

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

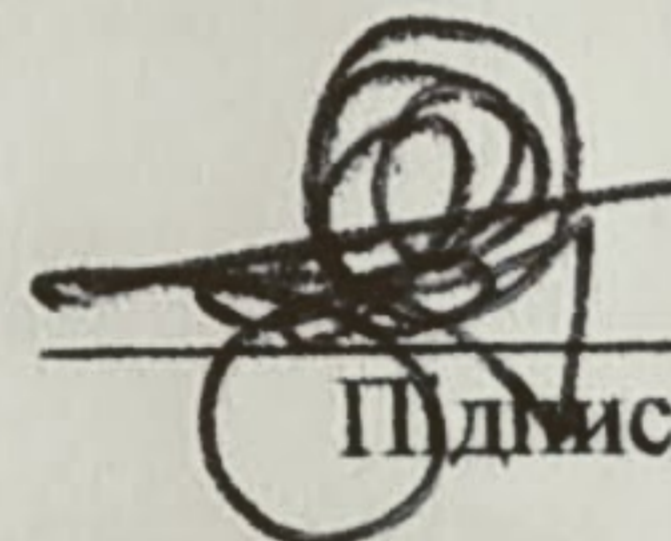
на тему Система оптимального підбору товарів

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

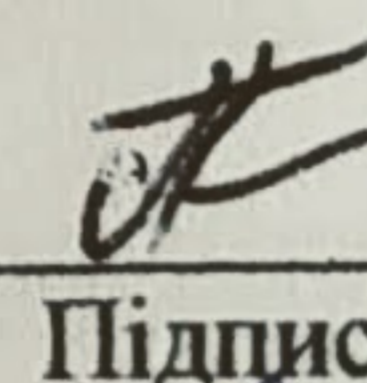
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

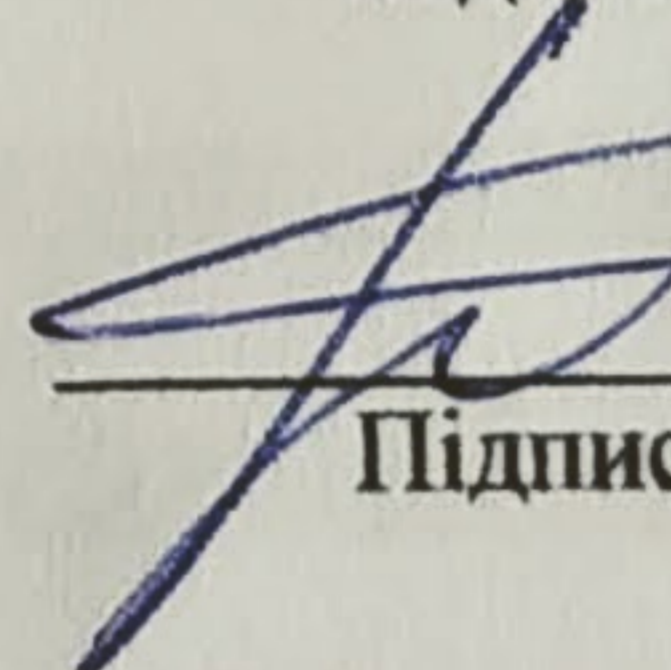
І.С. Яцунь
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада


Підпис

С.С. Петровський
Ініціали, прізвище

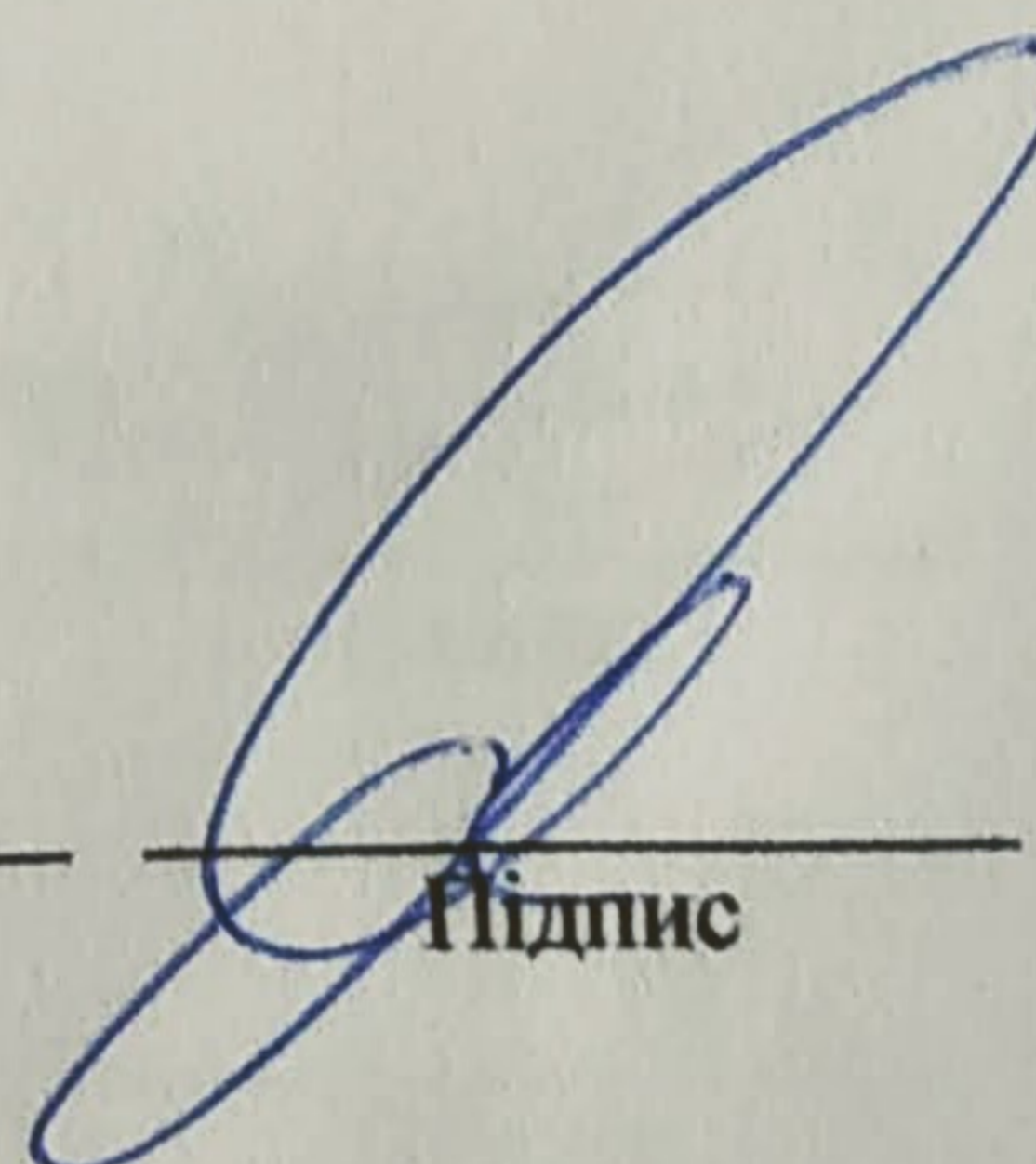
Нормоконтроль: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада


Підпис

Р.О. Багрії
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор


Підпис

О.В. Бармак
Ініціали, прізвище

08 червня 2021 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

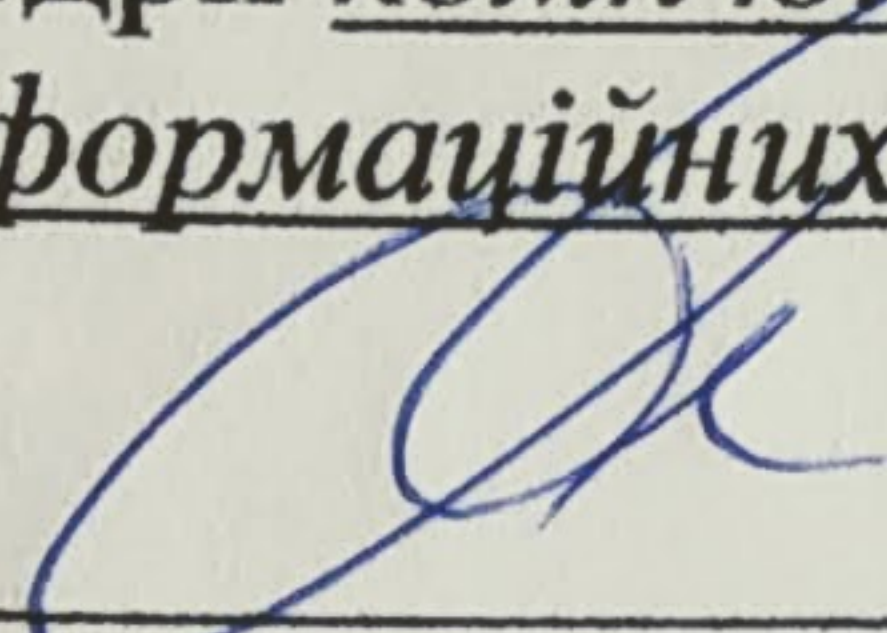
Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій



(підпис)

д.т.н., професор О.В. Бармак

« 08 » лютого 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Система оптимального підбору товарів»

2. Завдання видано студентці Яцунь Івану Сергійовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КНІТ Петровський Сергій Степанович
(посада, прізвище, ім'я, по батькові)

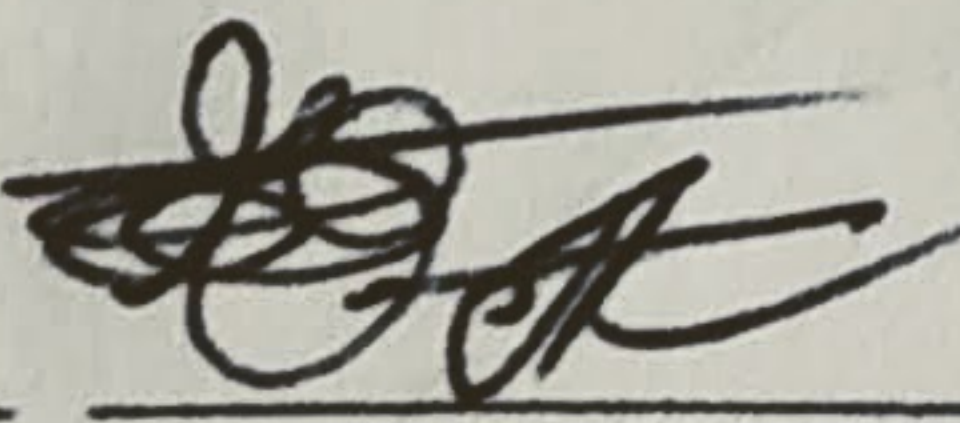
4. Затверджено наказом університету від « 05 » лютого 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – створення інформаційної системи для підбору оптимальних комп'ютерних комплектуючих по заданим бажаним характеристикам. При підборі комплектуючих слід враховувати мінімальні та рекомендовані характеристики збірок що необхідні певною цільовою програмою. Потрібно реалізувати виконання функцій додавання та редагування всіх необхідних для роботи системи даних, й одержання наступних результатів роботи системи: вивід інформації про збірки комплектуючих, цільові програми, та підібрана цільова збірка по вибраній цільовій програмі.

Виконавець: студент 4 курсу, група КН-17-2

Курс, група виконавця



Підпис

І.С. Яцунь

Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ

Науковий ступінь, посада


Підпис

С.С. Петровський

Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Система оптимального підбору товарів»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-2 Яцунь Іван Сергійович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КНІТ Петровський Сергій Степанович


Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
50	29	4	9	1

Метою кваліфікаційної роботи є створення інформаційної системи для підбору оптимальних комп'ютерних комплектуючих по заданим бажаним характеристикам.

Результатом виконання кваліфікаційної роботи є база даних та інформаційна система. Створена інформаційна система може використовуватися для підбору комп'ютерних комплектуючих або цілих збірок комп'ютера, з функцією оптимального підбору товарів по вказаним критеріям. Для розробки даної системи використовувалась реляційна БД MS SQL Server та платформа ASP.NET.

Виконавець: студент 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

І.С. Яцунь
Ініціали, прізвище

Вступ

Тема моєї роботи: створення інформаційної системи для підбору комплектуючих запчастин комп'ютеру. Інформаційна система для підбору – це не магазин, він лише підбирає і рекомендує товар, але може враховувати ціни з інших магазинів. Підбір комплектуючих має свої специфікації і відмінності від, наприклад, «підбору домашньої техніки». Основна відмінність – абсолютно інші критерії підбору. Кожна запчастина має свій набір характеристик, а також - сумісність з іншими запчастинами, що разом з ціною буде складати критерії для вибору запчастини для рекомендації. Важливим критерієм для вибору комп'ютерної техніки є її можливість використання для запуску і користування конкретним програмним забезпеченням : робочими програмами, іграми, програмним забезпеченням зі спеціальними функціями. Саме тому також важливий підбір комплектуючих через критерій можливості запуску конкретних програм, для яких вказані рекомендовані характеристики техніки.

Тобто ціль кваліфікаційної роботи актуальна, багатьом покупцям комп'ютерних запчастин буде цікаво скористатись рекомендаціями системи, що і дало мені ідею вибрати саме цю тему для кваліфікаційної роботи.

Розділ 1. Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

В залежності від задач, що ставляться перед комп'ютером, різні конфігурації архітектури та запчастин будуть показувати різні результати. Через цей факт, та через вартість комп'ютерних запчастин доцільно було б підійти до обрання комп'ютера з розумом. Цю задачу можна назвати Оптимізацією підбору збірки комп'ютера.

Апаратне забезпечення [6] — це електронні частини пристрою для обчислення (можливо механічні), з яких складається система (також мережа). Апаратним забезпеченням можна вважати: пристрої з функціоналом виводу будь-якої інформації та її вводу, різних електронні прилади з електронними схемами, як електричні так і аналогові, пристрої живлення, апаратура для проведення діагностичних робіт, аналізу та тестування, та різноманітні допоміжні компоненти.

Більшість запчастин у складі системи вміщені у корпус комп'ютера, що допомагає механічно зафіксувати і захистити їх при переміщенні, так і від електростатичних розрядів. Важлива також і структура корпусу, так як вона змінює переміщення повітря всередині нього, що необхідно для охолодження компонентів у випадку нагрівання. Також корпус впливає на захист від електромагнітного випромінення, генерованого електричними схемами. У різних корпусах може бути різна кількість простору для різних запчастин, наприклад у великих корпусах є вистачає простору для додаткових твердих дисків і іншої додаткової периферії, що може бути необхідним. Корпус комп'ютера часто змінюється під естетичні цілі користувача, або під особливі потреби.

Часто у корпусі комп'ютера знаходиться блок живлення, що перетворює змінний струм із місцевої мережі на необхідну комп'ютеру напругу. Від вбудованих же акумуляторів працюють ноутбуки, а блок живлення використовується як пристрій для зарядки.

Системна плата — одна з найважливіших частин комп'ютера, так як вона об'єднує в одне ціле всі основні запчастини, а саме: центральний процесор, оперативну пам'ять та жорсткий диск, підключення засобів вводу-виводу. Розглянемо детальніше компоненти, об'єднані системною платою:

1. Центральний процесор – електронний компонент, за допомогою якого здійснюється більшість основних обчислень та на якому зав'язана здатність до роботи комп'ютера, та виконуваних на ньому програм. В процесі роботи процесор зазвичай швидко нагрівається, і через це необхідно охолодження радіатором з допомогою одного або декількох вентиляторів; також використовують і системи охолодження з використанням рідин. З часом процесори комп'ютера почали вміщувати також і вбудований графічний співпроцесор. Тактова частота процесора – це величина, що позначає швидкість виконання процесів, і позначається як правило у мегагерцах або гігагерцах. Досить часто сучасні процесори можна також «розігнати», що понаднормово збільшить частоту», але також і збільшити при цьому тепловиділення відповідно. Також це називають «оверклокінгом».

2. Мікросхеми підтримки, або чіпсет (англ. chipset), складається з північного та південного моста. Головний функціонал чіпсета: забезпечення зв'язку між іншими запчастинами системної плати, такими як: процесором, оперативною пам'яттю, пристроями вводу-виводу, та шинами розширення.

3. Оперативна пам'ять (RAM - англ. Random Access Memory), це елемент куди поміщуються код і необхідні для швидкого доступу дані виконуваних програм. Оперативна пам'ять виготовляється за стандартом DIMM, що установлює стандартні об'єми пам'яті модулів в 2, 4, 8, і 16 гігабайт.

4. Мікросхеми постійної пам'яті (ROM - англ. Read Only Memory), це деталь в якій знаходиться *firmware*, або базова система вводу - виводу — це основна системна програма, що виконується, коли на комп'ютер включається або він перезапускається. Такий процес називається *bootstrapping*[en], і являє собою початковий запуск системи і підготовку до початку роботи.

5. Шини розширення, що використовуються для під'єднання зі слотами плати, що реалізують додаткову функціональність.

6. Гальванічний елемент для живлення напівпостійної пам'яті BIOS.

7. Мікросхеми контролера відео, якщо системна плата має під'єднаний монітор через відеовихід.

8. Інші контролери і інтерфейси вводу-виводу.

9. Карта розширення або плата розширення — це плата зі стандартизованим роз'ємом, що вставляється у слот на системній платі для розширення функціональності комп'ютера. Наприклад, таким способом можна додати до системи звукову плату, вставивши її у слот розширення.

10. Пристрої для накопичення та збереження інформації:

- Жорсткий диск — пристрій для збереження інформації по принципу магнітного запису.
- Твердотільний накопичувач (SSD) — запам'ятовуючий пристрій на основі мікросхем пам'яті, являється швидшим ніж жорсткий диск, і тому вважається кращим варіантом, хоча вартість на одиницю пам'яті більша.
- Оптичний накопичувач — пристрій для зчитування інформації з оптичних носіїв, також можливий і запис інформації (CD-ROM або DVD-ROM). [6]

Для оптимізації підбору збірки комп'ютера, було вирішено відбирати підходящі збірки шляхом зрівняння характеристик збірки з мінімальними та рекомендованими необхідними характеристиками цільової програми. Цими характеристиками являються:

- Частота роботи процесора;
- Кількість ядер процесора;
- Об'єм відео пам'яті відео карти;
- Частота роботи пам'яті відео карти;
- Об'єм оперативної пам'яті;

– Об'єм вільної пам'яті на диску

Враховуючи актуальність даного напрямку, в якості теми для кваліфікаційної роботи була прийнята тема «Система оптимального підбору товарів» та мова програмування C# з фреймворком ASP.NET, в якості базової платформи для реалізації.

Отже основними параметрами предметної області для задачі, що розглядається є:

Програмний продукт	Параметрами є: назва, зображення , розробник, мінімальні характеристики, рекомендовані характеристики, категорія
Збірка комп'ютера	Параметрами є: ціна(приблизна), процесор, оперативна пам'ять, Жорсткий диск, Відеокарта
Користувач	Параметрами є: логін, пароль, роль

1.2 Аналіз інформаційного забезпечення предметної області

1.2.1 Аналіз існуючого програмного забезпечення предметної області

E-Katalog - каталог описів і цін на різноманітну побутову і комп'ютерну техніку, електроніку, товари для дому та офісу. За допомогою цього сервісу можна підібрати і купити необхідну модель за найкращою ціною в порівнянню з різними інтернет-магазинами. У цьому сервісі можна знайти багато інформації, що пригодиться у виборі товарів, параметри для підбору, детальні описи товарів, пошук товару за різними критеріями, відгуки користувачів, рейтинг товарів, рекомендації експертів, каталог брендів і багато інших. [2]

На E-Katalog є функція порівняння товарів по характеристикам, Що схожа на функцію моєї інформаційної системи для підбору комплектуючих запчастин комп'ютеру. На рисунку 1.1 Можна побачити як виглядає зрівняння декількох процесорів.

Сравнение Процессоров [Сохранить таблицу](#)

✕ Удалить
✕ Удалить
✕ Удалить
✕ Удалить





	Intel Core i3 Comet Lake [i3-10100 BOX]	Intel Core i5 Comet Lake [i5-10400 BOX]	AMD Ryzen 3 Matisse [3100 BOX]	AMD A-Series Bristol Ridge [A8-9600 OEM]
<p>Добавить модель для сравнения</p> <p>Бренд <input type="text"/></p> <p>Серия <input type="text"/></p> <p>Модель <input type="text"/></p> <p><input type="checkbox"/> показать устаревшие</p> <p>Сортировка по цене</p> <p>только отличающиеся все параметры</p>	 <p>от 3 499 до 4 679 грн. Сравнить цены 28 →</p>	 <p>от 4 999 до 7 100 грн. Сравнить цены 33 →</p>	 <p>от 3 249 до 3 954 грн. Сравнить цены 12 →</p>	 <p>от 1 459 до 1 959 грн. Сравнить цены 13 →</p>
	☹️ 😊 😊 😊 😊 😊 😊			
Комплектуется кулером ?	+	+	+	-
Серия ?	Core i3	Core i5	Ryzen 3	A-Series
Кодовое название ?	Comet Lake	Comet Lake	Matisse (Zen 2)	Bristol Ridge
Разъем (Socket) ?	Intel LGA 1200	Intel LGA 1200	AMD AM4	AMD AM4
Кол-во ядер ?	4 cores	6 cores	4 cores	4 cores
Кол-во потоков ?	8 threads	12 threads	8 threads	4 threads
Hyper-threading ?	+	+	-	-
SMT (многопоточность) ?	-	-	+	-
Тактовая частота ?	3.6 ГГц	2.9 ГГц	3.6 ГГц	3.1 ГГц
Частота TurboBoost / TurboCore ?	4.2 ГГц	4.2 ГГц	3.9 ГГц	3.4 ГГц

Рисунок 1.1 – Зрівняння процесорів на «E-Katalog»

Hotline – ще один сервіс схожий на E-Katalog. Hotline.ua це інтернет - сервіс, що допомагає покупцеві знайти, порівняти і визначитися з вибором товару, магазину і зробити покупку. Цей сервіс рекламує наявні у різноманітних магазинах товари. Хотлайн це популярний сервіс, який регулярно відвідує понад 250 тис. користувачів. Користуючись сервісами Хотлайн можна значно спростити пошук, вибір та порівняння, що значно допоможе з покупкою і економією.[1] Система порівняння товарів на Hotline зображена на рисунку 1.2.

🏠 / Компьютеры, Сети / Процессоры / Сравнение

Сравнение 4 товаров

+ добавить другие товары
🗑️ очистить выбор





	 Intel Core i3-10100F (BX8070110100F) 58 предложений 2 627 - 3 569 грн 2 986,36 грн	 AMD Ryzen 5 2600 (YD2600BBAFBOX) 27 предложений 4 301 - 6 867 грн 4 970,13 грн	 Intel Core i5-10400F (BX8070110400F) 66 предложений 4 656 - 5 927 грн 5 168,27 грн	 AMD Ryzen 9 5900X (100-100000061WOF) 43 предложения 18 331 - 23 855 грн 19 419,98 грн
Дата добавления товара	12 октября 2020 г.	16 апреля 2018 г.	20 мая 2020 г.	9 октября 2020 г.
Общая оценка	☆☆☆☆ 4	☆☆☆☆ 24	☆☆☆☆ 4	—
Тип	Процессор	Процессор	Процессор	Процессор
Модель	Intel Core i3-10100F	AMD Ryzen 5 2600	Intel Core i5-10400F	AMD Ryzen 9 5900X
Тип разъема	Socket 1200	Socket AM4	Socket 1200	Socket AM4
Частоты				
Тактовая частота, ГГц	3,6 (Turbo 4,3)	3,4 (Turbo 3,9)	2,9 (Turbo 4,3)	3,7 (4,8 Turbo)
Частота системной шины	8 ГГц	нет данных	8 ГГц	—
Кэш-память				
Объем кэш-памяти третьего уровня, МБ	6	16	12	64

Рисунок 1.2 – Зрівняння процесорів на Hotline»

Е-Katalog

Преваги:

- Зручний сервіс
- Зрозумілий у користуванні
- Великий каталог товарів, магазинів
- Краща функція зрівняння товарів

Недоліки:

- Застарілий дизайн
- Багато магазинів з сумнівною репутацією, через що можливе шахрайство
- Можлива обманна ціна (спеціальне зниження ціни на сервісі, коли справжня ціна вища)

Hotline

Переваги:

- Зручний фільтр
- Приємний дизайн
- Є система бонусів
- Актуальний моніторинг цін

Недоліки:

- Часті скарги на видалення та редагування негативних коментарів
- Багато магазинів з сумнівною репутацією, через що можливе шахрайство
- Маніпуляція рейтингами магазинів

Данні сайти – сервіси мають на меті допомогти підібрати товар, і магазин для його покупки, але вони надають характеристики, ціни різних магазинів і відгуки, а не підбирають оптимальний товар по запиту користувача. Тому створення алгоритму який буде підбирати оптимальний варіант по характеристикам і ціні враховуючи потреби користувача є актуальним завданням.

1.2.2 Аналіз сучасних засобів створення програмного забезпечення

На даний момент є величезна кількість різноманітних фреймворків, платформ, мов, та різноманітних технологій веб розробки. Можна виділити такі популярні фреймворки для веб розробки[7]:

Мова PHP

- Symphony
- Zend
- Laravel
- Kohana
- Yii

Мова RUBY

- Ruby on Rails (явний лідер)
- Sinatra

Мова Python

- Django
- Plone
- Twisted
- Flask
- Tornado

Мова Java

- Spring MVC
- JSF
- GWT

Мова Kotlin

- Spring
- Ktor

Мова C#

- ASP.NET (ASP.NET MVC) и .Net Framework

Було вирішено розглянути детальніше 3 найбільш відомих мені мов програмування: .NET, та PHP .

Microsoft .NET [4] – технологія програмування, розроблена компанією Microsoft як платформа для створення різноманітних програм, включаючи також і веб-додатки. Ця технологія в великій мірі є розвитком та продовженням принципу та методики початих технологією програмування Java. Важливою ідеєю .NET є об'єднання різноманітних служб, написаних різними мовами. Ця здатність виставляється Microsoft як перевага .NET, хоча технологія Java уже мала такий функціонал. .NET складається з двох важливих частин– середовище виконання (по суті віртуальна машина) та інструментарій розробки.

.NET – являється технологією з наявністю крос-платформи, і на даний момент доступна реалізація для таких платформ як Microsoft Windows, FreeBSD (також від Microsoft), варіант технології для ОС Linux в проекті Mono, DotGNU .

ASP.NET [6] — це фреймворк який являється частиною платформи Microsoft .NET. Це технологія створення веб-додатків, веб застосунків та сервісів від компанії Microsoft. Її можна вважати розвитком більш давньої технології Microsoft ASP.

Розробляти та працювати над проектами .NET-програм можна у різних середовищах проектування: Visual Studio, SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse також має додаток для розробки .NET-програм. Деякі програми можна розробляти навіть в текстовому

редакторі та використовувати консольний компілятор. На платформі .NET в якості бази даних краще за все використовувати MS SQL Server.

ASP.NET будується на принципі Common Language Runtime: розробники мають можливість писати код для ASP.NET, використовуючи велику кількість різноманітних мов програмування, з яких деякі входять в комплект .NET Framework (C#, JavaScript .NET, і Visual Basic .NET), і ще більше інших мов можуть бути установленні додатково (Python, PHP, Perl, Smalltalk, Ruby).

На ASP.NET можлива розробка проектів з реалізацією шаблону Model-View-Controller (ASP.NET MVC Framework).

PHP [5] (англ. PHP: Hypertext Preprocessor – PHP: гіпертекстовий препроцесор) – це скриптова мова програмування, що була розроблена для створення HTML сторінок шляхом генерації із веб сервера. PHP являється одною з найпоширеніших мов програмування для сфери веб-додатків. PHP підтримується переважною більшістю хостинг-провайдерів.

Суть роботи PHP у інтерпретації коду веб-сервером як HTML-код, що після цього вже буде переданий на пристрій користувача. Велика різниця у порівнянні зі мовою JavaScript в тому, що клієнт не отримує PHP-код, а лише готовий html файл що отримує браузер. Даний спосіб передачі даних є великим плюсом зі сторони безпеки, але він також робить важчою інтерактивну роботу веб сторінок. Однак можливо використовувати PHP код для генерування JavaScript файлу, який вже може бути виконаним браузером користувача.

Laravel [6] – це безплатний фреймворк з принципом відкритого коду. Він був створений Тейлором Отвеллом, з ціллю розробки веб-додатків по структурному шаблону MVC. На створення цього фреймворку був виказаний великий вплив інших PHP фреймворків. Створений він був з ціллю бути сильною альтернативою Codeigniter, у якому на думку розробника не було достатньо корисних засобів для ефективної розробки веб сайтів. В своїй основі Laravel має велику кількість іншого популярного фреймворку – Symfony.

Велику популярність даний фреймворк завоював у західних програмістів. Laravel користується менеджером пакетів Composer, за допомогою якого він може легко встановлювати і підключати різноманітні компоненти для використання у веб додатках. Також Laravel реалізовує шаблон ActiveRecord - Eloquent ORM, що дозволяє йому встановити відношення між різноманітними об'єктами бази даних веб додатку та побудувати зручні запити для управління даними різного типу. Механізм авто підвантаження класів дає можливість не підключати файли вручну через include, и зупиняє загрузку невикористаних компонентів. Має зручну систему міграцій, що допомагає опростити оновлення і розвертання веб-додатків.

В Laravel використовується шаблон Blade, який дозволяє об'єднувати html код та код php, за допомогою якого можна робити нескладні представлення вашого веб-додатку використовуючи особливу структуру коду.

1.3 Постановка задачі

Метою кваліфікаційної роботи є розробка інформаційної системи для підбору комплектуючих запчастин комп'ютеру на платформі .NET , що виконує наступні основні функції:

- виведення інформації про комплектуючі та програмне забезпечення;
- адмін-панель для додавання нових одиниць комплектуючих та програмного забезпечення;
- редагування і видалення вже створених даних;
- робота з особистими даними (реєстрація в системі, авторизація);
- виведення оптимальних товарів по заданим критеріям;

Розділ 2. Проектування структури інформаційної системи

2.1 Аналіз та автоматизація обробки інформаційних потоків

При розробці інформаційної системи для підбору комплектуючих запчастин комп'ютеру необхідно реалізувати наступні бізнес-процеси:

- Робота з інформацією про збірки;
- Робота з підбором збірки;
- Робота з вибором програм для підбору;
- Робота з адмін-панеллю;
- Робота з інформацією про програми;
- Робота з інформацією про збірки;

Робота з інформацією про збірки - надає можливість користувачу інформаційної системи отримувати інформацію про доступні на інформаційної системи збірки, а саме:

- Інформацію про існуючі збірки.
- Інформацію про характеристики комплектуючих збірки.

Доступ до цієї інформації відкритий для всіх користувачів
Діаграму дій для даного бізнес-процесу зображено на рисунку 2.1.

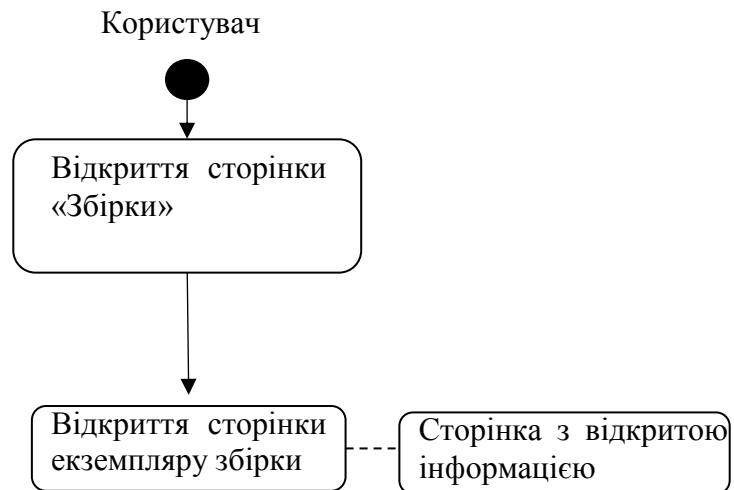


Рисунок 2.1 – Діаграма дій бізнес-процесу «інформацією про збірки».

Робота з підбором збірки - надає можливість користувачу використовувати підбір збірок по вибраним цільовим програмам, а саме:

- вибирати критерії підбору збірки через Мінімальні або Рекомендовані характеристики цільової програми.
- Виконувати підбір збірки по заданим параметрам.

Діаграма дій даного бізнес-процесу зображена на рисунку 2.2.

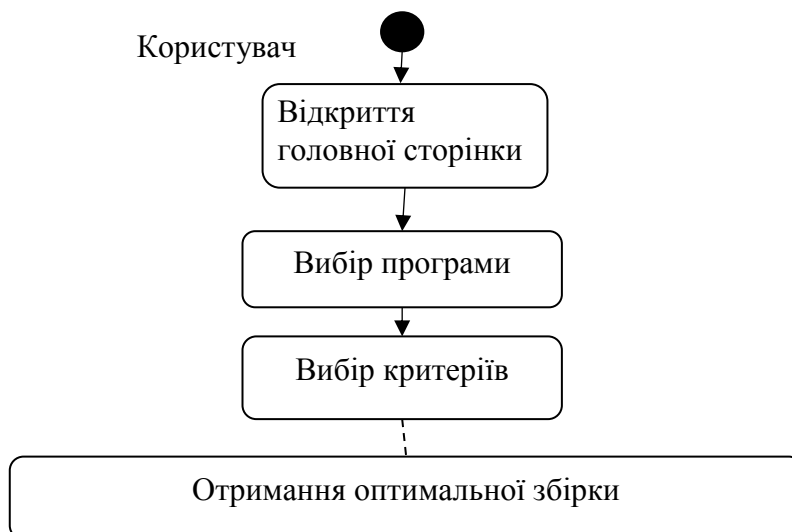


Рисунок 2.2 – Діаграма дій бізнес-процесу «Робота з підбором збірок».

Робота з адмін-панеллю надає можливість працівнику інформаційної системи (адміністратору) додавати та редагувати інформацію про збірки та програми. Він повинен мати можливість:

- Заходити в адмін-панель за допомогою логіну та пароля.
- Переглядати існуючі збірки та програми.
- Додавати нові збірки та програми.
- Редагувати збірки та програми.
- Видаляти збірки та програми.

Діаграма дій даного бізнес-процесу зображена на рисунку 2.3.



Рисунок 2.3 – Діаграма дій бізнес-процесу «Робота з адмін-панеллю».

Робота з вибором програм для підбору - надає можливість користувачу інформаційної системи вибрати цільові програми для підбору збірок.

Алгоритм дій користувача:

- Користувач вибирає підбір товарів по програмі
- З доступних програм користувач вибирає потрібну.

- Користувач бачить рекомендовані характеристики для програми.
- Виконується підбір збірок по програмі.

Робота з інформацією про програми – надає можливість користувачу інформаційної системи отримувати інформацію про програми.

Алгоритм дій користувача:

- Користувач переходить на сторінку «Програми».
- З доступних програм користувач вибирає потрібну.
- Користувач переглядає інформацію про програму.

На початку роботи з системою користувачу виводиться головна сторінка, де він може вибрати цільові програми для підбору збірок. Звідси він може перейти на сторінку «Збірки» та «Програми», де відображається інформація про збірки, та програми відповідно. Також з стартової сторінки адміністратор може зайти в адмін панель.

На рисунку 2.1 зображена UML-діаграма сайту, що зображує описані бізнес-процеси.

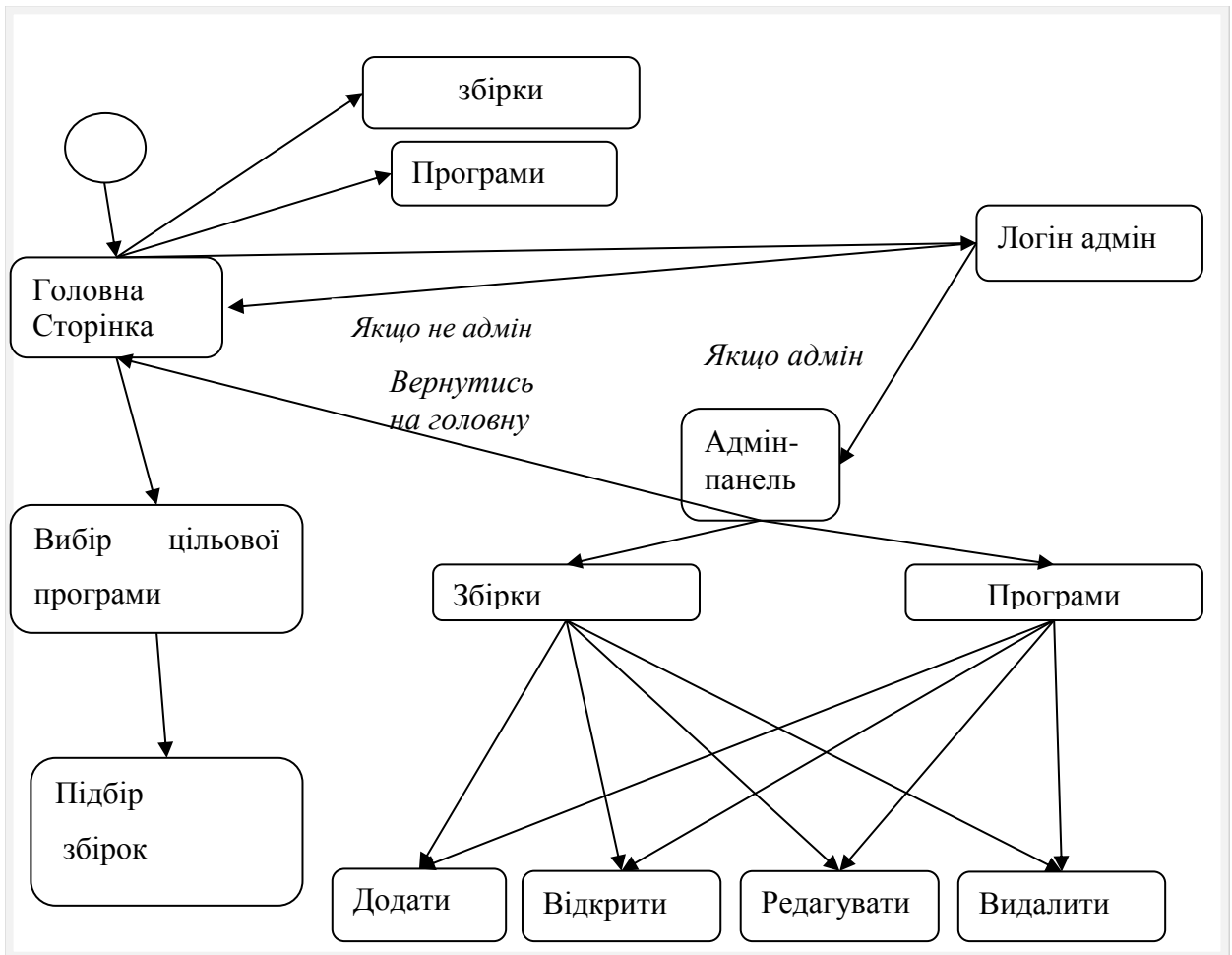


Рисунок 2.4 – UML-діаграма сайту.

Для збереження даних по розглянутих процесах та послідовності дій користувача системи, необхідно використати БД MSSQL.

2.2 Розробка структури інформаційної системи

Користувачі заносяться в таблицю «users» де є логін, пароль, і статус адміна. (таблиця 2.1)

Програми заносяться в таблицю «programs», де є назва, тип, опис, зображення, розробник, мінімальна частота процесора, рекомендована частота процесора, мінімальна кількість ядер, рекомендована кількість ядер, мінімальний об'єм оперативної пам'яті, рекомендований об'єм оперативної пам'яті, мінімальний об'єм постійної пам'яті, рекомендований об'єм постійної

пам'яті, мінімальний об'єм візуальної пам'яті, рекомендований об'єм візуальної пам'яті, мінімальна частота пам'яті відеокарти, рекомендована частота пам'яті відеокарти. (таблиця 2.2) .

Збірки заносяться в таблицю «setups», де є назва, опис, середня ціна, частота процесора, кількість ядер процесора, об'єм оперативної пам'яті, необхідний об'єм диску, Об'єм відеопам'яті, Частота відео пам'яті, Назва процесору, назва оперативної пам'яті, назва Відеокарти. (таблиця 2.3)

Таблиця 2.1 – Атрибути таблиці «User»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	login	varchar	Логін
3	password	varchar	Пароль
4	Role	bool	Чи Адмін

Таблиця 2.2 – Атрибути таблиці «programs»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	Name	varchar	Назва
3	Image	varchar	зображення
4	Description	varchar	Опис
5	Category	varchar	Категорія
6	Developer	varchar	Розробник
7	ProcessorMin	float	мінімальна частота процесора
8	ProcessorRec	float	рекомендована частота процесора
9	ProcessorCoreMin	float	мінімальна кількість ядер
10	ProcessorCoreRec	float	рекомендована кількість ядер

11	RamMin	float	мінімальний об'єм оперативної пам'яті
12	RamRec	float	рекомендований об'єм оперативної пам'яті
13	StorageMin	float	мінімальний об'єм постійної пам'яті
14	StorageRec	float	рекомендований об'єм постійної пам'яті
15	VideocardMin	float	мінімальний об'єм візуальної пам'яті
16	VideocardRec	float	рекомендований об'єм візуальної пам'яті
17	VideocardFreqMin	float	мінімальна частота пам'яті відеокарти
18	VideocardFreqRec	float	рекомендована частота пам'яті відеокарти

Таблиця 2.3 – Атрибути таблиці «Setup»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	Name	varchar	Назва
3	Description	varchar	Опис
4	AvgPrice	float	середня ціна
5	Processor	float	частота процесора
6	ProcessorCore	float	кількість ядер процесора
7	RAM	float	об'єм оперативної пам'яті
7	Storage	float	необхідний об'єм диску
	Videocard	float	Об'єм відеопам'яті
	VideocardFreq	float	Частота відео пам'яті
	NameProcessor	varchar	Назва процесору
	NameRAM	varchar	назва оперативної пам'яті
	NameVideocard	varchar	назва Відеокарти

Отже, зазначений обсяг таблиць є достатнім для обробки всіх необхідних даних для роботи з ігровим інтернет - магазином.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

C# — це мова програмування що повноправно може вважатися об'єктно-орієнтованою, також вона має вбудовану систему безпечної типізації .NET. Мова належить компанії Microsoft.[6]

Синтаксис мови C# є близьким C, C++, але не лише являється розвитком цих мов, також він ввібрав в себе елементи мов Java і Pascal. Так як мова C# є об'єктно-орієнтованою, вона має правило статичної типізації, підтримується поліморфізм, атрибути, перевантаження операторів, події, властивості, винятки, коментарі у форматі XML, вказівники на функції певних класів. Ця мова багато чого перейняла від своїх ідейних батьків, — а саме мов C++, C, Pascal, і Java — C#, але при цьому вирішила багато попередніх проблем, що були у попередників. Наприклад, уже недоступна така можливість множинне успадкування класів, що мала свої проблеми при деяких ситуаціях.[8]

Було обрано мову C# через її зручність, мій досвід в роботі з нею та фреймворк ASP.NET, який використовує цю мову, що я обрав для розробки проекту.

2.3.2 Вибір фреймворку

Для розробки проекту було вирішено обрати ASP.NET, тому що цей фреймворк являється частиною платформи Microsoft .NET, створений для роботи з мовою C#, та підходить до мого проекту.

На ASP.NET можлива розробка проектів з реалізацією шаблону Model-View-Controller (ASP.NET MVC Framework).

2.3.4 Вибір системи керування базами даних

Є величезна кількість різноманітних СКБД для створення систем обліку, наприклад MS SQL Server, MySQL, Oracle, і інші, але для проекту ASP.NET MVC краще всього підходить MS SQL Server.

Microsoft SQL Server – це СКБД від компанії Microsoft. Для функціонування цієї СКБД використовується мова Transact-SQL, що була створена спільно Microsoft та Sybase. Transact-SQL відповідає стандарту ANSI/ISO що визначає структуру SQL запитів та розширень. Microsoft SQL Server може використовуватися для обробки даних будь яких розмірів, і ідеально підходить до проекту ASP.NET MVC .[9]

Відповідно до переваг і недоліків різноманітних СКБД було обрано MS SQL Server, так як вона підходить під платформу .Net.

Розділ 3. Програмна реалізація

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок

Проект використовує фреймворк ASP.NET MVC, який реалізує шаблон проектування MVC. Суть цього шаблону полягає в розділенні програми на моделі, контролери та представлення. У Моделі реалізується бізнес-логіка програми. У представленні знаходиться вивід даних системи. Контролер реалізує зв'язок між моделями та представленнями.

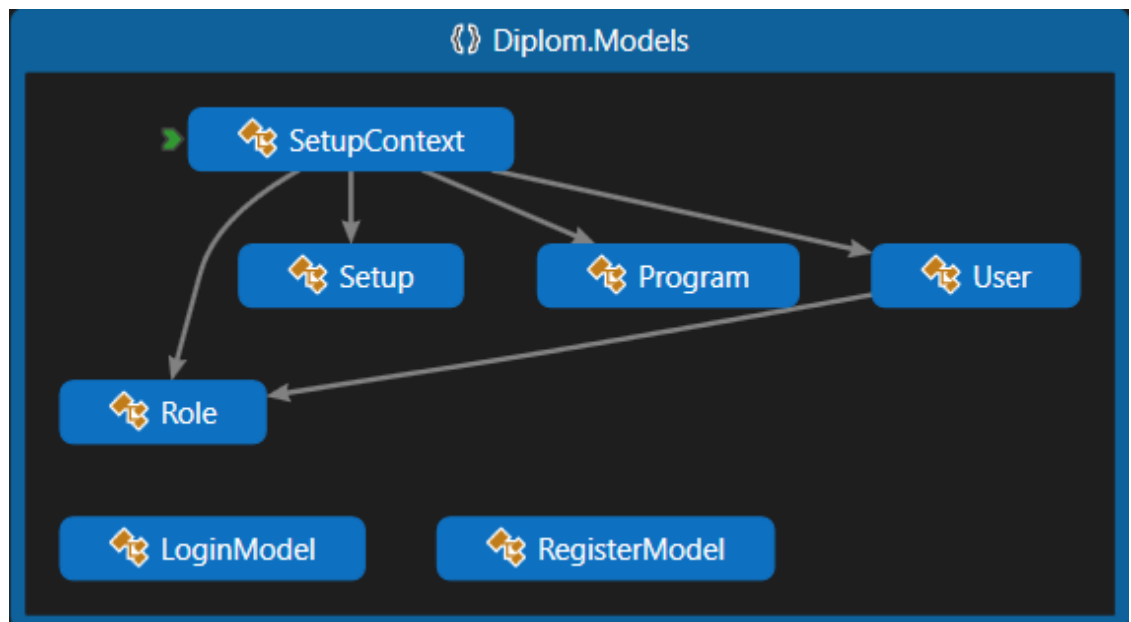


Рисунок 3.1 – Моделі проекту

Даний проект має наступні моделі (рисунок 3.1):

- AuthModels – модель, яка відповідає за бізнес логіку пов'язану з роботою системи авторизації. Включає в себе моделі LoginModel та RegisterModel;
- LoginModel – ця модель описує структуру класу, що використовується для виконання входу на сайт по зареєстрованому акаунту(рисунок 3.2);

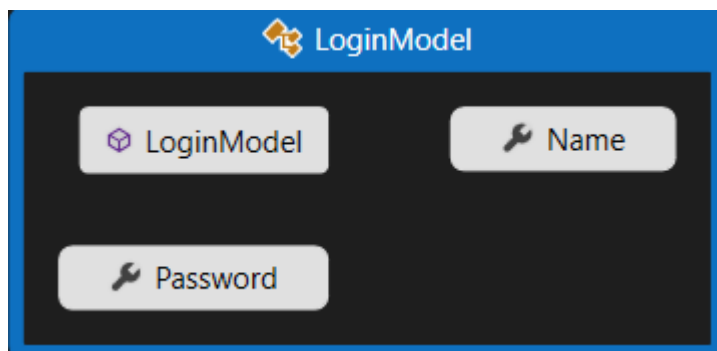


Рисунок 3.2 – LoginModel

- RegisterModel – модель, що описує структуру, що використовується для виконання реєстрації на сайт (рисунок 3.3);

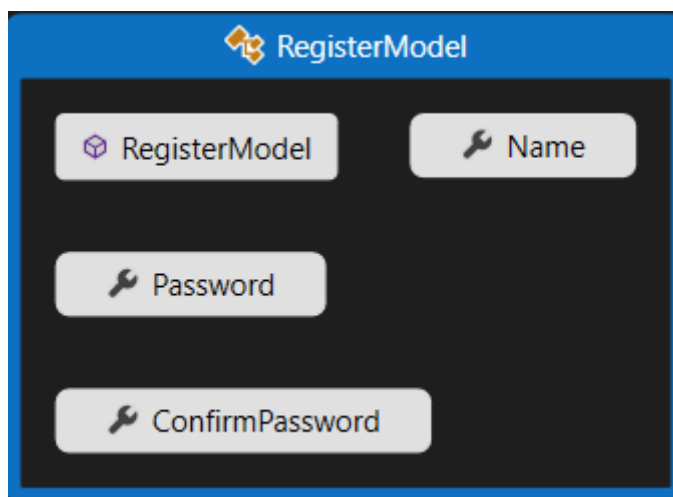


Рисунок 3.3 – RegisterModel

- Program – в цій моделі описується логіка і структура класу Program, І відповідає за бізнес логіку пов'язану з цільовими програмами для

підбору(рисунок 3.4);

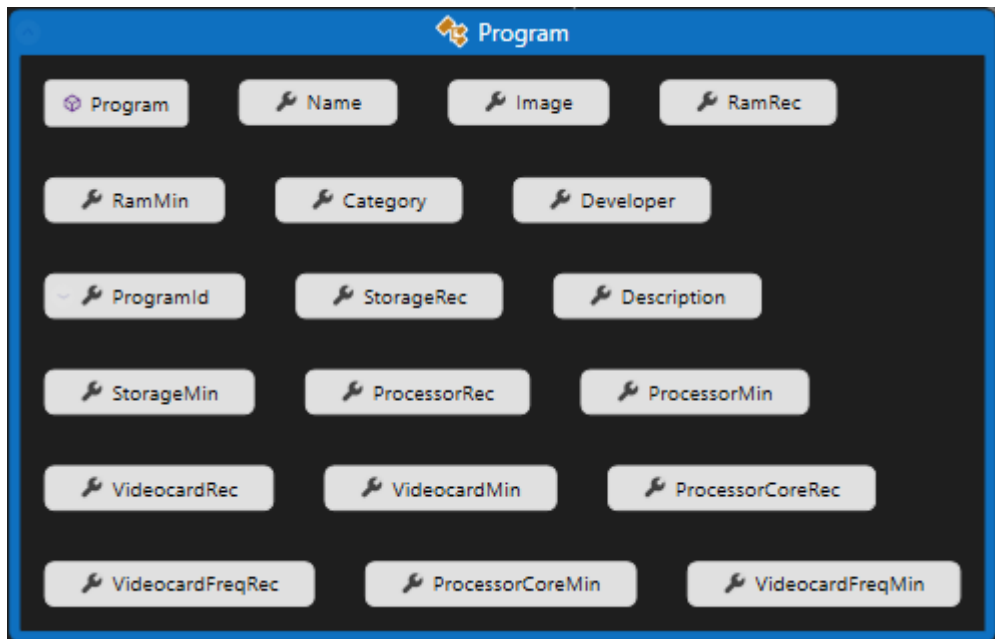


Рисунок 3.4 – Program

- User – є моделлю, відповідаючою за бізнес логіку, пов'язану з функціоналом користувачів (рисунок 3.5);

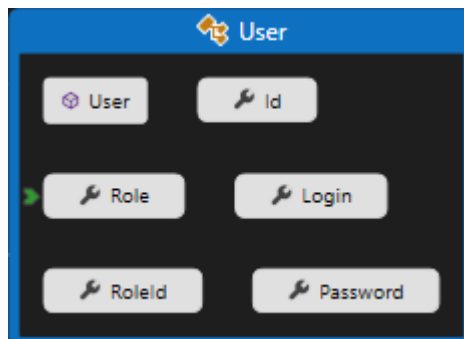


Рисунок 3.5 – User

- Setup – є моделлю, відповідаючою за структуру елементів збірки комп'ютерів та бізнес логіку пов'язану з ними (рисунок 3.6);

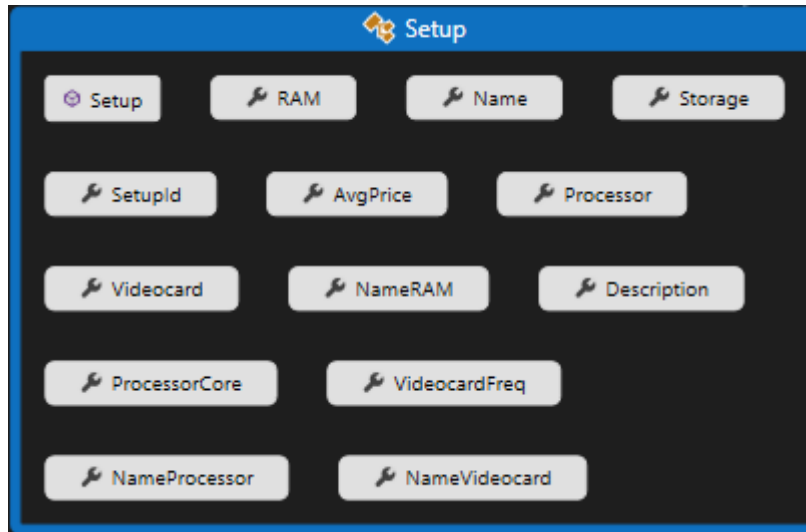


Рисунок 3.6 – Setup

- SetupContext – є моделлю – контекстом даних для зв'язування інших моделей проекту з базою даних (рисунок 3.7);

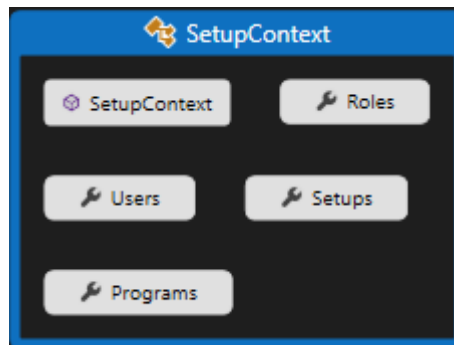


Рисунок 3.7 – Setup

В проекті є наступні представлення:

Account – група представлень, що відповідають за управління акаунтом:

- register - Відповідає за реєстрацію нового користувача, введення необхідних даних;
- login - Відповідає за вхід на сайт під зареєстрованим акаунтом;

Admin - група представлень, що відповідають за функціонування панелі Адміністратора:

- AddProgram – Відповідає за створення нового екземпляру класу Program, з введенням всіх необхідних даних;
- AddSetup – Відповідає за створення нового екземпляру класу Setup, та введенням всіх необхідних даних;
- DeleteProgram - Відповідає за видалення об'єкту програми;
- DeleteSetup - Відповідає за видалення об'єкту Збірки;
- DetailsPrograms - Відповідає за відображення усіх параметрів об'єкту програми;
- DetailsSetups – Відповідає за відображення усіх параметрів об'єкту збірки;
- EditProgram – Відповідає за редагування об'єкту класу програми, відображає попередні параметри об'єкту;
- EditSetup – Відповідає за редагування об'єкту класу збірки, відображає попередні параметри об'єкту;
- Index – Відповідає за відображення головної сторінки адмін-панелі
- Programs – Відповідає за відображення всіх елементів класу програми в адмін-панелі;
- Setups – Відповідає за відображення всіх елементів класу збірки в адмін-панелі;

Home - група представлень, що відповідають за функціонування основної функціональності інформаційної системи:

- DetailsPrograms – Відповідає за відображення параметрів об'єкту програми, що доступні користувачу;
- DetailsSetups – Відповідає за відображення параметрів об'єкту збірки, що доступні користувачу;

- Find – Відповідає за відображення результату підбору збірки;
- Index – Відповідає за відображення головної сторінки сайту, з формою для проведення знаходження оптимальної збірки під обрану цільову програму;
- Programs – Відповідає за відображення всіх елементів класу Program за допомогою карток;
- Setups – Відповідає за відображення всіх елементів класу Setup за допомогою карток;
- UserLogin – Відповідає за відображення панелі із входом/реєстрацією та відображенням поточного користувача;
- Layout – майстер – сторінка, навігаційна сторінка, використовується всіма іншими сторінками;

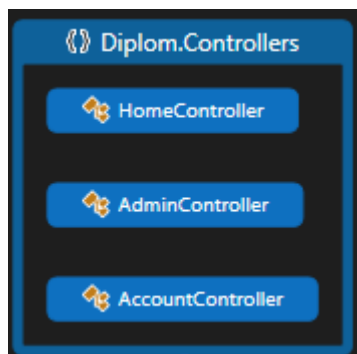


Рисунок 3.8 – Контролери

Також у системі є наступні контролери(рисунок 3.8):

- HomeController – Відповідає за основні функції інформаційної системи, вивід інформації про Програми та Збірки, та підбір оптимальної збірки під обрану цільову програму (рисунок 3.9);

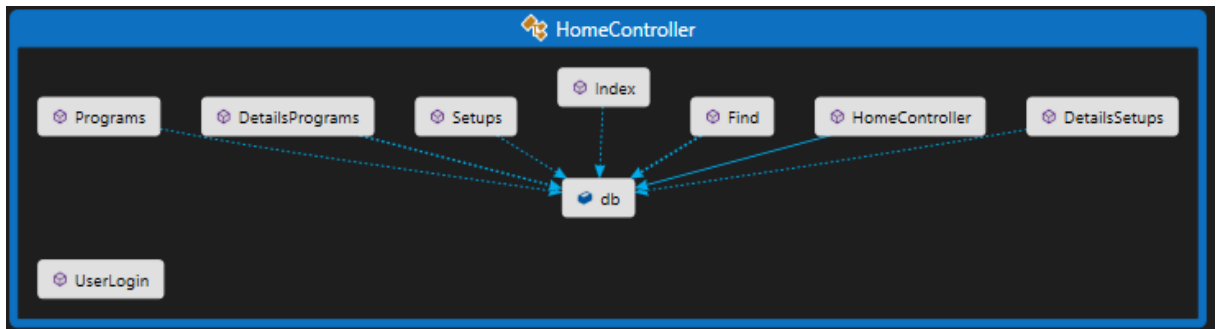


Рисунок 3.9 – методи контролеру HomeController

- AdminController – Відповідає за роботу функціоналу адміністратора: занесення в базу даних нових даних про програми та збірки, їх зміна та видалення (рисунок 3.10);

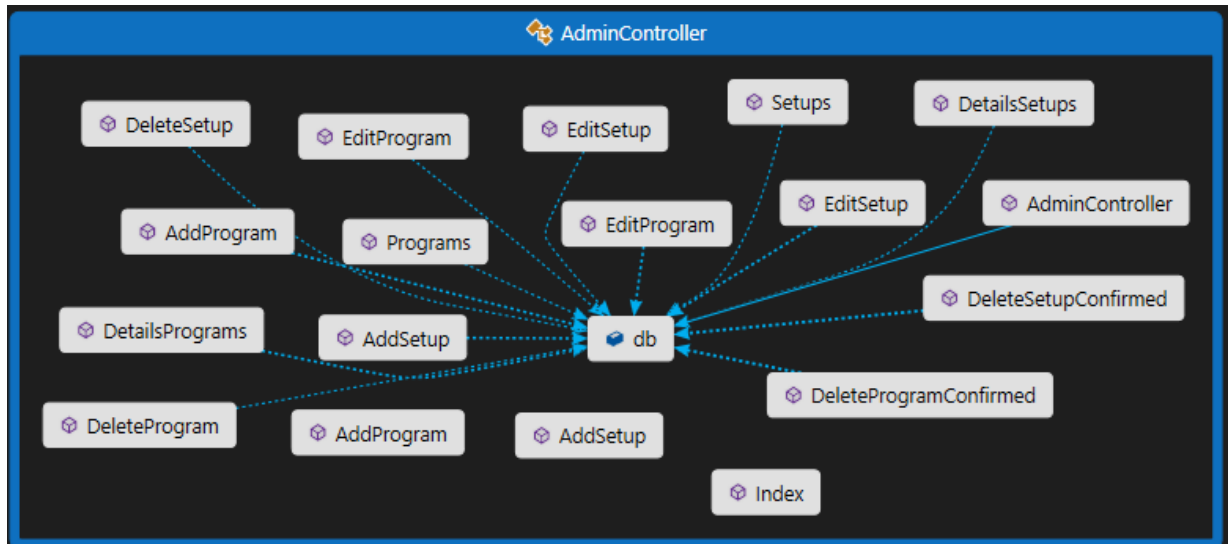


Рисунок 3.10 – методи контролеру AdminController

- AccountController – Відповідає за функціонування системи авторизації, створення акаунту та входу в акаунт (рисунок 3.11);

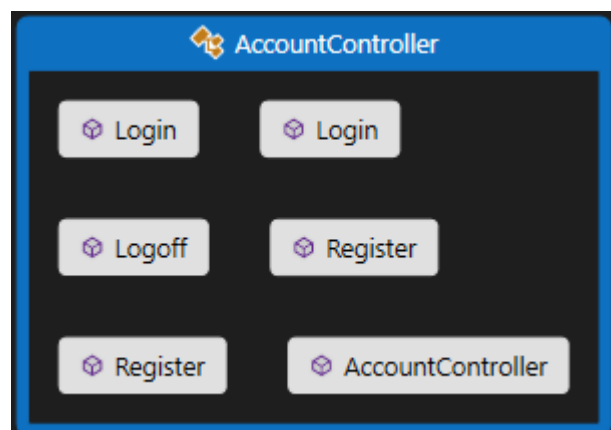
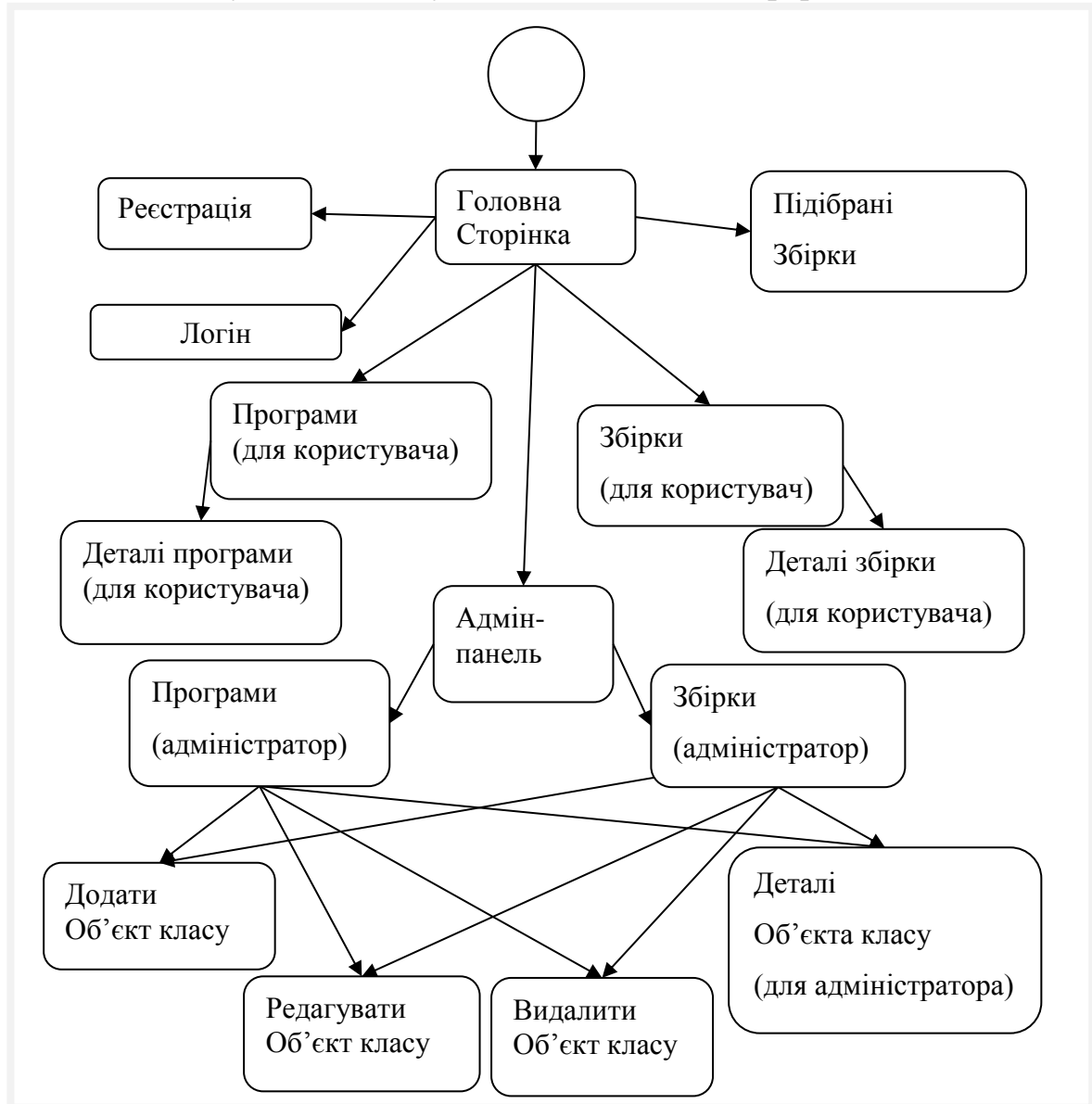


Рисунок 3.11 – методи контролеру AccountController

3.2 Розробка програмних модулів

На рисунку 3.12 зображена Функціональна схема розробленої інформаційної системи.

Рисунок 3.12 – Функціональна схема інформаційної системи



Діаграма класів контролерів на рисунку 3.13. Методи цих контролерів відповідають за функціонування відповідних View сторінок.

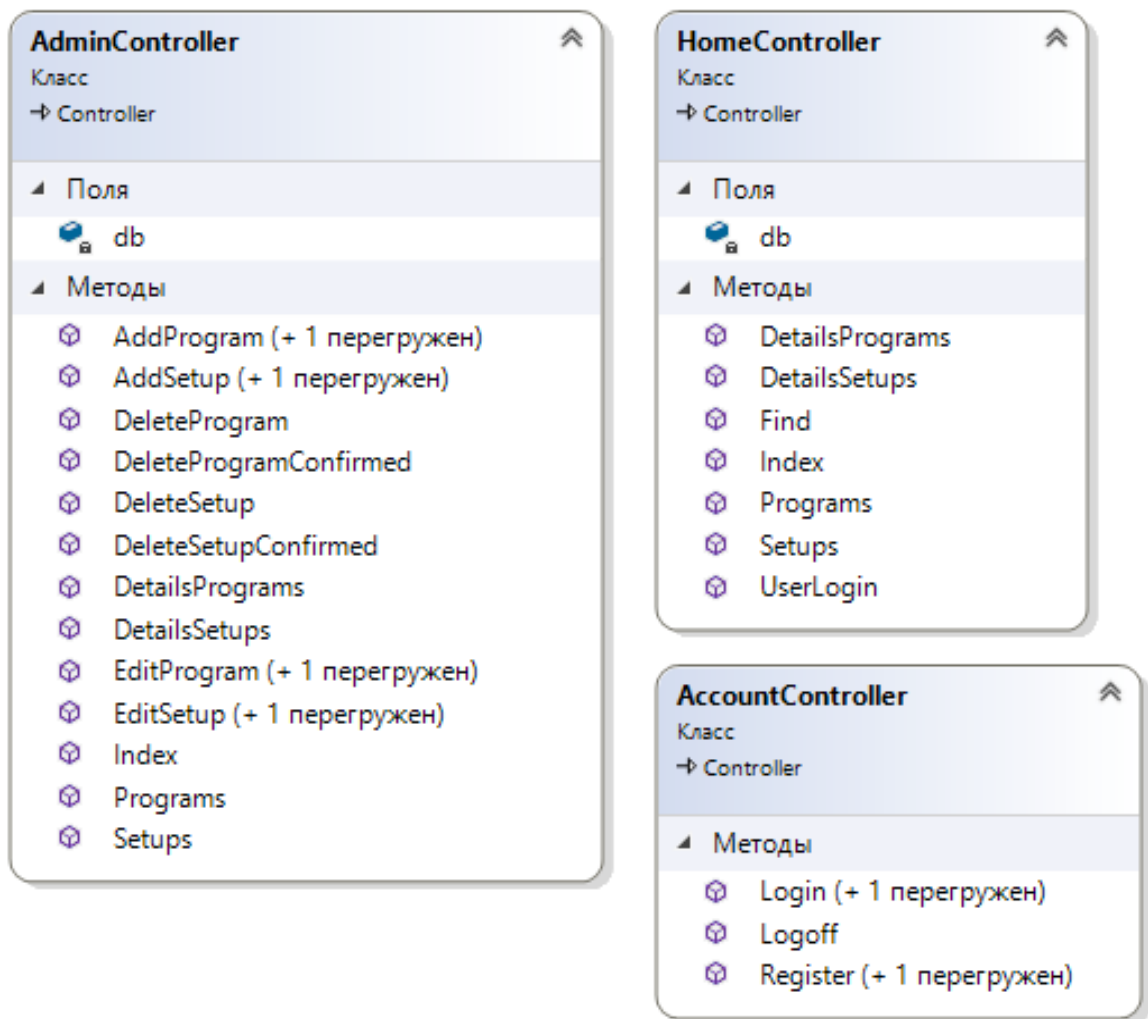


Рисунок 3.13 – Діаграма класів контролерів

Основний функціонал проекту реалізований в класі HomeController – А саме в методах Index та Find. Метод Index реалізує роботу головної сторінки та передає данні про доступні цільові програми. Він дає запит бази даних на отримання списку всіх елементів Program, та передає їх у представлення через ViewBag. Сам підбір збірок реалізує метод Find, що бере список збірок з бази та в залежності від обраного типу підбору (Мінімальний чи рекомендований) відбирає підходящі збірки. З підходящих збірок формується список, який сортується по ціні.

Таким чином клієнту видається список збірок, який задовольняє вибрані критерії, починаючи з самого дешевого варіанту. Клієнт має можливість обрати або самий дешевий підходящий варіант, або переглянути кращі варіанти залежно від бюджету. Робота методу Find зображена на рисунку 3.14. Фрагмент коду із алгоритмом:

```

if(type.Equals("Minimal"))
{
    foreach (Setup set in setups)
    {
        if(set.Processor >= a.ProcessorMin && set.ProcessorCore >=
            a.ProcessorCoreMin && set.RAM >= a.RamMin
            && set.Videocard >= a.VideocardMin && set.VideocardFreq >=
            a.VideocardFreqMin)
        {
            SortedSetups.Add(set);
        }
    }
}
if(type.Equals("Recomended"))
{
    foreach (Setup set in setups)
    {
        if (set.Processor >= a.ProcessorRec && set.ProcessorCore >=
            a.ProcessorCoreRec && set.RAM >= a.RamRec
            && set.Videocard >= a.VideocardRec && set.VideocardFreq >=
            a.VideocardFreqRec)
        {
            SortedSetups.Add(set);
        }
    }
}
SortedSetups = SortedSetups.OrderBy(x => x.AvgPrice).ToList();

```

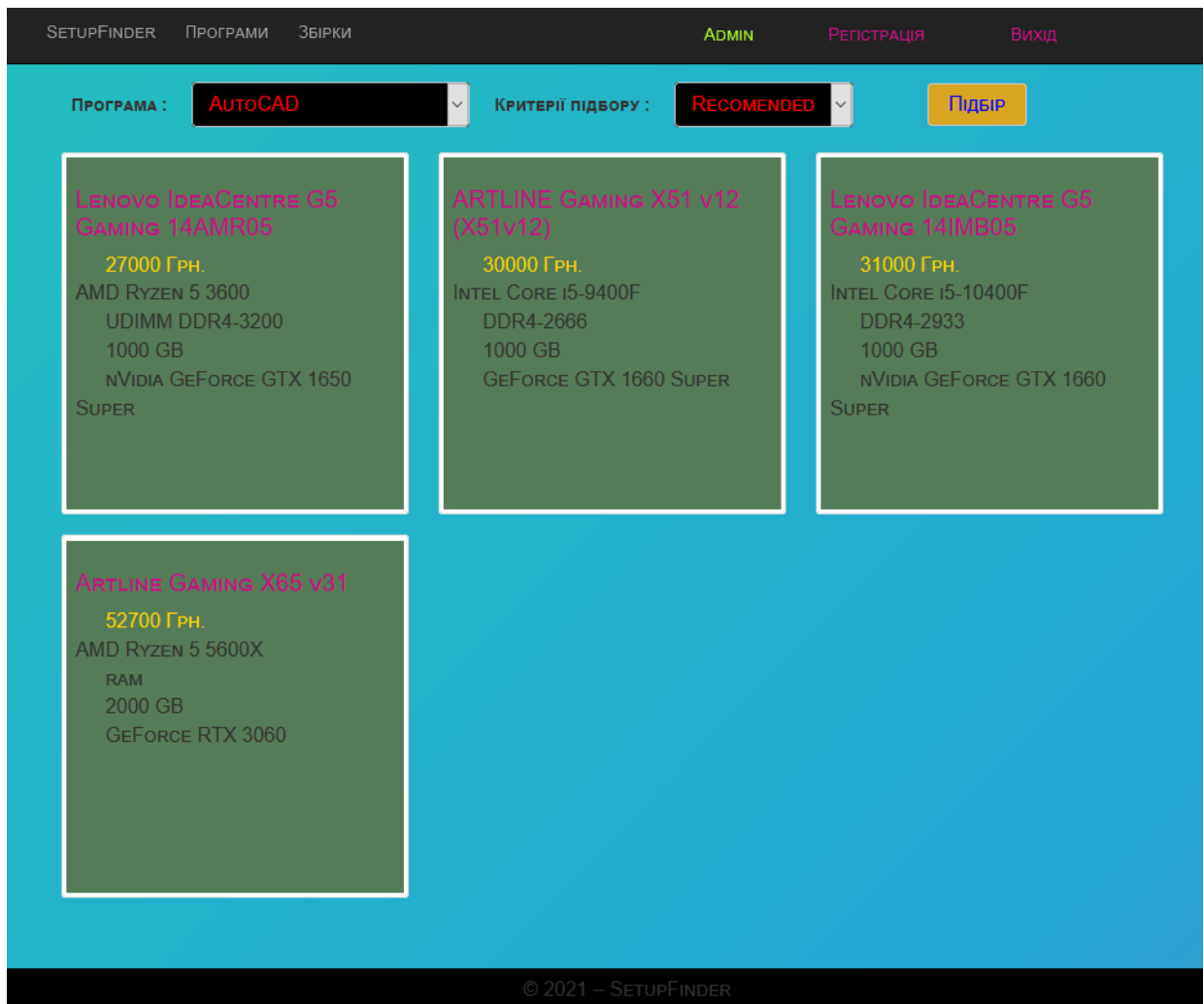


Рисунок 3.14 – робота методу Find

AccountController – Відповідає за функціонування системи авторизації, створення акаунту та входу в акаунт. Цей функціонал реалізований в методах Login, Register та Logoff. Інформація про користувача передається в базу, і за замовчанням задає йому роль користувача. Роль адміністратора може задаватись лише напряму в базу даних, для безпеки. Робота методу Register зображена на рисунку 3.15.

The screenshot shows a web application interface for registration. At the top, there is a dark navigation bar with the following items: 'SETUPFINDER', 'ПРОГРАМИ', 'ЗБІРКИ', 'ADMIN', 'РЕГІСТРАЦІЯ', and 'ВИХІД'. Below this, the main content area has a blue background. The title 'РЕЄСТРАЦІЯ' is displayed in large white letters, with 'REGISTERMODEL' underneath it. The registration form consists of three white input fields labeled 'NAME', 'PASSWORD', and 'CONFIRMPASSWORD'. Below these fields is a white button with the text 'ЗАРЕЄСТРУВАТИСЯ'. At the bottom left of the form area, there is a blue link labeled 'НАЗАД'.

Рисунок 3.15 – робота методу Register

`AdminController` – Відповідає за роботу функціоналу адміністратора: занесення в базу даних нових даних про програми та збірки, їх зміна та видалення. У цьому контролері є велика кількість різноманітних методів, що реалізують роботу необхідних операцій, так відображення існуючих програм та збірок реалізовані методами `Programs` та `Setups` відповідно, додавання нових елементів реалізовані методами `AddProgram`, `AddSetups`, редагуються існуючі елементи методами `Edit`, видаляються в методах `Delete`.

Деталі зі всією інформацією про елементи моделі можна отримати з допомогою методу `Details`.

Роботу методу `Programs` можна побачити на рисунку 3.16

NAME	DESCRIPTION	CATEGORY	DEVELOPER	
CYBERPUNK 2077	SCAM	RPG GAME	CD PROJECT RED	EDIT DETAILS DELETE
ZOOM	ZOOM	SKYPE	ZOOM	EDIT DETAILS DELETE
BLENDER	3D МОДЕЛИРОВАНИЕ	3D МОДЕЛИРОВАНИЕ	BLENDER FOUNDATION	EDIT DETAILS DELETE
SKETCHUP	SKETCHUP	3D МОДЕЛИРОВАНИЕ	SKETCHUP FOUNDATION	EDIT DETAILS DELETE
SOLIDWORKS	SOLIDWORKS	3D МОДЕЛИРОВАНИЕ	SOLIDWORKS DEV	EDIT DETAILS DELETE
AUTOCAD	AUTOCAD	3D МОДЕЛИРОВАНИЕ	AUTODEK	EDIT DETAILS DELETE
INVENTOR	INVENTOR	3D МОДЕЛИРОВАНИЕ	AUTODESK	EDIT DETAILS DELETE
TINKERCAD	TINKERCAD	3D МОДЕЛИРОВАНИЕ	TINKERCAD	EDIT DETAILS DELETE
DOTA 2	DOTA	MOBA GAME	VALVE	EDIT DETAILS DELETE
DARK SOULS 3	YOU DIED	RPG GAME	FROMSOFTWARE	EDIT DETAILS DELETE
DARK SOULS 2	DARK SOULS 2	RPG GAME	FROMSOFTWARE	EDIT DETAILS DELETE
MICROSOFT FLIGHT SIMULATOR	MICROSOFT FLIGHT SIMULATOR	SIMULATOR	MICROSOFT	EDIT DETAILS DELETE

© 2021 – SETUPFINDER

Рисунок 3.16 – робота методу Programs

3.3 Тестування інформаційної системи

Проведемо тестування різноманітних можливостей і функцій інформаційної системи, з допомогою тест-кейсів на таблицях.

Перевірка працездатності інтерфейсу користувача на головній сторінці.

Author:	Specification ID:	Priority:	Producer:	Developer:
Яцунь І.С	1	1	S.Petrovsky	Яцунь І.С
OVERVIEW:				
Даний тест комплект перевіряє працездатність інтерфейсу користувача				
GLOBAL SETUP AND ADDITIONAL INFO:				
1. Запустити роботу сервера				
2. Зайти на сайт http://localhost:52735/Index				
3. Головна сторінка				

TC ID/Priority	01	1
IDEA: Правильне функціонування кнопок навігації сайту		
SETUP AND ADDITIONAL INFO:		

Головна сторінка містить навігацію для інших сторінок та частину інформації	
Revision History	
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші по черзі на всі елементи навігації.	Відкриваються відповідні сторінки з вмістом відповідним до назви елемента навігації і назви сторінки.

TC ID/Priority	02	1
IDEA: Правильне функціонування функції підбору		
SETUP AND ADDITIONAL INFO:		
При виборі цільової програми та критеріїв підбору система видає список збірок що підходять цим критеріям		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Вибрати цільову програму, вибрати критерій, натиснути на кнопку Find	Виводиться список програм з характеристиками відповідними заданим критеріям, розсортовані від меншої ціни до більшої

Перевірка працездатності інтерфейсу користувача на конкретних сторінках сайту

Author: Яцунь І.С	Specification ID: 2	Priority: 2	Producer: S.Petrovsky	Developer: Яцунь І.С
OVERVIEW:				

Даний тест комплект перевіряє працездатність інтерфейсу користувача
GLOBAL SETUP AND ADDITIONAL INFO:
1. Запустити роботу сервера
2. Зайти на сайт http://localhost:52735/
3. Головна сторінка

TC ID/Priority	03	2
IDEA: Правильне функціонування відображення списку Програм		
SETUP AND ADDITIONAL INFO:		
Сторінка програм відображає список програм		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Перейти на сторінку Програм	Список програм відображається правильно.

TC ID/Priority	04	2
IDEA: Правильне функціонування посилань на конкретну програму		
SETUP AND ADDITIONAL INFO:		
Сторінка програми містить деталі про програму		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на будь яку програму на сторінці програм	Посилання переводить користувача на сторінку програми, інформація про програму відображається коректною.

TC ID/Priority	05	2
-----------------------	-----------	----------

IDEA: Правильне відображення списку Збірок	
SETUP AND ADDITIONAL INFO: Сторінка збірки відображає списки збірок	
Revision History	
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку посилання Збірок	Кнопка переводить користувача на список доступних збірок

TC ID/Priority	06	2
IDEA: Правильне функціонування посилань на конкретну збірку		
SETUP AND ADDITIONAL INFO: Сторінка збірки містить деталі про збірки		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на будь яку збірку на сторінці збірок	Посилання переводить користувача на сторінку збірки, інформація про збірку відображається коректною.

Перевірка працездатності функцій та інтерфейсу адміністратора

Author: Яцунь І.С	Specification ID: 3	Priority: 3	Producer: S.Petrovsky	Developer: Яцунь І.С
OVERVIEW: Даний тест комплект перевіряє працездатність функцій та інтерфейсу адміністратора				

GLOBAL SETUP AND ADDITIONAL INFO:

1. Запустити роботу сервера
2. Зайти на сайт <http://localhost:52735/Admin>
3. Сторінка Адміністратора

TC ID/Priority	07	3
IDEA: Правильне функціонування сторінки Programs		
SETUP AND ADDITIONAL INFO: Сторінка програм відображає список програм для адміністратора		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Programs	Посилання переводить користувача на сторінку зі списком програм для адміністратора, в якому коректною відображається інформація про програми

TC ID/Priority	08	3
IDEA: Правильне функціонування сторінки Setups		
SETUP AND ADDITIONAL INFO: Сторінка програм відображає список збірок для адміністратора		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Setups	Посилання переводить користувача на сторінку зі списком збірок для

	адміністратора, в якому коректною відображається інформація про збірок
--	--

TC ID/Priority	09	3
IDEA: Правильне функціонування сторінки AddProgram		
SETUP AND ADDITIONAL INFO: Сторінка AddProgram дозволяє створювати нові екземпляри програм		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Створити програму 2. Ввести коректні дані у відповідну форму	У базі даних буде створено новий об'єкт моделі Program з введеними параметрами

TC ID/Priority	10	3
IDEA: Правильне функціонування сторінки AddSetup		
SETUP AND ADDITIONAL INFO: Сторінка AddSetup дозволяє створювати нові екземпляри збірки		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Створити збірку 2. Ввести коректні дані у відповідну форму	У базі даних буде створено новий об'єкт моделі Setup з введеними параметрами

TC ID/Priority	11	3
IDEA: Правильне функціонування сторінки DeleteProgram		

SETUP AND ADDITIONAL INFO:	
Сторінка DeleteProgram дозволяє видалити об'єкт Program, id якого було передано методу.	
Revision History	
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Delete біля необхідного екземпляра моделі	Цей елемент буде видалено з бази даних

TC ID/Priority	12	3
IDEA: Правильне функціонування сторінки EditProgram		
SETUP AND ADDITIONAL INFO:		
Сторінка EditProgram дозволяє змінити параметри об'єкту Program, id якого було передано методу.		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Нажати на кнопку Edit у відповідного елемента моделі. 2. ввести деякі зміни до параметрів елемента	Параметри елемента були змінені

TC ID/Priority	13	3
IDEA: Правильне функціонування сторінки EditSetup		
SETUP AND ADDITIONAL INFO:		
Сторінка EditSetup дозволяє змінити параметри об'єкту Setup, id якого було передано методу.		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Нажати на кнопку Edit у відповідного елемента моделі. 2. Ввести деякі зміни до параметрів елемента	Параметри елемента були змінені

3.4 Інструкція користувача

Перше, що бачить користувач, зайшовши на сайт – це стартова сторінка, зображена на рисунку 3.17.

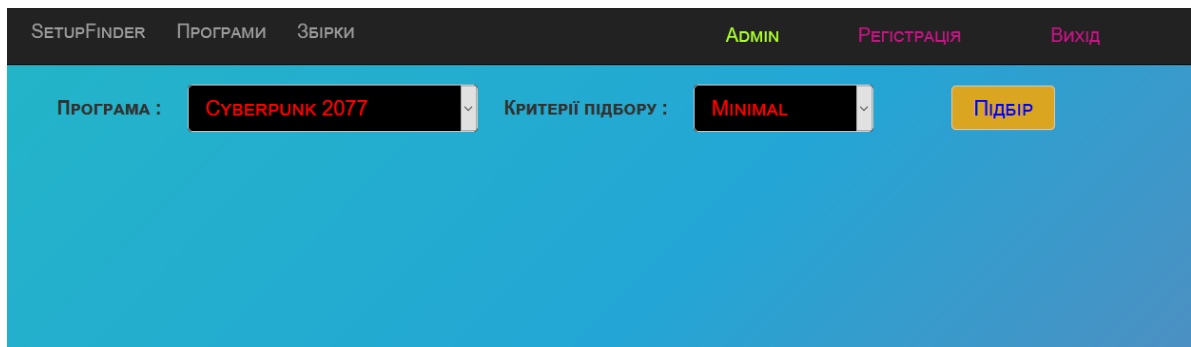


Рисунок 3.17 – Стартова сторінка

Далі користувач може зайти сторінку зі списком програм та Списком Збірок, що є у інформаційній системі Ці сторінки зображені на рисунках 3.18 та 3.19.

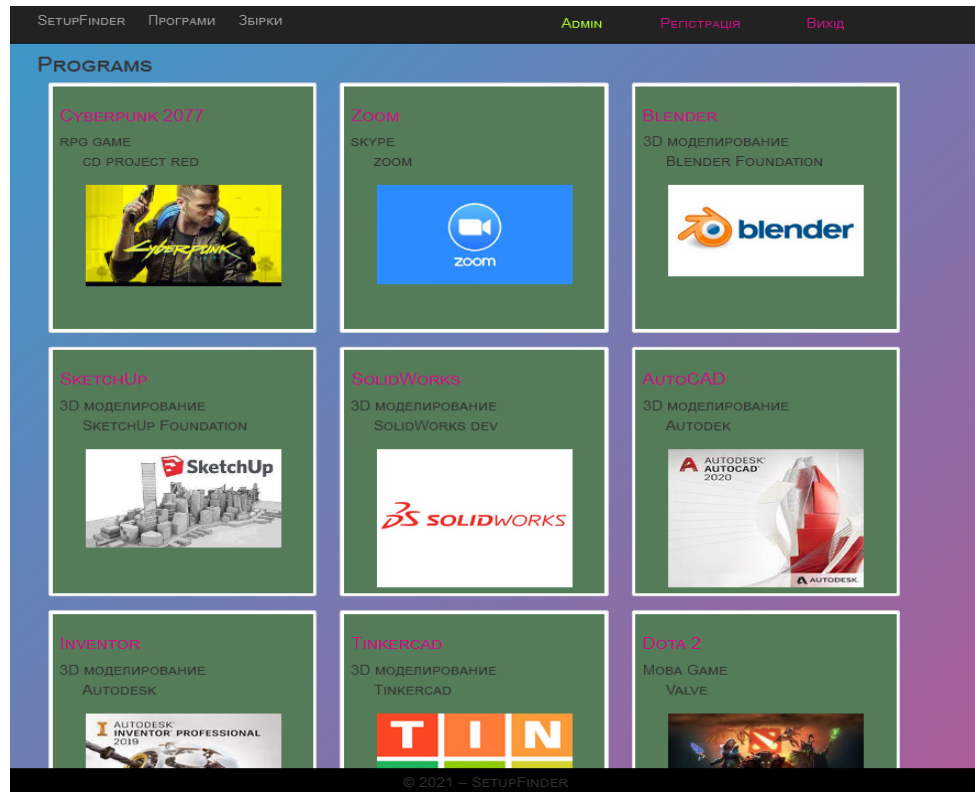


Рисунок 3.18 – Сторінка Programs

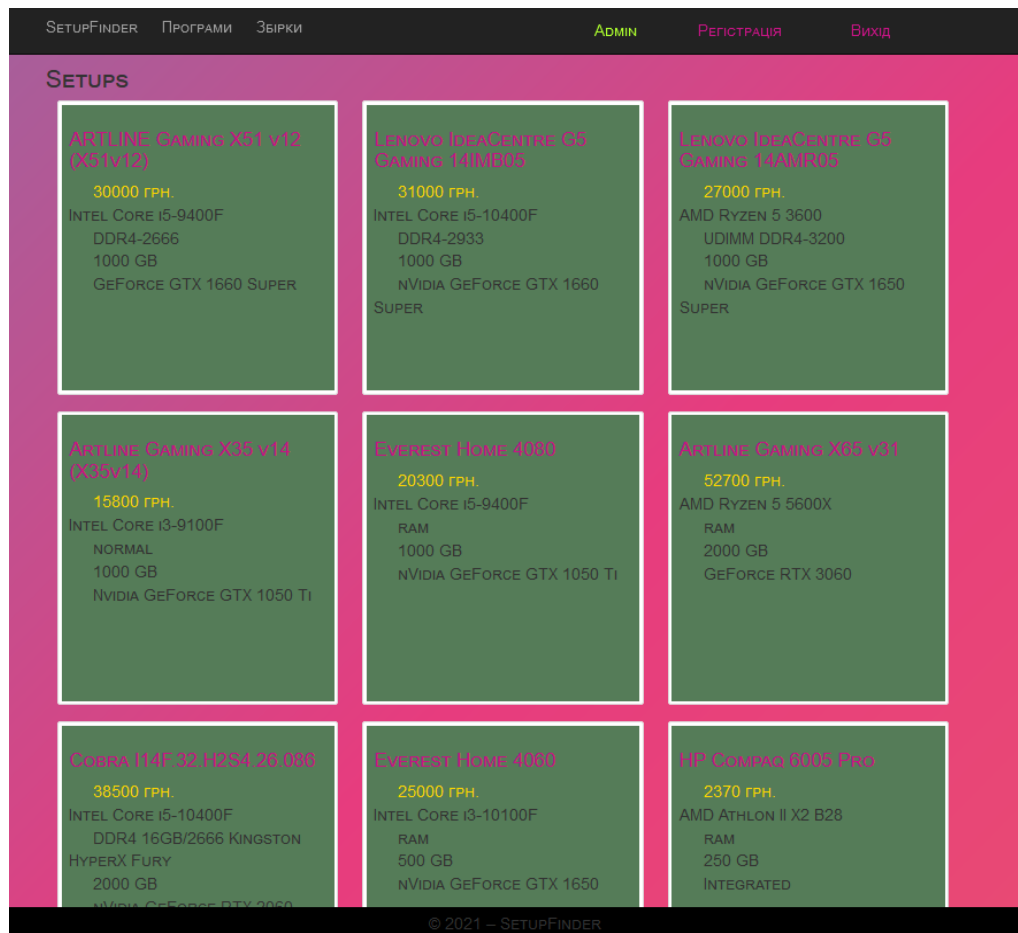


Рисунок 3.19 – Сторінка Setups


Ознайомившись зі списком програм або збірок, користувач може переглянути детальнішу інформацію про ці об'єкти, перейшовши через посилання на їх назві до деталей конкретного елемента (рисунок 3.20)

Рисунок 3.20 – Сторінка DetailsPrograms

SETUPFINDER ПРОГРАМИ ЗБІРКИ ADMIN РЕПСТРАЦІЯ Вихід

DETAILSPROGRAMS

PROGRAM

NAME	INVENTOR
IMAGE	
DESCRIPTION	INVENTOR
CATEGORY	3D МОДЕЛИРОВАНИЕ
DEVELOPER	AUTODESK
PROCESSORMIN	25
PROCESSORREC	30
PROCESSORCORE...	4
PROCESSORCORE...	6
RAMMIN	16
RAMREC	32
STORAGEMIN	30
STORAGEREC	40
VIDEOCARDMIN	1
VIDEOCARDREQ...	3000
VIDEOCARDREC	4
VIDEOCARDREQ...	7000

НАЗАД

© 2021 – SETUPFINDER

Крім цього, зі стартової сторінки користувач може перейти на сторінку з логіном або реєстрацією. Сторінка з реєстрацією зображена на рисунку 3.21.

SETUPFINDER ПРОГРАМИ ЗБІРКИ ADMIN РЕЄСТРАЦІЯ Вихід

РЕЄСТРАЦІЯ

REGISTERMODEL

NAME

PASSWORD

CONFIRMPASSWORD

[НАЗАД](#)

Рисунок 3.21 – Сторінка Реєстрації

Сторінка з логіном для користувача зображена на рисунку 3.22

SETUPFINDER ПРОГРАМИ ЗБІРКИ ADMIN РЕЄСТРАЦІЯ Вихід

ЛОГІН

NAME

PASSWORD

[НАЗАД](#)

Рисунок 3.22 – Логін

Основна функція інформаційної системи – підбір збірки по цільовій програмі. Користувач може провести цю операцію перейшовши на головну сторінку (рисунок 3.17), вибравши цільову програму та критерій підбору (мінімальні або рекомендовані) та натиснувши кнопку Filter. Результат цієї

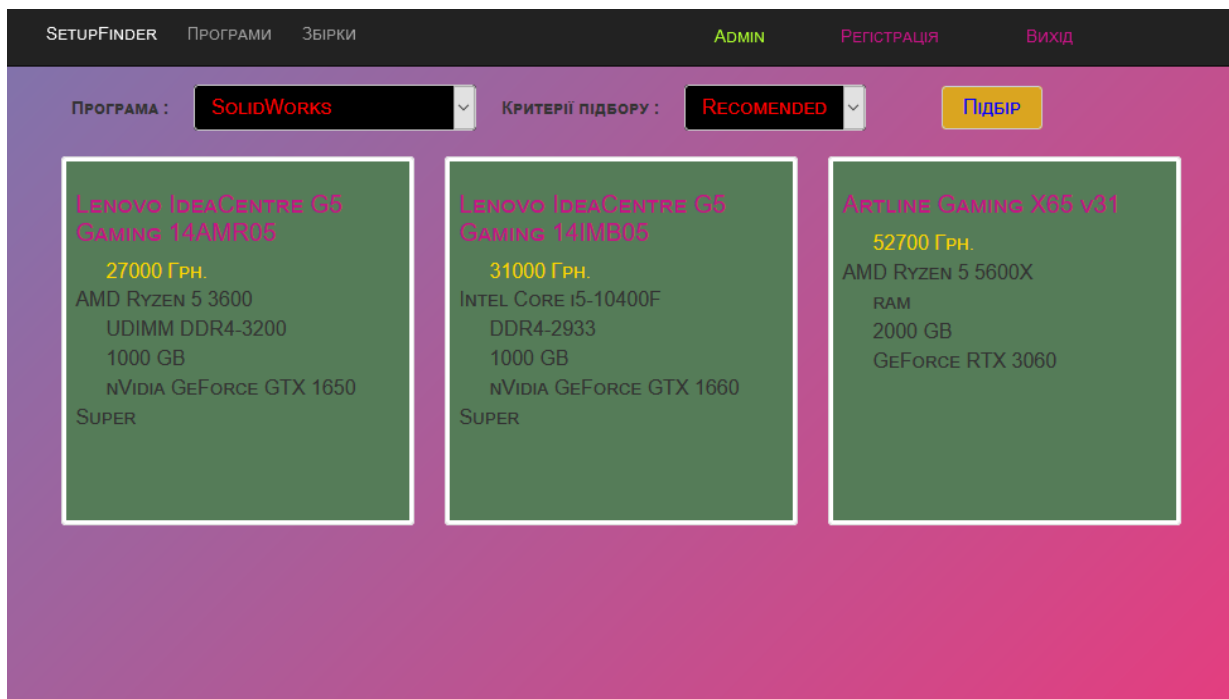


Рисунок 3.23 – Виконаний підбір збірки

Отже, користування інформаційною системою є інтуїтивно зрозумілим та не вимагає додаткових знань, умінь та навичок, окрім необхідних для звичайного користування комп'ютером.

3.5 Вимоги до апаратних та програмних засобів

Мінімальні вимоги до браузера клієнта :

- підтримка HTML 5;
- підтримка CSS 3;

Мінімальні вимоги до серверу:

- сервер IIS.
- фреймворк ASP.NET MVC 5
- NET 3.5
- System.Web.Mvc
- System.Web.Routing
- System.Web.Abstractions

- Бібліотеки DLL маршрутизації та абстракцій (у випадку відсутності пакета SP1)

Мінімальні вимоги до СКБД:

- СКБД MSSQL SERVER 2017+

Додаткові вимоги до системи:

- для роботи вимагається підключення до Інтернету;
- не підходить Internet Explorer та інші застарілі браузерери або старі версії сучасних браузерів;
- сайт не створений для перегляду на мобільних пристроях або інших пристроях з маленьким розміром екрану.

Висновки

В результаті виконання даної кваліфікаційної роботи було створено інформаційну систему оптимального підбору збірок комп'ютера за вибраними цільовими програмами. Створена вона була за допомогою платформи .Net, фреймворку ASP.NET MVC і бази даних Microsoft SQL Server . Всі вимоги, поставлені до роботи, були виконані, а саме була створена база даних та застосунок, який її використовує.

Створена система виконує наступні функції:

- робота з відкритою інформацією (виведення інформації про цільові програми та збірки комплектуючих комп'ютера зареєстрованих в системі);
- робота з підбором оптимальної збірки з урахуванням цільової програми та критеріями підбору(мінімальні або рекомендовані);
- робота з панеллю адміністратора (додавання цільових програм та збірок комп'ютерних комплектуючих, редагування та видалення цієї інформації);
- робота з особистими даними (реєстрація в системі, авторизація);
- робота з внутрішніми даними системи (внутрішня інформація про цільові програми, та збірки комп'ютера);

Дана система може бути використана для підбору оптимальної збірки комп'ютера за певною програмою, що вважається цільовою для оптимізації роботи ПК. Її можна вдосконалити, додавши можливість пошуку, сортування програм та збірок, удосконаливши алгоритм підбору, удосконаливши інтерфейс та зробивши його більш привабливим, а також швидшим.

Перелік посилань

1. Hotline – Режим доступу: <https://hotline.ua/>
2. E-Katalog – Режим доступу: <https://ek.ua/>
3. Wikipedia. – API <https://ru.wikipedia.org/wiki/API>
4. Головний сайт платформи .NET – <http://www.microsoft.com/net/dd>
5. Головний сайт платформи PHP – <http://www.php.net/>
6. Апаратне забезпечення ПК <https://bondarenko.dn.ua/aparatne-zabezpechennya-pk/>
7. Найкращі backend фреймворки в 2020 році
<https://top.studio/stati/luchshie-backend-frejmworki-v-2020-godu/>
8. C# документація <https://docs.microsoft.com/en-us/dotnet/csharp/>
9. Microsoft SQL Server
https://uk.wikipedia.org/wiki/Microsoft_SQL_Server

Додатки

Додаток А – Лістинг коду системи

```

namespace Diplom.Controllers
{
    public class HomeController : Controller
    {
        private SetupContext db = new
        SetupContext();
        public ActionResult Index()
        {
            ViewBag.UserName =
            User.Identity.Name;
            SelectList programs = new
            SelectList(db.Programs, "ProgramId", "Name");
            ViewBag.Programs = programs;
            List<String> Types = new
            List<String>()
            {
                "Minimal",
                "Recomended",
            };
            SelectList SelectTypes = new
            SelectList(Types);
            ViewBag.Types = SelectTypes;
            return View();
        }
        public ActionResult Find(int?
        program, string type)
        {
            ViewBag.UserName =
            User.Identity.Name;
            SelectList programs = new
            SelectList(db.Programs, "ProgramId", "Name");
            ViewBag.Programs = programs;
            List<String> Types = new
            List<String>()
            {
                "Minimal",
                "Recomended",
            };
            SelectList SelectTypes = new
            SelectList(Types);
            ViewBag.Types = SelectTypes;
            List<Setup> SortedSetups = new
            List<Setup>();
            IQueryable<Setup> setups = db.Setups;
            Program a =
            db.Programs.Find(program);
            if(type.Equals("Minimal"))
            {
                foreach (Setup set in setups)
                {
                    if(set.Processor >=
                    a.ProcessorMin && set.ProcessorCore >=
                    a.ProcessorCoreMin && set.RAM >= a.RamMin
                    && set.Videocard >=
                    a.VideocardMin && set.VideocardFreq >=
                    a.VideocardFreqMin)
                    {
                        SortedSetups.Add(set);
                    }
                }
            }
            if(type.Equals("Recomended"))
            {
                foreach (Setup set in setups)
                {
                    if (set.Processor >=
                    a.ProcessorRec && set.ProcessorCore >=
                    a.ProcessorCoreRec && set.RAM >= a.RamRec
                    && set.Videocard >=
                    a.VideocardRec && set.VideocardFreq >=
                    a.VideocardFreqRec)

```

```

                {
                    SortedSetups.Add(set);
                }
            }
        }
        SortedSetups = SortedSetups.OrderBy(x
        => x.AvgPrice).ToList();
        ViewBag.SortedSetups = SortedSetups;

        return View();
    }
    public ActionResult Programs()
    {
        IQueryable<Program> programs =
        db.Programs;
        return View(programs);
    }
    public ActionResult Setups()
    {
        IQueryable<Setup> setups = db.Setups;
        return View(setups);
    }
    [HttpGet]
    public ActionResult DetailsPrograms(int?
    id)
    {
        IQueryable<Program> programs =
        db.Programs;
        Program a = db.Programs.Find(id);
        ViewBag.Image = a.Image;
        if (a == null)
        {
            return HttpNotFound();
        }
        return View(a);
    }
    [HttpGet]
    public ActionResult DetailsSetups(int?
    id)
    {
        Setup a = db.Setups.Find(id);
        if (a == null)
        {
            return HttpNotFound();
        }
        return View(a);
    }
    public ActionResult UserLogin()
    {
        ViewBag.UserName =
        User.Identity.Name;
        return PartialView();
    }
}

namespace Diplom.Controllers
{
    public class AccountController : Controller
    {
        public ActionResult Login()
        {
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Login(LoginModel
        model)
        {
            if (ModelState.IsValid)

```

```

    {
        // поиск пользователя в бд
        User user = null;
        using (SetupContext db = new
SetupContext())
        {
            user =
db.Users.FirstOrDefault(u => u.Login ==
model.Name && u.Password == model.Password);
        }
        if (user != null)
        {
            FormsAuthentication.SetAuthCookie(model.Name,
true);
            return
RedirectToAction("Index", "Home");
        }
        else
        {
            ModelState.AddModelError("",
"Пользователя с таким логином и паролем нет");
        }
        return View(model);
    }

    public ActionResult Register()
    {
        return View();
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult
Register(RegisterModel model)
    {
        if (ModelState.IsValid)
        {
            User user = null;
            using (SetupContext db = new
SetupContext())
            {
                user =
db.Users.FirstOrDefault(u => u.Login ==
model.Name);
            }
            if (user == null)
            {
                // создаем нового
пользователя
                using (SetupContext db = new
SetupContext())
                {
                    db.Users.Add(new User {
Login = model.Name, Password = model.Password,
RoleId = 1 });
                    db.SaveChanges();

                    user = db.Users.Where(u
=> u.Login == model.Name && u.Password ==
model.Password).FirstOrDefault();
                }
                // если пользователь удачно
добавлен в бд
                if (user != null)
                {
                    FormsAuthentication.SetAuthCookie(model.Name,
true);
                    return
RedirectToAction("Index", "Home");
                }
            }
        }
        else
        {
            ModelState.AddModelError("",
"Пользователь с таким логином уже существует");
        }
        return View(model);
    }
    public ActionResult Logoff()
    {
        FormsAuthentication.SignOut();
        return RedirectToAction("Index",
"Home");
    }
}

namespace Diplom.Controllers
{
    public class AdminController : Controller
    {
        private SetupContext db = new
SetupContext();
        // GET: Admin
        [Authorize(Roles = "admin")]
        public ActionResult Index()
        {
            return View();
        }
        [Authorize(Roles = "admin")]
        public ActionResult Programs()
        {
            IQueryable<Program> programs =
db.Programs;
            return View(programs);
        }
        [Authorize(Roles = "admin")]
        public ActionResult Setups()
        {
            IQueryable<Setup> setups = db.Setups;
            return View(setups);
        }
        [HttpGet]
        [Authorize(Roles = "admin")]
        public ActionResult DetailsPrograms(int?
id)
        {
            IQueryable<Program> programs =
db.Programs;
            Program a = db.Programs.Find(id);
            ViewBag.Category = a.Category;
            ViewBag.Developer = a.Developer;
            ViewBag.Image = a.Image;
            if (a == null)
            {
                return HttpNotFound();
            }
            return View(a);
        }
        [HttpGet]
        [Authorize(Roles = "admin")]
        public ActionResult DetailsSetups(int?
id)
        {
            Setup a = db.Setups.Find(id);
            if (a == null)
            {
                return HttpNotFound();
            }
            return View(a);
        }
        [HttpGet]
        [Authorize(Roles = "admin")]
        public ActionResult AddProgram()
        {

```

```

        return View();
    }
    [HttpPost]
    [Authorize(Roles = "admin")]
    public ActionResult AddProgram(Program
program, HttpPostedFileBase programImg)
    {
        // if (ModelState.IsValid)
        // {
        if (programImg != null)
        {
            var fileName =
Path.GetFileName(programImg.FileName);
            var directoryToSave =
Server.MapPath(Url.Content("~/Pictures/Programs")
);
            var pathToSave =
Path.Combine(directoryToSave, fileName);
            programImg.SaveAs(pathToSave);
            program.Image = fileName;
        }
        db.Programs.Add(program);
        db.SaveChanges();
        // }
        return RedirectToAction("Programs");
    }
    [HttpGet]
    [Authorize(Roles = "admin")]
    public ActionResult AddSetup()
    {
        return View();
    }
    [HttpPost]
    [Authorize(Roles = "admin")]
    public ActionResult AddSetup(Setup setup)
    {
        if (ModelState.IsValid)
        {
            db.Setups.Add(setup);
            db.SaveChanges();
        }
        return RedirectToAction("Setups");
    }
    [HttpGet]
    [Authorize(Roles = "admin")]
    public ActionResult DeleteSetup(int? id)
    {
        Setup a = db.Setups.Find(id);
        if (a == null)
        {
            return HttpNotFound();
        }
        return View(a);
    }
    [HttpPost, ActionName("DeleteSetup")]
    [Authorize(Roles = "admin")]
    public ActionResult
DeleteSetupConfirmed(int? id)
    {
        Setup a = db.Setups.Find(id);
        if (a == null)
        {
            return HttpNotFound();
        }
        db.Setups.Remove(a);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    [HttpGet]
    [Authorize(Roles = "admin")]
    public ActionResult DeleteProgram(int?
id)
    {
        Program a = db.Programs.Find(id);
        if (a == null)
        {
            return HttpNotFound();
        }
        return View(a);
    }
    [HttpPost, ActionName("DeleteProgram")]
    [Authorize(Roles = "admin")]
    public ActionResult
DeleteProgramConfirmed(int? id)
    {
        Program a = db.Programs.Find(id);
        if (a == null)
        {
            return HttpNotFound();
        }
        if (a.Image != null)
        {
            var fileName =
Path.GetFileName(a.Image);
            var directoryToDelete =
Server.MapPath(Url.Content("~/Pictures/Programs")
);
            var pathToDelete =
Path.Combine(directoryToDelete, fileName);
            System.IO.File.Delete(pathToDelete);
        }
        db.Programs.Remove(a);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    [HttpGet]
    [Authorize(Roles = "admin")]
    public ActionResult EditProgram(int? id)
    {
        if (id == null)
        {
            return HttpNotFound();
        }
        Program program =
db.Programs.Find(id);
        if (program != null)
        {
            return View(program);
        }
        return HttpNotFound();
    }
    [HttpPost]
    [Authorize(Roles = "admin")]
    public ActionResult EditProgram(Program
program, HttpPostedFileBase programImg)
    {
        /*
        if (game.Image != null)
        {
            var fileName =
Path.GetFileName(game.Image);
            var directoryToDelete =
Server.MapPath(Url.Content("~/Pictures/Games"));
            var pathToDelete =
Path.Combine(directoryToDelete, fileName);
            System.IO.File.Delete(pathToDelete);
        }
        if (productImg != null)
        {
            var fileName =
Path.GetFileName(productImg.FileName);
            var directoryToSave =
Server.MapPath(Url.Content("~/Pictures/Games"));

```

```

        var pathToSave =
Path.Combine(directoryToSave, fileName);
productImg.SaveAs(pathToSave);
        game.Image = fileName;
    }
    game.CategoryID = category;
    game.DeveloperID = developer;

    db.Games.Add(game);
    db.SaveChanges();
*/
if (programImg != null)
{
    var fileName =
Path.GetFileName(programImg.FileName);
    var directoryToSave =
Server.MapPath(Url.Content("~/Pictures/Programs")
);
        var pathToSave =
Path.Combine(directoryToSave, fileName);
    programImg.SaveAs(pathToSave);
    program.Image = fileName;
}
db.Entry(program).State =
EntityState.Modified;
db.SaveChanges();

```

```

        return RedirectToAction("Index");
}
[HttpGet]
[Authorize(Roles = "admin")]
public ActionResult EditSetup(int? id)
{
    if (id == null)
    {
        return HttpNotFound();
    }
    Setup setup = db.Setups.Find(id);
    if (setup != null)
    {
        return View(setup);
    }
    return HttpNotFound();
}
[HttpPost]
[Authorize(Roles = "admin")]
public ActionResult EditSetup(Setup
setup)
{
    db.Entry(setup).State =
EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}

```

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Яцунь І. С. на захист дипломного проекту (роботи)
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Система оптимального підбору товарів

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



(підпис)

САВЕНКО О. С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Яцунь І. С. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 3,12 %, добре 34,38 %, задовільно 62,50 %.

шкалою ЄКТС: А 5,45 %, В 12,73 %, С 16,36 %, D 10,91 %, E 54,55 %.

Методист факультету

(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент

Яцунь І. С. Рецензував кваліфікаційну роботу
на тему "Система оптимального
підбору товарів" у відповідності до завдань
у програмі оцінки.

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

(підпис)

Петровський С. С.
(прізвище та ініціали)

" 11 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Яцунь І. С. допускається до захисту цього проекту

Завідувач кафедри

(назва)

" 11 " 06 2021 р.

(підпис, прізвище, ініціали)

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 58.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 11%**

ID: 92683 Название: Система оптимального підбору товарів Добавлено в БД: 2021-06-08 Авторы: І.С. Яцунь Руководители: С.С. Петровський Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	34058	338	20649 (61%)	220 (65%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы
90911	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-18 Авторы: Яцунь І.С. Руководители: Скрипник Т.К. Консультанты: Оponentы:	19851 (58.0%)	215 (64.0%)
90176	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-11 Авторы: Яцунь І.С. Руководители: Скрипник Т.К. Консультанты: Оponentы:	18541 (54.0%)	201 (59.0%)

Ім'я користувача:
Кафедра КН

ID перевірки:
1008228073

Дата перевірки:
08.06.2021 13:03:08 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.06.2021 13:22:28 EEST

ID користувача:
100005671

Назва документа: Диломна Яцунь Lite

Кількість сторінок: 48 Кількість слів: 5738 Кількість символів: 43210 Розмір файлу: 1.46 MB ID файлу: 1008301751

7.44% Схожість

Найбільша схожість: 4.25% з джерелом з Бібліотеки (ID файлу: 1008298363)

2.2% Джерела з Інтернету

90

Сторінка 50

7.08% Джерела з Бібліотеки

47

Сторінка 50

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему *Система оптимального підбору товарів*

Галузь знань *12 – Інформаційні технології*
Шифр і назва галузі знань
Спеціальність *122 – Комп'ютерні науки*
Шифр і назва спеціальності
Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Версія шаблону © CHONG 2021

Виконав: *студент 4 курсу, група КН-17-2* *І.С. Яцунь*
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: *к.т.н., доцент кафедри КНІТ* *С.С. Петровський*
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: *к.т.н., доцент кафедри КНІТ* *Р.О. Багрій*
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КНІТ, д.т.н., професор *О.В. Бармак*
Підпис Ініціали, прізвище
_____ 2021 р.

Хмельницький 2021

Вступ

Тема моєї роботи: створення інформаційної системи для підбору комплектуючих запчастин комп'ютеру. інформаційна система для підбору – це не магазин, він лише підбирає і рекомендує товар, але може враховувати ціни з інших магазинів. Підбір комплектуючих має свої специфікації і відмінності від, наприклад, «підбору домашньої техніки». Основна відмінність – абсолютно інші критерії підбору. Кожна запчастина має свій набір характеристик, а також - сумісність з іншими запчастинами, що разом з ціною буде складати критерії для вибору запчастини для рекомендації. Важливим критерієм для вибору комп'ютерної техніки є її можливість використання для запуску і користування конкретним програмним забезпеченням : робочими програмами, іграми, програмним забезпеченням зі спеціальними функціями. Саме тому також важливий підбір комплектуючих через критерій можливості запуску конкретних програм, для яких вказані рекомендовані характеристики техніки.

Тобто ціль кваліфікаційної роботи актуальна, багатьом покупцям комп'ютерних запчастин буде цікаво скористатись рекомендаціями системи, що і дало мені ідею вибрати саме цю тему для кваліфікаційної роботи.

Розділ 1. Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

В залежності від задач, що ставляться перед комп'ютером, різні конфігурації архітектури та запчастин будуть показувати різні результати. Через цей факт, та через вартість комп'ютерних запчастин доцільно було б підійти до обрання комп'ютера з розумом. Цю задачу можна назвати Оптимізацією підбору збірки комп'ютера.

Апаратне забезпечення [6] — це електронні частини пристрою для обчислення (можливо механічні), з яких складається система (також мережа). Апаратним забезпеченням можна вважати: пристрої з функціоналом виводу будь-якої інформації та її вводу, різних електронні прилади з електронними схемами, як електричні так і аналогові, пристрої живлення, апаратура для проведення діагностичних робіт, аналізу та тестування, та різноманітні допоміжні компоненти.

Більшість запчастин у складі системи вміщенні у корпус комп'ютера, що допомагає механічно зафіксувати і захистити їх при переміщенні, так і від електростатичних розрядів. Важлива також і структура корпусу, так як вона змінює переміщення повітря всередині нього, що необхідно для охолодження компонентів у випадку нагрівання. Також корпус впливає на захист від електромагнітного випромінювання, генерованого електричними схемами. У різних корпусах може бути різна кількість простору для різних запчастин, наприклад у великих корпусах є вистачає простору для додаткових твердих дисків і іншої додаткової периферії, що може бути необхідним. Корпус комп'ютера часто змінюється під естетичні цілі користувача, або під особливі потреби.

Часто у корпусі комп'ютера знаходиться блок живлення, що перетворює змінний струм із місцевої мережі на необхідну комп'ютеру напругу. Від вбудованих же акумуляторів працюють ноутбуки, а блок живлення використовується як пристрій для зарядки.

Системна плата — одна з найважливіших частин комп'ютера, так як вона об'єднує в одне ціле всі основні запчастини, а саме: центральний процесор, оперативну пам'ять та жорсткий диск, підключення засобів вводу-виводу. Розглянемо детальніше компоненти, об'єднані системною платою:

1. Центральний процесор – електронний компонент, за допомогою якого здійснюється більшість основних обчислень та на якому зав'язана здатність до роботи комп'ютера, та виконуваних на ньому програм. В процесі роботи процесор зазвичай швидко нагрівається, і через це необхідно охолодження радіатором з допомогою одного або декількох вентиляторів; також використовують і системи охолодження з використанням рідин. З часом процесори комп'ютера почали вміщувати також і вбудований графічний співпроцесор. Тактова частота процесора – це величина, що позначає швидкість виконання процесів, і позначається як правило у мегагерцах або гігагерцах. Досить часто сучасні процесори можна також «розігнати», що понаднормово збільшить частоту», але також і збільшити при цьому тепловиділення відповідно. Також це називають «оверклокінгом».

2. Мікросхеми підтримки, або чіпсет (англ. chipset), складається з північного та південного моста. Головний функціонал чіпсета: забезпечення зв'язку між іншими запчастинами системної плати, такими як: процесором, оперативною пам'яттю, пристроями вводу-виводу, та шинами розширення.

3. Оперативна пам'ять (RAM - англ. Random Access Memory), це елемент куди поміщуються код і необхідні для швидкого доступу дані виконуваних програм. Оперативна пам'ять виготовляється за стандартом DIMM, що установлює стандартні об'єми пам'яті модулів в 2, 4, 8, і 16 гігабайт.

4. Мікросхеми постійної пам'яті (ROM - англ. Read Only Memory), це деталь в якій знаходиться firmware, або базова система вводу - виводу — це основна системна програма, що виконується, коли на комп'ютер включається або він перезапускається. Такий процес називається bootstrapping[en], і являє собою початковий запуск системи і підготовку до початку роботи.

5. Шини розширення, що використовуються для під'єднання зі слотами плати, що реалізують додаткову функціональність.

6. Гальванічний елемент для живлення напівпостійної пам'яті BIOS.

7. Мікросхеми контролера відео, якщо системна плата має під'єднаний монітор через відеовихід.

8. Інші контролери і інтерфейси вводу-виводу.

9. Карта розширення або плата розширення — це плата зі стандартизованим роз'ємом, що вставляється у слот на системній платі для розширення функціональності комп'ютера. Наприклад, таким способом можна додати до системи звукову плату, вставивши її у слот розширення.

10. Пристрої для накопичення та збереження інформації:

- Жорсткий диск — пристрій для збереження інформації по принципу магнітного запису.
- Твердотільний накопичувач (SSD) — запам'ятовуючий пристрій на основі мікросхем пам'яті, являється швидшим ніж жорсткий диск, і тому вважається кращим варіантом, хоча вартість на одиницю пам'яті більша.
- Оптичний накопичувач — пристрій для зчитування інформації з оптичних носіїв, також можливий і запис інформації (CD-ROM або DVD-ROM). [6]

Для оптимізації підбору збірки комп'ютера, було вирішено відбирати підходящі збірки шляхом зрівняння характеристик збірки з мінімальними та рекомендованими необхідними характеристиками цільової програми. Цими характеристиками являються:

- Частота роботи процесора;
- Кількість ядр процесора;
- Об'єм відео пам'яті відео карти;
- Частота роботи пам'яті відео карти;
- Об'єм оперативної пам'яті;
- Об'єм вільної пам'яті на диску

Враховуючи актуальність даного напрямку, в якості теми для кваліфікаційної роботи була прийнята тема «Система оптимального підбору товарів» та мова програмування C# з фреймворком ASP.NET, в якості базової платформи для реалізації.

Отже основними параметрами предметної області для задачі, що розглядається є:

Програмний продукт	Параметрами є: назва, зображення, розробник, мінімальні характеристики, рекомендовані характеристики, категорія
Збірка комп'ютера	Параметрами є: ціна(приблизно), процесор, оперативна пам'ять, Жорсткий диск, Відеокарта
Користувач	Параметрами є: логін, пароль, роль

1.2 Аналіз інформаційного забезпечення предметної області

1.2.1 Аналіз існуючого програмного забезпечення предметної

області

E-Katalog - каталог описів і цін на різноманітну побутову і комп'ютерну техніку, електроніку, товари для дому та офісу. За допомогою цього сервісу можна підібрати і купити необхідну модель за найкращою ціною в порівнянні з різними інтернет-магазинами. У цьому сервісі можна знайти багато інформації, що пригодиться у виборі товарів, параметри для підбору, детальні описи товарів, пошук товару за різними критеріями, відгуки користувачів, рейтинг товарів, рекомендації експертів, каталог брендів і багато інших. [2]

На E-Katalog є функція порівняння товарів по характеристикам,

7

Що схожа на функцію моєї інформаційної системи для підбору комплектуючих запчастин комп'ютеру. На рисунку 1.1 Можна побачити як виглядає зрівняння декількох процесорів.



Рисунок 1.1 – Зрівняння процесорів на «E-Katalog»

Hotline – ще один сервіс схожий на E-Katalog. Hotline.ua це інтернет - сервіс, що допомагає покупцеві знайти, порівняти і визначитися з вибором товару, магазину і зробити покупку. Цей сервіс рекламує наявні у різноманітних магазинах товари. Хотлайн це популярний сервіс, який регулярно відвідує понад 250 тис. користувачів. Користуючись сервісами Хотлайн можна значно спростити пошук, вибір та порівняння, що значно допоможе з покупкою і економією.[1] Система порівняння товарів на Hotline зображена на рисунку 1.2.

Дата добавления товара	10 декабря 2020 г.	16 января 2021 г.	20 мая 2021 г.	9 января 2021 г.
Объем памяти	32, 32, 32, 32 GB	32, 32, 32, 32 GB	32, 32, 32, 32 GB	—
Тип	Процессор	Процессор	Процессор	Процессор
Модель	AMD Ryzen 5 2600	AMD Ryzen 5 2600	AMD Ryzen 5 2600	AMD Ryzen 5 2600
Платформа	Socket AM4	Socket AM4	Socket AM4	Socket AM4
Частота	—	—	—	—
Полная мощность TDP	65 Вт/65 Вт	65 Вт/65 Вт	65 Вт/65 Вт	65 Вт/65 Вт
Частота тактовой частоты	3,7 ГГц	3,7 ГГц	3,7 ГГц	—
Кэш-память	—	—	—	—
Объем накопителя (встроенный)	0	16	16	16

Рисунок 1.2 – Зрівняння процесорів на Hotline»

E-Katalog

Преваги:

- Зручний сервіс
- Зрозумілий у користуванні
- Великий каталог товарів, магазинів
- Краща функція зрівняння товарів

Недоліки:

- Застарілий дизайн
- Багато магазинів з сумнівною репутацією, через що можливе шахрайство
- Можлива обманна ціна (спеціальне зниження ціни на сервісі, коли справжня ціна вища)

Hotline

Переваги:

- Зручний фільтр
- Приємний дизайн
- Є система бонусів
- Актуальний моніторинг цін

Недоліки:

- Часті скарги на видалення та редагування негативних коментарів
- Багато магазинів з сумнівною репутацією, через що можливе шахрайство
- Маніпуляція рейтингами магазинів

Данні сайти – сервіси мають на меті допомогти підібрати товар, і магазин для його покупки, але вони надають характеристики, ціни різних магазинів і відгуки, а не підбирають оптимальний товар по запиту користувача. Тому створення алгоритму який буде підбирати оптимальний варіант по характеристикам і ціні враховуючи потреби користувача є актуальним завданням.

1.2.2 Аналіз сучасних засобів створення програмного забезпечення

На даний момент є величезна кількість різноманітних фреймворків, платформ, мов, та різноманітних технологій веб розробки. Можна виділити такі популярні фреймворки для веб розробки[7]:

Мова PHP

- Symphony
- Zend
- Laravel
- Kohana
- Yii

Мова RUBY

- Ruby on Rails (явний лідер)
- Sinatra

Мова Python

- Django
- Plone
- Twisted
- Flask
- Tornado

Мова Java

- Spring MVC
- JSF
- GWT

Мова Kotlin

- Spring
- Ktor

Мова C#

- ASP.NET (ASP.NET MVC) и .Net Framework

Було вирішено розглянути детальніше 3 найбільш відомих мені мов програмування: .NET, та PHP .

Microsoft .NET [4] – технологія програмування, розроблена компанією Microsoft як платформа для створення різноманітних програм, включаючи також і веб-додатки. Ця технологія в великій мірі є розвитком та продовженням принципу та методики початих технологією програмування Java. Важливою ідеєю .NET є об'єднання різноманітних служб, написаних різними мовами. Ця здатність виставляється Microsoft як перевага .NET, хоча технологія Java уже мала такий функціонал. .NET складається з двох важливих частин– середовище виконання (по суті віртуальна машина) та інструментарій розробки.

.NET – являється технологією з наявністю крос-платформи, і на даний момент доступна реалізація для таких платформ як Microsoft Windows, FreeBSD (також від Microsoft), варіант технології для ОС Linux в проекті Mono, DotGNU .

ASP.NET [6] — це фреймворк який являється частиною платформи Microsoft .NET. Це технологія створення веб-додатків, веб застосунків та сервісів від компанії Microsoft. Її можна вважати розвитком більш давньої технології Microsoft ASP.

Розробляти та працювати над проектами .NET-програм можна у різних середовищах проектування: Visual Studio, SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse також має додаток для розробки .NET-програм. Деякі програми можна розробляти навіть в текстовому

редакторі та використовувати консольний компілятор. На платформі .NET в якості бази даних краще за все використовувати MS SQL Server.

ASP.NET будується на принципі Common Language Runtime: розробники мають можливість писати код для ASP.NET, використовуючи велику кількість різноманітних мов програмування, з яких деякі входять в комплект .NET Framework (C#, JavaScript .NET, і Visual Basic .NET), і ще більше інших мов можуть бути установленні додатково (Python, PHP, Perl, Smalltalk, Ruby).

На ASP.NET можлива розробка проектів з реалізацією шаблону Model-View-Controller (ASP.NET MVC Framework).

PHP [5] (англ. PHP: Hypertext Preprocessor – PHP: гіпертекстовий препроцесор) – це скриптова мова програмування, що була розроблена для створення HTML сторінок шляхом генерації із веб сервера. PHP являється одною з найпоширеніших мов програмування для сфери веб-додатків. PHP підтримується переважною більшістю хостинг-провайдерів.

Суть роботи PHP у інтерпретації коду веб-сервером як HTML-код, що після цього вже буде переданий на пристрій користувача. Велика різниця у порівнянні зі мовою JavaScript в тому, що клієнт не отримує PHP-код, а лише готовий html файл що отримує браузер. Даний спосіб передачі даних є великим плюсом зі сторони безпеки, але він також робить важчою інтерактивну роботу веб сторінок. Однак можливо використовувати PHP код для генерування JavaScript файлу, який вже може бути виконаним браузером користувача.

Laravel [6] – це безплатний фреймворк з принципом відкритого коду. Він був створений Тейлором Отвеллом, з ціллю розробки веб-додатків по структурному шаблону MVC. На створення цього фреймворку був виказаний великий вплив інших PHP фреймворків. Створений він був з ціллю бути сильною альтернативою Codeigniter, у якому на думку розробника не було достатньо корисних засобів для ефективної розробки веб сайтів. В своїй

основі Laravel має велику кількість іншого популярного фреймворку – Symfony.

Велику популярність даний фреймворк завоював у західних програмістів. Laravel користується менеджером пакетів Composer, за допомогою якого він може легко встановлювати і підключати різноманітні компоненти для використання у веб додатках. Також Laravel реалізовує шаблон ActiveRecord - Eloquent ORM, що дозволяє йому встановити відношення між різноманітними об'єктами бази даних веб додатку та побудувати зручні запити для управління даними різного типу. Механізм авто підвантаження класів дає можливість не підключати файли вручну через include, и зупиняє загрузку невикористаних компонентів. Має зручну систему міграцій, що допомагає опростити оновлення і розвертання веб-додатків.

В Laravel використовується шаблон Blade, який дозволяє об'єднувати html код та код php, за допомогою якого можна робити нескладні представлення вашого веб-додатку використовуючи особливу структуру коду.

1.3 Постановка задачі

Метою кваліфікаційної роботи є розробка інформаційної системи для підбору комплектуючих запчастин комп'ютеру на платформі .NET , що виконує наступні основні функції:

- виведення інформації про комплектуючі та програмне забезпечення;
- адмін-панель для додавання нових одиниць комплектуючих та програмного забезпечення;
- редагування і видалення вже створених даних;
- робота з особистими даними (реєстрація в системі, авторизація);
- виведення оптимальних товарів по заданим критеріям;

Розділ 2. Проектування структури інформаційної системи

2.1 Аналіз та автоматизація обробки інформаційних потоків

При розробці інформаційної системи для підбору комплектуючих запчастин комп'ютеру необхідно реалізувати наступні бізнес-процеси:

- Робота з інформацією про збірки;
- Робота з підбором збірки;
- Робота з вибором програм для підбору;
- Робота з адмін-панеллю;
- Робота з інформацією про програми;
- Робота з інформацією про збірки;

Робота з інформацією про збірки - надає можливість користувачу інформаційної системи отримувати інформацію про доступні на інформаційної системи збірки, а саме:

- Інформацію про існуючі збірки.
- Інформацію про характеристики комплектуючих збірки.

Доступ до цієї інформації відкритий для всіх користувачів
Діаграму дій для даного бізнес-процесу зображено на рисунку 2.1.



Рисунок 2.1 – Діаграма дій бізнес-процесу «інформацією про збірки».

Робота з підбором збірки - надає можливість користувачу використовувати підбір збірок по вибраним цільовим програмам, а саме:

- вибирати критерії підбору збірки через Мінімальні або Рекомендовані характеристики цільової програми.
- Виконувати підбір збірки по заданим параметрам.

Діаграма дій даного бізнес-процесу зображена на рисунку 2.2.

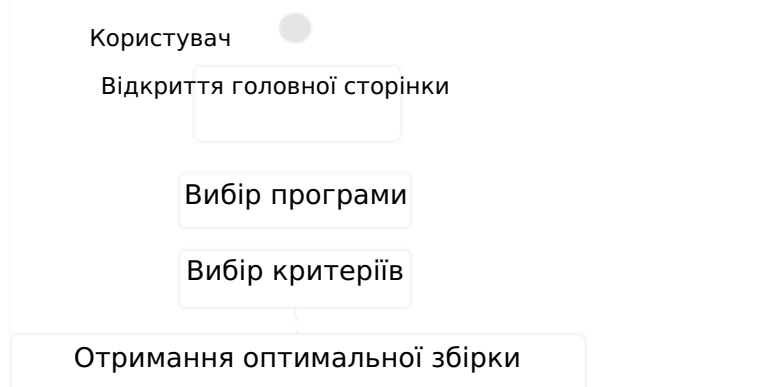


Рисунок 2.2 – Діаграма дій бізнес-процесу «Робота з підбором збірок».

Робота з адмін-панеллю надає можливість працівнику інформаційної системи (адміністратору) додавати та редагувати інформацію про збірки та програми. Він повинен мати можливість:

- Заходити в адмін-панель за допомогою логіну та пароля.
- Переглядати існуючі збірки та програми.
- Додавати нові збірки та програми.
- Редагувати збірки та програми.
- Видаляти збірки та програми.

Діаграма дій даного бізнес-процесу зображена на рисунку 2.3.

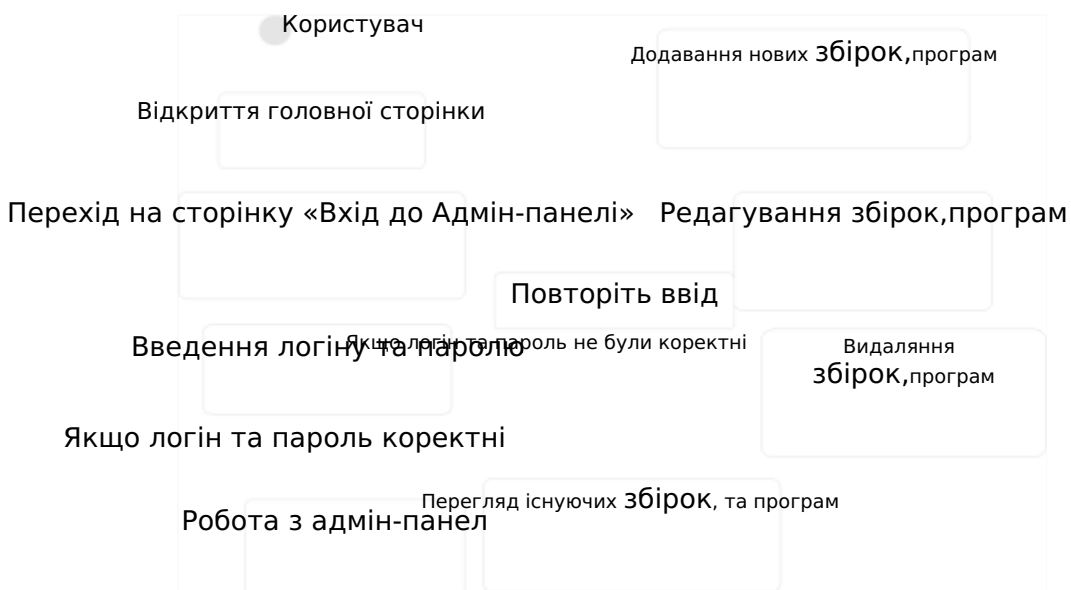


Рисунок 2.3 – Діаграма дій бізнес-процесу «Робота з адмін-панеллю».

Робота з вибором програм для підбору - надає можливість користувачу інформаційної системи вибирати цільові програми для підбору збірок.

Алгоритм дій користувача:

- Користувач вибирає підбір товарів по програмі
- З доступних програм користувач вибирає потрібну.
- Користувач бачить рекомендовані характеристики для програми.
- Виконується підбір збірок по програмі.

Робота з інформацією про програми – надає можливість користувачу інформаційної системи отримувати інформацію про програми.

Алгоритм дій користувача:

- Користувач переходить на сторінку «Програми».
- З доступних програм користувач вибирає потрібну.
- Користувач переглядає інформацію про програму.

На початку роботи з системою користувачу виводиться головна сторінка, де він може вибрати цільові програми для підбору збірок. Звідси він може перейти на сторінку «Збірки» та «Програми», де відображається інформація про збірки, та програми відповідно. Також з стартової сторінки адміністратор може зайти в адмін панель.

На рисунку 2.1 зображена UML-діаграма сайту, що зображує описані бізнес-процеси.



Рисунок 2.4 – UML-діаграма сайту.

Для збереження даних по розглянутих процесах та послідовності дій користувача системи, необхідно використати БД MSSQL.

2.2 Розробка структури інформаційної системи

Користувачі заносяться в таблицю «users» де є логін, пароль, і статус адміна. (таблиця 2.1)

Програми заносяться в таблицю «programs», де є назва, тип, опис, зображення, розробник, мінімальна частота процесора, рекомендована частота процесора, мінімальна кількість ядер, рекомендована кількість ядер, мінімальний об'єм оперативної пам'яті, рекомендований об'єм оперативної пам'яті, мінімальний об'єм постійної пам'яті, рекомендований об'єм постійної пам'яті, мінімальний об'єм візуальної пам'яті, рекомендований об'єм візуальної пам'яті, мінімальна частота пам'яті відеокарти, рекомендована частота пам'яті відеокарти. (таблиця 2.2) .

Збірки заносяться в таблицю «setups», де є назва, опис, середня ціна, частота процесора, кількість ядер процесора, об'єм оперативної пам'яті, необхідний об'єм диску, Об'єм відеопам'яті, Частота відео пам'яті, Назва процесору, назва оперативної пам'яті, назва Відеокарти. (таблиця 2.3)

Таблиця 2.1 – Атрибути таблиці «User»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	login	varchar	Логін
3	password	varchar	Пароль
4	Role	bool	Чи Адмін

Таблиця 2.2 – Атрибути таблиці «programs»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	Name	varchar	Назва

3	Image	varchar	зображення
4	Description	varchar	Опис
5	Category	varchar	Категорія
6	Developer	varchar	Розробник
7	ProcessorMin	float	мінімальна частота процесора
8	ProcessorRec	float	рекомендована частота процесора
9	ProcessorCoreMin	float	мінімальна кількість ядер
10	ProcessorCoreRec	float	рекомендована кількість ядер
11	RamMin	float	мінімальний об'єм оперативної пам'яті
12	RamRec	float	рекомендований об'єм оперативної пам'яті
13	StorageMin	float	мінімальний об'єм постійної пам'яті
14	StorageRec	float	рекомендований об'єм постійної пам'яті
15	VideocardMin	float	мінімальний об'єм візуальної пам'яті
16	VideocardRec	float	рекомендований об'єм візуальної пам'яті
17	VideocardFreqMin	float	мінімальна частота пам'яті відеокарти
18	VideocardFreqRec	float	рекомендована частота пам'яті відеокарти

Таблиця 2.3 – Атрибути таблиці «Setup»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	Name	varchar	Назва
3	Description	varchar	Опис
4	AvgPrice	float	середня ціна
5	Processor	float	частота процесора
6	ProcessorCore	float	кількість ядер процесора
7	RAM	float	об'єм оперативної пам'яті

7	Storage	float	необхідний об'єм диску
	Videocard	float	Об'єм відеопам'яті
	VideocardFreq	float	Частота відео пам'яті
	NameProcessor	varchar	Назва процесору
	NameRAM	varchar	назва оперативної пам'яті
	NameVideocard	varchar	назва Відеокарти

Отже, зазначений обсяг таблиць є достатнім для обробки всіх необхідних даних для роботи з ігровим інтернет - магазином.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

C# — це мова програмування що повноправно може вважатися об'єктно-орієнтованою, також вона має вбудовану систему безпечної типізації .NET. Мова належить компанії Microsoft.[6]

Синтаксис мови C# є близьким C, C++, але не лише являється розвитком цих мов, також він ввібрав в себе елементи мов Java і Pascal. Так як мова C# є об'єктно-орієнтованою, вона має правило статичної типізації, підтримується поліморфізм, атрибути, перевантаження операторів, події, властивості, винятки, коментарі у форматі XML, вказівники на функції певних класів. Ця мова багато чого перейняла від своїх ідейних батьків, — а саме мов C++, C, Pascal, і Java — C#, але при цьому вирішила багато попередніх проблем, що були у попередників. Наприклад, уже недоступна така можливість множинне успадкування класів, що мала свої проблеми при деяких ситуаціях.[8]

Було обрано мову C# через її зручність, мій досвід в роботі з нею та фреймворк ASP.NET, який використовує цю мову, що я обрав для розробки проекту.

2.3.2 Вибір фреймворку

Для розробки проекту було вирішено обрати ASP.NET, тому що цей фреймворк являється частиною платформи Microsoft .NET, створений для роботи з мовою C#, та підходить до мого проекту.

На ASP.NET можлива розробка проектів з реалізацією шаблону Model-View-Controller (ASP.NET MVC Framework).

2.3.4 Вибір системи керування базами даних

Є величезна кількість різноманітних СКБД для створення систем обліку, наприклад MS SQL Server, MySQL, Oracle, і інші, але для проекту ASP.NET MVC краще всього підходить MS SQL Server.

Microsoft SQL Server – це СКБД від компанії Microsoft. Для функціонування цієї СКБД використовується мова Transact-SQL, що була створена спільно Microsoft та Sybase. Transact-SQL відповідає стандарту ANSI/ISO що визначає структуру SQL запитів та розширень. Microsoft SQL Server може використовуватися для обробки даних будь яких розмірів, і ідеально підходить до проекту ASP.NET MVC .[9]

Відповідно до переваг і недоліків різноманітних СКБД було обрано MS SQL Server, так як вона підходить під платформу .Net.

Розділ 3. Програмна реалізація

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок

Проект використовує фреймворк ASP.NET MVC, який реалізує шаблон проектування MVC. Суть цього шаблону полягає в розділенні програми на моделі, контролери та представлення. У Моделі реалізується бізнес-логіка програми. У представленні знаходиться вивід даних системи. Контролер реалізує зв'язок між моделями та представленнями.

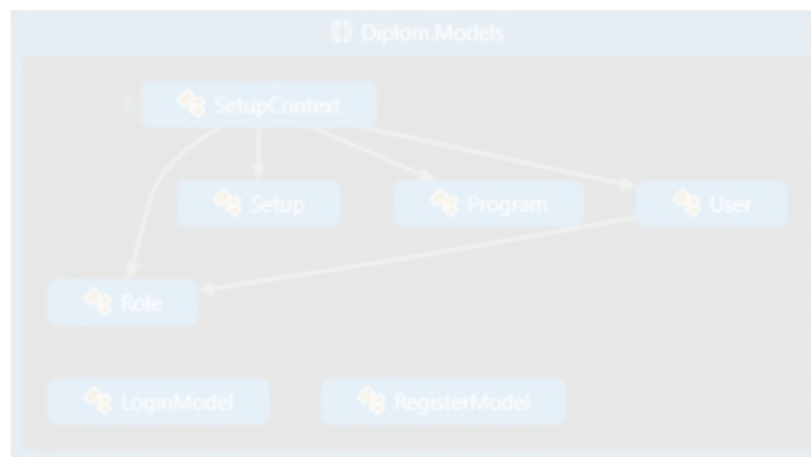


Рисунок 3.1 – Моделі проекту

Даний проект має наступні моделі (рисунок 3.1):

- AuthModels – модель, яка відповідає за бізнес логіку пов'язану з роботою системи авторизації. Включає в себе моделі LoginModel та RegisterModel;
- LoginModel – ця модель описує структуру класу, що використовується для виконання входу на сайт по зареєстрованому акаунту(рисунок 3.2);

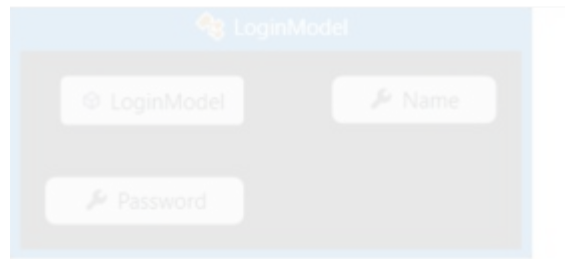


Рисунок 3.2 – LoginModel

- RegisterModel – модель, що описує структуру, що використовується для виконання реєстрації на сайт (рисунок 3.3);

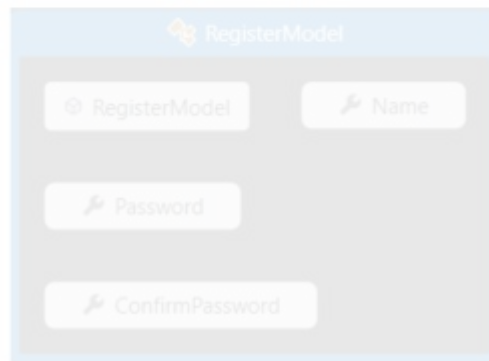


Рисунок 3.3 – RegisterModel

- Program – в цій моделі описується логіка і структура класу Program, І відповідає за бізнес логіку пов'язану з цільовими програмами для

підбору(рисунок 3.4);

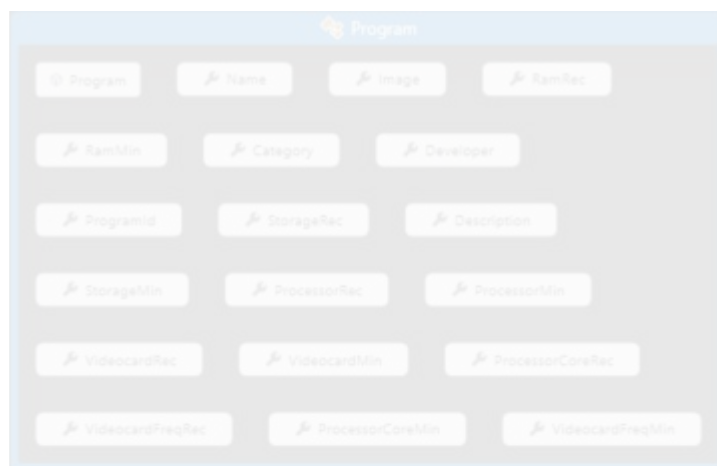


Рисунок 3.4 – Program

- User – є моделлю, відповідаючою за бізнес логіку, пов’язану з функціоналом користувачів (рисунок 3.5);



Рисунок 3.5 – User

- Setup – є моделлю, відповідаючою за структуру елементів збірки комп'ютерів та бізнес логіку пов'язану з ними (рисунок 3.6);

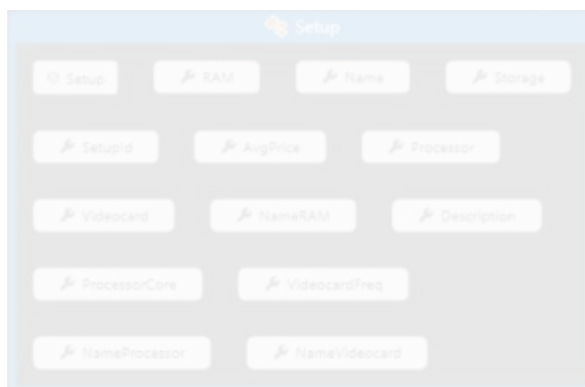


Рисунок 3.6 – Setup

- SetupContext – є моделлю – контекстом даних для зв'язування інших моделей проекту з базою даних (рисунок 3.7);



Рисунок 3.7 – Setup

В проекті є наступні представлення:

Account – група представлень, що відповідають за управління акаунтом:

- register - Відповідає за реєстрацію нового користувача, введення необхідних даних;
- login - Відповідає за вхід на сайт під зареєстрованим акаунтом;

Admin - група представлень, що відповідають за функціонування панелі Адміністратора:

- AddProgram – Відповідає за створення нового екземпляру класу Program, з введенням всіх необхідних даних;
- AddSetup – Відповідає за створення нового екземпляру класу Setup, та введенням всіх необхідних даних;
- DeleteProgram - Відповідає за видалення об'єкту програми;
- DeleteSetup - Відповідає за видалення об'єкту Збірки;
- DetailsPrograms - Відповідає за відображення усіх параметрів об'єкту програми;
- DetailsSetups – Відповідає за відображення усіх параметрів об'єкту збірки;
- EditProgram – Відповідає за редагування об'єкту класу програми, відображає попередні параметри об'єкту;
- EditSetup – Відповідає за редагування об'єкту класу збірки, відображає попередні параметри об'єкту;
- Index – Відповідає за відображення головної сторінки адмін-панелі
- Programs – Відповідає за відображення всіх елементів класу програми в адмін-панелі;
- Setups – Відповідає за відображення всіх елементів класу збірки в адмін-панелі;

Home - група представлень, що відповідають за функціонування основної функціональності інформаційної системи:

- DetailsPrograms – Відповідає за відображення параметрів об'єкту програми, що доступні користувачу;
- DetailsSetups – Відповідає за відображення параметрів об'єкту збірки, що доступні користувачу;
- Find – Відповідає за відображення результату підбору збірки;
- Index – Відповідає за відображення головної сторінки сайту, з формою для проведення знаходження оптимальної збірки під обрану цільову програму;
- Programs – Відповідає за відображення всіх елементів класу Program за допомогою карток;
- Setups – Відповідає за відображення всіх елементів класу Setup за допомогою карток;
- UserLogin – Відповідає за відображення панелі із входом/реєстрацією та відображенням поточного користувача;
- Layout – майстер – сторінка, навігаційна сторінка, використовується всіма іншими сторінками;



Рисунок 3.8 – Контролери

Також у системі є наступні контролери(рисунок 3.8):

- HomeController – Відповідає за основні функції інформаційної системи, вивід інформації про Програми та Збірки, та підбір оптимальної збірки під обрану цільову програму (рисунок 3.9);



Рисунок 3.9 – методи контролеру HomeController

- AdminController – Відповідає за роботу функціоналу адміністратора: занесення в базу даних нових даних про програми та збірки, їх зміна та видалення (рисунок 3.10);



Рисунок 3.10 – методи контролеру AdminController

- AccountController – Відповідає за функціонування системи авторизації, створення акаунту та входу в акаунт (рисунок 3.11);

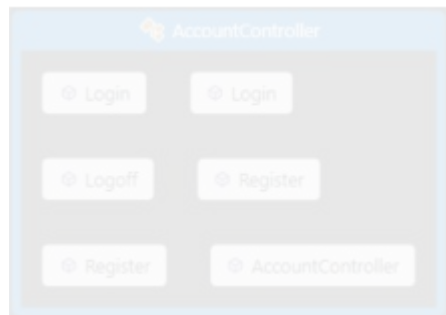
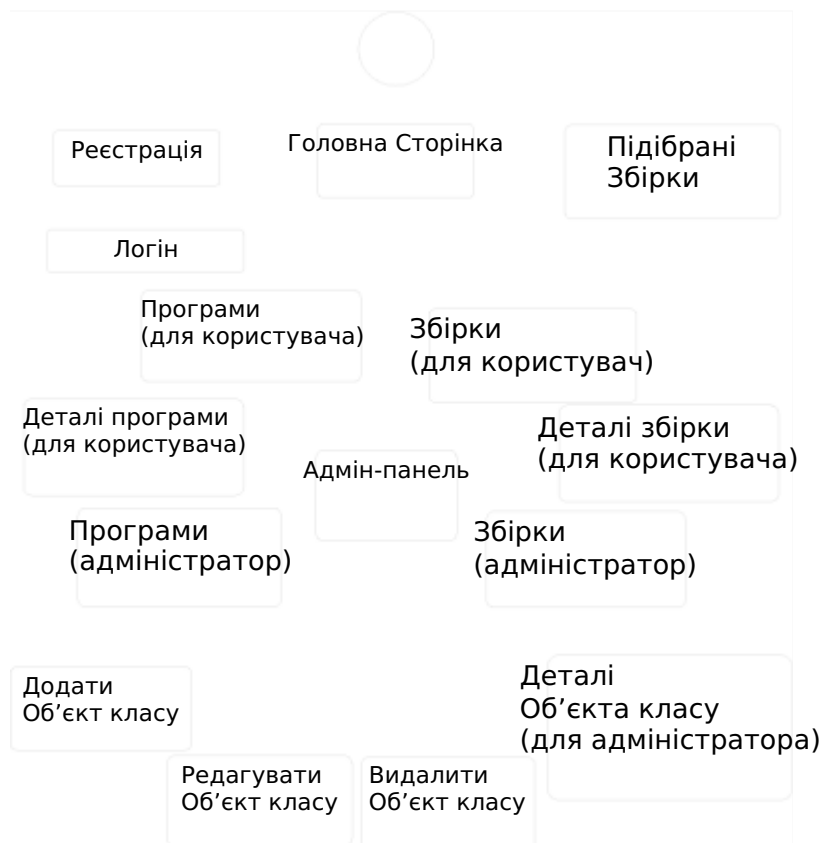


Рисунок 3.11 – методи контролеру AccountController

3.2 Розробка програмних модулів

На рисунку 3.12 зображена Функціональна схема розробленої інформаційної системи.

Рисунок 3.12 – Функціональна схема інформаційної системи



Діаграма класів контролерів на рисунку 3.13. Методи цих контролерів відповідають за функціонування відповідних View сторінок.

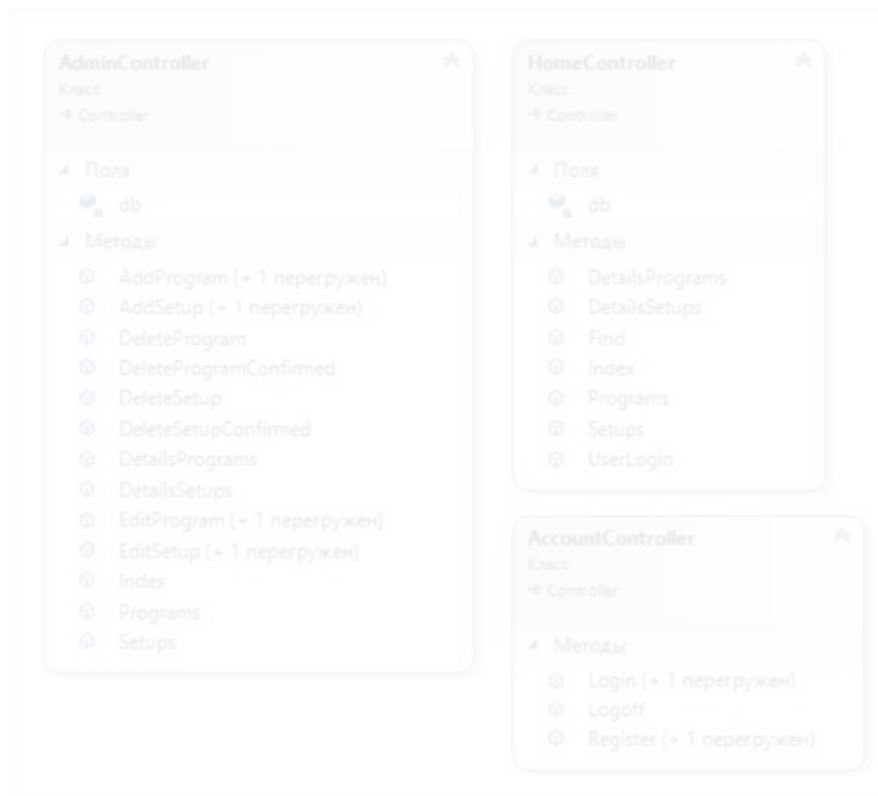


Рисунок 3.13 – Діаграма класів контролерів

Основний функціонал проекту реалізований в класі HomeController – А саме в методах Index та Find. Метод Index реалізує роботу головної сторінки та передає данні про доступні цільові програми. Він дає запит бази даних на отримання списку всіх елементів Program, та передає їх у представлення через ViewBag. Сам підбір збірок реалізує метод Find, що бере список збірок з бази та в залежності від обраного типу підбору (Мінімальний чи рекомендований) відбирає підходящі збірки. З підходящих збірок формується список, який сортується по ціні.

Таким чином клієнту видається список збірок, який задовольняє вибрані критерії, починаючи з самого дешевого варіанту. Клієнт має можливість обрати або самий дешевий підходящий варіант, або переглянути кращі варіанти залежно від бюджету. Робота методу Find зображена на рисунку 3.14. Фрагмент коду із алгоритмом:

```
if(type.Equals("Minimal"))
{
    foreach (Setup set in setups)
    {
        if(set.Processor >= a.ProcessorMin && set.ProcessorCore >=
            a.ProcessorCoreMin && set.RAM >= a.RamMin
            && set.Videocard >= a.VideocardMin && set.VideocardFreq >=
            a.VideocardFreqMin)
        {
            SortedSetups.Add(set);
        }
    }
}
if(type.Equals("Recommended"))
{
    foreach (Setup set in setups)
    {
        if (set.Processor >= a.ProcessorRec && set.ProcessorCore >=
            a.ProcessorCoreRec && set.RAM >= a.RamRec
            && set.Videocard >= a.VideocardRec && set.VideocardFreq >=
            a.VideocardFreqRec)
        {
            SortedSetups.Add(set);
        }
    }
}
SortedSetups = SortedSetups.OrderBy(x => x.AvgPrice).ToList();
```

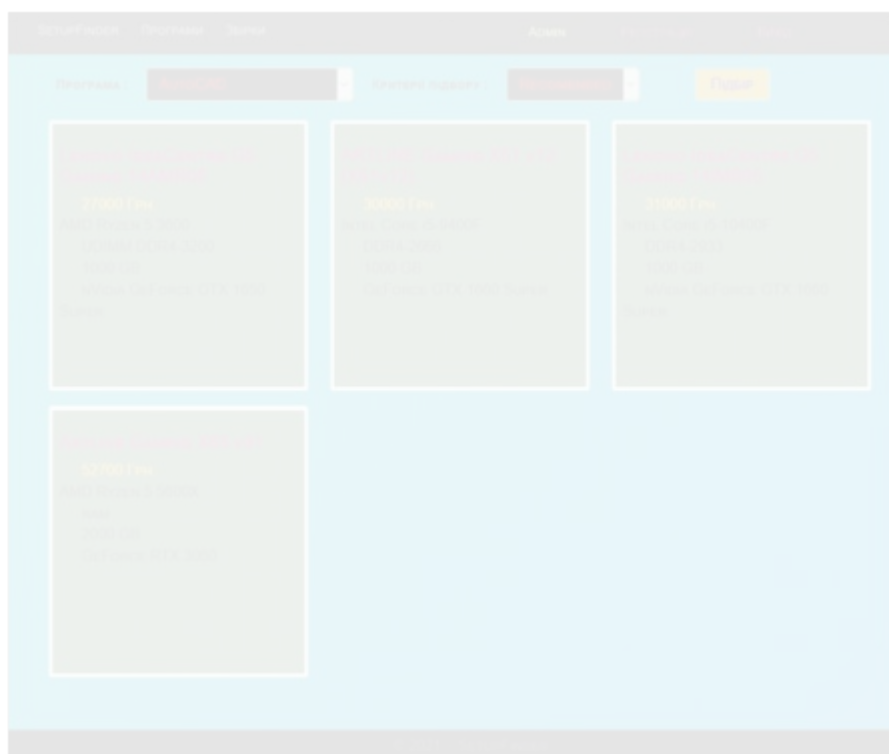


Рисунок 3.14 – робота методу Find

AccountController – Відповідає за функціонування системи авторизації, створення акаунту та входу в акаунт. Цей функціонал реалізований в методах Login, Register та Logout. Інформація про користувача передається в базу, і за замовчанням задає йому роль користувача. Роль адміністратора може задаватись лише напряму в базу даних, для безпеки. Робота методу Register зображена на рисунку 3.15.



Рисунок 3.15 – робота методу Register

AdminController – Відповідає за роботу функціоналу адміністратора: занесення в базу даних нових даних про програми та збірки, їх зміна та видалення. У цьому контролері є велика кількість різноманітних методів, що реалізують роботу необхідних операцій, так відображення існуючих програм та збірок реалізовані методами Programs та Setups відповідно, додавання нових елементів реалізовані методами AddProgram , AddSetups, редагуються існуючі елементи методами Edit, видаляються в методах Delete.

Деталі зі всією інформацією про елементи моделі можна отримати з допомогою методу Details.

Роботу методу Programs можна побачити на рисунку 3.16



Рисунок 3.16 – робота методу Programs

3.3 Тестування інформаційної системи

Проведемо тестування різноманітних можливостей і функцій інформаційної системи, з допомогою тест-кейсів на таблицях.

Перевірка працездатності інтерфейсу користувача на головній сторінці.

Author:	Specification ID:	Priority:	Producer:	Developer:
Яцунь І.С	1	1	S.Petrovsky	Яцунь І.С
OVERVIEW:				
Даний тест комплект перевіряє працездатність інтерфейсу користувача				
GLOBAL SETUP AND ADDITIONAL INFO:				
1. Запустити роботу сервера				
2. Зайти на сайт http://localhost:52735/Index				
3. Головна сторінка				
TC ID/Priority		01	1	
IDEA: Правильне функціонування кнопок навігації сайту				
SETUP AND ADDITIONAL INFO:				
Головна сторінка містить навігацію для інших сторінок та частину інформації				

Revision History	
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші по черзі на всі елементи навігації.	Відкриваються відповідні сторінки з вмістом відповідним до назви елемента навігації і назви сторінки.

TC ID/Priority	02	1
IDEA: Правильне функціонування функції підбору		
SETUP AND ADDITIONAL INFO:		
При виборі цільової програми та критеріїв підбору система видає список збірок що підходять цим критеріям		

Revision History	
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Вибрати цільову програму, вибрати критерій, натиснути на кнопку Find	Виводиться список програм з характеристиками відповідними заданим критеріям, розсортовані від меншої ціни до більшої

Перевірка працездатності інтерфейсу користувача на конкретних сторінках сайту

Author:	Specification ID:	Priority:	Producer:	Developer:
Яцунь I.C	2	2	S.Petrovsky	Яцунь I.C

OVERVIEW:

Даний тест комплект перевіряє працездатність інтерфейсу користувача

GLOBAL SETUP AND ADDITIONAL INFO:

- Запустити роботу сервера
- Зайти на сайт <http://localhost:52735/>
- Головна сторінка

TC ID/Priority	03	2
IDEA: Правильне функціонування відображення списку Програм		
SETUP AND ADDITIONAL INFO:		
Сторінка програм відображає список програм		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Перейти на сторінку Програм	Список програм відображається правильно.

TC ID/Priority	04	2
IDEA: Правильне функціонування посилань на конкретну програму		
SETUP AND ADDITIONAL INFO:		
Сторінка програми містить деталі про програму		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на будь яку програму на сторінці програм	Посилання переводить користувача на сторінку програми, інформація про програму відображається коректною.

TC ID/Priority	05	2
IDEA: Правильне відображення списку Збірок		
SETUP AND ADDITIONAL INFO:		
Сторінка збірки відображає списки збірок		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку посилання Збірок	Кнопка переводить користувача на список доступних збірок

TC ID/Priority	06	2
IDEA: Правильне функціонування посилань на конкретну збірку		
SETUP AND ADDITIONAL INFO:		
Сторінка збірки містить деталі про збірки		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на будь яку збірку на сторінці збірок	Посилання переводить користувача на сторінку збірки, інформація про збірку відображається коректною.

Перевірка працездатності функцій та інтерфейсу адміністратора

Author: Яцунь I.C	Specification ID: 3	Priority: 3	Producer: S.Petrovsky	Developer: Яцунь I.C
OVERVIEW:				
Даний тест комплект перевіряє працездатність функцій та інтерфейсу адміністратора				
GLOBAL SETUP AND ADDITIONAL INFO:				
1. Запустити роботу сервера				
2. Зайти на сайт http://localhost:52735/Admin				
3. Сторінка Адміністратора				

TC ID/Priority	07	3
IDEA: Правильне функціонування сторінки Programs		
SETUP AND ADDITIONAL INFO:		
Сторінка програм відображає список програм для адміністратора		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	
Execution Part		
PROCEDURE	EXPECTED RESULT	
1. Натиснути лівою кнопкою миші на	Посилання переводить користувача на	

кнопку Programs	сторінку зі списком програм для адміністратора, в якому коректною відображається інформація про програми
-----------------	--

TC ID/Priority	08	3
IDEA: Правильне функціонування сторінки Setups		
SETUP AND ADDITIONAL INFO: Сторінка програм відображає список збірок для адміністратора		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Setups	Посилання переводить користувача на сторінку зі списком збірок для адміністратора, в якому коректною відображається інформація про збірок

TC ID/Priority	09	3
IDEA: Правильне функціонування сторінки AddProgram		
SETUP AND ADDITIONAL INFO: Сторінка AddProgram дозволяє створювати нові екземпляри програм		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	

Execution Part	
PROCEDURE	EXPECTED RESULT
1. Натиснути лівою кнопкою миші на кнопку Створити програму	У базі даних буде створено новий об'єкт моделі Program з введеними параметрами
2. Ввести коректні дані у відповідну форму	

TC ID/Priority	10	3
IDEA: Правильне функціонування сторінки AddSetup		
SETUP AND ADDITIONAL INFO: Сторінка AddSetup дозволяє створювати нові екземпляри збірки		

Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	
Execution Part		
PROCEDURE	EXPECTED RESULT	
1. Натиснути лівою кнопкою миші на кнопку Створити збірку	У базі даних буде створено новий об'єкт моделі Setup з введеними параметрами	
2. Ввести коректні дані у відповідну форму		
TC ID/Priority	11	3
IDEA: Правильне функціонування сторінки DeleteProgram		
SETUP AND ADDITIONAL INFO:		
Сторінка DeleteProgram дозволяє видалити об'єкт Program, id якого було передано методу.		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	
Execution Part		
PROCEDURE	EXPECTED RESULT	
1. Натиснути лівою кнопкою миші на кнопку Delete біля необхідного екземпляра моделі	Цей елемент буде видалено з бази даних	
TC ID/Priority	12	3
IDEA: Правильне функціонування сторінки EditProgram		
SETUP AND ADDITIONAL INFO:		
Сторінка EditProgram дозволяє змінити параметри об'єкту Program, id якого було передано методу.		
Revision History		
Created on: 09.05.2021 by Яцунь I.C	Новий тест-кейс	
Execution Part		
PROCEDURE	EXPECTED RESULT	
1. Нажати на кнопку Edit у відповідного елементу моделі.	Параметри елементу були змінені	
2. ввести деякі зміни до параметрів елементу		

TC ID/Priority	13	3
IDEA: Правильне функціонування сторінки EditSetup		
SETUP AND ADDITIONAL INFO:		
Сторінка EditSetup дозволяє змінити параметри об'єкту Setup, id якого було передано методу.		
Revision History		
Created on: 09.05.2021 by Яцунь І.С	Новий тест-кейс	
Execution Part		
PROCEDURE	EXPECTED RESULT	
1. Нажати на кнопку Edit у відповідного елемента моделі.	Параметри елемента були змінені	
2. Ввести деякі зміни до параметрів елемента		

3.4 Інструкція користувача

Перше, що бачить користувач, зайшовши на сайт – це стартова сторінка, зображена на рисунку 3.17.



Рисунок 3.17 – Стартова сторінка

Далі користувач може зайти сторінку зі списком програм та Списком Збірок, що є у інформаційній системі Ці сторінки зображені на рисунках 3.18 та 3.19.

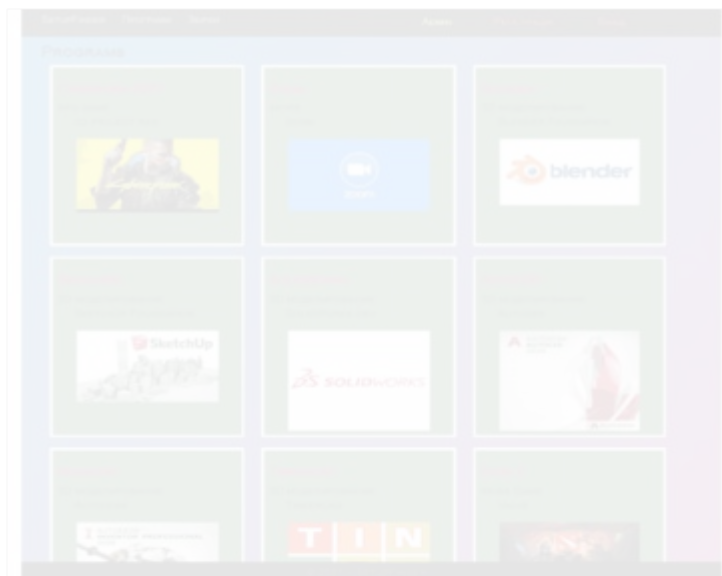


Рисунок 3.18 – Сторінка Programs

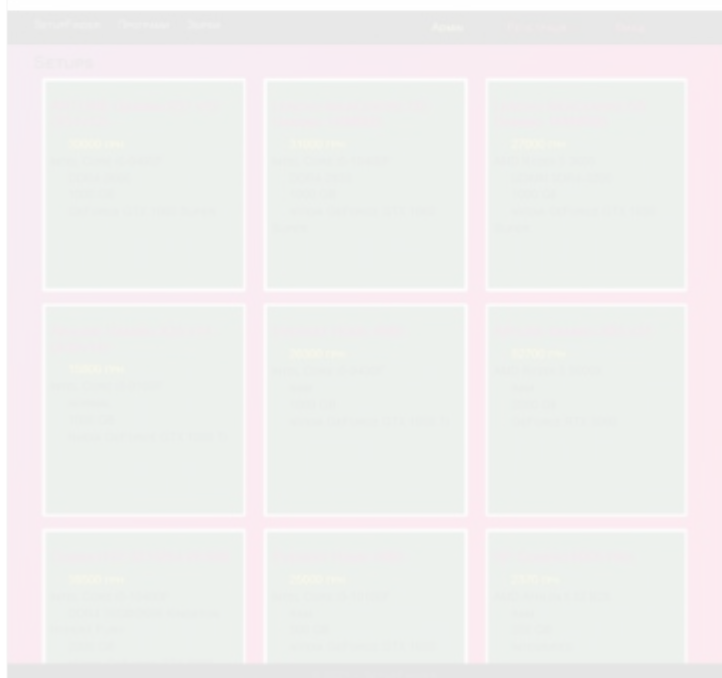


Рисунок 3.19 – Сторінка Setups

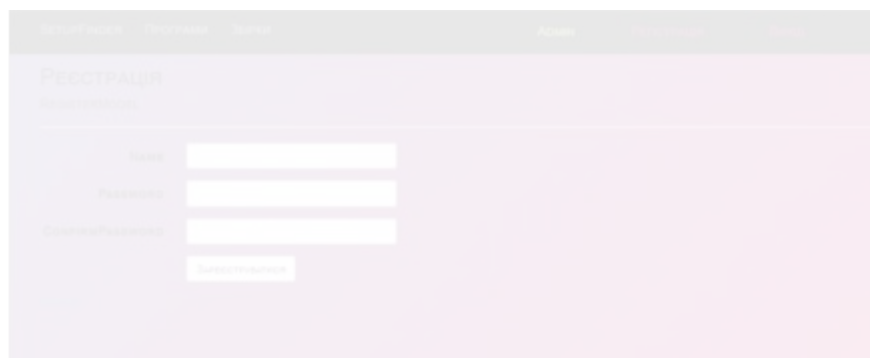


Рисунок 3.21 – Сторінка Реєстрації

Сторінка з логіном для користувача зображена на рисунку 3.22



Рисунок 3.22 – Логін

Основна функція інформаційної системи – підбір збірки по цільовій програмі. Користувач може провести цю операцію перейшовши на головну сторінку (рисунок 3.17), вибравши цільову програму та критерій підбору (мінімальні або рекомендовані) та натиснувши кнопку Filter. Результат цієї

45

операції зображений на рисунку 3.23

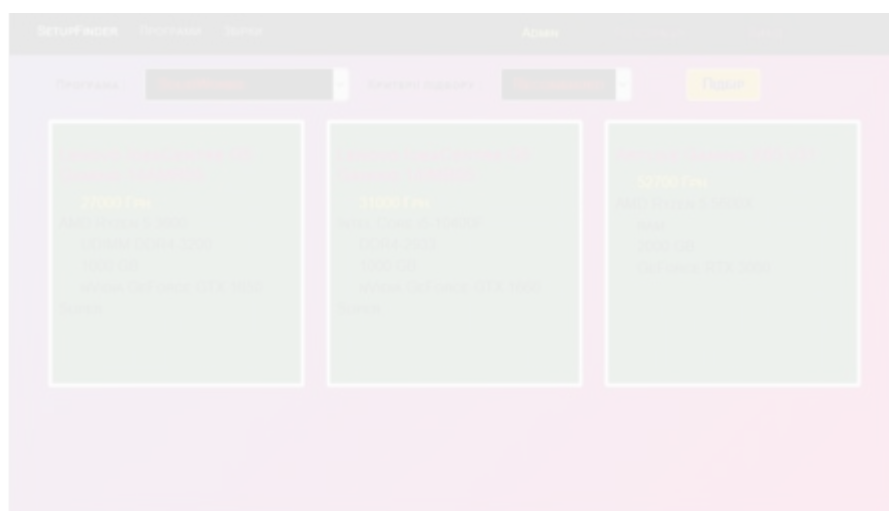


Рисунок 3.23 – Виконаний підбір збірки

Отже, користування інформаційною системою є інтуїтивно зрозумілим та не вимагає додаткових знань, умінь та навичок, окрім необхідних для звичайного користування комп'ютером.

3.5 Вимоги до апаратних та програмних засобів

Мінімальні вимоги до браузера клієнта :

- підтримка HTML 5;
- підтримка CSS 3;

Мінімальні вимоги до серверу:

- сервер IIS.

- фреймворк ASP.NET MVC 5
- NET 3.5
- System.Web.Mvc
- System.Web.Routing
- System.Web.Abstractions
- Бібліотеки DLL маршрутизації та абстракцій (у випадку відсутності пакета SP1)

Мінімальні вимоги до СКБД:

- СКБД MSSQL SERVER 2017+

Додаткові вимоги до системи:

- для роботи вимагається підключення до Інтернету;
- не підходить Internet Explorer та інші застарілі браузерери або старі версії сучасних браузерів;
- сайт не створений для перегляду на мобільних пристроях або інших пристроях з маленьким розміром екрану.

-

Висновки

В результаті виконання даної кваліфікаційної роботи було створено інформаційну систему оптимального підбору збірок комп'ютера за вибраними цільовими програмами. Створена вона була за допомогою платформи .Net, фреймворку ASP.NET MVC і бази даних Microsoft SQL Server. Всі вимоги, поставлені до роботи, були виконані, а саме була створена база даних та застосунок, який її використовує.

Створена система виконує наступні функції:

- робота з відкритою інформацією (виведення інформації про цільові програми та збірки комплектуючих комп'ютера зареєстрованих в системі);
- робота з підбором оптимальної збірки з урахуванням цільової програми та критеріями підбору(мінімальні або рекомендовані);
- робота з панеллю адміністратора (додавання цільових програм та збірок комп'ютерних комплектуючих, редагування та видалення цієї інформації);
- робота з особистими даними (реєстрація в системі, авторизація);
- робота з внутрішніми даними системи (внутрішня інформація про цільові програми, та збірки комп'ютера);

Дана система може бути використана для підбору оптимальної збірки комп'ютера за певною програмою, що вважається цільовою для оптимізації роботи ПК. Її можна вдосконалити, додавши можливість пошуку, сортування програм та збірок, удосконаливши алгоритм підбору, удосконаливши інтерфейс та зробивши його більш привабливим, а також швидшим.

Перелік посилань

1. Hotline – Режим доступу: <https://hotline.ua/>
2. E-Katalog – Режим доступу: <https://ek.ua/>
3. Wikipedia. – API <https://ru.wikipedia.org/wiki/API>
4. Головний сайт платформи .NET – <http://www.microsoft.com/net/dd>
5. Головний сайт платформи PHP – <http://www.php.net/>
6. Апаратне забезпечення ПК <https://bondarenko.dn.ua/aparatne-zabezpechennya-pk/>
7. Найкращі backend фреймворки в 2020 році <https://top.studio/stati/luchshie-backend-frejmvorki-v-2020-godu/>
8. C# документація <https://docs.microsoft.com/en-us/dotnet/csharp/>
9. Microsoft SQL Server https://uk.wikipedia.org/wiki/Microsoft_SQL_Server

13	Студентська робота	ID файлу: 1005677564	Навчальний заклад: Khmelnytskyi National University	0.35%
21	Студентська робота	ID файлу: 1005411484	Навчальний заклад: Khmelnytskyi National University	0.14%

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система оптимального підбору товарів

Автор: студент групи КН-17-2 Яцунь Іван Сергійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: доцент кафедри КНІТ Петровський С.С.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 7,4 % і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



С.С. Петровський

Гарант ОП



О. В. Мазурець

Завідувач кафедри КНІТ



О. В. Бармак

РЕЦЕНЗІЯ
на кваліфікаційну роботу бакалавра

Студента: групи КН-17-2 Яцуня Івана Сергійовича

За темою: Система оптимального підбору товарів

1. Актуальність і значення теми: Тему підбору збірок комплектуючих комп'ютера можна вважати актуальною, так як існує велика кількість можливих варіантів збірок комп'ютера, а також багато різних необхідних характеристик в залежності від цільової програми.

2. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: Запропоновані моделі, підходи та алгоритми виконують свою роботу, але є малофункціональними. Є багато можливостей для вдосконалення. Засоби розробки інформаційної системи є актуальними та підходять для виконання поставленої задачі.

3. Оцінка розробленої інформаційної системи, її практичність та економічна доцільність: Розроблена система виконує свою задачу та працює стабільно, але її функціонал обмежений. Систему можна використовувати практично та вона є економічно доцільною, не потребує додаткових затрат при використанні.

4. Загальний висновок та оцінка: Значних проблем у системі не виявлено, вона виконує свою роботу та більшість необхідних функцій, відповідає темі. Недостатками можна вважати обмежений функціонал та низьку варіативність роботи, є можливості для покращення роботи системи.

Робота заслуговує на оцінку « добре »

Рецензент к. ер.-м. н., доц. Яценко Н.О. Дату 08.06.2014