**ЗМІСТ**

[ВСТУП 7](#_Toc500574876)

[1. ЗАДАЧА РОЗБОРКИ СИСТЕМИ A/B ТЕСТУВАННЯ 8](#_Toc500574877)

[1.1. Мета розробки системи A/B тестування 8](#_Toc500574878)

[1.2. Застосування системи A/B тестування 9](#_Toc500574879)

[1.3. Опис підсистем системи A/B тестування 9](#_Toc500574880)

[1.3.1. Web-застосунок системи A/B тестування. 10](#_Toc500574881)

[1.3.2. API системи A/B тестування. 10](#_Toc500574882)

[1.3.3. СКБД Microsoft SQL Server. 11](#_Toc500574883)

[2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ 12](#_Toc500574884)

[2.1. Огляд роботи аналогів, їх переваг та недоліків 12](#_Toc500574886)

[3. МЕТОДИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ 16](#_Toc500574887)

[3.1. Скрапінг HTML сторінки 16](#_Toc500574888)

[3.2. Методи маркетингових досліджень 17](#_Toc500574889)

[4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ 22](#_Toc500574890)

[4.1. Архітектура системи 22](#_Toc500574894)

[4.1.1. Опис Web-інтерфейсу системи A/B тестування. 23](#_Toc500574895)

[4.1.2. Архітектура API системи A/B тестування. 24](#_Toc500574896)

[4.1.3. Опис бази даних системи A/B тестування. 24](#_Toc500574897)

[4.2. Програмні засоби розроблення системи 29](#_Toc500574898)

[4.2.1. ASP.NET Web API. 31](#_Toc500574901)

[4.2.2. Microsoft SQL Server. 32](#_Toc500574902)

[4.2.3. Angular. 34](#_Toc500574903)

[4.2.4. JQuery. 34](#_Toc500574904)

[5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ 36](#_Toc500574905)

[5.1. Системні вимоги та інсталяція програмного забезпечення 36](#_Toc500574907)

[5.1.1. Апаратні та програмні вимоги. 36](#_Toc500574908)

[5.1.2. Інсталяція програмного забезпечення. 37](#_Toc500574909)

[5.2. Сценарії роботи користувача з системою 41](#_Toc500574910)

[6. ЕКОНОМІКО-ОРГАНІЗАЦІЙНИЙ РОЗРАХУНОК 46](#_Toc500574911)

[6.1. Розрахунок трудомісткості розробки та впровадження програмного продукту….. . 46](#_Toc500574913)

[6.2. Кошторис витрат на розробку та впровадження програмного продукту 52](#_Toc500574914)

[6.3. Визначення економічного ефекту від застосування програмного продукту….. . 58](#_Toc500574915)

[6.4. Визначення ціни розробки 60](#_Toc500574916)

[6.5. Техніко-економічне обґрунтування розробки програмного продукту на основі функціонально вартісного аналізу 61](#_Toc500574917)

[ВИСНОВКИ 72](#_Toc500574918)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 73](#_Toc500574919)

# **ВСТУП**

Системи A/B тестування є потужним і надійним інструментом для маркетингового дослідження. Як відомо, в бізнесі не буває статичних станів. Підприємство повинно постійно розвиватися, щоб відповідати поточній ринковій ситуації, потребам клієнтів і власників. Щоб прибутковість проекту хоча б не падала, підприємцю необхідно постійно розширювати асортимент, збільшувати охоплення аудиторії за допомогою реклами та публікації корисного контенту, покращувати поведінкові метрики сайту і коефіцієнт конверсії.

Одним з інструментів розвитку веб-проектів є A/B-тестування. Цей метод дозволяє вимірювати переваги аудиторії і впливати на ключові показники ефективності сайту, включаючи конверсію, час перебування користувачів на сторінці, середню суму замовлення, показник відмов і інші метрики.

A/B-тестування — метод маркетингового дослідження, суть якого полягає в тому, що контрольна група елементів порівнюється з набором тестових груп, в яких один або декілька показників були змінені, для того, щоб з'ясувати, які зі змін покращують цільовий показник. Хороший приклад — це дослідження впливу колірної схеми, розташування та розміру елементів інтерфейсу на конверсію сайту.

Метод часто використовується при оптимізації веб-сторінок відповідно до заданої мети. Тестуються дві дуже схожі сторінки (сторінка А і сторінка В), які відрізняються лише одним елементом або декількома елементами (тоді метод називають A/B/N-тестування). Сторінки А і В показуються користувачам почергово в рівних пропорціях, при цьому відвідувачі, як правило, не знають про це. По закінченню певного часу або при досягненні певного статистично значимого числа показів порівнюються числові показники мети і визначається найкращий варіант сторінки.

# **ЗАДАЧА РОЗБОРКИ СИСТЕМИ A/B ТЕСТУВАННЯ**

Метою цієї роботи є побудова системи, що має можливість створювати задачі для проведення A/B тестування, а також на основі налаштувань збирати статистичні дані з web-сторінок та відображати результати у зрозумілому для користувача вигляді.

* 1. Мета розробки системи A/B тестування

Метою розробки є створення системи A/B тестування, яка буде акумулювати інформацію з сайтів для подальшого збереження та можливості експорту даних з системи агрегації до інших систем. Система має упорядковану структуру та об’єднує зібрані дані на одному ресурсі. Це дасть можливість користувачам системи отримати дані, які зберігаються на сайтах та використовувати зібрану інформацію з метою наповнення свого сайту контентом.

Система має використовувати:

* серверну технологію ASP.NET Web API [1];
* базу даних MS SQL;
* Angular [2] - Typescript фреймворк у якості клієнтської технології.

Задачі, які повинна виконувати вищеописана система:

* створення нової задачі для даної сторінки користувача;
* розпізнавання HTML [3] елементів сторінки для зручного вибору елементів сторінки;
* впровадження механізму заміни HTML елементів на сторінку користувача;
* механізм збору статистичних даних зі сторінки користувача;
* зручне відображення результатів A/B тестування.
  1. Застосування системи A/B тестування

Важливо також виділити основні задачі програмного продукту та визначити, в яких цілях він буде використовуватися користувачами.

Цільовою аудиторією системи є власники сайтів, у яких є потреба досягти максимальної продуктивності в роботі свого сайту. За допомогою системи A/B тестування користувачі зможуть визначати, яке саме наповнення їх сайту буде приносити більшу користь та прибуток.

Після того, як користувач визначиться з сайтом для обробки, створить задачі тестування та підключить на свій сайт автоматично згенерований скрипт - система починає роботу. Кожен перегляд сторінки, кожне натискання на досліджуваний матеріал буде зберігатися та аналізуватися, щоб згодом продемонструвати користувачу який контент на його сторінці буде максимально вигідним.

* 1. Опис підсистем системи A/B тестування

Таку систему можна розділити на дві незалежні підсистеми (рис. 1.1):

* система налаштування та створення задачі A/B тестування, яка дозволяє переглядати результати проведеного дослідження;
* система аналізу та збору статистичних даних, з можливістю динамічної підміни елементів.

У першій підсистемі вхідними даними є сторінка користувача, на якій будуть проводитися дослідження. Після проведення налаштувань вихідними даними, які будуть вхідними в другій підсистемі, повинні бути селектор елементу, який не буде змінюватись протягом досліджень, та два варіанти HTML коду цього елементу.

Як сказано вище, вхідними даними для другої підсистеми будуть вихідні дані першої. За допомогою Javascript [4] файлу, який користувач має розмістити на вказаній сторінці будуть проводитися дослідження. Після впровадження користувачем скрипту система повинна почати роботу. При відкриті сторінки користувача, будь-яким відвідувачем сайту, повинна відбутись заміна HTML елементу. Відображення одного елементу повинно відбутися для першої половини відвідувачів, та іншого елементу для другої половини. Таким чином, 50% користувачів будуть бачити один елемент, а інші 50% - інший.

Кожне відвідування сторінки та кожне натискання досліджуваного елементу повинні реєструватися системою та зберігатися для відображення в першій підсистемі для можливості подальшого статистичного аналізу.

* + 1. **Web-застосунок системи A/B тестування.** Web-застосунок розроблений з використанням технології Angular, а також додаткових бібліотек Javascript/Typescript, які забезпечують взаємодію користувача з системою, роблять інтерфейс більш інтуїтивно зрозумілим, а також дозволяють розробнику організовувати взаємодію серверної та клієнтської частин. Він призначений для створення завдань A/B тестування, перегляду вже створених завдань, а також отримання результатів A/B тестування.
    2. **API системи A/B тестування.** API - це серверна частина системи, інтерфейс, який дозволяє в обмеженому вигляді працювати з базою даних. Конкретніше, він має методи для взаємодії з клієнтською частиною, а також з сайтом користувача:
* створення задачі A/B тестування;
* видалення задачі A/B тестування;
* отримання результатів A/B тестування;
* реєстрація натискань та переглядів на досліджуваній сторінці користувача.
  + 1. **СКБД Microsoft SQL Server.** Система A/B тестування використовує Microsoft SQL Server — комерційну систему керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

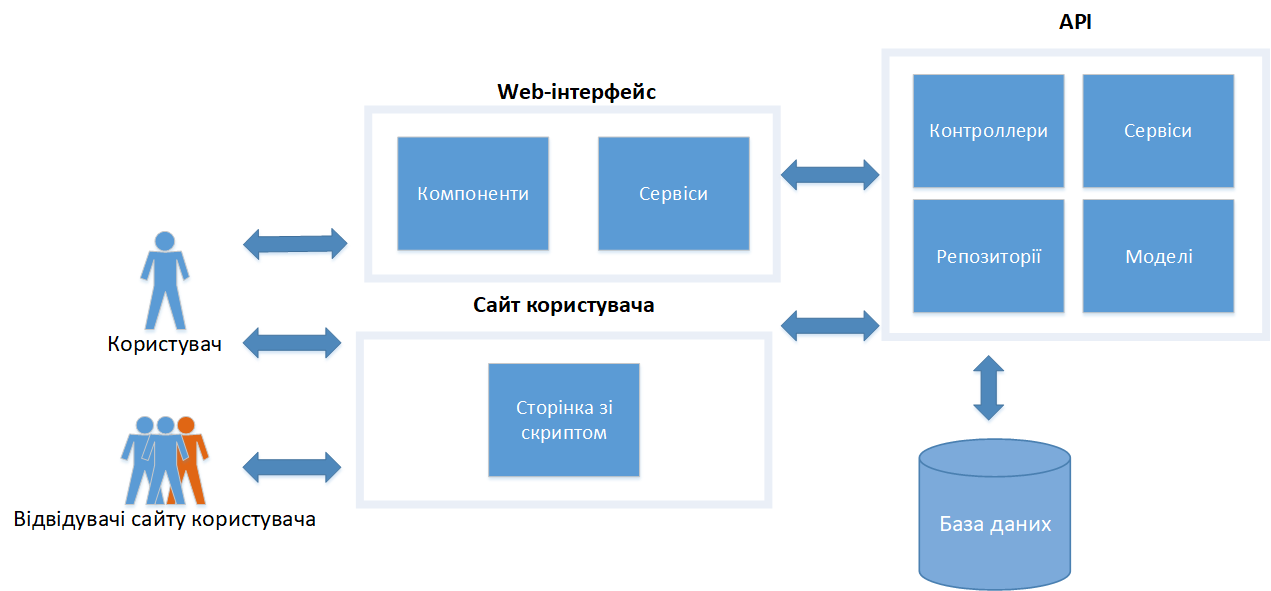


Рисунок 1.1 – Схема взаємодії компонентів системи A/B тестування

# **ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ**

В останні роки з’явилась достатня кількість програмного забезпечення, яка частково розв’язує задачу A/B тестування. Серед них такі програмні засоби:

* Convert;
* LPGenerator;
* Google Web Scraper;
* Optimizely;
* AB TASTY;
* Unbounce;
* Clickthroo.

Головним недоліком усіх програмних продуктів є неможливість обходу сайту за заданими посиланнями.

1. 1. Огляд роботи аналогів, їх переваг та недоліків

Першим популярним інструментом для A/B тестування є Convert (рис. 2.1). Сервіс дозволяє наступні види тестування: А/В, спліт-тестування, багатофакторне тестування.

Сервіс надає непоганий онлайн-редактор, в якому легко без спеціальних знань змінювати елементи сторінки для тестування. Окремо налаштовується тестування під мобільні пристрої.

Варіантів для тестування може бути безліч. Кожному можна привласнити унікальне ім'я і задати свої налаштування. При цьому, є функція копіювання попереднього варіанту для більш швидкого його редагування.

Наступним популярним інструментом є LPgenerator. Інструменти даного тестування представлені, як частина функціоналу по створенню посадкової сторінки. Сервіс дозволяє лише A / B-тестування.

Присутній онлайн-редактор, з великою кількістю шаблонів для генерації посадкової сторінки з нуля. Однак, власну сторінку редагувати в ньому не можна.

Одним з найсильніших за функціоналом сервісів для тестування є Optimizely (рис. 2.2). Підтримує А / В тестування і багатофакторний аналіз.

При створенні проекту тестування пропонується вибрати одну з трьох платформ на якій буде проводитися аналіз.

Зручний онлайн-редактор сторінки дає можливість змінювати варіанти сайту для тестування прямо в вікні сервісу.

Так само є можливість налаштування цілей тестування.

Після налаштування цілей і запуску кампанії, реалізоване відстеження тестування в режимі реального часу.

Досить потужним сервісом для тестування є AB TASTY (рис. 2.3). Доступні наступні види тестування: тестування захищених сторінок, тестування перенаправлення, багатосторінкові тестування, тунелі.

Зручна настройка сегментації і персоналізаціі. Є онлайн редактор, як графічний, так і HTML, CSS, JavaScript. Легке налаштування кампанії і наочне відстеження тестування і досягнення цілей в реальному часі. Сервіс дозволяє відслідковувати дійсно величезну кількість показників.

Наступний сервіс Unbounce - по суті є західним аналогом LPgenerator. Являє собою потужний інструмент для створення landing сторінок, який включає в себе А/В тестування. Однак, через це власний, вже готовий сайт, протестувати вийде. Є графічний редактор з величезною базою редагованих шаблонів посадкових сторінок.

Сервіс VWO має весь функціонал, який може знадобитися для тестування сайту. Види тестування: А / В, спліт-тестування, багатофакторне тестування.

Налаштування цілей, дає можливість автоматизувати тестування безлічі варіантів до досягнення певних показників. Кампанію можна налаштувати як для десктопних комп’ютерів, так і для версії на будь-якому з мобільних пристроїв. Зручний онлайн-редактор і швидке налаштування величезної кількості варіантів.

Після запуску кампанії відразу можна почати відстежувати тестування в наочних звітах.

Так само сервіс пропонує «Ідеї» - це готові набори визначені встановлення і цілі для тестування. При чому, сервіс аналізує ваш сайт і видає вам ідеї, які найбільш підійдуть саме для вашого сайту.

Наступним сервісом є сайт Clickthroo, який дозволяє тільки А/В тестування.

Є онлайн редактор і набори шаблонів. Можливо відстеження тестування в реальному часі, настройка гео-таргетингу і відстеження всієї аналітики.

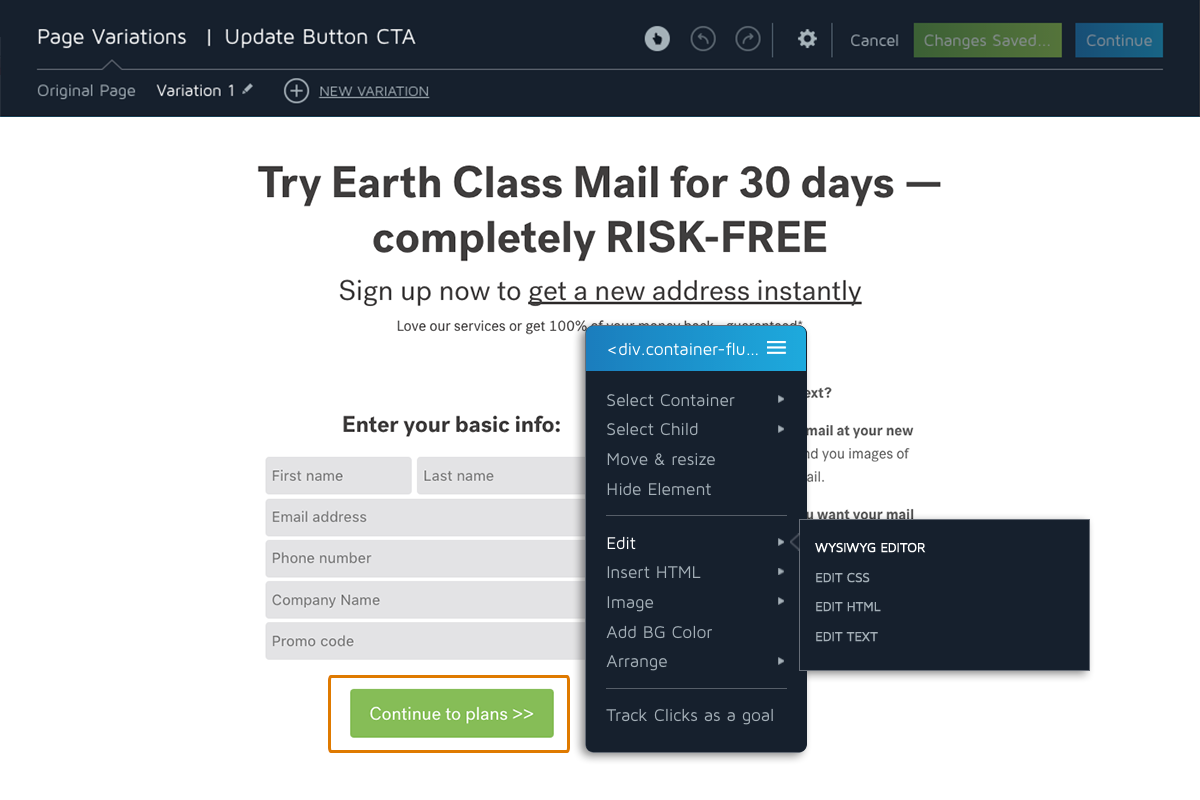


Рисунок 2.1 – Діалогове вікно налаштувань для Convert

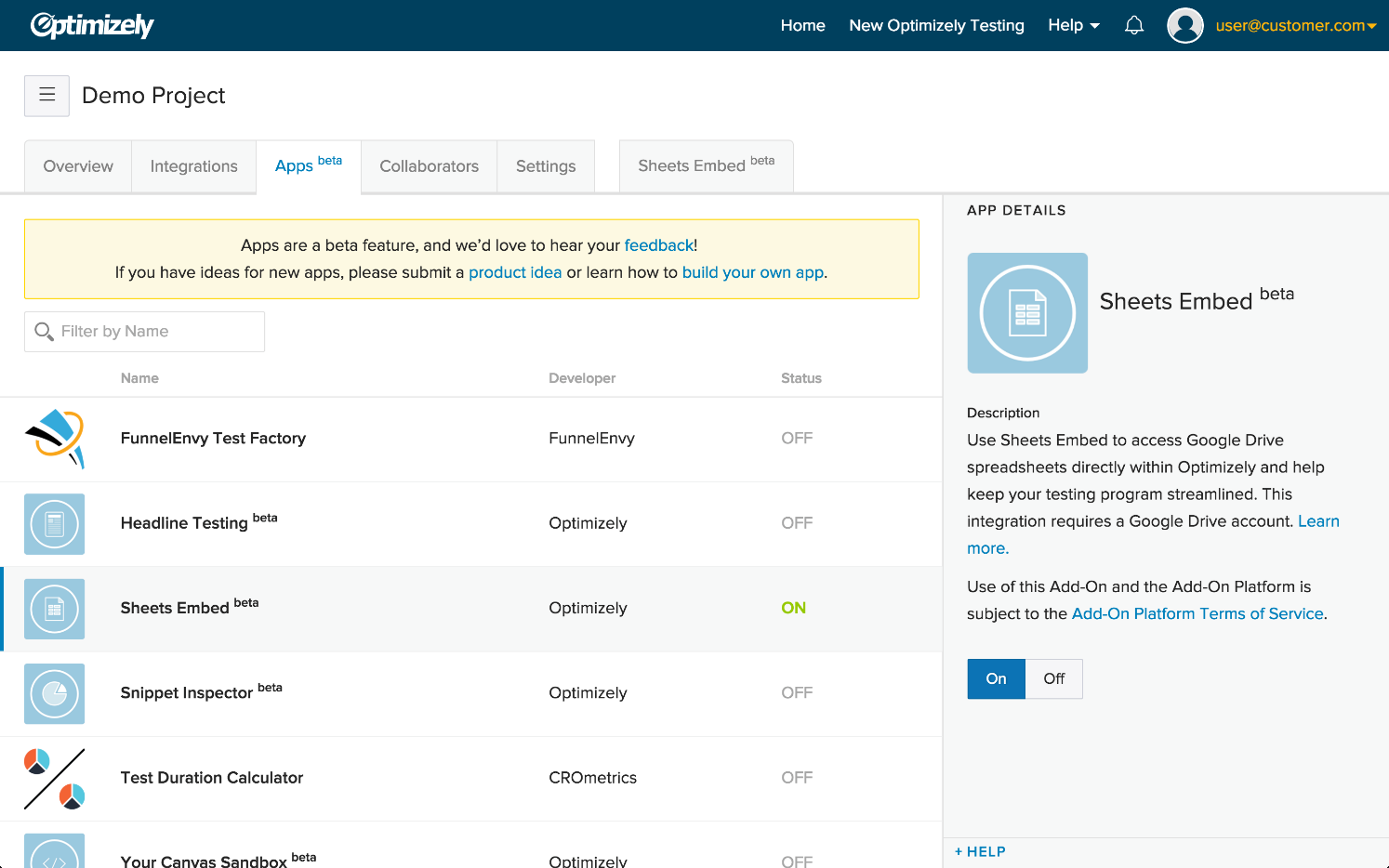


Рисунок 2.2 – Діалогове вікно налаштувань для Optimizely

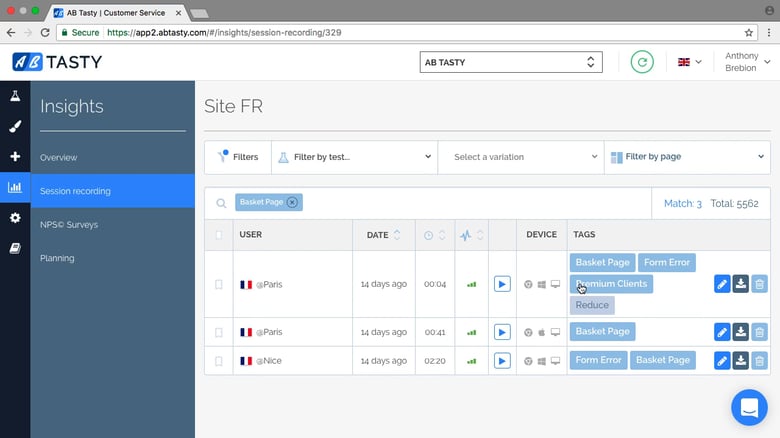


Рисунок 2.3 – Діалогове вікно налаштувань для AB Tasty

# **МЕТОДИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ**

У процесі розробки системи агрегації контенту використовувались наступні методи: методи скрапінгу HTML сторінки, методи маркетингового дослідження, зокрема метод A/B тестування.

* 1. Скрапінг HTML сторінки

Скрапінг являє собою метод вилучення інформації з веб-сайтів. Як правило, це досягається за рахунок або безпосередньо реалізації протоколу передачі гіпертексту (на якому базується Web) або за допомогою веб-браузера.

Скрапінг тісно пов'язаний з веб-індексацією, яка індексує інформацію в Інтернеті за допомогою бота і є універсальною методикою, прийнятою більшістю пошукових систем. На противагу цьому, скрапінг більше фокусується на перетворенні неструктурованих даних в мережі, як правило, в форматі HTML, в структуровані дані, які можуть зберігатися і аналізуватися в центральній локальній базі даних або таблиці. Скрапінг також пов'язаний з веб-автоматизацією, яка імітує людський перегляд інформації в інтернеті за допомогою спеціалізованого програмного забезпечення.

Існують наступні методи скрапінгу:

1. Копіювання та вставка інформації користувачем: іноді навіть найкраща технологія скрапінгу не може замінити людину, особливо, коли на сайті явно встановлений бар’єр для запобігання автоматизованого скрапінгу.
2. Регулярні вирази: простий, але ефективний підхід для отримання інформації з веб-сторінок за допомогою регулярних виразів.
3. HTTP програмування: статичні і динамічні веб-сторінки можуть бути отримані шляхом HTTP-запитів на віддалений веб-сервер.
4. HTML парсинг: багато веб-сайтів мають великі колекції сторінок, створених динамічно з використанням бази даних, як структурованого джерела. Цей метод визначає структуру сторінки та створює об’єкти, виходячи з цієї структури.
5. DOM (Document Object Model) аналіз: вбудовуючись у повноцінний веб-браузер, наприклад, Internet Explorer або Mozilla, програма може отримати динамічний контент, згенерований на стороні клієнта.
6. Веб-скрапінг: є багато інструментів, які можуть бути використані для веб-скрапінгу. Ці інструменти надають інтерфейс для того, щоб точно визначити структуру сторінки, а також зберегти вибрані структури у базі даних.
7. Аналіз веб-сторінки з використанням комп’ютерного зору або комп’ютерного бачення: є програмне забезпечення яке працює за допомогою машинного навчання і комп'ютерного зору, яке намагається ідентифікувати і отримувати інформацію з веб-сторінок шляхом інтерпретації сторінок візуально так, як це зробив би користувач.
   1. Методи маркетингових досліджень

Найважливішими методами проведення маркетингових досліджень є: спостереження, опитування, експеримент і визначена група кількісних методів.

Спостереження - це метод збирання первинної маркетингової інформації про об'єкт, що досліджується. Спостереження здійснюється за обраними групами людей, ситуаціями.

До проведення спостережень застосовують різноманітні підходи: пряме і побічне спостереження, відкрите і приховане, структуроване і неструктуроване, за допомогою людини або механічних засобів.

Пряме спостереження передбачає безпосереднє спостереження за об'єктом, наприклад за покупцем у магазині. При проведенні побічного спостереження вивчаються результати певної поведінки, а не сама поведінка.

Відкрите спостереження передбачає, що люди знають про те, що за ними спостерігають. Необхідно пам'ятати, що присутність спостерігачів певною мірою впливає на поведінку людей, за якими спостерігають, тому необхідно прагнути звести його до мінімуму.

Найбільш прийнятим в таких умовах є приховане спостереження, коли досліджуваний об'єкт не знає, що за ним спостерігають.

При проведенні структурованого спостереження наперед визначається, що підлягає спостереженню. Всі інші види спостереження ігноруються. Структуроване спостереження використовується для перевірки результатів, отриманих іншими методами, та їх уточнення.

Коли проводиться неструктуроване спостереження, спостерігач фіксує в досліджуваному епізоді всі види поведінки. Такий тип спостереження часто використовується при проведенні розвідувальних досліджень.

Наприклад, компанія, яка випускає будівельний інструмент, може направити своїх співробітників для вивчення направлень і частоти використання певного інструменту при будівництві будинків. Результати спостережень використовуються при вдосконаленні цього інструменту.

Щоб отримати необхідну й об'єктивну інформацію і не пропустити будь-яких важливих факторів, необхідно наперед сумлінно розробити план спостережень. У ньому необхідно передбачити терміни, засоби збирання інформації, відповідальних осіб.

Найбільш важливими етапами спостережень є:

* визначення цілі;
* визначення суб'єкта і об'єкта;
* вибір методів спостереження;
* підготовка технічних документів (тиражування карток, інструкцій для споживачів, підготовка письмових приладів, технічного обладнання).

У процесі реалізації плану здійснюється проведення спостережень, збирання даних, накопичення інформації. В першу чергу необхідно правильно підібрати спостерігачів. Найважливіші вимоги, що висуваються до спостерігача - добросовісність, увага, терплячість.

Фіксація результатів спостереження здійснюється у вигляді:

* короткочасного запису, що проводиться по "гарячих слідах";
* реєстрації інформації;
* щоденника, до якого систематично вносяться всі необхідні відомості;
* фото-, відео-, звукозаписів.

Контроль спостереження можна здійснювати одним з методів нижче:

* проведення розмови з учасниками ситуації;
* звернення до документів, пов'язаних з певною подією;
* повторення спостереження.

Звіт про спостереження містить:

* документацію про час, місце і обставини проведення спостереження;
* інформацію про роль спостерігача і ефективність обраних методів;
* характеристику осіб, за якими проводиться спостереження;
* власні інтерпретації спостерігача;
* оцінку надійності отриманих результатів.

Одним з найважливіших методів проведення маркетингових досліджень, поряд з експериментом або в його рамках, залишається опитування населення і спеціалістів. Шляхом опитування здійснюється систематичне збирання інформації про осіб, яких опитують. Опитування здійснюється за допомогою особистих контактів, телефону, або поштою.

Особисте опитування проводиться в процесі бесіди безпосередньо з опитуваним і дає змогу отримати найбільш обширну і достовірну інформацію. Персональне інтерв'ю має свої плюси і мінуси.

Переваги: більш репрезентативна вибірка, більшу кількість питань може бути задано, можлива оперативна перевірка. Недоліки: дорого, помилки, викликані особистістю інтерв'юера або обманом зі сторони опитуваного.

Нерідко метод опитування використовується для отримування результатів впливу реклами. Інтерв'юер, використовуючи тести на впізнання реклами, розглядаючи разом з досліджуваним спеціалістом раніше бачений ним журнал, опитує, які рекламні оголошення є знайомими. Більш достовірні дані про ефективність реклами дають тести на запам'ятовування: людина, яка заявила, що вона уже знайома з представленою їй на даний момент торгівельною маркою, або іншим елементом реклами, повинна розповісти про її зміст більш детально. Так можна оцінити якість запам'ятовування, міру її достовірності. Але і такі тести ще не дають підстав для висновків про готовність до укладання угоди, до покупки товару.

Ще один крок вперед порівняно з тестами - опитування на тему запропонованої реклами. Можливі прості питання типу: "Чи подобається вам ця реклама?", "Чи цікава вона для вас?", "Чи довіряєте ви їй?". Можливі й побічні методи. Наприклад, людину наперед просять вибрати із ряду предметів той, який їй найбільше подобається. Зміна вибору на користь про рекламованого предмета - критерій оцінки ефективності реклами.

Для оцінки дієвості реклами використовуються і проективні методи. Один з них - тест на словесні асоціації: якщо ті, або інші слова з проекту рекламного повідомлення, що надається у випадковій послідовності серед інших слів, викликають у людей невірні або небажані асоціації, то вони виключаються із реклами. Тести на сприйняття допомагають визначити ступінь зрозумілості, легкість зчитування основних змістових блоків реклами.

Відомі також тести, опитування, що мають своєю ціллю аналіз іміджу - ставлення населення до певного продукту або підприємства, що здійснює опитування. Прагнення підприємств і реальність у цьому випадку найбільш різко розходяться. Підприємства часто бувають шоковані тим, що думає населення насправді, тому такі тести і опитування найбільш повчальні.

Детально і диференційовано оцінити якість і ефективність рекламного повідомлення дає можливість анкетне опитування, до якого можна залучити як безпосередньо споживачів, так і спеціалістів- експертів.

Опитування по телефону відносно просте, дешеве і не займає багато часу.

Недоліки - певна частина населення може не мати телефону.

Найбільш повну і добротну інформацію можна отримати експериментальним шляхом. Причому, процес експерименту проводиться ніби в три етапи, рівнозначними самостійними видами опитування: експлораторне (розвідувальне), дескриптивне (описувальне), каузальне (експериментальне) дослідження.

Зазначимо, що різниця між категоріями досліджень має відносний характер, а самі категорії можна розцінити як ступені деякого єдиного процесу. Причому, кожний вид досліджень відповідає певному типу маркетингових проблем.

Отже, розглянувши методи маркетингового дослідження можна сказати, що A/B тестування задовольняє багатьом потребам сучасного бізнесу. Воно є одним з методів спостереження, при якому досліджувана група людей не знає про проведення дослідження. A/B-тест повинен мати певний результат, який піддається вимірюванню, наприклад, кількість продажів, число людей, реєстрації, кількість натискань на той чи інший елемент. Багато компаній використовують «експериментальний» підхід до прийняття маркетингових рішень. Це стає все більш поширеною практикою, і досвід росте в цій галузі. Є багато A/B-тестувань тематичних досліджень, які показують, що практика тестування стає все більш популярною також у малому і середньому бізнесі.

# **ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ**

Система A/B тестування розроблена в середовищі Visual Studio 2017 з використанням технологій ASP.NET Web API та Angular.

2. 2. Архітектура системи

Система складається з двох основних компонентів: компонент, що виконує аналіз даних, та веб-інтерфейс. Взаємодія цих компонентів та основних модулів представлена на наступній діаграмі (рис. 4.1).

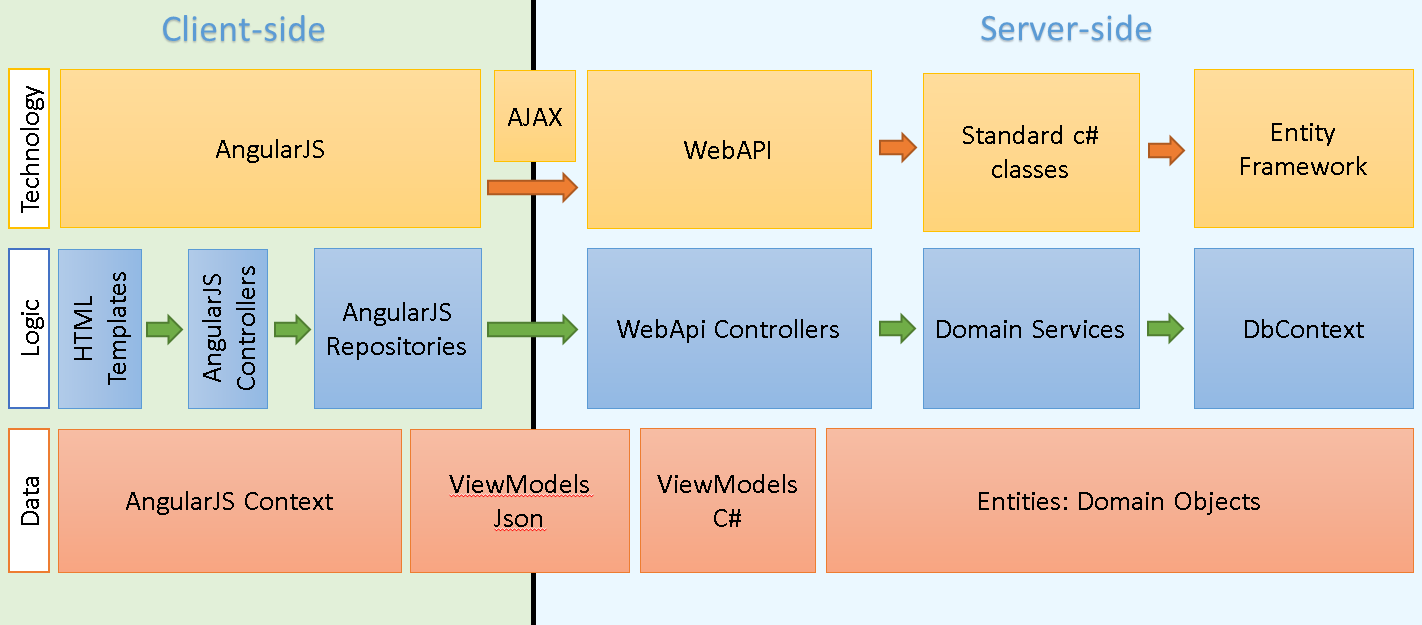


Рисунок 4.1 Архітектура та взаємодія основних модулів системи

Обидва модулі є незалежними один від одного, але й в той самий час кожен з них виконує функцію, яка відповідає за надання даних для іншого модуля.

Система включає низку модулів, які складають основний функціонал:

* “HomeController” — основний модуль системи. Відповідає за створення та налаштування задач A/B тестування.
* “AccountContoller” — модуль, що відповідає за збереження даних користувачів системи, надає можливість реєстрації та авторизації.
* “ABTestContoller” — модуль, що відповідає за механізм відображення А та В елементів на сторінці клієнта, а також сбір та аналіз дій відвідувачів сайту кліента.
* “ScrappingController” — модуль, який дозволяє отримати сторінку користувача, та відобразити її у первинному вигляді.
* “ABContext” — модуль, що надає можливість працювати з БД.

На рисунку 4.2 зображені функції користувача системи та функції самої системи.

Під час побудови задачі A/B тестування відбувається наступний процес:

Користувач надає сторінку, для якої будуть відбуватися створення налаштування задачі A/B тестування.

Система аналізує сторінку, та надає можливість створити нову задачу.

Після налаштувань, користувач зберігає задачу. Система створює всі необхідні початкові дані для початка A/B тестування.

Система генерує файл, який користувач повинен розмістити на досліджуваній сторінці.

Після того, як користувач вставив файл на свою сторінку система постійно відслідковує дії відвідувача сайту, та зберігає ці дані для подальшого відображення клієнту.

* + 1. **Опис Web-інтерфейсу системи A/B тестування.** Web-інтерфейс використовує бібліотеку Angular та Ajax технології для того, щоб користувач працював з інтерфейсом без перерв на перезавантаження сторінок при надходженні нового запиту на роботу. Для роботи з DOM використано JavaScript бібліотеку jQuery.

Задачами даного модуля є:

* формування завдання A/B тестування;
* перегляд створених задач;
* перегляд результатів A/B тестування;

За допомогою даного модулю користувач має можливість створити нове завдання для A/B тестування та дізнатися результати вже створеного раніше.

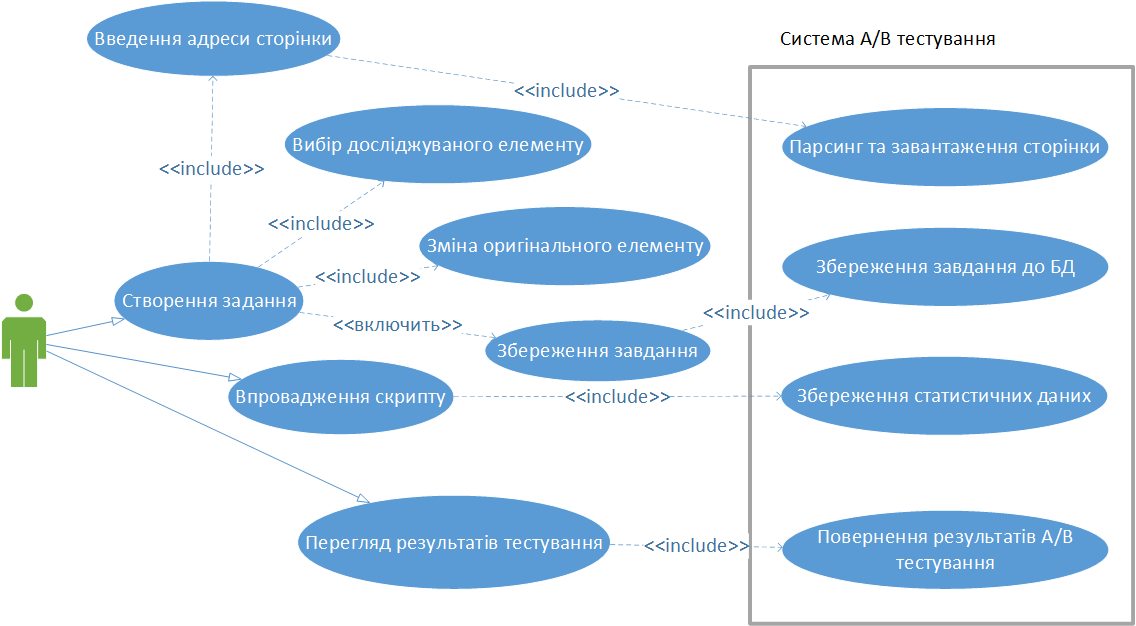


Рисунок 4.2 – Діаграма прецедентів

* + 1. **Архітектура API системи A/B тестування.** API системи A/B тестування складається з трьох модулів: модуль скрапінгу, модуль роботи с базою даних та модуль A/B тестування. Модуль скрапінгу дозволяє розпарсити сторінку для того, щоб зробити можливим відображення сайта користувача всередині сторінки налаштування A/B системи. Модуль роботи з базою даних дозволяє працювати з описаною вище описаною MS SQL базою даних.
    2. **Опис бази даних системи A/B тестування.** Базою даних для системи A/B тестування була обрана СКБД Microsoft SQL Server з таких причин:
* контроль за надлишковістю даних;
* несуперечливість даних;
* підтримка цілісності бази даних;
* цілісність описується за допомогою обмежень;
* незалежність прикладних програм від даних;
* спільне використання даних;
* підвищений рівень безпеки.

Система керування базами даних — комплекс програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

База даних системи A/B тестування з таких таблиць: ABTests, Users, UserClaims, Logins, Roles, UserRoles, UserTokens (рисунок 4.4).



Рисунок 4.3 – Архітектура API

На рисунку 4.5 представлений приклад даних записаний до таблиці “ABTests”.

Ця колекція складається з таких полей:

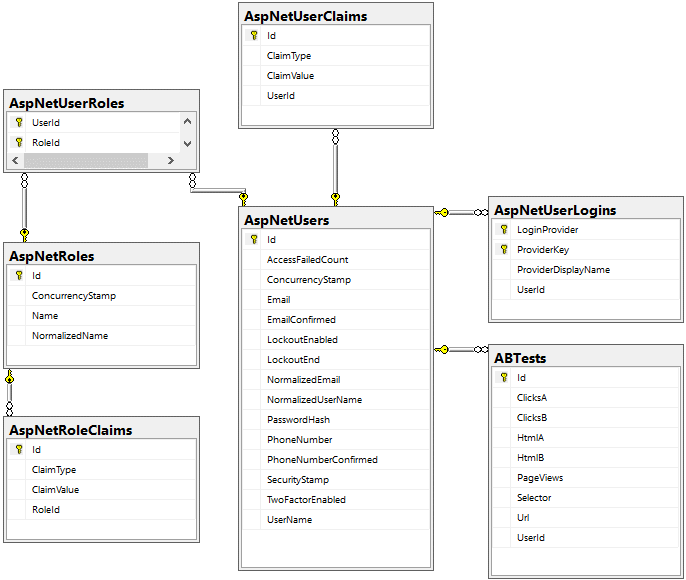
* «Id» – ідентифікатор запуску;
* «Selector» – селектор вибраного елемента;
* «Url» – посилання на сторінку, з якої отримані данні;
* «HtmlA» – html оригінального елементу;
* «HtmlB» – html зміненого елементу;
* «ClicksA» – кількість натискань на оригінальний елемент;
* «ClicksB» – кількість натискань на змінений елемент;
* «PageViews» – кількість переглядів, вибраної для A/B тестування сторінки;
* «UserId» – ідентифікатор користувача, який створив задачу A/B тестування.

Всі дані про створення нової задачі, а також уся аналітика, яка проводиться протягом A/B тестування зберігається у данній таблиці.

На рисунку 4.6 представлена таблиця, яка має назву «Users».

Ця таблиця зберігає інформацію про користувача для його аутентифікації. Вона складається з таких полів:

* «Id» – ідентифікатор користувача;
* «UserName» – логін користувача, який вводиться при вході у застосунок;
* «SecurityStamp» – згенерований код захисту, яке повинен змінюватись при будь-якій зміні даних користавача;
* «Email» – поштова адреса користувача;
* «EmailConfirmed» – поле, яке свідчить яке свідчить про факт підтвердження поштової адреса;
* «PhoneNumber» – номер телефону;
* «PhoneNumberConfirmed» – поле, яке свідчить про факт підтвердження номер телефону;
* «TwoFactorEnabled» – визначає чи увімкнена двофакторна аутентифікація;
* «LockoutEndDateUtc» – значення дати й часу, коли разблокування закінчиться; в будь-який час в минулому вважається не заблокований;
* «LockoutEnabled» – поле, яке свідчить чи заблокований користувач;
* «AccessFailedCount» – лічильник невдачних спроб логіну;
* «Roles» – список ролей, які виконує користувач;
* «PasswordHash» – пароль у захешованому вигляді;
* «Logins» – список, який містить інформацію про логіни користувача;
* «Claims» – список заявок;
* «FirstName» – ім’я користувача;
* «LastName» – прізвище користувача.

Рисунок 4.4 – Схема бази даних

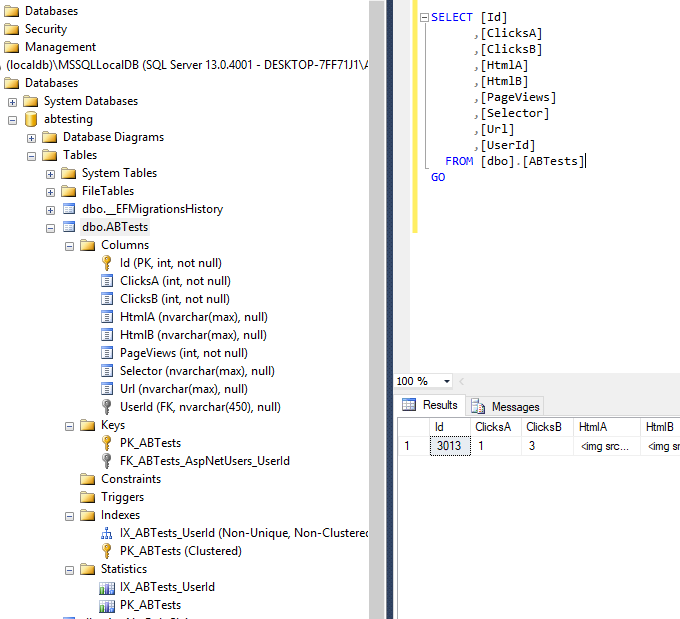
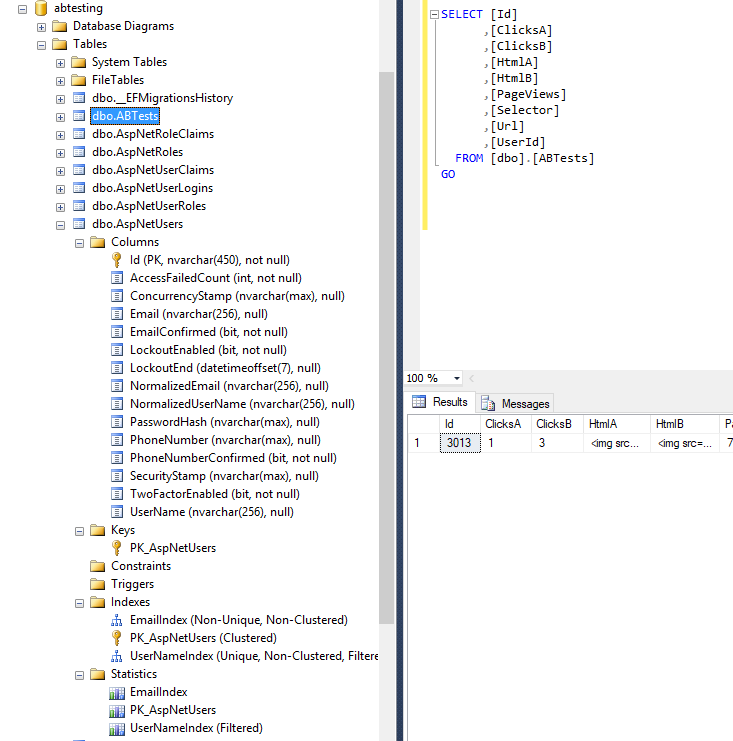


Рисунок 4.5 – Дані у таблиці “ABTests”

 Рисунок 4.6 – Дані у таблиці «Users»

* 1. Програмні засоби розроблення системи

Для написання програмного продукту використано наступні методи та засоби розробки.

Платформа ASP.NET Core собою представляє технологію від компанії Microsoft, призначену для створення різного роду веб-додатків: від невеликих веб-сайтів до великих веб-порталів і веб-сервісів.

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна.

ASP.NET Core є кросплатформенним opensource-проектом. Весь код проекту доступний на GitHub, і тому кожен може бачити реалізацію всього функціоналу цієї платформи і робити до неї зміни, якщо це буде потрібно.

Додаток ASP.NET Core може працювати з двома виконуваними середовищами: .NET Core і з повною версією фреймворка .NET [9].

.NET Core представляє модульну крос-платформену виконуючу середу, яка спрощує розгортання програми на мові C# [10]. .NET Core програми можна запускати не тільки на ОС Windows, але і на Linux і MacOS.

Завдяки модульності всі необхідні компоненти веб-додатки можуть завантажуватися як окремі модулі через пакетний менеджер Nuget.

ASP.NET Core - більш оптимізована для використання в хмарних сервісах і має вбудовану підтримку впровадження залежностей.

Angular [11] — JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з однієї HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Фреймворк працює зі сторінкою HTML, що містить додаткові атрибути і пов'язує області вводу або виводу сторінки з моделлю, яка є звичайними змінними JavaScript. Значення цих змінних задаються вручну або отримуються зі статичних або динамічних JSON-даних.

Angular спроектований з переконанням, що декларативне програмування найкраще пасує для побудови інтерфейсів користувача та опису програмних компонентів, в той час як імперативне програмування пасує для опису бізнес-логіки. Фреймворк адаптує та розширює традиційний HTML, щоб забезпечити двосторонню прив'язку даних для динамічного контенту, що дозволяє автоматично синхронізувати модель та вид. У результаті AngularJS зменшує роль DOM-маніпуляцій з метою підвищення продуктивності і спрощення тестування.

* 2. 1. **ASP.NET Web API.** ASP.NET — технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET і розвитком старішої технології Microsoft ASP. У цей час останньою версією цієї технології є ASP.NET 5.

ASP.NET зовні багато в чому зберігає схожість із старішою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі. NET і, отже, використовує всі нові можливості, що надаються цією платформою.

У якості мови програмування для реалізації системи було взято мову С#, що має підтримку усіх необхідних технік програмування. Мова С# є універсальною та високорівневою мовою програмування, придатною для створення програм будь-якої складності. Вона є однією з найбільш перспективних у світі мов програмування, яка на практиці підтвердила свою надійність та масштабованість. Ці причини є основними факторами, що визначили вибір цієї мови.

Таким чином, в огляді методів реалізації було визначено та обґрунтовано основні методи та технології, які будуть використані при розробці програмного продукту: модульне програмування на мові С# з використанням об’єктно-орієнтованої техніки, а також застосування знань з інформаційних систем.

ASP.NET MVC Framework — фреймворк для створення веб-застосунків, який реалізує шаблон Model-view-controller. Цей фреймворк доданий Microsoft в ASP.NET.

Шаблон архітектури Model-View-Controller (MVC) розділяє додаток на три основні компоненти: модель, вигляд і контролер. Платформа ASP.NET MVC являє собою альтернативу схемі веб-форм ASP.NET при створенні веб-додатків. Платформа ASP.NET MVC є платформою відображення з широкими можливостями тестування і, подібно до додатків на основі веб-форм, інтегрована з існуючими функціями ASP.NET, наприклад з головними сторінками і перевіркою достовірності на основі членства. Платформа MVC визначається в збірці System.Web.Mvc.

MVC являє собою стандартний шаблон розробки, знайомий багатьом фахівцям. Деякі типи веб-додатків мають переваги при створенні на платформі MVC. Для інших може бути доцільно використання традиційної схеми програми ASP.NET, заснованої на веб-формах і зворотного передачі. У деяких випадках можливе поєднання двох підходів: застосування однієї схеми не виключає використання іншого. Шаблон розробки MVC представлений на рисунку 4.8.

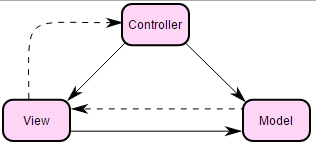


Рисунок 4.8 – Шаблон розробки MVC

* + 1. **Microsoft SQL Server.** Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву TRANSACT-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC)—інтерфейс взаємодії застосунків з СУБД. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє застосункам під керування Java з'єднуватися з Microsoft SQL Server.

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL—це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server підтримує надлишкове дублювання даних за трьома сценаріями:

* знімок: виконується «знімок» бази даних, який сервер відправляє одержувачам.
* історія змін: всі зміни бази даних безперервно передаються користувачам.
* синхронізація з іншими серверами: Бази даних декількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість вирішення протиріч між базами даних.

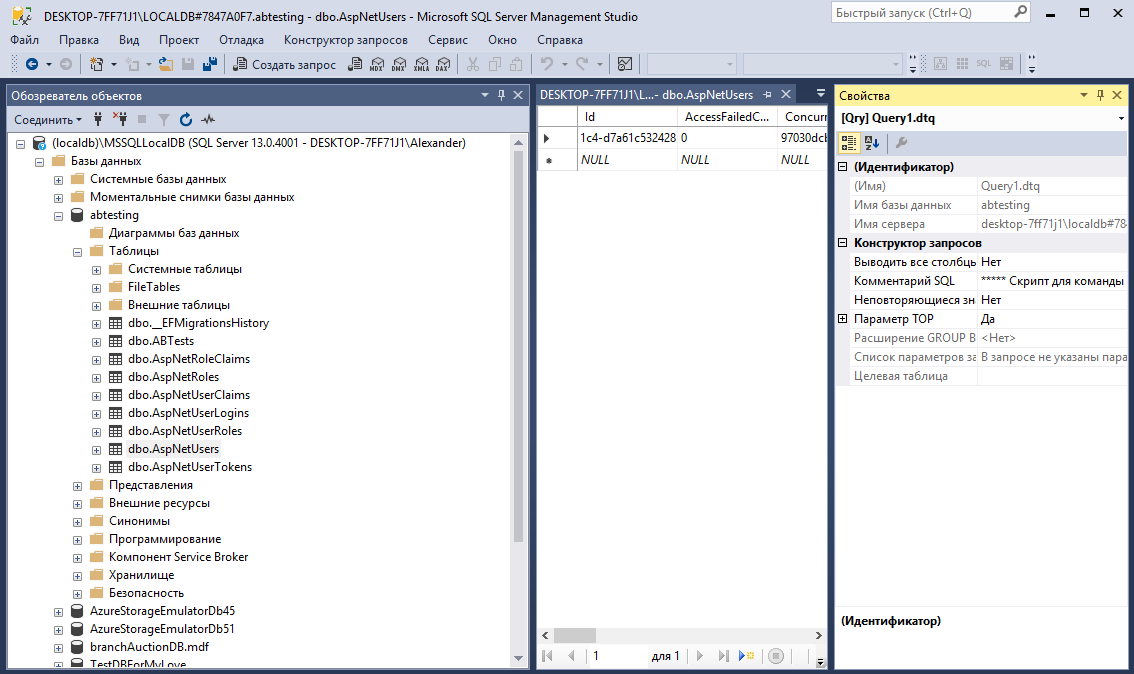


Рисунок 4.9 – Стартове вікно Microsoft SQL Server Management Studio

* + 1. **Angular.** Angular представляє фреймворк від компанії Google для створення клієнтських додатків. Перш за все він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків. В цьому плані Angular є спадкоємцем іншого фреймворка AngularJS. У той же час Angular це не нова версія AngularJS, а принципово новий фреймворк.

Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі.

Однією з ключових особливостей Angular є те, що він використовує TypeScript в якості мови програмування.

* + 1. **JQuery.** jQuery – популярна JavaScript-бібліотека з відкритим вихідним кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресіґом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

jQuery є вільним програмним забезпеченням під ліцензією MIT (до вересня 2012 було подвійне ліцензування під MIT та GNU General Public License другої версії).

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

# **МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ**

Для того, щоб почати працювати з системою A/B тестування потрібно ознайомитись з системними вимогами, а також виконати ряд операцій, щоб встановити її на сервері.

1. 1. Системні вимоги та інсталяція програмного забезпечення

Перед початком інсталяції програмного забезпечення треба впевнитись, що комп’ютер відповідає заданим системним вимогам. Вимоги формувались у відповідності до середньостатистичних, тобто таких, які відповідають можливостям більшості підприємств.

* + 1. **Апаратні та програмні вимоги.** Апаратні вимоги для коректної роботи програмного забезпечення для комп’ютера:
* процесор: Pentium з тактовою частотою не менше 1Ггц або аналогічний (рекомендується Pentium з тактовою частотою 2 Ггц або аналогічний);
* ОЗУ: не менше 256 МБ (рекомендується 1024 МБ);
* екран: 800 x 600, 256 кольорів (рекомендується 1280 x 768, high color (32-розрядний колір));
* Програмні вимоги для коректної роботи програмного забезпечення для комп’ютера:
* встановлена ОС Windows 7 / Windows 8 / Windows 10;
* .NET Framework 4.0 або вище;
* встановлена база даних MS SQL.

Система коректно функціонує на комп'ютерах з операційними системами:

* Windows Server 2008;
* Windows 7;
* Windows 8;
* Windows Server 2012;
* Windows 10.
  + 1. **Інсталяція програмного забезпечення.** Приступаючи до роботи з програмою, користувачу насамперед необхідно включити «Служби IIS» (Internet Information Services). Для цього необхідно зробити наступні дії:
* закрийте всі програми, які працюють в даний час;
* відкрийте панель керування та двічі клацніть «Установка й видалення програм»;
* на лівій панелі клацніть «Додавання та видалення компонентів Windows». Запуститься майстер компонентів Windows;
* оберіть «Служби IIS» [20] і натисніть кнопку «Далі». Якщо прапорець вже встановлений, то служби IIS вже встановлені.

Також необхідно встановити СКБД MS SQL Server:

* завантажуємо MS SQL Server з сайту (рисунок 5.1);
* запускаємо інсталяцію завантаженого файлу.

Після установки «Служб IIS» необхідно створити Web-сайт:

1. Відкрийте «Диспетчер IIS» (рисунок 5.2).
2. На панелі «Підключення» клацніть правою кнопкою миші вузол «Сайти» в дереві, потім виберіть команду «Додати веб-сайт» (рисунок 5.3);
3. У вікні зображеному на рисунку 5.4 необхідно зробити такі дії:
   1. У діалоговому вікні «Додавання веб-сайту» в полі «Ім'я веб-сайту» введіть зрозуміле ім'я веб-сайту.
   2. Клацніть пункт «Вибрати», якщо слід вибрати пул додатків, відмінний від пулу, зазначеного в полі «Пул додатків». У діалоговому вікні «Вибір пулу додатків» виберіть пул додатків в списку «Пул додатків», а потім натисніть кнопку «OK».
   3. У полі «Фізичний шлях» введіть фізичний шлях до папки веб-сайту або натисніть кнопку огляду (...), щоб вибрати папку у файловій системі. Якщо фізичний шлях, вказаний на кроці 5, веде до загального мережного ресурсу, клацніть Підключитися як, щоб вказати облікові дані, що мають дозвіл на доступ до ресурсу з цього шляху. Якщо не використовуються певні облікові дані, виберіть параметр Користувач веб-додатки (наскрізна перевірка справжності) у діалоговому вікні «Підключитися як».
   4. У списку «Тип» виберіть протокол для веб-сайту.
   5. Значення за замовчуванням в полі IP-адреса - Значення не привласнені. Якщо необхідно вказати статичний IP-адресу для веб-сайту, введіть IP-адресу в полі IP-адресу.
   6. У текстовому полі «Порт» введіть номер порту.
   7. Додатково можна ввести ім'я заголовка вузла для веб-сайту в полі «Тема вузла».
   8. Якщо не слід вносити зміни в параметри сайту і слід зробити його негайно доступним, встановіть прапорець Негайний запуск сайту.
4. Натисніть кнопку ОК.



Рисунок 5.1 – Завантаження Microsoft SQL Server з сайту

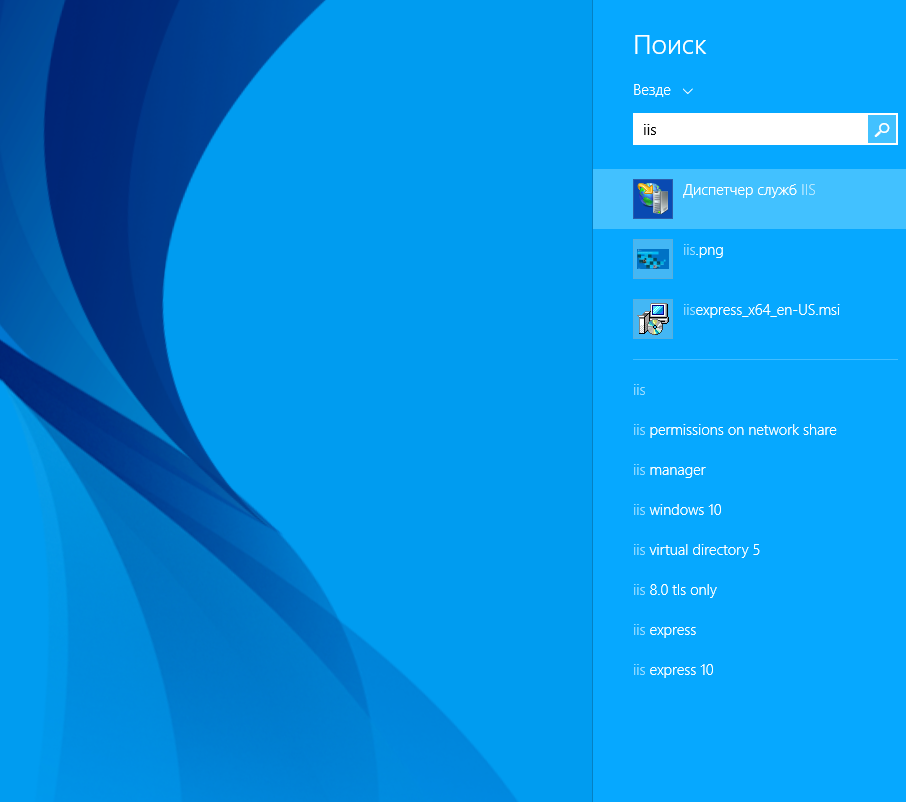


Рисунок 5.2 – Відкриття «Служб IIS»

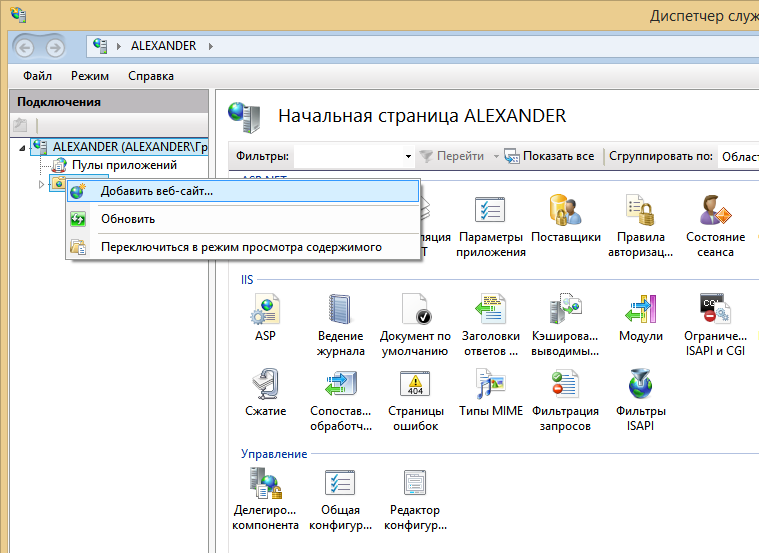


Рисунок 5.3– Додавання нового сайту

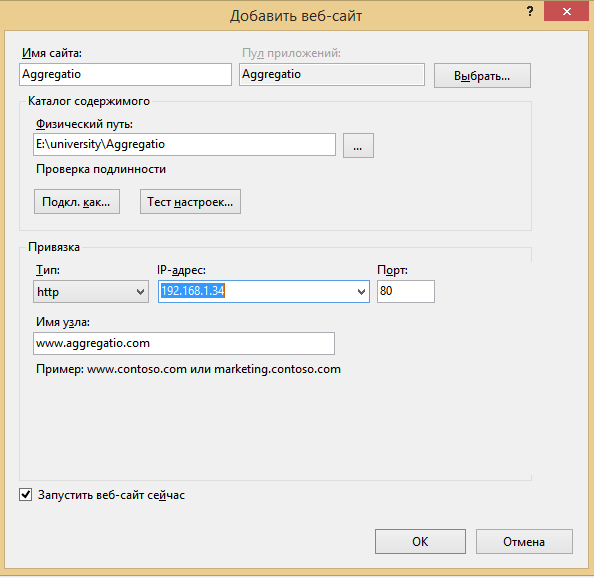


Рисунок 5.4 – Введення даних нового сайту

* 1. Сценарії роботи користувача з системою

Користувач взаємодіє з системою A/B тестування через Web-інтерфейс.

Можливості розробленої системи:

* створення завдання на A/B тестування для заданої сторінки Web-ресурсу;
* перегляд списку завдань;
* перегляд отриманих даних;
* видалення завдання з усіма отриманими даними;
* доступ до отриманих даних через API.

Сценарій роботи користувача представлено у вигляді демонстрації перерахованих вище можливостей системи.

Створення нового завдання включає в себе такі кроки:

1. Введення адреси сторінки (рисунок 5.6).
2. Вибір досліджуваного елементу (рисунок 5.7).
3. Після створення завдання користувач перенаправляється на сторінку генерації скрипта для сайту користувача (рисунок 5.8).

Після того, як користувач системи розмістить скрипт на своїй сторінці він зможе побачити результати на сторінці результатів A/B тестування(рис. 5.9).

Також користувач має можливість видаляти завдання на сторінці результатів.

Також користувач має можливість автентифікації, щоб його отримані дані булі доступні тільки йому. Перший екран, який він бачить у застосунку це логін (рисунок 5.10).

Якщо у користувача немає аккаунту то він має можливість зареєструватись, натиснувши кнопку «Register as a new user?» (рисунок 5.11).

Результати роботи зберігаються в базу даних MS SQL. Вони представлені на рисунку 5.12. Саме ці дані буде повертати API.

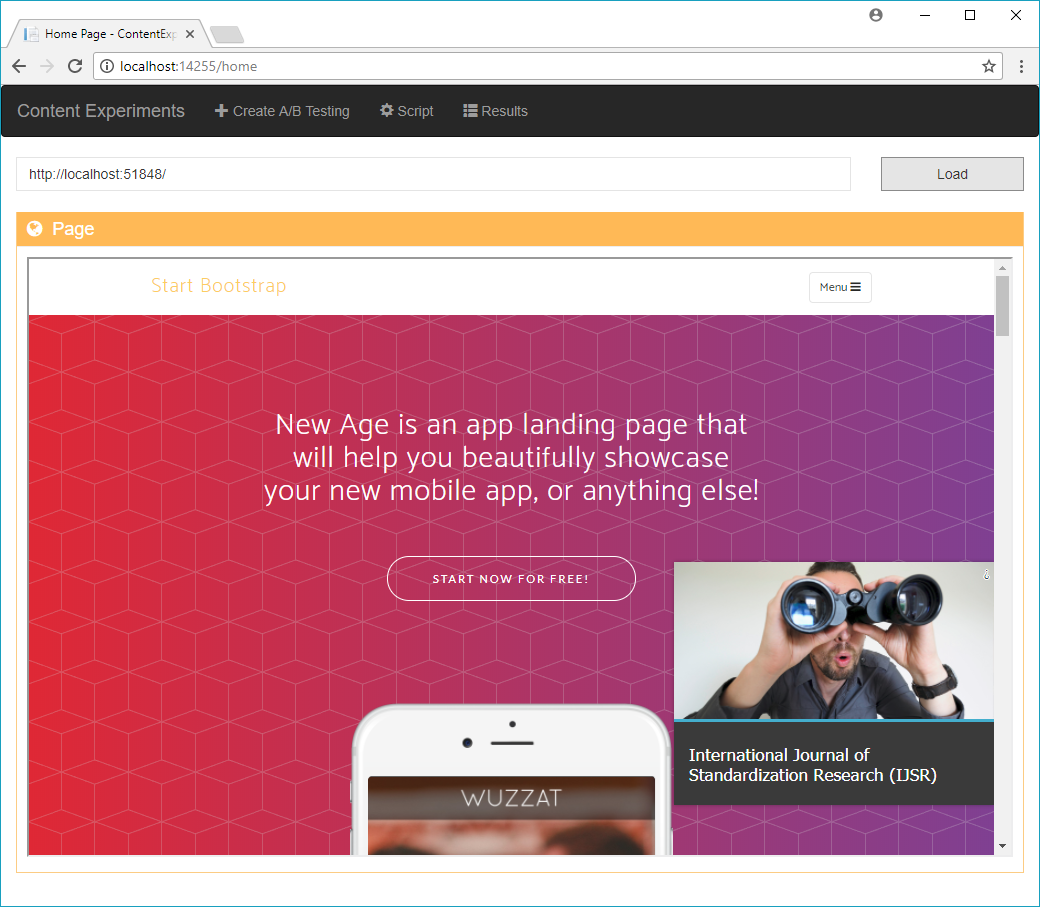


Рисунок 5.6 – Сторінка налаштування конфігурації – перший етап

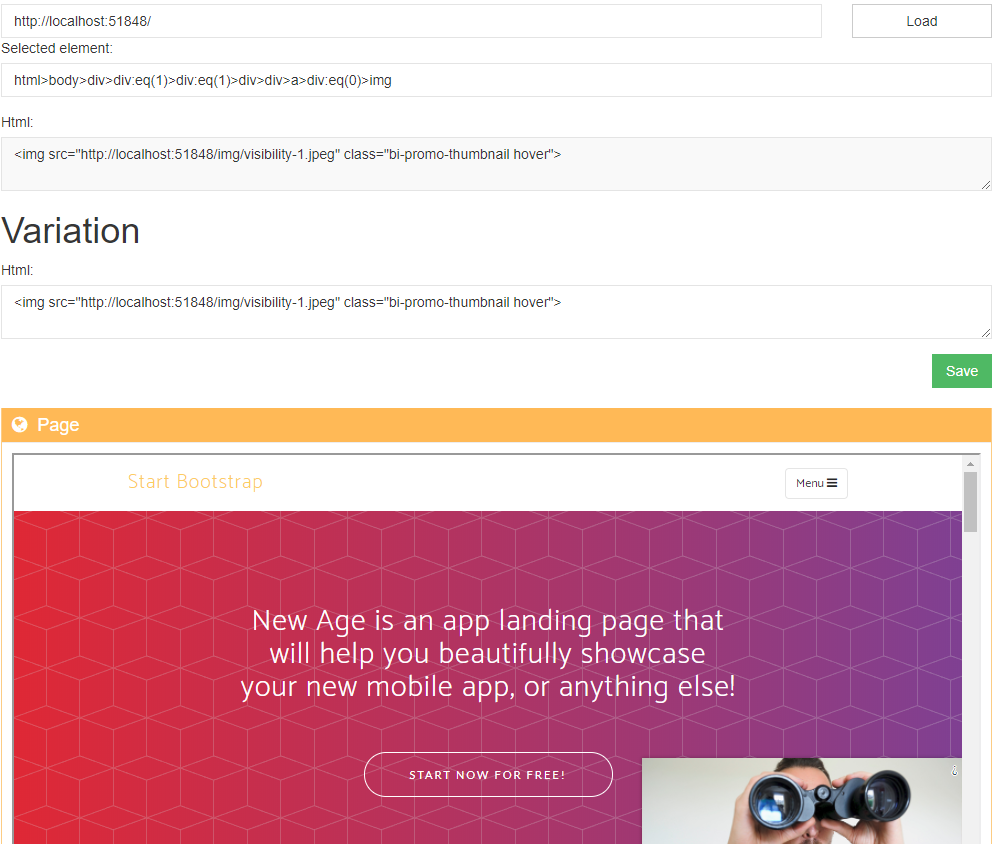


Рисунок 5.7 – Сторінка налаштування конфігурації – другий етап

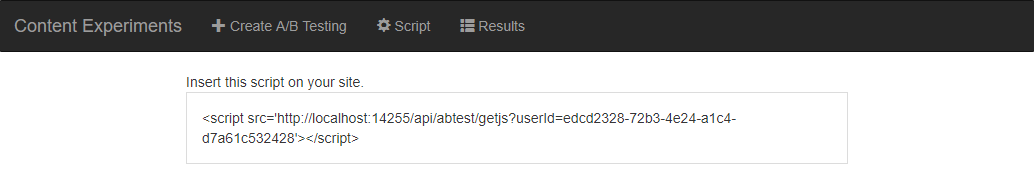


Рисунок 5.8 – Сторінка генерації скрипта для сайту користувача

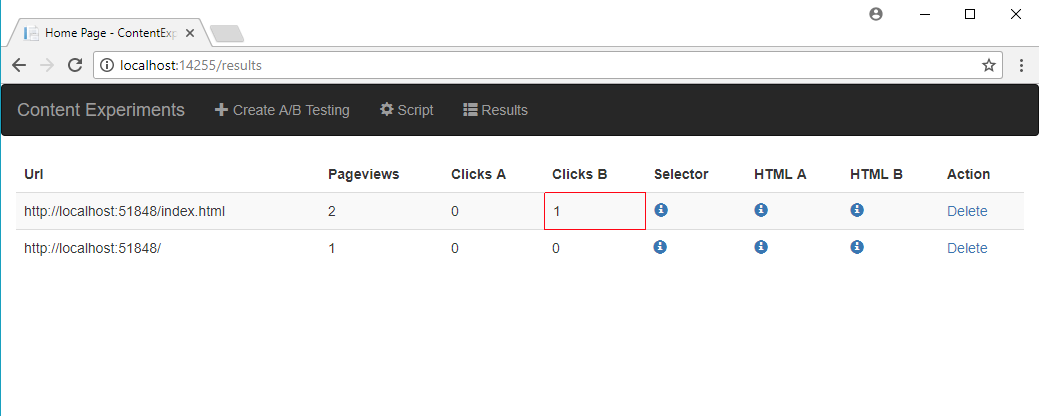


Рисунок 5.9 – Сторінка зі списком усіх завдань користувача

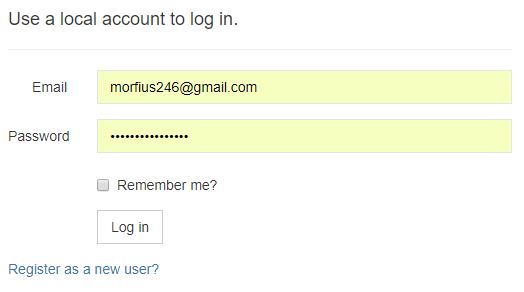


Рисунок 5.10 – Логін

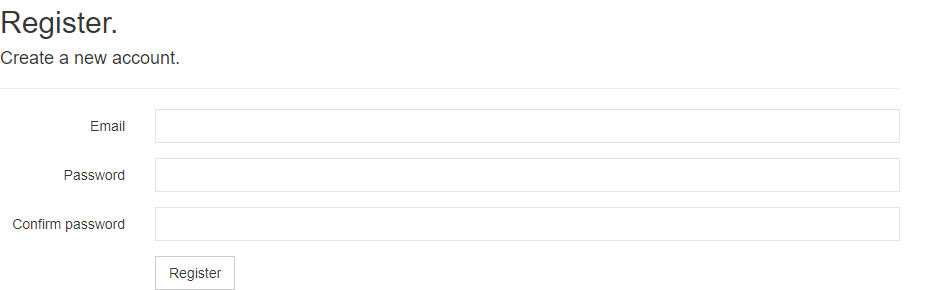


Рисунок 5.11 – Форма реєстрації

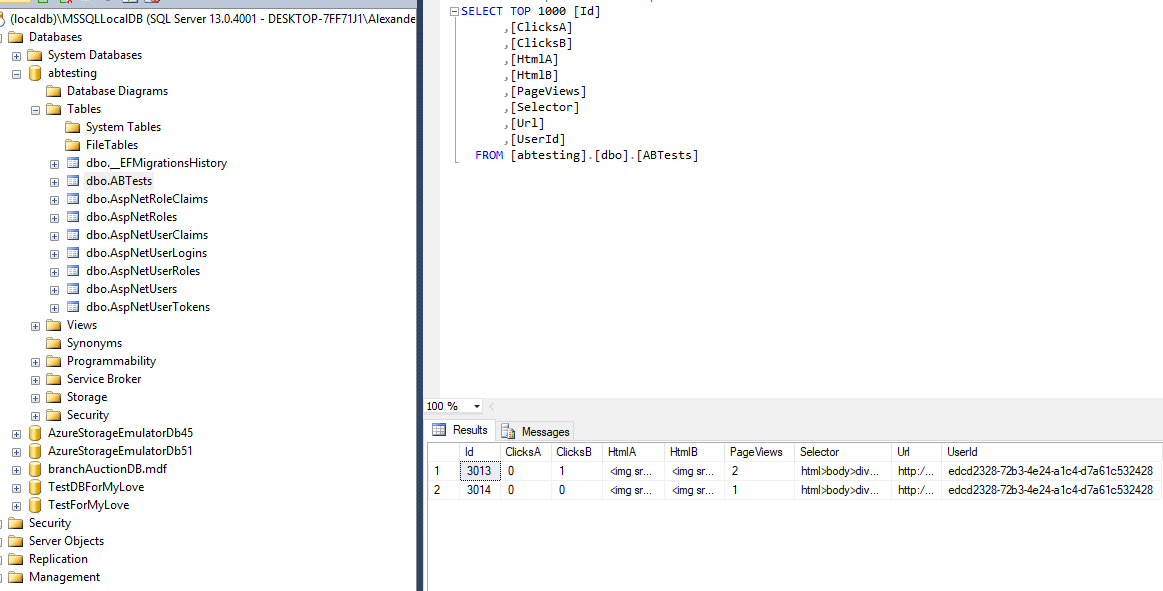


Рисунок 5.12 – Дані в базі даних MS SQL

# 

# **ЕКОНОМІКО-ОРГАНІЗАЦІЙНИЙ РОЗРАХУНОК**

1. 1. Розрахунок трудомісткості розробки та впровадження програмного продукту

Трудомісткість розробки та впровадження програмного продукту (ПП) на тему “Система A/B тестування” визначається для таких стадій розробки:

технічне завдання (ТЗ);

ескізний проект (ЕП);

технічний проект (ТП**);**

робочий проект (РП);

впровадження (Вп).

Всі ці стадії мають місце тільки при розробці дуже великих і складних програмних продуктів, у переважній більшості випадків деякі стадії можуть бути відсутні. Наприклад, може бути відсутньою стадія ЕП, тоді трудомісткість цієї стадії (ТЕП) враховується в трудомісткості ТП (Т’ТП) (формула 6.1).

(6.1)

Т’ТП = ТЕП + ТТП

Стадії ТП і РП можуть об’єднуватися в техноробочий проект (ТРП), тоді його трудомісткість складає розраховується за формулою 6.2.

(6.2)

ТТРП = 0,85 ТТП + ТРП

Трудомісткість розробки програмного продукту розраховується на основі типових норм часу на програмування.

На стадіях ТЗ та ЕП трудомісткість в людино-годинах визначається залежно від типу завдання і ступеню новизни.

Вхідні дані для економіко-організаційного розрахунку приведені в таблиці 6.1

Таблиця 6.1 — Вхідні дані

|  |  |
| --- | --- |
| Назва | Значення |
| Кількість макетів (наборів даних) вхідної інформації (банк даних) (ЗІ – 1) | 1 |
| Кількість різновидів форм вихідної інформації | 1 |
| Ступінь новизни групи завдань | «В» |
| Складність алгоритму | 2 |
| Складність організації контролю вхідної і вихідної інформації, яка характеризується такими групами | 12/22 |
| Мова програмування | С# |
| Використання стандартних модулів | 25% |
| Програмний продукт | стандартний |
| Запланований термін розробки | 0,5 року |

Для економічних завдань на стадіях “Технічний проект”, “Робочий проект” і “Впровадження” трудомісткість може бути розрахована залежно від кількості різноманітних форм вхідної і вихідної інформації за формулою 6.3.

*Tр = a\*kb\*lc,*  (6.3)

де k – кількість макетів вхідної інформації;

l – кількість різноманітних форм вихідної інформації;

а, b, с – коефіцієнти.

При використанні інформації різних видів розраховується поправочний коефіцієнт (формула 6.4).

*,*  (6.4)

де – поправочні коефіцієнти;

m – кількість наборів даних ЗІ.

Далі наведено значення поправочних коефіцієнтів для технічного і робочого проектів, враховуючи, що ступінь новизни – “В” та група складності 2.

Значення Кп для стадії “Технічний проект”:

Кп = =1

Розрахунок Кп для стадії “Робочий проект”:

Кп = = 1,1.

Поправковий коефіцієнт, який враховує складність контролю вхідної і вихідної інформації: Kск  = 1.

При використанні мов програмування низького рівня, норми часу для стадії “Робочий проект” потрібно скоригувати з урахуванням коефіцієнта Км. У нашому випадку Км = 1.

Коли при розробці ПП використовуються стандартні модулі і (або) пакети прикладних програм (ППП), типові програми, норми часу корегують за допомогою коефіцієнта Кст, значення якого залежить від процентного відношення використаних пакетів і програм, і для стадій РП і ВП приведено у таблиці 6.2.

Таблиця 6.2 – Поправочні коефіцієнти при використанні типових проектних рішень, типових програм і стандартних модулів на стадіях “Робочий проект” і “Впровадження”

|  |  |
| --- | --- |
| Ступінь використання ППП, типових програм, стандартних модулів | Кст |
| 60% і вище | 0,5 |
| 40...60% | 0,6 |
| 25...40% | 0,7 |
| 20...25% | 0,8 |

У нашому випадку поправочний коефіцієнт Кст = 0,8 при використанні типових програм і стандартних модулів із ступенем використання ПП 25% на стадіях “Робочий проект”, “Впровадження”.

При розробці стандартного ПП норму часу слід коректувати за допомогою коефіцієнта Кст.п рівного 1,2.

Коефіцієнти для розрахунку трудомісткості розробки ПП на стадіях “Технічний проект”, “Робочий проект”, “Впровадження” приведені в   
таблиці 6.3.

Таблиця 6.3 – Коефіцієнти для розрахунку трудомісткості розробки ПП на стадіях “Технічний проект”, “Робочий проект”, “Впровадження”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Комплекс завдань | Стадія розробки | Для розробника постановки задачі | | | Для розробника програмного забезпечення | | |
| a | b | c | a | b | c |
| Управління науково-технічною інформацією | ТП | 23,08 | 0,42 | 0,39 | 10,23 | 0,46 | 0,23 |
| РП | 8,35 | 0,43 | 0,44 | 54,90 | 0,37 | 0,40 |
| Вп | 8,30 | 0,47 | 0,37 | 8,09 | 0,45 | 0,41 |

Тр(ТП)= 23,08 \* 10,42 \* 10,39 + 10,23 \* 10,46 \* 10,23 = 33,31 люд./дні.

Тр(РП)= 8,35 \* 10,43 \* 10,44 + 54,9 \* 10,37 \* 10,40 = 63,25 люд./дні.

Тр(Вп)= 8,3\*10,47 \* 10,37 + 8,09 \* 10,45 \* 10,41 = 16,39 люд./дні.

Загальна трудомісткість програмування завдань визначається за формулою 6.5:

*Тз = Тр \* Кn \* Кск \* Км \*Кст \* Кст.п* (6.5)

Тз (ТЗ) = 61 \* 1,2 = 73,2 люд./дні;

Тз (ЕП) = 72 \* 1,2 = 84,6 люд./дні;

Тз (ТП) = 33,31 \* 1 \* 1 \* 1 \* 0,8 \* 1,2 =31,98 люд./дні;

Тз (РП) = 63,25 \* 1,1 \* 1 \* 1 \* 0.8 \* 1,2 = 66,79 люд./дні;

Тз (Вп) = 16,39 \* 1 \* 0,8 \* 1,2 = 15,73 люд./дні;

Тз = 73,2 + 84,6 + 31,98 + 66,79 + 15,73 = 272,3 люд./дні.

ТТРП = (ТЕП + ТТП) \* 0,85 + ТРП = (84,6 + 31,98) \* 0,85 + 66,79 = 165,88 люд./дні.

Кореговане значення Тз:

Тз = Тз(ТЗ) + ТТРП + Тз(Вп) = 73,2 + 165,88 + 15,73 = 254,81 люд./дні.

Кількість людей, які беруть участь в розробці, розраховується за формулою 6.6:

, (6.6)

де R – кількість людей;

Тз – загальна трудомісткість (людино-дня);

n – кількість робочих днів в одному році, n = 228;

T – період розробки, в роках, Т = 0,5.

R = 254,81 / (228 \* 0,5) = 3 (людини).

Це керівник роботи, програміст та інженер-впровадник.

У таблиці 6.4 приведений розрахунок трудомісткості розробки ПП групи завдань “Планування і контроль виконання робіт”.

Таблиця 6.4 – Розрахунок трудомісткості розробки ПП для всіх стадій

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Найменування | Стадії розробки | | | | | | | | | |  | |
| ТЗ | ЕП | ТП | | РП | | Вп | | | Всього | |
| Постановка завдання | Розробка програм | Постановка завдання | Розробка програм | Постановка завдання | | Розробка програм |
| Трудомісткість, людино-дни | 61 | 72 | 33,31 | | 63,25 | | 16,36 | | | 245,92 | |
| в т.ч.  постановка задачі | 16 | 12 | 3,31 | - | 3,25 | - | 6,36 | | - | - | |
| Розробка програм | 45 | 60 | - | 30 | - | 60 | - | | 10 | - | |
| Поправочні коефіцієнти на:  Види інформації Кп | - | - | 1 | | 1,1 | | - | | | - | |
| Складність контролю інформації Кск | - | - | 1 | | 1 | | 1 | | | - | |
| Мова програмування Км. | - | - | - | | 1 | | - | | | - | |
| Використання стандартних модулів Кст | - | - | - | | 0,8 | | 0,8 | | | - | |
| Розробку стандартних ПП Кст.п | 1,2 | 1,2 | 1,2 | | 1,2 | | 1,2 | | | - | |
| Скоригована трудомісткість | 73,2 | 84,6 | 31,98 | | 66,9 | | 15,73 | | | 272,3 | |
| Трудомісткість з урахуванням об’єднання стадій розробки | 73,2 | 165,88 | | | | | | 15,73 | | | 254,81 | |
| Кількість працівників | 1 | 2 | | | | | | 1 | | | 3 | |
| Тривалість розробки, років | 0,08 | 0,35 | | | | | | 0,07 | | | 0,5 | |

* 1. Кошторис витрат на розробку та впровадження програмного продукту

Кошторис витрат розробляється виконавцем робіт на основі нормативів трудомісткості розробки і впровадження програмного продукту і затверджується замовником робіт або органом, який забезпечує фінансування робіт. Витрати, які включаються в собівартість ПП, групуються відповідно до їх економічного змісту за такими статтями:

* витрати на оплату праці;
* відрахування на соціальні заходи;
* витрати на спеціальне устаткування;
* витрати на службові відрядження;

Проведемо розрахунок Кошторису витрат за вищезазначеними статтями.

Витрати на оплату праці. До цієї статті витрат належать витрати на виплату основної і додаткової заробітної плати виконавців, обчислені згідно системам оплати праці, які прийняті в організації, включаючи всі види матеріальних і грошових доплат.

Основна заробітна плата розраховується на основі даних про трудомісткість робіт, і посадових окладів основних виконавців. Інформацію про трудомісткість окремих стадій знаходиться у таблиці 6.5.

Таблиця 6.5 – Трудомісткість виконання робіт.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Стадія | Трудомісткість люд/дні | | | Всього, люд/дні |
| Керівник роботи | Програміст | Інженер- впровадник |  |
| ТЗ | 73 | - | - | 73 |
| ТРП | 17 | 149 | - | 166 |
| Вн | - | **-** | 16 | 16 |
| Всього | 90 | 149 | 16 | 255 |

Заробітну плату визначають, виходячи з місячних окладів, враховуючи тривалість умовного місяця (21.1 – при 5-денному робочому тижні). Результати розрахунків основної заробітної плати виконавців знаходиться в таблиці 6.6.

Таблиця 6.6 – Основна заробітна плата виконавців.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Посада | Місячна ставка, грн | Денна зар. плата, грн | Трудомісткість, люд-дні | Основна зар. плата, грн |
| Керівник роботи | 15000 | 710,90 | 90 | 63981 |
| Програміст | 12000 | 568,72 | 149 | 84739,28 |
| Інженер-впровадник | 9000 | 426,54 | 16 | 6824,64 |
| Всього | | | 255 | 155544,92 |

Основна заробітна плата = 155544,92 грн.

Додаткова заробітна плата (оплата відпусток, премії, одноразові заохочення і тому подібне) розраховується згідно нормативу, який встановлює підприємство і який складає в нашому випадку = 20%.

Додаткова заробітна плата = 0,2 \* 155544,92 = 31108,98 грн.

Сума основної і додаткової заробітної плати складає витрати за статтею “Заробітна плата” або фонд оплати праці.

Фонд оплати праці = 155544,92 + 31108,984 = 186653,9 грн.

Відрахування на соціальні заходи. До цієї статті належать витрати, здійснювані в порядку і розмірах, передбачених законодавством України за 2017 рік:

* на обов’язкове державне пенсійне страхування – 32%;
* на обов’язкове соціальне страхування – 4%;
* на страхування на випадок безробіття – 1,5%.

Всього страхування – 37,5% від фонду оплати праці.

Пенсійне страхування = 0,32 \* 186653,9= 59729,25 грн.

Соціальне страхування = 0,04 \* 186653,9 = 7466,16 грн.

На випадок безробіття = 0,015 \* 186653,9= 2799,81 грн.

Всього страхування = 0,375 \* 186653,9= 69995,21 грн.

Матеріальні витрати. До цієї статті належать витрати на папір, канцелярське приладдя, картриджі і інші витратні матеріали. Ці витрати в середньому складають 3% від основної заробітної плати.

Матеріальні витрати = 0,03 \* 155544,92 = 4666,35 грн.

Витрати на спеціальне устаткування та на службові відрядження не передбачені.

Експериментально-виробничі витрати. До експериментально виробничих витрат відносять оплату машинного часу, пов’язаного з підготовкою і налагодженням програм. Витрати розраховуються, виходячи з кількості годинника машинного часу, необхідного для виконання потрібного об'єму обчислюваних робіт по темі і вартості однієї машинної години.

Кількість годин машинного часу для економічних завдань розраховується за формулою 6.7.

*Тр = a \* kb \* lc*, (6.7)

де k – кількість макетів вхідної інформації;

l - кількість різноманітних форм вихідної інформації;

а, b, с – коефіцієнти.

k = 1; l = 1; а = 8,16; b = 0,76; с = 0,57.

Тр = 8,16 \* 1 0,76 \* 1 0,57 = 8,16 годин.

Результати розрахунків приведені в таблиці 6.7.

Таблиця 6.7 – Витрати на оплату машинного часу.

|  |  |  |  |
| --- | --- | --- | --- |
| Роботи, які виконуються на ЕОМ | Тривалість виконання робіт, години | Вартість однієї машино-години, грн. | Сума витрат, грн. |
| Написання програмного коду | 6,16 | 4,8 | 29,57 |
| Тестування програми | 2 | 4,8 | 9,6 |
| Разом | | | 39,17 |

Накладні витрати. Витрати на цією статтею охоплюють витрати на оплату праці управління персоналу з нарахуваннями, оплату службових відряджень, консультаційно-інформаційних витрат, ремонт і техобслуговування інших основних фондів, окрім ПК, оренду приміщення і ін. Ці витрати розраховуються за нормативом, встановленим на підприємстві у відсотках до заробітної плати, які можуть представляти від 50 до 100% (для КПІ – 67%), або 20% від статті 11 “Повна вартість роботи, виконаної власними силами”.

У нашому випадку витрати по цій статті не передбачені.

Прибуток. Прибуток визначається у відсотках від суми витрат. Прибуток складає 10% від суми витрат (формула 6.8).

*Св = (Фзп + Свсего + Ззмат + Зекс)* (6.8)

Св = (186653,9 + 69995,21 +4666,35 +39,19) = 261354,65 грн.

Прибуток=0,1 \* Св = 0,1 \* 261354,65 = 26135,465 грн.

Податок на прибуток (30% від прибутку):

Податок на прибуток = 0,3 \* 26135,465 = 7840,64 грн.

Загальні витрати. Загальні витрати обчислюються як сума витрат і прибутку (формула 6.9).

*З = Пр + Под + Св* (6.9)

З = 26135,465 +7840,64 +261354,65 = 295330,755 грн.

Податок на додану вартість (ПДВ). ПДВ обчислюється у розмірі 20% від загальних витрат (формула 6.10).

*ПДВ = З \* 0.2* (6.10)

ПДВ = 0,2 \* 295330,755 = 59066,151 грн.

Повна вартість роботи, виконаної власними силами. Повна вартість роботи обчислюється як сума загальних витрат і ПДВ (формула 6.11).

*Сума загальних витрат = ПДВ + З* (6.11)

Сума загальних витрат = 59066,151 + 295330,755= 354396,906 грн.

Загальні підсумки витрат зводяться в Кошторис вартості роботи і приведені в таблиці 6.8.

Таблиця 6.8 – Кошторис вартості роботи.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(вигляд, тема роботи і номер реєстрації)

Джерело фінансування\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Замовник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Співвиконавці\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Термін виконання робіт: почало\_\_\_\_\_\_\_\_\_\_, закінчення\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Стаття витрат | Норматив | | Сума  грн | | Питома вага статті %% |
| 1.Заробітна плата, всього  зокрема: основна  додаткова |  | | 186653,9 | | 52,67 |
| 155544,92 | | 43,89 |
| 31108,98 | | 8,78 |
| 2.Відрахування на соц. заходи  зокрема:  на обов’язкове державне пенсійне страхування  на обов’язкове соціальне страхування  на страхування на випадок безробіття | 37.5% | | 69995,22 | | 19,75 |
| 32,0% | | 59729,25 | | 16,86 |
| 4,0% | | 7466,16 | | 2,11 |
| 1,5% | | 2799,81 | | 0,79 |
| 3. Матеріали |  | | 4666,35 | | 1,32 |
| 4. Спеціальне устаткування | Не передбачені | | | | |
| 5. Відрядження | Не передбачені | | | | |
| 6. Експериментально-виробничі витрати |  | 39,17 | | 0,02 | |
| 7. Накладні витрати | Не передбачені | | | | |
| 7а. Сума витрат |  | | 261354,65 | | 73,75 |
| 8. Прибуток  зокрема податок на прибуток | 10% від п.7а | | 26135,47 | | 7,38 |
| 30% від п.8 | | 7840,64 | | 2,22 |
| 9. Загальні витрати |  | | 295330,76 | | 83,34 |
| 10. ПДВ | 20% від п.9 | | 59066,15 | | 16,67 |
| 11. Повна вартість роботи, виконаної власними силами |  | | 354396,91 | | 100 |
| 12. Договірна ціна |  | | 354396,91 | | 100 |

Дата складання Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

кошторису Економіст\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Визначення економічного ефекту від застосування програмного продукту

Річний економічний ефект від використання ПП розраховується за формулою 6.12.

*ЕР = Е - ЕН \* К*,(6.12)

де ЕР – річна економія від функціонування ПП;

ЕН – нормативний коефіцієнт ефективності капіталовкладень (для програмного забезпечення може складати 0,33);

К – об’єм інвестицій, пов’язаних із створенням і впровадженням ПП.

Об’єм інвестицій складається з витрат на розробку і впровадження ПП і вартості придбання і монтажу технічних засобів, необхідних для використання ПП.

Економія витрат праці на рішення завдань із застосуванням ПП розраховується за формулою 6.13.

, (6.13)

де m – кількість типів завдань, які вирішує даний ПП;

C1i і С2i – витрати на рішення і-того типу завдань без ПП і із застосуванням ПП;

Qi – кількість завдань і-го типу, які розв’язуються протягом року.

У загальному випадку значення C1i і С2i складаються з витрат на оплату праці користувачів ПП і на оплату машинного часу (формула 6.14).

|  |  |
| --- | --- |
| *Сji =[Tji \*cji\*(1+нд/100)\*(1+нвід/100)\*(1+ннв)] + +Тмчji\*cмч]\*(1+нПДВ/100),* | (6.14) |

де Tji – витрати часу на вирішення завдань без ПП (j=1) і із застосуванням ПП (j=2), годин;

cji – погодинна заробітна плата фахівця, який вирішує задачу і-го типу без ПП (j=1) і з ПП (j=2), залежно від складності завдання і кваліфікації;

Нд, Нвід, Ннв, Нпдв – нормативи відповідно додаткової зарплати, відрахувань на соціальні заходи. Накладних витрат і нормативу ПДВ;

Нд=30%;

Нвід=37,5%;

Ннв=67%;

Нпдв=20%;

Tмчji – витрати машинного часу на вирішення завдань без ПП (j=1) і з ПП (j=2), годин;

Смч – вартість одної годинни роботи ЕОМ.

У таблиці 6.9 приведені розрахунки економічного ефекту (розрахунок ведеться 5-денним робочим тижнем з 8-годинним робочим день).

При розрахунках використані такі дані: 21.1 робочих днів в місяць, 8 годинний робочий день. Вартість однієї години роботи на ЕОМ = 4,8 грн.

Даний програмний продукт передбачений для побудови дидактичних зв’зяків між статтями онлайн-енциклопедії, тобто вирішує лише одну задачу.

Таблиця 6.9 – Вхідні дані для розрахунку економічного ефекту.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Час на розвиток задачі | | Витрати машинного часу | | Кількість виконавців | | Заробітна плата | | Кількістьзадач в рік |
| без ПП | з ПП | без ПП | з ПП | без ПП | з ПП | без ПП | з ПП |
| 40 | 12 | 40 | 12 | 3 | 1 | 53,3 | 53,3 | 20 |

Розрахунок за формулою 6.14:

С11 = (((40 \* 3 \* 53,3 \* (1 + 0,3) \* (1 + 0,375) \* (1 + 0,67)) + 40 \* 4,8) \* (1 + 0,2) = (19092,85 + 192) \* 1,2= 23141,82 грн/година.

С12 = (((12 \* 1 \* 53,3 \* (1 + 0,3) \* (1 + 0,375) \* (1 + 0,67)) + 12 \* 4,8) \* (1 + 0,2) = (1909,28 + 57,6) \* 1,2= 2360,25 грн/година.

Економія витрат за формулою 6.13:

Е = (23141,82 - 2360,25) \* 20 = 415631,28грн/рік.

К = 354396,91 грн.

Річний економічний ефект рівний за формулою 6.12:

Ер= 415631,28 - 0,33 \* 354396,91= 298680,3 грн.

* 1. Визначення ціни розробки

Серед різних методів ціноутворення на ранніх стадіях проектування ПП достатньо поширений метод лімітних цін. При цьому визначаються нижня і верхня межа ціни.

Нижня межа ціни. Захищає інтереси розробника ПП і передбачає, що ціна повинна покрити витрати розробника, пов'язані з розробкою і впровадженням ПП, і забезпечити прийнятний рівень рентабельності, не нижче за те, що він має при впровадженні вже розроблених ПП .

Нижня межа ціни розраховується за формулою 6.15.

*ЦНМ = Сп\*(1 + рн/100)\*(1 +нПДВ/100),* (6.15)

де Сп – повна собівартість розробки ПП;

Нпдв - норматив ПДВ, 20%;

рн - нормативний рівень рентабельності, 5%.

Цнм = 354396,91 \* (1+0,05) \* (1+0,2) = 446540,11 грн.

Верхня межа ціни. Захищає інтереси споживача і визначається тією ціною, яку споживач згоден заплатити за продукцію з кращою для нього якістю. Верхня межа може бути розрахований, виходячи з рівня якості ПП (Кяк) або на базі економічного ефекту від впровадження нового ПП.

Верхня межа ціни розраховується за формулою 6.16.

*ЦВМ = Сп + Kе \* Е \*Т,* (6.16)

де Сп – повна собівартість розробки та впровадження ПП, грн;

Ке – доля економічного ефекту, яка залишається в розпорядженні розробника (найчастіше – 0,3);

Е – очікуваний економічний ефект від застосування ПП;

Т – термін застосування програмного продукту, роки.

Цвм = 354396,91 + 0,3 \* 298680,3 \* 5= 802417,36 грн.

* 1. Техніко-економічне обґрунтування розробки програмного продукту на основі функціонально вартісного аналізу

Вихідні положення. Формулюються задачі вдосконалення ПП (в процесі його модернізації) або вимоги, яким повинен відповідати новий ПП. Визначається ціль ФВА відносно цього ПП. Якщо ПП, що досліджується, є частиною складної системи, то необхідно привести структурну модель системи і визначити місце і значення елемента, який буде досліджуватися, відносно системи. Якщо ПП, що досліджується, є складною системою, то для проведення ФВА можна взяти більш простий елемент (модуль) системи. При цьому необхідно в роботі привести структурну модель всієї системи і обґрунтувати вибір даного елемента як об’єкта дослідження.

В основі ФВА лежить функціональний підхід, згідно з яким об’єктом аналізу є не сам ПП, а функції, які він виконує. ФВА проводиться в два етапи:

* функціональний аналіз;
* вартісний аналіз.

Обґрунтування функцій об’єкту. На основі вивчення організаційно-економічної суті об’єкту, досліджується (в нашому випадку програмний продукт), обґрунтовуються основні функції, які буде реалізовувати об’єкт (наприклад обробка інформації, зберігання інформації та інше). При цьому будують функціональну модель об’єкту (рисунок 6.1).

**Головна функція**

**F0** – Проведення A/B тестування

**Основні функції**

**F1** – Завантаження скрипту

**F2** – Підміна елементів

**F3** – Збір статистичної інформації

**F4** – Збереження та аналіз данних зв’язків

Рисунок 6.1. — Функціональна модель

Головна функція реалізує ціль розробки. Основні функції – ті, заради яких об’єкт створюється. Кожна з основних функцій може мати декілька варіантів реалізації. Функції і їх варіанти повинні бути чітко і повно описані. Вони використовуються для опису морфологічної карти (рис. 6.2)

На основі цієї карти виконують якісну оцінку варіантів. Для обмеження кількості варіантів, що аналізуються, будують позитивно-негативну матрицю, в якій оцінюють переваги і недоліки варіантів рішень. Результати аналізу наведено у таблиці 6.10.

За даними таблиці 6.10 проводиться порівняльний аналіз всіх можливих варіантів реалізації функцій ПП. Варіанти, які мають суттєві недоліки, не відповідають з різних причин умовам технічного завдання, виключаються з подальшого розгляду. А з варіантами, які залишаються, буде проводитись технічна і економічна порівняльна оцінка.

**Функції**

F1

F2

F3

F4

**Варіанти реалізації**

а) Повне завантаження як окремий етап

б) Завантаження частинами як окремий етап

в) Завантаження у режимі реального часу

а) Паралельний аналіз даних під час завантаження

а) Засобами сховища даних

б) Програмними засобами

а) У режимі реального часу

б) Як окремий етап

Рисунок 6.2. Морфологічна карта

Таблиця 6.10 – Позитивно-негативна матриця варіантів реалізації функцій.

|  |  |  |  |
| --- | --- | --- | --- |
| Основна функція | Варіанти реалізації | Переваги | Недоліки |
| F1 | а | Найоптимальніший спосіб | Найбільш складний у реалізації |
| б | Може базуватися на попередніх розробках | Недостатня швидкість |
| в | Легко реалізувати | Недостатня швидкість |
| F2 | а | Забезпечує високу швидкість виконання | Потребує потужне апаратне забезпечення |
| F3 | a | Найшвидше виконання | Потребує велику кількість оперативної пам’яті |
| б | Більш легко реалізувати | Більш низька продуктивність |
| F4 | a | Більш легко реалізувати | Дещо нижча продуктивність |
| б | Дещо вища продуктивність | Необхідність виконати весь процес заздалегідь, що потребує багато часу |

Обґрунтування системи параметрів ПП і оцінка його рівня якості. Основні положення з вибору і обґрунтування системи параметрів ПП, які будуть використовуватися для розрахунків рівня якості виробу і сама методика проведення розрахунків рівня якості викладена.

Для характеристики ПП, який розробляється використовуються наступні параметри:

* X1 – об’єм пам’яті на жорсткому диску, який займає встановлений программний продукт, Кб;
* X2 – потреби в об'ємі оперативної пам’яті, який необхідний для роботи програми, Мб;
* X3 – час, що витрачається на завантаження скрипту, с;
* X4 – час, що витрачається на побудову зв’язків, с;
* X5 – результат побудови, доля одиниці;
* X6 – можливість розширення, доля одиниці.

Коефіцієнт вагомості варіанта реалізації функції обчислюється за формулою 6.17.

, (6.17)

де *bі -* вагомість і-го параметра за результатами оцінок експертів, і обчислюється за формулою 6.18.

, (6.18)

де *аij* - коефіцєнти переваги, дані усіма експертами по і-му параметру.

Результати розрахунку вагомості параметрів наведені у таблиці 6.11.

Таблиця 6.11 – Вагомість параметрів.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Х1 | Х2 | Х3 | Х4 | Х5 | Х6 | bi | φі |
| Х1 | 1 | 1,5 | 0 | 0,5 | 0,5 | 0 | 3,5 | 0,127 |
| Х2 | 0,5 | 1 | 0,5 | 1 | 0,5 | 0 | 3,5 | 0,127 |
| Х3 | 0 | 1 | 1 | 0 | 0,5 | 0,5 | 3 | 0,109 |
| Х4 | 1,5 | 1 | 0 | 1 | 1,5 | 1,5 | 6,5 | 0,236 |
| Х5 | 1,5 | 1,5 | 1,5 | 0,5 | 1 | 1,5 | 7,5 | 0,273 |
| Х6 | 0 | 0 | 1,5 | 0,5 | 0,5 | 1 | 3,5 | 0,127 |
| Сума |  | | | | | | 27,5 | 1 |

Результати розрахунку показників технічного рівня варіантів виконання функцій наведені у таблиці 6.12.

Таблиця 6.12 – Показники технічного рівня варіантів виконання функцій.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Основ-на функ-ція | Варіант реалізації функції | Параметри, які приймають участь в реалізації | Абсолютне значення параметра | Оцінка параметра в балах, bi | Коефіцієнт вагомості. φі | Показник технічного рівня, Ктр (Fi) |
| F1 | а | Х1 | 6,75 | 7 | 0,127 | 0,889 |
| Х2 | 1,75 | 2 | 0,236 | 0,472 |
| б | Х1 | 9,67 | 10 | 0,127 | 1,270 |
| Х2 | 2,52 | 3 | 0,236 | 0,708 |
| в | Х1 | 11,23 | 11 | 0,127 | 1,397 |
| Х2 | 0,89 | 1 | 0,236 | 0,236 |
| F2 | а | Х3 | 8,78 | 7 | 0,127 | 0,889 |
| Х4 | 1,64 | 2 | 0,236 | 0,472 |
| F3 | а | Х4 | 2,03 | 2 | 0,273 | 0,546 |
| Х5 | 0,57 | 1 | 0,127 | 0,127 |
| б | Х4 | 1,57 | 2 | 0,273 | 0,546 |
| Х5 | 0,65 | 1 | 0,127 | 0,127 |
| F4 | а | Х6 | 11,18 | 11 | 0,236 | 2,596 |
| б | Х6 | 6,75 | 7 | 0,236 | 1,652 |

На основі порівняльного аналізу варіантів реалізації функцій за їх перевагами і недоліками і коефіцієнтів вагомості параметрів вибираємо наступні варіанти реалізації функцій:

1. F1б + F2а + F3а + F4а.
2. F1а + F2а + F3 + F4а.

Показник рівня якості k-того варіанта реалізації основних функцій виробу розраховується за формулою 6.19.

*Ктрк = Ктр(F1k) + Ктр(F2k) + ... + Ктр(Fzk)* (6.19)

де Ктр(F1k) – показник технічного рівня першої функцій k-того варіанту реалізації основних функцій виробу.

 = (1,270 +0,708)+(0,889+0,472)+( 0,546+0,127)+ 2,596 = 6,608.

 = (1,397+ 0,236)+(0,889+0,472)+( 0,546+0,127)+ 2,596 = 6,263.

Найкращим на етапі функціонального аналізу є варіант, якому відповідає найбільше значення коефіцієнта технічного рівня (формула 6.20).

(6.20)

де Kmpj – коефіцієнт технічного рівня j-того варіанта реалізації основних функцій виробу.

Як видно з розрахунків, кращим є перший варіант, у якого коефіцієнт технічного рівня має максимальне значення рівне 6,608.

Методика проведення вартісного аналізу. Аналіз функцій, які реалізуються програмним продуктом, повинен бути доповнений вартісним аналізом. Для цього розраховуються витрати, які необхідні для розробки ПП, тобто визначається функціонально-необхідна вартість виробу по всіх варіантах реалізації, які досліджуються. Якщо проводиться ФВА робіт, які мають науково-дослідний характер (наприклад, створення програмного продукту), то функціонально-необхідні витрати визначаються шляхом розрахунку кошторису витрат на проведення науково-дослідних робіт. Методика розрахунку кошторису витрат на НДДКР.

Далі наведено укрупнену методику визначення функціонально-необхідних витрат на розробку програмного продукту.

Функціонально необхідні витрати на створення ПП (Сф) визначаються за формулою 6.21.

*Сф = Сз + Свід+ См + Сн,* (6.21)

де Сз – оплата праці розробників, грн.;

Свід – відрахування на соціальні заходи (37,5% від фонду оплати праці), грн.;

См – вартість машинного часу, необхідного для розробки і налагодження ПП, грн.;

Сн – накладні витрати в розмірі 50-150% від витрат на оплату праці, грн.

Опис вхідних даних до формули 6.21:

* Сз – основна заробітна плата розробників, становить 186653,9 грн.;
* Свід – 37,5% від фонду оплати праці грн., становить 69995,21 грн.;
* См – вартість машинного часу становить 39,17 грн.;
* Сп – накладні витрати в розмірі 50-150% від витрат на оплату праці, грн., витрати за цією статтею не передбачені.

Розрахунок згідно формули 6.21:

= 186653,9 + 69995,21 + 39,17 + 0 = 256688,28 грн.

Вартісний аналіз варіантів реалізації функцій завершується визначенням коефіцієнта техніко-економічного рівня кожного варіанта (Kтерj), який розраховується за формулою 6.22.

, (6.22)

де Kтерj – коефіцієнт технічного рівня j-того варіанту;

Cфj – величина функціонально-необхідних витрат j-того варіанту, грн.

 = 9,535/8140,08= 1,2 • 10-3 ;

 = 9,027/8140,08= 1,1 • 10-3 ;

 = 8,827/8140,08= 1,0 • 10-4.

Найкращий варіант визначається за максимальним значенням коефіцієнта техніко-економічного рівня.

= 1,0 • 10-3 .

На заключному етапі проведення ФВА, якщо з’явиться можливість визначити базовий варіант, то бажано встановити область ефективного використання найбільш ефективного і базового варіантів. Для цього розраховують зведені витрати по базовому і новому варіантах (Зв) за формулами 6.23 і 6.24.

, (6.23)

, (6.24)

де ,  – річні зведені витрати відповідно по базовому і новому варіантах, грн.;

, – питомі поточні витрати на одиницю продукції відповідно по базовому і новому варіантах, грн.;

, – питомі капітальні витрати відповідно по базовому і новому варіантах, грн.;

 – витрати на розробку виробу, визначаються за кошторисом витрат, грн.;

 – нормативний коефіцієнт ефективності;

 – річні продуктивні можливості нового варіанту (наприклад, кількість розрахунків у рік певної задачі за допомогою ПП), шт.

Розрахунок за формулами 6.23 та 6.24:

 = (40 \* 3 \* 53,3 \* 1,3 \* 1,375 \* 1,67 + 40 \* 4,8) \* 1,2 = 23141,82 грн.;

= (12 \* 53,3 \* 1,3 \* 1,375 \* 1,67 + 12 \* 4,8) \* 1,2 = 2360,25 грн.;

= ;

= ;

= 261354,65 / 20 = 13067,73 грн.;

= 261354,65 / 20 = 13067,73 грн.;

 = 0,33;  = 0,33;

 = 261354,65 грн.;

 = 20 шт., = 20 шт.;

= (23141,82 + 0,33 \* 13067,73) \* 20 =   
= 549083,41 грн.;

= (2360,25 + 0,33 \* 13067,73) \* 20 + 0,33 \* 261354,65 = 219700 грн.

Основні техніко-економічні показники базового і нового варіантів зведені у таблиці 6.13.

Таблиця 6.13 – Основні техніко-економічні показники.

|  |  |  |  |
| --- | --- | --- | --- |
| Показник | Одиниці виміру | Варіанти | |
| Базовий | Новий |
| Питомі поточні витрати на одиницю продукції | Гривні | 23141,82 | 2360,25 |
| Питомі капітальні витрати | Гривні | 13067,73 | 13067,73 |
| Коефіцієнт ефективності | Безрозмірний | 0,33 | 0,33 |
| Річні продуктивні можливості | Штуки | 20 | 20 |
| Витрати на розробку виробу | Гривні | 261354,65 | 261354,65 |
| Річні зведені витрати | Гривні | 549083,41 | 219700 |

Область ефективного використання варіантів зображено на рисунку 6.3.



Рисунок 6.3 – Область ефективного використання

Отже з проведеного аналізу видно, що найкращим є варіант реалізації програмного продукту при завантаженні повного об’єму даних енциклопедії в окремому етапі (при цьому виконуючи паралельну обробку даних), вибірці множини зв’язаних з цільовою статей засобами сховища даних, а також пошук фактів і побудова зв’язків у режимі реального часу. Саме цей варіант і було реалізовано у даній роботі.

Як видно з рисунку 6.3 базовий варіант доцільно використовувати лише у випадках, коли задача використання виникає менше ніж 14 разів, інакше рекомендовано використовувати новий варіант програми.

# **ВИСНОВКИ**

У ході виконання даної роботи було успішно розроблено систему A/B тестування, що дозволяє створювати та налаштовувати задачі A/B тестування, відслідковувати дії відвідувачів сайту користувача та зберігати статистичні дані у вигляді результатів A/B тестування. Було застосовано новітні та високопродуктивні інструменти розробки, які значно прискорюють роботу користувацького інтерфейсу, а також час відклику від API [12]. Система має простий користувацький інтерфейс, який дозволяє користувачам без додаткових знань користуватися системою.

Розроблено систему аналізу сторінки користувача, та можливість зручного вибору HTML елементу без додаткових технічних знань.

Розроблено механізм відображення A та B елементів на сторінці користувача та механізм збору статистичних даних.

Розглянуто переваги використання даної системи у сучасних бізнес процесах, наведено приклади існуючих систем, надано опис методів маркетингового дослідження, зокрема методу A/B тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Troelsen A. Pro C# 5.0 and the .NET 4.5 Framework / A. Troelsen. – NY.: Apress, 2012. – 1463p.
2. Лабберс П. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development / П. Лабберс, О. Брайан, С. Френк., 2011. – 272 с.
3. Sharp J. Microsoft Visual C# 2012 Step by Step (Step By Step (Microsoft)) / J. Sharp. – C.: O’Reilly, 2013. – 775p.
4. Freeman A. Pro ASP.NET MVC 4 / A. Freeman. – NY.: Apress, 2012. – 717p.
5. Skonnard A. Essential XML Quick Reference / Aaron Skonnard., 2002. – 430 с. – (Addison-Wesley).
6. Sai Srinivas S. JavaScript and JSON Essentials / Sriparasa Sai Srinivas., 2013. – 452 с. – (Amazon).
7. API [Електронний ресурс] – Режим доступу до ресурсу: - http://www.asp.net/web-api.
8. Шевчук Ф. jQuery / Фнтон Шевчук. – Харків, 2012. – 122 с. – (Nix solutions).
9. Рихтер Д. CLR via С#. Программирование на платформе Microsot .NET Framework 4.0 на языке С#. 3-е изд / Д. Рихтер. — СПб.: Питер, 2012. — 928с.
10. Miner D. MapReduce Design Patterns / D. Miner, A. Shook., 2012. – 250 с. – (O'Reilly Media).
11. Professional Microsoft IIS 8 / K.Schaefer, J. Cochran, S. Forsyth, D. Glendenning., 2013. – 947 с. – (John Wiley & Sons).