

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повна назва факультету)

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

(назва освітнього ступеня)

на тему: _____

Виконав(ла): студент(ка) _____ курсу, групи _____
спеціальності _____

(шифр і назва спеціальності)

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
20__

АНОТАЦІЯ

Магістерська робота на тему «Розробка розширення для перекладу тексту для Magento 2 на мові PHP» Бережний Богдан Ігорович – Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПм–61 // Тернопіль, 2020.

Дана робота містить 68 сторінки, 26 рисунків, список використаної літератури з 18 найменувань та 3 додатки.

MAGENTO 2, TRANSLATION INLINE, ІНТЕРНЕТ-МАГАЗИН, ЛОКАЛІЗАЦІЯ, NGINX, REST API, PHP, UI COMPONENTS, PRODUCTION MODE, DEVELOPER MODE.

В атестаційній роботі розглянуті проблеми локалізації інтернет-магазинів на платформі Magento 2, проведено дослідження предметної області та пошук методів для вирішення проблеми. Описано етапи проектування та конструювання програмного забезпечення, яке вирішує проблеми, які виникають при роботі над перекладами. Наведено приклад розгортання та експлуатації системи.

Метою магістерської роботи є розробка розширення для багатомовних інтернет-магазинів на платформі Magento 2, яке дозволить працювати над локалізацією сайту без залучення сторонніх розробників, та вести роботи одразу ж на production сервері.

Під час виконання даної роботи використано мову програмування PHP із застосуванням технологій Magento Framework, середовищем розробки обрано PHP Storm, веб-сервер Apache 2 та база даних MySQL .

Результатом розробки стало розширення, яке дозволяє локалізувати інтернет-магазин без залучення розробників, одразу ж на production сервері, при цьому не зупиняючи роботу сайту.

ANNOTATION

Master's thesis on "Development of an extension for text translation for Magento 2 on PHP" Bereznyi Bohdan Ihorovych - the Ivan Pulyuy National Technical University of Ternopil, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, SPm-61 group // Ternopil, 2020.

This work contains 68 pages, 26 figures, a list of references from 18 titles, and 3 appendices.

MAGENTO 2, TRANSLATION INLINE, INTERNET SHOP, LOCALIZATION, NGINX, REST API, PHP, UI COMPONENTS, PRODUCTION MODE, DEVELOPER MODE.

The certification work considers the problems of localization of online stores on the platform Magento 2, conducted a study of the subject area, and search for methods to solve the problem. The stages of designing and constructing software that solves problems that arise when working on translations are described. An example of system deployment and operation is given.

The purpose of the master's thesis is to develop an extension for multilingual online stores on the Magento 2 platform, which will allow you to work on localizing the site without the involvement of third-party developers and work immediately on the production server.

In this work, the PHP programming language using Magento Framework technologies was used, PHP Storm, Apache 2 web server, and MySQL database were chosen as the development environment.

The result of the development was an extension that allows you to localize the online store without the involvement of developers, immediately on the production server, without downtime the site.

ЗМІСТ

ВСТУП.....	5
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
1 АНАЛІЗ ВИМОГ ТА ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Патентування програмного забезпечення.....	8
1.2 Опис системи.....	10
1.3 Аналіз вимог.....	12
1.4 Визначення акторів та прецедентів системи.....	13
2. ПРОЕКТУВАННЯ ТА ОПИС АРХІТЕКТУРИ.....	20
2.1 Концептуальна архітектура системи.....	20
2.2 Взаємодія між компонентами.....	23
2.3 Специфікації для основних класів.....	24
3 РОЗРОБКА, РОЗГОРТАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ.....	27
3.1 Створення базової структури.....	27
3.3 Тестування класів та методів.....	38
3.4 Інтеграційне тестування.....	39
3.5 Розгортання програмної системи та системні вимоги.....	40
3.6 Опис типових схем використання системи.....	41
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	48
4.1 Охорона праці.....	48
4.2 Безпека в надзвичайних ситуаціях.....	51
ВИСНОВКИ.....	54
СПИСОК ЛІТЕРАТУРИ.....	56
ДОДАТКИ.....	58

ВСТУП

В даний час більшість інтернет-магазинів є багатомовними та підтримують різні валюти, багато хто керується простим принципом, чим більше мов матиме сайт – тим ширшим буде коло користувачів яких він зможе охопити. Здебільшого такі магазини активно розвивають контент та працюють над SEO, для них важливо, щоб магазин був повністю локалізованим. В Magento 2 велика частина даних змінюється через адмін-панель, але не весь контент може бути перекладений цим методом. Для прикладу переклад для кнопок, посилань та елементів меню може бути зроблений лише тоді, коли режим роботи сайту – developer, це означає що робота над цією областю контенту є недоступною для звичайних користувачів, внаслідок чого ці завдання переходять до розробників, а це в свою чергу сповільнює та збільшує вартість локалізації .

Метою проекту є – пошук методів спрощення та здешевлення процесу локалізації, та подальше проектування та розробка розширення, яке відповідатиме поставленим вимогам та дасть можливість перекласти роботу над перекладами з розробників на контент-менеджерів чи звичайних користувачів, які мають потрібний рівень доступу. Важливою вимогою є здатність програми працювати за умови роботи сайту в режимі – production, тобто не заважати клієнтам вибрати товар та оформляти замовлення, даний процес повинний бути непомітним для звичайних користувачів інтернет-магазину.

Об'єктом дослідження виступає процес перекладу тексту за допомогою функціоналу під назво «Translate Inline», який потребує додаткового вивчення та вдосконалення.

Предметом дослідження є – проектування та розробка розширення для перекладу тексту для Magento 2.

Вибір на користь платформи Magento був зумовлений тим, що Magento 2 – є одною з найбільших та найінноваційніших платформ для електронної комерції, та

в порівнянні з іншими платформами тут знаходяться дійсно великі клієнти в яких зазвичай є потреби в сторонніх розширеннях.

Провівши дослідження базового функціоналу Magento 2 під назвою «Translate Inline», було вирішено залишити поточну логіку для перекладу тексту та розробити зручний та зрозумілий користувацький інтерфейс, який розширюватиме можливості модуля та дасть змогу керувати перекладами з адмін-панелі сайту.

Інтерфейс буде максимально зрозумілий користувачеві, тому що розробка на Magento – це розширення функціоналу платформи, а завдяки продуманій архітектурі та використанню уже готових компонентів, все що нам потрібно це використати їх у своєму розширенні. Таким чином ми за короткий час матимемо багатофункціональне розширення, яке матиме уже добре знайомий користувачам Magento 2 інтерфейс.

Розширення матиме велику цінність та практичне значення для багатомовних інтернет-магазинів, так як дасть можливість швидкої та дешевої локалізації сайту та відкине потребу в залученні сторонніх розробників для цього процесу. Це зекономить багато часу, який йде на комунікацією між працівником, який надає переклад для того чи іншого слова та розробником.

Актуальність проблеми та основні методи локалізації інтернет-магазину також розглянуто в публікації «Як перекласти інтернет-магазин на платформі Magento 2 на українську мову?».

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	База даних.
ПЗ	Програмне забезпечення.
CLI	Обробляє команди до комп'ютерної програми у вигляді рядків тексту. Програма, яка обробляє інтерфейс, називається інтерпретатором командного рядка або процесором командного рядка.
PHP	Це скриптова мова програмування, яка була створена для генерації HTML-сторінок на стороні веб-сервера. Вона є однією з найпопулярніших мов, які використовуються у сфері веб-розробок.
REST	Це архітектурний стиль, який визначає набір обмежень, що використовуються для створення веб-служб. Веб-служби, що відповідають архітектурному стилю REST, називаються RESTful Web services, забезпечують взаємодію між комп'ютерними системами в Інтернеті.
SQL (structured query language)	Це мова програмування призначена для взаємодії користувача з базами даних, що використовується для створення запитів, оновлення і керування реляційними базами даних, створення схеми БД та її модифікації.
UI (User Interface)	Користувацький інтерфейс.

1 АНАЛІЗ ВИМОГ ТА ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Патентування програмного забезпечення

В багатьох ІТ-компаніях чи підприємців виникає питання пов'язане із охороною своїх нематеріальних прав, адже майже вся вартість ІТ-підприємства – це вартість об'єктів інтелектуальної власності, якими воно володіє. Так у випадку копіювання конкурентом вашої розробки, у вас не буде доказів того, що це дійсно ваша розробка.

Для того щоб, захистити свою власності, ІТ-компанії повинні використовувати комплекс заходів, а саме:

- Залучення інструментів, які надає авторське право;
- Реєстрація торговельних марок;
- Правильне оформлення документів для передачі прав на ПЗ;
- Створення та нагляд над виконанням положень про комерційну таємницю;
- Патент на програмне забезпечення.

Найскладнішим в реалізації є патентування ПЗ, проте існуючий патент на програмне забезпечення має багато переваг:

- Найвищий рівень охорони інтелектуальної власності;
- Конкурентам не вдасться трішки відредагувати код та потім представити це як власну розробку;
- Патент автоматично заборонятиме отримання патентів на подібні розробки.
- Мінуси патентування ПЗ:
 - Складна процедура реєстрації;
 - Велика вартість оформлення прав і необхідність підтримки патент в активному стані;
 - Довготривалість процедури.

В кожній країні процедура реєстрації патенту на програмне забезпечення може відрізнятись.

Патент на програмне забезпечення можна отримати у вигляді патенту на винахід і патенту на корисну модель.

Нижче наведено базові кроки, які необхідно пройти для того щоб, отримати патент на ПЗ:

1. Перш за все потрібно здійснити патентний пошук за вашою розробкою. Даний етап не є обов'язковим, проте він дозволить виявити однакові чи подібні розробки;

2. Оформити документи та матеріали заявки для подальшої передачі в патентний офіс країни, після чого сплачується мито за заявку;

3. Тепер можна здійснити експертизу. Експертиза передбачає два етапи: формальний, та кваліфікаційний;

4. Після проведення експертизи отримуємо повідомлення про реєстрацію, та оплачуємо додаткове мито за видачу патенту;

5. Надалі потрібно буде платити періодичні внески для підтримки чинності патенту. Якщо пропустити оплату – патент автоматично втрачає свою чинність.

Максимальним терміном дії патенту є – 20 років.

Для перевірки наявних патентів використано міжнародну патентну базу Espacenet. Даний тип продукту відноситься до групи зображеної на Рисунку 1.1.1.



Рисунок 1.1.1 – Класифікація ПЗ

Результат пошуку схожих ПП зображено на рисунку Рисунку 1.1.2.

The screenshot shows the Espacenet search results for 'magento 2 translation system'. The search results list 26 results, with the first result selected. The selected result is '1. DATA TRANSLATION SYSTEM AND METHOD FOR MULTI-PLATFORM E-COMMERCE DISTRIBUTION SYSTEM' by VIOLET IO INC [US], filed on 2018-04-13. The page also displays bibliographic data, including applicants (VIOLET IO INC [US]), inventors (SCHULZ BRANDON [US], ZABEL RHEN [US]), and classifications (IPC: G06Q30/06; CPC: G06Q30/0619 (EP,US)). A front-page drawing from US2019318402A1 is shown, illustrating a system architecture with components like Product DB, Operational DB, Product Distribution Platform, and various applications (Media Apps, VR Apps, AR Apps, Voice Apps, Other Apps) connected via SYNC and API interfaces.

Рисунок 1.1.2 – Пошук по патентній базі

Після проведення пошуку в патентних базах, патентів, які б перешкождали розробці розширення для перекладу тексту для Magento 2 не виявлено, а це означає що на цьому етапі можна приступити до аналізу вимог до системи.

1.2 Опис системи

Magento 2 – це відкрита система для ведення електронної комерції в мережі інтернет. На даний момент це остання версія провідної платформи електронної комерції корпоративного класу, що використовується понад 200 000 інтернет-магазинів. За оцінками, кожен четвертий бізнес використовує програмне забезпечення Magento.

Magento 2 має чудовий функціонал під назвою «Inline Translate»[2]. Приклад роботи з функціоналом зображено на рисунку 1.2.1.

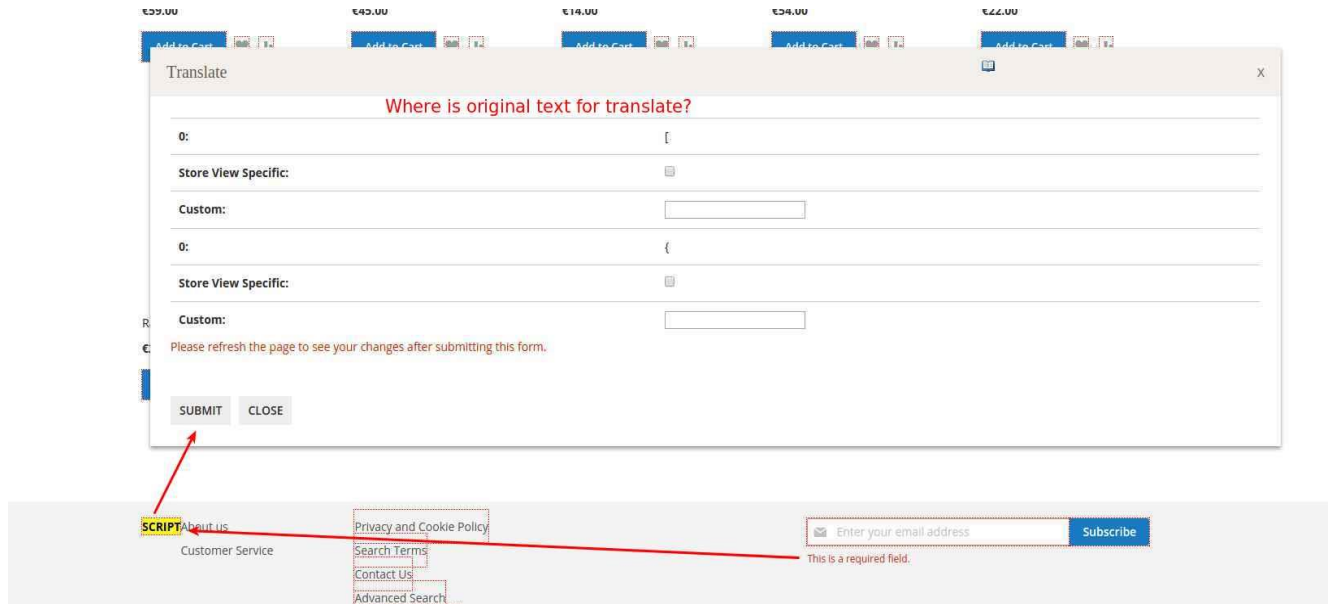


Рисунок 1.2.1 – Приклад роботи з функціоналом «Inline Translate»

Він дозволяє робити переклад контенту та зберігає зміни у базі даних Magento 2. Мінусом цього функціоналу є те, що він не працює коли магазин знаходиться у «Production» режимі, тобто безкорисним на «живому» сайті.

Потрібно спроектувати модуль «Magefan_Translation» який дозволить користувачу в панелі адміністратора Magento (System > Tool > Translation Management) працювати з цим функціоналом. Тобто дозволить користувачу:

- Переглядати записи;
- Редагувати записи;
- Додавати нові записи.

Вимоги до реалізації:

1. Списки (grid) та форми повинні бути реалізовані через UI компоненти;

2. Діставати об'єкти потрібно буде через репозиторію (клас який потрібно буде створити);
3. Додати підтримку REST API, щоб дати змогу отримувати список перекладів та редагувати їх;
4. Додати можливість експорту-імпорту.

1.3 Аналіз вимог

Вимоги до програмної системи часто класифікуються як функціональні, нефункціональні й вимоги предметної області.

Функціональні вимоги до розроблюваної системи:

1. Система повинна надавати користувачу можливість створення, редагування, видалення перекладів;
2. Всі переклади повинні виводитись в одній таблиці з можливістю пошуку та фільтрації даних;
3. Експорт усіх та вибраних даних;
4. Імпорт даних;
5. Робота з кешом сайту.

Нефункціональні вимоги до розроблюваної системи:

1. Вимоги до продукту:
 - 1.1. Розроблюваний продукт має бути розроблений на платформі Magento 2;
 - 1.2. Продукт має бути доступним лише тим користувачам які мають відповідний дозвіл на читання, писання чи видалення даних;
 - 1.3. Продукт повинен стабільно працювати коли сайт знаходиться в «production» режимі;
 - 1.4. Розширення повинно підтримувати усі мови встановлені на сайті;
 - 1.5. Система має працювати без перебоїв.

2. Організаційні вимоги:

2.1. Розробка системи та створення супутньої документації виконується на основі стандарту операційних систем які були вказані в вимогах до продукту;

2.2. Розробка системи включає в себе всі фази життєвого циклу від ідеї до реалізації готового продукту;

2.3. Розробка система має відбуватись на таких мовах програмування як, Java Script, PHP.

3. Зовнішні вимоги:

3.1. Система не повинна працювати з даними користувачів;

3.2. Для користування системою, клієнт має погодитись з політикою користування даною системою та увімкнути розширення.

Вимоги до реалізації:

1. Списки (grid) та форми повинні бути реалізовані через UI компоненти;

2. Діставати об'єкти потрібно буде через репозиторію (клас який потрібно буде створити);

3. Додати підтримку REST API, щоб дати змогу отримувати список перекладів та редагувати їх;

4. Додати можливість експорту-імпорту.

1.4 Визначення акторів та прецедентів системи

До основних функцій системи можна віднести: додання, оновлення і видалення перекладів, фільтрацію, пошук, імпорт та експорт даних.

Після проведення аналізу предметної області можна виявити наступних акторів (Рисунок 1.4.1):

– Адміністратор: може створювати, редагувати, видаляти переклади та здійснювати імпорт, експорт, фільтрацію та пошук даних., створює ролі користувачів та надає дозволи для кожної з ролей;

– Користувач: залежно від дозволу може створювати, редагувати, видаляти переклади та здійснювати імпорт, експорт, фільтрацію та пошук даних.

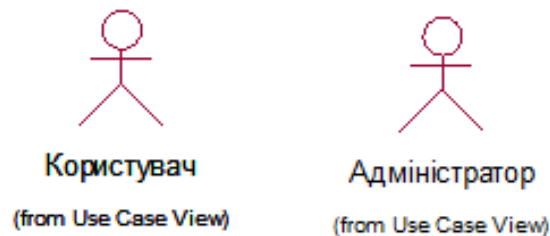


Рисунок 1.4.1 – Актори системи

Після створення акторів системи ми можемо створити варіанти використання для цих акторів:

1. Варіанти використання адміністратора

1.1. Створити переклад

– Призначення: створення нового перекладу для тексту на вітрині чи адмін-панелі магазину;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче створити переклад йому потрібно натиснути на кнопку «Add Translation» після ввести у форму оригінальний текст для перекладу, переклад, та вибрати «Store View» для якого призначений переклад;

– Вихід: список перекладів буде оновлено, після очищення кешу переклад відобразиться на вітрині магазину.

1.2. Редагування перекладу

– Призначення: якщо адміністратор хоче змінити переклад чи «Store View»;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче змінити інформацію про переклад, він повинен вибрати рядок з потрібним записом у таблиці, та натиснути кнопку «Edit»,

або ж змінити переклад за допомогою «Inline Edit», внести зміни та натиснути кнопку «Save» або «Save and Continue»;

– Вихід: список перекладів буде оновлено, після очищення кешу переклад відобразиться на вітрині магазину.

1.3. Видалення перекладу

– Призначення: якщо адміністратор хоче видалити переклад;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче видалити переклад, він повинен вибрати в таблиці один або декілька записів та натиснути на кнопку «Delete Translation», після чого підтвердити дію;

– Вихід: переклад буде видалений, після очищення кешу переклад зникне з сайту.

1.4. Створення користувача

– Призначення: якщо адміністратор хоче створити нового користувача та присвоїти йому певну роль у системі;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче створити нового користувача він повинен перейти в пункт меню System -> Permissions -> All Users заповнити форму з контактними даними, вказати користувацьку роль та зберегти данні;

– Вихід: створиться новий користувач з доступом до адмін-панелі сайту.

1.5. Створення користувацької ролі

– Призначення: якщо адміністратор хоче обмежити доступ до певних ресурсів для деяких користувачів;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче створити чи змінити дозволи для ролі користувача, йому необхідно перейти в пункт меню System -> Permissions -> User Roles обрати потрібну роль, встановити права на бажане розширення та натиснути кнопку «Save»;

– Вихід: дозволи для користувачів з цією роллю будуть змінені.

1.6. Експорт даних

- Призначення: якщо адміністратор хоче експортувати дані до іншого магазину;
- Актор: адміністратор;
- Вхід: якщо адміністратор хоче експортувати дані до іншого магазину йому потрібно вибрати переклади з таблиці (в іншому випадку експортнуться усі дані) натиснути на кнопку «Export» та вибрати формати даних csv або excel;
- Вихід: буде сформований та завантажений файл з вибраними даними у відповідному форматі.

1.7. Пошук даних

- Призначення: якщо адміністратор хоче знайти переклад;
- Актор: адміністратор;
- Вхід: якщо адміністратор хоче знайти переклад він повинен ввести текст у полі пошуку та натиснути на іконку пошуку
- Вихід: список перекладів буде оновлено й відображено в таблиці відповідно до критеріїв пошуку.

1.8. Фільтрування даних

- Призначення: якщо адміністратор хоче відібрати данні;
- Актор: адміністратор;
- Вхід: якщо адміністратор хоче відібрати данні, він повинен задати параметри для фільтрації такі як «Store View» та натиснути на кнопку «Search»;
- Вихід: список перекладів буде оновлено й відображено в таблиці відповідно до фільтру.

1.9. Імпорт даних

- Призначення: якщо адміністратор хоче додати нові переклади з іншого сайту;
- Актор: адміністратор;
- Вхід: якщо адміністратор хоче додати нові переклади з файлу йому необхідно натиснути на кнопку «Import» після чого вибрати файл з перекладом та

натиснути на кнопку «Import Translation»;

- Вихід: переклади додаються до таблиці, та відобразяться на сайті.

2. Варіанти використання користувача

2.1. Створити переклад

– Призначення: створення нового перекладу для тексту на вітрині чи адмін-панелі магазину;

- Актор: користувач;

– Вхід: якщо користувач хоче створити переклад йому потрібно натиснути на кнопку «Add Translation» після ввести у форму оригінальний текст для перекладу, переклад, та вибрати «Store View» для якого призначений переклад;

– Вихід: список перекладів буде оновлено, після очищення кешу переклад відобразиться на вітрині магазину.

2.2. Редагування перекладу

– Призначення: якщо адміністратор хоче змінити переклад чи «Store View»;

- Актор: користувач;

– Вхід: якщо користувач хоче змінити інформацію про переклад, він повинен вибрати рядок з потрібним записом у таблиці, та натиснути кнопку «Edit», або ж змінити переклад за допомогою «Inline Edit», внести зміни та натиснути кнопку «Save» або «Save and Continue»;

– Вихід: список перекладів буде оновлено, після очищення кешу переклад відобразиться на вітрині магазину.

2.3. Видалення перекладу

- Призначення: якщо адміністратор хоче видалити переклад;

- Актор: користувач;

– Вхід: якщо адміністратор хоче видалити переклад, він повинен вибрати в таблиці один або декілька записів та натиснути на кнопку «Delete Translation», після чого підтвердити дію;

- Вихід: переклад буде видалений, після очищення кешу переклад зникне

з сайту.

2.4. Експорт даних

– Призначення: якщо адміністратор хоче експортувати дані до іншого магазину;

– Актор: користувач;

– Вхід: якщо адміністратор хоче експортувати дані до іншого магазину йому потрібно вибрати переклади з таблиці (в іншому випадку експортнуться усі дані) натиснути на клавішу «Export» та вибрати формати даних csv або excel;

– Вихід: буде сформований та завантажений файл з вибраними даними у відповідному форматі.

2.5. Пошук даних

– Призначення: якщо адміністратор хоче знайти переклад;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче знайти переклад він повинен ввести текст у полі пошуку та натиснути на іконку пошуку

– Вихід: список перекладів буде оновлено й відображено в таблиці відповідно до критеріїв пошуку.

2.6. Імпорт даних

– Призначення: якщо адміністратор хоче додати нові переклади з іншого сайту;

– Актор: адміністратор;

– Вхід: якщо адміністратор хоче додати нові переклади з файлу йому необхідно натиснути на кнопку «Import» після чого вибрати файл з перекладом та натиснути на кнопку «Import Translation»;

– Вихід: переклади додаються до таблиці, та відобразяться на сайті.

Діаграми варіантів використання зазвичай називають діаграмами поведінки, що використовуються для опису набору дій (випадків використання), які деякі системи чи певна системи (суб'єкт) повинні або можуть виконувати у співпраці з

одним або кількома зовнішніми користувачами системи (акторами). Кожен випадок використання повинен дати певний спостережуваний та цінний результат для суб'єктів або інших зацікавлених сторін системи.

Специфікації UML 2.0 до 2.4 також описують діаграму використання як спеціалізацію діаграми класів, а діаграма класів - це структурна діаграма.

Діаграми варіантів використання насправді схожі - вони обидві є діаграмами поведінки, оскільки вони описують поведінку системи, а також вони є структурними діаграмами - як особливий випадок діаграми класів, де класифікатори обмежуються різними акторами, або варіантами використання, які пов'язані асоціаціями.

Усі дані про акторів і варіанти використання формують таку діаграму варіантів використання (Рисунок 1.4.2), з якої ми бачимо який функціонал необхідно реалізувати в системі.

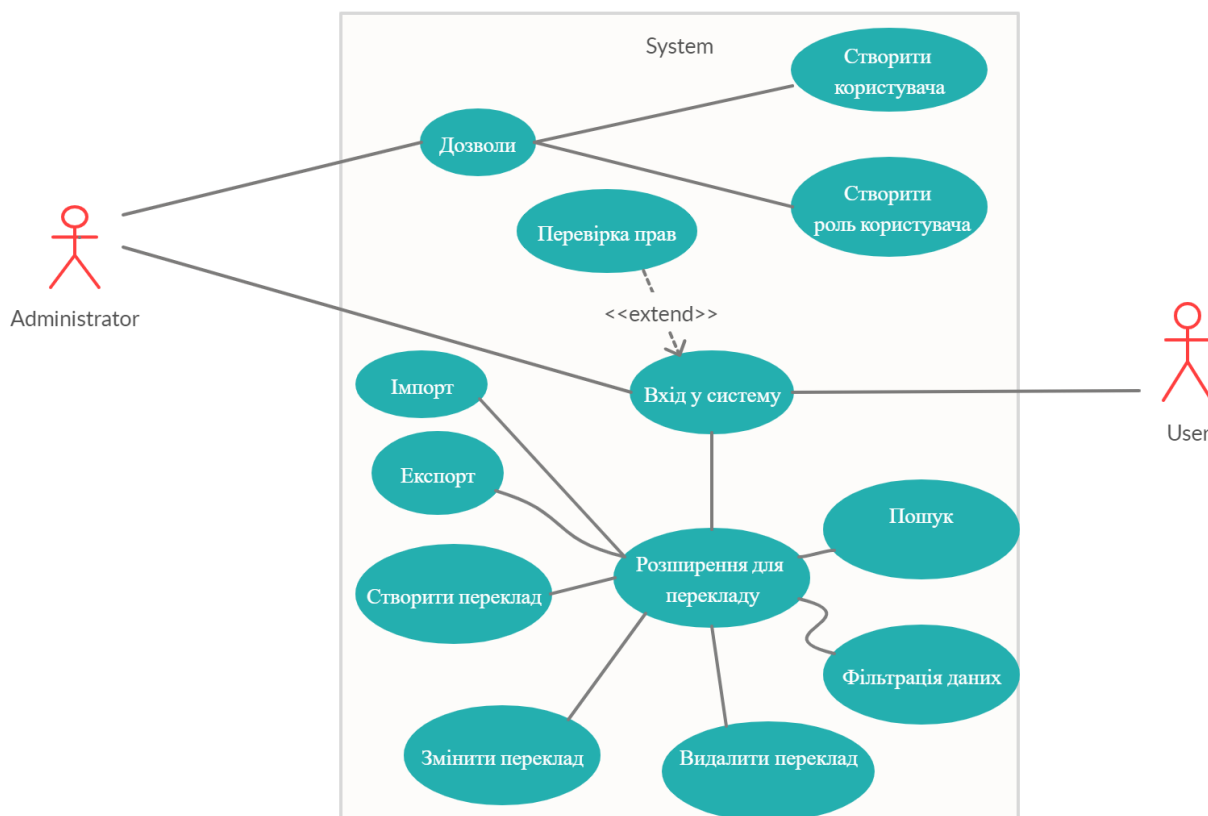


Рисунок 1.4.2 – Діаграма варіантів використання

2. ПРОЕКТУВАННЯ ТА ОПИС АРХІТЕКТУРИ

2.1 Концептуальна архітектура системи

Magento 2 є модульною системою та складається з багатьох компонентів, для прикладу: розширення, теми, бібліотеки, словники для перекладу тексту. Magento_Framework надає розробникам готовий набір інструментів для роботи з продуктами, категоріями, користувачами та замовленнями. Вам не потрібно реалізовувати власне підключення до БД чи придумувати метод кешування, більшість функцій є одразу з коробки. Для розширення логіки та функціоналу потрібно використовувати наслідування та перевизначити уже готові компоненти.

Розширення та теми Magento 2 використовуються для кастомізації базового вигляду та логіки: в основному розширення відповідають за бізнес-логіку, а теми надають функції для створення зовнішнього вигляду сайту.

Кожен модуль являє собою групу файлів, що містить Блоки, Допоміжні класи, Контролери, Моделі, вони є незалежними та не впливають на існуючий функціонал, якщо це не передбачено в самому розширенні.

Кожне розширення являє собою незалежну одиницю, яка в будь-який момент може бути вимкнена чи увімкнена за умови незалежності від інших модулів. Залежність від інших модулів прописується у файлі оголошення модуля, в секції оголошення залежностей, для цього потрібно вказати назву розширення у форматі VendorName_ModuleName. Додатково можна вказати порядок для залежностей. Таким чином архітектура системи все більше стає схожою на мікро сервіси, які можуть бути абсолютно незалежними.

Модулі в Magento 2 реалізовані за допомогою архітектури MVVM зображеної на рисунку 2.1.1.

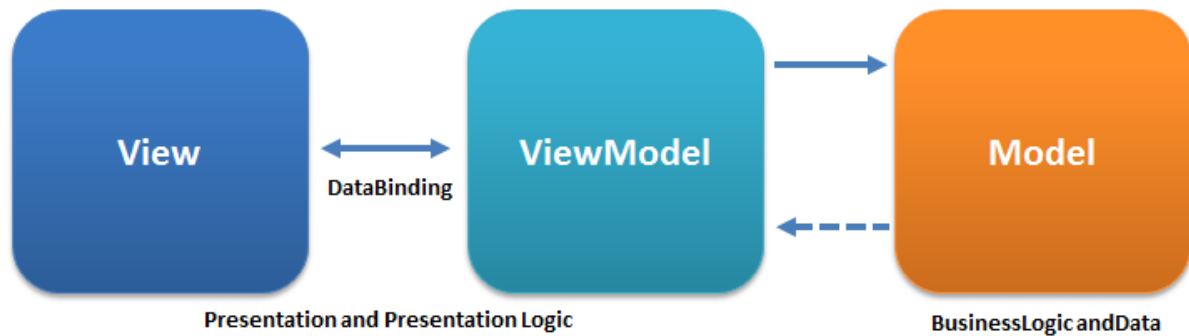


Рисунок 2.1.1 – архітектура MVVM

Model-View-ViewModel (MVVM) – це структурний шаблон проектування, який розділяє об'єкти на три окремі групи:

- Моделі містять дані програми. Зазвичай це структурні або прості класи.
- Вигляд (View) відображають візуальні елементи та елементи керування на екрані. Зазвичай вони є підкласами UIView.
- Модель вигляду (ViewModel) – перетворюють інформацію про модель у значення, які можуть відобразитися у поданні. Зазвичай це класи, тому їх можна передавати як посилання.

ViewModel призначена для:

- Комунікації між моделлю та вікном;
- Нагляду за даними, які змінює користувач.

MVVM призначена для відокремлення бізнес логіки від графічного інтерфейсу.

На перший погляд, MVVM здається дуже схожим на шаблон Model-View-Presenter, оскільки обидва вони чудово справляються із абстракцією стану та поведінки виду. Презентаційна модель абстрагує подання, незалежно від конкретної платформи користувальницького інтерфейсу, тоді як шаблон MVVM

був створений для спрощення програмування користувацьких інтерфейсів, керованого подіями.

Якщо шаблон MVP означав, що доповідач безпосередньо повідомляв представлення, що відображати, у MVVM ViewModel виставляє потоки подій, до яких подання можуть прив'язуватися. Так, ViewModel більше не потрібно містити посилання на View, як і Presenter. Це також означає, що всі інтерфейси, які вимагає шаблон MVP, тепер не потрібні.

Представлення також повідомляють ViewModel про різні дії. Таким чином, шаблон MVVM підтримує двосторонню прив'язку даних між View і ViewModel, і між View і ViewModel існує взаємозв'язок «багато до одного». View має посилання на ViewModel, але ViewModel не має інформації про View. Споживач даних повинен знати про виробника, але виробник - ViewModel - не знає і не хвилюється про те, хто споживає дані.

Власне архітектура Magento 2 зображена на рисунку 2.1.2.

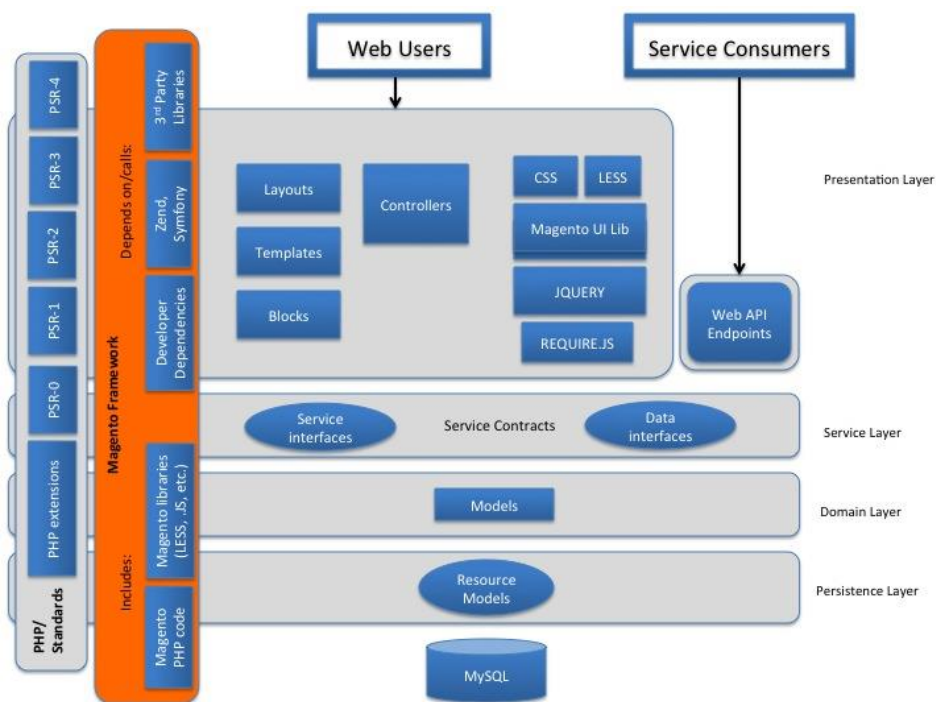


Рисунок 2.1.2 – Архітектура Magento 2

Згідно з офіційною документацією [2], архітектура Magento 2 розбита на 4 шари:

- Presentation Layer – це верхній шар який містить всі View елементи (layouts, blocks, templates, css, js) і контролери. Presentation Layer зазвичай викликає сервісний рівень використовуючи сервіс контракти, але залежачи від імплементації він може перетинатись з бізнес-логікою;

- Service layer – це прошарок між шаром презентації та доменним шаром. Він реалізує контракти на обслуговування, які визначаються за допомогою інтерфейсів PHP. Контракти на обслуговування дозволяють додавати або змінювати модель ресурсів бізнес логіки, використовуючи файл введення залежності (di.xml). Службовий рівень також використовується для надання доступу до API (REST / SOAP або інших модулів);

- Domain layer – відповідає за бізнес логіку, яка не містить інформації, що стосується ресурсів або бази даних. Доменний рівень також може включати договори на обслуговування. Кожна модель даних на рівні доменного шару залежить від ресурс-моделі, яка відповідає за доступ до бази даних;

- Persistence layer – описує ресурс-модель, яка відповідає за діставання та зміну даних у базі даних за допомогою CRUD-запитів. Тут також реалізовані додаткові можливості бізнес логіки, наприклад, валідація даних та реалізація функцій бази даних.

Так як розробка на Magento 2 ведеться шляхом розширення функціоналу за рахунок модулів в нашому розширенні ми будемо використовувати саме MVVM архітектуру. а не створюємо архітектуру з нуля, головним завданням є побудувати розширення відповідно до архітектурних вимог Magento 2

2.2 Взаємодія між компонентами

В нашому випадку розширення використовуватиме REST API, яке надає наступні методи для комунікації з модулем:

```

public function getById($id);

/**
 * @param Data\TranslationInterface $translation
 * @return mixed
 */
public function save(\Magefan\Translation\Api\Data\TranslationInterface
$translation);

public function
delete(\Magefan\Translation\Api\Data\TranslationInterface $translation);

public function deleteById($id);

public function get($id);

public function create($data);

public function update($id, $data);

```

За допомогою методів наведених вище можна виконати всі доступні операції з перекладом через API.

2.3 Специфікації для основних класів

Діаграма класу UML - це графічне позначення, що використовується для побудови та візуалізації об'єктно-орієнтованих систем. Діаграма класів в Уніфікованій мові моделювання (UML) - це тип статичної структурної діаграми, що описує структуру системи, допомогою:

- класів;
- атрибутів;
- операцій;
- взаємозв'язків між об'єктами.

Клас – це проект об'єкта. Предмети та класи йдуть рука об руку. Ми не можемо говорити про одне, не говорячи про інше. І вся суть об'єктно-орієнтованого проектування стосується не об'єктів, а класів, оскільки ми

використовуємо класи для створення об'єктів. Отже, клас описує, яким буде об'єкт, але це не сам об'єкт.

Насправді класи описують тип об'єктів, тоді як об'єкти є придатними екземплярами класів. Кожен об'єкт побудований з одного набору креслень і, отже, містить однакові компоненти (властивості та методи). Стандартне значення полягає в тому, що об'єкт є екземпляром класу та об'єкта - об'єкти мають стани та поведінку.

Діаграма класів розширення для перекладу тексту зображена на рисунку 2.3.1.

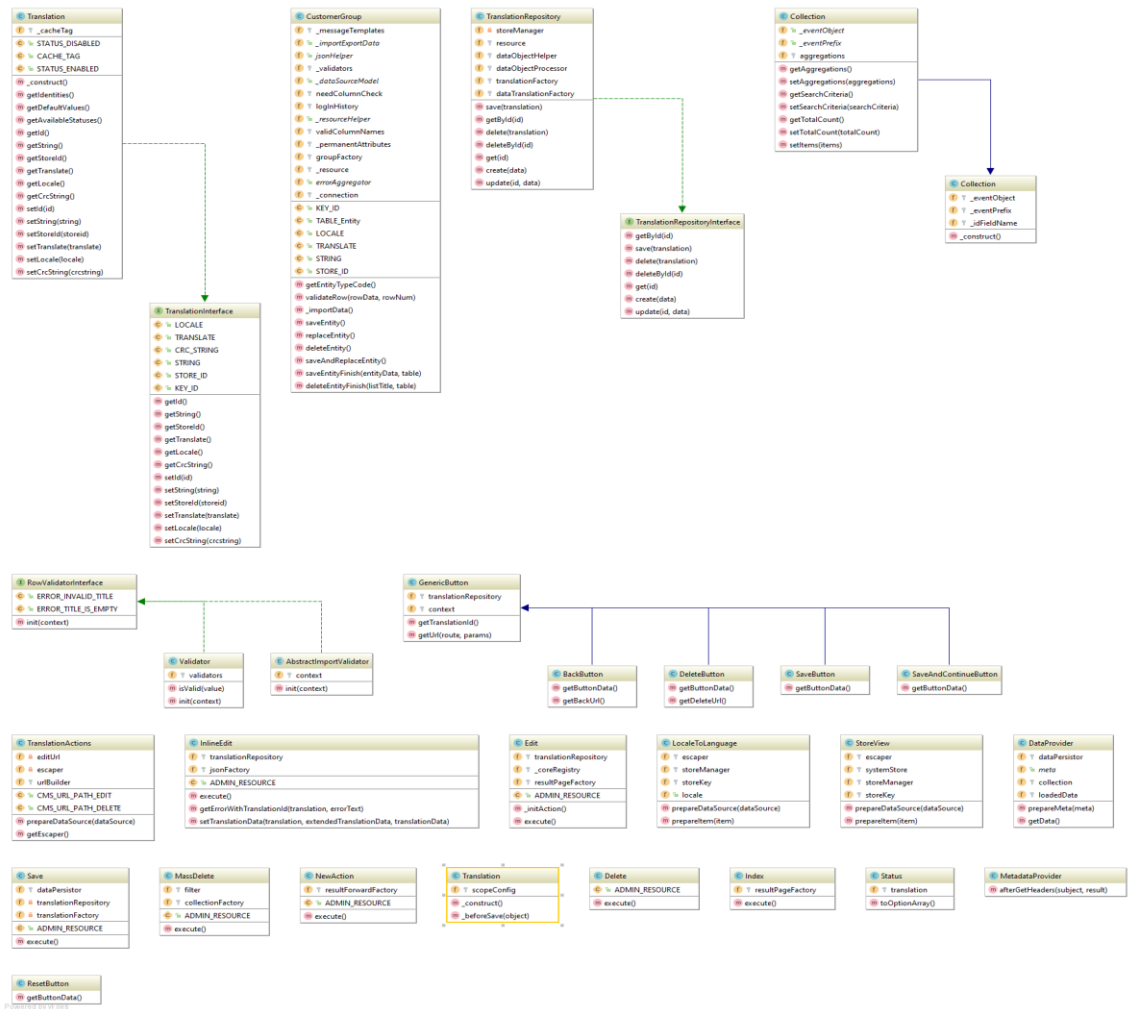


Рисунок 2.3.1 – Діаграма класів

Для того, щоб відобразити обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах використаємо діаграму розгортання проекту зображену на рисунку 2.3.2

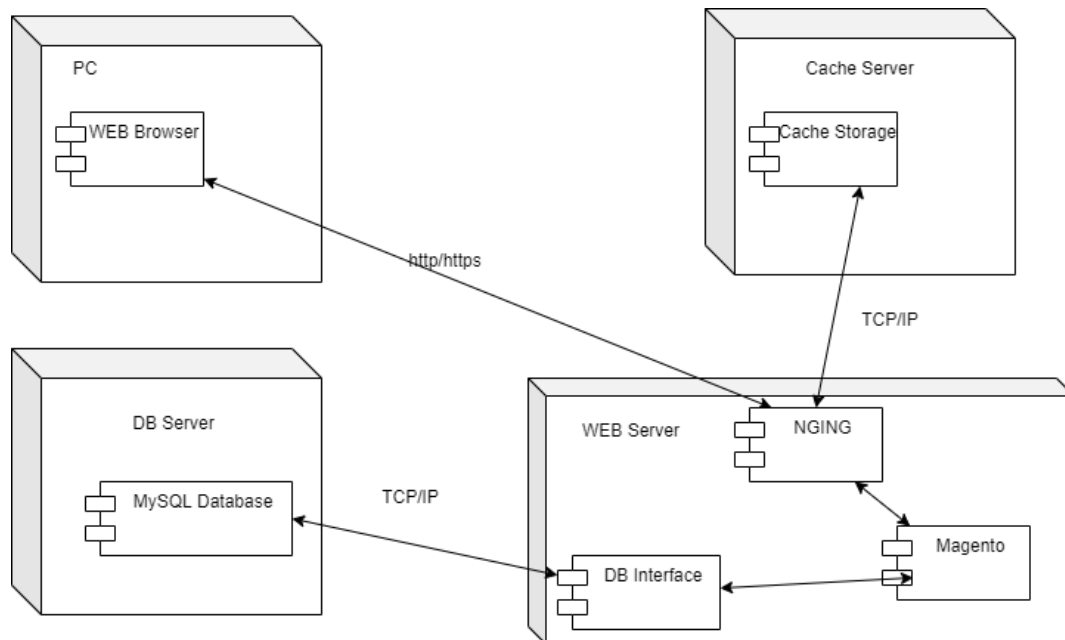


Рисунок 2.3.2 – Діаграма розгортання

3 РОЗРОБКА, РОЗГОРТАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Створення базової структури

Для того, щоб створити базову структуру модуля (див. рисунок 3.1.1) Magento 2 скористаємось сервісом Mage2Gen – це генератор модулів для Magento 2. Завдяки зручному інтерфейсу користувач може генерувати базовий код для модуля Magento 2, починаючи від базової структури необхідної для реєстрації модуля в системі та закінчуючи платіжними методами чи методами доставки.

Generated files (click a file to view its contents. Live reloads)

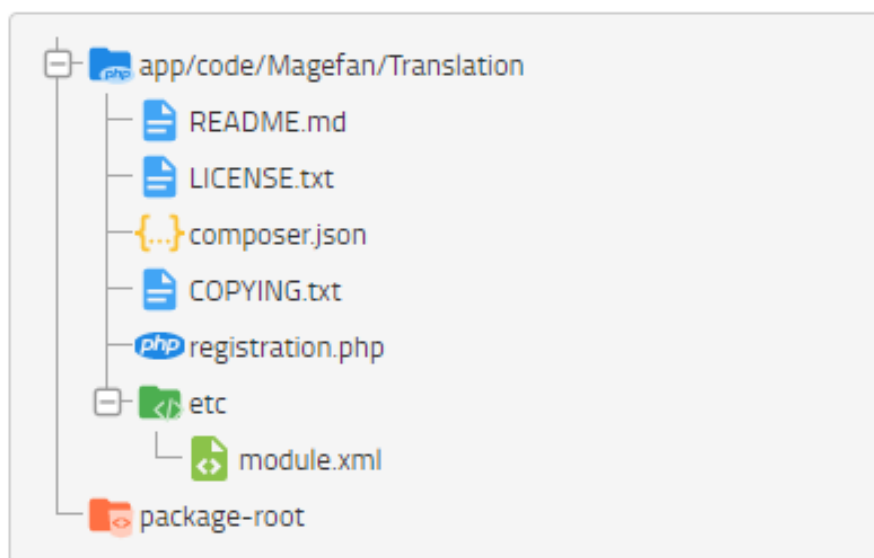


Рисунок 3.1.1 – Базова структура модуля Magento 2

Тепер нам потрібно створити файл конфігурації меню з іменем menu.xml за шляхом /app/code/Magefan/Translation/etc /adminhtml та додати у нього наступний запис:

```
<menu>
  <add id="Magefan_Translation::translation_translation" title="Translation by Magefan" translate="title" module="Magefan_Translation" parent="Magento_Backend::system_tools" sortOrder="10" dependsOnModule="Magefan_Translation" action="translation/translation">
```

```
resource="Magefan_Translation::translation_translation"/>
</menu>
```

Після цього в адмін-панелі Magento з'явиться пункт меню «Translation by Magefan».

Далі нам необхідно описати інтерфейс та репозиторію. Реалізація інтерфейсу:

```
interface TranslationInterface
{
    const KEY_ID = 'key_id';
    const STRING = 'string';
    const STORE_ID = 'store_id';
    const TRANSLATE = 'translate';
    const LOCALE = 'locale';
    const CRC_STRING = 'crc_string';

    public function getId();
    public function getString();
    public function getStoreId();
    public function getTranslate();
    public function getLocale();
    public function getCrcString();
    public function setId($id);
    public function setString($string);
    public function setStoreId($storeid);
    public function setTranslate($translate);
    public function setLocale($locale);
    public function setCrcString($crcstring);
}
```

Після чого можна припустити до реалізації репозиторії. Особливу увагу потрібно приділити коментарям вони повинні чітко описувати вхідні та вихідні параметри, в іншому випадку API не працюватиме. Реалізації репозиторії:

```
interface TranslationRepositoryInterface
{
    public function getById($id);

    /**
     * @param Data\TranslationInterface $translation
     * @return mixed
     */
    public function save(\Magefan\Translation\Api\Data\TranslationInterface
    $translation);

    public function delete(\Magefan\Translation\Api\Data\TranslationInterface
    $translation);

    public function deleteById($id);
```

```

public function get($id);

public function create($data);

public function update($id, $data);
}

```

Для отримання даних з бази даних потрібно створити ресурс-модель та вказати потрібну для підключення таблицю.

```

class Translation extends \Magento\Framework\Model\ResourceModel\Db\AbstractDb
{
    protected $scopeConfig;

    public function __construct(
        \Magento\Framework\Model\ResourceModel\Db\Context $context,
        \Magento\Framework\App\Config\ScopeConfigInterface $scopeConfig
    ) {
        parent::__construct($context);
        $this->scopeConfig = $scopeConfig;
    }

    /**
     * Define main table
     */
    protected function _construct()
    {
        $this->_init('translation', 'key_id');
    }

    protected function _beforeSave(\Magento\Framework\Model\AbstractModel $object)
    {
        if ($object->getData('mf_locale')) {
            $object->setData('locale', $object->getData('mf_locale'));
        }

        $storeId = $object->getData('store_id') ?: 0;
        $crcString = $object->getData('string') ?: '';

        if ($storeId > 0) {
            $object->setData('locale', $this->scopeConfig-
>getValue('general/locale/code', ScopeInterface::SCOPE_STORE, $storeId));
        }
        $object->setData('crc_string', crc32($crcString));
        return parent::_beforeSave($object);
    }
}

```

Тепер коли ми маємо інтерфейс, який говорить нам як ми повинні використовувати данні і власне самі данні з ресурс-моделі, ми можемо створити модель Translation яка імплементуватиме описаний вище інтерфейс.

```

class Translation extends AbstractModel implements TranslationInterface,

```

IdentityInterface

```

{
    const STATUS_ENABLED = 1;
    const STATUS_DISABLED = 0;
    const CACHE_TAG = 'magefan_translation';
    protected $_cacheTag = self::CACHE_TAG;
    protected function _construct()
    {
        $this->_init(\Magefan\Translation\Model\ResourceModel\Translation::class);
    }

    public function getIdentities()
    {
        return [self::CACHE_TAG . '_' . $this->getId()];
    }

    public function getDefaultValues()
    {
        $values = [];
        return $values;
    }

    public function getAvailableStatuses()
    {
        return [self::STATUS_ENABLED => __('Enabled'), self::STATUS_DISABLED =>
        __('Disabled')];
    }

    public function getId()
    {
        return parent::getData(self::KEY_ID);
    }

    public function getString()
    {
        return $this->getData(self::STRING);
    }

    public function getStoreId()
    {
        return $this->getData(self::STORE_ID);
    }

    public function getTranslate()
    {
        return $this->getData(self::TRANSLATE);
    }

    public function getLocale()
    {
        return $this->getData(self::LOCALE);
    }

    public function getCrcString()
    {
        return $this->getData(self::CRC_STRING);
    }

    public function setId($id)
    {
        return $this->setData(self::KEY_ID, $id);
    }
}

```

```

public function setString($string)
{
    return $this->setData(self::STRING, $string);
}

public function setStoreId($storeid)
{
    return $this->setData(self::STORE_ID, $storeid);
}

public function setTranslate($translate)
{
    return $this->setData(self::TRANSLATE, $translate);
}

public function setLocale($locale)
{
    return $this->setData(self::LOCALE, $locale);
}
public function setCrcString($crcstring)
{
    return $this->setData(self::CRC_STRING, $crcstring);
}
}

```

Після чого створимо модель репозиторії та імплементуємо відповідний інтерфейс:

```

class TranslationRepository implements TranslationRepositoryInterface
{
    protected $resource;
    protected $translationFactory;
    protected $dataObjectHelper;
    protected $dataObjectProcessor;
    protected $dataTranslationFactory;
    private $storeManager;

    public function __construct(
        ResourceTranslation $resource,
        TranslationFactory $translationFactory,
        Data\TranslationInterfaceFactory $dataTranslationFactory,
        DataObjectHelper $dataObjectHelper,
        DataObjectProcessor $dataObjectProcessor,
        StoreManagerInterface $storeManager
    ) {
        $this->resource = $resource;
        $this->translationFactory = $translationFactory;
        $this->dataObjectHelper = $dataObjectHelper;
        $this->dataTranslationFactory = $dataTranslationFactory;
        $this->dataObjectProcessor = $dataObjectProcessor;
        $this->storeManager = $storeManager;
    }

    public function save(\Magefan\Translation\Api\Data\TranslationInterface
    $translation)
    {
        if ($translation->getStoreId() === null) {

```

```

        $storeId = $this->storeManager->getStore()->getId();
        $translation->setStoreId($storeId);
    }
    try {
        $this->resource->save($translation);
    } catch (\Exception $exception) {
        throw new CouldNotSaveException(
            __('Could not save the translation: %1', $exception-
>getMessage()),
                $exception
            );
    }
    return $translation;
}

public function getById($id)
{
    $translation = $this->translationFactory->create();
    $translation->load($id);
    if (!$translation->getId()) {
        throw new NoSuchEntityException(__('Translation with id "%1" does not
exist.', $id));
    }
    return $translation;
}

public function delete(\Magefan\Translation\Api\Data\TranslationInterface
$translation)
{
    try {
        $this->resource->delete($translation);
    } catch (\Exception $exception) {
        throw new CouldNotDeleteException(__('
Could not delete the translation: %1',
            $exception->getMessage()
        ));
    }
    return true;
}

public function deleteById($id)
{
    return $this->delete($this->getById($id));
}

public function get($id)
{
    $translation = $this->translationFactory->create();
    $translation->load($id);
    if (!$translation->getId()) {
        throw new NoSuchEntityException(__('Translation with id "%1" does not
exist.', $id));
    }
    return \GuzzleHttp\json_encode($translation->getData());
}

public function create($data)
{
    try {
        $data = json_decode($data, true);
        $item = $this->translationFactory->create();
        $item->setData($data)->save();
    }
}

```



```

        return json_encode($item->getData());
    } catch (\Exception $e) {
        return false;
    }
}

public function update($id, $data)
{
    try {
        $item = $this->translationFactory->create();
        $item->load($id);

        if (!$item->getId()) {
            return false;
        }
        $data = json_decode($data, true);
        $item->addData($data)->save();
        return json_encode($item->getData());
    } catch (\Exception $e) {
        return false;
    }
}
}

```

Щоб все описане вище працювало нам потрібно зв'язати інтерфейси з моделями за допомогою файлу /etc/di.xml (ін'єкція залежностей).

```

<?xml version="1.0" ?>

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd"
>
    <type name="Magento\Ui\Model\Export\MetadataProvider">
        <plugin disabled="false"
name="Magefan_Translation_Plugin_Magento_Ui_Model_Export_MetadataProvider"
sortOrder="10"
type="Magefan\Translation\Plugin\Magento\Ui\Model\Export\MetadataProvider"/>
    </type>
    <preference for="Magefan\Translation\Api\Data\TranslationInterface"
type="Magefan\Translation\Model\Translation" />
    <preference for="Magefan\Translation\Api\TranslationRepositoryInterface"
type="Magefan\Translation\Model\TranslationRepository" />
    <virtualType name="TranslationGridDataProvider"
type="Magento\Framework\View\Element\UiComponent\DataProvider\DataProvider">
        <arguments>
            <argument name="collection" xsi:type="object"
shared="false">Magefan\Translation\Model\ResourceModel\Translation\Collection</arg
ument>
        </arguments>
    </virtualType>
    <type
name="Magento\Framework\View\Element\UiComponent\DataProvider\CollectionFactory">
        <arguments>
            <argument name="collections" xsi:type="array">
                <item name="translation_translation_listing_data_source"
xsi:type="string">Magefan\Translation\Model\ResourceModel\Translation\Grid\Collect
ion</item>
            </argument>
        </arguments>
    </type>

```

```

</type>
<type
name="Magefan\Translation\Model\ResourceModel\Translation\Grid\Collection">
  <arguments>
    <argument name="mainTable" xsi:type="string">translation</argument>
    <argument name="eventPrefix"
xsi:type="string">translation grid collection</argument>
    <argument name="eventObject"
xsi:type="string">translation collection</argument>
    <argument name="resourceModel"
xsi:type="string">Magefan\Magefan\Model\ResourceModel\Translation</argument>
  </arguments>
</type>
<type name="Magento\Framework\Model\Entity\RepositoryFactory">
  <arguments>
    <argument name="entities" xsi:type="array">
      <item name="Magefan\Translation\Api\Data\TranslationInterface"
xsi:type="string">Magefan\Translation\Api\TranslationRepositoryInterface</item>
    </argument>
  </arguments>
</type>
</config>

```

Після описаних вище етапів ми реалізували базові функції CRUD для роботи з базою даних.

Тепер нам потрібно створити контролери для роботи з формою (рис 3.1.2).

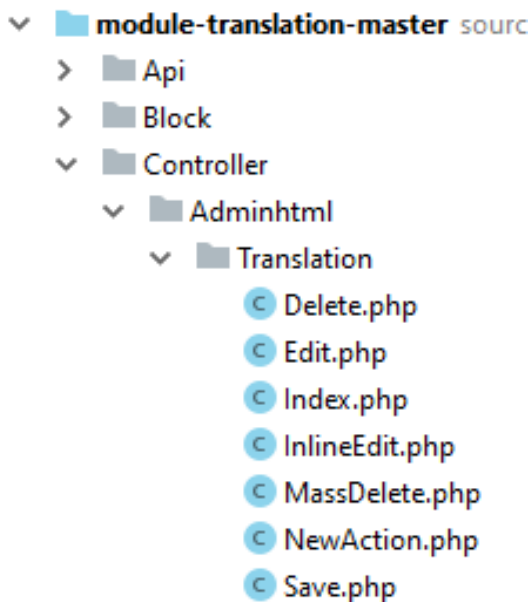


Рисунок 3.1.2 – Перелік контролерів

Для забезпечення безпеки потрібно створити файл у директорії etc під назвою acl.xml (Access Control List), та додати власні ресурси до дерева ресурсів, після

чого доступ до розширення може бути обмежений в налаштуваннях ролей користувачів.

```
<?xml version="1.0" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Acl/etc/acl.xsd">
  <acl>
    <resources>
      <resource id="Magento_Backend::admin">
        <resource id="Magefan_Translation::magefan" title="Magefan"
sortOrder="51">
          <resource id="Magefan_Translation::translation_translation"
title="Translation" sortOrder="10">
            <resource id="Magefan_Translation::save" title="Save
Translation" translate="title" sortOrder="10" />
            <resource id="Magefan_Translation::translation_delete"
title="Delete Translation" translate="title" sortOrder="20" />
          </resource>
        </resource>
      </resource>
    </resources>
  </acl>
</config>
```

На цьому етапі ми можемо використовувати дані контролери для реалізації CRUD методів в адмін-панелі. Для того, щоб створити інтерфейс користувача в адмін-панелі Magento використовує UI Components за допомогою яких ми можемо реалізувати:

- Створення;
- Видалення;
- Редагування;
- Сортування;
- Пошук;
- Фільтрацію;
- Експорт;
- Імпорт.

Після реалізації UI Components ми можемо переходити до етапу тестування.

В результаті конструювання ми отримаємо структуру проекту зображену на рисунку 3.1.3



Рисунок 3.1.3 – Структура проекту

3.2 Аутентифікація на основі токенів.

Щоб здійснити звернення через веб-АРІ від клієнта, наприклад мобільного додатка, потрібно надати маркер доступу під час виклику. Токен діє як електронний ключ, що дозволяє отримати доступ до АРІ.

Magento видає такі типи токенів доступу:

1. Integration - Продавець визначає, до яких ресурсів Magento буде доступ для інтеграції. Він триває доти, поки його не буде скасовано вручну;
2. Admin – Продавець визначає рівень доступу користувача. За замовчування токен буде дійсний протягом чотирьох годин;
3. Customer – Magento надає доступ до ресурсів з анонімним методом доступу. Продавці не можуть редагувати ці налаштування. Таке підключення дійсне протягом однієї години.

Коли продавець створює та активує інтеграцію Magento генерує consumer key, consumer secret, access token, та access token secret. Всі ці ключі використовуються для OAuth-based аунтифікації, проте аунтифікація за токеном вимагає лише access token.

Для створення токена доступу потрібно виконати наступні дії:

1. Увійдіть в панель адміністратора та натисніть Система> Розширення> Інтеграції, щоб відобразити сторінку Інтеграції.
2. Клацніть Додати нову інтеграцію, щоб відобразити сторінку Нова інтеграція.
3. Введіть унікальне ім'я для інтеграції в поле Ім'я. Потім введіть пароль адміністратора в поле Ваш пароль. Всі інші поля залиште порожніми.
4. Клацніть на вкладку API. Виберіть ресурси Magento, до яких може отримати доступ інтеграція. Ви можете вибрати всі ресурси або вибрати власний список.
5. Натисніть Зберегти, щоб зберегти зміни та повернутися на сторінку Інтеграції.
6. Клацніть посилання Активувати в сітці, яке відповідає новоствореній інтеграції.
7. Клацніть Дозволити.

Magento надає окремий токен-сервіс для адміністраторів та клієнтів. Коли ви запитуєте токен у однієї з цих служб, служба повертає унікальний токен доступу в обмін на ім'я користувача та пароль для облікового запису Magento.

Структура веб-API Magento дозволяє гостьовим користувачам отримувати доступ до ресурсів, які дозволені для користувачів з анонімним рівнем дозволу. Гостьові користувачі – це користувачі, які Framework не може провести автентифікацію за допомогою існуючих механізмів автентифікації. Як гостьовому користувачу вам не потрібно використовувати токен, але ви можете вказати його у виклику веб-API для ресурсу з анонімним дозволом. Обмеження доступу до анонімних веб-API містить список API, для яких не потрібен токен.

Облікові записи адміністраторів повинні бути авторизовані через двохфакторну аунтифікацію, тобто вхід повинен бути підтвердженим через ваш смартфон. Деякі оператори можуть вимагати декількох дзвінків.

Наведемо приклад отримання токена за допомогою curl:

```
curl -X POST "https://magento.host/index.php/rest/V1/integration/customer/token" \
  -H "Content-Type:application/json" \
  -d '{"username":"customer@example.com", "password":"customer_password"}'
```

Запит поверне стрічку наступного виду «asdf3hjk1p5iuuytre».

3.3 Тестування класів та методів

Для того щоб перевірити якість нашого коду використаємо Magento Extension Quality Program Coding Standard (Magento EQP), яка дозволяє перевіряти код на відповідність стандарту, а також виявляти такі недоліки:

- Виконання SQL запитів у циклі;
- Використання небезпечних функцій;
- Використання супер-глобальних змінних;

- Надмірна складність коду;
- Невиправдані завантаження колекцій.

Після виконання тестування отримані результати зображені на рисунку 3.3.1

```
FILE: /path/to/your/extension/Model/Object.php
-----
FOUND 0 ERRORS AND 10 WARNINGS AFFECTING 10 LINES
-----
46 | WARNING | Use of protected class members is discouraged.
51 | WARNING | Use of protected class members is discouraged.
56 | WARNING | Use of protected class members is discouraged.
61 | WARNING | Use of protected class members is discouraged.
66 | WARNING | Use of protected class members is discouraged.
71 | WARNING | Use of protected class members is discouraged.
109 | WARNING | Use of protected class members is discouraged.
159 | WARNING | Direct object instantiation (object of
    |         | \Magento\Framework\DataObject) is discouraged in
    |         | Magento 2.
222 | WARNING | The use of function is_null() is discouraged; use
    |         | strict comparison "=== null" instead.
256 | WARNING | Direct throw of Exception is discouraged. Use
    |         | \Magento\Framework\Exception\LocalizedException
    |         | instead.
```

Рисунок 3.3.1 – Результат тестування

Згідно з результатами тестування критичних помилок не виявлено.

3.4 Інтеграційне тестування

Інтеграційні тести вимагають середовища Magento для виконання, тому перед їх виконанням потрібно трохи підготуватися. Після підготовки системи ми можемо виконати тести, використовуючи інтерфейс командного рядка або всередині IDE, наприклад PhpStorm.

Перш ніж використовувати тестовий фреймворк інтеграції Magento, ми повинні підготувати тестове середовище. Передумовами тестового середовища є:

- Виділена база даних для тестової інтеграції;
- Конфігурація бази даних налаштована під тести;

– Конфігурація PHPUnit відповідає меті виконання тесту на інтеграцію.

Після виконання кроків описаних вище можемо запустити Unit Test за допомогою команди:

```
php ./vendor/bin/phpunit -c dev/tests/unit/phpunit.xml.dist
app/code/Magefan/Translation /Test/Unit/Model/Translation.php.
```

Результат виконання Unit тесту зображений на рисунку 3.4.2.

```
..... 4 / 4 (100%)
Time: 406 ms, Memory: 4.00MB
OK (4 tests, 4 assertions)
```

Рисунок 3.4.2 – Результат виконання Unit тесту

3.5 Розгортання програмної системи та системні вимоги

Вимоги до серверної частини:

- ОС Linux;
- не менше ніж 8Гб ОЗП;
- SSD 128 GB.

Вимоги до клієнтської частини:

- iOS/Android/Windows/macOS.

Додаткові вимоги:

- наявне підключення до мережі Інтернет.

Програмні вимоги:

- БД: MySQL 5.6, 5.7.
- Версія PHP: від 7.1.3.
- Встановленні PHP розширення: ext-bcmath, ext-ctype, ext-curl, ext-dom, ext-gd, ext-hash, ext-iconv, ext-intl, ext-mbstring, ext-openssl, ext-pdo_mysql, ext-simplexml, ext-soap, ext-spl, ext-xsl, ext-zip, lib-libxml.
- Веб-сервер: Apache 2/Nginx.

Для того, щоб встановити розширення для перекладу тексту потрібно виконати наступні вимоги:

- Встановлене Magento версії 2.2.6 і вище;
- Composer 1.6 і вище;
- SSH з дозволом на виконання.

Далі виконати наступні команди:

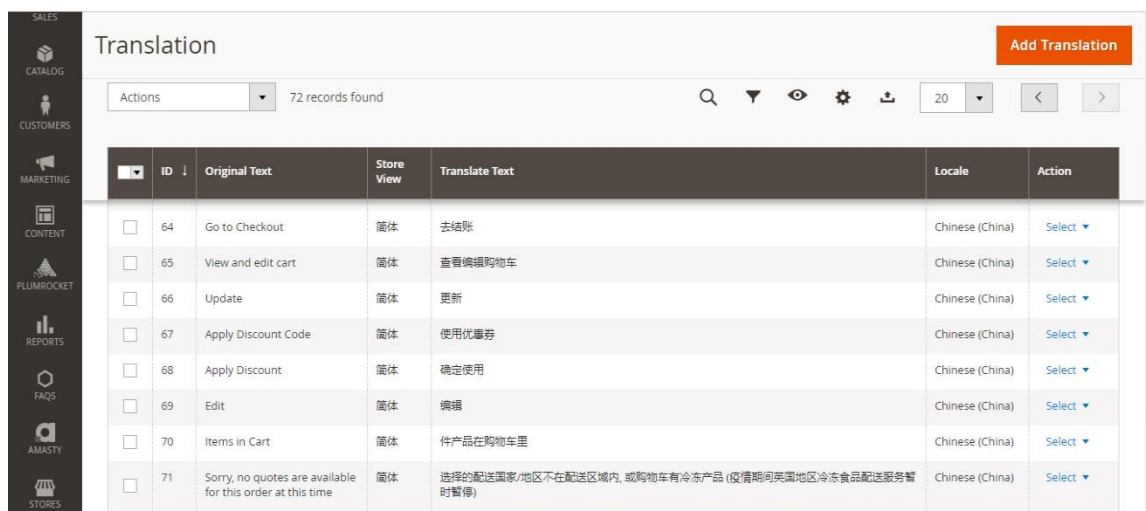
- `composer require magefan/module-translation;`
- `php bin/magento setup:upgrade;`
- `php bin/magento setup:di:compile;`
- `php bin/magento setup:static-content:deploy.`

Після чого розширення стане доступним в адмін-панелі Magento 2.

3.6 Опис типових схем використання системи

Завдяки наступним крокам ми перевіримо працездатність системи, пройдемо по типових схемах використання ПЗ та протестуємо функціонал.

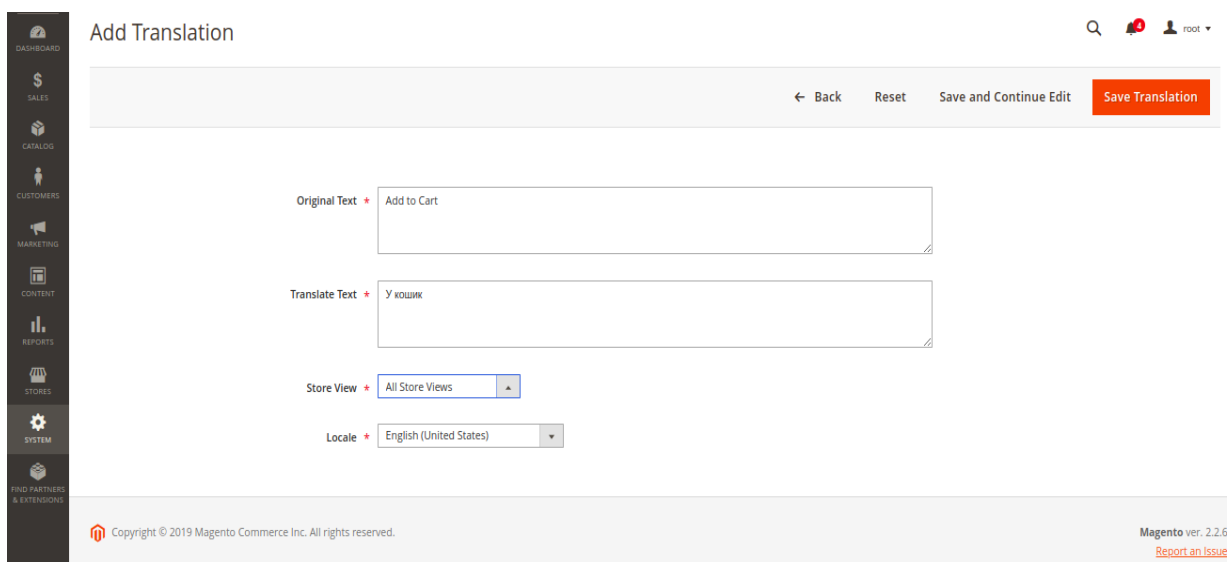
Для того щоб відкрити модуль потрібно зайти у панель-адміністратора, після чого у вкладці System вибрати опцію Translate, після чого відкриється сторінка з таблицею перекладів рисунок 3.6.1.



	ID ↓	Original Text	Store View	Translate Text	Locale	Action
<input type="checkbox"/>	64	Go to Checkout	简体	去结账	Chinese (China)	Select ▼
<input type="checkbox"/>	65	View and edit cart	简体	查看编辑购物车	Chinese (China)	Select ▼
<input type="checkbox"/>	66	Update	简体	更新	Chinese (China)	Select ▼
<input type="checkbox"/>	67	Apply Discount Code	简体	使用优惠券	Chinese (China)	Select ▼
<input type="checkbox"/>	68	Apply Discount	简体	确定使用	Chinese (China)	Select ▼
<input type="checkbox"/>	69	Edit	简体	编辑	Chinese (China)	Select ▼
<input type="checkbox"/>	70	Items in Cart	简体	件产品在购物车里	Chinese (China)	Select ▼
<input type="checkbox"/>	71	Sorry, no quotes are available for this order at this time	简体	选择的配送国家/地区不在配送区域内, 或购物车有冷冻产品 (疫情期间英国地区冷冻食品配送服务暂时暂停)	Chinese (China)	Select ▼

Рисунок 3.6.1 – Таблица перекладів

Для того щоб додати новий переклад потрібно, клікнути на кнопку «Add Translation» після чого відкриється сторінка з формою для додавання запису що зображена на рисунку 3.6.2.



Add Translation

← Back Reset Save and Continue Edit **Save Translation**

Original Text *

Translate Text *

Store View *

Locale *

Copyright © 2019 Magento Commerce Inc. All rights reserved. Magento ver. 2.2.6 [Report an Issue](#)

Рисунок 3.6.2 – Сторінка додавання нового перекладу

Для прикладу роботи модуля змінимо текст кнопки «Add to Cart» на «У кошик» заповнивши поля «Original Text» та «Translate Text» відповідно. Після чого потрібно обрати «Store View» – магазин, та «Locale» - мову магазину для якої створитися переклад. Вигляд кнопок після перекладу зображено на рисунку 3.6.3.

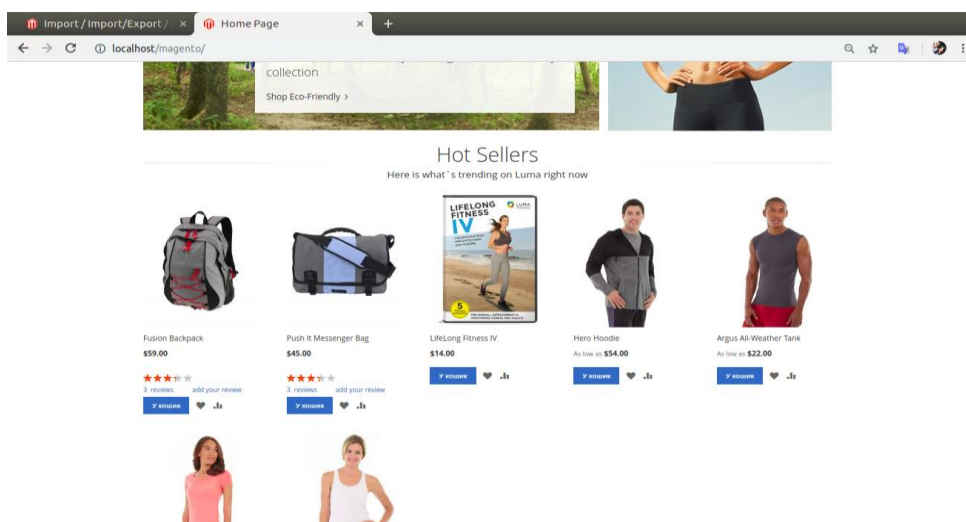


Рисунок 3.6.3 – Застосування перекладу

Для того, щоб змінити існуючий переклад в селекті Action вибираємо кнопку Edit, після чого відкриється форма редагування перекладу що зображена на рисунку 3.6.4.

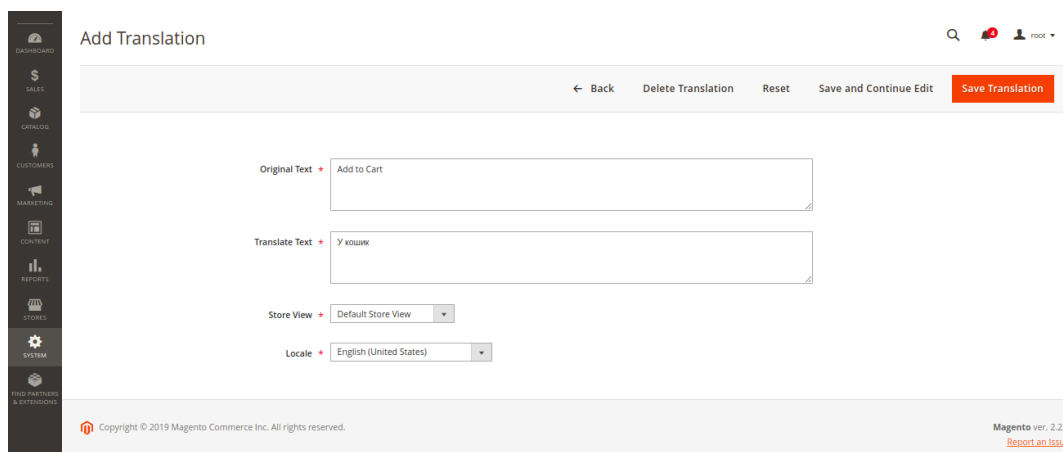


Рисунок 3.6.4 – Форма редагування перекладу

З форми редагування ми також можемо видалити поточний переклад натиснувши кнопку «Delete Translation». Якщо користувач під час редагування тексту схоче відмінити свої зміни він може натиснути на кнопку «Reset» яка поверне значення до редагування.

З допомогою чекбоксів ми можемо видалити одразу ж декілька записів, спочатку позначивши записи які потрібно видалити після чого вибрати «Actions» та «Delete». Приклад видалення записів зображено на рисунку 3.6.5.

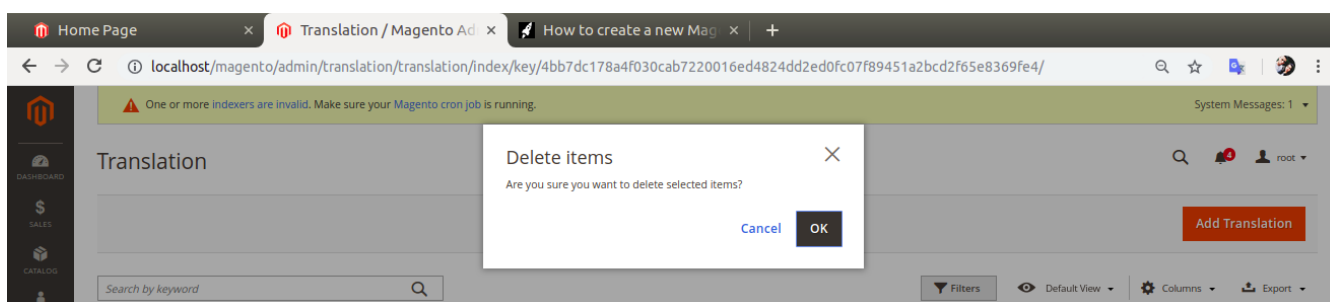


Рисунок 3.6.5 – Видалення записів

Щоб знайти переклад можна скористатись наступними фільтрами зображеними на рисунку 3.6.6.

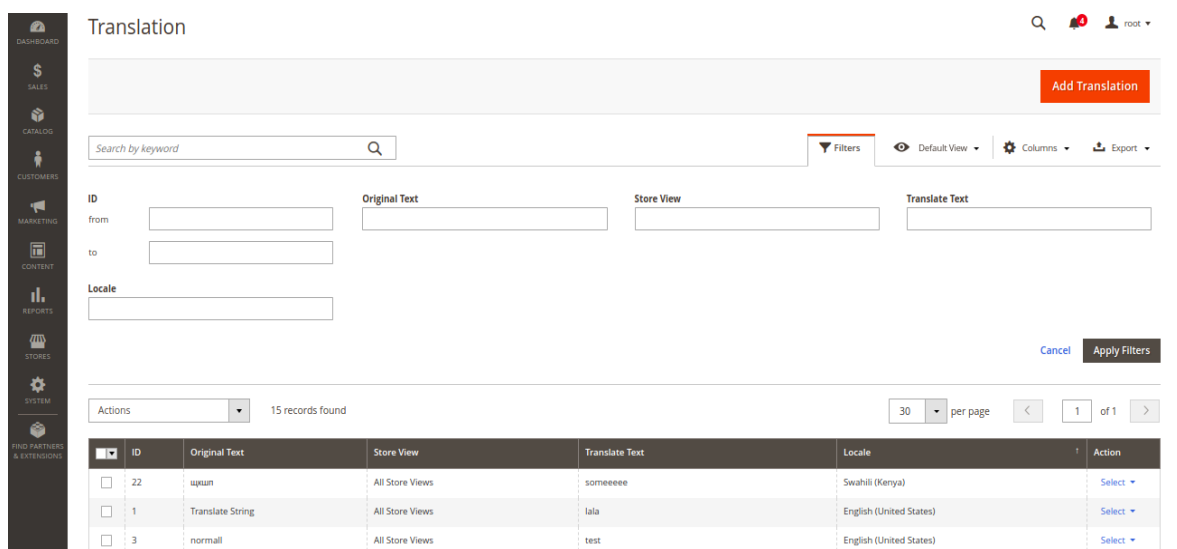


Рисунок 3.6.6 – Пошук

Результат пошуку перекладу по ID 1 зображено на рисунку 3.6.7.

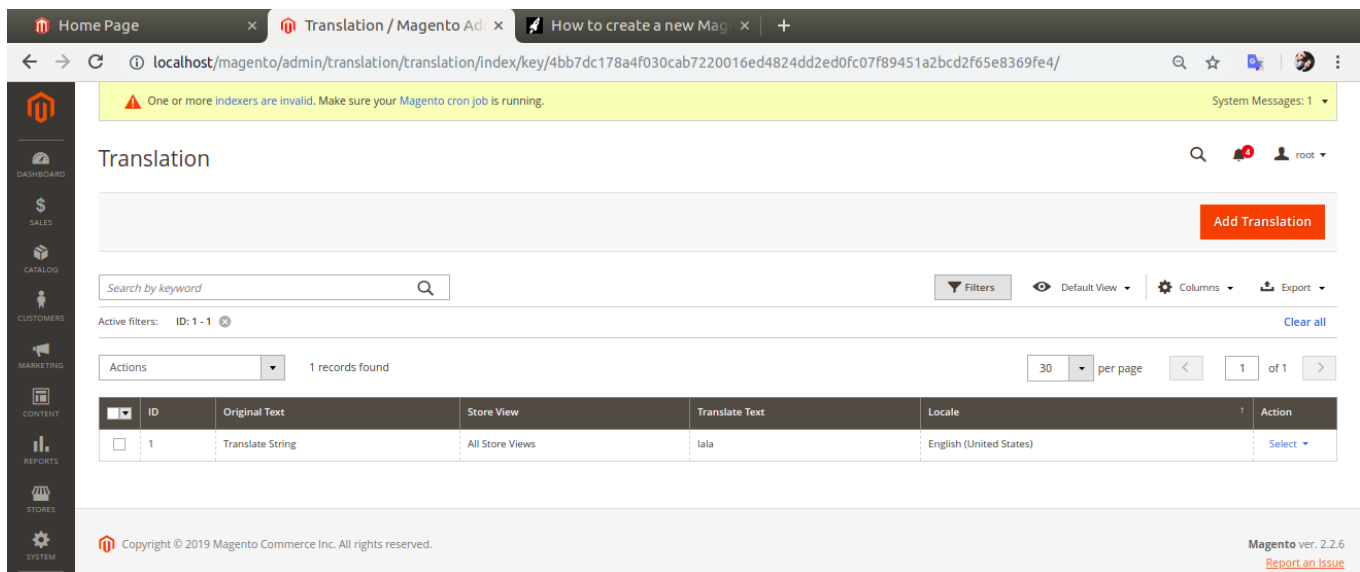


Рисунок 3.6.7 – Пошук по ID

У пункті «Columns» ми можемо вибрати які поля таблиці повинні відображатися у таблиці

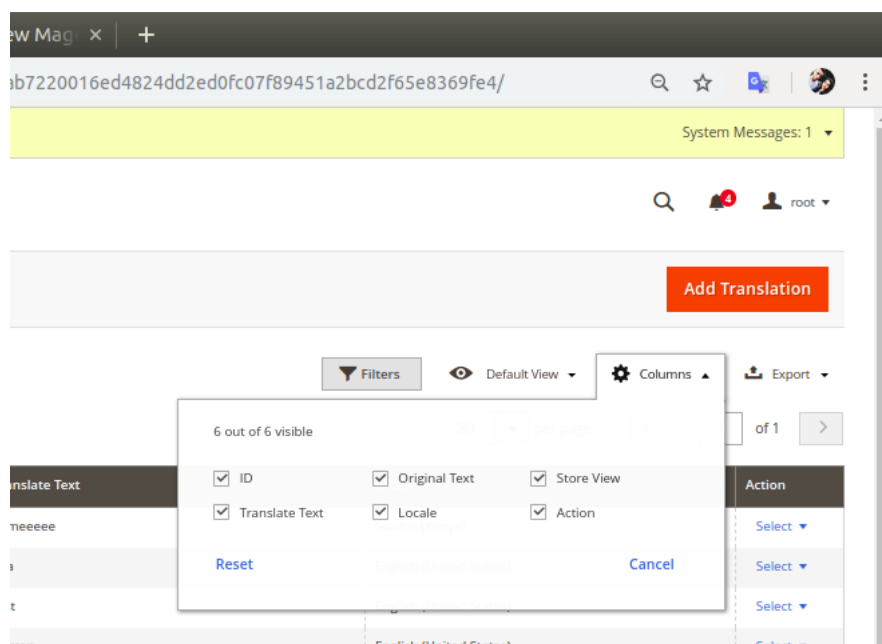


Рисунок 3.6.8 – Налаштування полів таблиці

У списку «Per page» ми можемо обрати кількість записів що відобразяться у таблиці за раз, або вказати свої значення. Для того щоб відсортувати записи потрібно клікнути на потрібне поле у заголовку таблиці

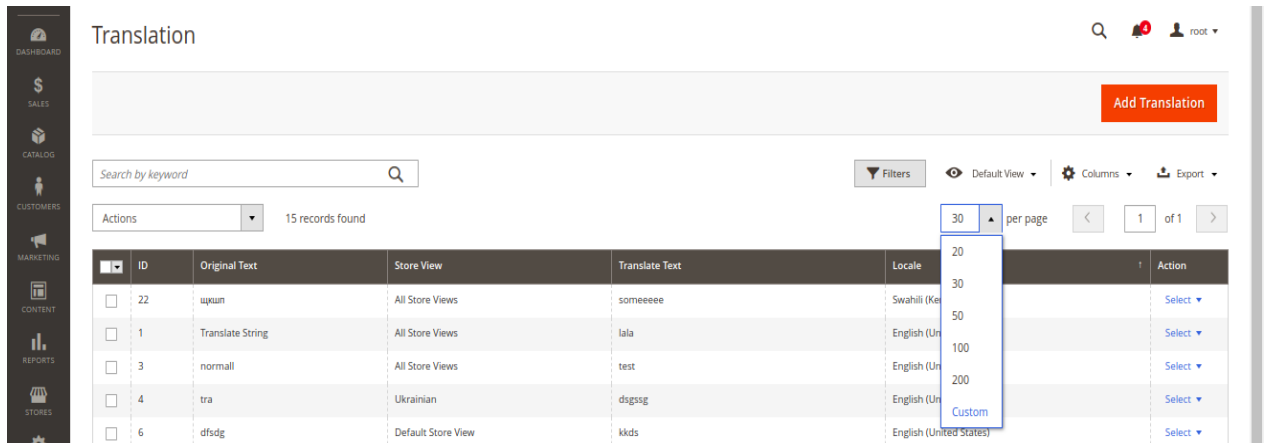


Рисунок 3.6.9 – Налаштування кількості записів на сторінці

Для експорту записів потрібно натиснути на кнопку «Export» та обрати потрібний формат «Excel XML», чи «CSV». Для того щоб експортувати лише кілька записів потрібно вибрати за допомогою чекбоксів потрібні записи, після зробити дії описані вище.

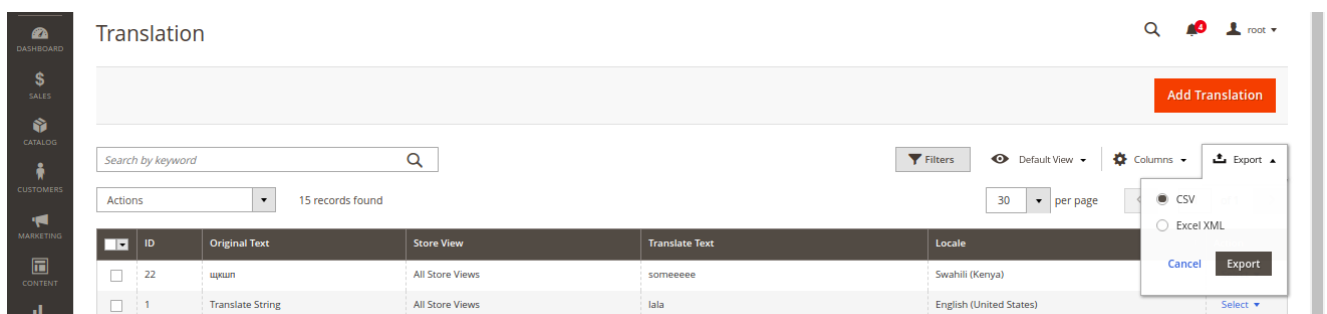


Рисунок 3.6.10 – Експорт записів

Експортований файл зображений на рисунку 3.6.11

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
44	43 Response Time	0	Response Time	en_US											
45	44 CPU Speed	0	CPU Speed	en_US											
46	45 Disk Speed	0	Disk Speed	en_US											
47	46 Brand	0	Brand	en_US											
48	47 New Customers		3 新的客户	zh_Hans_CN											
49	48 Registered Customers	0	3 已注册客户	zh_Hans_CN											
50	49 Registered Customers		3 已注册的客户	zh_Hans_CN											
51	50 审查及付款		3 付款	zh_Hans_CN											
52	51 Place Order		3 提交订单	zh_Hans_CN											
53	52 Qty		3 数量	zh_Hans_CN											
54	53 Item in Cart		3 件产品	zh_Hans_CN											
55	55 含。税	0	3 含税	zh_Hans_CN											
56	56 不包括/消费税		3 不含税	zh_Hans_CN											
57	57 Review & Payments		3 支付	zh_Hans_CN											
58	58 Payment Method:		3 付款方式:	zh_Hans_CN											
59	59 All Categories		3 全部目录	zh_Hans_CN											
60	60 Ship To		3 收件人	zh_Hans_CN											
61	61 Status		3 订单状态	zh_Hans_CN											
62	62 You subscribe to "General Subscriptio		3 您已订阅"普通订阅"	zh_Hans_CN											
63	63 Newsletters		3 订阅邮件资讯	zh_Hans_CN											
64	64 Go to Checkout		3 去结账	zh_Hans_CN											
65	65 View and edit cart		3 查看编辑购物车	zh_Hans_CN											
66	66 Update		3 更新	zh_Hans_CN											
67	67 Apply Discount Code		3 使用优惠券	zh_Hans_CN											
68	68 Apply Discount		3 确定使用	zh_Hans_CN											
69	69 Edit		3 编辑	zh_Hans_CN											

Рисунок 3.6.11 – Результат экспорту

Для імпорту записів потрібно перейти на сторінку «Export» з вкладки «System», заповнити поля, та завантажити файл для імпорту.

The screenshot displays the 'Import' settings interface in the Magento Admin. The left sidebar contains navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Styles, System, and Products & Extensions. The main content area is titled 'Import' and includes a 'Check Data' button. The 'Import Settings' section shows 'Entity Type' set to 'Magefan Translation' with a 'Download Sample File' link. The 'Import Behavior' section includes 'Import Behavior' set to '-- Please Select --', 'Stop on Error' set to 'Stop on Error', and 'Allowed Errors Count' set to 10. The 'Field separator' is set to a comma, and the 'Multiple value separator' is also set to a comma. The 'Fields enclosure' checkbox is unchecked. The 'File to Import' section has a 'Choose File' button and a note 'No file chosen'. The 'Images File Directory' field is empty, with a note below it: 'For Type "Local Server" use relative path to Magento installation, e.g. var/export, var/import, var/export/some/dir'.

Рисунок 3.6.12 – Імпорт перекладів

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці.

Зазвичай процес локалізації інтернет-магазину є довготривалим та виснажливим, так як приходиться працювати з багатьма мовами та великим обсягом даних, через це існує ймовірність припуститися помилки. Усім добре відомо, що одним із головних методів забезпечення ефективної роботи з комп'ютером є забезпечення належних умов праці, так як при недостатньому освітлені, просторі чи неякісному дисплеї, людина відчуватиме фізичний та з часом психологічний дискомфорт, який обов'язково відобразиться на результатах роботи. Для того щоб запобігти цьому, потрібно дотримуватись ряду наказів та стандартів.

В наказі № 207 від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [16], якраз описується частина вимог, яких потрібно дотримуватись при роботі з екранними пристроями.

Відповідно до третьої частини наказу [16], робоче місце працівника має відповідати наступним вимогам:

1. Робоче місце має мати достатні розміри, щоб працівник мав простір для зміни робочого положення та рухів.
2. Усе випромінювання від екранних пристроїв має бути знижене до гранично допустимого рівня з погляду безпеки та охорони здоров'я працівників.
3. Усі елементи робочого місця та їх розташування мають відповідати ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.
4. Освітлення робочого місця має створювати відповідний контраст між екраном і навколишнім середовищем та відповідати вимогам ДСанПН 3.3.2.007-98.

5. Мікроклімат виробничих приміщень має підтримуватись на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [17].

6. Робочий стіл чи поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, бути гнучкою під час розміщення екрана, клавіатури, документів чи устаткування.

7. Робоче крісло має бути стійким і дозволяти працівнику легко рухатися та займати зручне положення. Сидіння має регулюватися по висоті, спинка сидіння – по висоті, та з можливістю нахилу. Для зручності слід передбачати підніжку для тих, кому це необхідно.

Також потрібно пам'ятати про безпеку, відповідно до четвертої частини наказу [16], потрібно дотримуватись наступних мінімальних вимог безпеки:

1. Перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень.

2. Після закінчення роботи пристрої слід відключати від живлення.

3. При виникненні аварійної ситуації необхідно в той же час відключити пристрій від електричної мережі.

4. Не допускається:

– Ремонтувати чи виконувати технічне обслуговування, і налагодження екранних пристроїв на робочому місці працівника під час роботи з екранними пристроями;

– Вимикати захисні пристрої чи проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;

– Працювати з несправними екранними пристроями, у яких під час роботи виникають нехарактерні сигнали, мигання та інші несправності.

5. Під час виконання робіт з комп'ютером, пов'язаних з нервово-емоційним напруженням мають дотримуватись оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99 [17].

Для забезпечення безпеки відповідно до п'ятої частини наказу «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [16], усі екранні пристрої повинні відповідати наступним мінімальним вимогам безпеки:

Відповідно до третьої частини наказу [16], робоче місце працівника інтернет-магазину, який працює над контентним наповненням сайту має відповідати наступним вимогам:

1. Екранні пристрої не мають бути джерелом ризику для працівників.
2. Усе випромінювання має бути зведене до мінімального рівня з погляду безпеки і охорони здоров'я працівників.
3. Символи на дисплеї мають бути чіткими, відповідного розміру. Між символами і рядками символів повинна бути правильна відстань.
4. Зображення на дисплеї має бути стабільним, без миготінь або інших видів несправності.
5. Яскравість та контрастність символів має легко регулюватися, а також швидко адаптуватися до навколишніх умов.
6. Під час вибору монітора, слід надавати перевагу тим пристроям, які мають можливість повороту та нахилу екрану.
7. При потребі монітор може бути закріпленим на окремому столі чи підставці.
8. При виборі монітора надавайте перевагу дисплеям з матовим покриттям, щоб мінімізувати відблискування або відбивання світла.
9. При виборі клавіатури, слід надавати перевагу тій, яка відкидається і є автономною, щоб працівник міг вибрати зручну робочу позу й уникнути втоми рук.
10. Поверхня клавіатури має бути матовою, щоб уникнути віддзеркалювання.
11. Устаткування, яке входить до робочої станції, не повинно виділяти надлишкового тепла.
12. Під час розробки, вибору, замовлення та модифікації програмного забезпечення, а також під час розробки завдань, що передбачають використання

устаткування з екранними пристроями, роботодавець має керуватися таким програмним забезпеченням, яке відповідає розв'язуваним завданням і є простим у використанні, а де необхідно - адаптованим до рівня знань і досвіду працівника.

Отже, для безпечної та ефективної роботи контент-менеджера інтернет-магазину забезпечено належні умови праці, починаючи від робочого місця та його оснащення, та закінчуючи мікрокліматом робочого середовища, відповідно до вимог чинного законодавства.

4.2 Безпека в надзвичайних ситуаціях

Однією з найпоширеніших надзвичайних ситуацій є пожежа, тому розглянемо порядок дій при виникненні даної ситуації на підприємстві. Усі дії описані у Наказі [18] від 03.10.2017 № 1417 «Про затвердження Правил пожежної безпеки в Україні».

Згідно наказу [18] «Про затвердження Правил пожежної безпеки в Україні» порядок дій у разі пожежі має наступний вигляд:

1. При виявленні ознак пожежі, потрібно негайно повідомити про це пожежну службу зателефонувавши за номером 101, та надати інформацію про місцезнаходження об'єкта, вказати кількість поверхів будинку, місце виникнення надзвичайної ситуації, обстановку на пожежі, приблизну кількість та наявність людей, а також повідомити своє прізвище; за можливості вжити заходів щодо евакуювання людей, гасіння пожежі вогнегасниками та збереження матеріальних цінностей; якщо пожежа виникла на підприємстві, повідомити про неї керівника чи відповідну компетентну посадову особу та чергового; при необхідності викликати інші аварійно-рятувальні служби.

2. Працівник відповідальний за безпеку в надзвичайних ситуаціях, що прибув на місце пожежі, зобов'язаний: перевірити, чи викликана оперативно-рятувальна служба цивільного захисту, повідомити про надзвичайну ситуацію власника

підприємства; у разі загрози життю людей чим в той же час організувати їх евакуацію, використовуючи для цього доступні методи й засоби; вивести за межі небезпечної зони всіх працівників, не пов'язаних з ліквідуванням пожежі; припинити роботи в будинку (якщо це допускається технологічним процесом на виробництві та не призведе до погіршення ситуації), крім робіт, пов'язаних із заходами щодо ліквідування пожежі; здійснити в разі необхідності відключення електроенергії (за винятком систем протипожежного захисту), зупинення транспортуючих пристроїв, апаратів, перекриття газових, сировинних, водяних та парових комунікацій, зупинення систем вентиляції в аварійному та суміжних з ним приміщеннях (за винятком пристроїв протидимового захисту) та здійснити додаткові заходи, для того щоб, запобігти розвитку пожежі та задимленню будинку; перевірити включення систем оповіщення людей в разі виникнення надзвичайної ситуації, протидимового захисту, установок пожежогасіння; організувати зустріч підрозділів Оперативно-рятувальної служби цивільного захисту та надати їм допомогу у виборі оптимального шляху для під'їзду до епіцентру пожежі та в установці техніки на зовнішні джерела водопостачання; Паралельно з гасінням пожежі організувати евакуацію та захист матеріальних цінностей; контролювати дотримання безпеки праці працівниками, які беруть участь у локалізації пожежі.

3. При прибутті на пожежу пожежно-рятувальних підрозділів повинен бути забезпечений безперешкодний доступ до території об'єкта, за винятком випадків, коли чинним законодавством встановлений особливий порядок допуску.

4. Після прибуття пожежно-рятувальних підрозділів, інженерно-технічний персонал підприємства та адміністрація підприємства зобов'язані брати участь у консультуванні керівника ліквідації пожежі з приводу технологічних та конструктивних особливостей об'єкта, де сталась пожежа, прилеглих будівель та пристроїв, організувати залучення сил та засобів об'єкта до вжиття необхідних заходів, пов'язаних із ліквідацією пожежі та попередженням її розгання.

При виникненні пожежі вже на першій стадії виділяється теплота, токсичні продукти згоряння, можливі обвалення конструкцій. Тому слід враховувати необхідність евакуації людей у визначені строки. Показником ефективності процесу евакуації є час, протягом якого люди можуть при необхідності залишити окремі приміщення і будинок загалом. Безпека евакуації досягається тоді, коли тривалість евакуації людей з окремих приміщень і будинку в цілому менша критичної тривалості пожежі, що становить небезпеку для людини. Критичною тривалістю пожежі вважається час досягнення при пожежі небезпечних для людини температур і зменшення вмісту кисню у повітрі. Не вважаються евакуаційними виходи, що пов'язані з механічним приводом (ліфти, ескалатори), сходи, які не розташовані у сходових клітках. Евакуаційних виходів з приміщень кожного поверху має бути не менше двох. Мінімальна ширина шляхів евакуації — не менше 1 м, дверей – 0,8 м. Двері на шляхах евакуації мають відчинятися у напрямку виходу зі споруди. У кожному приміщенні (поверху) слід вивісити план евакуації людей і матеріальних цінностей.

Первинні засоби пожежогасіння — це вогнегасники, пісок, ковдри, лопати, сокири та ін. Широко застосовують пінні, порошкові та вуглекислотні вогнегасники. Кожний працівник повинен знати місце розташування первинних засобів пожежогасіння і вміти ними користуватися, працівники повинні знати правила поведінки при пожежі, шляхи евакуації. Легкозаймісті та горючі рідини дозволяється зберігати у спеціально відведених місцях, у межах їх потреби. Мастильні матеріали, легкозаймісті та горючі рідини повинні зберігатися окремо від інших матеріалів і речовин, у спеціальних ємкостях.

Отже, під час виникненням надзвичайної ситуації найголовнішим є швидке та розумне прийняття рішень, адже час на прийняття рішення може коштувати комусь життя. Саме тому й існують особи відповідальні за безпеку в надзвичайних ситуація так, як вони добре ознайомлені з правилами та порядком дій в таких випадках.

ВИСНОВКИ

У ході виконання дипломного проекту було досліджено проблему локалізації багатомовних інтернет-магазинів на платформі Magento 2, в ході досліджень було знайдено метод вирішення проблеми, а саме розширення базового функціоналу та побудова графічного інтерфейсу для роботи з перекладами.

Далі був проведений пошук патентів за схожою тематикою, за результатами якого схожих патентів не знайдено, тому можна приступати до етапу проектування. В результаті спроектовано програмний продукт «Magefan_Translation» за допомогою передових методів проектування програмного забезпечення, розроблено діаграми варіантів використання, класів, діяльності, які в подальшому допомогли на етапі конструювання системи.

Програмне забезпечення для перекладу тексту для багатомовних інтернет-магазинів на платформі Magento 2 сконструйовано на основі аналізу предметної області та описі архітектури програмної системи. Розширення розроблене відповідно до архітектурних вимог Magento 2.

Програмне забезпечення пройшло перевірку на якість коду та на інтеграційне тестування. У результаті розроблено модуль, який дозволить перекласти рутинну роботу з локалізації магазину з розробників на контент-менеджерів інтернет-магазину, що дозволяє пришвидшити, спростити та здешевити даний процес.

Створено інструкцію користувача, в якій детально описаний основний функціонал системи, такий як методи створення, редагування та видалення, та додатковий – пошук, фільтрація, імпорт та експорт даних. Для кращого розуміння роботи системи для кожного опису окремого функціоналу наведений скріншот з виконанням тієї чи іншої операції.

В подальшому система може бути розширена шляхом додавання нового функціоналу, який б зчитував усі файли з перекладами з модулів та тем та виводив їх данні в одну таблицю з урахування пріоритету того чи іншого перекладу.

СПИСОК ЛІТЕРАТУРИ

1. Magento 2.3 Developer Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://devdocs.magento.com>.
2. Translate Inline [Електронний ресурс] – Режим доступу до ресурсу: https://docs.magento.com/m2/ee/user_guide/system/translate-inline.html.
3. Пасічник О. Г. Основи веб-дизайну [Текст]: навч. посібн. / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко. – К. : Вид. група ВНУ, 2009 – 336 с.
4. Грофф Д.Р. SQL: полное руководство [Текст] : пер. с англ. / Джеймс Р. Грофф, Пол Н. Вайнберг. – К. : Издательская группа ВНУ, 1999. – 608 с.
5. Пьюривал Сэмми. Основы разработки веб-приложений [Текст] / Сэмми Пьюривал. – СПб. : Питер, 2015. – 272 с.
6. Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 [Текст] / Робин Никсон. – СПб. : Питер, 2016. – 768 с.
7. Колисниченко Денис. PHP и MySQL. Разработка Web-приложений [Текст] / Денис Колисниченко. – СПб. : БХВ-Петербург, 2015. – 593 с.
8. Прохоренок Николай. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера [Текст] / Николай Прохоренок, Владимир Дронов. – СПб. : БХВ-Петербург, 2015. – 766 с.
9. Бенедетти Райан. Изучаем работу с jQuery [Текст] / Райан Бенедетти, Ронан Крэнли. – СПб. : Питер, 2012. – 508 с.
10. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения [Текст] / Б. Хоган. – СПб. : Питер, 2014. – 320 с.
11. Stefanov Stoyan. JavaScript for PHP Developers [Текст] / Stoyan Stefanov. – O'Reilly, 2013. – 144 с.
12. Флэнаган Дэвид. JavaScript. Подробное руководство, 5-е издание [Текст] / Дэвид Флэнаган. – М. : Символ-Плюс, 2009. – 992 с.

13. Prettyman Steve. Learn PHP 7: Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL [Текст] / Steve Prettyman. – Apress, 2015. – 316 с.

14. Пасічник В. В. Веб-технології [Текст]: підруч. / В. В. Пасічник, О.В. Пасічник, Д. І. Угрин. – Львів : Магнолія 2013. – 336 с.

15. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (Освітньо-професійна програма - «Програмне забезпечення систем», Освітньо-наукова програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.: М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк - Тернопіль: ТНТУ, 2020-51с.

16. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>.

17. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text>.

18. Про затвердження Правил пожежної безпеки в Україні [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0252-15#Text>.

ДОДАТКИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ІМЕНІ ІВАНА ПУЛЮЯ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку проекту

«Розробка розширення для перекладу тексту для Magento 2»

Виконавець:

ст. гр. СПм-61

Бережний Б.І.

(підпис)

Керівник проекту:

Михалик Д.М

(підпис)

ЗМІСТ

1. ПІДСТАВИ ДО РОЗРОБКИ	61
2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	61
3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	61
3.1 Функціональні вимоги	61
3.2 Технічні вимоги	62
3.3 Програмні вимоги	62
4. ЕТАПИ РОЗРОБКИ	63
5. СУПРОВІДНА ДОКУМЕНТАЦІЯ	63
6. ПОРЯДОК ЗДАЧІ ПРОЕКТУ	64
7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ	64

1. ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану підготовки магістрів за спеціальністю 121 «Інженерія програмного забезпечення», та згідно наказу на виконання дипломної роботи студента-магістра.

Тема проекту: «Розробка розширення для перекладу тексту для Magento 2 на мові PHP».

Термін виконання: до __.__._____р.

2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Інформаційна система призначена для створення перекладів для з різних мовних пакетів встановлених у Magento 2.

Інформаційна система буде корисною для багатомовних інтернет магазинів.

Інформаційна система дозволить пришвидшити та покращити процес перекладу, оскільки всі дані про переклади зберігатимуться в одній таблиці яка має можливість фільтрування та пошуку даних, та на відмінну від стандартних засобі переклад можна робити коли магазин знаходиться у «Production mode».

3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Функціональні вимоги

У системі повинні бути реалізовані наступні можливості:

- Створення перекладу;
- Редагування перекладу;
- Видалення перекладу;
- Пошук;
- Фільтрація;
- Імпорт даних;
- Експорт даних;
- ACL.

Для операцій створення, редагування та видалення повинно бути реалізоване REST API.

3.2 Технічні вимоги

Вимоги до серверної частини:

- ОС Linux;
- не менше ніж 16Гб ОЗП;
- SSD 240 GB.

Вимоги до клієнтської частини:

- iOS/Android/Windows/macOS.

Додаткові вимоги:

- наявне підключення до мережі Інтернет.

3.3 Програмні вимоги

БД: MySQL 5.6, 5.7.

Версія PHP: від 7.1.3.

Встановленні PHP розширення: ext-bcmath, ext-ctype, ext-curl, ext-dom, ext-gd, ext-hash, ext-iconv, ext-intl, ext-mbstring, ext-openssl, ext-pdo_mysql, ext-simplexml, ext-soap, ext-spl, ext-xsl, ext-zip, lib-libxml.

Веб-сервер: Apache 2/Nginx.

4. ЕТАПИ РОЗРОБКИ

Розробка інформаційної системи проводиться в наступному порядку:

- аналіз предметної області, виявлення акторів та варіантів використання системи;
- вибір засобів розробки та архітектури системи;
- проектування архітектури;
- розробка програмного забезпечення системи;
- тестування інформаційної системи на реальних даних;
- оформлення супровідної документації;
- здача проекту.

Результати виконання кожного етапу проекту погоджуються з керівником проекту.

5. СУПРОВІДНА ДОКУМЕНТАЦІЯ

Для інформаційної системи повинні бути розроблені наступні документи:

- пояснювальна записка до проекту;

- презентація проекту;
- рецензія на проект;
- диск з проектом.

Пояснювальна записка до проекту оформляється згідно діючих вимог до нормоконтролю проектів.

6. ПОРЯДОК ЗДАЧІ ПРОЕКТУ

Розроблена інформаційна системи повинна відповідати вимогами, що складаються з перерахованих у п.3.1 цього документу характеристик.

Для задачі проекту необхідно підготувати весь перелік документів зазначений у п.5 цього документу.

Приймання проекту проводиться спеціально створеною комісією в термін зазначені в п.1 цього документу.

7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ

Назва етапу	Відмітка*
Аналіз предметної області	
Архітектура системи	
Проектування база даних	
Використання системи	
Супровідна документація	

* відмітки про виконання етапу ставляться керівником проекту

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

**ТЕРНОПІЛЬ
2020**

УДК 004.4 **Б.І. Бережний**

(Тернопільський національний технічний університет імені Івана Пулюя)

ЯК ПЕРЕКЛАСТИ ІНТЕРНЕТ-МАГАЗИН НА ПЛАТФОРМІ MAGENTO 2 НА УКРАЇНСЬКУ МОВУ?

UDC 004.4 **B.I. Berezhnyi**

HOW TO TRANSLATE AN ONLINE STORE ON THE MAGENTO 2 PLATFORM INTO UKRAINIAN?

Ключові слова: Magento 2, переклад, інтернет-магазин.

Весь контент у Magento 2 можна розділити на два види: статичний та динамічний. До статичного можна віднести системні файли, розширення, теми та інші. Даний контент практично не змінюється, тому для нього існує багато готових мовних пакетів. Отже найлегшим методом перекладу в даному випадку є скачування та встановлення готових словників.

Для перекладу динамічного контенту знадобиться більше часу, так як він є унікальним, та змінювати його потрібно в декількох місцях.

Прикладом динамічного контенту можуть служити:

- Продукти;
- CMS сторінки та блоки;
- Віджети;
- Сторонні розширення.

Більшість динамічного контенту перекладається в адмін панелі сайту, але є дані, які неможливо змінити шляхом редагування інформації в адмін панелі. Саме тому існують декілька методів для перекладу контенту, кожен з яких підходить в тій чи іншій ситуації.

Якщо вам необхідно перекласти cms block, продукт чи будь-яку іншу інформацію, яка створена та має можливість редагування безпосередньо з адмін-панелі, все що потрібно це переключитись на відповідне store view у випадку з продуктом або створити cms block з тим же ідентифікатором як і в блока для якого створюється переклад, після чого обрати потрібне store view та змінити контент.

Щоб додати переклад для статичних файлів можна скористатись словником перекладу. Це файл (.csv) із значеннями, які відокремлені комами, який має щонайменше два стовпці: базову фразу англійською (en_US) та переклад цієї фрази іншою мовою. Приклад перекладу з англійської (en_US) на українську (uk_UA):

"Add to Cart", "Купити",
"Add to Compare", "Порівняти",
"Sign In", "Увійти", Magento_Customer

Magento має інструмент для перекладу тексту під назвою Translate Inline, він дозволяє здійснювати переклад безпосередньо в браузері. Коли ви включите інструмент Translate Inline, на сторінці сайту біля слів, які доступні для перекладу будуть з'являтися іконки у вигляді словника. Щоб перевести потрібне слово, натисніть на іконку словника, після чого у спливаючому вікні введіть свій варіант перекладу. Оновіть сторінку, щоб побачити новий переклад.

Саме за допомогою використання вище наведених методів можна додати будь-яку мову для інтернет-магазину на платформі Magento 2.

