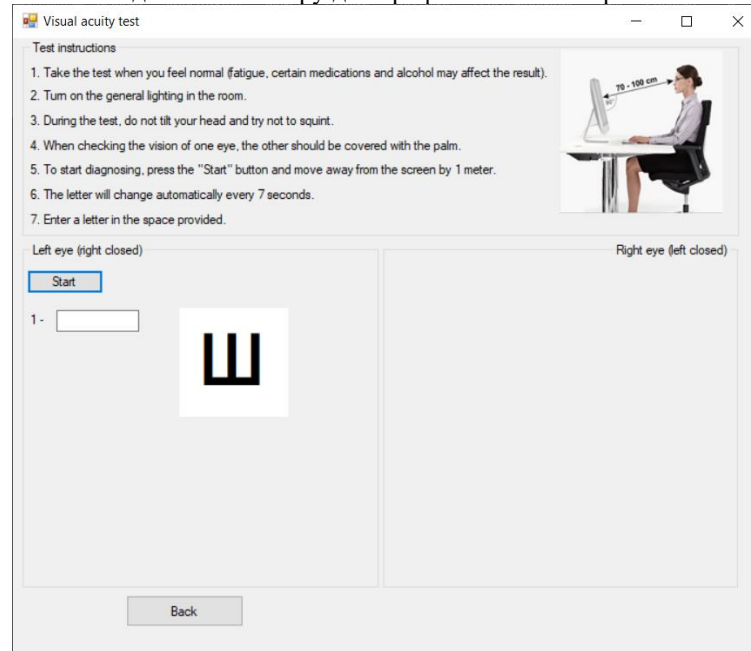


Рисунок 3.16 – Початок діагностики гостроти зору

Після того, як з'являться останній (9-ий) оптотип діагностику зору лівого ока буде закінчено. У області «Right eye» автоматично з'явиться кнопка «Start».

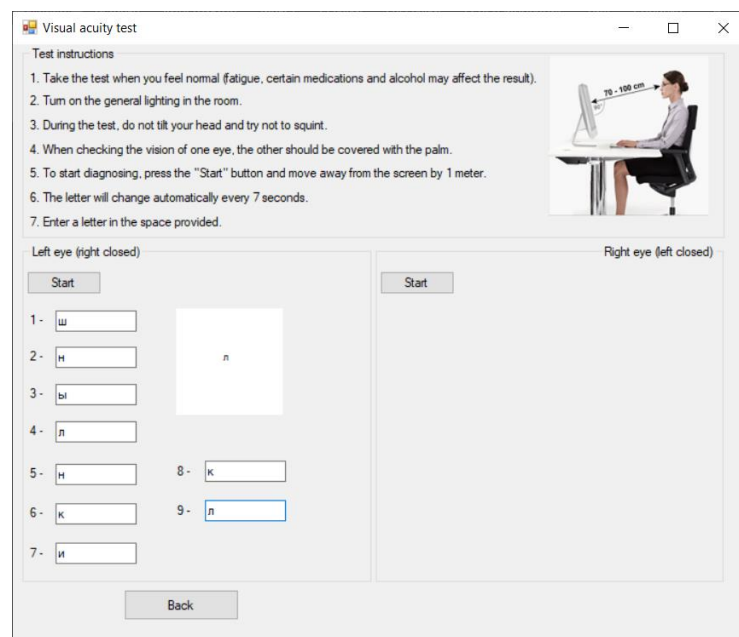


Рисунок 3.17 – Закінчення діагностики лівого ока

По натиску на кнопку розпочнеться процес діагностування правого ока за аналогічним принципом.

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

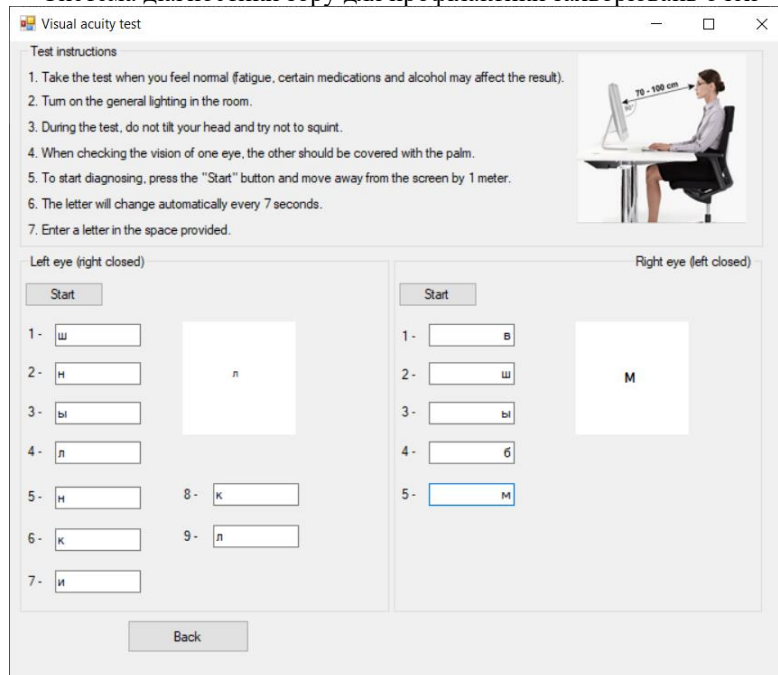


Рисунок 3.18 – Процес діагностики правого ока

По завершенню діагностики правого ока з'являється кнопка «Get result», яка підрахує гостроту зору кожного ока та, використовуючи Байєсівський класифікатор, розрахує та поверне загальний результат стану зору користувача.

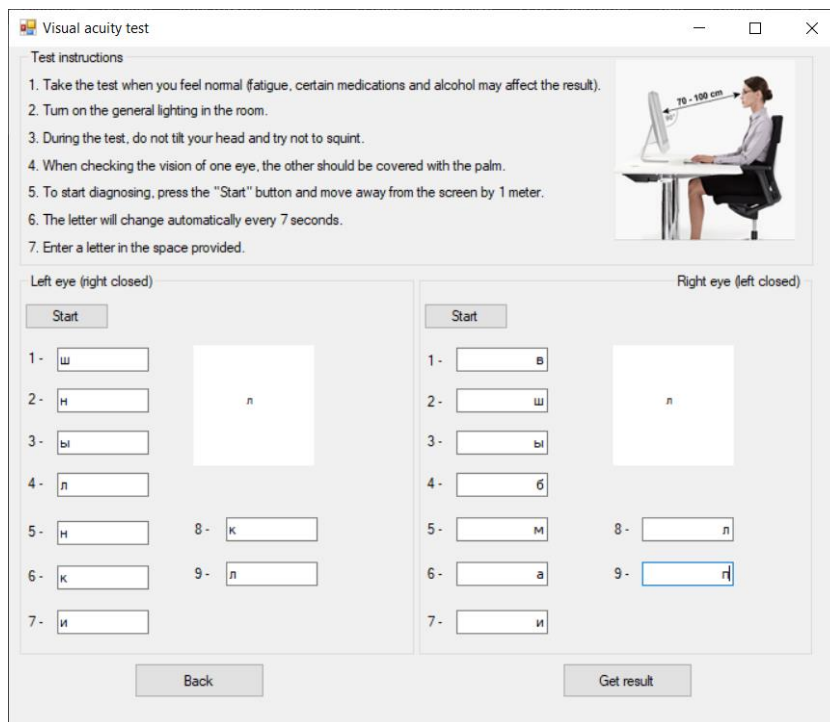


Рисунок 3.19 – Закінчення діагностики правого ока

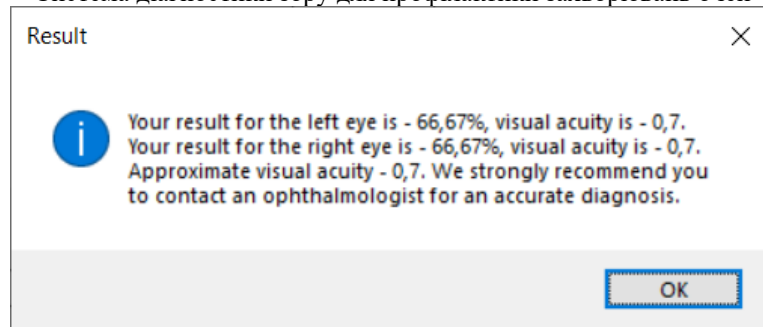


Рисунок 3.20 – Отриманий результат діагностики гостроти зору

Діагностика кольоровідчуття

Для того, щоб розпочати діагностику кольоровідчуття необхідно на головному меню систему натиснути на зображення у відповідній області. По натисканню відбудеться перехід до вікна діагностування (рис. 3.21).

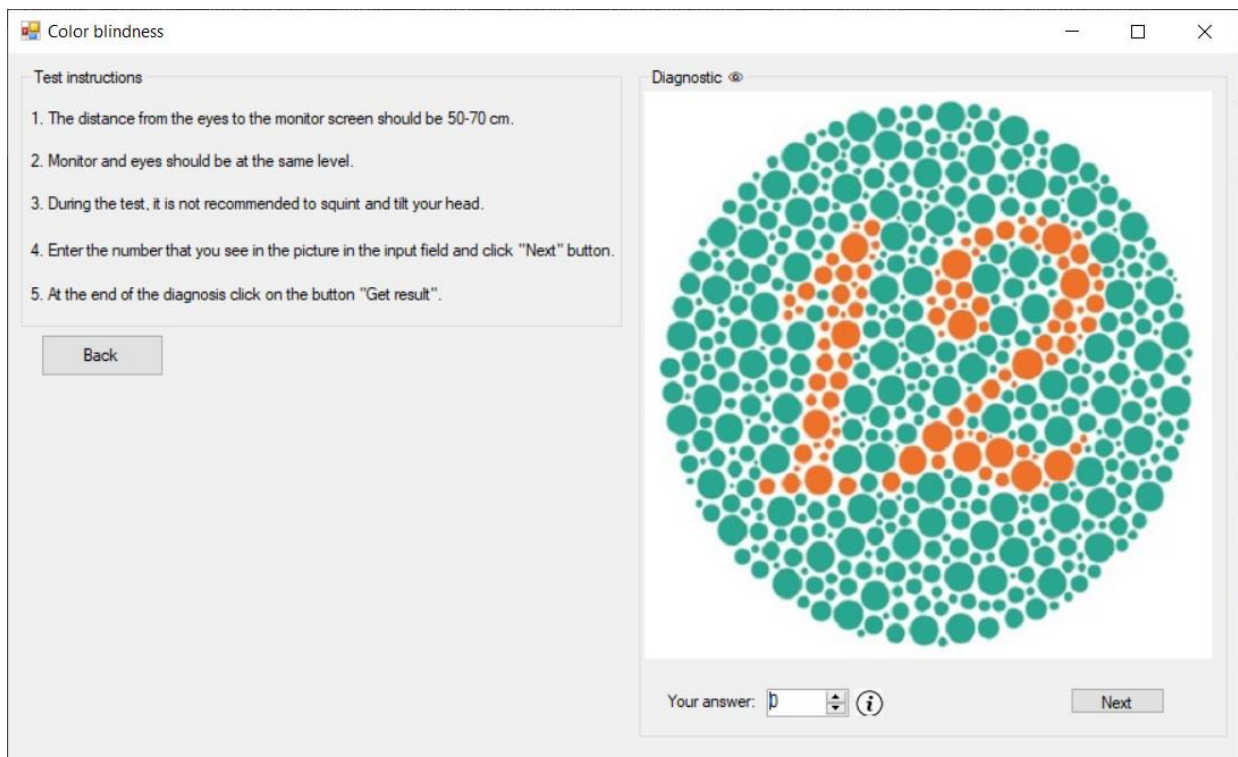


Рисунок 3.21 – Вікно діагностування кольоровідчуття

У верхній області вікна розміщено правила та інструкцію проведення діагностики. Для того, щоб розпочати діагностику потрібно ввести значення з першої таблиці Ішихари. Перша таблиця є перевіркою – оптотип з цієї таблиці можуть розрізняти як здорові люди, так і люди з патологією. Цю таблицю зазвичай використовують для того, щоб перевірити чи не симулює пацієнт

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
наявність дальтонізму. Вказавши відповідь необхідно натиснути на кнопку «Next», щоб перейти до наступної таблиці.

У інформаційній вказівці, яка знаходиться біля поля вводу відповіді, повідомляється про те, що якщо користувач не може розрізнити оптотип на зображенні необхідно у полі відповіді вказати 0.

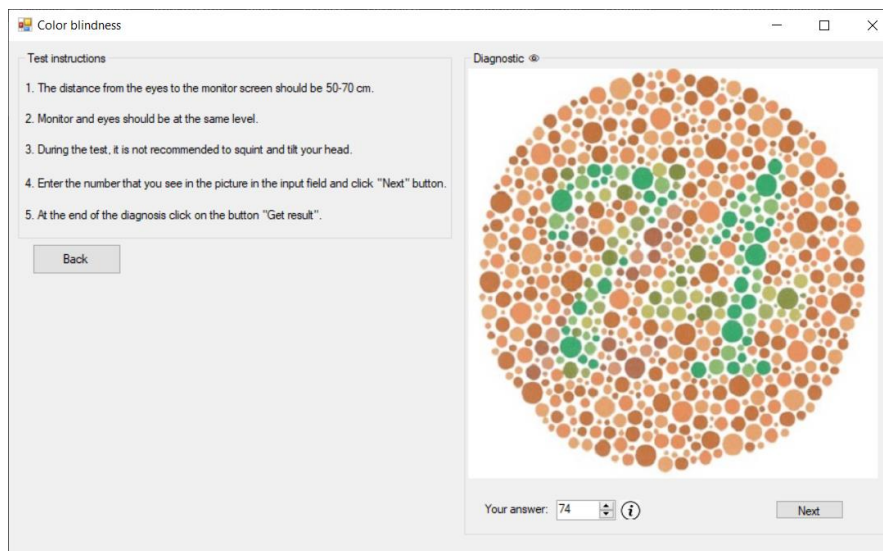


Рисунок 3.22 – Приклад другої таблиці Ішихари

Таблиці, у яких оптотипи зображено зеленим кольором на червоному тлі використовуються для діагностування протанопії.

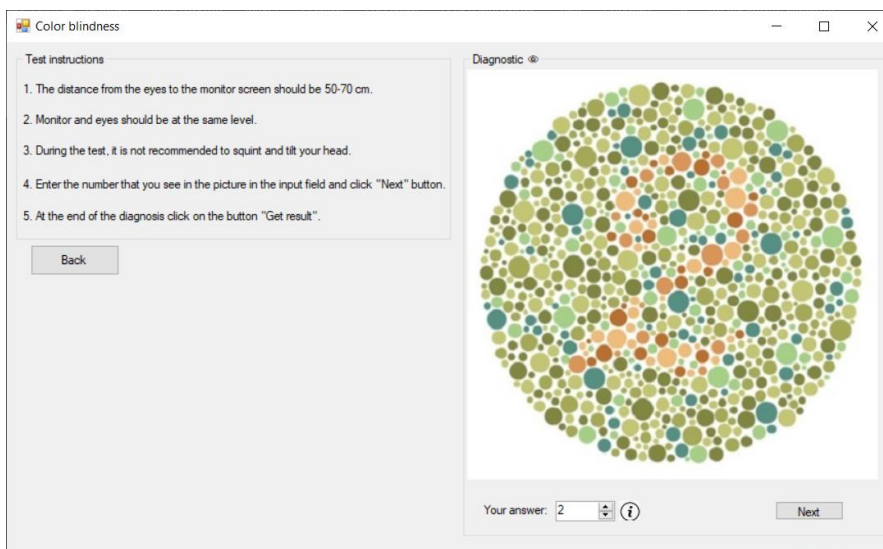


Рисунок 3.23 – Приклад третьої таблиці Ішихари

Таблиці, у яких оптоптики зображено червоним кольором на зеленому тлі використовуються для діагностування дейтеранопії.

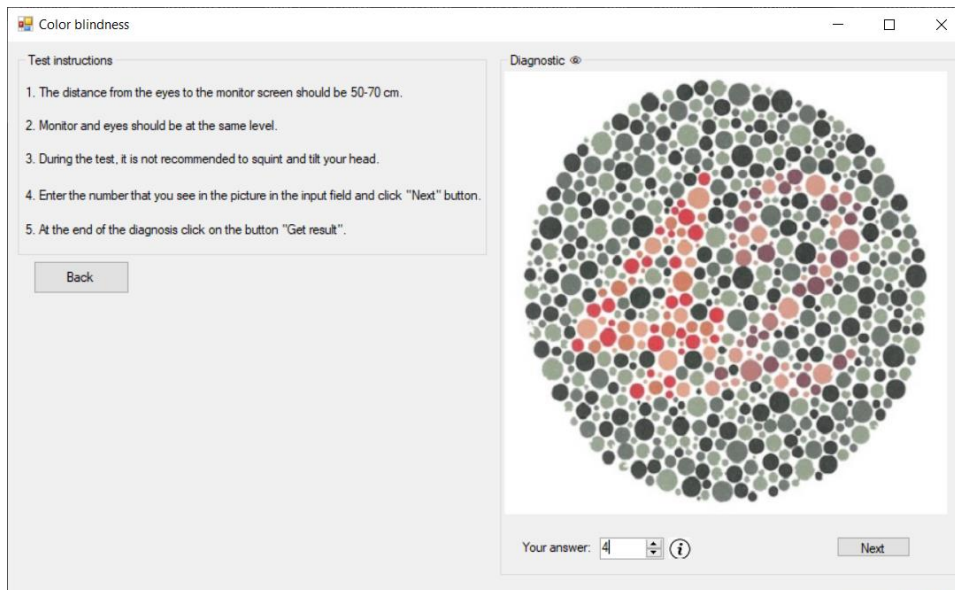


Рисунок 3.24 – Приклад шостої таблиці Ішихари

Після введення останнього оптоптику та натиску на кнопку «Next» з'являється кнопка «Get result», яка використовуючи Байєсівський класифікатор, розрахує та поверне загальний результат стану зору користувача.

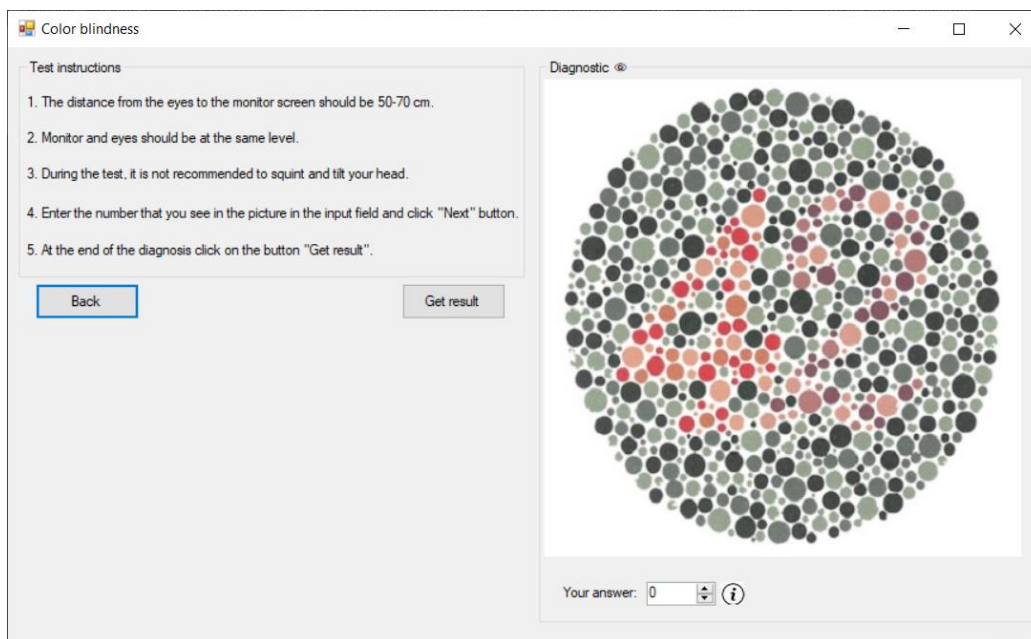


Рисунок 3.25 – Завершення діагностування кольоровідчуття

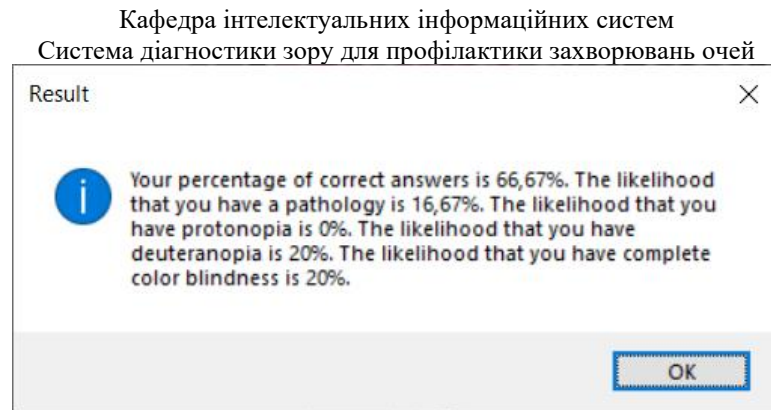


Рисунок 3.26 – Отриманий результат діагностики кольоровідчуття

Діагностика контрастної чутливості

Для того, щоб розпочати діагностику контрастної чутливості необхідно на головному меню систему натиснути на зображення у відповідній області. По натисканню відбудеться перехід до вікна діагностування (рис. 3.27).

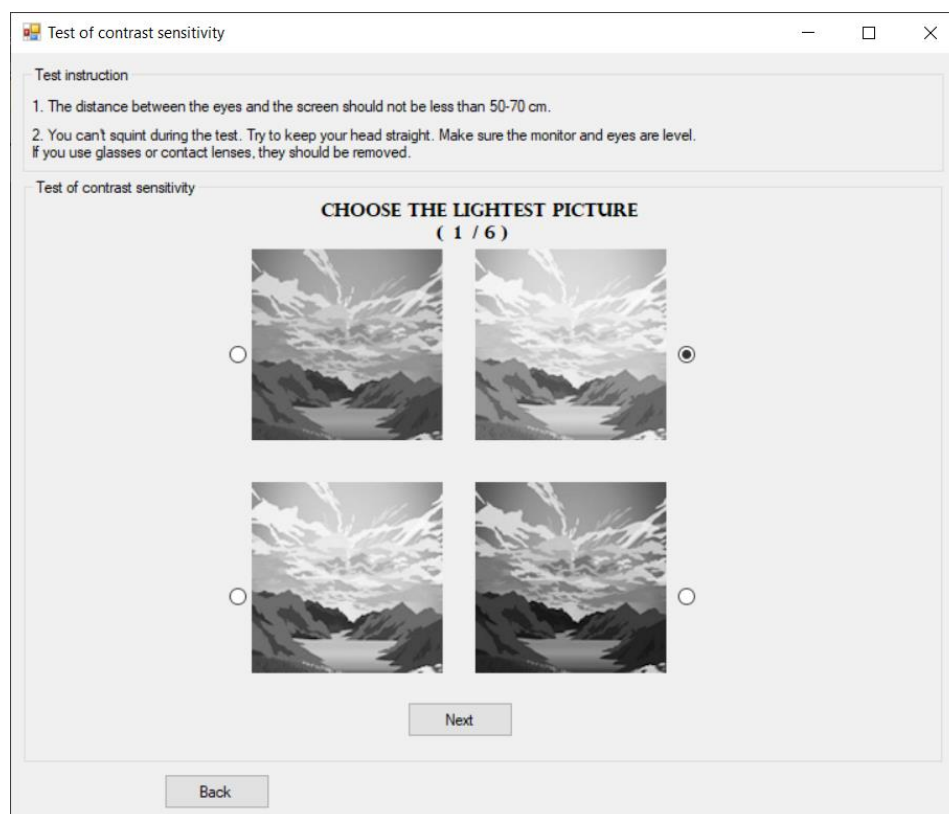


Рисунок 3.27– Вікно діагностики контрастної чутливості

У верхній частині вікна розміщено вказівки для проходження діагностики. Для виконання діагностики необхідно слідуючи вказівкам, які виділені жирним шрифтом, обирати підходящі під опис зображення. Спершу необхідно обрати

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
найсвітліше зображення з представлених. Далі необхідно обрати найяскравіше
зображення (рис. 3.28).

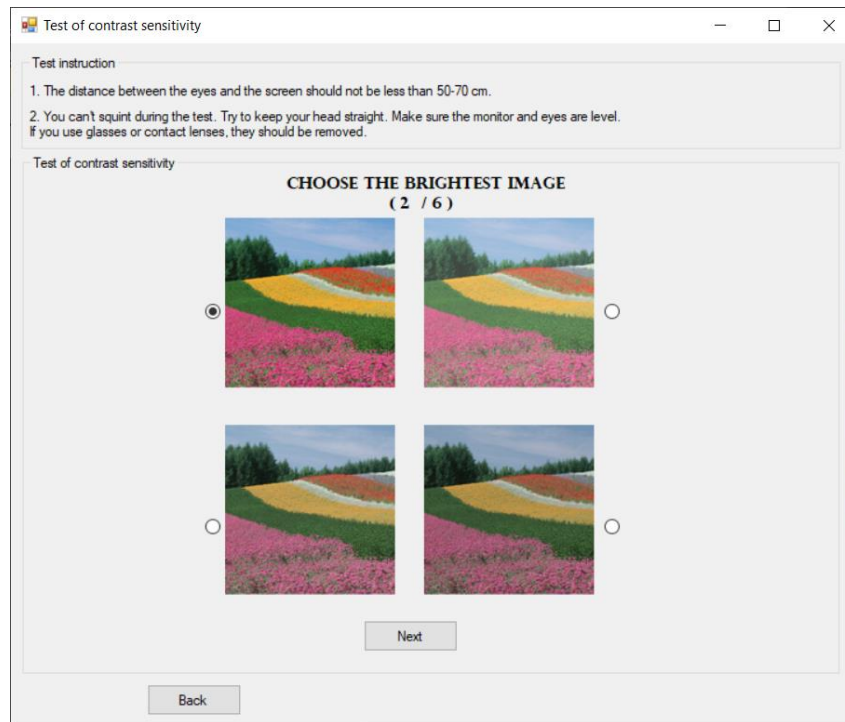


Рисунок 3.28 – Діагностика контрастної чутливості

На наступному кроці необхідно обрати зображення, яке є найменш
яскравим з представлених варіантів.

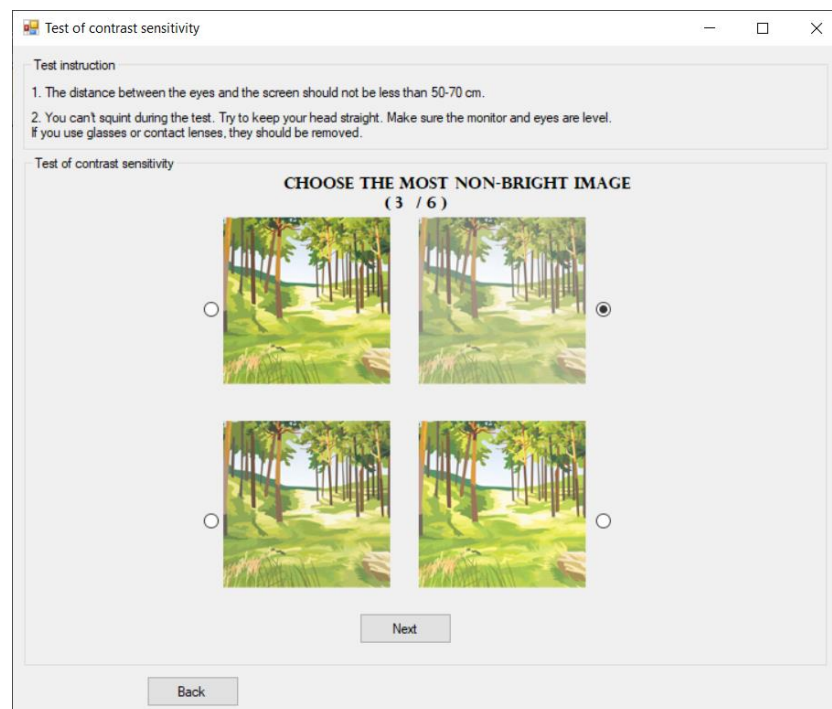


Рисунок 3.29 – Діагностика контрастної чутливості

Далі необхідно обрати два зображення, які є однаковими за рівнем контрасту.

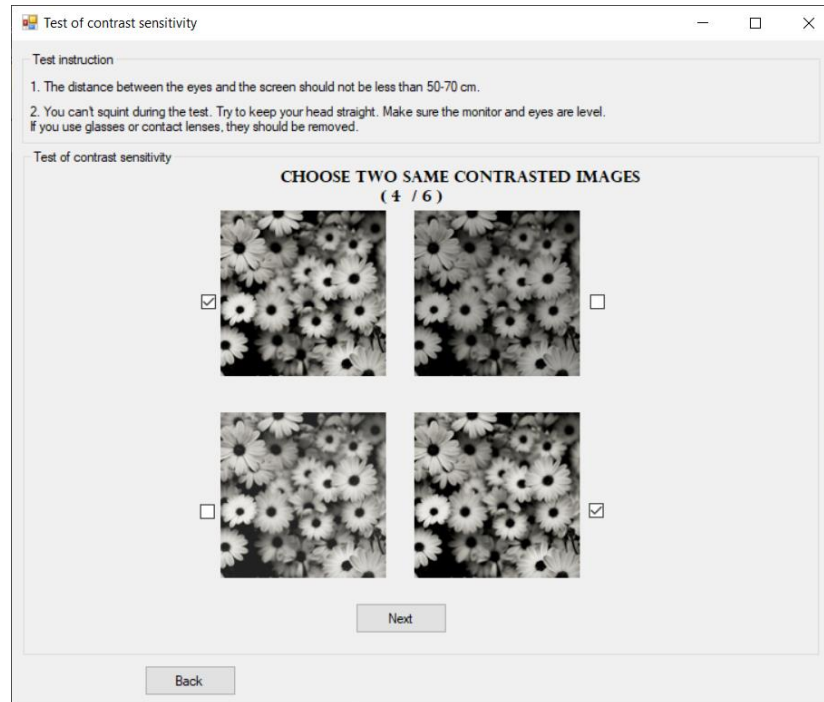


Рисунок 3.30 – Діагностика контрастної чутливості

Після цього обраємо найтемніше зображення.

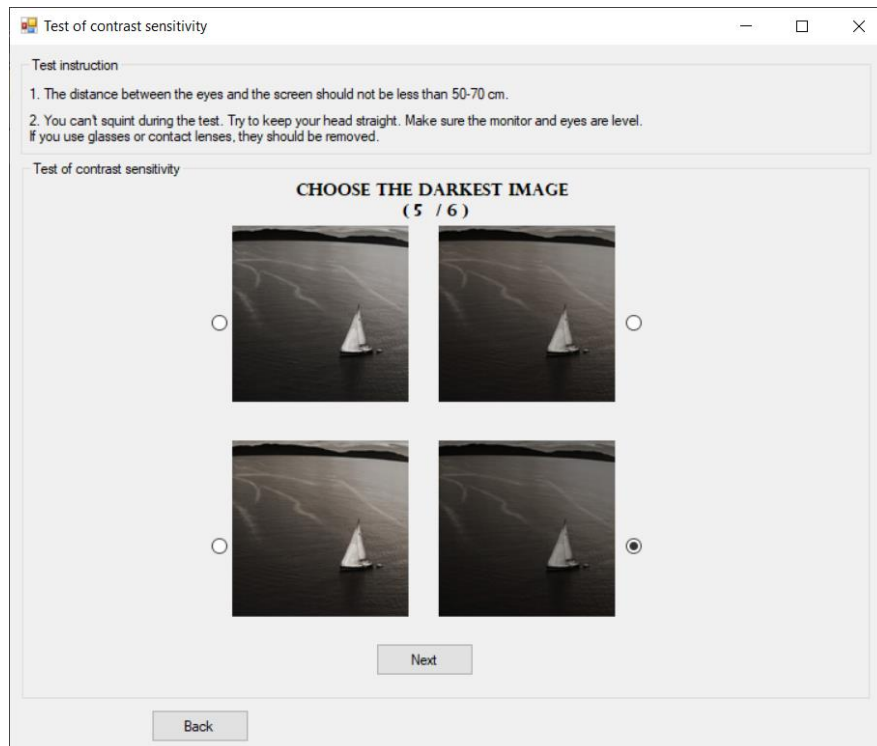


Рисунок 3.31 – Діагностика контрастної чутливості

На останньому кроці необхідно обрати найяскравіше зображення.

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

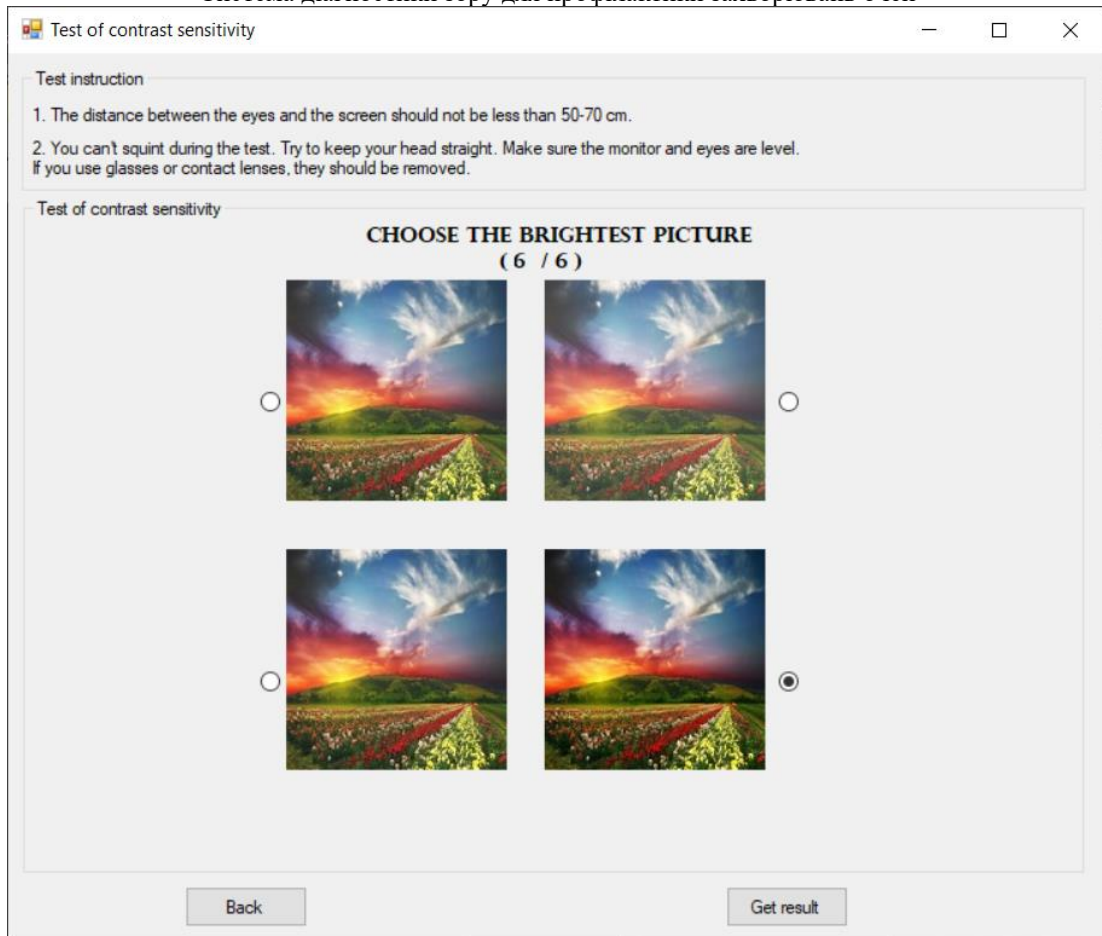


Рисунок 3.32 – Діагностика контрастної чутливості

Після чого з'являється кнопка «Get result», яка використовуючи Байєсівський класифікатор, розрахує та поверне висновок про ступінь контрастної чутливості очей користувача.



Рисунок 3.33 – Результат діагностики контрастної чутливості

Дуохромний тест

Для того, щоб розпочати перевірку зору на наявність далекозорості або короткозорості необхідно на головному меню натиснути на зображення у області

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
«Duochrome test». По натисканню відбудеться перехід до вікна діагностування
(рис. 3.34).

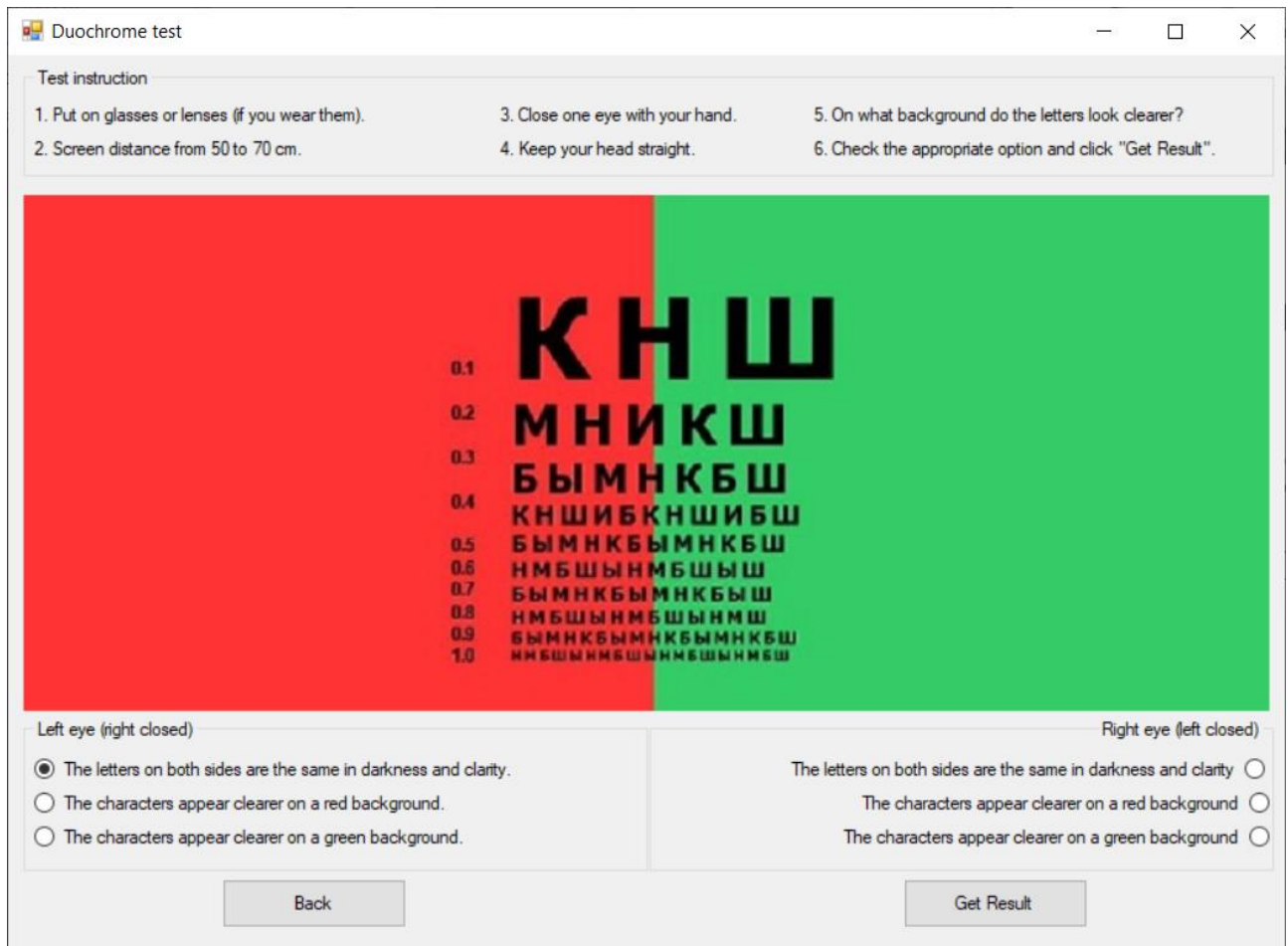


Рисунок 3.34 – Вікно перевірки зору на наявність далекозорості або короткозорості

У верхній області вікна розміщено правила та інструкцію проведення діагностики. У середній області вікна розміщено тест-об'єкт, який являє собою червоно-зелене поле з нанесеними на ньому чорними опто типами. Висновок робиться на підставі того, на полі якого кольору пацієнт краще розрізняє опто типи. У нижній області вікна діагностики розміщено опції для кожного ока.

Для виконання діагностики необхідно закрити праве око та обрати найбільш підходящий варіант, з перелічених, для лівого ока у відведеній області. Потом повторити процедуру для другого ока і натиснути на кнопку « Get result», для того щоб отримати висновок проведеного діагностування.

Для людей, у яких відсутня патологія (міопія або гіперметропія) оптопти буде видно однаково чітко з обох сторін. Для людей з далекозорістю (гіперметропією) оптопти будуть чіткішими на зеленому фоні. Для людей з короткозорістю (міопією) оптопти будуть чіткішими на червоному фоні. Одним з рідкісних, але можливих випадків є анізотропія (антисметропія). У цьому випадку людина може бути короткозорою на одне око і далекозорою на інше. Анізотропія є станом, при якому заломлююча здатність (згинання світла) в обох очей суттєво відрізняються.

Після натиску на кнопку «Get result» з'явиться повідомлення з результатом діагностування.

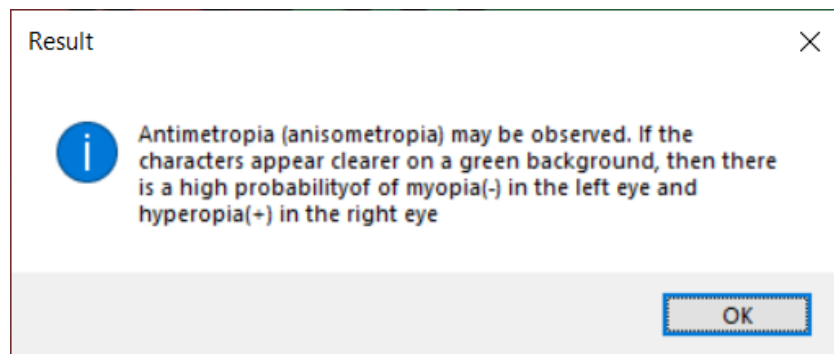


Рисунок 3.35 – Результат перевірки зору на наявність далекозорості або короткозорості

Діагностування наявності астигматизму

Для того, щоб розпочати перевірку зору на наявність астигматизму необхідно на головному меню натиснути на зображення у області «Vision test for astigmatism». По натисканню відбудеться перехід до вікна діагностування (рис. 3.36).

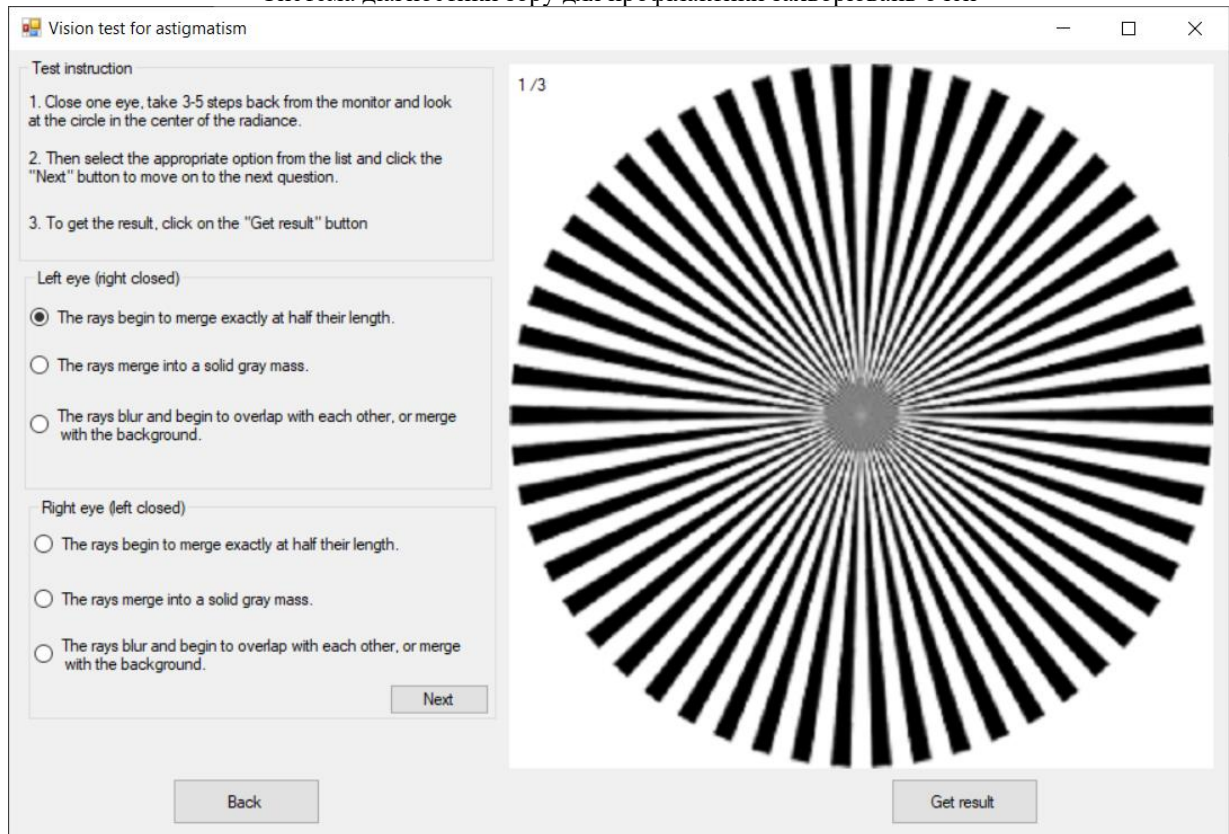


Рисунок 3.36 – Вікно перевірки зору на наявність астигматизму

У лівій верхній області вікна розміщено правила та інструкцію проведення діагностики. Усього для проведення діагностики використовується три варіанти тестувань, один з яких – зірка Сіменса. Для кожного ока необхідно обрати найбільш підходящу опцію та натиснути на кнопку «Next», щоб перейти до наступного тесту.

Діагностування наявності астигматизму у користувача починається з зірки Сіменса (рис. 3.36). Якщо стан зору є неідеальним, то, не доходячи до центру, промені розпливаються і починають перекриватися між собою.

На наступному тест-об'єкті (рис. 3.37) необхідно звернути увагу, чи не стають деякі з ліній темнішими, ніж інші.

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

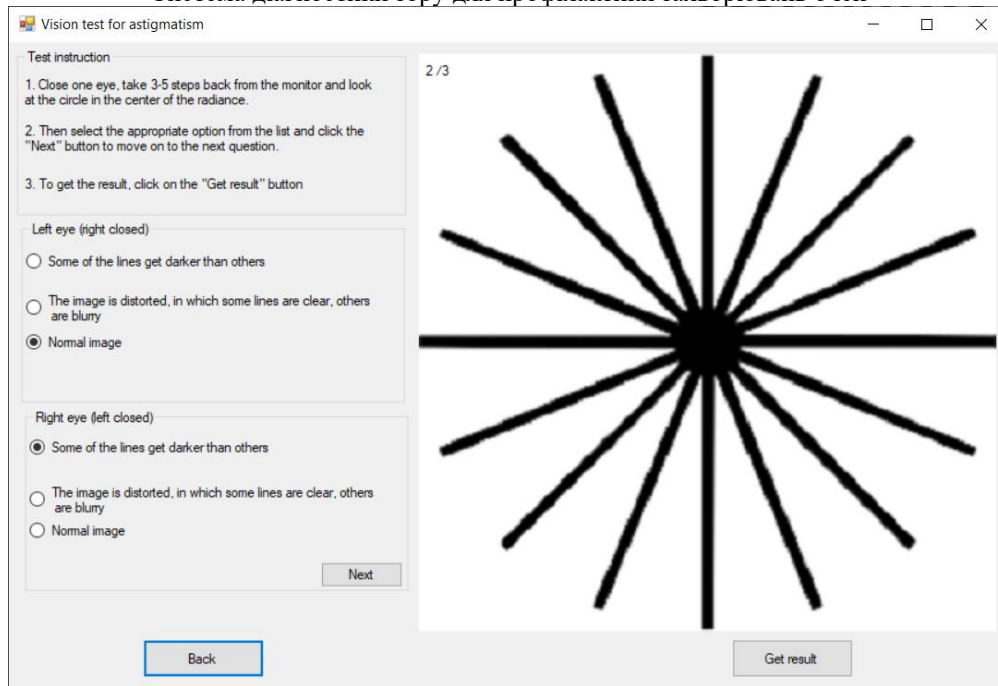


Рисунок 3.37 – Перевірка зору на наявність астигматизму

На останньому тест-об'єкті (рис. 3.38) необхідно звернути увагу, чи є ліновані квадрати однаково чорними. Якщо один і більше квадратів виглядають сірими, то є ймовірність наявності астигматизму. У випадку, коли усі ліновані квадрати є однаково чорними – астигматизм, швидше за все, відсутній.

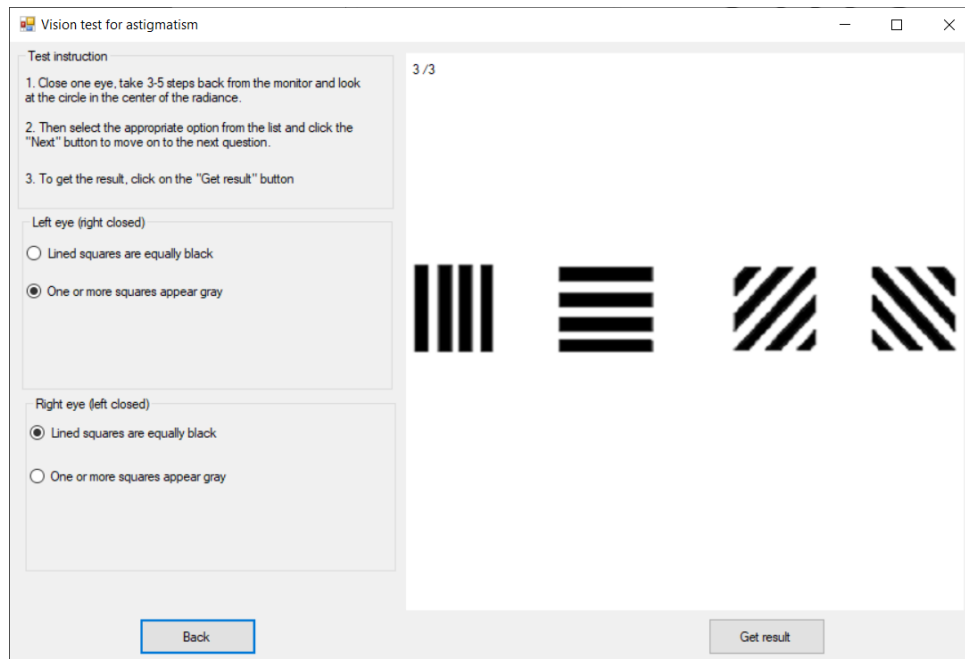


Рисунок 3.38 – Перевірка зору на наявність астигматизму

Після натиску на кнопку «Get result» з'явиться повідомлення з результатом діагностування.

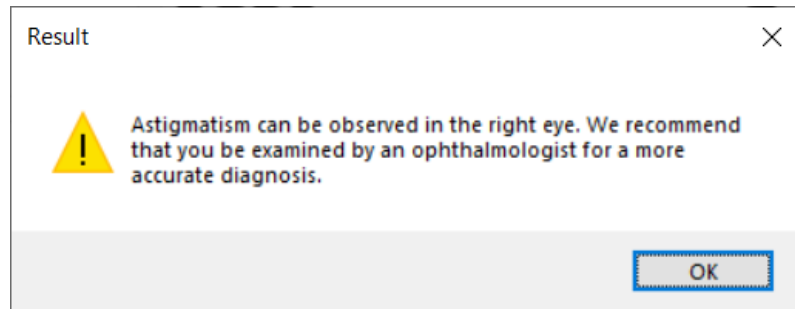


Рисунок 3.39 – Результат перевірки зору на наявність астигматизму

Діагностування патології макули. Тест Амслера

Для того, щоб розпочати перевірку зору на наявність патології макули необхідно на головному меню натиснути на зображення у області «Macular pathology. Amuser's test». По натисканню відбудеться перехід до вікна діагностування (рис. 3.40).

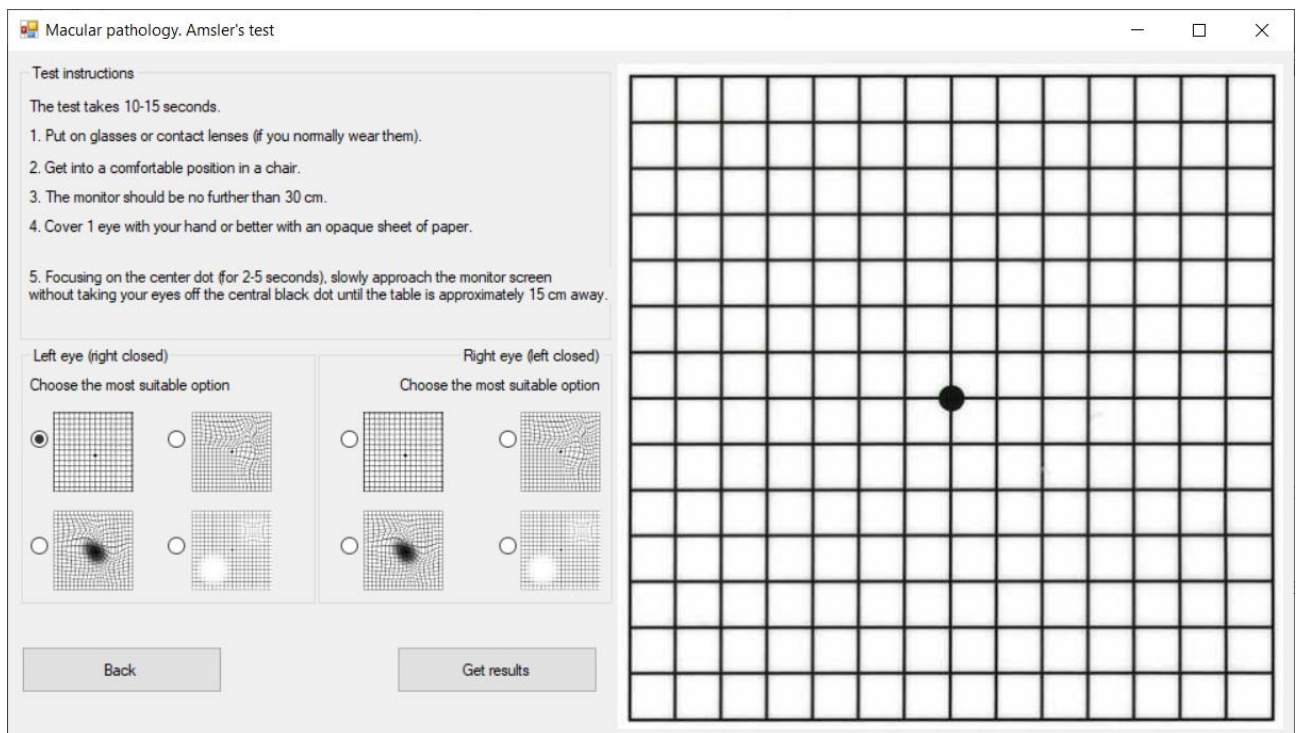


Рисунок 3.40 – Вікно перевірки зору на наявність патології макули

У лівій верхній області вікна розміщено правила та інструкцію проведення діагностики. У правій області вікна розміщено тест-об'єкт, який

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
використовується для діагностування патології макули. Необхідно, слідуючи вказівкам, перевірити кожне око та обрати найбільш схоже до побаченого зображення. При перевірці необхідно звернути увагу, чи немає зон, де малюнок спотворюється, затуманюється, знебарвлюється та обрати найбільш підходяще під побачене зображення. При наявній патології макули користувач побаче зображення спотвореним.

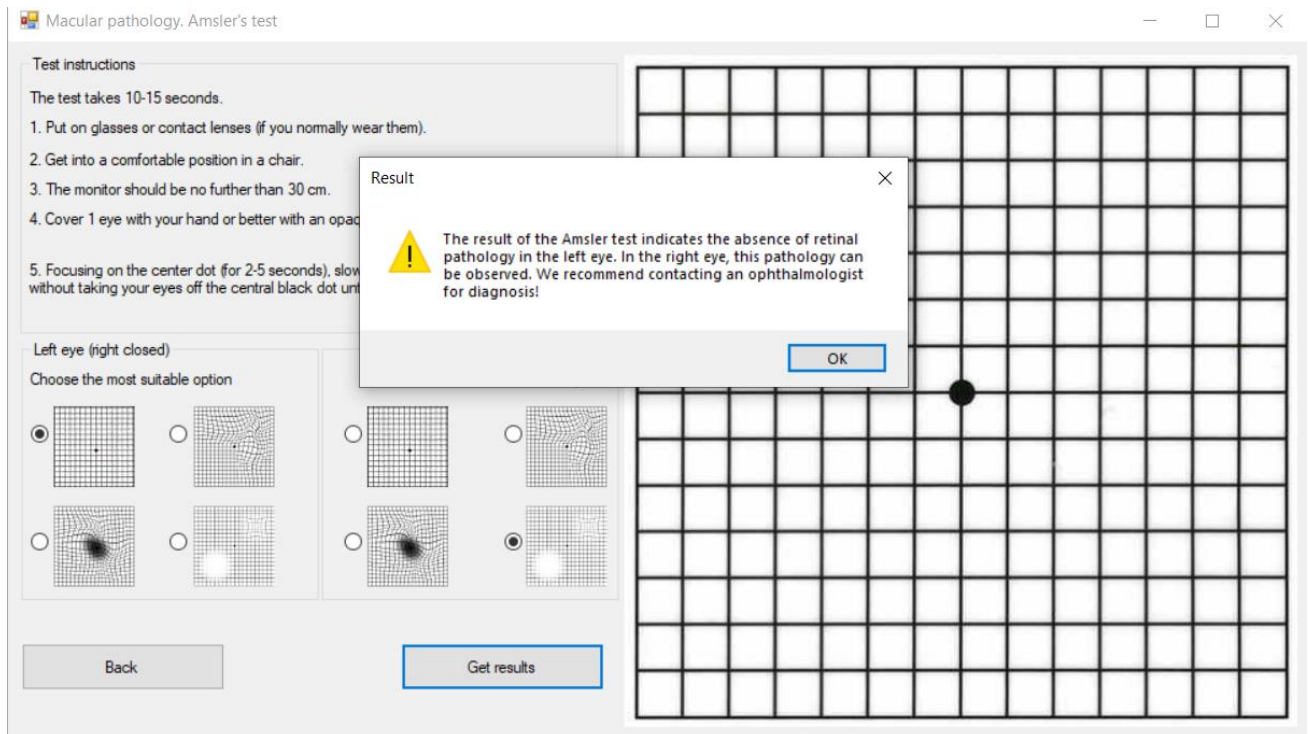


Рисунок 3.41 – Результат перевірки зору на наявність патології макули

Висновки до розділу 3

У результаті роботи було створено систему, яка надає можливість користувачеві виконувати діагностування зору у таких напрямках, як: гострота зору, кольоровідчуття, перевірка контрастної чутливості, перевірка на наявність далекозорості або короткозорості, перевірка на наявність астигматизму та перевірка на наявність патології макули.

При створенні системи діагностики зору для профілактики захворювань очей для класифікації результатів було використано наївний класифікатор Байеса. Наївний класифікатор широко використовується у медичній діагностиці, адже його легко здійснити. Завдяки наївному класифікатору розроблена система

здатна достатньо точно виконувати діагностування патологій та порушень стану зору користувача.

При реалізації системи для діагностики зору для профілактики захворювань зору було вирішено створити БД `vision_diagnostics_db` для зберігання результатів тестувань користувача, які будуть використовуватися для подальшого аналізування та відображення статистики.

Завдяки використанню БД при створенні системи отримано набір даних, який складається з результатів та висновків діагностування різних користувачів, який у майбутньому можна використовувати для аналізу стану зору користувачів різної вікової категорії.

Спеціальний розділ

ОХОРОНА ПРАЦІ

до кваліфікаційної роботи

на тему:

«СИСТЕМА ДІАГНОСТИКИ ЗОРУ ДЛЯ ПРОФІЛАКТИКИ ЗАХВОРЮВАНЬ ОЧЕЙ»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810114

Виконав студент 4-го курсу, групи 401

Клішевська А.В.

(підпис, ініціали та прізвище)

«__» _____ 2022 р.

Консультант ст. викладач

(наук. ступінь, вчене звання)

Макарова О.В.

(підпис, ініціали та прізвище)

«__» _____ 202_ р.

Миколаїв – 2022

4 ОХОРОНА ПРАЦІ

Охорона праці – це система забезпечення безпеки роботи, збереження життя та здоров'я працюючих у процесі всієї їх трудової діяльності.

Охорона праці сьогодні є однією з головних систем на підприємствах різного спрямування будь-яких форм власності. Дотримуючись елементарних правил безпечного виконання робіт та вимог охорони праці, роботодавець може досягти істотного зниження виробничого травматизму, практично до 95%. Відсоток ймовірності, що залишився, складе людський фактор, від якого складно піти. Саме таке відсоткове співвідношення випадків травматизму виникають через недотримання вимог охорони праці як з боку роботодавця, так і з боку працівника.

В даній кваліфікаційній роботі створено систему діагностики зору для профілактики захворювань очей. З метою створення безпечних і здорових умов праці при створенні цієї програми виконується аналіз умов на робочому місці.

Метою роботи є створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях та забезпечення безпеки людини у надзвичайних ситуаціях.

Відповідно до мети виділені наступні **завдання**:

1. Виконати опис серверного приміщення, робочого місця адміністратора та виробничого обладнання.
2. Оцінити умови праці у серверному приміщенні.
3. Сформулювати рекомендації щодо поліпшення умов праці на робочому місці адміністратора.

Приміщення серверної кімнати закладу ресторанного типу розташовано на другому поверсі п'ятиповерхової будівлі. Розміри приміщення складають $a \times b \times H = 6,5 \times 6,0 \times 3,5$ м. У приміщенні влаштовано два металопластикових вікна (з потрійними склопакетами) розмірами $c \times d = 2,0 \times 2,4$ м.

Приміщення має сучасний офісний інтер'єр. Стеля виконана у вигляді підвісної конструкції із синтетичного матеріалу світло-сірого кольору. Стіни

мають гладку поверхню кольору біла слонова кістка. Підлога має покриття із паркету світло-жовтого кольору. Вікна обладнані світлозахисними пристроями у вигляді внутрішніх регульованих жалюзі.

У приміщенні розташовано 6 робочих місць, обладнаних сучасними персональними комп'ютерами з необхідними периферійними пристроями, один лазерний принтер, 6 телефонів. Для зберігання робочої документації та науково-технічної літератури передбачена шафа. Загальний вигляд обраного виробничого приміщення представлено на рис. 4.1.

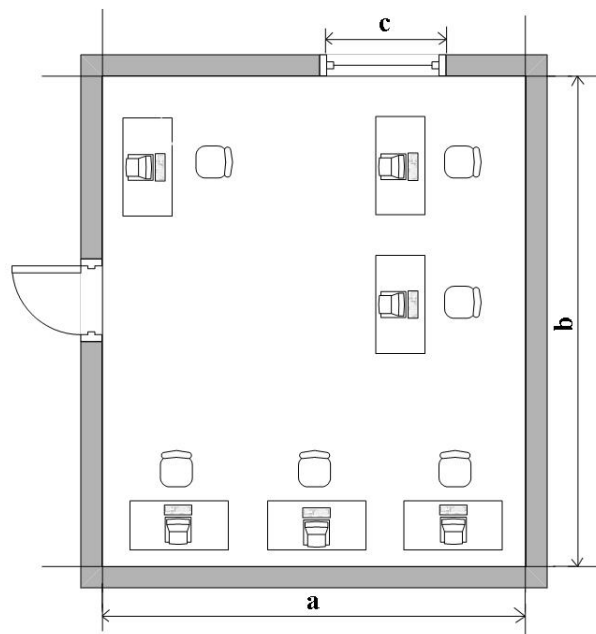


Рисунок 4.1 – Загальний вигляд серверного приміщення

Напруга джерела живлення електроспоживної техніки – 220 В з глухо заземленою нейтралю, з частотою 50 Гц згідно НПАОП 0.00-1.28-2010 [18]. Електромережа виконана у вигляді трипровідної з дотриманням усіх вимог нормативних документів. За безпекою ураження електричним струмом приміщення відноситься до приміщень без підвищеної небезпеки ураження електричним струмом.

Мікрокліматичні умови у літній період забезпечується системою кондиціонування, потужності якої вистачає для забезпечення комфортних умов праці. У зимовий період опалення здійснюється центральною системою, яка

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей
забезпечує необхідний тепловий режим відповідно до вимоги ДБН В.2.5-67:2013 [19].

Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, які нормуються санітарними нормами й правилами ДБН В.2.5-28-2006 [20]. Для управління освітленням користуватися одним або декількома вимикачами і розташовувати їх поряд з дверима на висоті 1,5 м від рівня підлоги.

Пожежна безпека в обраному виробничому приміщенні забезпечується дотриманням вимог НПАОП 0.00-1.28-10 [18].

У приміщенні технічного відділу причинами шуму і вібрації є апарати, і прилади устаткування: друкуючі пристрої, комп'ютери, кондиціонер та ін.

Наведені у табл. 4.1 відомості є вихідними даними для виконання подальших розрахункових робіт даного розділу дипломної роботи бакалавра.

Таблиця 4.1 – Фактори умов праці на робочому місці серверної

№	Параметри	Позначення	Розмірність	Значення
1	Довжина приміщення	a	м	6,5
2	Ширина приміщення	b	м	6,0
3	Висота приміщення	H	м	3,5
4	Ширина вікна	c	м	2,0
5	Висота вікна	d	м	2,4
6	Кількість вікон	n_6	шт.	2
7	Висота верхнього краю вікна відносно умовної робочої поверхні	h	м	2,5
8	Відстань розрахункової точки (робочої поверхні) до зовнішньої стіни	l	м	4,8
9	Висота карнізу протилежної будівлі відносно підвіконня	H'	м	8,0
10	Відстань до затіняючої будівлі	D	м	28,0

Продовження таблиці 4.1

№	Параметри	Позначення	Розмірність	Значення
11	Орієнтація світлових отворів	ОСО		ПНЗ
12	Точність зорових робіт	T	-	Середньої точності (IV розряд зорової роботи)
13	Вид світлопропускаючого матеріалу	-	-	Подвійне листове віконне скло
14	Вид віконної рами	-	-	Спарена дерев'яна рама
15	Сонцезахисні пристрої	-	-	Внутрішні регульовані жалюзі
16	Стан стелі	-	-	Підвісна стеля світло-сірого кольору
17	Стан стін	-	-	Гладка поверхня кольору – біла слонова кістка
18	Стан поверхні підлоги	-	-	Паркет світло-жовтого кольору
19	Кількість робочих місць	<i>пл</i>	шт.	5

Для інтегральної оцінки умов праці [21] в обраному виробничому приміщенні слід скористатися даними табл. 4.1 та здійснити оцінку питомої ваги кожного із представлених там факторів виробничого середовища та трудового процесу, скориставшись наведеними у додатку А даного розділу роботи критеріями бальної оцінки.

У табл. 4.2 представлені параметри, що необхідні для інтегральної оцінки умов праці.

x_{ni} – нормативне значення i – того фактору умов праці (прийняті значення відповідають оптимальному (допустимому) класу умов праці згідно з Гігієнічною класифікацією);

x_{abi} – дійсне значення i – того фактору умов праці;

x_{x_i} – оцінка i – того фактору умов праці, балів;

t_i – тривалість дії i – того фактору умов праці, хв.;

t_{num_i} – відносна тривалість дії i – того фактору умов праці (за прийнятої тривалості робочої зміни $t_p = 480$ хв.), хв., тобто:

$$t_{num_i} = \frac{t_i}{t_p} = \frac{t_i}{480}; \quad (4.1)$$

x_{ϕ_i} – фактична оцінка питомої ваги i – того фактору умов праці, балів, а саме:

$$x_{\phi_i} = x_{x_i} t_{num_i} = x_{x_i} \frac{t_i}{480}. \quad (4.2)$$

За даними табл. 4.2 визначаємо елемент умов праці, який одержав у балах найбільшу оцінку x_{max} . Принципово таких елементів може бути декілька.

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

Таблиця 4.2 – Параметри, що необхідні для розрахунку оцінки умов праці на робочому місці

№ з/п	Фактор умов праці на робочому місці	Нормоване значення фактора $x_{нi}$	Оцінка фактора		Тривалість дії фактора		Фактична оцінка питомої ваги фактора x_{fi}
			Абсолютна $x_{абi}$	У балах x_{xi}	Хвилин t_i	У долях робочої зміни $t_{пит i}$	
1	Температура повітря на робочому місці °С - теплий період	23...25	25	1	480	1	1
		21...23	23	1	480	1	1
2	Відносна вологість повітря на РМ, %	40..60	50	1	480	1	1
3	Швидкість руху повітря на РМ, м/с	<0,2	0,2	2	420	0,875	1,75
4	Освітленість на РМ, лк	400	200	3	300	0,625	1,85
5	Мінімальний розмір об'єкта розпізнавання, мм	>1	0,5	2	300	0,625	1,25
6	Виробничий шум, дБА	<1	0,9	1	480	1	1
7	Інтенсивність теплового випромінювання, Вт/м ²	≤140	50	2	240	0,5	1
8	Токсична речовина, озон, поліхромовані біфеніли, протипожежні бромовані компоненти, кратність перевищення ГДК	≤1	-	1	480	1	1
9	Виробничий пил (паперовий), кратність перевищення ГДК	≤1	0,6	2	480	1	2
10	Робоче місце (РМ), поза та переміщення просторі у	РМ стаціонарне, маса переміщення до 5 кг	РМ стаціонарне, маса переміщення до 5 кг	1	480	1	1
11	Кількість важливих об'єктів спостереження	<5	3	1	480	1	1
12	Тривалість зосередженого спостереження, % часу зміни	<25	55	3	360	0,75	2,25

Таким елементом являється елемент x_{15} , який пов'язаний з тривалістю роботи за добу, тобто $x_{max} = x_4 = x_{12} = 3$. Даний елемент вважається визначаючим.

Далі розраховується:

1. Середній бал усіх елементів крім визначаючого \bar{x} , балів:

$$\bar{x} = \frac{\sum_{i=1}^{n-1} x_{\phi i}}{n-1}, \quad (4.3)$$

де n – фактична кількість врахованих елементів умов праці (у даному випадку $n = 13$).

Тоді:

$$\bar{x} = \frac{\sum_{i=1}^{n-1} x_{\phi i}}{n-1} = \frac{1+1+1+0,875+0,625+0,625+1+0,5+1+1+1+1+0,75}{13-1} = \frac{11,75}{12} = 0,98, \quad (4.4)$$

2. Інтегральна бальна оцінка умов праці на робочому місці [21] у відділі I_n , балів.

$$I_n = 10 \times \left(x_{max} + \bar{x} \frac{6-x_{max}}{6} \right) = 10 \times \left(3 + 0,98 \times \frac{6-3}{6} \right) = 39,8 \quad (4.5)$$

Отримане значення інтегральної оцінки умов праці порівнюємо зі значеннями, наведеними в додатку Б даного розділу магістерської наукової роботи і робимо висновок відносно категорії важкості праці.

Згідно даними додатку Б умови праці на визначеному робочому місці [21] відносяться до III категорії, коли спостерігається відхилення від ГДК і ГДР факторів трудового процесу та допустимих величин психофізіологічних факторів

В наступному підрозділі представлено матеріали щодо визначення параметрів системи вентиляції, призначеної для забезпечення необхідної чистоти повітряного середовища в серверному приміщенні.

Розрахунок припливно-витяжної вентиляції виконується в залежності від розв'язуваних завдань для конкретного приміщення. При цьому основними

параметрами, які регулюються вентиляційними агрегатами, є вологість, температура. При монтажі витяжки забезпечується одноразовий повітрообмін, додавання припливної вентиляції обмін робить дворазовим.

Таблиця 4.3 – Дані для визначення повітропродуктивності припливно-витяжної системи вентиляції

$t_{вх},$ °C	$t_{вих},$ °C	$d_{вх},$ г/кг с.п.	$d_{вих},$ °C г/кг с.п.	$C_{вх},$ мг/м ³	$C_{вих},$ мг/м ³	Q, кВт	W, кг/год	M, г/год	V, м ³
15	23	10,5	15,0	305	1450	40	11,0	2850	650

Принципова схема припливно-витяжної системи вентиляції представлена на рис. 4.2.

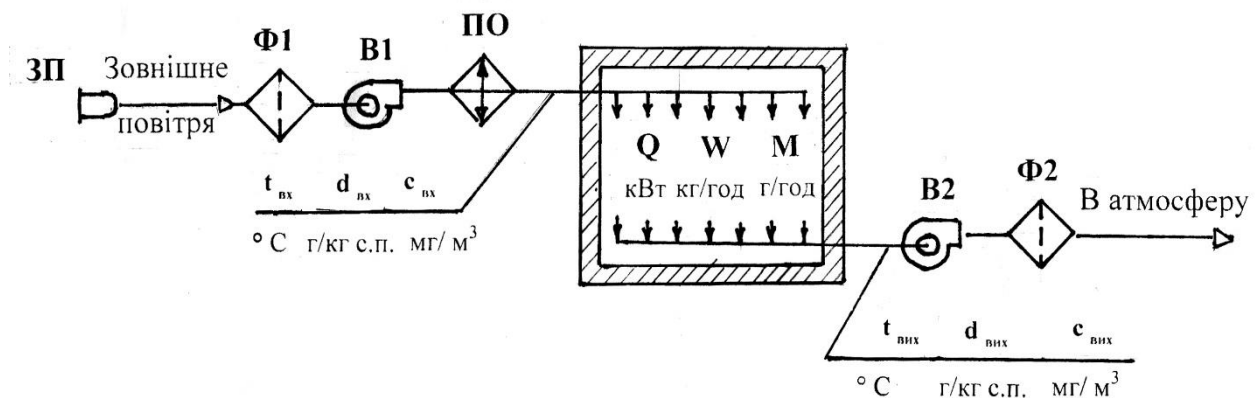


Рисунок 4.2 – Принципова схема припливно-витяжної системи вентиляції: ЗП – забірний пристрій; Ф1, Ф2 – фільтри, В1, В2 – вентилятори; ПО – повітроохолодник; Q, W, M – відповідно теплоприпливи, вологондлишки та надходження шкідливих речовин у виробниче приміщення.

Розрахунок виконується для теплого періоду року за умови рівності повітропродуктивностей припливної та витяжної гілок припливно-витяжної системи вентиляції у такій послідовності:

Повітропродуктивність системи вентиляції, що необхідна для компенсації теплоприпливів у виробничому приміщенні V_B , м³/год.

$$V_B = 3600 \cdot Q / (c_p \rho_p (t_{вих} - t_{вх})) \quad (4.6)$$

де: $c_{\text{п}}$ – ізобарна теплоємність повітря (за нормальних умов $c_{\text{п}} = 1,01$ кДж/(кг·°C));

$\rho_{\text{п}}$ – густина повітря (за нормальних умов $\rho_{\text{п}} = 1,205$ кг/м³).

$$V_{\text{в}} = 3600 \cdot 40 / (1,01 \cdot 1,205 (23 - 15)) = 14790 \text{ м}^3/\text{год}. \quad (4.7)$$

Повітропродуктивність системи вентиляції, що необхідна для компенсації вононадлишків у виробничому приміщенні $V_{\text{в}}^I, \text{ м}^3/\text{год}$.

$$V_{\text{в}}^I = 1000 \cdot W / (\rho_{\text{п}} (d_{\text{вих}} - d_{\text{вх}})), \quad (4.8)$$

де: $c_{\text{п}}$ – ізобарна теплоємність повітря (за нормальних умов $c_{\text{п}} = 1,01$ кДж/(кг·°C));

$\rho_{\text{п}}$ – густина повітря (за нормальних умов $\rho_{\text{п}} = 1,205$ кг/м³).

$$V_{\text{в}}^I = 1000 \cdot W / (\rho_{\text{п}} (d_{\text{вих}} - d_{\text{вх}})) = 1000 \cdot 11,0 / (1,205 (15 - 10,5)) = 2029 \text{ м}^3/\text{год}. \quad (4.9)$$

Повітропродуктивність системи вентиляції, що необхідна для компенсації надходжень шкідливих компонентів повітряного середовища виробничого приміщення $V_{\text{в}}^{II}, \text{ м}^3/\text{год}$.

$$V_{\text{в}}^{II} = 1000 \cdot M / (c_{\text{вих}} - c_{\text{вх}}) = 1000 \cdot 2850 / (1450 - 305) = 2490 \text{ м}^3/\text{год} \quad (4.10)$$

Кратність циркуляції повітря (кратність повітрообміну), що забезпечується при роботі системи свентиляції $\kappa, \text{ год}^{-1}$.

$$\kappa = V_{\text{в}} / V_{\text{п}} = 14790 / 650 = 23 \text{ год}^{-1}. \quad (4.11)$$

Отримане значення кратності циркуляції слід порівняти з його нормативним значенням $\kappa_{\text{н}}$ і, у разі перебільшення нормативного значення відносно розрахункового, слід зробити перерахунок повітропродуктивності системи вентиляції, а саме:

$$V_{\text{в}} = \kappa_{\text{н}} \cdot V_{\text{п}} \text{ м}^3/\text{год}. \quad (4.12)$$

У результаті отримано кратність циркуляції повітря 23 год^{-1} , норма кратності повітрообміну серверної кімнати $5-10 \text{ год}^{-1}$ [22], отже отримане значення є задовольняючим.

Висновки до розділу 4

Виконано аналіз умов праці в серверному приміщенні розробки системи для діагностики зору. Перевірено забезпечення вимог охорони праці. Виявлено,

що оцінка умов праці на робочому місці відноситься до III категорії, коли спостерігається робота, що відхиляються від ГДК і ГДР та допустимих рівнів психофізіологічних факторів. З метою їх покращення розраховано припливно-втяжну вентиляцію кратністю циркуляції повітря 23 год^{-1} , норма кратності повітрообміну серверної кімнати $5-10 \text{ год}^{-1}$ [23], отже отримане значення є задовольняючим.

Для покращення роботи підприємства були проведені наступні заходи:

- додано освітлення на робочому місці;
- облаштування поверхонь інтер'єру з максимально можливим коефіцієнтом відбиття;
- зменшено тривалість зосередженого спостереження;
- змінено режим праці та відпочинку на обґрунтований.

ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи було створено систему діагностики зору для профілактики захворювань очей, яка надає можливість користувачеві виконувати діагностування зору у таких напрямках, як: гострота зору, кольоровідчуття, перевірка контрастної чутливості, перевірка на наявність далекозорості або короткозорості, перевірка на наявність астигматизму та перевірка на наявність патології макули. Проаналізовано актуальність та проблеми сталого розвитку діагностики зору. У ході роботи було розглянуто аналоги онлайн-ресурсів, які надають можливість самостійного діагностування зору. Було проаналізовано предметну сферу для більш чіткого розуміння як краще реалізувати інформаційну систему. Досліджено основні принципи створення застосунку для проведення діагностики зору, використовуючи методи та алгоритми Data Mining.

При створенні системи діагностики зору для профілактики захворювань очей для класифікації результатів було використано наївний класифікатор Байєса. Завдяки наївному класифікатору розроблена система здатна достатньо точно виконувати діагностування патологій та порушень стану зору користувача. У медицині з їх допомогою побудовані експертні системи для постановки діагнозів на основі правил, що описують поєднання різних симптомів різних захворювань. Правила допомагають вибирати показання (протипоказання), передбачати наслідки призначеного курсу лікування.

При реалізації системи для діагностики зору для профілактики захворювань зору було вирішено створити БД, у якій зберігатимуться результати тестувань користувача, які у подальшому будуть використовуватися для аналізування та відображення статистики.

В результаті виконання кваліфікаційної роботи були досягнуті поставлені цілі та було зроблено висновок, що система відпрацьовує коректно та повертає досить точний та зрозумілий результат користувачу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Reasons Why Your Eyes Are Important. *Ultralase*: веб-сайт. URL: <https://www.ultralase.com/> (дата звернення: 02.05.2022).
2. Симптоми захворювання очей *Центр Ока*: веб-сайт. URL: <https://centeroka.com.ua/uk/> (дата звернення: 02.05.2022).
3. Визначення гостроти зору *Оптималь*: веб-сайт. URL: <https://optimalclinic.com/uk> (дата звернення: 03.05.2022).
4. Порушення кольоровідчуття: причини, типи і опис, способи корекції, відгуки *Likar.net.ua*: веб-сайт. URL: <http://likar.net.ua/> (дата звернення: 03.05.2022).
5. Таблиця Рабкіна з відповідями для водіїв *Все про зір*: веб-сайт. URL: <http://kompendium.com.ua/> (дата звернення: 03.05.2022).
6. Джек Дальтон *Wikipedia*: веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0_%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0 (дата звернення: 04.05.2022).
7. Тест на контрастную чувствительность Спета/Ричмана *Good Clinical Practice*: веб-сайт. URL: <https://ichgcp.net/ru> (дата звернення: 04.05.2022).
8. Дуохромний тест *Офтальмологическая клиника доктора Куренкова*: веб-сайт. URL: <https://www.visus-novus.ru/> (дата звернення: 05.05.2022).
9. Перевірка зору на астигматизм *Оптик Сити*: веб-сайт. URL: <https://www.optic-city.ru/> (дата звернення: 10.05.2022).
10. Тест на стигматизм *Эксимер*: веб-сайт. URL: <https://excimerclinic.ru/> (дата звернення: 15.05.2022).
11. Діагностика та лікування вікової макулодистрофії (ВМД) *Центр Ока*: веб-сайт. URL: <https://centeroka.com.ua/uk/> (дата звернення: 05.05.2022).
12. Амслер, Марк *Wikipedia*: веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0_%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0

- [%B2%D0%BD%D0%B0%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0](#) (дата звернення: 05.05.2022).
13. Що таке наївний байєс? веб-сайт. URL: <https://uk.theastrologypage.com/> (дата звернення: 17.05.2022).
14. Naive Bayes Classifier Towards data science: веб-сайт. URL: <https://towardsdatascience.com/> (дата звернення: 07.05.2022).
15. Мова C# і платформа .Net Metanit: веб-сайт. URL: <https://metanit.com/> (дата звернення: 20.05.2022).
16. Графічний інтерфейс користувача Wikipedia: веб-сайт. URL: <https://uk.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0> (дата звернення: 07.05.2022).
17. Введення у Entity Framework Metanit: веб-сайт. URL: <https://metanit.com/> (дата звернення: 20.05.2022).
18. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин: Закон України від 14 лют. 2018 р. № 207. URL: <https://zakon.rada.gov.ua/laws/show/z0293-10#Text> (дата звернення: 26.05.2022).
19. Опалення, вентиляція та кондиціонування: накази Міністерства регіонального розвитку, будівництва та житлово комунального господарства України від 01 січ. 2014 р. № 410. URL: <http://kbu.org.ua/assets/app/documents/dbn2/100.1.%20%D0%94%D0%91%D0%9D%20%D0%92.2.5-67~2013.%20%D0%9E%D0%BF%D0%B0%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F,%20%D0%B2%D0%B5%D0%BD%D1%82%D0%B8%D0%BB%D1%8F%D1%86%D1%96%D1%8F%20%D1%82%D0%B0%20%D0%BA%D0%BE%D0%BD%D0%B4%D0%B8%D1%86%D1%96.pdf> (дата звернення: 26.05.2022).

20. Природне і штучне освітлення: наказ Мінрегіону України від 01 вер. 2012 р. № 438 URL: <https://www.minregion.gov.ua/wp-content/uploads/2017/12/95.3.-DBN-V.2.5-28-2006.-Zmina-N2.-Prirodne-i-shtuchne-osv.pdf> (дата звернення: 26.05.2022).
21. Оцінка категорії важкості праці. StudFiles: вебсайт. URL: <https://studfile.net/> (дата звернення: 26.05.2022).
22. Визначення необхідного повітрообміну приміщень. Рекомендації до проектування. Ventus: вебсайт. URL: <https://vents.ua/> (дата звернення: 28.05.2022).
23. Rish I. An empirical study of naive Bayes classifier / I. Rish. – IBM Research Division, 2001. – pp. 1-6.
- 24.10. McCallum A, Nigam K (1998) A comparison of event models for Naive Bayes text classification. In: AAI-98 workshop on learning for text categorization. AAAI Press, Menlo Park, pp 41–48.
25. Lewis D (1998) Naive Bayes at forty: the independence assumption in information retrieval. In: Proceedings of the 10th European conference on machine learning (ECML-98), Chemnitz. Springer, Berlin, pp 4–15.
26. Що таке Data Mining? *MAD Data*: веб-сайт URL : <https://maddata.agency/> (дата звернення: 28.05.2022).
27. Zhang H. The Optimality of Naive Bayes. URL: <https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf> (дата звернення: 28.05.2022)
28. Naive Bayes. URL: https://scikit-learn.org/stable/modules/naive_bayes.html (дата звернення: 29.05.2022)
29. Наскільки важлива регулярна перевірка зору. *СМ-Клініка*: веб-сайт URL: <https://www.smclinic-spb.ru/> (дата звернення: 29.05.2022)
- 30.6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R. *Analytics Vidhya*: веб-сайт URL: <https://www.analyticsvidhya.com/blog/> (дата звернення: 29.05.2022)

ДОДАТОК А

Код програмної реалізації авторизації

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class Login : Form
    {
        string salt = "QWERTYUIOPASDFGHJKLZXCVBNM";
        public static readonly UnicodeEncoding ByteConverter = new UnicodeEncoding();

        MD5 md5;
        SqlConnection cn;
        SqlCommand cmd;
        SqlDataReader dr;

        public Login()
        {
            InitializeComponent();
        }
        private void Login_Load(object sender, EventArgs e)
        {
            cn = new SqlConnection(@"Data Source=DESKTOP-7P35733;Initial
Catalog=vision_diagnostics_db;Integrated Security=True");
            cn.Open();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form1 f1 = new Form1();
            f1.Show();
            this.Close();
        }
        static string HashPassword(string password)
        {
            byte[] bytes = Encoding.ASCII.GetBytes("2011" + password);
            using (MD5 md5 = new MD5CryptoServiceProvider())
            {
                byte[] hash = md5.ComputeHash(bytes);
                StringBuilder sb = new StringBuilder(hash.Length * 2);
                foreach (var b in hash)
                {
                    sb.Append(b.ToString("x2"));
                }
                return sb.ToString();
            }
        }

        public bool login_valid()
        {

```

```

const string myConnectionString = @"Data Source = DESKTOP-7P35733; Initial
Catalog = vision_diagnostics_db; Integrated Security = True";

string login = textBox1.Text;
string password = HashPassword(textBox2.Text);

bool authenticated;
using (SqlConnection connection = new SqlConnection(myConnectionString))
{
    SqlCommand command = new SqlCommand();
    command.CommandText = "SELECT COUNT(*) FROM [customers] WHERE [login] =
@login AND [password] = @password";
    command.Connection = connection;
    command.Parameters.AddWithValue("@login", login);
    command.Parameters.AddWithValue("@password", password);

    connection.Open();
    authenticated = 1 == (int)command.ExecuteScalar();
}

if (!authenticated)
{
    return false;
}
else {
    return true;
}
}

public bool UserExists(string login, string password)
{
    cn = new SqlConnection(@"Data Source=DESKTOP-7P35733;Initial
Catalog=vision_diagnostics_db;Integrated Security=True");
    cn.Open();
    cmd = new SqlCommand("SELECT login, password FROM [customers] ", cn);
    SqlDataReader rd = cmd.ExecuteReader();

    bool comand = false;
    while (rd.Read())
    {
        string userLogin = rd["login"].ToString();
        string passwordDB = rd["password"].ToString();
        if (login == userLogin && password == passwordDB)
        {
            cn.Close();
            comand = true;
            return comand;
        }
    }
    return comand;
}

private void button2_Click(object sender, EventArgs e)
{
    bool user;
    if (textBox1.Text != string.Empty || textBox2.Text != string.Empty)
    {
        if (login_valid() == true)
        {
            MessageBox.Show("You have successfully logged in!", "Welcome",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            MainForm main = new MainForm(textBox1.Text);

```

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

```
main.Show();
this.Hide();

}
else if (textBox1.Text == "")
{
    MessageBox.Show("Please enter your username", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else if (textBox2.Text == "")
{
    MessageBox.Show("Please enter your password", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
    textBox2.Text = "";
    MessageBox.Show("Check if the username entered is correct and re-enter
your password", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    Recovery recover = new Recovery(textBox1.Text.ToString());
    recover.Show();
    this.Hide();
}
}
```

ДОДАТОК Б

Код програмної реалізації реєстрації

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class SignUp : Form
    {
        bool needRedRect = false;

        public SignUp()
        {
            InitializeComponent();
        }
        private void SignUp_Paint(object sender, PaintEventArgs e)
        {
            if (needRedRect)
            {
                Pen pen = new Pen(Color.Red, 4);
                Rectangle rect = new Rectangle(textBox2.Location, textBox2.Size);
                Rectangle rect2 = new Rectangle(textBox3.Location, textBox3.Size);
                e.Graphics.DrawRectangle(pen, rect);
                e.Graphics.DrawRectangle(pen, rect2);
            }
        }
        static string HashPassword(string password)
        {
            byte[] bytes = Encoding.ASCII.GetBytes("2011" + password);
            using (MD5 md5 = new MD5CryptoServiceProvider())
            {
                byte[] hash = md5.ComputeHash(bytes);
                StringBuilder sb = new StringBuilder(hash.Length * 2);
                foreach (var b in hash)
                {
                    sb.Append(b.ToString("x2"));
                }
                return sb.ToString();
            }
        }

        public string generateRegisterCode()
        {
            Random rnd = new Random();
            string alphabet = "aedfjbntlvwzsqertyophkxcbm".ToUpper();

            var sb = new StringBuilder();
            sb.Append(alphabet[rnd.Next(0, alphabet.Length)]);
            sb.Append(rnd.Next(0, 10));
            sb.Append(rnd.Next(0, 10));
        }
    }
}

```

```

sb.Append(rnd.Next(0, 10));
sb.Append(alphabet[rnd.Next(0, alphabet.Length)]);
sb.Append(alphabet[rnd.Next(0, alphabet.Length)]);
sb.Append(rnd.Next(0, 10));
sb.Append(rnd.Next(0, 10));
return sb.ToString();
}
public bool UserExists(string login)
{
    int countDis = 0;
    using (SqlConnection connection = new SqlConnection(@"Data Source=DESKTOP-
7P35733;Initial Catalog=vision_diagnostics_db;Integrated Security=True"))
    {
        connection.Open();
        var sql = $"SELECT COUNT(*) FROM [customers] WHERE login = @login ";
        using (var sqlComm = new SqlCommand(sql, connection))
        {
            sqlComm.Parameters.AddWithValue("@login", login);
            countDis = int.TryParse(sqlComm.ExecuteScalar()?.ToString(), out int
temp) ? temp : 0;
        }
    }
    return countDis == 1;
}
public void addCustomer(string username, string password, string code)
{
    if (UserExists(username) == false)
    {
        using (SqlConnection connection = new SqlConnection(@"Data Source=DESKTOP-
7P35733;Initial Catalog=vision_diagnostics_db;Integrated Security=True"))
        {
            using (SqlCommand command = new SqlCommand())
            {
                command.Connection = connection;
                command.CommandType = CommandType.Text;
                command.CommandText = "INSERT into [customers] (login, password,
registerCode) VALUES (@username, @userpassword, @code)";
                command.Parameters.AddWithValue("@username", username);
                command.Parameters.AddWithValue("@userpassword", password);
                command.Parameters.AddWithValue("@code", code);
                try
                {
                    connection.Open();
                    int recordsAffected = command.ExecuteNonQuery();
                    MessageBox.Show("You have successfully registered!", "Welcome",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    MainForm mainForm = new MainForm(username);
                    mainForm.Show();
                    this.Hide();
                }
                catch (SqlException)
                {
                    MessageBox.Show("An error occurred while registering, please try
again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                finally
                {
                    connection.Close();
                }
            }
        }
    }
}

```

```

    }
  }
}
else
{
    MessageBox.Show("A user with this login already exists. Please change your
login and try again.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void button1_Click(object sender, EventArgs e)
{
    Form1 f1 = new Form1();
    f1.Show();
    this.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    string username, password;
    string path = @"c:\tmp\RegisterCode.txt";

    if (textBox1.Text != string.Empty && textBox2.Text != string.Empty &&
textBox3.Text != string.Empty)
    {
        if (textBox2.Text == textBox3.Text)
        {
            username = textBox1.Text;
            password = HashPassword(textBox2.Text);
            string code = generateRegisterCode();
            try
            {
                addCustomer(username, password, code);
            } catch (Exception ex)
            {
                MessageBox.Show(ex.StackTrace);
            }
            MessageBox.Show("Your register code is - " + code.ToString(),
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
            using (FileStream fs = File.Create(path))
            {
                byte[] info = new UTF8Encoding(true).GetBytes(code);
                fs.Write(info, 0, info.Length);
            }
        }
        else
        {
            needRedRect = true;
            Invalidate();
            MessageBox.Show("Passwords do not match, please try again!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("All fields are expected to be filled!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}

```

```
    }  
  }  
  private void SignUp_Load(object sender, EventArgs e){  
  }  
}
```

ДОДАТОК В

Код програмної реалізації відновлення паролю користувача

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class Recovery : Form
    {
        public string user, connection = @"Data Source=DESKTOP-7P35733;Initial
Catalog=vision_diagnostics_db;Integrated Security=True";
        SqlConnection cn;
        SqlCommand cmd;
        SqlDataReader dr;
        bool needRedRect = false;

        public Recovery(string user)
        {
            InitializeComponent();
            this.user = user;
            textBox1.Text = user.ToString();
            label3.Visible = false;
            label4.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;
            button3.Visible = false;
            Tooltip tooltip1 = new Tooltip();
            tooltip1.SetToolTip(pictureBox1, "If you have forgotten your registration code,
you can find it along the path c:\\tmp\\RegisterCode.txt");
        }

        static string HashPassword(string password)
        {
            byte[] bytes = Encoding.ASCII.GetBytes("2011" + password);
            using (MD5 md5 = new MD5CryptoServiceProvider())
            {
                byte[] hash = md5.ComputeHash(bytes);
                StringBuilder sb = new StringBuilder(hash.Length * 2);
                foreach (var b in hash)
                {
                    sb.Append(b.ToString("x2"));
                }
                return sb.ToString();
            }
        }

        public bool UserExists(string login, string code)
        {

```



```

cn = new SqlConnection(@"Data Source=DESKTOP-7P35733;Initial
Catalog=vision_diagnostics_db;Integrated Security=True");
cn.Open();
cmd = new SqlCommand("SELECT login, registerCode FROM [customers] ", cn);
SqlDataReader rd = cmd.ExecuteReader();

bool comand = false;
while (rd.Read())
{
    string userLogin = rd["login"].ToString();
    string userCode = rd["registerCode"].ToString();
    if (login == userLogin && code == userCode)
    {
        cn.Close();
        comand = true;
        return comand;
    }
}
return comand;
}

public void updateCustomer(string username, string password)
{
    const string myConnectionString = @"Data Source = DESKTOP-7P35733; Initial
Catalog = vision_diagnostics_db; Integrated Security = True";

    using (SqlConnection connection = new SqlConnection(myConnectionString))
    {
        using (SqlCommand command = new SqlCommand())
        {
            connection.Open();
            var cmd = new SqlCommand("UPDATE [customers] SET [password]=@password
WHERE login =@login");
            cmd.Parameters.AddWithValue("@password", password);
            cmd.Parameters.AddWithValue("@login", username);

            try
            {
                cmd.Connection = connection;
                cmd.ExecuteNonQuery();
                connection.Close();
                MessageBox.Show("You have successfully changed your password!" ,
>Password changed!", MessageBoxButtons.OK, MessageBoxIcon.Information);
                Login login = new Login();
                login.Show();
                this.Hide();
            }
            catch (SqlException e)
            {
                MessageBox.Show("An error occurred while trying to change your
password, please try again!"+ e.StackTrace, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
            finally
            {
                connection.Close();
            }
        }
    }
}
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    Login login = new Login();
    login.Show();
    this.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    string username = textBox1.Text;
    string registerCode = textBox2.Text;
    string password = "";
    if(textBox3.Text != string.Empty && textBox4.Text != string.Empty)
    {
        if (textBox3.Text == textBox4.Text)
        {
            password = HashPassword(textBox3.Text);
            try
            {
                updateCustomer(username, password);
            } catch (Exception ex)
            {
                MessageBox.Show(ex.StackTrace);
            }
        }
        else
        {
            needRedRect = true;
            Invalidate();
            MessageBox.Show("Passwords do not match, please try again!", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("All fields are expected to be filled!", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void Recovery_PAint(object sender, PaintEventArgs e)
{
    if (needRedRect)
    {
        Pen pen = new Pen(Color.Red, 4);
        Rectangle rect = new Rectangle(textBox3.Location, textBox3.Size);
        Rectangle rect2 = new Rectangle(textBox4.Location, textBox4.Size);
        e.Graphics.DrawRectangle(pen, rect);
        e.Graphics.DrawRectangle(pen, rect2);
    }
}

private void toolTip1_Popup(object sender, PopupEventArgs e){}
private void Recovery_Load(object sender, EventArgs e){}
private void pictureBox1_Click(object sender, EventArgs e){}

private void button1_Click(object sender, EventArgs e)
{
    string username, registerCode;
}

```

```
if (textBox1.Text != string.Empty && textBox2.Text != string.Empty)
{
    username = textBox1.Text;
    registerCode = textBox2.Text;
    bool exist = false;
    try
    {
        exist = UserExists(username, registerCode);
    } catch (Exception ex)
    {
        MessageBox.Show(ex.StackTrace);
    }
    if (exist)
    {
        label3.Visible = true;
        label4.Visible = true;
        textBox3.Visible = true;
        textBox4.Visible = true;
        button3.Visible = true;
    }
    else
    {
        MessageBox.Show("User is not found. Check the correctness of the login
and code!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    MessageBox.Show("All fields are expected to be filled!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}
```

ДОДАТОК Г

Код програмної реалізації головного меню системи

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class MainForm : Form
    {
        public string user;
        public MainForm(string user)
        {
            InitializeComponent();
            this.user = user;
        }

        private void MainForm_Load(object sender, EventArgs e){}
        private void label2_Click(object sender, EventArgs e){}
        private void groupBox2_Enter(object sender, EventArgs e) { }

        private void pictureBox1_Click(object sender, EventArgs e) //visual act. test
        {
            Visual_acuity acuity = new Visual_acuity(user);
            acuity.Show();
            this.Hide();
        }

        private void pictureBox2_Click(object sender, EventArgs e)//color blind
        {
            Blindness blindness = new Blindness(user);
            blindness.Show();
            this.Hide();
        }

        private void pictureBox3_Click(object sender, EventArgs e) //contrast
        {
            Contrast_sensitivity sensitivity = new Contrast_sensitivity(user);
            sensitivity.Show();
            this.Hide();
        }

        private void pictureBox4_Click(object sender, EventArgs e) // duochrome
        {
            Duochrome duochrome = new Duochrome(user);
            duochrome.Show();
            this.Hide();
        }

        private void pictureBox5_Click(object sender, EventArgs e) //astigmatism
        {
            Astigmatism astigmatism_test = new Astigmatism(user);
            astigmatism_test.Show();
            this.Hide();
        }
    }
}
```

```
}  
  
private void pictureBox6_Click(object sender, EventArgs e) // macular pathology  
{  
    Amsler amsler_test = new Amsler(user);  
    amsler_test.Show();  
    this.Hide();  
}  
  
private void button1_Click(object sender, EventArgs e) // personal account  
{  
    Account account = new Account(user);  
    account.Show();  
    this.Hide();  
}  
  
private void button2_Click(object sender, EventArgs e) //close  
{  
    DialogResult result;  
    result = MessageBox.Show("Do you really want to exit the application?",  
"Attention", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  
    if (result == System.Windows.Forms.DialogResult.Yes)  
    {  
        Close();  
    }  
}  
}  
}
```

ДОДАТОК Д

Код програмної реалізації діагностики наявності патології макули

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class Astigmatism : Form
    {
        string username;
        double answ_1_left = 0, answ_2_left = 0, answ_3_left = 0, answ_1_right = 0,
answ_2_right = 0, answ_3_right = 0;
        public Astigmatism(string user)
        {
            InitializeComponent();
            this.username = user;
            button4.Visible = false;
        }

        private void Astigmatism_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e) //next
        {
            if (radioButton1.Checked) answ_1_left = 1;
            if (radioButton4.Checked) answ_1_right = 1;
            pictureBox1.Image = Properties.Resources.astigmat_2;
            int numb = Int32.Parse(label4.Text);
            label4.Text = (++numb).ToString();
            radioButton1.Text = "Some of the lines get darker than others";
            radioButton2.Text = "The image is distorted, in which some lines are clear,
others \n are blurry";
            radioButton3.Text = "Normal image";
            radioButton4.Text = "Some of the lines get darker than others";
            radioButton5.Text = "The image is distorted, in which some lines are clear,
others \n are blurry";
            radioButton6.Text = "Normal image";
            button1.Visible = false;
            button4.Visible = true;
        }

        private void button4_Click_1(object sender, EventArgs e)
        {
            if (radioButton3.Checked) answ_2_left = 1;
            if (radioButton6.Checked) answ_2_right = 1;
            pictureBox1.Image = Properties.Resources.astigmat_3;
        }
    }
}

```

```

int numb = Int32.Parse(label4.Text);
label4.Text = (++numb).ToString();
radioButton1.Text = "Lined squares are equally black";
radioButton2.Text = "One or more squares appear gray";
radioButton4.Text = "Lined squares are equally black";
radioButton5.Text = "One or more squares appear gray";
radioButton3.Visible = false;
radioButton6.Visible = false;
button1.Visible = false;
button4.Visible = false;
}

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e) //back
{
    MainForm main = new MainForm(username);
    main.Show();
    this.Close();
}

private void button3_Click(object sender, EventArgs e) //result
{
    string left_eye = "", right_eye = "", conclusion = "";
    DateTime date = DateTime.Today;
    if (radioButton1.Checked) answ_3_left = 1;
    if (radioButton4.Checked) answ_3_right = 1;

    double result_left = (answ_1_left + answ_2_left + answ_3_left) / 3 * 100;
    double result_right = (answ_1_right + answ_2_right + answ_3_right) / 3 * 100;

    double total = (answ_1_left + answ_2_left + answ_3_left + answ_1_right +
answ_2_right + answ_3_right) / 6 * 100;

    if (total == 100)
    {
        MessageBox.Show("Congratulations! Astigmatism is absent in both eyes!",
"Result", MessageBoxButtons.OK, MessageBoxIcon.Information);
        left_eye = "normal"; right_eye = "normal"; conclusion = "Astigmatism is
absent in both eyes";
    }
    else if(result_left < 100 && result_right > 80)
    {
        MessageBox.Show("Astigmatism can be observed in the left eye. We recommend
that you be examined by an ophthalmologist for a more accurate diagnosis. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        right_eye = "normal"; left_eye = "pathology"; conclusion = "Astigmatism can
be observed in the left eye";
    }
    else if (result_right < 100 && result_left > 80)
    {
        MessageBox.Show("Astigmatism can be observed in the right eye. We recommend
that you be examined by an ophthalmologist for a more accurate diagnosis. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        left_eye = "normal"; right_eye = "pathology"; conclusion = "Astigmatism can
be observed in the right eye";
    }
    else if (result_right < 30 && result_left < 30)
    {

```


ДОДАТОК Е

Код програмної реалізації діагностики на наявність далекозорості або короткозорості

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class Contrast_sensitivity : Form
    {
        string username; int i = 1;
        float correct_answer;

        public Contrast_sensitivity(string user)
        {
            InitializeComponent();
            this.username = user;
            button2.Visible = false; button4.Visible = false;
            button5.Visible = false; button6.Visible = false; button7.Visible = false;
            checkBox1.Visible = false; checkBox2.Visible = false; checkBox3.Visible = false;
            checkBox4.Visible = false;
        }

        private void Contrast_sensitivity_Load(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (radioButton2.Checked)
            {
                correct_answer++;
            }
            i++;
            label3.Text = "CHOOSE THE BRIGHTEST IMAGE";
            label4.Text = i.ToString();
            pictureBox1.Image = Properties.Resources._3_1;
            pictureBox2.Image = Properties.Resources._3_2;
            pictureBox3.Image = Properties.Resources._3_3;
            pictureBox4.Image = Properties.Resources._3_4;
            button3.Visible = false;
            button4.Visible = true;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked)
            {
                correct_answer++;
            }
        }
    }
}

```

```

i++;
label3.Text = "CHOOSE THE MOST NON-BRIGHT IMAGE";
label4.Text = i.ToString();
pictureBox1.Image = Properties.Resources._4_1;
pictureBox2.Image = Properties.Resources._4_2;
pictureBox3.Image = Properties.Resources._4_3;
pictureBox4.Image = Properties.Resources._4_4;
button4.Visible = false;
button5.Visible = true;
}

private void button5_Click(object sender, EventArgs e)
{
    if (radioButton2.Checked)
    {
        correct_answer++;
    }
    i++;
    label3.Text = "CHOOSE TWO SAME CONTRASTED IMAGES";
    checkBox1.Visible = true; checkBox2.Visible = true; checkBox3.Visible = true;
checkBox4.Visible = true;
    radioButton1.Visible = false; radioButton2.Visible = false; radioButton3.Visible
= false; radioButton4.Visible = false;

    label4.Text = i.ToString();
    pictureBox1.Image = Properties.Resources._5_1;
    pictureBox2.Image = Properties.Resources._5_2;
    pictureBox3.Image = Properties.Resources._5_3;
    pictureBox4.Image = Properties.Resources._5_4;
    button5.Visible = false;
    button6.Visible = true;
}

private void button6_Click(object sender, EventArgs e)
{
    if(checkBox1.Checked && checkBox4.Checked)
    {
        correct_answer++;
    }
    i++;
    label3.Text = "CHOOSE THE DARKEST IMAGE";
    checkBox1.Visible = false; checkBox2.Visible = false; checkBox3.Visible = false;
checkBox4.Visible = false;
    radioButton1.Visible = true; radioButton2.Visible = true; radioButton3.Visible =
true; radioButton4.Visible = true;

    label4.Text = i.ToString();
    pictureBox1.Image = Properties.Resources._6_1;
    pictureBox2.Image = Properties.Resources._6_2;
    pictureBox3.Image = Properties.Resources._6_3;
    pictureBox4.Image = Properties.Resources._6_4;
    button6.Visible = false;
    button7.Visible = true;
}

private void button7_Click(object sender, EventArgs e)
{
    if (radioButton4.Checked)
    {
        correct_answer++;
    }
}

```

```

}
i++;
label3.Text = "CHOOSE THE BRIGHTEST PICTURE";
label4.Text = i.ToString();
pictureBox1.Image = Properties.Resources._7_1;
pictureBox2.Image = Properties.Resources._7_2;
pictureBox3.Image = Properties.Resources._7_3;
pictureBox4.Image = Properties.Resources._7_4;
button7.Visible = false;
button2.Visible = true;

}

private void button2_Click(object sender, EventArgs e)
{
    DateTime date = DateTime.Today;
    string conclusion = "";
    string left_eye, right_eye;

    if (radioButton4.Checked)
    {
        correct_answer++;
    }

    float result = correct_answer / 6 * 100;

    if(result == 100)
    {
        MessageBox.Show("Congratulations! Contrasting sensitivity is " +
Math.Round(result,2).ToString() + "%.", "Result", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        left_eye = right_eye = "normal";
        conclusion = "High contrast sensitivity.";
    }
    else if( result >= 50)
    {
        MessageBox.Show("Your score is above average. The contrast sensitivity is "
+ Math.Round(result, 2).ToString() + "%. Vision needs a mandatory examination. Visit an
ophthalmologist to avoid vision problems.", "Result", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        left_eye = right_eye = "pathology";
        conclusion = "Reduced contrast sensitivity.";
    }
    else
    {
        MessageBox.Show("Your score is below average. The contrast sensitivity is "
+ Math.Round(result, 2).ToString() + "%. Vision needs a mandatory examination. Visit an
ophthalmologist to avoid vision problems.", "Result", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        left_eye = right_eye = "pathology";
        conclusion = "Reduced contrast sensitivity.";
    }
}

using (SqlConnection connection = new SqlConnection(@"Data Source=DESKTOP-
7P35733;Initial Catalog=vision_diagnostics_db;Integrated Security=True"))
{
    using (SqlCommand command = new SqlCommand())
    {
        command.Connection = connection; // <== lacking
        command.CommandType = CommandType.Text;
    }
}

```

Кафедра інтелектуальних інформаційних систем
Система діагностики зору для профілактики захворювань очей

```
command.CommandText = "INSERT into [diagnostic] (username, date, test,
left_eye, right_eye, total_result, conclusion) VALUES (@username, @date,@test, @left_eye,
@right_eye, @total_result, @conclusion)";
command.Parameters.AddWithValue("@username", username);
command.Parameters.AddWithValue("@date", date);
command.Parameters.AddWithValue("@test", "Contrast sensitivity test");
command.Parameters.AddWithValue("@left_eye", left_eye);
command.Parameters.AddWithValue("@right_eye", right_eye);
command.Parameters.AddWithValue("@total_result", Math.Round(result, 2));
command.Parameters.AddWithValue("@conclusion", conclusion);
try
{
    connection.Open();
    int recordsAffected = command.ExecuteNonQuery();
}
catch (SqlException) { }
finally
{
    connection.Close();
}
}
}

private void button1_Click(object sender, EventArgs e)
{
    MainForm main = new MainForm(username);
    main.Show();
    this.Close();
}
}
```

ДОДАТОК Ж

Код програмної реалізації діагностики контрастної чутливості

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace VisionDiagnosticsApplication
{
    public partial class Duochrome : Form
    {
        double answ_left = 0, answ_right = 0;
        string username;
        public Duochrome(string user)
        {
            InitializeComponent();
            this.username = user;
        }

        private void Duochrome_Load(object sender, EventArgs e)
        {
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MainForm main = new MainForm(username);
            main.Show();
            this.Close();
        }

        private void radioButton6_CheckedChanged(object sender, EventArgs e)
        {
        }

        private void radioButton5_CheckedChanged(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DateTime date = DateTime.Today;
            string left_eye = "", right_eye = "", conclusion = "";
            if (radioButton1.Checked && radioButton6.Checked)
            {
                answ_left = 1; answ_right = 1;
            } else if (radioButton1.Checked)
        }
    }
}

```

```

{
    answ_left = 1;
}
else if (radioButton6.Checked)
{
    answ_right = 1;
}
double result = (answ_left + answ_right) / 2 * 100;

if(result == 100)
{
    MessageBox.Show("The duochrome test determined that your vision is
emmetropia (normal). ", "Result", MessageBoxButtons.OK, MessageBoxIcon.Information);
    left_eye = right_eye = "normal"; conclusion = "Emmetropia";
}
else if(result >= 50)
{
    if(radioButton2.Checked && radioButton6.Checked)
    {
        MessageBox.Show("If the characters appear clearer on a red background,
then there is a high probability of myopia(-) in the left eye. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        left_eye = "pathology"; right_eye = "normal"; conclusion = "Myopia(-) in
left eye";
    }
    else if(radioButton3.Checked && radioButton6.Checked)
    {
        MessageBox.Show("If the characters appear clearer on a green background,
then there is a high probability of hyperopia(+) in the left eye. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        left_eye = "pathology"; right_eye = "normal"; conclusion = "Hyperopia(+)
in left eye";
    }
    else if (radioButton1.Checked && radioButton4.Checked)
    {
        MessageBox.Show("If the characters appear clearer on a red background,
then there is a high probability of myopia(-) in the right eye. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        right_eye = "pathology"; left_eye = "normal"; conclusion = "Myopia(-) in
right eye";
    }
    else if (radioButton1.Checked && radioButton3.Checked)
    {
        MessageBox.Show("If the characters appear clearer on a green background,
then there is a high probability of hyperopia(+) in the right eye. ", "Result",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        right_eye = "pathology"; left_eye = "normal"; conclusion = "Hyperopia(+)
in right eye";
    }
}
else
{
    if(radioButton2.Checked && radioButton5.Checked)
    {
        MessageBox.Show("If the characters appear clearer on a red background,
then there is a high probability of myopia(-).", "Result", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        right_eye = "pathology"; left_eye = "pathology"; conclusion = "Myopia(-
)";
    }
    else if(radioButton3.Checked && radioButton4.Checked)
    {

```

