

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Медіасистем та технологій
Рівень вищої освіти другий (магістерський)
Спеціальність 186 Видавництво та поліграфія
Тип програми Освітньо-професійна
Освітня програма Технології електронних мультимедійних видань
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ _____
(підпис)

« 28 » жовтня 2019 р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Кондратьєву Олексію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів інтеграції інтернет-магазинів з системою ІС:Підприємство

Затверджена наказом по університету від 25 жовтня 2019р. № 1550 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 грудня 2019 р.

3. Вихідні дані до роботи

Інструменти обміну даними між веб-ресурсами та системою ІС:Підприємство
Технології обміну даними між веб-ресурсами та системою ІС:Підприємство
Алгоритмізація роботи системи

4. Перелік питань, що потрібно опрацювати в роботі

Вступ; Аналіз існуючих аналогів; Опис обміну даними між ІС і інтернет-магазином;
Планування дослідження; Експериментальна частина; Економічна частина; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1) Вступ; 2) Актуальність роботи; 3) Задачі, що необхідно вирішити; 4) Огляд існуючих методів синхронізації; 5) Огляд існуючих аналогів; 6) Алгоритмізація роботи системи; 7) Побудова свого модулю обміну даними; 8) Економічна частина; 9) Висновки

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Вовк О.В.		
Економічна частина	зав. каф. ЕК Полозова Т.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання на атестаційну роботу	28.10-30.10.2019	викон
2	Аналіз стану проблеми обміну даними між ІС:Підприємством та WEB-ресурсом	31.10- 06.11.2019	викон
3	Проведення теоретичних досліджень	07.11-14.11.2019	викон
4	Проведення експериментальних досліджень	15.11- 22.11.2019	викон
5	Економічна частина	23.11-28.11.2019	викон
6	Оформлення пояснювальної записки	29.11-05.12.2019	викон
7	Оформлення графічної частини	06.12-09.12.2019	викон

Дата видачі завдання 28 жовтня 2019 р.

Студент

(підпис)

Кондратьєв О. В.

Керівник роботи

(підпис)

доцент Вовк О.В.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 60 сторінок, 19 рисунків, 6 таблиць, 25 використаних літературних джерел.

1С:ПІДПРИЄМСТВО, ІНТЕРНЕТ-МАГАЗИН, СИНХРОНІЗАЦІЯ, ДАНІ, XML, WEB-СЕРВІСИ, FTP, SFTP, HTTP, HTTPS.

Метою магістерської атестаційної роботи є підвищення ефективності інтеграції між веб-ресурсом та системою 1С:Підприємство за допомогою розробки модуля синхронізації даних.

Об'єктом дослідження роботи є автоматизація обміну даними між сайтом та 1С.

Предметом дослідження роботи є програмні засоби для автоматизації обміну даними між сайтом та 1С.

Результатами атестаційної роботи є розроблені методики вирішення існуючих проблем при обміні даними між веб-ресурсом та системою 1С:Підприємство. Розроблені рекомендації щодо обміну даними. Створений свій модуль обміну даними між системами. Спираючись на розроблені методики можливо зменшити необхідний час на обмін даними між веб-ресурсом та системою 1С:Підприємство, скоротити роботу менеджерів.

РЕФЕРАТ

Пояснительная записка содержит 60 страниц, 19 рисунков, 6 таблиц, 25 использованных литературных источников.

1С:ПРЕДПРИЯТИЕ, ИНТЕРНЕТ-МАГАЗИН, СИНХРОНИЗАЦИЯ, ДАННЫЕ, XML, WEB-СЕРВИСЫ, FTP, SFTP, HTTP, HTTPS.

Целью магистерской аттестационной работы является повышение эффективности интеграции между веб-ресурсом и системой 1С:Предприятие с помощью разработки модуля синхронизации данных.

Объектом исследования данной работы является автоматизация обмена данными между веб-ресурсом и системой 1С:Предприятие.

Предмет исследования – программные средства автоматизации обмена данными между веб-ресурсом и системой 1С:Предприятие.

Результатами аттестационной работы являются разработанные методики решения существующих проблем при обмене данными между веб-ресурсом и системой 1С:Предприятие. Разработанные рекомендации по обмену данными. Созданный модуль обмена данными между системами. Опираясь на разработанные методики возможно уменьшить необходимое время на обмен данными между веб-ресурсом и системой 1С:Предприятие, сократить работу менеджеров.

ABSTRACT

Explanatory note 60 pages, 19 pictures, 6 tables, 25 sources.

1C: PIDPRIMSTVO, INTERNET STORE, SYNCHRONIZATION, DATA, XML, WEB-SERVICE, FTP, SFTP, HTTP, HTTPS.

The purpose of the master's attestation work is to improve the integration between Web resources and systems 1C: Enterprise by developing data synchronization module.

The object of study of this work is to automate data exchange between Web resources and systems 1C: Enterprise.

Subject of research - software for automation of data exchange between Web-resources and systems 1C: Enterprise.

Certifying the results of the work are developed methods of solving the existing problems in the communication between a web resource and system 1C: Enterprise. The recommendations on data exchange. Created data exchange between modules. Relying on the developed methodology is possible to reduce the time required for the exchange of data between the web and the system 1C: Enterprise, to reduce the work of managers.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ЛІТЕРАТУРИ ТА АНАЛОГІВ ЗА ТЕМОЮ РОБОТИ	11
1.1 Аналіз існуючих методів	11
1.2 Аналіз аналогів	18
1.2.1 ПЗ «shopModx»	18
1.2.2 ПЗ «mSkлад»	19
1.2.3 «NeoSeo Обмін з 1С» для OpenCart	20
1.3 Постановка задачі дослідження	22
2 ОПИС ОБМІНУ ДАНИМИ МІЖ 1С І ІНТЕРНЕТ-МАГАЗИНОМ	23
3 ПЛАНУВАННЯ ДОСЛІДЖЕННЯ	29
3.1 Опис розроблюваного загального алгоритму роботи web-додатки	30
3.2 Опис розроблюваного алгоритму авторизації у системі синхронізації даних з 1С	30
3.3 Опис розроблюваного алгоритму зміни конфігурацій системи	33
3.4 Опис розроблюваного алгоритму синхронізації	33
4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	37
4.1 Вибір мови програмування	37
4.2 Вибір інструментальних засобів для реалізації БД	38
4.3 Архітектурне проектування програмного забезпечення	39
5 ЕКОНОМІЧНА ЧАСТИНА	49
5.1 Характеристика науково-дослідного рішення	49
5.2 Етапи виконання НДР, їх трудомісткість та заробітна плата	49
5.3 Розрахунок одноразових витрат на розробку НДР	52
5.4 Оцінка результатів науково-дослідної роботи	55
ВИСНОВКИ	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	59

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних.

ЗШ – змінна шаблону.

СКЗ – система керування змістом.

ККК – керування корпоративним контентом.

ПЗ – програмне забезпечення.

П. П. – програмний продукт.

ФПС – фреймворк програмної системи.

FTP (File Transfer Protocol) – протокол передачі файлів.

SFTP (SSH File Transfer Protocol) – захищений протокол передачі файлів.

MD5 – контрольна сума.

ВСТУП

З появою інтернету переважна частина життя людини поступово перемістилися у віртуальний світ. У сучасній рутині для придбання потрібної речі не обов'язково виходити з дому, адже існує безліч інтернет-магазинів, що займаються різними категоріями товарів.

Інтернет-магазин - web-додаток, що займається поширенням товарів за допомогою мережі Інтернет. Дозволяє користувачу онлайн, в своєму браузері або через мобільний додаток, сформувані замовлення на покупку, вибрати спосіб оплати та доставки замовлення, сплатити замовлення. При цьому весь цикл продаж товарів здійснюється дистанційним способом.

Інтернет-магазин істотно полегшує завдання як для покупця, якому відтепер немає необхідності витратити свій час та кошти на проїзд точки продажу, так і продавцеві, якому більше немає потреби витратити гроші на оренду приміщення під магазин. Більш того продавець більше не обмежений у виборі регіону розміщення свого підприємства, адже для доставки замовлених товарів можна вдатися до послуг різних поштових сервісів.

Завдяки розміщенню свого каталогу товарів в мережі відпадає необхідність у створенні стендів, вітрин. Власник може поширювати свій каталог, не стикаючись з труднощами великого обсягу товарів в маленькому приміщенні. Але якщо існує необхідність не тільки в web-додатку, а й в реальному магазині, який би в точності або частково повторював асортимент свого інтернет-варіанта, або, маючи свою точку продажу, власник має потребу вивести свій каталог в мережу. У такому випадку існує необхідність в системі управління великими обсягами товарів, наприклад, 1С.

1С: Підприємство - програмний продукт компанії «1С», призначений для автоматизації бухгалтерського та управлінського обліків на підприємстві

Технологічна платформа «1С: Підприємство» являє собою програмну оболонку над базою даних. Використовуються бази на основі DBF-файлів в 7.7,

власний формат 1CD. Платформа має свій внутрішній мову програмування, що забезпечує, крім доступу до даних, можливість взаємодії з іншими програмами за допомогою OLE і DDE, в версіях 7.7, 8.0 і 8.1 - за допомогою COM-з'єднання [1, 2].

Дане ПО як не можна краще підходить для маніпулювання великими обсягами даних, формуванням каталогів товарів, знижок, бази клієнтів, обліку товарів на складі і т.д. На сьогоднішній день безліч компаній використовують 1С в своїх фірмах і хотіли б мати інтеграцію з цим продуктом і в своїх магазинах. Але це неможливо реалізувати силами тільки 1С, тому що дана система не є БД в широкому сенсі слова, в ній немає стандартних засобів для інтерпретування SQL команд, за допомогою яких і відбувається обмін інформацією між БД і стороннім сервісом.

1 АНАЛІЗ ЛІТЕРАТУРИ ТА АНАЛОГІВ ЗА ТЕМОЮ РОБОТИ

1.1 Аналіз існуючих методів

Найкращим варіантом системи обліку для більшості онлайн - магазинів є 1С - програма для автоматизації роботи підприємств, що дозволяє якісно вирішити проблему обліку товарів, замовлень і ефективно побудувати документообіг. Тому більшість інтернет-магазинів використовує для ведення обліку систему «1С: Підприємство» [3]. А значить їм необхідно вести 2 бази одночасно - одну у себе в програмі 1С, а другу в адміністративній панелі сайту. Виходить, що це створює подвійне навантаження для менеджерів. Їм спочатку необхідно побачити замовлення в адміністративній панелі сайту і потім вже перенести його в 1С в ручному режимі. Кожну зміну в базі 1С потрібно вручну переносити в базу сайту і це безумовно не сприяє ефективній життєдіяльності проекту.

Тому для автоматизації онлайн-бізнесу і оптимізації робочих процесів застосовується синхронізація даних 1С і сайту. Синхронізація інтернет-магазину з 1С дозволить завантажувати товари, замовлення, інформацію про клієнтів і інші дані, які необхідні в 1С. Пропоную розглянути ключові переваги цього рішення.

Головна перевага інтеграції магазину з 1С - це автоматичне завантаження товарів з 1С на сайт разом з усіма необхідними даними і параметрами. Синхронізований з 1С сайт завжди підтримує актуальність даних про товарні залишки, відповідно, вирішується питання своєчасності і повноти оновлення інформації каталогів інтернет-магазину. Замовлення і дані клієнтів також переносяться в 1С автоматично.

В результаті інтернет-магазин отримує такі бонуси:

– інтеграція магазину з 1С дозволяє значно заощадити час співробітників (або скоротити їх кількість);

– з її допомогою стає можливим ефективно мінімізувати кількість помилок ручного введення.

Додатковою перевагою стане збільшення швидкості обробки замовлень (якщо менеджер не витрачає час на перенесення даних клієнта з сайту в 1С, а відразу може побачити замовлення в даній системі обліку, він явно зможе витратити час на більш важливу роботу - наприклад, швидше відповісти на листи клієнтів).

Актуальність даних про товарні залишки на сайті завжди потішить його відвідувачів (і вони не будуть забирати час у менеджерів, замовляючи товари, «яких вже два роки як немає в наявності»).

Формалізованого модуля, який би дозволив завантажувати з будь-якого інтернет-магазину, в будь-яку програму 1С, на жаль, не існує. Це обумовлено тим, що програми 1С бувають різними: торгівля, управління торговим підприємством, управління виробничим підприємством, управління невеликою фірмою і т.д. Також, як і бувають різні системи, які використовуються на сайтах і для кожної необхідний свій підхід[4, 5].

Можна виділити основні способи обміну даними між сайтом і 1С [6].

Використання HTTP-сервісів в 1С.

Приклади застосування HTTP-сервісів такі ж, як і для веб-сервісів. Наприклад, двосторонній обмін даними 1С з інтернет-магазином. З боку сайту в 1С передаються замовлення, з боку 1С на сайт - документи оплати і документ, що підтверджує факт відвантаження товару.

HTTP-сервіси схожі на наявні в платформі веб-сервіси, але володіють декількома перевагами:

- більш просте створення клієнтського додатка;
- зменшений обсяг переданих даних;
- менша потреба в обчислювальних потужностях;
- велика націленість на роботу в мобільних пристроях.

Між веб-сервісами та HTTP-сервісами існує різниця в структурі запитів, якими обмінюються програми. Веб-сервіси (SOAP) - POST запити, HTTP-

сервіси - GET запити. Також, як і при роботі з веб-сервісами, для роботи з HTTP-сервісами необхідна установка веб-сервера Apache.

Підтримка REST-інтерфейсу в 1С.

Завдяки платформі і унікальності цього механізму, це найбільш зручне рішення питання інтеграції 1С із зовнішніми системами. REST-інтерфейс дозволяє створювати нові об'єкти, видаляти їх, читати і редагувати. Найбільш часто REST-інтерфейс застосовується в наступних випадках:

- інтеграції з веб-додатками (інтернет-магазини, веб-портали тощо);
- обміну даними із зовнішнім стороннім додатком;
- необхідність розширення можливості застосування 1С сторонніми засобами без доопрацювання самої конфігурації.

Застосування формату XML в 1С.

XML-формат є в деякому роді універсальним форматом і широко використовується у всіх конфігураціях 1С. 1С: Підприємство підтримує роботу з XML-документами за допомогою функцій вбудованого мови програмування. Завдяки цьому XML-формат широко використовується у власних розробках.

Формат широко використовується при обміні з віддаленими підрозділами та з інтернет-сайтами, при завантаженні виписок з банків і прайс-листів від постачальників, при вивантаженні даних в звітні органи і т.д. В 1С є універсальні обробки для вивантаження даних в форматі XML.

Перевага при обміні зі сторонніми додатками полягає в тому, що це дуже поширений формат і підтримується більшістю програмних продуктів, незалежно від структури бази даних.

Всі описані вище формати обміну даними в 1С призначені для того, щоб підготувати запитовані дані і передати їх сторонньому додатку. Або запросити необхідні дані у зовнішнього додатку, отримати їх і передати для обробки на підставі внутрішніх алгоритмів 1С. Зазвичай для інтеграції з веб-системою на стороні 1С використовують стандартний обмін з web-сайтом в форматі CommerceML. Дане рішення є стабільним і перевіреним на багатьох проектах.

Було проведено аналіз систем інтеграції для потреб нашого власного продукту, тому пропоную розглянути і порівняти формати обміну CommerceML і інших способів інтеграції.

Обмін з сайтом по формату CommerceML.

У типових конфігураціях 1С:Підприємство існує два типи обміну, заснованого на форматі CommerceML:

– обмін за схемою Постачальник-Покупець. Обмін за схемою Постачальник-Покупець можна використовувати для обміну з сайтом, якщо база 1С виступатиме в якості Постачальника, а сайт в якості Покупця. Але обмін по цій схемі можливо використовувати лише в ручному режимі, завантаження і вивантаження проводиться вручну. Також вручну необхідно обробляти завантажені замовлення і формувати на них відповіді;

– web-розширення. В асортименті програмних продуктів 1С є компонент web-розширення для платформи 1С:Підприємство. Дане рішення засноване на технології Web Forms, яка інтегрує веб-форму, елемент управління і джерело даних. Для доступу до даними елементи управління використовують технологію ADO.NET, а призначений для користувача інтерфейс працює на ASP.NET.

Основний недолік цієї технології - обмежений дизайн компонентів веб-форм, сайт повинен використовувати ASP.NET, необхідність додаткового ліцензування та фактично прямий доступ в базу даних.

Використовувати COM інтерфейс. Використання COM інтерфейсу передбачає наявність у сайту com-інтерфейсу, до якого можна підключитися.

Зв'язок по COM-інтерфейсу підтримується багатьма додатками в середовищі Windows, проте, для обміну з web-сайтом це досить екзотичне рішення. Для реалізації такого рішення також потрібно наявність програміста, який володіє технологіями COM.

Використовувати web-сервіси 1С [7]. Для web-сервісів необхідно відкривати порт зі сторони 1С, що становить потенційну загрозу проникнення в базу з Інтернету.

Зручніше для обміну 1С з сайтом використовувати вбудовану в платформу 1С:Підприємство технологію web-сервіси. Але використання цього рішення відштовхує компанії через необхідність відкривати доступ до 1С з Інтернету.

Перейти на 1С 8.3. У версії 8.3 1С: Підприємство реалізована підтримка SSL, сертифікатів у web-сервісах і об'єктах вбудованої мови використовують FTP і HTTP-з'єднання. В даному випадку в web-сервісах вже забезпечується необхідний рівень безпеки доступу до даних. Для даної платформи поки що не реалізовані типові конфігурації, що обмежує її поширення.

Універсальний обмін XML. Універсальний механізм обміну [8]. XML гнучко налаштовується без втручання програміста з допомогою конфігурації «Конвертація даних». Але не дозволяє здійснювати обмін в автоматичному режимі. А також, в даному варіанті обміну не відслідковуються зміни об'єктів. Тому доводиться вивантажувати всі об'єкти, навіть якщо вони не змінювалися. У кращому випадку для документів можна встановити інтервал вивантаження.

Самописний обмін. 1С вивантажує файли формату txt , xml або csv , які передаються на сайт за протоколами http або ftp. Сайт обробляє отримані файли. Самописний обмін дозволяє досить гнучко описати всі правила і алгоритми обміну, проте він добре працює при обміні невеликими обсягами даних, при великих обсягах починаються проблеми з продуктивністю.

Веб-сервер на стороні 1С. Вбудована в платформу 1С:Підприємство технологія web-сервісів дозволяє створити конфігурацію з повноцінною CMS-системою генеруючої по запиті html-код. Таким чином кардинально вирішується питання обміну з сайтом, його по суті справи немає, так як сайт працює на базі 1С. Дане рішення потенційно володіє низькою продуктивністю.

У таблиці 1.1 приведено порівняння наведених рішень за такими параметрами: продуктивність, відмовостійкість, безпеку, ергономіка.

Таблиця 1.1 – Порівняння варіантів обмінів

Тип обміну	Переваги	Недоліки
CommerceML	Є в стандартному постачанні 1С. Прийнятний рівень безпеки, немає доступу в базу 1С з інтернет.	При роботі по схемі Постачальник-Покупець необхідно вручну ініціювати обмін. Надмірність даних протоколу знижує продуктивність. Дані не зашифровані. Великі обсяги даних можуть викликати відмову в роботі. Немає стандартних засобів автоматичного моніторингу процесу обміну.
Web -розширення	Прямий доступ в базу 1с дозволяє спростити процес налагодження. Включено в деякі стандартні поставки.	Під кожного користувача треба купувати ліцензію 1С. Продуктивність сильно залежить від швидкості доступу до бази 1С. Прямий доступ в базу, потенційна загроза доступу до даними в 1 С з інтернет а . Падіння сервера 1С викличе падіння сайту.
підключається DLL	Продуктивність залежить тільки від обсягу даних Прийнятний рівень безпеки, доступу в базу 1С з інтернет а немає. Трафік шифрується при обміні по протоколу sftp . За порівнянні з іншими варіантами найбільш висока відмовостійкість при передачі великих обсягів даних.	Трудомістке написання бібліотеки. Підійде тільки для реалізації транспорту даних.
COM-інтерфейс	Прийнятний рівень безпеки, доступу в базу 1С з інтернет немає.	Продуктивність сильно залежить від швидкості доступу до бази 1С. Можливі збої при частих таймаут. Трудомістке написання COM інтерфейсу.
Web -сервіси 1С	Прямий доступ в базу 1с дозволяє спростити процес налагодження.	Продуктивність сильно залежить від швидкості доступу до бази 1С Відкрито порти доступу в базу, є потенційні загрози Можливі збої при збоях в базі 1С

Продовження таблиці 1.1

Тип обміну	Переваги	Недоліки
1С 8.3	Високий ступінь безпеки, підтримка протоколів шифрування Є кошти для підвищення рівня відмовостійкості.	Необхідна міграція з поточної платформи 1С.
Універсальний обмін XML	Прийнятний рівень безпеки, доступу до бази 1С з інтернет немає.	Ініціація обміну за замовчуванням здійснюється оператором. Обробка великих файлів правил сильно знижує продуктивність. Дані передаються у відкритому вигляді. При великих обсягах даних можливі збої.
Веб-сервер на стороні 1С	Ні промислових впроваджень. Рішення на рівні прототипу.	Продуктивність нижче, ніж у звичайних веб-серверів Низький рівень безпеки, відкритий порт доступу до бази даних 1С. Сайт не працює при збоях з базою 1С.

Кожен з описаних вище варіантів має свої переваги і недоліки. Який з них вибрати в кінцевому підсумку залежить від багатьох факторів. У кожному разі вирішується індивідуально.

З досвіду найбільш оптимальним є рішення, що використовує відразу кілька варіантів обміну для різних ситуацій.

До коли постав вибір який тип обміну використовувати в масштабному проекті зі створення готової b2b системи з універсальною інтеграцією в більшість конфігурацій 1С, на основі глибокого аналізу був обраний формат CommerceML з допрацьованим функціоналом. Саме він поєднує в собі гнучкість настройки універсального обміну XML, високу автоматизацію і підвищену продуктивність. В результаті у зазначеній вище системі інтернет-дистрибуції був використаний оптимізований CommerceML формат для обміну сайту з базами 1С: Підприємство. При цьому була можливість гнучкого налаштування об'єктів обміну без програмування, шляхом додавання об'єктів в пакет XDTO. Великі обсяги даних система передає по протоколу sftp, що помітно підвищує відмово стійкість і гарантує безпеку.

1.2 Аналіз аналогів

1С досить велика система, призначена для вирішення величезної кількості завдань, включаючи фінансові розрахунки, менеджмент персоналу, а також формування виробничих списків товарів і послуг підприємства.

Дана система досить давно використовується на підприємствах для обліку продукції, що випускається, і зарекомендувала себе як надійний інструмент в даній області. За той час поки підприємство функціонує, в 1С накопичується колосальний обсяг інформації. І коли стає завдання про побудову, web-додатку який б мав доступ до даної інформації, наприклад, інтернет-магазину, виникає проблема в способах отримання даних з 1С, зміна їх і збереженні назад. Розглянемо декілька існуючих модулів обміну між веб-ресурсом та системою 1С:Підприємство.

1.2.1 ПЗ «shopModx»

Даний модуль є потужним інструментом, призначеним для швидкого запуску інтернет-магазинів (рис. 1.1) [9, 10].

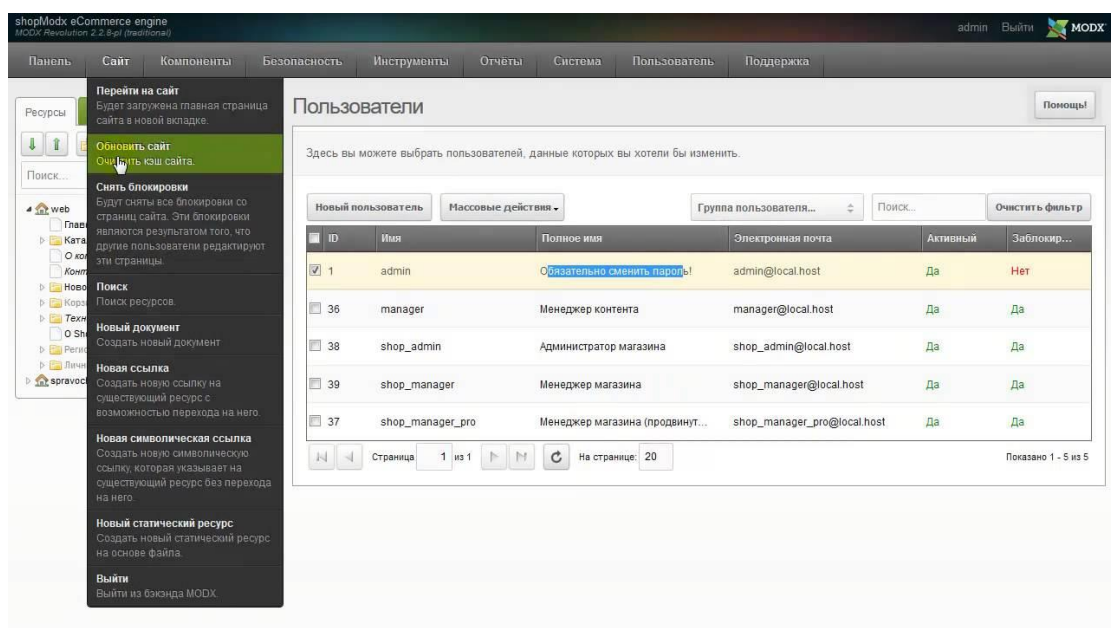


Рисунок 1.1 – Приклад роботи з користувачами через модуль «shopModx»

Він володіє інструментом для синхронізації з 1С але тільки по протоколу SOAP. Повністю порушує структуру MODX, вносячи зміни в ядро і структуру таблиць БД. При установці видаляє всі дані, після видалення подальша робота з CMS неможлива.

Переваги:

- швидкий обмін даними з БД;
- надання функціоналу базової роботи інтернет магазину.

Недоліки:

- для обміну даними використовується застарілий протокол.
- повністю змінює структур CMS.
- робота з великим файлами.

1.2.2 ПЗ «mSkлад»

Утиліта, призначена виключно для вивантаження даних в один батьківський ресурс. Має дуже гнучку систему налаштувань по інтеграції з 1С (рис. 1.2), але яка базується виключно на файлах XML [11, 12].

Id	источник	тип поля	цель	активный
1	Артикул	поле msProduct	article	да
2	Изготовитель	поле msProduct	vendor	да
3	Цена	поле msProduct	price	да
4	Количество	поле msProduct		нет
5	Штрихкод	поле msProduct		нет
6	Вес	поле msProduct	weight	да
7	Бренд	поле msProduct	vendor	нет
8	Цвет	поле msProduct	color	да
9	Полное наименование	поле msProduct	longtitle	да

<<< << Страница 1 из 1 >> >>> На странице: 20 Показано 1 - 9 из 9

Рисунок 1.2 – Приклад конфігурацій «mSkлад»

Перевага: повна інтеграція з 1С.

Недоліки:

- для синхронізації використовується виключно файли XML;

- модуль не здатний відтворити структуру каталогів;
- робота з великим файлами.

1.2.3 «NeoSeo Обмін з 1С» для Opencart

Даний модуль використовується для оновлення товарів в CMS «Opencart». Він дозволяє виконувати такі функції:

- автономна робота по обміну замовленнями з різних торгових точок;
- вивантаження з Інтернет-магазину на OpenCart в 1С контрагента який бере участь в замовленні;
- можливість створити і оновлювати дерево категорій номенклатури на сайті, без повторного вивантаження;
- редагувати товарні позиції прямо в 1С з подальшим оновленням інформації на сайті;
- оновлення номенклатури на сайті, без повторного вивантаження;
- вивантаження характеристик в опції;
- вивантаження залишків по товарах і окремо по складах;
- вивантаження довжини, ширини, висоти, ваги кожного товару.

Але також модуль дуже складний в налаштуванні, він має багато вкладок по різним опціям (рис. 1.3).

Він дозволяє вивантажувати усі замовлення за певними параметрами, також отримувати відповіді від 1С про зміну статусу замовлення в автоматичному режимі. Має дуже багато функцій для роботи з характеристиками та атрибутами, є можливість вказувати категорії, якщо це потрібно змінити, оновлювати параметри фільтрів, а також інвалідувати «кеш».

При проведенні аналізу було виявлено, що модуль дуже складний в налаштуванні. Користувач, що ніколи не працював з подібними системами нічого в ньому не зрозуміє зовсім. Також потрібна тісна співпраця з 1С програмістом.

The screenshot shows the 'Товары' (Goods) configuration page in the NeoSeo 1C 164 application. The page includes several settings:

- Синхронизация товаров:** Рекомендуются синхронизировать только по артикулу (Set to 'По артикулу').
- Добавлять новые товары:** Выключите, если вам нужна синхронизация остатков и/или цен только по имеющимся в магазине товарам (Set to 'Включено').
- Начальный статус новых товаров:** Если нужна дополнительная обработка товаров, пришедших из 1С, то выключите (Set to 'Включено').
- Начальный статус наличия новых товаров:** Укажите статус наличия, который нужно выставлять созданным продуктам (Set to 'В наличии').
- Спец. Тег для статуса если товара нет на складе:** Укажите имя тега отвечающего за значение статуса товара которого нет в наличии. Оставьте поле пустым если Вам не нужно учитывать статус отсутствия товара (Empty field).
- Статус по умолчанию для товаров которых нет в наличии:** Статус устанавливается если присутствует спец. тег для статусов (Set to 'Ожидание 2-3 дня').
- Обновлять наименование:** Выключите, если наименование вам нужно только при создании товара, а дальше вы будете редактировать его в самом магазине (Set to 'Включено').
- Обновлять описание:** Выключите, если описание вам нужно только при создании товара, а дальше вы будете редактировать его в самом магазине (Set to 'Включено').

Рисунок 1.3 – Пример параметрів, які можна налаштувати

Підводячи підсумок можна сказати, що кожен модуль має свої недоліки й переваги, але потрібно виділити їх спільні проблеми:

- складність в налаштуванні, кінцевому користувачу буде дуже складно в усьому цьому розібратися;
- часто – руйнування структури сайту;
- складність роботи з «великими» файлами, так як може не вистачити оперативної пам'яті у сервера для коректної роботи;
- якщо сталася помилка після створення товару (наприклад проблема з зображенням, або параметрами, які встановлюються після створення товару), то неможливо відмінити його створення – цей товар буде не правильно відображатися на ресурсі;
- складність роботи з зображеннями, так як 1С може їх вивантажувати тільки повністю по всім товарам;
- складність роботи з великим файлами.

Виходячи з перерахованих недоліків необхідно розробити програмний модуль обміну між веб-ресурсом та системою 1С:Підприємство, який вирішить деякі з перерахованих проблем обміну.

1.3 Постановка задачі дослідження

Метою магістерської атестаційної роботи є підвищення ефективності інтеграції між веб-ресурсом та системою 1С:Підприємство за допомогою розробки модуля синхронізації даних.

Таким чином, маємо змогу сформулювати наступну гіпотезу: «Модуль обміну даними між веб-ресурсом та системою 1С:Підприємство, створений в ході науково дослідницької роботи, сприяє підвищенню ефективності роботи інтернет-магазину та вирішую проблеми обміну даними».

Для практичної реалізації сформульованої мети були поставлені та вирішені наступні задачі:

- обґрунтування необхідності проведення дослідження обміну даними між сайтом та 1С;
- проведення аналізу існуючих методів обміну даними між сайтом та 1С;
- опис обраних методів синхронізації;
- детальний аналіз методів обміну даними;
- визначення найкращих методів обміну даними;
- розробка ефективного модулю обміну даними;
- оцінка ефективності розробленого модулю.

Об'єктом дослідження роботи є автоматизація обміну даними між сайтом та 1С.

Предметом дослідження роботи є програмні засоби автоматизації обміну даними між сайтом та 1С.

2 ОПИС ОБМІНУ ДАНИМИ МІЖ 1С І ІНТЕРНЕТ-МАГАЗИНОМ

За своїм родом діяльності, я часто займаюся впровадженням нових програмних продуктів для автоматизації бізнесу. При цьому важливим етапом роботи завжди є інтеграція систем між собою. І особливо тут варто виділити одну з найбільш концептуальних потреб - інтеграцію облікової системи з інтернет-магазином.

При цьому CMS і особливості реалізації інтернет-магазинів можуть бути вкрай різними, в якості облікової системи найчастіше виступає 1С, але й тут є аналоги, які можуть дуже сильно відрізнятися один від одного. Є основні принципи і певні стандарти обміну даними, загальні для всіх типів програмного забезпечення [13].

Автоматична інтеграція між інтернет-магазином і обліковою системою допомагає вирішити цілий ряд питань:

- позбутися від проблем, пов'язаних з людським фактором;
- знизити число можливих помилок при обміні даних;
- знизити завантаженість співробітників і направити їх активність безпосередньо на продажі і роботу з клієнтами;
- забезпечити актуальність інформації як для відвідувачів магазину, так і для продавців компанії. Особливо актуально при великому товарообігу.

Розглянемо основні етапи обміну даними. Процес обміну даними між програмними системами складається з 3 основних частин:

- ініціація обміну;
- вивантаження даних на сайт;
- вивантаження даних з сайту.

Якщо перший етап відбувається один раз, тобто обмін налаштований, включений і працює, то етапи 2 і 3 повторюються регулярно.

Ініціація - це важливий етап, основна частина роботи в рамках якого, проводиться до першого вивантаження даних на сайт. Необхідно налаштувати

бази даних під певні стандарти, перевірити інформацію по суті (щоб не вивантажувати помилкові або надмірні відомості) і т.д.

Далі на наступних етапах обмін буде здійснюватися автоматично і регулярно. Але перше вивантаження даних повинно бути відповідним чином підготовлене, і тільки потім обмін ініціюється - проводиться перше вивантаження і після перевірок включається постійний обмін.

Розглянемо основні параметри, що вивантажуються на. З облікової системи на сайт вивантажуються такі відомості:

- назва товарної позиції;
- ієрархія товару;
- артикул і код товару;
- опис;
- характеристики (колір, розмір і т.д)
- штрих-код товару;
- ціни, акції, знижки;
- товарні залишки.

Деякі параметри з переліку зрозумілі інтуїтивно. Інші розглянемо докладніше.

Ієрархія - ієрархію створюють як окремий довідник в 1С. Вбудовану ієрархію 1С використовувати не рекомендується. Справа в тому, що структура каталогу товарів на сайті і довідника в 1С дуже часто виявляється різною. При цьому саме таким чином зручніше будувати роботу: для покупців - один варіант структури, для роботи в 1С - інший. Відповідно, для вивантаження в магазин необхідний окремий параметр «Група товарів на сайті», де буде відображатися ієрархія каталогу в інтернет-магазині.

Ціни, знижки та акції - ціни для інтернет-магазину рекомендується також зберігати як окремий тип ціни (довідник "Типи цін"), так як вони нерідко відрізняються від вартості товару в оптових або роздрібних продажах. При цьому окремо потрібно враховувати можливість продажу зі знижками і в рамках акцій.

Зазвичай «Акція» в картці товару в 1С включаються за допомогою прапорця («галочка»). Необхідно цей параметр також переносити в інтернет-магазин, де акційні товари будуть виділятися відповідним чином.

Для ціни зі знижкою необхідно реалізувати окремий параметр. При цьому в магазин вивантажуються 2 ціни: без знижки і зі знижкою. Це потрібно для реалізації поширеного рішення: перекреслена ціна без знижки і нова - зі знижкою [14].

Товарні залишки - основна проблема вивантаження залишків в магазин-це неможливість 100% синхронізувати дані. Ви можете вивантажувати залишки з 1С на сайт дуже часто, але все одно, в межах профілю системи актуалізація залишків буде відбуватися швидше. Ця проблема буде вирішена у роботі. Також, необхідно передбачити додаткову перевірку оформленого замовлення з сайту на предмет наявності всіх позицій.

Також слід продумати варіанти вирішення на випадок відсутності потрібного товару при замовленні через сайт.

Описи товарів і фото - описи вивантажувати з 1С в більшості випадків не має сенсу. На сайті вони повинні красиво відображатися в форматі HTML, в описах нерідко бувають посилання, таблиці, різні варіанти форматування тексту. Довідники 1С не підтримують всі ці нюанси. А тому набагато зручніше завантажувати опису відразу на сайт.

Фотографії також нерідко прагнуть зберігати в 1С і завантажувати на сайт, що є великою помилкою.

1. Графічні файли порівняно багато «важать», в результаті обмін даними буде значно сповільнюватися, при цьому буде потрібно залучити значні серверні потужності.

2. Відображення фото в CMS може бути некоректним, файли «обрізаються» або «стискаються» системою під задані параметри, а в деяких CMS взагалі не відображаються після вивантаження без додаткових дій контент-менеджера.

3. Фото, як і опису товару, оновлюються вкрай рідко.

За перерахованими причин фото товарів рекомендується вивантажувати на сайт через FTP – не має захисту через шифрування, тому рекомендується використовувати SFTP, а у самому файлі вивантаження вказувати шлях до файлу на сервері де встановлений сам сайт. Далі система вже сама створить мініатюри і стисне файл. Опис рекомендується завантажувати відразу в CMS, як і інші дані для сайту, відповідальність за які несе контент-менеджер.

При цьому важливо не зловживати розмірами фотографій. Зрозуміло, що всім хочеться показати товар якомога якісніше. Але занадто великі фото - це файли величезних розмірів. В результаті значно сповільнюється завантаження сторінки сайту для користувачів, а користі - ніякої, так як ніхто (за винятком, можливо, конкурентів) не вивантажують фото і переглядають його з великим збільшенням. А тому фотографії необхідно оптимізувати до вивантаження на сайт (навіть якщо ваша CMS підтримує автоматичну оптимізацію, великі файли будуть займати зайве місце на хостингу). І при оформленні контент-менеджер обов'язково повинен відслідковувати такі параметри, як назва файлу (найкраще осмислене та використовувати кириличні символи) і заповнювати тег alt (відображається якщо не видно картинку, корисний для пошукової оптимізації).

Вивантаження з магазину в 1С - з інтернет-магазину в облікову систему обов'язково вивантажується один тип документів - замовлення від клієнта разом з усіма необхідними даними (ПІБ, email, телефон, адреса, фінансова інформація для юридичних осіб і т.д.).

Тут варто зупинитися на двох важливих етапах:

- вивантаження замовлення;
- проведення замовлення.

У процесі вивантаження необхідно коректно з бази даних магазину в 1С передати всі дані:

- ПІБ і контакти (при цьому важливо уникати дублювання в разі повторних замовлень);
- адреса доставки або точку самовивозу;

- дані про оплату;
- дані про товари та їх кількості.

Оплата в інтернет-магазині може бути простою (оплачується повна сума безготівково або при отриманні товару) і складною (частина суми оплачується подарунковим купоном, балами з програми лояльності і т.д.). Всі ці відомості також повинні коректно передаватися в 1С.

Дуже важливо, щоб всі відомості з замовлення передавалися в 1С без спотворень. Наприклад, товар був проданий з певною знижкою, в замовленні в 1С повинна відображатися ціна зі знижкою, навіть якщо вона виявиться індивідуальною (з урахуванням бонусних балів і т.д.). Нерідко виникають ситуації, коли при автоматичному обміні даних в 1С використовується ціна з довідника цін для інтернет-магазину, в результаті в обліковій системі ціни і сума замовлення можуть не збігатися з тим, що було на сайті.

При проведенні замовлення може виникнути описана вище ситуація: в замовленні присутній вже проданий товар. Необхідно передбачити таку можливість і реалізувати шляхи вирішення. Зазвичай при цьому замовлення проводиться з залишками «в мінус» або не проводиться взагалі. І в будь-якому випадку про проблему повідомляється відповідальний співробітник, який в ручному режимі вирішує питання з покупцем (домовляються про повернення коштів або збільшення терміну поставки, якщо товар можна швидко замовити). Після оформлення замовлення і проведення його в 1С процес продажу переходить до питань перевірки оплати та здійснення доставки (адресну або до точки самовивозу).

І тут важливо передбачити, яким чином покупець буде отримувати повідомлення про статус замовлення.

Варіант 1. Повідомлення будуть розсилатися з 1С (на email, за допомогою sms або телефонії)

Варіант 2. Статус замовлення відображається в інтернет-магазині і повідомлення відправляються з сайту.

У першому випадку обмін даних не потрібний, вся робота проводиться в рамках звичайного бізнес-процесу продажу. У другому випадку необхідно реалізувати вивантаження з 1С в інтернет-магазин такого параметра, як статус замовлення.

Що необхідно передбачити при організації обміну даними - до початку інтеграції облікової системи та сайту необхідно чітко визначити бізнес-процес продажу. Продумати всі етапи роботи з покупцями, це допоможе визначити, які відомості вам потрібно передавати в який бік.

Що важливо враховувати при складанні техзавдання:

- які дані необхідно вивантажувати в кожну сторону;
- з якою періодичністю буде проводитися обмін;
- передбачити можливі колізії: проблеми із залишками, можливість подвоєння даних, інші складності. Все це повинно бути передбачено, співробітники компанії повинні знати, як діяти в разі виникнення колізій.

Отже, можна зробити висновок, що технічні нюанси реалізації інтеграції залежать від обраної конфігурації 1С і особливостей інтернет-магазину. Але незалежно ні від чого, якщо при обговоренні модулю і складанні технічного завдання будуть враховані описані в розділі етапи і враховані «підводні камені», інтеграція пройде швидко, а наступний обмін даними буде працювати якісно і ефективно.

3 ПЛАНУВАННЯ ДОСЛІДЖЕННЯ

Алгоритм це чітка послідовність дій або інструкцій яка описує як досягти бажаного результату. Щоб алгоритм міг послідовно виконуватися не тільки людиною, але й деяким технічним засобом, алгоритм має бути формалізованим. Усі команди, які може виконати пристрій називаються системою команд виконавця (СКІ) [15, 16].

Алгоритм має бути зрозумілим виконавцю, який його виконує, тобто містити приписи про виконання тільки таких дій і про перевірку тільки таких властивостей об'єкта, які входять в СКІ. Також алгоритм повинен бути дискретним, тобто виконуватися команди алгоритму повинні послідовно, результативним, тобто алгоритм повинен забезпечувати можливість отримання результату за кінцеве число кроків, певним, тобто повинні бути точні відомості про те, що після виконання кожної чергової команди буде завершено виконання алгоритму, або про те, яка наступна команда повинна буде виконатися після поточної.

Алгоритми можуть бути задані в декілька засобів:

– за допомогою словесного опису– описує алгоритм за допомогою неформалізованих речень природної мови, з використанням професійних понять, термінів, залежностей і знаків [17];

– за допомогою графічного опису – спосіб представлення алгоритму за допомогою геометричних фігур, які називаються блоками [18];

– за допомогою псевдокоду – мова опису алгоритмів, що використовує ключові слова імперативних мов програмування.

Єдиною системою програмної документації стандартизовано два методи опису алгоритмів програм: за допомогою блок-схем і за допомогою схем. Загальним гідністю цього методу є його незалежність від мови реалізації, що дозволяє домогтися переносимості алгоритмічного забезпечення. У цьому методі також реалізовані всі базові структури управління.

Блок-схеми є найдоступнішим і наочним способом алгоритмізації розроблюваної системи синхронізації даних з ІС [19].

3.1 Опис розроблюваного загального алгоритму роботи web-додатки

Загальний алгоритм роботи web-додатки складається з наступних дій:

- за допомогою браузера, користувач робить запит (GET-запит) на сервер. Для того щоб відправити запит, браузер використовує HTTP;
- сервер обробляє GET-запит: якщо запитувана сторінка існує - сервер повертає HTML розмітку, якщо запитувана сторінка не існує - сервер повертає номер помилки;
- браузер обробляє відповідь і відображає сторінку або номер помилки.

Блок-схема загального алгоритму роботи системи синхронізації даних з ІС зображена на рисунку 3.1.

3.2 Опис розроблюваного алгоритму авторизації у системі синхронізації даних з ІС

Алгоритм авторизації працює наступним чином.

1. Користувач відправляє GET – запит до сервера, у відповідь отримує форму авторизації.
2. Користувач заповнює форму і натискає на кнопку для відправки даних на сервер. Виконується POST-запит.
3. Сервер обробляє POST-запит: якщо введені дані є валідними - сервер виконує авторизацію і повертає відповідь користувачеві, в іншому випадку сервер повертає код помилки.
4. Браузер відображає дані, доступні тільки авторизованному користувачеві, інакше відображає повідомлення про помилки.

Блок-схема алгоритму авторизації представлена на рисунку 3.2.

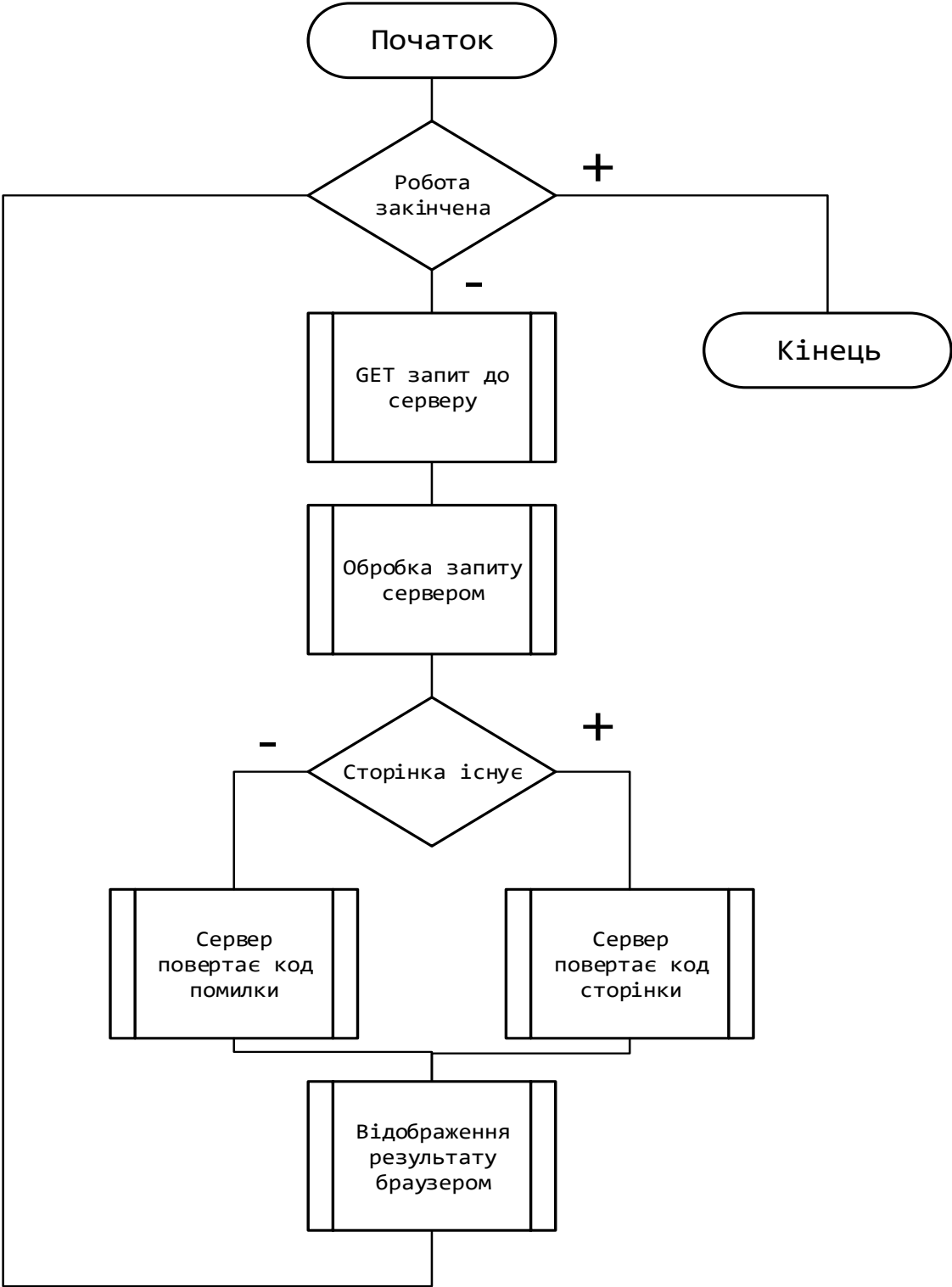


Рисунок 3.1 – Загальний алгоритм роботи web-додатку

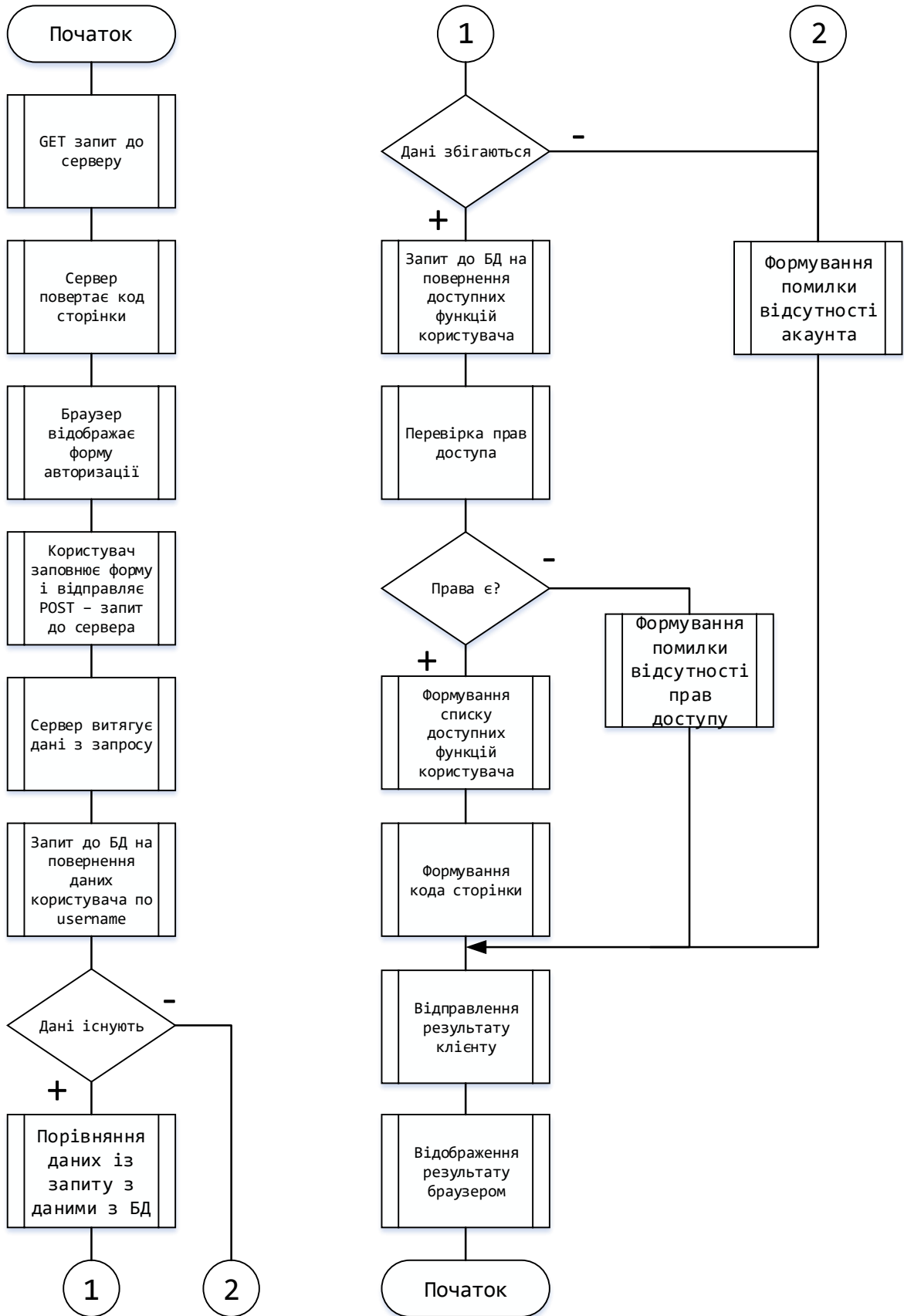


Рисунок 3.2 – Алгоритм авторизації системи синхронізації даних з 1С

3.3 Опис розроблюваного алгоритму зміни конфігурацій системи

Алгоритм зміни конфігурацій працює наступним чином.

1 Користувач авторизується у системі, а також проходить автентифікацію.

2 Користувач відправляє GET – запит до сервера, у відповідь отримує список доступних конфігурацій.

3 Користувач змінює конфігурацію, та натискає кнопку зберегти.

4 На сервер відправляється POST – запит.

5 Сервер змінює конфігурацію.

Блок-схема алгоритму зміни конфігурацій представлена на рисунку 3.3.

3.4 Опис розроблюваного алгоритму синхронізації

Змінювати існуючий алгоритм синхронізації ми не будемо, тому що він відповідає усім вимогам і добре себе зарекомендував у існуючих модулях.

Алгоритм синхронізації файлів з 1С працює таким чином.

А. Початок сеансу.

Вивантаження даних починається з того, що система "1С: Підприємство" відправляє http-запит наступного виду:

`http://<сайт>/<шлях>/1c_exchange.php?type=catalog&mode=checkauth.`

У відповідь система управління сайтом передає системі «1С: Підприємство» три рядки (використовується роздільник рядків "\ n"):

- слово "success";
- ім'я Cookie;
- значення Cookie.

Усі наступні запити до системи управління сайтом з боку "1С: Підприємства" містять в заголовку запиту ім'я і значення Cookie.

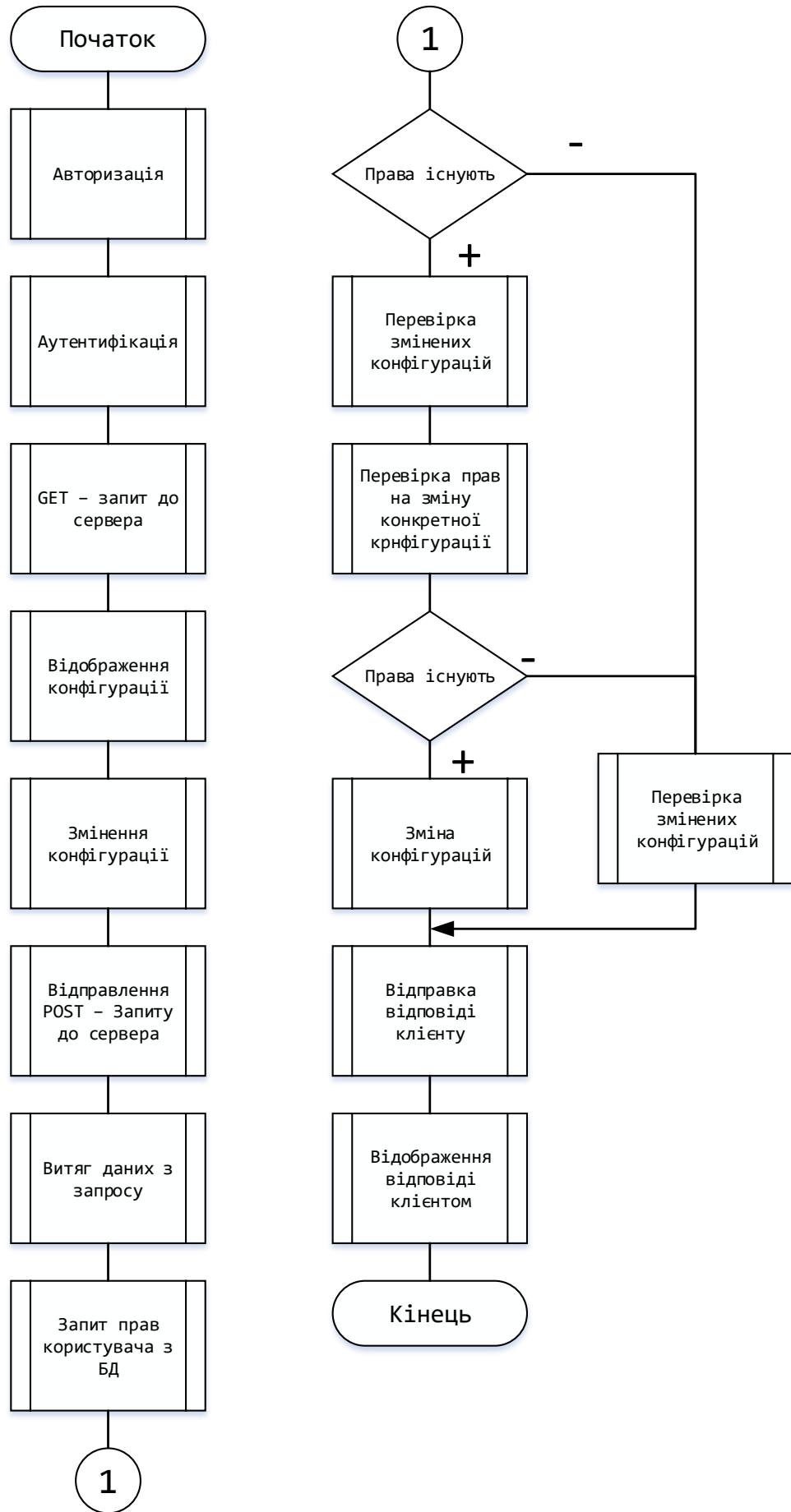


Рисунок 3.3 – Алгоритм зміни конфігурацій

В. Запит параметрів від сайту.

Далі слід запит наступного виду:

`http:// <сайт> / <шлях> / 1c_exchange.php?type=catalog&mode=init`

У відповідь система управління сайтом передає два рядки:

– `zip = yes`, якщо сервер підтримує обмін в `zip`-форматі - в цьому випадку на наступному кроці файли повинні бути упаковані в `zip`-форматі або `zip = no` - в цьому випадку на наступному кроці файли не упаковуються і передаються кожен окремо;

– `file_limit = <число>`, де `<число>` - максимально допустимий розмір файлу в байтах для передачі за один запит. Якщо системі "1С: Підприємство" знадобиться передати файл більшого розміру, його слід розділити на фрагменти.

С. Вивантаження на сайт файлів обміну.

Потім "1С: Підприємство" запитами з параметрами виду:

`http:// <сайт> / <шлях> / 1c_exchange.php ?type = catalog & mode =file &filename =<імя файлу>` вивантажує на сайт файли обміну в форматі CommerceML 2, посилаючи вміст файлу або його частини у вигляді POST.

У разі успішного запису файлу система управління сайтом видає рядок "success".

Д. Покрокове завантаження даних.

На останньому кроці за запитом з "1С: Підприємства" здійснюється покрокове завантаження даних за запитом з параметрами виду `http:// <сайт> / <шлях> / 1c_exchange.php?type=catalog&mode=import&filename=<імя файлу>`

Під час завантаження система управління сайтом може відповідати в одному з наступних варіантів.

1. Якщо в першому рядку міститься слово "progress" - це означає необхідність послати той же запит ще раз. В цьому випадку у другому рядку буде повернутий поточний статус обробки, обсяг завантажених даних, статус імпорту і т.д.

2. Якщо у відповідь передається рядок зі словом "success", то це буде означати повідомлення про успішне закінчення обробки файлу. Саме на цьому кроці відбувається найбільша кількість проблем. Такі як:

- великий файл імпорту;
- велика кількість товарів;
- надмірність переданих даних;
- велика кількість параметрів, що треба створити.

Саме ці проблеми будуть вирішені далі.

4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

4.1 Вибір мови програмування

Вибір буде здійснюватися серед трьох мов програмування: С# [20], PHP [21], Java [22, 23].

Java - це мова програмування розроблена і підтримувана корпорацією Oracle. Надає найбільш високу продуктивність і захист даних з усіх. Своїх власних фреймворків для web-розробки не має але досить багато написано сторонніми розробниками. Серед головних переваг варто відзначити кросплатформеність, що дозволяє мові однаково працювати під будь-який ОС.

С# - мова програмування розроблена та підтримувана компанією Microsoft. Дає весь необхідний спектр інструментів необхідні при проектуванні і розробці додатків (будь то веб-додаток або десктопних систем) під ОС Windows. Головний недолік полягає в повному провалі що стосується web-розробки, так як написані системи на даній мові працюють лише на серверах під управління ОС Windows, які абсолютно не підлягають застосуванню у корпоративній сфері.

PHP - найбільш поширена мова програмування для створення web-додатків. Підтримується більшістю хостинг-провайдерів. Має найбільшу швидкість розробки серед всіх. Розроблений групою ентузіастів і поширюється під власною ліцензією не сумісною з GNU GPL. Також є кросплатформенною.

Для порівняння факторів між перерахованими вище мовами програмування застосований метод варіантних мереж, результати якого представлені в таблиці 4.1.

За результатами порівняння трьох мов була обрана мова PHP.

Таблиця 4.1 – Порівняння мов програмування

Критерії Варіанти	Переносимість	Час розробки	Організація управлінням БД	Швидкість	Разом
	Вага = 4	Вага = 5	Вага = 5	Вага = 4	
C#	2	3	5	5	68
PHP	5	5	5	4	86
Java	5	3	5	5	80

4.2 Вибір інструментальних засобів для реалізації БД

Серед програмних засобів розробки, які мають характеристики необхідні для реалізації БД розроблюваного програмного продукту, доступними є наступні:

- Oracle [24];
- MYSQL [25].

MySQL - це реляційна база даних, робота з даними в якій здійснюється за допомогою SQL запитів. Основними перевагами цього типу БД є швидкість і простота у використанні. Ця БД повністю реалізована на мові PHP, тому її можливо встановитися на любий сервер.

Oracle Database або Oracle RDBMS - об'єктно-реляційна система управління базами даних (СКБД).

Для порівняння цих програмних продуктів скористаємося методом варіантних мереж. Оцінимо за п'ятибальною шкалою наступні характеристики програмних продуктів.

- 1 Продуктивність СУБД.
- 2 Рівень безпеки даних.
- 3 Швидкість розробки програмного продукту.
- 4 Рівень реалізації механізму збережених процедур.
- 5 Наявність розширень стандартної мови SQL.
- 6 Надані можливості адміністрування бази даних.
- 7 Вимоги до апаратних ресурсів.

В круглих дужках вказані коефіцієнти вагомості, тобто важливості, кожної характеристики, визначені методом експертних оцінок.

Рішення поставленого завдання вибору програмного забезпечення методом варіантних мереж показано в таблиці 4.2.

Таблиця 4.2 – Вибір програмного забезпечення методом варіантних мереж

Засіб розробки	Характеристика							Разом
	1(4)	2(3)	3(4)	4(5)	5(4)	6(2)	7(3)	
Oracle	5	5	3	5	5	5	1	105
MySQL	4	4	5	4	5	4	5	111

З результатів аналізу, проведеного методом варіантних мереж, слід, що краща СУБД для реалізації серверної частини БД системи синхронізації даних з 1С, MySQL.

4.3 Архітектурне проектування програмного забезпечення

Розроблювана система синхронізації даних має складатися з таких модулів:

а) змінення конфігурації:

- 1) модуль змінення даних;
- 2) модуль розмежування прав доступу користувачів;
- 3) модуль додавання користувачів;
- 4) модуль присвоєння прав доступу;

б) синхронізації:

- 1) модуль парсингу файлів:
 - XML;
 - електронні таблиці;
- 2) модуль створення об'єктів MODX;
- 3) модуль збору статистики.

Модулі описані раніше згруповані за виконавчими функціями.

4.4 Проведення дослідження

Почнемо розробку самого модулю на прикладі фреймворку Yii 2.

Yii - це високопродуктивний компонентний PHP фреймворк, призначений для швидкої розробки сучасних веб-додатків. Це універсальний фреймворк і може бути задіяний у всіх типах веб-додатків. Завдяки його компонентній структурі і відмінній підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів, як портали, форуми, CMS, магазини або RESTful-додатки.

Для вирішення завдання скористаємося стандартною технологією, яку надає IC, це - протокол обміну між системою "IC: Підприємство" і сайтом, і інструкція розташована за адресою http://v8.1c.ru/edi/edi_std/131/. У даній інструкції вже описаний алгоритм, за яким необхідно реалізувати контролер, що ми зараз і зробимо.

Для початку створимо контролер ApiController в app\controllers (рис 4.1).

Після початку сеансу обміну даними, IC генерує по черзі запити checkauth, init, file і import, після кожного запиту чекає відповідь, якщо відповідь успішна, то переходить до наступного запиту, якщо негативна - то перериває обмін.

Для чого потрібен екшн index? - це буде редіректна або 404 сторінка, щоб не перевизначати дефолтний екшн контролера (рис. 4.2).

Далі, для роботи нам постійно потрібні будуть локальні змінні і змінна сесії (рис 4.3).

Оскільки IC буде відправляти дію контролера в get параметрі \$mode, ми функцією __constuct перетворюємо цю змінну в дію і викликаємо цю дію, коли така є в контролері. Можете визначити тут список дозволених дій, а решта редирект на 302 або 404 (правильніше все ж на 404). Змінну \$_GET['mode'] варто прочистити, наприклад регулярним виразом.

Тепер визначимо функцію checkauth.

Мета цієї функції авторизувати користувача, який звернувся у наше API. Насправді, ІС відправляє логін і пароль для доступу до API (в програмі для цього є відповідні поля). Авторизація проводиться стандартним способом, користувач, який підключається, повинен бути заздалегідь створений в базі даних (рис. 4.4).

Після успішної авторизації, необхідно відповісти на питання архівування (підтримується на сервері методи роботи з архівами чи ні) (рис. 4.5).

Далі сервер ІС передає запит з параметром `mode = file`. Функція `file` повинна зберігати файли імпорту (наприклад, в папку `runtime`) (рис. 4.6).

Змінну `$_GET['filename']` теж необхідно прочистити через регулярний вираз, по суті усі GET параметри необхідно проводити через функції очищення.

Далі залишається лише виконати парсинг даних, дана задача залежить від конкретних умов і завдань і логіка розбору буде своя на кожному сайті. Тому приведено зальну конструкцію методу `import` (рис. 4.7).

```
<?php
namespace app\controllers;
use Yii;
use yii\rest\Controller;
use yii\web\Response;
use yii\web\Session;
class ApiController extends Controller
{
    public function action_index() {
    }
    public function action_checkauth() {
    }
    public function action_init() {
    }
    public function action_file() {
    }
    public function action_import() {
    }
}
?>
```

Рисунок 4.1 – Контролер ApiController

```
public function action_index()
{
    return $this->redirect('/', 302);
}
```

Рисунок 4.2 – Дефолтний екшн контролера ApiController

```

private $mode;
private $filename;
private $session;
public function __construct($id, $module, $config = [])
{
    $this->session = new Session;
    $this->session->open();
    if (!empty($_GET['mode'])) {
        $mode = 'action_' . $_GET['mode'];
    }
    else
    {
        $mode = 'action_index';
    }
    $this->$mode();
    parent::__construct($id, $module, $config);
}

```

Рисунок 4.3 – Назначення локальних змінних ApiController

```

public function action_checkauth()
{
    if ((Yii::$app->getSecurity()->validatePassword($password, $hash)) &
    (!empty($identity)))
    {
        echo "success\n";
        echo $this->session->id."\n";
        echo $this->session->name."\n";
        exit;
    }
    else
    {
        echo "failure\n";
        exit;
    }
}

```

Рисунок 4.4 – Перевірка валідності переданих логіна та паролю

```

public function action_init()
{
    $zip = extension_loaded('zip') ? 'yes' : 'no';
    echo 'zip=' . $zip . "\n";
    echo "file_limit=0\n";
    exit;
}

```

Рисунок 4.5 – Перевірка роботи з архівами

```

public function action_file()
{
    $data = file_get_contents('php://input');
    $path = Yii::getAlias("@app")."/runtime/files/";
    $filename = $_GET['filename'];
    file_put_contents($path.$filename, $data);
    if(file_exists($path.$filename)) {
        $zip = new ZipArchive;
        if($res=$zip->open($path.$filename, ZipArchive::CREATE)) {
            $zip->extractTo($path);
            $zip->close();
            unlink($path.$filename);
            echo "success\n";
            exit;
        }
        else {
            echo "failure\n";
            exit;
        }
    }
    else {
        echo "failure\n";
        exit;
    }
}

```

Рисунок 4.6 – Збереження та розархівування переданих файлів імпорту

```

public function action_import()
{
    $filename=$_GET['filename'];
    $path=Yii::getAlias("@app")."/web/upload/files/";
    $connection=Yii::$app->db;
    $transaction = $connection->beginTransaction();
    try {
        ...
        ...
        $transaction->commit();
    } catch(\Exception $e) {
        $transaction->rollBack();
        echo "failure\n";
        throw $e;
        exit;
    }
    unlink($path.$filename);
    Yii::$app->cache->flush();
    echo "success\n";
    exit;
}

```

Рисунок 4.7 – Обробка файлів імпорту

Проведемо розрахунки середнього часу, який необхідний для синхронізації. Час обміну даними між нашим API і 1С:Підприємство сильно обмежується з боку 1С. Так середня статистика за часом підготовки даних з боку 1С (до відправки даних на сайт) може займати наступний час:

- 20 тис. позицій - 10 хвилин;
- 60 тис. позицій - 15 хвилин;
- 80 тис. позицій - 25 хвилин.

Після підготовки даних програмою 1С, відбувається закачування даних на сайт, швидкість якої може обмежуватися тільки швидкістю каналу передачі даних. Якщо файли передаються в архіві:

- 20 тис. позицій - 2 хвилини;
- 60 тис. позицій - 5 хвилин;
- 80 тис. позицій - 8 хвилин.

Якщо файли передаються без архіву:

- 20 тис. позицій - 6 хвилин;
- 60 тис. позицій - 10 хвилин;
- 80 тис. позицій - 14 хвилин.

Завершальним етапом йде час читання отриманого файлу XML вже безпосередньо на сайті, і запис в базу даних. Час тут не суттєвий, і зазвичай сервер дуже швидко «читає» дані, в залежності від кількості параметрів у

товару. Порахуємо приблизний час необхідний на створення усіх товарів, якщо час необхідний на створення(перевірку) 1 товару буде 1 секунда.

Порахуємо середній час необхідний на повну синхронізацію товарів:

$$- 10 + 2(6) + 20000 \times 0,0016 = 44(48)\text{хвилин};$$

$$- 15 + 5(10) + 60000 \times 0,0016 = 116(121)\text{хвилин};$$

$$- 25 + 8(14) + 80000 \times 0,0016 = 161(167)\text{хвилин}.$$

Як можна побачити числа дуже великі, розглянемо варіанти рішення цієї проблеми. Першим варіантом буде співпраця з 1С програмістом. Так як за замовченням найчастіше завжди вивантажуються усі товари, що є на ресурсі – це не дуже добре, тому що через це ми маємо надлишкові данні, вивантажуються список товарів за якими не було операцій (зміна залишків, ціни, параметрів, опцій), приклад даних зображений на рис. 4.8-4.9.

Кожен товар несе в собі велику кількість текстової інформації, а у файлі ще кожна опція по товару має окремий запис. Ми би мали великий виграш у часі обміну як би цих товарів не було. Рішення цієї проблеми існує, але воно потребує співпраці з 1С програмістом – потрібно налаштувати вивантаження таким чином, щоб вивантажувалися тільки товари там опції за якими були якісь зміни. Це суттєво зменшить кількість товарів та вагу файлів імпорту.

Ще один варіант який я можу запропонувати при цій проблемі – це звіряти MD5 суму товару на сайті та суму товару що передається.

Контрольна сума - деяке значення, розраховане по набору даних шляхом застосування певного алгоритму і використовується для перевірки цілісності даних при їх передачі або зберіганні. Також контрольні суми можуть використовуватися для швидкого порівняння двох наборів даних на нееквівалентність: з великою ймовірністю різні набори даних матимуть нерівні контрольні суми.

Час котрий необхідний для пошуку товару в базі даних та звірка MD5 сум товарів значно менша, аніж перезапис усіх властивостей товару.

```

▼<Товар>
  ▼<Ид>
    а04d5b3b-3d7f-11e8-a3ee-99861ddb57d0#a04d5b3d-3d7f-11e8-a3ee-99861ddb57d0
  </Ид>
  <ШтрихКод>2000000023670</ШтрихКод>
  <Артикул>9760000001</Артикул>
  <Наименование>Безрукавка Паппи, малиновый</Наименование>
  <Наличие>false</Наличие>
  <БазоваяЕдиница Код="796" НаименованиеПолное="Штука" МеждународноеСокращение="РСЕ">шт.</БазоваяЕдиница>
  ▼<ХарактеристикиТовара>
    ▼<ХарактеристикаТовара>
      <Наименование>Размер</Наименование>
      <Значение>6 лет</Значение>
    </ХарактеристикаТовара>
  </ХарактеристикиТовара>
  ▼<ЗначенияРеквизитов>
    ▼<ЗначениеРеквизита>
      <Наименование>ВидНоменклатуры</Наименование>
      <Значение>Товар</Значение>
    </ЗначениеРеквизита>
    ▼<ЗначениеРеквизита>
      <Наименование>ТипНоменклатуры</Наименование>
      <Значение>Товар</Значение>
    </ЗначениеРеквизита>
    ▼<ЗначениеРеквизита>
      <Наименование>Полное наименование</Наименование>
      <Значение>Безрукавка Паппи, малиновый</Значение>
    </ЗначениеРеквизита>
  </ЗначенияРеквизитов>
  ▼<Группы>
    <Ид>243caedd-59e3-11e8-a048-860507b9b884</Ид>
  </Группы>
</Товар>

```

Рисунок 4.8 – Приклад товару в файлі «import.xml»

```

-<Предложение>
  -<Ид>
    54fe972e-1fd5-11e9-a2c3-f962e5f0b1c9#54fe9730-1fd5-11e9-a2c3-f962e5f0b1c9
  </Ид>
  <ШтрихКод>2000000099460</ШтрихКод>
  <Артикул>1673000002</Артикул>
  <Наименование>Гольф Теона, голубой</Наименование>
  <Наличие>true</Наличие>
  <БазоваяЕдиница Код="796" НаименованиеПолное="Штука" МеждународноеСокращение="РСЕ">шт.</БазоваяЕдиница>
  -<Цены>
    -<Цена>
      <Представление>216 грн за шт.</Представление>
      <ИдТипаЦены>1ea5aaf9-ba51-11e2-afd3-f46d04bed993</ИдТипаЦены>
      <ЦенаЗаЕдиницу>216</ЦенаЗаЕдиницу>
      <Валюта>грн</Валюта>
      <Единица>шт.</Единица>
      <Коэффициент>1</Коэффициент>
    </Цена>
  </Цены>
  <Количество>2</Количество>
  -<ЗначенияСвойств>
    -<ЗначенияСвойства>
      <Ид>92e9e8f2-3b0c-11e3-ae40-000c29cde7a2</Ид>
      <Наименование>Размер</Наименование>
      <Значение>92e9e8f5-3b0c-11e3-ae40-000c29cde7a2</Значение>
    </ЗначенияСвойства>
  </ЗначенияСвойств>
</Предложение>
-<Предложение>

```

Рисунок 4.9 – Приклад товару в файлі «offers.xml»

Тепер розробимо найкращий з методів обміну даними – через API 1С. Він дозволяє вирішити проблеми пересилки файлів, швидкості виконання, підтримки та простоти коду. Приклад API наведений на рисунку 4.10.

```
{
  "version": "1.0",
  "author": "Krepkih A.V.",
  "date": "15.03.2019",
  "methods": [
    {
      "name": "APIInfo",
      "call": "/getAPIInfo",
      "comment": ""
    },
    {
      "name": "ProductsList",
      "call": "/getProductsList",
      "comment": "offset / limit"
    },
    {
      "name": "ProductsListCount",
      "call": "/getProductsListCount",
      "comment": ""
    },
    {
      "name": "SubProductsList",
      "call": "/getSubProductsList",
      "comment": "offset / limit"
    },
    {
      "name": "SubProductsListCount",
      "call": "/getSubProductsListCount",
      "comment": ""
    },
    {
      "name": "ColorList",
      "call": "/getColorList",
      "comment": ""
    },
    {
      "name": "BaseColorList",
      "call": "/getBaseColorList",
      "comment": ""
    },
    {
      "name": "MaterialList",
      "call": "/getMaterialList",
      "comment": ""
    },
    {
      "name": "BaseMaterialList",
      "call": "/getBaseMaterialList",
      "comment": ""
    }
  ]
}
```

Рисунок 4.10 – Список доступних методів та параметрів

Як можна побачити все дуже просто, ми бачимо назву методу (поле «name»), список параметрів (поле «comment») котрі можна передати та як до цього методу можна звернутися (поле «call»).

Створимо для цього API універсальний клас.

Так як при кожному виклику нам потрібно створювати HTTP з'єднання, то нам необхідно у методі __construct це і зробити (рис. 4.11).

На рис. 4.12 наведено звернення до методів у API 1С.

```
public function __construct()
{
    $client = new Client([
        'responseConfig' => [
            'format' => Client::FORMAT_JSON
        ],
    ]);
    $this->request = $client->createRequest()
        ->addHeaders(['Authorization' => 'Basic ' . base64_encode($this->login . ':' . $this->password)]);
}
```

Рисунок 4.11 – Список доступних методів та параметрів

```
public function run($method, array $data = null)
{
    $this->request->setUrl('http://nblight.srvtch.com:48888/zavod01/hs/ss_new/' . $method);
    if (!empty($data)) {
        $this->request->setData($data);
    }
    try {
        $response = $this->request->send();
        if ($response->isOk) {
            return $response->data;
        }
    } catch (InvalidConfigException $e) {
        Yii::warning($e->getMessage());
        return false;
    } catch (Exception $e) {
        Yii::warning($e->getMessage());
        return false;
    }
}
```

Рисунок 4.12 – Метод для звернення до методів у API 1С

Як можна побачити цей спосіб більш простий більш простий на стороні сайту, але має свої недоліки – програміст 1С має буду кваліфікованим у розробці API на стороні 1С, так як там своя мова програмування – це може стати не простим завданням.

Гіпотеза доведена - модуль обміну даними між веб-ресурсом та системою 1С:Підприємство, створений в ході науково дослідницької роботи, сприяє підвищенню ефективності роботи інтернет-магазину та вирішую проблеми обміну даними.

Тепер з 1С вивантажуються тільки товари за якими був якийсь «рух» у самій системі. Розрахуємо час необхідний на підготовку файлів:

- 1 тис. позицій – 25 секунд;
- 3 тис. позицій - 1 хвилина;
- 5 тис. позицій - 2 хвилини.

Якщо файли передаються в архіві:

- 1 тис. позицій – 15 секунд;
- 3 тис. позицій - 30 секунд;
- 5 тис. позицій – 45 секунд.

Якщо файли передаються без архіву:

- 1 тис. позицій – 25 секунд;
- 3 тис. позицій - 38 секунд;
- 5 тис. позицій - 52 секунди.

Порахуємо середній час необхідний на повну синхронізацію товарів:

- $0,41 + 0,25(0,41) + 1000 \times 0,0016 = 2,26(2,42)$ хвилин;
- $1 + 0,5(0,63) + 3000 \times 0,0016 = 6,3(6,43)$ хвилин;
- $2 + 0,75(0,86) + 5000 \times 0,0016 = 10,75(10,86)$ хвилин.

Як можна побачити час, необхідний на оновлення товарів став меншим.

Отже, під час проведення експерименту були вирішені такі проблеми обміну даними, як час виконання, надлишковість даних, простота налаштування. Проблема пов'язана з часом виконання обміну була вирішена завдяки тому, що додане порівняння MD5 сум товару у файлі та товару на сайті і якщо вони однакові товар не оновлюється. Надлишковість була вирішена за допомогою видалення товарів за якими не було ніяких змін.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Характеристика науково-дослідного рішення

В атестаційній роботі розглянутий один із можливих підходів до створення модулю обміну даними між 1С та веб-ресурсом. Було розроблено структурну схему модулю, виконане комп'ютерне макетування. Реалізація програмного модулю обміну даними між 1С та веб-ресурсом дозволяє:

- зменшити похибку ручного вводу даних;
- зменшити кількість необхідних людей на підприємстві;
- зменшити втрати через неактуальність даних на веб-ресурсі.

5.2 Етапи виконання НДР, їх трудомісткість та заробітна плата

У процесі виконання науково-дослідної роботи був проведений огляд існуючих методів обміну даними, на основі аналізу спеціальної літератури розглянуті принципи побудови та схеми роботи існуючих модулів, розроблено програмне забезпечення, проведено макетування спеціального програмного забезпечення універсального модулю обміну даними.

Умовно науково-дослідну роботу (НДР) можна розділити на три етапи: підготовчий, основний і заключний.

На стадії виконання підготовчого етапу були виконані підбір і вивчення літератури та існуючих модулів для проведення відповідних до постановки задачі робіт. Проведено пошук інформації в Internet.

На етапі виконання основної частини НДР були виконані такі роботи:

- огляд існуючих аналогічних програмних засобів;
- розробка універсального модулю обміну даними на базі персонального комп'ютера.

У заключній частині проводяться: аналіз результатів виконання НДР, складання звіту по НДР, захист звіту.

Найбільш складною й відповідальною частиною при плануванні НДР є розрахунок трудомісткості робіт, тому що трудові витрати часто становлять основну частину вартості науково-дослідних робіт і безпосередньо впливають на строки розробки.

Дану роботу виконували два фахівця: middle PHP developer, PHP developer. За даними сайту по пошуку праці місячна заробітна плата middle PHP developer складає 22000 грн., PHP developer – 12000 грн.

Проведемо розрахунок трудовитрат і заробітної плати виконавця робіт.

Середньоденна заробітна плата виконавця робіт ($Z_{\text{ср.дн.}}$) розраховується за формулою:

$$Z_{\text{ср.дн.}} = \frac{Z_{\text{ср.міс.}}}{n}, \quad (5.1)$$

де $Z_{\text{ср.міс.}}$ – середньомісячна зарплата виконавця роботи;

n – число робочих днів у місяці, ($n=22$).

Середньоденна заробітна плата керівника робіт складає:

$$Z_{\text{ср.дн.}} = \frac{22000}{22} = 1000 \text{ (грн)}.$$

Середньоденна заробітна плата інженера-метролога складає:

$$Z_{\text{ср.дн.}} = \frac{12000}{22} = 545,45 \text{ (грн)}.$$

Етапи виконання НДР, перелік і зміст робіт, трудомісткість їх виконання, заробітна плата виконавців робіт представлені в табл. 5.1.

Таблиця 5.1 – Розрахунок трудовитрат і заробітної плати виконавця робіт

Перелік робіт	Кількість виконавців	Посада виконавця	Трудоємність робіт, люд.-днів	Середньоденна заробітна плата, грн.	Сума заробітної плати, грн.
1. Підготовчий етап	1	PHP developer	1	545,45	545,45
1.1. Розробка та затвердження ТЗ	1	Middle PHP developer	1	1000	1000
1.2 Підготовка довідкових матеріалів та даних для виконання НДР	1	PHP developer	2	545,45	1090,9
2. Основний етап	-	-	-	-	-
2.1 Постановка задачі	1	Middle PHP developer	1	1000	1000
2.2 Розробка теоретичної частини: методи обміну даними	1	PHP developer	3	545,45	1631,35
2.2 Огляд існуючих засобів обміну даними	1	PHP developer	1	545,45	545,45
2.3 Розробка універсального модулю обміну на базі ПК	2	PHP developer, Middle PHP developer	10	545,45 1000	15454,5
2.4 Алгоритмізація розв'язання задачі ПЗ	2	PHP developer, Middle PHP developer	2	545,45 1000	3090,9
2.5 Тестові розрахунки на ПК	1	PHP developer	1	545,45	545,45
3. Заключний етап	-	-	-	-	-
3.1 Аналіз результатів проведення роботи	1	PHP developer	1	545,45	545,45
3.2 Формування висновків та пропозицій за темою дослідження	1	PHP developer	1	545,45	545,45
3.3 Технічне оформлення звіту про виконання НДР	1	PHP developer	1	545,45	545,45
3.3 Захист звіту на технічній раді	1	PHP developer	1	545,45	545,45
Всього	2	-	26	-	27085,8

5.3 Розрахунок одноразових витрат на розробку НДР

Калькуляція собівартості розраховується відповідно до існуючих нормативних актів України. До складу калькуляції входять такі статті витрат:

- матеріальні витрати;
- витрати на оплату праці;
- єдиний соціальний внесок;
- амортизація основних засобів (вартість машинного часу);
- витрати на спожиту електроенергію;
- інші витрати.

До інших витрат відносяться адміністративні витрати (водопостачання, водовідведення, опалення, освітлення) та вартість послуг зв'язку.

Матеріальні витрати визначаються витратами на матеріали, визначені їх потребою для виконання робіт, і цін, що діють на момент складання калькуляції. Матеріальні витрати розраховуються за такою формулою:

$$M = \sum_{j=1}^n Q_j \times C_j, \quad (5.2)$$

де M – сумарні витрати на матеріали, в тому числі малоцінні предмети, що швидко зношуються (носії, папір, канцелярське приладдя тощо), або на літературу, яка необхідна для проведення роботи, тощо;

Q_j – кількість використаних одиниць j -го виду матеріалів, $j=(1 \div n)$;

C_j – ціна одиниці j -го виду матеріалів.

Розрахунок матеріальних витрат представлено в табл. 5.2.

Витрати на оплату праці розраховуються виходячи з необхідного для виконання робіт складу й кількості працівників, а також із середньомісячної заробітної плати. Відповідно до проведених розрахунків витрати на оплату праці виконавців роботи дорівнюють 27085,8 грн.

Таблиця 5.2 – Розрахунок матеріальних витрат

Найменування	Од. вим.	Кількість	Ціна, грн	Сума, грн.
Олівець механічний	шт.	2	20	40
Папір	уп.	1	150	150
Степлер	шт.	1	30	30
Скріпки для степлеру	уп.	1	10	10
Флешка	шт.	1	350	350
Заправка для картриджу	шт.	1	400	400
Усього	-	7	-	980

Єдиний внесок на загальнодержавне соціальне страхування (ЄСВ) – консолідований страховий внесок, збір якого здійснюється в систему загальнообов’язкового державного соціального страхування в обов’язковому порядку і на регулярній основі з метою забезпечення захисту у випадках, передбачених законодавством, прав застрахованих осіб і членів їх сімей на отримання страхових виплат (послуг) за діючими видами загальнообов’язкового державного соціального страхування.

Для об’єкта дослідження ставка єдиного соціального внеску дорівнює 22% від витрат на оплату праці, тобто розмір ЄСВ дорівнює 5958,87 грн.

При виконанні НДР застосовувалось наступне обладнання: 2 комп’ютери вартістю 12000 грн. та принтер вартістю 5000 грн (термін служби для амортизації 4-5 роки). Вищенаведене устаткування є власністю організації виконавця, тому доцільно розрахувати суму амортизаційних відрахувань на період виконання НДР. Амортизація основних засобів:

$$AB = \sum_{k=1}^L \frac{BO_k}{TE_k} \times T, \quad (5.3)$$

де AB – сума амортизаційних відрахувань, нарахованих під час проведення науково-дослідницької роботи;

BO_k – вартість основних засобів k -го виду;

TE_k – термін експлуатації основних засобів k -го виду, днів;

T – термін науково-дослідницької роботи, днів;

L – кількість видів обладнання.

Підставивши відомі значення у (5.3), визначимо величину амортизаційних відрахувань. Отже маємо:

$$AB = \frac{2 \times 12000 \times 26}{528} + \frac{5000 \times 5}{528} = 1229,16 \text{ (грн.)}$$

Витрати на використану обладнанням електроенергію:

$$Z_e = M \cdot t \cdot T_{кВт}, \quad (5.4)$$

де M – потужність устаткування, тобто кількість енергії, споживаної за одиницю часу (кВт/година);

t – кількість годин використання устаткування за період проведення науково-дослідницької роботи;

$T_{кВт}$ – тариф, тобто вартість використання 1 кВт електроенергії.

Споживна потужність комп'ютера складає 1,5 кВт та принтера 0,8 кВт за годину. Тариф споживачів, складає 1,80065 грн./кВтгодин (без ПДВ). Підставивши значення у (5.4), визначимо величину витрат на спожиту електроенергію:

$$Z_e = 1,5 \times 208 \times 1,8 \times 2 + 0,8 \times 208 \times 1,8 = 1422,72 \text{ грн.}$$

До інших статей витрат відносяться такі:

– адміністративні витрати: (водопостачання, водовідведення, освітлення, опалення), які прийнято рахувати у розмірі 20% від витрат на оплату праці;

– вартість оплати послуг зв'язку.

Вартість оплати послуг зв'язку становитиме:

а) Інтернет – із розрахунку 300 грн. на місяць (безлімітний пакет); всього 150 грн. за 1 місяць виконання НДР;

б) телефон – із розрахунку 120 грн на місяць; всього 120 грн. за 1 місяць.

За час виконання НДР витрати на відрядження, аутсорсинг, інформаційні послуги та маркетингові заходи не мали місця.

Результати розрахунку кошторису витрат, тобто одноразових витрат, на виконання НДР наведені в табл. 5.3.

Таблиця 5.3 – Кошторис витрат на розробку НДР

№ з/п	Стаття витрат	Сума, грн.
1	Заробітна плата	27085,8
2	Єдиний соціальний внесок (22,0 % від п.1)	5958,87
3	Матеріальні витрати	980
4	Амортизація основних засобів	1229,16
5	Витрати на спожиту електроенергію	1422,72
6	Інші витрати, у тому числі:	-
6.1	адміністративні витрати (20% від п.1)	13542,9
6.2	вартість послуг зв'язку	450
	Всього витрати	50 669,45

Таким чином, кошторис витрат на виконання даної НДР відбиває сумарні витрати за статтями п.1 ÷ п.6 та складає 50 669,45 грн.

5.4 Оцінка результатів науково-дослідної роботи

З огляду на проектно-дослідницький характер роботи і відсутності конкретних об'єктів впровадження наведені загальні дані економічного ефекту впровадження розробки. Структурна схема очікуваного ефекту від реалізації запропонованих заходів наведена на рис. 5.1.

Практична значущість отриманих результатів полягає у тому, що запропоноване програмне рішення забезпечує:

а) підвищення ефективності бізнес-процесів, завдяки:

1) зменшенню необхідного часу для оновлення інформації на веб-ресурсі;

2) високій актуальності інформації на веб-ресурсі;

б) підвищення керованості бізнес-процесів, завдяки:

- 1) створенню повної достовірної бази даних за товарами на веб-ресурсі, з'являється можливість вести повний наскрізний облік товарів;
- 2) автоматизованому формуванню замовлень на веб-ресурсі;
- 3) зменшенню витрат часу менеджера по продажах з 4 годин до 1 години, зменшенню витрат часу контент-менеджера з 5 днів до 1 дня.



Рисунок 5.1 – Структурна схема очікуваного ефекту

Стає можливим регулярний і оперативний аналіз даних, що вносяться на веб-ресурс. Даний програмний продукт дає можливість зменшити кількість менеджерів на підприємстві, так як автоматизується частка їх обов'язків.

Користувач сайту отримує актуальні дані про товари (послуги), а також можливість слідкувати за станом його замовлення в реальному часі без звернення до менеджера.

Таким чином, впровадження розробки забезпечить:

- зменшення похибок ручного вводу, так як контент-менеджеру не треба буде створювати товари (послуги) на обох ресурсах;
- підвищення достовірності даних про товари (послуги), їх наявність, кількість, ціну, характеристики;
- зниження економічних втрат на 5% в рік за рахунок зниження неактуальної інформації на веб-ресурсі.

ВИСНОВКИ

Результатом атестаційної магістерської роботи є сформовані методи покращення синхронізації між веб-ресурсом та 1С:Підприємство.

Під час дослідження предметної області було проаналізована проблему синхронізації даних між 1С та стороннім web-додатком. Проведено огляд аналогічних систем синхронізації даних. При огляді аналогів, були виявлені їх переваги та недоліки.

Розглянуто сучасні методи обміну даними між с веб-ресурсом та 1С:Підприємство. Проведено їх аналіз та порівняння, побудована порівняльна таблиця існуючих методів обміну.

Спроектвана функціональна модель системи, розроблені алгоритми синхронізації, а також алгоритми роботи всієї системи. Побудовані діаграми ілюструють схеми інформаційних зв'язків. Спроектвана і реалізована архітектура системи синхронізації даних з 1С. Проведено детальне проектування системи. Обґрунтовано вибір мови реалізації.

Розроблено основні елементи синхронізації з 1С: Підприємство при обміні через файли та за допомогою API. Вирішенні проблеми, які були в інших модулях.

В економічній частині роботи здійснено розрахунки по формуванню витрат на проведення НДР. Таким чином, впровадження розробки забезпечить:

- зменшення похибок ручного вводу, за рахунок того, що контент-менеджеру не треба буде створювати товари (послуги) на обох ресурсах;
- підвищення достовірності даних про товари (послуги), їх наявність, кількість, ціну, характеристики;
- зниження економічних втрат на 5% в рік за рахунок зниження неактуальної інформації на веб-ресурсі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Wikimedia Foundation, Inc. Интернет-магазин. URL: <https://uk.wikipedia.org/wiki/Интернет-магазин> (дата звернення 01.11.2019).
2. Wikimedia Foundation, Inc. 1С:Підприємство. URL: <https://uk.wikipedia.org/wiki/1С:Підприємство> (дата звернення 01.11.2019).
3. Wikimedia Foundation, Inc. Центр Автоматизации Нургуш. Функциональные возможности 1С:Бухгалтерии. URL <http://www.nurgush.ru/horz/br8/about.php> (дата звернення 04.11.2019).
4. Кондратьев А.В., Вовк А.В. Интеграция интернет-магазинов с системой 1С:Предприятие // Полиграфические, мультимедийные и web-технологии: материалы IV Международной научно-технической конференции (14-17 мая 2019, г. Харьков). С. 90-91.
5. Кондратьев А.В., Вовк А.В. Синхронизация системы 1С:Предприятие с интернет-магазином // Полиграфические, мультимедийные и web-технологии: материалы IV Международной научно-технической конференции (14-17 мая 2019, г. Харьков). С. 176-179.
6. ООО «1С» Система программ. URL: <http://v8.1c.ru/> (дата звернення 22.11.2019).
7. Wikimedia Foundation, Inc. 1С-Битрикс. URL: <https://uk.wikipedia.org/wiki/1С-Битрикс> (дата звернення 05.11.2019).
8. 1С. 1С-битрикс. URL: <https://www.1c-bitrix.ru/> (дата звернення 08.11.2019).
9. MODX. shopmodx 0.2.0-beta. URL: <https://modx.com/extras/package/shopmodx> (дата звернення 08.11.2019).
10. MODXClub. Документация по MODX. MODXClub. URL: <https://modxclub.ru/topics/dokumentacziya-po-shopmodx-1283.html> (дата звернення 11.11.2019).
11. Wikimedia Foundation, Inc. SOAP. URL: <https://ru.wikipedia.org/wiki/SOAP> (дата звернення 12.11.2019).

12. MODStore. mSkлад. <https://modstore.pro/packages/import-and-export/msklad> (дата звернення 14.11.2019).
13. MODXpro. Синхронизация с 1С. <https://modx.pro/development/6817-synchronization-with-1c/> (дата звернення 14.11.2019).
14. BusinesStudio. Нотация IDEF0. URL: <http://www.businessstudio.ru/wiki/docs/current/doku.php/ru/csdesign/bpmodeling/idef0> (дата звернення 15.11.2019).
15. Wikimedia Foundation, Inc. Алгоритм. URL: <https://ru.wikipedia.org/wiki/Алгоритм> (дата звернення 15.11.2019).
16. AlgoritMyYakovlev. Система команд исполнителя (СКИ). URL: <https://sites.google.com/site/algoritmyyakovlev/ispolniteli/sistema-komand-ispolnitela-ski> (дата звернення 17.11.2019).
17. BookKbsu. Что такое словесный способ записи алгоритмов. URL: http://book.kbsu.ru/theory/chapter7/1_7_5.html (дата звернення 17.11.2019).
18. Студентик.РФ. Способы описания алгоритмов. URL: <http://www.студентик.рф/content/способы-описания-алгоритмов> (дата звернення 17.11.2019).
19. Wikimedia Foundation, Inc. Блок-схема. URL: <https://ru.wikipedia.org/wiki/Блок-схема> (дата звернення 18.11.2019).
20. Microsoft Inc. C#. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/csharp> (дата звернення 19.11.2019).
21. The PHP Group. PHP. URL: <http://php.net/manual/ru/intro-what-is.php> (дата звернення 22.11.2019).
22. Oracle Inc. Java Software. URL: <https://www.oracle.com/java/index.html> (дата звернення 22.11.2019).
23. Oracle Inc. Home. URL: <https://www.oracle.com/ru/index.html> (дата звернення 22.11.2019).
24. Oracle Inc. Oracle DB 12c. URL: <https://www.oracle.com/database/index.html>. (дата звернення 23.11.2019).
25. Microsoft Inc. SQL Server 2016. URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016> (дата звернення 23.11.2019).