

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ „ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра ІСМ

КУРСОВА РОБОТА

з дисципліни « Проектування інформаційних систем »

НА ТЕМУ :

« Система підбору оптимальної конфігурації комп'ютера »

Виконав студент групи КН-311

Гадзало Олег

Оцінка	Балів	Дата



Номер залікової книжки

17080633

Керівник проекту

доцент кафедри ІСМ Басюк Т.М.

Львів 2020

## ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ

Завдання до виконання	Дата
Анотація	16.02.2020
Вступ	06.05.2020
Розділ 1. Аналіз літературних джерел та предметної галузі	16.02.2020
1.1. Основні засади дослідження	16.02.2020
1.2. Аналіз відомих засобів вирішення проблеми	16.02.2020
1.3. Функціональність програмної системи	16.02.2020
Висновок до першого розділу	16.02.2020
Розділ 2. Проектування системи	16.02.2020
2.1. Опис системних вимог згідно з методологією RUP	16.02.2020
2.2. Деталізація функцій та аналіз категорій користувачів	16.02.2020
2.2.1. Діаграма варіантів використання	16.02.2020
2.2.2. Діаграма класів	16.02.2020
2.2.3. Діаграми кооперації	16.02.2020
2.2.4. Діаграма послідовності	16.02.2020
2.2.5. Діаграма діяльності	16.02.2020
2.2.6. Діаграма компонентів	16.02.2020
Висновок до другого розділу	16.02.2020
Розділ 3. Вибір засобів реалізації та конструювання системи	06.05.2020
3.1. Вибір програмного рішення	06.05.2020
3.2. Аналіз контрольного прикладу	06.05.2020
Висновок до третього розділу	06.05.2020
Висновки	06.05.2020
Список використаних джерел	16.02.2020
Додатки	06.05.2020

Завдання прийнято до виконання



(підпис студента)

Керівник проекту

к.т.н., доц. Басюк Т.М.

## АНОТАЦІЯ

В курсовій роботі розглянуто розробку системи підбору оптимальної конфігурації комп'ютера. Досліджено актуальність створення даної системи. Проведено аналіз основних аналогів системи з виділенням їхніх плюсів і мінусів для визначення необхідного функціоналу програмної системи з її атрибутами. Також розроблений розгорнутий текстовий опис основних системних вимог до заданої системи за допомогою технології RUP. Було проведено аналіз вимог та опис функцій системи та сформувалось відповідне їх представлення. Було створено програмний продукт, за допомогою якого буде вирішено дану проблематику та визначені задачі.

## ЗМІСТ

Розділ 1. Аналіз літературних джерел та предметної галузі.....	7
1.1. Основні засади дослідження.....	7
1.2. Аналіз відомих засобів вирішення проблеми.....	16
1.3. Функціональність програмної системи.....	19
Висновок до першого розділу.....	23
Розділ 2. Проектування системи.....	24
2.1. Опис системних вимог згідно з методологією RUP.....	24
2.2. Деталізація функцій та аналіз категорій користувачів.....	26
2.2.1. Діаграма варіантів використання.....	26
2.2.2. Діаграма класів.....	27
2.2.3. Діаграми кооперації.....	28
2.2.4. Діаграма послідовності.....	29
2.2.5. Діаграма діяльності.....	30
2.2.6. Діаграма компонентів.....	31
Висновок до другого розділу.....	32
Розділ 3. Вибір засобів реалізації та конструювання системи.....	33
3.1. Вибір програмного рішення.....	33
3.2. Аналіз контрольного прикладу.....	35
Висновок до третього розділу.....	39
Висновки.....	40
Список використаних джерел.....	41
Додатки.....	42

## ВСТУП

**Актуальність дослідження.** Існує декілька способів отримати новий комп'ютер. Самий простий з них – придбання фабричної збірки. Комп'ютер, зібраний виробником, як правило, характеризується відмінною збіркою, надійністю та тихою роботою.

Ті, хто купує комп'ютер не перший раз, віддають перевагу збірці пристрою з комплектуючих.

- Самостійна збірка. Але на сьогоднішній день не дозволяє економити на вартості та досягти більш високої продуктивності пристрою. Якщо Ви володієте відповідними знаннями та навичками. Причому, це повинні бути не поверхневі знання про те, які деталі забезпечують роботу ПК, а професійні знання про характеристики, котрими повинна володіти кожна комплектуюча, про сумісність тих чи інших деталей, і впевнене володіння навиками збірки, виключаючи можливість мимовільного пошкодження електронних схем та інших деталей. Інакше, ви не лише ризикуєте переплатити, кілька разів купляючи ті чи інші комплектуючі, але і навряд чи зможете забезпечити надійну продуктивну роботу комп'ютера, чого можна досягнути лише зібравши максимально збалансовану систему.
- Професійна збірка на ваше замовлення. Самий підходящий варіант в тому випадку, якщо ви самі не володієте необхідними навичками. Але при цьому все ж необхідно бути гранично обережним, довіряти збірку слід людині або фірмі, які користуються гарною репутацією. Ви повинні чітко пояснити збирачеві, які обов'язки будуть покладатися на майбутній комп'ютер. щоб він розумів, які характеристики пристрою є основоположними.

**Об'єкт дослідження.** Підбір та визначення комплектацій персонального комп'ютера, на вимогу користувача за заданими параметрами.

**Мета роботи.** Покращити існуючі рішення даної проблеми, визначити їх позитивні та негативні сторони. Сконструювати модель кращого вирішення даної проблематики на основі дослідженого матеріалу.

**Практичне значення.** Практичне значення отриманих результатів полягає у розробці етапів аналізу й оцінки ефективності системи підбору оптимальної конфігурації комп'ютера, пропозицій відповідних критеріїв підвищення її ефективності.

## **Розділ 1. Аналіз літературних джерел та предметної галузі**

### **1.1. Основні засади дослідження**

У сьогоднішніх реаліях важко уявити життя без машин, зокрема комп'ютера, до яких вже прив'язалось людство і стало залежними від нього. Існує цілий ряд переваг того, що надає нам комп'ютер. По-перше, це значно спрощує організацію охорони офісів, підприємств, компаній, житлових комплексів або стоянок. Якщо раніше необхідно було на кожному поверсі або по всьому периметру розставляти охорону для того, щоб постійно спостерігати за подіями, то сьогодні завдяки комп'ютеризації, досить просто встановити камери спостереження і все, що відбувається спостерігати на моніторі комп'ютера. У виробництві, наприклад, комп'ютери використовуються від проектування деталей до конструювання цілого виробу, його виробництва і продажу. Широко використовуються системи автоматизованого проектування архітектурних споруд: будівель, мостів, гребель. Дорогі експерименти в науці і техніці можна повністю замінити розрахунками на комп'ютері. [1]

По-друге, сучасні комп'ютери (ноутбуки, нетбуки) можна без проблем брати з собою у відрядження, на відпочинок або просто на прогулянку в парк. Їх компактні розміри, легка вага і можливість працювати декілька годин без мережі, дають ще більше свободи діловій людині. Це також дає можливість бути завжди на зв'язку. Адже з Вами завжди і електронна пошта, і Інтернет, і всі офісні додатки. Більше того, вся інформація, яка може знадобитися в відрядженні, вміщається просто в невеликому комп'ютері.

Комп'ютери бувають різні: для офісної праці, для ігор, ноутбуки, нетбуки і так далі. І усі вони складаються так чи інакше з різних складових, комплектуючих і мають різні типи конфігурацій. Розглянемо наступним чином, з чого саме складається персональний комп'ютер.

### **Материнська плата (CPU)**

Це текстолитова пластина з припаяними до неї мікросхемами і роз'ємами виконує складальну функцію, об'єднуючи всі інші елементи комп'ютера. Завдяки їй організовуються всі складні процеси і виконуються завдання. Навіть комп'ютерні миша і клавіатура працюють так, як вони працюють, тому що обмінюються інформацією з іншими пристроями через системну плату. Працездатність всього комп'ютера залежить від неї. Та й швидкість — теж. Тому дуже важливо при зборі комп'ютера враховувати пропускну здатність шини системної плати. [2]

Головні елементи материнської плати:

- Чіпсет (набір мікросхем, сполучний компонент для інших елементів);
- Північний міст (поєднує процесор з іншими компонентами);
- Південний міст (підключає компоненти, яким не потрібна висока швидкість)
- BIOS (мікросхема зі стартовим ПЗ для запуску операційної системи)



Рис.1.1 Материнська плата



Положення при установці, кількість пристроїв, що підключаються, тип роз'ємів і багато іншого визначається форматом системної плати. Материнські плати бувають різних форматів. Ось найпоширеніші:

- Mini ITX.
- Micro ATX (mATX).
- ATX.

Найкомпактніша плата — Mini ITX, йде з інтегрованим процесором, рідко коли використовується при самостійному зборі комп'ютера. Наступна за розміром — mATX. Відмінна плата для офісного або домашнього робочого комп'ютера. ATX — найбільша і функціональна плата, до неї можна підключити набагато більше пристроїв. Підходить для професійних робочих комп'ютерів (для дизайну, програмування, роботи з відео і інших занять) і ігрових системників. Всі дані між компонентами, встановленими на материнській платі, повинні якось передаватися, щоб комп'ютер взагалі функціонував. Для цього і використовуються шини — групи провідників, по яких пересилаються команди від одного компонента до іншого. [3]

### **Центральний Процесор (GPU)**

Процесором називається пристрій, що здатен обробляти програмний код і визначати основні функції комп'ютера з обробки інформації, тобто процесор виконує основні процеси в комп'ютері.

Конструктивно процесори можуть виконуватися як у вигляді однієї великої монокристалічної інтегральної мікросхеми — чіпа, так і у вигляді декількох мікросхем, блоків електронних плат і пристроїв.

Найчастіше процесор має вигляд чіпа, розташованого на материнській платі. На самому чіпі написана його марка, його тактова частота (кількість можливих операцій, які він може виконати за одиницю часу) і виробник.

Сьогодні мікропроцесори й процесори вміщують у собі мільйони транзисторів й інших елементів електронної логіки і становлять собою складні високотехнологічні електронні пристрої. Персональний комп'ютер містить у своєму складі чимало різних процесорів. Вони входять до складу систем

введення виведення контролерів пристроїв. Кожен пристрій, будь то відеокарта, системна шина чи ще щось, обслуговується своїм власним процесором або процесорами. Однак архітектуру і конструктивне виконання персонального комп'ютера визначає процесор або процесори, що контролюють і обслуговують системну шину й оперативну пам'ять, і, що більш важливо, виконують об'єктний код програм. Такі процесори прийнято називати центральними, або головними процесорами (Central Point Unit — CPU). На основі архітектури центральних процесорів будується архітектура материнських плат і проектується архітектура й конструкція комп'ютера.

**Центральний процесор (ЦП)** — це основний компонент сучасного комп'ютера, призначений для керування всіма його пристроями та виконання арифметичних і логічних операцій над ними.

ЦП — це компактний модуль (розміром близько 5 x 5 x 0,3 см), що вставляється в гніздо на материнській платі. Напівпровідниковий кристал сучасного процесора містить мільйони чи навіть мільярди транзисторів. Процесор вставляється в спеціальне місце — сокет на системній (материнській) платі, яка, у свою чергу, розміщується в системному блоці. Для охолодження процесорів застосовують кулери — пристрої, які складаються з вентилятора і радіатора. На процесор встановлюють радіатор (зазвичай з алюмінію чи міді), а на нього — вентилятор, що забезпечує притік повітря до радіатора.[4]



Рис.1.2 Сучасний ЦП Intel core i7 2600K

## Відеокарта

**Відеокарта** (відома також як відеоадаптер, графічна плата, графічний адаптер, графічна карта) - важлива і дуже складна складова частина комп'ютера. Сучасні відеокарти є свого роду спеціалізованими комп'ютерами, що складаються з власного процесора, оперативної пам'яті, BIOS і інших компонентів, які за своєю структурою і організацією взаємодії пристосовані для максимально ефективного вирішення одного завдання - обробки і формування графічних даних, а також їх виведення на монітор.

Основними розробниками відеокарт є американська компанія **Nvidia** і канадська **ATI Technologies**, придбана в 2006 році американською компанією AMD. Відеокарти від Nvidia представлені брендом *GeForce*. Графічні плати ATI відомі всім під назвою *Radeon*.

### Сучасна графічна карта складається з наступних частин:

- **Графічний процесор** (графічне ядро GPU (Graphics processing unit - графічний процесорний пристрій) - процесор, що займається розрахунками та формуванням графічної інформації, що виводиться на монітор, є основою відеокарти і по своїй складності практично не поступається центральному процесору комп'ютера, а іноді і перевершує його;

- **Відеопам'ять** - виконує роль своєрідного буфера, в який тимчасово поміщаються зображення, що виводяться на монітор, створюються та постійно змінюються графічним ядром. У цей буфер поміщаються також елементи, необхідні для формування цих зображень;

- **Відеоконтролер** - відповідає за правильне формування і передачу потрібної інформації з відеопам'яті на RAMDAC.

- **RAMDAC** (Random Access Memory Digital-to-Analog Converter) або **цифро-аналоговий перетворювач (ЦАП)** - пристрій, що здійснює перетворення цифрових результатів роботи відеокарти в аналоговий сигнал, який відображається на моніторі. Можливостями цього пристрою визначається кількість відображуваних кольорів, насиченість картини та ін. Цифрові монітори, проектори та інші пристрої, які підключаються до

цифрових роз'ємів відеокарти, використовують власні цифро-аналогові перетворювачі і від RAMDAC відеокарти не залежать;

- **Відео-ПЗУ (Video ROM)** - мікросхема, що містить в собі базову систему введення-виведення відеокарти, а інакше кажучи її BIOS - сукупність правил і алгоритмів, визначених виробником, за яким складові частини відеокарти працюють і взаємодіють між собою. [5]

- **Система охолодження** - пристрій, що здійснює відвід і розсіювання тепла від відеопроцесора, відеопам'яті та інших компонентів графічної плати з метою забезпечення нормального температурного режиму їхньої роботи.



Рис.1.3 Відеокарта від MSI

## Оперативна пам'ять (RAM)

Оперативна пам'ять (ОЗУ, RAM - Random Access Memory) - швидка енергозалежна пам'ять комп'ютера з довільним доступом, в якій здійснюються більшість операцій обміну даними між пристроями. Є енергозалежною, тобто при відключенні живлення, всі дані на ній стираються. Оперативна пам'ять є сховищем всіх потоків інформації, які необхідно обробити процесору або ж вони чекають в оперативній пам'яті своєї черги. Всі пристрої, зв'язується з оперативною пам'яттю через системну шину, а з нею в свою чергу обмінюються через кеш або ж безпосередньо. [6]

Random Access Memory - пам'ять з довільним (прямим) доступом. Значить це те, що при необхідності, пам'ять може безпосередньо звернутися до одного, необхідного блоку, не зачіпаючи при цьому інші. Швидкість довільного

доступу не змінюється від місця знаходження потрібної інформації, що є величезним плюсом.



Рис.1.4 Дві планки ОЗУ DDR3

### **Жорсткий диск (HDD та SSD)**

**Жорсткий диск** (накопичувач на жорстких магнітних дисках (НЖМД), "вінчестер", англ. - hard disk drive (HDD) - пристрій для зберігання інформації, в якому використовується принцип магнітного запису. Всередині жорсткого диску запис даних здійснюється на жорсткі пластини, виготовлені з легкометалевого сплаву або скла, вкриті шаром спеціального магнітного матеріалу (найчастіше - двоокисом хрому). Залежно від конструкції, в *HDD* можуть використовуватися одна або кілька таких пластин, що швидко обертаються на одній осі. За рахунок обертання створюється своєрідний підпір повітря, завдяки якому зчитувальні головки не торкаються поверхні пластин, хоч і знаходяться дуже близько від них (всього кілька нанометрів). Це гарантує надійність запису та зчитування даних. При зупинці пластин головки переміщуються за межі їх поверхні, тому механічний контакт між головками та пластинами практично виключений. Така конструкція забезпечує довговічність запам'ятовуючих пристроїв цього типу. [7]

Крім пластин, до складу HDD входить накопичувач, привод і блок електроніки.

Завдяки високій надійності роботи і відносно невисокій вартості, жорсткі диски є найпоширенішим пристроєм зберігання інформації.

У розмовній мові **HDD** часто називають "вінчестером" або скорочено "вінтом". Цей термін колись давно був запозичений у мисливського гвинтівкового патрона "30-30 Winchester", популярного в США на момент створення першого жорсткого диска, який у той час носив співзвучну з ним назву "30-30".



Рис.1.5 HDD накопичувач

## SSD

**SSD (solid state-drive)** або твердотільний накопичувач - запам'ятовуючий пристрій відносно нового типу, який працює на основі використання мікросхем пам'яті і на відміну від HDD не містить рухомих частин.

Цей тип пристроїв порівняно з HDD має ряд переваг: відсутність будь-якої вібрації і шуму, низьке енергоспоживання, більш висока швидкість роботи при невеликих розмірах, стійкість до температурних коливань і механічного впливу й ін.

Найбільшими *недоліками* SSD є їх висока вартість і швидкість зношування (зазвичай, близько 10 тис. циклів перезапису, у більш дорогих

виробах - до 100 тис.). Останнє обов'язково повинне враховуватися при їх експлуатації. Не рекомендується проводити дефрагментацію таких носіїв (це ніяк не прискорить пошук інформації), розміщувати на них файл підкачки, а також здійснювати інші дії, пов'язані з їх "невиправданим" використанням. Серед операційних систем сімейства Windows тільки Windows 7 враховує ці особливості. При використанні більш ранніх версій ОС Windows термін служби **SSD** скорочується.

Ймовірно, пройшовши низку вдосконалень, носії SSD з часом витіснять класичні жорсткі диски. Але поки для рядового користувача останні залишаються кращим варіантом з точки зору як довговічності, так і вартості.



Рис.1.6 SSD накопичувач

### Блок живлення

Блок живлення — вторинне джерело живлення, призначене для забезпечення живлення електроприладу електричною енергією, при відповідності вимогам її параметрів: напруги, струму, і т. д. шляхом перетворення енергії інших джерел живлення. Комп'ютерний блок живлення — блок живлення (вторинне джерело живлення), призначений для забезпечення вузлів компютера електричною енергією постійного струму. У його завдання входить перетворення мережевої напруги до заданих значень напруги живлення, її стабілізація і захист від

незначних завад з боку електричних мереж живлення. Також, будучи забезпечений вентилятором, він бере участь в охолоджуванні системного блоку. [8]



Рис.1.7 Блок живлення

## 1.2. Аналіз відомих засобів вирішення проблеми

Проблема людей у тому, що вони не бажають розібратись у комплектуючих під час покупки персонального комп'ютера, а надають перевагу готовим збіркам, які, по-перше набагато дорожчі за свою вартість і, по-друге, можуть виникнути проблеми та нестиковки між складовими. І тому зараз існує дуже мало сервісів, які надають людям змогу самому вибрати собі комплектуючі, чи можливо надати їм допомогу у виборі та їх сумісності. Розглянемо два з відомих сервісів для вирішення цієї проблеми. Це сервіс сайту «digitalfury.pro» та «telemart.ua».

Отже розпочнемо з сервісу «telemart.ua».



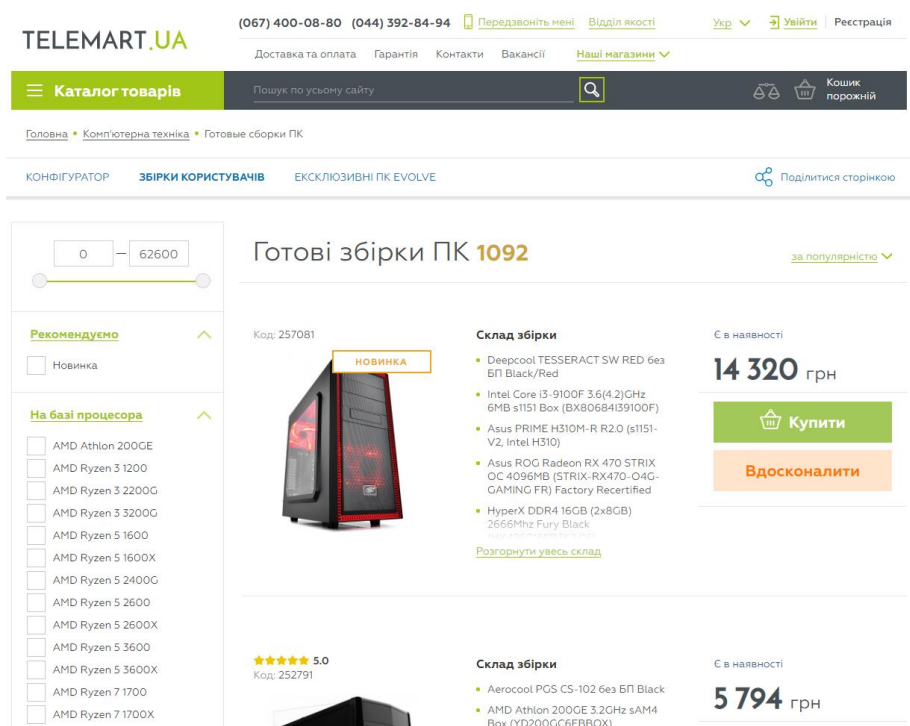


Рис. 1.8 Головне вікно «telemart.ua»

Виділимо основні плюси даного сайту:

- Зручне меню вибору комплектуючих;
- Можливість автозбірки за запитом користувача;
- Вибір улюбленого бренда, функціональності та призначення персонального комп'ютера;
- Видимі ціни та наявність даної деталі на складі чи у магазині;
- Готовий опис, характеристика та конфігурації комплектуючих;
- Пропонування користувачеві периферії до вашого ПК;
- Наявність опису самого сервісу, контакти його розробників та можливість відслідковування новин у всіх соціальних мережах.

Даний сервіс являється майже ідеальним взірцем для максимально вигідної допомоги користувачеві у збірці ПК чи просто покупки окремих деталей.

Основні мінуси даного сервісу:

- Рекламні банери по боках сайту;
- Не дуже широкий асортимент товару;
- Недостача деяких комплектуючих;

Тепер розглянемо, що нам запропонує сервіс «digitalfury.pro».

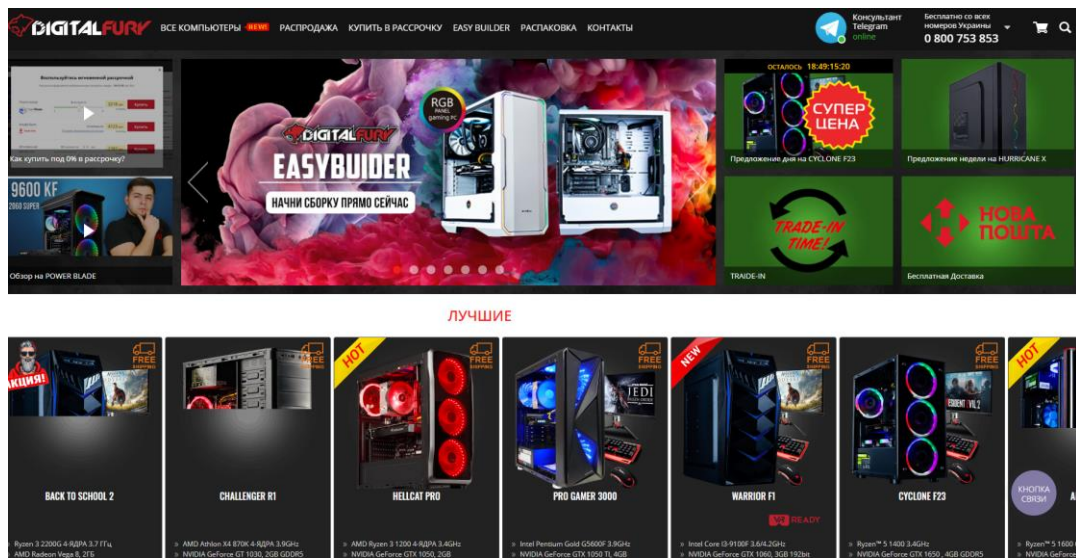


Рис.1.9 Головне вікно «digitalfury.pro»

Виділимо основні плюси:

- Відсутність рекламних банерів;
- Мінімалістичність;
- Зручний конфігуратор, в якому потрібно тільки вказати ваш E-mail, та контакти, вибрати потрібні задачі, для яких вам потрібний ПК, і через деякий час вам прийдуть запропоновані збірки комп'ютерів та конфігурацій;
- Можливість вибрати готовий комп'ютер, та змінити в ньому будь-які налаштування на ваш вибір;
- Цілодобова консультація.

У даного сервісу виявилось не так багато позитивних сторін, як у попередника, але даний сайт теж заслуговує уваги. Основні мінуси при тому, як і у минулого сервісу. Це:

- Недостача деяких комплектуючих;
- Невеликий асортимент товару та готових збірок;
- Відсутність вибору периферії.

Дані сервіси є чудовими та зручними у використанні, допомагають швидко та максимально якісно і вигідно підібрати комплектуючі та збірки користувачеві.

### 1.3. Функціональність програмної системи

Перелік функцій, які потрібно включити до проектованої системи:

1. Функція перегляду комплектації - дозволяє користувачеві за моделлю ПК переглянути його конфігурації;
2. Функція зміни комплектації - дозволяє змінити конфігурації даного ПК;
3. Функція вибору фільтрів – дозволяє відфільтрувати ПК за бажанням користувача;
4. Функція вибору ексклюзивних ПК – дозволяє вибрати та переглянути лімітовані версії комп'ютерів;
5. Функція інструкції – допомагає користувачеві з навігацією на сервері;
6. Функція реєстрації – дозволяє користувачеві створити обліковий запис;
7. Функція реклами своїх соціальних мереж – дозволяє користувачеві відслідковувати новини чи зв'язатися з адміністрацією;
8. Функція кошика – дозволяє зберегти бажані збірки чи деталі;
9. Функція спецпропозицій – дозволяє користувачеві ввести свій E-mail, щоб отримувати різні цікаві варіанти ПК;
10. Функція графіку роботи – дозволяє користувачеві бачити, коли працює даний сервіс;
11. Функція цілодобової підтримки – допоможе користувачеві у будь-який час дня;
12. Функція покупки – дозволяє користувачеві безпосередньо купити те, що він бажає без додавання цього у кошик;
13. Функція тестування – дозволяє користувачеві побачити, наскільки його збірка буде ефективна у тих чи інших задачах;
14. Функція порівняння – дозволяє користувачеві порівняти потужності збірок або окремих комплектуючих;

15. Функція бонусів – дозволяє отримати користувачеві певну знижку при замовленні від певної суми;
16. Функція «передзвоніть мені» - дозволяє користувачеві залишити свій телефон, у випадку якщо сервер впав, або онлайн підтримка у даний момент недоступна;
17. Функція «шерінгу» - дозволяє користувачеві поділитися даним сервісом, якщо він забажає;
18. Функція оцінювання – дозволяє користувачеві оцінити якість сервісу;
19. Функція «книги скарг та побажань» - дозволяє користувачеві в розширеному вигляді допомогти з покращенням сервісу порадою;
20. Функція завантаження фото і відео – дозволяє користувачеві поділитись і показати іншим своє розпакування, і чи все прийшло у хорошому стані;
21. Функція переходу – дозволяє користувачеві перейти до партнерів сервісу чи інших виробників;
22. Функція «гаманець» - дозволяє користувачеві залишити свої кошти на самому сервісі, і не вводити після покупки дані з карти чи оплачувати при отримуванні;
23. Функція трекінгу – дозволяє відслідкувати своє замовлення на карті;
24. Функція «побудови» - дозволяє користувачеві самому скласти бажану збірку ПК;
25. Функція замовлення – дозволяє замовити бажану деталь, якої немає в наявності або ж новітню.

З метою забезпечення додаткової інформації про кожну функцію використовується поняття атрибутів функції.

*Таблиця 1.1 Атрибути функцій*

№	Статус	Пріоритет	Трудоємність	Ризик	Стабільність	Цільова версія
---	--------	-----------	--------------	-------	--------------	----------------

1	Включена	Критична	Низький рівень	Низький	Висока	v1.1
2	Включена	Критична	Низький рівень	Низький	Висока	v1.1
3	Пропонов ана	Корисна	Середній рівень	Низький	Висока	v1.2
4	Включена	Важлива	Середній рівень	Середній	Низька	v1.1
5	Пропонов ана	Корисна	Низький рівень	Середній	Висока	v1.2
6	Пропонов ана	Важлива	Середній рівень	Середній	Низька	v1.3
7	Пропонов ана	Корисна	Середній рівень	Середній	Середня	v1.2
8	Пропонов ана	Корисна	Високий рівень	Високий	Низька	v1.3
9	Включена	Важлива	Низький рівень	Середній	Низька	v1.1
10	Включена	Важлива	Низький рівень	Низький	Середня	v1.2
11	Включена	Важлива	Середній рівень	Низький	Низька	v1.1
12	Включена	Корисна	Середній	Низький	Низька	v1.2

			рівень			
13	Пропонована	Корисна	Середній рівень	Низький	Середня	v1.1
14	Включена	Критична	Низький рівень	Низький	Середня	v1.1
15	Включена	Критична	Низький рівень	Низький	Низька	v1.1
16	Пропонована	Корисна	Середній рівень	Низький	Низька	v1.2
17	Включена	Важлива	Середній рівень	Середній	Середня	v1.1
18	Пропонована	Корисна	Низький рівень	Середній	Висока	v1.2
19	Пропонована	Важлива	Середній рівень	Середній	Середня	v1.3
20	Пропонована	Корисна	Середній рівень	Середній	Низька	v1.4
21	Пропонована	Корисна	Високий рівень	Високий	Середня	v1.3
22	Включена	Важлива	Низький рівень	Середній	Низька	v1.4
23	Включена	Важлива	Низький рівень	Низький	Висока	v1.4

24	Включена	Важлива	Середній рівень	Низький	Середня	v1.1
25	Включена	Корисна	Середній рівень	Низький	Висока	v1.2

Після проведеного аналізу було вирішено реалізувати негайно функції № 1, 2, 3, 4, 10, 11, 12, 13. Відкласти до наступної версії функції № 5, 6, 7, 9. Додатково дослідити функцію № 8.

## **Висновок**

У даному розділі було проаналізовано персональний комп'ютер, а саме усі його складові, їх види та підвиди. Знайдено та продемонстровано проблему у людей, її основні методи вирішення. Було представлено два сервіси, які допомагають боротися з даною проблемою. Застосовано механізм використання функцій, за для зручного опису системи, яку необхідно буде спроектувати, наведено та описано основні функції даного сервісу.

## Розділ 2. Проектування системи

### 2.1. Опис системних вимог згідно з методологією RUP

1) Зацікавлені особи прецеденту та їх вимоги: в цій секції опису подається, що повинна зробити ПС для усіх можливих учасників процесу.

- Сервісний центр: хоче отримати інформацію про замовлення, тип збірки ПК, комплектацію, інформацію про клієнта, укласти контракт, отримати предоплату при замовленні і отримати оплату при продажі комп'ютера;
- Клієнт: хоче швидко оформити своє замовлення з мінімальними витратами часу;
- Комп'ютерна майстерня: хоче мати інформацію про замовлення клієнта;
- Склад: хоче використати нереалізовану продукцію;

2) Користувач ПС тобто основний актор цього прецеденту: це сервісний центр, який опрацьовує замовлення клієнтів, склад та комп'ютерна майстерня, яка отримує та реалізовує замовлення клієнтів за допомогою ПС яка має бути розроблена.

3) Передумови прецеденту (preconditions): це перелік подій які завжди повинні виконуватися до початку сценарію поточного прецеденту. Передумови не перевіряються у даному сценарії прецеденту, вони вважаються результатом успішного виконанням деякого іншого прецеденту. Наприклад, це можуть бути наступні умови

- Сервіс повинен успішно пройти процедуру автентифікації в ПС
- Майстерня повинен успішно пройти процедуру автентифікації в ПС
- Клієнт повинен успішно пройти процедуру автентифікації в ПС

4) Основний успішний сценарій: ця секція специфікації описує «сценарій успіху». Тобто дії що призводять до успішного завершення подій в основному процесі, наприклад:

1. Клієнт оформляє замовлення, підписує контракт і вносить передоплату;
2. ПС прсилає замовлення до комп'ютерної майстерні;
3. Сервіс забирає готовий ПК з майстерні і доставляє її в магазин;



4. Сервіс доставляє ПК до клієнта;
5. Клієнт оплачує замовлення;
- 5) Розширення основного сценарію або альтернативні потоки: у цій секції вказуються всі інші можливі варіанти розвитку подій в ПС, які призводять до успішного завершення основного сценарію або альтернативні сценарії, що приводять до некоректного завершення прецеденту. При цьому після виконання обробки усіх можливих розширень прецеденту ПС повинна забезпечити повернення користувача в основний сценарій, якщо в ПС не передбачено альтернативний хід подій. Наприклад, розширення основного сценарію:
  1. Клієнт відмінює замовлення;
  2. Комп'ютерна майстерня повідомляє про неможливість виконання замовлення; Менеджер сервісного центру зв'язується з покупцем і пропонує інші варіанти комплектації або інші типи персонального комп'ютера.
  3. Замовлення під час доставки получило пошкодження; Менеджер сервісного центру зв'язується з покупцем та повідомляє про пошкодження. Клієнт або відмовляється від замовлення або оформлює його зі знижкою.
  4. ПС повідомляє центр про помилку і відмінює доставку замовлення.
  5. ПС повідомляє майстерню про помилку і відмінює доставку замовлення.
- 6) Пост-умови (postconditions): це перелік умов, які завжди повинні виконуватися у випадку успішного виконання основного сценарію (тобто коли задовольняються інтереси усіх зацікавлених осіб із п.2), наприклад.
  - дані щодо продажу оброблені та збережені в ПС;
  - роздруковано потрібний чек;
  - транзакція успішно зафіксована в базі даних ПС;
- 7) Спеціальні СВ: ця секція опису містить перелік нефункціональних вимог, атрибутів якості або певних обмеження в функціонування ПС, що є пов'язаними із даним прецедентом. Наприклад, це може бути:

- Необхідно забезпечити 100% надійність обробки всіх транзакцій,
- Потрібно забезпечити можливість локалізації інтерфейсу користувача ПС

8) Список необхідних технологій та додаткових пристроїв: типовим прикладом можуть бути технічні вимоги, які висувають зацікавлені особи для технологій вводу/виводу даних тощо.

- ПС має бути розроблена як Web-орієнтована система для обробки замовлень менеджерами в дилерських центрах і на заводах;

## 2.2. Деталізація функцій та аналіз категорій користувачів

### 2.2.1. Діаграма варіантів використання

**1.**Визначити загальні межі та контекст предметної області, яка моделюється на початкових етапах проектуванні системи.

**2.**Сформулювати загальні вимоги до функціонування поведінки ПС.

**3.**Розробити початкову концептуальну модель системи для подальшої її деталізації у формі логічних та фізичних моделей.

**4.**Підготувати початкову документацію для взаємодії розробників з замовниками та користувачами.

Система представляється у вигляді множини акторів, які взаємодіють з системою за допомогою варіантів використання.

**Варіант використання** –це послідовність дій, які повинні бути виконані системою при взаємодії з відповідним актором.

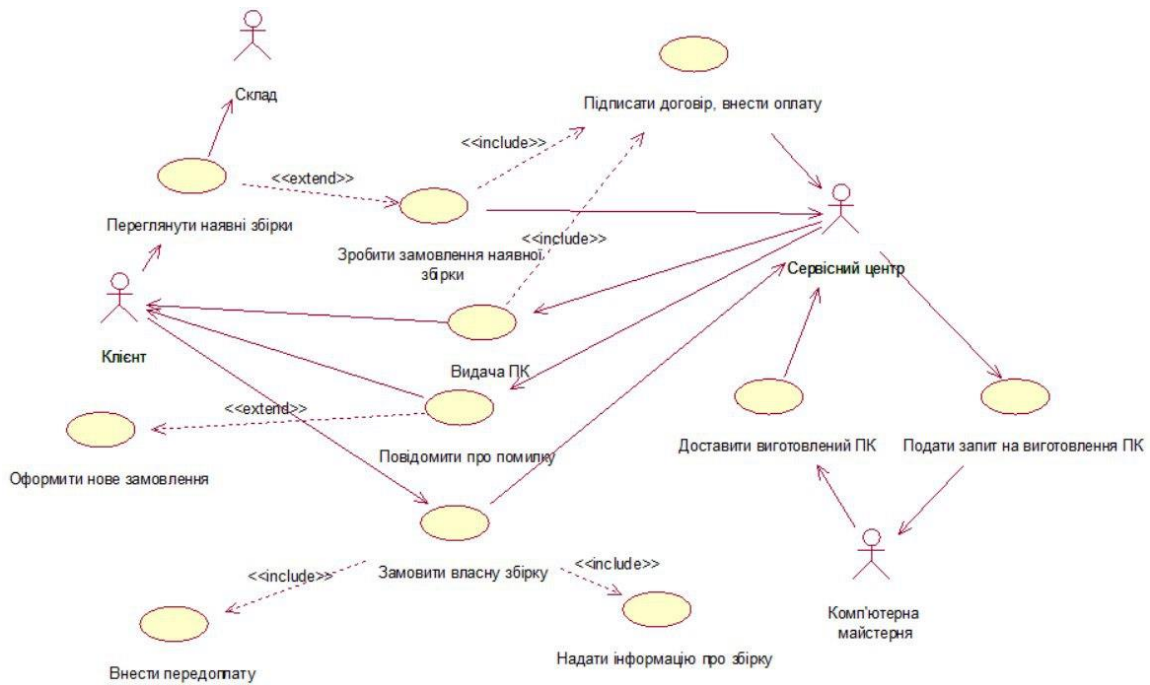


Рис 2.1 Діаграма варіантів використання системи керування комп'ютерною сервісною мережею.

Система починає роботу з бізнес актора Клієнт, який робить замовлення. Замовлення, в свою чергу, пересилається до Комп'ютерної майстерні та Сервісу для підпису контракту і внести передоплату. Обов'язковість цього вказує відношення `<<include>>`. Після цього актор Сервіс подає заявку у майстерню і вона доставляє ПК. У разі виникнення проблем на стороні Майстерні, актор Сервіс повідомляє Клієнта про помилку. Відношення зі стереотипом `<<extend>>` вказує на те що, Клієнт може відмовитися від оформлення нового замовлення. Також Клієнт може відмовитися від замовлення, після чого ПК скасовує дане замовлення у Сервіса та Майстерні. У разі успішної доставки Клієнт оплачує замовлення у Сервіса.

### 2.2.2. Діаграма класів

Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів

та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

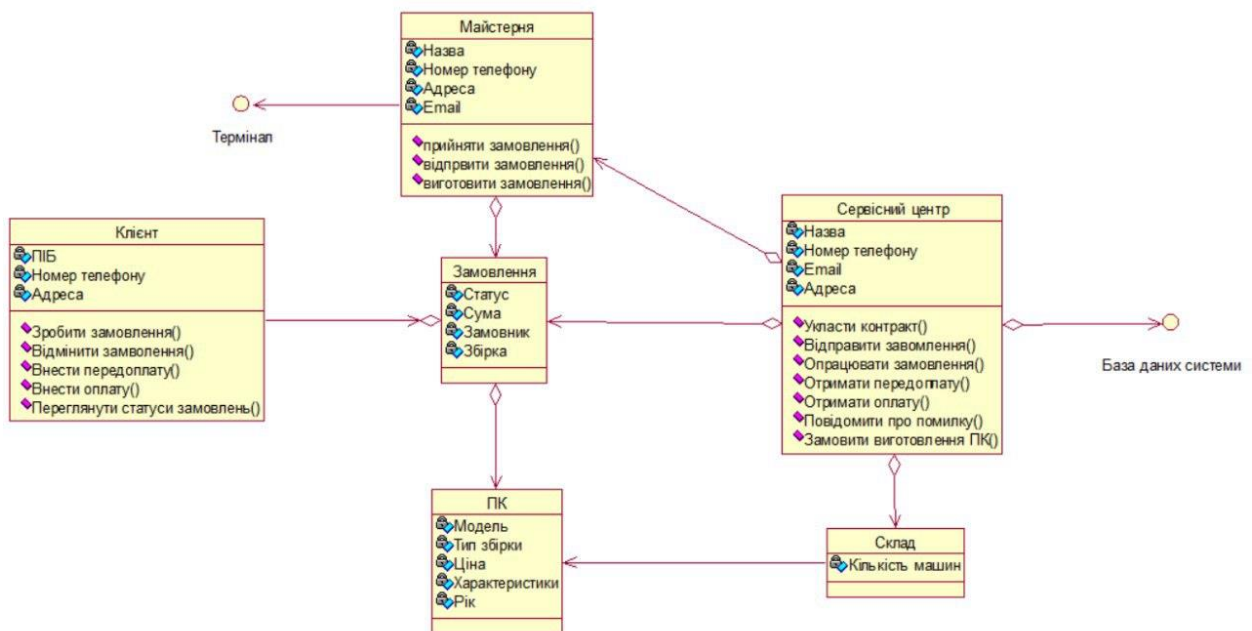


Рис 2.2 Діаграма класів системи керування мережею сервісного центру.

### 2.2.3. Діаграма кооперації

Головна особливість діаграми кооперації полягає в можливості графічно представити не тільки послідовність взаємодії, але й усі структурні відносини між об'єктами, які беруть участь у цій взаємодії.

На відміну від діаграми послідовності, на діаграмі кооперації зображаються тільки відносини між об'єктами, що грають певні ролі у взаємодії. З іншого боку, на цій діаграмі не вказується час у вигляді окремого виміру. Тому послідовність взаємодій і паралельних потоків може бути визначена за допомогою порядкових номерів.

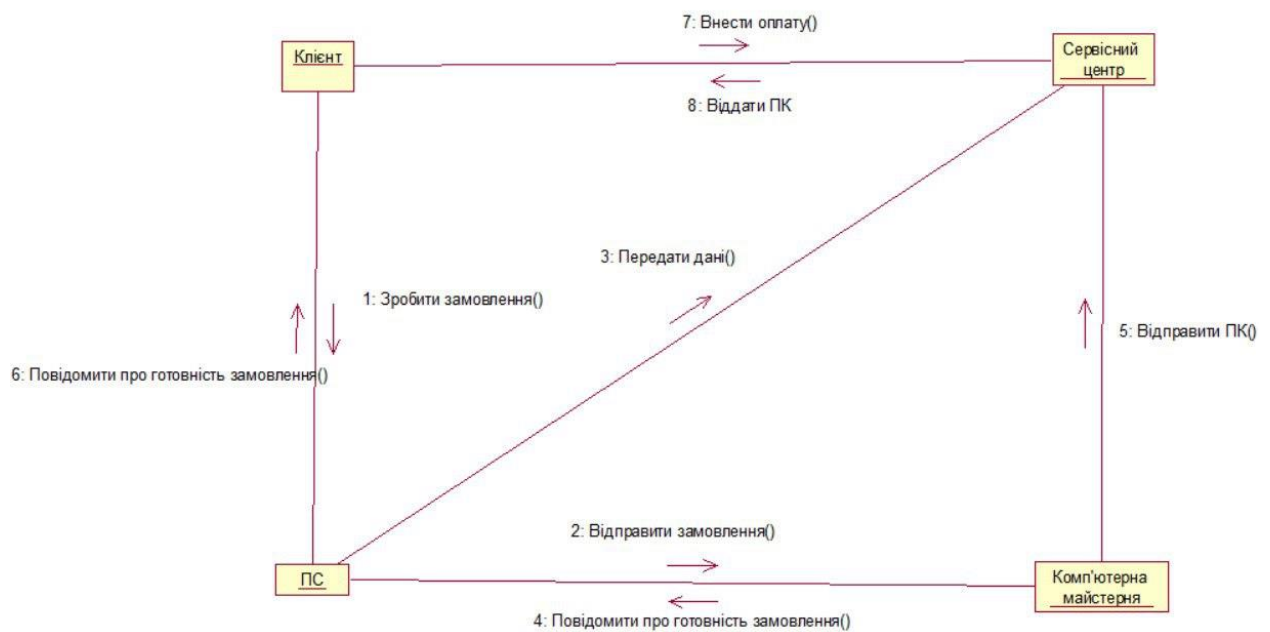


Рис 2.3 Діаграма кооперації системи керування мережею сервісного центру.

#### 2.2.4. Діаграма послідовності

Діаграма послідовності — відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

Іншими словами, діаграма послідовностей відображає часові особливості передачі і прийому повідомлень об'єктами.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями.

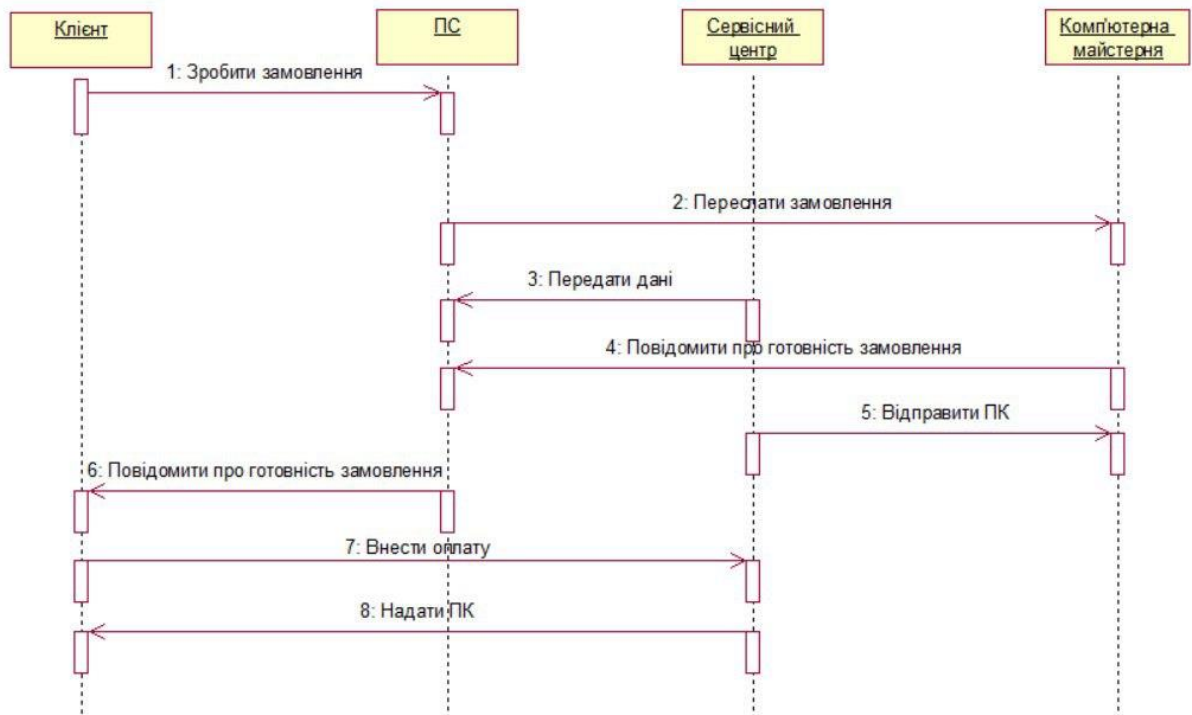


Рис 2.4. Діаграма послідовності системи керування мережею сервісного центру.

### 2.2.5. Діаграма діяльності

Діаграма діяльності (діаграма активності) дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів. Зазначені послідовності можуть являти собою альтернативні галузей процесу обробки даних або галузям, які можуть виконуватися паралельно.

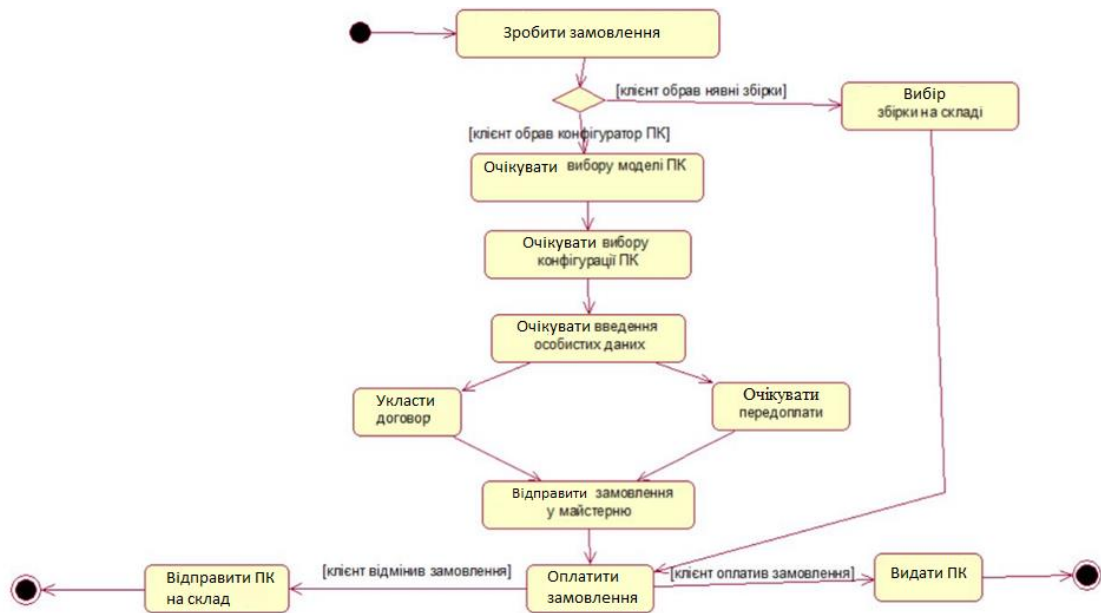


Рис 2.5 Діаграма діяльності системи керування авто дилерською мережею.

## 2.2.6. Діаграма компонентів

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

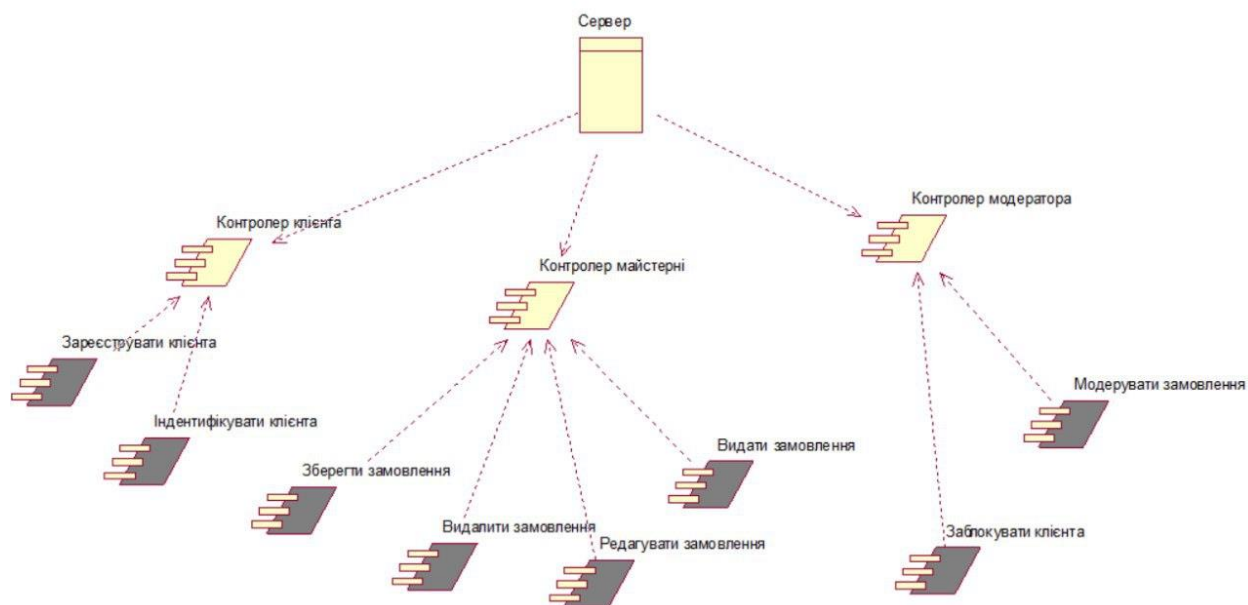


Рис 2.6. Діаграма компонентів системи керування мережею сервісного центру.

## Висновок

За допомогою методології RUP, було спроектовано ряд діаграм, де було описано нашу предметну область. Основні діаграми це:

- Діаграма класів;
- Діаграма послідовності;
- Діаграма компонентів;
- Діаграма кооперації;
- Діаграма діяльності;
- Діаграма варіантів використання.



## Розділ 3. Вибір засобів реалізації та конструювання системи

### 3.1. Вибір програмного рішення

JavaScript ("JS" скорочено) це повноцінна динамічна мова програмування, яка, у застосуванні до HTML документу, може надати динамічну інтерактивність на веб-сайтах. Вона була винайдена Бренданом Ейхом, співзасновником проекту Mozilla, the Mozilla Foundation, та Mozilla Corporation.

JavaScript має надзвичайно багато застосувань. Ви можете розпочати з малого: створити "каруселі", галереї зображень, динамічні макети сторінок, відповіді на натиски кнопок, тощо. Із досвідом, ви зможете створювати ігри, 2D та 3D графіку, складні застосунки з використанням баз даних та багато іншого!

JavaScript доволі компактна та гнучка мова. Розробники забезпечили велике розмаїття інструментів, що доповнюють основу мови JavaScript, які відкривають величезну кількість додаткового функціоналу з мінімальними зусиллями. Серед них:

Програмні інтерфейси (APIs) для браузерів — API, які вбудовані у браузери, що надають функціонал на зразок динамічного створення HTML та застосування CSS-стилів, збір та обробка відео-потоків з вебкамери користувача, генерація 3D-графіки та аудіо-семплів.

API третіх осіб, що дозволяють розробникам інтегрувати у власні сайти функціонал інших провайдерів, таких як Twitter або Facebook.

Фреймворки та бібліотеки третіх осіб, які ви можете застосувати до вашого HTML, щоб прискорити створення сайтів та застосунків.

HTML - скорочення від "HyperText Mark-up Language" - перекладається як "Мова розмітка гіпертексту" (Гіпертекст - це текст, що не послідовно зв'язаний з іншими документами, тобто у вас є змога з першої сторінки

документу перейти на останню). Іншими словами HTML - це мова розмітки, або ще один спосіб зберігання інформації. За допомогою HTML ти позначаєш текст, вказуючи своєму веб-переглядачу, як він має розуміти позначений текст, так само як і на жорсткому диску інформація зберігається в блоках, кластерах, секторах, доріжках і тільки за допомогою, такої, визначеної структури твій комп'ютер розуміє, що треба, а що не треба зчитувати.

У HTML текст позначається за допомогою тегів. Кожен HTML документ буде складатися з деякої групи елементів, де кожен елемент буде визначатися (починатися та закінчуватися) певним тегом (Для деяких елементів кінцевий тег не є обов'язковим). Тег — це назва елемента, записана у кутових дужках (< >)

Кожен HTML тег має свою унікальну назву з визначеним синтаксисом, яка записується латинськими літерами і не чутливий до регістру.

CSS (аббревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів) - це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи веб-сторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML. Одна з головних переваг використання CSS - це можливість розділити зміст сторінки від її оформлення. Таке розділення дозволило покращити сприйняття та доступність змісту, забезпечити більшу гнучкість та контроль за відображенням змісту в різних умовах, зробити зміст більш структурованим та простим, прибрати повторення та ін. Власне це ж і була основна мета створення цієї технології.

Що дає використання CSS:

- Відображати один і той же документ в різних стилях.

- Декілька дизайнів сторінки для різних пристроїв. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на смартфоні меню буде внизу, під вмістом.
- Зменшення часу завантаження сторінок сайту за рахунок перенесення правил відображення в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а стилеві правила цих даних завантажуються браузером тільки один раз і кешуються.
- Простота подальшої зміни дизайну. Не потрібно правити кожну сторінку, а лише змінити CSS-файл.
- Додаткові можливості оформлення. Наприклад, за допомогою CSS-розмітки можна зробити так, щоб меню було завжди видно при скролінгу сторінки, або прибрати підкреслення у посилань.
- Дозволяє створювати складну і пропрацьовану техніку дизайну.

Як асинхронне подієве JavaScript-оточення, Node.js спроектований для побудови масштабованих мережеских додатків. У нижче наведений приклад "hello world", який може одночасно обробляти багато з'єднань. Для кожного з'єднання викликається функція зворотнього виклику, проте коли з'єднань немає Node.js засинає. Це контрастує з більш загальною моделлю в якій використовуються паралельні OS потоки. Такий підхід є відносно неефективним та дуже важким у використанні. Більше того, користувачі Node.js можуть не турбуватись про блокування процесів, оскільки немає жодних блокувань. Майже жодна з функцій у Node.js не працює напряму з I/O, тому процес не блокується ніколи. Оскільки нічого не блокується на Node.js легко розробляти масштабовані системи.

### 3.2. Аналіз контрольного прикладу

Розроблена система володіє простим і легким інтерфейсом щоб забезпечити легкодоступність для клієнта.

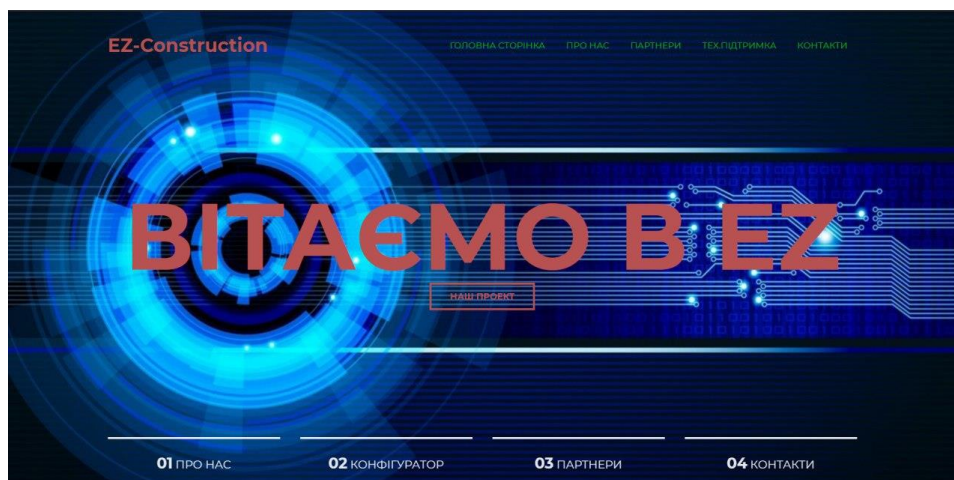


Рис 3.1 Інтерфейс системи

Це перше, що бачить користувач при вході на даний сайт – головне вікно.

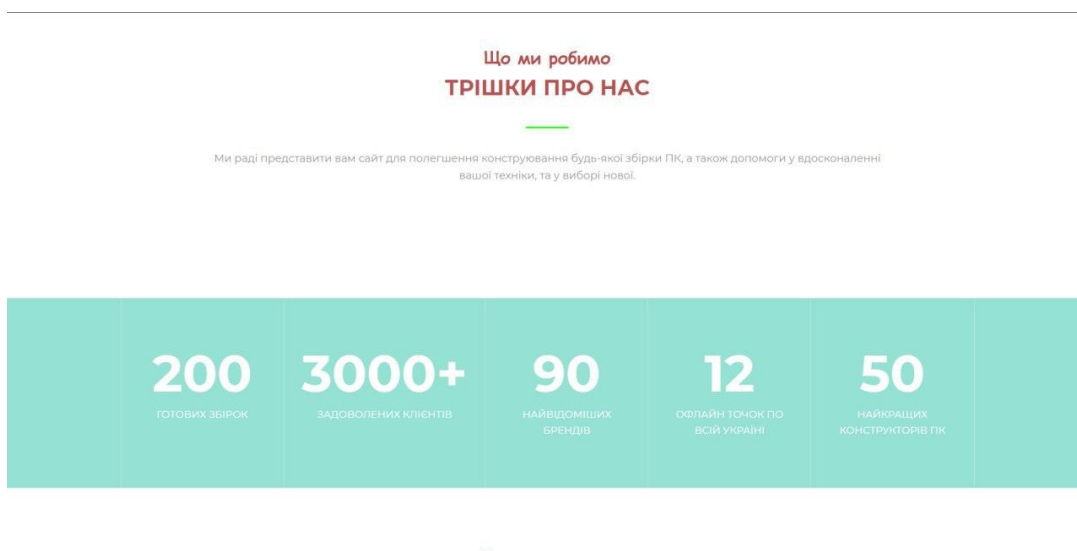


Рис 3.2 Інтерфейс системи

На головній частині сторінки зображено інформацію та деякі успіхи компанії.



Рис 3.3 Інтерфейс системи

Загальна інформація про компанії, з якими співпрацює даний сервіс.

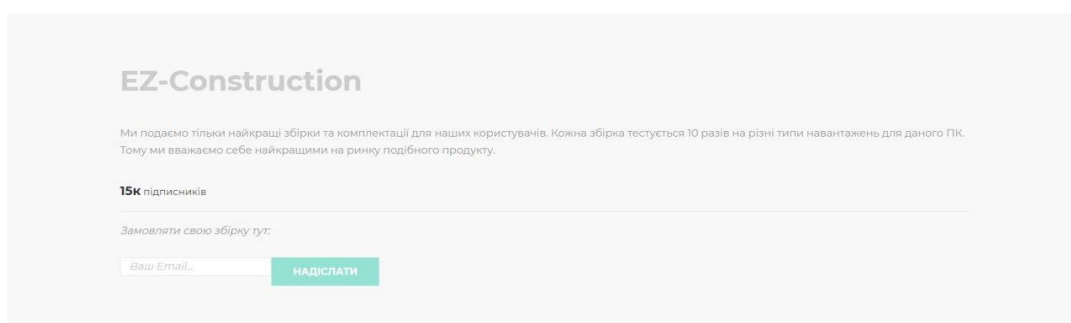


Рис. 3.4 Інтерфейс системи

Вікно відправлення замовлення на пошту. Після отримання замовлення дані відправляються на сервер за допомогою back-end складової сайту і відправляються інженерам компанії, які втілять замовлення у життя.



Рис 3.5 Відправленні замовлення

Підбірка комплектацій ПК.

\*Обов'язкове поле

Прізвище та ім'я \*

Ваша відповідь

Адрес електронної пошти \*

Ваша відповідь

Номер телефону

Ваша відповідь

Рис 3.6 Вигляд анкети

Для чого вам потрібен ПК? \*

☐ Ігровий

☐ Для офісної роботи

☐ Для монтажу\фотошопу та великих навантажень на ПК

Виберіть компанії, з якими бажаєте отримати ваш ПК. \*

☐ nVidia

☐ Asus

☐ GIGABYTE

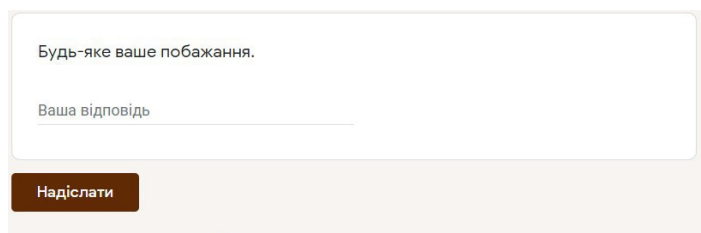
☐ MSI

☐ INTEL

☐ AMD

☐ Інше: \_\_\_\_\_

Рис 3.7 Вигляд анкети



Будь-яке ваше побажання.

Ваша відповідь

Надіслати

Рис 3.8 Вигляд анкети

## Висновок

В даному розділі було сформовано обґрунтування вибору програмного забезпечення і розроблено розгорнутий текстовий опис створеної системи.

## **Висновки**

В даній курсовій роботі мною було розглянуто основну проблематику теми розробки підбору оптимальної конфігурації комп'ютера. Досліджено актуальність створення даної системи. З метою вирішення даної проблеми було проведено аналіз основних аналогів системи з виділенням їхніх плюсів і мінусів для визначення необхідного функціоналу програмної системи з її атрибутами, за для розробки та всебічного покращення існуючих систем. А також розроблено розгорнутий текстовий опис основних системних вимог до системи за допомогою технології RUP, створено ряд діаграм, які послужили структурою для створення програмного забезпечення. Було проведено аналіз вимог та опис функцій системи та сформувалось відповідне їх представлення. Було застосовано тип інформаційної системи, де користувач напрямую контактує з даним сервісом через його програмне забезпечення. Визначились категорії користувачів та було побудовано узагальнену модель функціонування системи в навколишньому середовищі.



## Список використаних джерел

1. Системна плата і її головні елементи. URL: [<http://ipkey.com.ua/uk/faq/976-motherboard.html>]
2. Процесор, основні функції, характеристики. URL: [<https://alex-legend.livejournal.com/1270.html>]
3. Будова і основні характеристики відеокарти. URL: [[https://www.chaynikam.info/ukr/stat\\_gpu.html](https://www.chaynikam.info/ukr/stat_gpu.html)]
4. Оперативна пам'ять (ОЗУ) URL: [<http://www.xtechx.ru/c40-visokotehnologichni-spravochnik-hitech-book/operativnaya-pamyat-ozu-memory>]
5. Будова і основні характеристики жорсткого диску та SSD. URL: [[https://www.chaynikam.info/ukr/stat\\_hdd.html](https://www.chaynikam.info/ukr/stat_hdd.html)]
6. Блок живлення, основні функції та характеристики. URL: [<https://vika-vita-vika.livejournal.com/1070.html>]
7. Кабелі та перехідники для комп'ютера. URL: [<https://patronservice.ua/ukr/support/glossary/kabeli-i-perehodniki.html>]
8. Що таке материнська плата ПК, на що впливає. URL: [<https://patronservice.ua/ukr/support/glossary/kabeli-i-perehodniki.html>]

## Додатки

Index.html

```
<!doctype html>
```

```
<html lang="ua">
```

```
<head>
```

```
<meta charset="UTF-8" >
```

```
<link rel="stylesheet" href="app.css">
```

```
<link href="https://fonts.googleapis.com/css?
family=Kaushan+script|Montserrat:400,700&subset=cyrillic-ext"
rel="stylesheet">
```

```
<title>EZ-Constraction </title>
```

```
</head>
```

```
<body>
```

```
<header class="header">
```

```
<div class="container">
```

```
<div class="header__inner">
```

```
<div class="logo">EZ-Construction</div>
```

```
<nav class="nav">
```

```
<a class="nav__link" href="#">Головна сторінка</a>
```

```
<a class="nav__link" href="#">Про нас</a>
```

```
<a class="nav__link" href="#">Партнери</a>
```

```
<a class="nav__link" href="#">Тех.підтримка</a>
```

```
<a class="nav__link" href="#">Контакти</a>
```

```
</nav>
```

```
</div>
```

```
</div>
```

```
</header>
```

```
<div class="intro">
```

```
<div class="container">
```

```
<div class="intro__inner"></div>
```

```
<h1 class="intro__title"> Вітаємо в EZ</h1>
```

```
<a class="btn" href="#">Наш проект</a>
```

```
</div>
```

```
<div class="slider">
```

```
<div class="container">
```

```
<div class="slider__inner">
```

```
<div class="slider__item active">
```

```
<span class="slider__num">01</span>
```

```
Про нас
```

```
</div>
```

```
<div class="slider__item">
```

```
<span class="slider__num">02</span>
```

```
Конфігуратор
```

```
</div>
```

```
<div class="slider__item">
```

```
<span class="slider__num">03</span>
```

```
Партнери
```

```
</div>
```

```
<div class="slider__item">
```

```
<span class="slider__num">04</span>
```

```
Контакти
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<section class="section">
```

```
<div class="container">
```

```
<div class="section__header">
```

```
<h3 class="section__suptitle">Що ми робимо</h3>
```

```
<h2 class="section__title">Трішки про нас</h2>
```

```
<div class="section__text">
```

```
<p> Ми раді представити вам сайт для полегшення конструювання будь-
якої збірки ПК, а також допомоги у вдосконаленні вашої техніки, та у виборі
нової.
```

```
</p>
```

```
</div>
```

```
</div>
```

```
<div class="about">
```

```
<div class="about__item">
```

```
<div class="about__image">
```

```

```

```
</div>
```

```
</div>
```

```
<div class="about__item">
```

```
<div class="about__image">
```

```

```

```
</div>
```

```
</div>
```

```
</div>  
  </div>  
</section>
```

```
<div class="statistics">  
  <div class="container">  
    <div class="stat">  
      <div class="stat__item">  
        <div class="stat__count">200</div>  
        <div class="stat__text">Готових збірок</div>  
      </div>  
      <div class="stat__item">  
        <div class="stat__count">3000+</div>  
        <div class="stat__text">Задоволених клієнтів</div>  
      </div>  
      <div class="stat__item">  
        <div class="stat__count">90</div>  
        <div class="stat__text">Найвідоміших брендів</div>  
      </div>  
      <div class="stat__item">  
        <div class="stat__count">12</div>  
        <div class="stat__text">Офлайн точок по всій Україні</div>  
      </div>  
      <div class="stat__item">  
        <div class="stat__count">50</div>  
        <div class="stat__text">Найкращих конструкторів ПК</div>  
      </div>  
    </div>  
  </div>
```

</div>

<section class="section">

<div class="container">

<div class="section\_\_header">

<h3 class="section\_\_suptitle">Ми працюємо з</h3>

<h2 class="section\_\_title">Найкращими компаніями</h2>

</div>

<div class="services">

<div class="services\_\_item">

<div class="services\_\_title">nVidia</div>

<div class="services\_\_text">Корпорація nVIDIA (стилізовано nVIDIA) — американський виробник графічних процесорів, відеоадаптерів під торговими марками Riva TNT та GeForce, мультимедійних та комунікаційних пристроїв для ПК та ігрових консолей.</div>

</div>

<div class="services\_\_item">

<div class="services\_\_item">

<div class="services\_\_title">AMD</div>

<div class="services\_\_text">Advanced Micro Devices, Inc. (AMD) — компанія-виробник інтегрованої електроніки. Це другий найбільший постачальник x86 сумісних процесорів і великий постачальник флеш-пам'яті.

</div>

</div>

</div>

<div class="services\_\_item">

<div class="services\_\_item">

```

<div class="services__title">Intel</div>
<div class="services__text">Intel Corporation — найбільша у світі
компанія-виробник напівпровідникових елементів та пристроїв,
    найбільш відома як розробник та виробник x86-серії мікропроцесорів,
процесорів для IBM-сумісних персональних комп'ютерів.</div>
</div>
</div>
</div>
<hr>
<div class="services">
    <div class="servises__item">
        <div class="services__title">ASUS</div>
        <div class="services__text">ASUSTeK Computer Incorporated -
розташована в Тайвані міжнародна компанія, що виробляє материнські
плати, відеокарти, оптичні приводи, КПК, монітори, ноутбуки, сервери,
мережеве обладнання, мобільні телефони, комп'ютерні корпуси, компоненти
і системи охолодження.</div>
    </div>
    <div class="servises__item">
        <div class="servises__item">
            <div class="services__title">MSI</div>
            <div class="services__text">Micro-Star International - тайванська
компанія, заснована в 1986 році, володіє власними виробничими
потужностями і спеціалізується на виготовленні компонентів для IBM PC-
сумісних комп'ютерів, таких як материнські плати, відеокарти, геймерська
периферія, а також ноутбуків і ігрових комп'ютерів. </div>
        </div>
    </div>
    <div class="servises__item">
        <div class="servises__item">

```

<div class="services\_\_title">GIGABYTE</div>

<div class="services\_\_text">GIGABYTE Group - тайванська компанія з виробництва системних плат, відеокарт та іншої комп'ютерної техніки.

Заснована в 1986-ому році як група дослідників і розробників.

До 2008 року Gigabyte є холдингом, до складу якого входять:</div>

</div>

</div>

</div>

</div>

</section>

<footer class="footer">

<div class="container">

<div class="footer\_\_inner">

<div class="footer\_\_col">

<div class="footer\_\_logo">

EZ-Construction

</div>

<div class="footer\_\_text">Ми подаємо тільки найкращі збірки та комплектації для наших користувачів.

Кожна збірка тестується 10 разів на різні типи навантажень для даного ПК. Тому ми вважаємо себе найкращими на ринку

подібного продукту.</div>

<div class="footer\_\_social">

<div class="footer\_\_social-header">

<b>15к</b> підписників

</div>

<div class="footer\_\_social-content">



Замовляти свою збірку тут:

```
</div>
```

```
</div>
```

```
<form class="subscribe" action="/mail" method="post">
```

```
  <input class="subscribe__input" id="mail" type="email" name="name"
  onchange="write()" placeholder="Ваш Email...">
```

```
  <button class="subscribe__btn" type="submit" href="/mail" >
  Надіслати</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</footer>
```

```
<script>
```

```
function write() {
  let data = document.getElementById('mail').value;
  fs.writeFile('data.txt', data, function (err) {
    if (err) return console.log(err);
  });
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Index.js

```
const http = require('http')
```

```
const fs = require('fs')
```

```
const nodemailer = require("nodemailer");
```

```
http.createServer((req, res) =>{
```

```
  switch (req.url) {
```

```
    case '/':
```

```
      res.writeHead(200,{ 'Content-Type':'text/html'});
```

```
      let myFileHtml = fs.readFileSync('index.html');
```

```
      res.end(myFileHtml);
```

```
    case '/app.css':
```

```
      res.writeHead(200,{ 'Content-Type':'text/css'});
```

```
      let myFileCss = fs.readFileSync('css/normalize.css');
```

```
      res.end(myFileCss);
```

```
    case '/app.bg':
```

```
      res.writeHead(200,{ 'Content-Type':'img/jpg'});
```

```
      let myFileBg = fs.readFileSync('kursova/back2.jpg');
```

```
      res.end(myFileBg);
```

```
    case '/app.js':
```

```
      res.writeHead(200,{ 'Content-Type':'text/js'});
```

```
      let myFilejs = fs.readFileSync('js/main.js');
```

```
      res.end(myFilejs);
```

```
    case '/mail':
```

```
      var mail = fs.readFileSync('data.txt');
```

```
      var data ="Посилання для заповнення анкети для вибору ПК
```

```
      https://docs.google.com/forms/d/e/1FAIpQLSeVnYJYrBJMd_rcWu3C5iD8Uu_2o  
      DYu115srPJJaRCzEX2ev5A/viewform?usp=sf_link";
```

```

const transporter = nodemailer.createTransport({
  host: 'smtp.gmail.com',
  port: 587,
  secure: false,
  auth: {
    user: 'zhytelselakulyava001@gmail.com',
    pass: 'MAX80300KYmin'
  }
});

const mailer = message => {
  transporter.sendMail(message, (err, info) => {
    if(err) return console.log(err);
    console.log('Email sent: ', info)
  })
};

const message = {
  from: 'EZ-Construction<oleh.hadzalo.kn.2017@lpnu.ua>',
  to: "maxim.prisma.maidan@gmail.com",
  subject: 'New Order',
  html: data
};

mailer(message);

let myFileHt = fs.readFileSync('index.html');
res.end(myFileHt);

default:
  res.writeHead(404, {'Content-Type': 'text/plain'});
  res.end('404 Не знайдено');

```

```

    }
  }).listen(2000,() =>{ console.log('Server has been started....')});

```

Normalize.css

```

body {
  margin: 0;
  font-family:'Montserrat',sans-serif;
  font-size: 15px;
  line-height: 1.6;
  color: #b85151;
}
*,
*:before,
*:after {
  box-sizing: border-box;
}
h1,h2,h3,h4,h5,h6 {
  margin:0;
}

hr{
  display: block;
  width: 100%;
  height: 1px;
  margin: 40px 0;

  border: 0;
  background-color: #e5e5e5;
}

```

```
/*Контейнер*/
```

```
.container {  
    width: 100%;  
    max-width: 1200px;  
    margin: 0 auto;  
}
```

```
/* Интро*/
```

```
.intro{  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    width: 100%;  
    height: 100vh;  
    background:  
url("/app.bg") center no-repeat;  
    -webkit-background-size: cover;  
    background-size: cover;  
    text-align: center;  
  
}
```

```
.intro__inner{  
    width: 100%;  
    max-width: 880px;  
    margin: 0 auto;  
  
}
```

```
.intro__title{
  font-size: 150px;
  font-weight:700;
  text-transform:uppercase;
  line-height: 1;
}

/* Header */
.header {
  width: 100%;
  padding-top: 30px;
  position:absolute;
top:0;
  left: 0;
  right: 0;
  z-index: 1000;
}
.header__inner{
  display: flex;
  justify-content: space-between;
  align-items: center;
}

/* Logo */
.logo{
  font-size: 30px;
  font-weight:700;
  color: #b85151;
}

/* Nav */
```

```
.nav{
  font-size: 14px;
  text-transform: uppercase;
}

.nav__link{
  display: inline-block;
  vertical-align: top;
  margin: 0 15px;
  position: relative;
  color:#00B812;
  text-decoration: none;
  transition: color .2s linear;
}

.nav__link::after{
  content:"";
  display:block;
  width: 100%;
  height:3px;
  display: none;
  background-color: aqua;
  position: absolute;
  top:100%;
  left:0;
  z-index: 1;
}

.nav__link:hover{
  color:#B80200;
}

.nav__link:hover::after{
```

```
display: block;
}

/* Button*/
.btn{
display: inline-block;
vertical-align: top;
padding: 8px 30px;
border: 3px solid #b85151;
font-size: 14px;
font-weight: 700;
color: #b85151;
text-transform: uppercase;
text-decoration: none;
transition: background .1s linear, color .1s linear;
}

.btn:hover{
background-color: #ffffff;
color: aqua;
}

/* Slider*/
.slider{
width: 100%;

position: absolute;
bottom: 0;
left: 0;
z-index: 1;
```



```
}
```

```
.slider__inner {  
  display: flex;  
  justify-content: space-between;  
  
}
```

```
.slider__item{  
  width: 23%;  
  padding: 20px 0;  
  position: relative;  
  
  border-top: 3px solid #ffffff;  
  opacity: .9;  
  font-size: 18px;  
  color: #ffffff;  
  text-transform: uppercase;  
}
```

```
.slider_item.active {  
  opacity: 1;  
}
```

```
.slider__num{  
  font-size: 24px;  
  font-weight: 700;  
  color: #ffffff;
```

```
}
```

```
/* Section */
```

```
.section{  
  padding: 80px 0;  
}
```

```
.section__header{  
  width: 100%;  
  max-width: 950px;  
  margin: 0 auto 50px;  
  text-align: center;  
  
}
```

```
.section__suptitle{  
  font-size: 24px;  
  font-family: 'Kaushan Script',cursive;  
}
```

```
.section__title{  
  font-size: 30px;  
  font-weight: 700;  
  text-transform: uppercase;  
  
}
```

```
.section__title::after{  
  content: "";  
  display: block;  
  width: 60px;  
  height: 3px;  
  margin: 30px auto;  
  background-color: #22ff0a;  
}
```

```
.section__text{  
  font-size: 15px;  
  color: #999999;  
}
```

```
/*About*/
```

```
.about{  
  margin-top: 80px;  
  display: flex;  
  justify-content: space-between;  
  
}
```

```
.about__item{  
  width: 380px;  
  
}
```

```
.about__item:hover .about__image img{
  transform: translate3d(-10px, -10px, 0);
}
```

```
.about__item:hover .about__image img{
  opacity: .3;
}
```

```
.about__image{

  transition: transform .2s linear;

}
```

```
.about__image img{
  display: block;
  transition: opacity .1s linear;
}
```

```
/*Statistics*/
```

```
.statistics{
  background-color: #95e1d3;
}
```

```
.stat{
  display: flex;

}
```

```

.stat__item{
  flex: 1 1 0;
  padding: 70px 25px;

  border-left: 1px solid #b5eae0;

  text-align: center;
  color: white;
}

.stat__item:last-child{
  border-right: 1px solid #b5eae0;
}

.stat__count{
  margin-bottom: 10px;
  line-height: 1;
  font-size: 72px;
  font-weight: 700;
}

.stat__text{
  font-size: 14px;
  text-transform: uppercase;
}

/* Services*/

.services {

```

```
display: flex;
}
.services__item{
  flex: 1 1 0;
  padding-right: 35px;
  padding-left: 85px;
  position: relative;
}

.services__title{
  margin-bottom: 10px;
  font-size: 14px;
  color:#333333;
  text-transform: uppercase;
}

.services__text{
  font-size: 15px;
  color: #999;
}

/* Footer */

.footer{
  padding-top: 65px;
  background-color: #f8f8f8;
}
```

```
.footer__inner{
  padding-bottom: 65px;
}
```

```
.footer__logo{
  margin-bottom: 30px;
  font-size: 40px;
  font-weight: 700;
  color: #cccccc;
}
```

```
.footer__text{
  margin-bottom: 30px;
  font-size: 15px;
  color: #999999;
}
```

```
.footer__social{
  margin-bottom: 25px;
}
```

```
.footer__social-header{
  padding-bottom: 15px;
  border-bottom: 1px solid #e5e5e5;
  font-size: 14px;
  color: #333333;
}
```

```
.footer__social-header b {
```

```
font-size: 18px;  
}
```

```
.footer__social-content{  
font-size: 15px;  
color: #999999;  
font-style: italic;  
padding-top: 15px;  
}
```

```
.footer__social-content a{  
font-size: 18px;  
color: #95e1d3;  
}
```

```
/* Subscribe */  
.subscribe{  
width: 100%;  
max-width: 380px;  
display: flex;  
}
```

```
.subscribe__input{  
height: 14px;  
padding: 12px;  
border-right: 0;
```



```
background:#ffffff;  
border: 1px solid #e7e7e7;
```

```
font-family: 'Montserrat', sans-serif;  
font-size: 15px;  
line-height: 1.1;  
color: #333333;  
font-style: italic;  
font-weight: 300;  
}
```

```
.subscribe__input::placeholder{  
  color: #cccccc;  
}
```

```
.subscribe__input:focus{  
  outline: 0;  
  border-color: #95e1d3;  
}
```

```
.subscribe__btn{  
  width: 40%;  
  height: 40px;  
  background-color: #95e1d3;  
  padding: 12px 30px;  
  border:0;  
  cursor: pointer;  
  font-family: 'Montserrat', sans-serif;  
  font-size: 14px;  
  line-height: 1.1;  
  font-weight: 700;
```

```
color: #ffffff;  
text-transform: uppercase;  
  
transition: background .2s linear;  
}
```

```
.subscribe__btn:hover{  
  background-color: #74c7b8;  
}
```

```
.subscribe__btn:focus{  
  outline: 0;  
}
```