

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**на тему: «РОЗРОБКА МОДУЛЯ НАЛАШТУВАННЯ
ПОШТОВОГО ТРАНСПОРТНОГО ПРОТОКОЛУ
ДЛЯ CMS MAGENTO2»**

Виконав: студент 2 курсу, групи 8.1218
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми інженерія програмного забезпечення
(назва освітньої програми)
Д. С. Драсвський
(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.т.н., Мухін В.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної
математики, доцент, д.т.н., Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмна інженерія

Рівень вищої освіти магістр

Спеціальність інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А. О.

(підпис)

« _____ » _____ 2019 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ (СТУДЕНТЦІ)

Драєвському Дмитру Сергійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка модуля налаштування поштового
транспортного протоколу для CMS Magento2

керівник роботи (проекту) Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 29 » травня 2019 року № 811-с

2. Строк подання студентом роботи 27.12.2019

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Основні поняття.

2. Процес налаштування простого протоколу пересилання пошти для CMS Magento2.

3. Програмна реалізація модуля для простого протоколу пересилання пошти для CMS
Magento2.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 03.06.2019

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	25.06.2019	
2.	Збір вихідних даних.	30.08.2019	
3.	Обробка методичних та теоретичних джерел.	30.09.2019	
4.	Розробка першого і другого розділів.	31.10.2019	
5.	Розробка третього розділу.	25.11.2019	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	27.12.2019	
7.	Захист кваліфікаційної роботи.	15.01.2020	

Студент _____
(підпис)

Д. С. Драєвський _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

В. В. Мухін _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О. В. Кудін _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка модуля налаштування поштового транспортного протоколу для CMS Magento2»: 59 с., 19 рис., 1 табл., 8 джерел, 1 додаток.

РОЗРОБКА МОДУЛЯ НАЛАШТУВАННЯ ПОШТОВОГО ТРАНСПОРТНОГО ПРОТОКОЛУ ДЛЯ CMS MAGENTO2, CMS, MAGENTO2, PHP, SIMPLE MAIL TRANSPORT PROTOCOL, SMTP.

Актуальність досліджень в цьому напрямку підтверджується великим попитом на фахівців здатних зробити якісне та професійне налаштування поштового транспортного протоколу на виділеному сервері для своєчасного інформування своїх клієнтів інтернет-магазину за допомогою електронних листів.

Об'єкт дослідження – процес налаштування поштового транспортного протоколу для CMS Magento2.

Предмет дослідження – алгоритм налаштування поштового транспортного протоколу для CMS Magento2.

Мета роботи: розробити модуль для налаштування поштового транспортного протоколу для CMS Magento2.

Метод дослідження – методи системного аналізу, програмної інженерії та моделювання.

За результатами дослідження розроблено програмний модуль для налаштування поштового транспортного протоколу для CMS Magento2.

Одержані результати були використані в комерційній діяльності.

SUMMARY

Master's Qualification Thesis «The Development of the Mail Transport Protocol Configuration Module for CMS Magento2»: 59 pages, 19 figures, 1 tables, 8 references, 1 supplement.

THE DEVELOPMENT OF THE MAIL TRANSPORT PROTOCOL CONFIGURATION MODULE FOR CMS MAGENTO2, CMS, MAGENTO2, PHP, SIMPLE MAIL TRANSPORT PROTOCOL, SMTP.

The relevance of the study in this area is confirmed by the high demand for specialists capable of quality and professional customization of the mail transport protocol on a dedicated server for timely informing of their customers of e-commerce via e-mails.

Object of research – the process of the mail transport protocol configuration module for CMS Magento2.

Subject of research – the algorithm for the mail transport protocol configuration module for CMS Magento2.

The purpose of the thesis: develop module for mail transport protocol configuration of CMS Magento2.

Methods of research – methods of system analysis, software engineering and modeling.

According to the results of the study the program module for configuration of the mail transport protocol of CMS Magento2.

The obtained results were used in commercial activities.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ.....	7
1 Основні поняття.....	9
1.1 Поняття простого протоколу пересилання пошти	9
1.2 Налаштування простого протоколу пересилання пошти на прикладі Postfix для Ubuntu 18.04.....	16
1.3 Поверхневий огляд CMS Magento 2.....	19
2 Процес налаштування простого протоколу пересилання пошти для CMS Magento 2.....	26
2.1 Постановка задачі.....	26
2.2 Етапи розробки модуля для налаштування простого протоколу пересилання пошти для CMS Magento 2.....	27
2.3 Перелік технологій для розробки автоматизованого рішення для налаштування простого протоколу пересилання пошти	29
3 Програмна реалізація модуля для простого протоколу пересилання пошти для CMS Magento 2.....	31
Висновки.....	42
Перелік посилань.....	43
Додаток А Програмний код.....	44

ВСТУП

CMS Magento 2 – це платформа електронної комерції з відкритим кодом, написана на PHP. Це одна з найпопулярніших систем відкритої електронної комерції в мережі. Це програмне забезпечення створене за допомогою Zend Framework. Вихідний код Magento поширюється в рамках Ініціативи з відкритим кодом (OSI), затвердженої Ліцензією на відкрите програмне забезпечення (OSL) v3.0 .

CMS Magento 2, як і більшість інших e-Commerce платформ, являє собою не тільки продаж або купівлю товарів певного роду, а й комунікація зі своїми відвідувачами. Комунікація є невід’ємною частиною будь-якого сайту. Тому, так важливо мати можливість у простому та якісному налаштуванні зв’язку між власником електронного магазину та його клієнтами.

Актуальність досліджень в цьому напрямку підтверджується тим, що CMS Magento 2 є найбільшою платформою електронної комерції на даний час та нараховує більш ніж 250 тисяч створених електронних магазинів на її базі.

Поставлена задача полягає із декількох підзадач, а саме:

- а) реалізувати зручний користувачький інтерфейс для успішного налаштування поштового транспортного протоколу;
- б) отримати дані задані користувачем через інтерфейс;
- в) обробити дані та провести конфігурацію власного поштового транспортного протоколу;
- г) замінити стандартний поштовий транспортний протокол.

Метою дипломного проекту є: розробка модуля для налаштування поштового транспортного протоколу для CMS Magento2.

Для досягнення поставленої мети були визначені такі завдання дослідження:

а) проаналізувати існуючий метод відправлення електронної пошти за допомогою стандартного поштового транспортного протоколу для CMS Magento 2;

б) описати особливості та проблеми використання стандартного підходу для відправлення електронної пошти у CMS Magento 2;

в) розробити модуль для налаштування поштового транспортного протоколу для CMS Magento2.

1 ОСНОВНІ ПОНЯТТЯ

1.1 Поняття Простого Протоколу Пересилання Пошти

SMTP (англ. Simple Mail Transfer Protocol – простий протокол передачі пошти) – це широко використовуваний мережевий протокол, призначений для передачі електронної пошти в мережах TCP / IP.

SMTP вперше був описаний в RFC 821 (1982 рік); останнє оновлення в RFC 5321 (2008) включає масштабується розширення – ESMTP (англ. Extended SMTP). В даний час під «протоколом SMTP» як правило мають на увазі і його розширення. Протокол SMTP призначений для передачі вихідної пошти з використанням порту TCP 25.

У той час, як електронні поштові сервери та інші агенти пересилання повідомлень використовують SMTP для відправки та отримання поштових повідомлень, що працюють на призначеному для користувача рівні клієнтські поштові програми, зазвичай використовують SMTP тільки для відправки повідомлень на поштовий сервер для ретрансляції. Для отримання повідомлень клієнтські програми зазвичай використовують або POP (англ. Post Office Protocol – протокол поштового відділення), або IMAP (англ. Internet Message Access Protocol), або патентовані системи (такі як Microsoft Exchange і Lotus Notes / Domino) для доступу до профілю записи своєї поштової скриньки на сервері.

Електронна пошта представлена поштовим клієнтом (MUA, mail user agent – призначений для користувача поштовий агент) для поштового сервера (MSA, mail submission agent – агент відправки електронної пошти) за допомогою SMTP по TCP-порту 587. Звідти MSA доставляє пошту своїм агентам передачі повідомлень (MTA, mail transfer agent). Часто ці два агенти є просто різними зразками одного і того ж програмного забезпечення, запущеного з різними параметрами на одному пристрої. Локальна обробка

може бути проведена як на окремій машині, так і розділена між різними пристроями; в першому випадку залучені процеси мають загальний доступ до файлів, у другому випадку SMTP використовується для пересилання повідомлення внутрішньо, причому кожен хост налаштований на використання наступного пристрою в якості проміжного хоста.

Граничний МТА повинен знайти цільовий хост. Він використовує систему доменних імен (DNS) для пошуку записів поштового обмінника (mail exchanger – MX) домену одержувача (частина адреси, що знаходиться праворуч від символу @). Повертається запис поштового MX, містить ім'я цільового хоста. Потім МТА підключається до сервера обміну в якості SMTP-клієнта.

Як тільки мета MX приймає вхідне повідомлення, вона передає його агенту доставки пошти (mail delivery agent – MDA) для локальної доставки повідомлення. MDA передбачає можливість зберігати повідомлення у відповідному форматі поштової скриньки. Прийом пошти, знову ж таки, може бути проведений як декількома, так і одним комп'ютером. MDA може доставляти повідомлення прямо на зберігання або передавати їх по мережі за допомогою SMTP або будь-яких інших засобів, в тому числі протоколу локальної пересилання пошти (Local Mail Transfer Protocol – LMTP) – похідного від SMTP, призначеного для цієї мети.

Після доставки на локальний поштовий сервер повідомлення зберігається для пакетного пошуку по аутентифіцироваться поштових клієнтів (MUA). Повідомлення витягується додатками кінцевого користувача (поштовими клієнтами) з використанням протоколу IMAP (Internet Message Access Protocol), який полегшує доступ до повідомлень і управляє, зберігається поштою, або за допомогою протоколу POP (Post Office Protocol), який зазвичай використовує традиційний mbox-формат файлів, або фірмовими системами на кшталт Microsoft Exchange / Outlook або Lotus Notes / Domino. Клієнти мережевий пошти можуть використовувати будь-який метод, але протокол пошуку часто не відповідає офіційним стандартам.

SMTP визначає передачу повідомлення, а не його зміст. Таким чином, він задає оболонку повідомлення і її параметри (такі, як відправник оболонки), але не заголовок або тіло самого повідомлення. STD 10 і RFC 5321 визначають SMTP (оболонку), в той час як STD 11 і RFC 5322 – повідомлення (заголовок і тіло), офіційно званий форматом поштового повідомлення (Internet Message Format).

SMTP вимагає з'єднання – текстовий протокол, по якому відправник повідомлення зв'язується з одержувачем за допомогою видачі командних рядків і отримання необхідних даних через надійний канал, в ролі якого зазвичай виступає TCP-з'єднання (Transmission Control Protocol – протокол управління передачею). SMTP-сесія складається з команд, що посилаються SMTP-клієнтом, і відповідних відповідей SMTP-сервера. Коли сесія відкрита, сервер і клієнт обмінюються її параметрами. Сесія може включати нуль і більш SMTP-операцій (транзакцій).

SMTP-операція складається з трьох послідовностей команда / відповідь (див. Приклад нижче). Опис послідовностей:

- MAIL FROM встановлює зворотну адресу (тобто Return-Path, 5321.From, mfrom). Це адреса для повернутих листів;
- RCPT TO встановлює одержувача даного повідомлення. Ця команда може бути дана кілька разів, по одній на кожного одержувача. Ці адреси також є частиною оболонки;
- DATA для відправки тексту повідомлення. Це саме вміст листа, в протилежність його оболонці. Він складається з заголовка повідомлення і тіла повідомлення, розділених символом нового рядка. DATA, по суті, є групою команд, а сервер відповідає двічі: перший раз на саму команду DATA, для повідомлення про готовність прийняти текст; і другий раз після кінця послідовності даних, щоб прийняти або відхилити все лист.

Крім проміжних відповідей для DATA-команди, кожен відповідь сервера може бути позитивним (код відповіді 2xx) або негативним. Останній, в свою чергу, може бути постійним (код 5xx) або тимчасовим (код 4xx).

Відмова SMTP-сервера в передачі повідомлення – постійна помилка; в цьому випадку клієнт повинен відправити повернутий лист.

Початковим хостом (SMTP-клієнтом) може бути як поштовий клієнт кінцевого користувача (функціонально визначається як поштовий агент – MUA), так і агент пересилки повідомлень (MTA) на сервері, тобто сервер діє як клієнт у відповідній сесії для ретрансляції повідомлення. Повністю функціональні сервера підтримують черги повідомлень для повторної передачі повідомлення в разі помилок. [1]

MUA знає SMTP-сервер для вихідної пошти зі своїх налаштувань. SMTP-сервер, який діє як клієнт, тобто пересилає повідомлення, визначає, до якого сервера підключитися, переглядом ресурсу записів MX (Mail eXchange) DNS для домену кожного одержувача. У разі, якщо запис MX не знайдено, сумісні MTA (в повному обсязі) повертаються до простої A-записи.

SMTP-сервер, який діє як клієнт, встановлює TCP-з'єднання з сервером за розробленим для SMTP порту 25. MUA повинен використовувати порт 587 для підключення до агента надання повідомлень (MSA). Основна відмінність між MTA і MSA полягає в тому, що SMTP-аутентифікація обов'язкове тільки для останнього.

SMTP всього лише протокол доставки. Він не може на вимогу взяти повідомлення з віддаленого сервера. Для вилучення пошти та управління поштовою скринькою розроблені інші протоколи, такі як POP і IMAP. Проте, SMTP надає можливість почати на віддаленому сервері обробку черги повідомлень, при якій запитуюча система може отримувати всі спрямовані їй повідомлення (див. Remote Message Queue Starting нижче). POP і IMAP кращі, коли комп'ютер користувача включено не постійно, або ж тимчасово підключений до Інтернету.

Remote Message Queue Starting (запуск віддаленої черги повідомлень) – особливість SMTP, що дозволяє віддаленого хосту почати обробку черги повідомлень на сервері так, що він може отримувати призначені йому повідомлення за допомогою команди TURN. Однак ця особливість вважалась

небезпечною і була розширена в RFC 1985 командою ETRN, яка працює надійніше завдяки заснованому на інформації DNS методу аутентифікації.

ODMR (On-Demand Mail Relay – ретрансляція пошти на вимогу) – стандартизоване в RFC 2645 SMTP-розширення, що дозволяє проводити ретрансляцію повідомлення аутентифікованим користувачеві.

Поштовий клієнт повинен знати IP-адресу SMTP-сервера, який задається як частина конфігурації (звичайно у вигляді DNS-імені). Сервер буде доставляти вихідні повідомлення від особи користувача.

Адміністраторам сервера необхідно контролювати те, які клієнти можуть використовувати сервер. Це дозволяє їм боротися з такими зловживаннями, як спам. Зазвичай використовуються два рішення:

- у минулому багато систем вводили обмеження по місцю розташування клієнта, допускаючи до використання лише тих, чий IP-адреса був серед підконтрольних адміністраторам;

- сучасні сервери зазвичай пропонують альтернативну систему, що вимагає аутентифікацію клієнтів для отримання доступу.

В цьому випадку SMTP-сервер інтернет-провайдера не дозволить допуск користувачам «за межами» мережі провайдера. Точніше, сервер може допустити лише тих користувачів, чий IP-адреса надано даними провайдером, що еквівалентно вимогу з'єднання з Інтернетом за допомогою цього провайдера. Мобільний користувач часто може виявитися в мережі, відмінній від мережі свого провайдера, і тому повідомлення не будуть відправлятися.

У даної системи є кілька різновидів. Наприклад, SMTP-сервер організації може надавати доступ тільки користувачам тієї ж мережі, блокуючи інших користувачів. Також сервер може проводити ряд перевірок клієнтського IP-адреси. Ці методи зазвичай використовувалися організаціями та установами, наприклад університетами, для внутрішнього користування сервером. Однак, велика їх частина тепер використовує описані нижче методи аутентифікації.

Завдяки обмеженню доступу певним адресами, адміністратори сервера можуть легко визначити адресу будь-якого зловмисника. Якщо користувач може використовувати різних провайдерів для з'єднання з Інтернетом, цей вид обмеження стає недоцільним, а зміна налаштованої адреси SMTP-сервера вихідної пошти непрактично. Вкрай бажано мати можливість використовувати таку інформацію про екологічні атрибути Вашого клієнта, яка не потребує зміни.

Замість описаного раніше обмеження по місцю розташування, сучасні SMTP-сервери зазвичай вимагають аутентифікацію користувачів перед отриманням доступу. Ця система, будучи більш гнучкою, підтримує мобільних користувачів і надає їм фіксований вибір налаштованого сервера вихідної пошти.

Адміністратори сервера вибирають, який порт будуть використовувати клієнти для ретрансляції вихідної пошти – 25 або 587. Специфікації та багато сервери підтримують і той, і інший порти. Хоча деякі сервери підтримують порт 465 для безпечного SMTP, але краще використовувати стандартні порти і ESMTP-команди, якщо необхідна захищена сесія між клієнтом і сервером.

Деякі сервери налаштовані на відхилення всіх ретрансляцій по порту 25, але користувачам, які пройшли аутентифікацію по порту 587, дозволено перенаправляти повідомлення на будь-який дійсну адресу.

Деякі провайдери перехоплюють порт 25, перенаправляючи трафік на свій власний SMTP-сервер незалежно від адреси призначення. Таким чином, їх користувачі не можуть отримати доступ до сервера за межами провайдерської мережі по порту 25.

Деякі сервери підтримують аутентифікований доступ по додатковому, відмінному від 25, порту, дозволяючи користувачам з'єднуватися з ними, навіть якщо порт 25 заблокований.

Початкова специфікація SMTP не включала коштів для аутентифікації відправників. Згодом, в RFC 2554 було введено розширення. Розширення SMTP (ESMTP) надає поштових клієнтів можливість завдання механізму

забезпечення безпеки для сервера, аутентифікації і профілю безпеки SASL (Simple Authentication and Security Layer) для наступних передач повідомлень.

Продукти Microsoft реалізують власний протокол – SPA (Secure Password Authentication) за допомогою розширення SMTP-AUTH.

Однак, непрактичність широкого поширення реалізації і управління SMTP-AUTH означає, що проблема спаму не може бути вирішена за його допомогою.

Широке зміна SMTP, так само як і повна його заміна, вважаються не практичними через величезну інсталлированной бази SMTP. Internet Mail 2000 був одним з претендентів для такої заміни.

Спам функціонує завдяки різним чинникам, в тому числі не відповідають стандартам реалізації МТА, уразливості в захисті операційних систем (зокрема через постійним широкосмуговим підключенням), що дозволяє спамерам віддалено контролювати комп'ютер кінцевого користувача і посилати з нього спам.

Існує кілька пропозицій для побічних протоколів, які допомагають роботі SMTP. Дослідницька група Anti-Spam (The Anti-Spam Research Group – ASRG) – підрозділ Дослідницької групи Інтернет-технологій працює над поштової аутентифікацією і іншими пропозиціями для надання простий аутентифікації, яка буде гнучкою, легковажною і масштабованою. Недавня діяльність Інженерного ради Інтернету (IETF) включає в себе MARID (2004), який призвів до двох затвердженим IETF-експериментів в 2005, і DomainKeys Identified Mail в 2006. [4]

1.2 Налаштування Простого Протоколу Пересилання Пошти на прикладі Postfix для Ubuntu 18.04

Postfix є агентом передачі пошти (MTA) в Ubuntu за замовчуванням. Він розроблений щоб бути швидким, простим в адмініструванні і безпечним. Він сумісний з MTA sendmail. Цей розділ описує як встановити та налаштувати postfix. Тут також пояснюється як зробити його SMTP сервером, що використовує безпечні з'єднання (для безпечної передачі повідомлень).

Для установки postfix виконайте наступну команду:

```
sudo apt-get install postfix
```

Тепер саме час вирішити який формат поштової скриньки ви хочете використовувати. За замовчуванням postfix використовує формат mbox. Замість безпосереднього редагування файлу конфігурації ви можете використовувати команду postconf для настройки параметрів postfix. Параметри будуть збережені у файлі /etc/postfix/main.cf. Надалі якщо ви вирішите переконфігурувати окремі параметри ви можете як запустити команду, так і вручну виправити файл.

Для настройки формату поштової скриньки для Maildir:

```
sudo postconf -e 'home_mailbox = Maildir /'
```

SMTP-AUTH дозволяє клієнту ідентифікувати себе через механізм аутентифікації (SASL). Транспортний рівень безпеки (TLS) буде використовуватися для шифрування процесу аутентифікації. Після аутентифікації SMTP сервер дозволить клієнтові передавати пошту:

а) налаштуйте Postfix на SMTP-AUTH з використанням SASL (Dovecot SASL):


```

sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private / auth-client'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,
permit_mynetworks,
reject_unauth_destination'

```

б) далі створіть або отримаєте цифровий сертифікат для TLS;

в) як тільки у вас з'явився сертифікат, налаштуйте Postfix на використання TLS шифрування як для вхідної, так і для вихідної пошти:

```

sudo postconf -e 'smtp_tls_security_level = may'
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.example.com'

```

г) якщо ви використовуєте власний Центр сертифікації, для підпису сертифіката введіть:

```

sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'

```

Після виконання всіх команд Postfix налаштований на SMTP-AUTH і самопідписаний сертифікат створений для TLS шифрування.

Початкове налаштування postfix закінчено. Виконайте наступну команду для перезапуску різноманітних служб postfix:

```
sudo /etc/init.d/postfix restart
```

Postfix підтримує SMTP-AUTH як описано в RFC2554. Він заснований на SASL. Однак все-таки необхідно налаштувати аутентифікацію перед тим, як ви зможете використовувати SMTP-AUTH.

Postfix підтримує дві реалізації SASL: Cyrus SASL і Dovecot SASL. Щоб дозволити Dovecot SASL, потрібно встановити пакет dovecot-common. Для цього з терміналу введіть наступне:

```
sudo apt-get install dovecot-common
```

Далі вам потрібно відредагувати `/etc/dovecot/dovecot.conf`.

Щоб дозволити використовувати SMTP-AUTH клієнтам Outlook, в секції `auth default` файлу `/etc/dovecot/dovecot.conf` додайте "login":

```
mechanisms = plain login
```

Після того, як Dovecot налаштований, перезапустіть його:

```
sudo /etc/init.d/dovecot restart
```

Іншою опцією настройки Postfix для SMTP-AUTH є використання пакету `mail-stack-delivery` (раніше він називався `dovecot-postfix`). Цей пакет встановить Dovecot і налаштує Postfix для його використання спільно з SASL аутентифікацією і як агента доставки пошти (MDA). Пакет також налаштує Dovecot для IMAP, IMAPS, POP3 і POP3S.

Щоб встановити пакет, введіть в терміналі:

```
sudo apt-get install mail-stack-delivery
```

У вас тепер робочий поштовий сервер, але існує кілька опцій, які ви, можливо, захочете змінити в подальшому. Наприклад, пакет використовує сертифікат і ключ від `ssl-cert` пакета, і в робочому середовищі, ви повинні використовувати сертифікат і ключ, згенерований для хоста. Дивіться `дбав Сертифікати для додаткових деталей`.

Після того, як ви отримали замовлений сертифікат для сервера, замініть наступну опцію в `/etc/postfix/main.cf`:

```
smtpd_tls_cert_file = /etc/ssl/certs/ssl-mail.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-mail.key
```

Налаштування SMTP-AUTH завершено. [2]

1.3 Поверхневий огляд CMS Magento 2.

Magento – це найпопулярніша платформа електронної комерції в світі, яка обробляє понад 155 млрд. Доларів США загального обсягу товарів, що продаються щорічно на цій платформі. Є однією з найбільш часто використовуваних CMS для створення інтернет-магазинів в усьому світі завдяки своїй потужній масштабованій архітектурі.

Є багато причин, чому великі бренди довіряють Magento. Ось лише деякі з кращих функцій Magento 2:

а) Magento 2 сумісний з найсучаснішими технологіями, такими як PHP 7, що означає швидке завантаження (в порівнянні з Magento 1). Magento 2 може обробляти 10 млн. переглядів сторінок за годину;

б) мульти-магазин. Magento надає просту можливість створювати додаткові сайти за допомогою підкаталогів і варіантів дизайну;

в) легка панель адміністратора. Вона дуже зручна для настройки, створення товарів, навігації та фільтрації даних;

г) кращий процес покупок. Легке оформлення замовлення, зручний інтерфейс для мобільних пристроїв UI / UX і поліпшена функціональність пошуку;

д) нові можливості і сучасні технології. PHP7, HTML5, CSS3 і препроцесори LESS / SASS, RequireJS, налаштування/оновлення за допомогою Composer тощо;

е) масштабованість і модульність дозволяють розширювати Magento повністю під свої потреби;

ж) Magento має глобальну екосистему, що складається з понад 300 000 розробників і мережу з більш ніж 1000 висококваліфікованих партнерів по технологіям та рішенням.

Magento має один з найбільших списків функцій, який ви коли-небудь знайдете в термінах програмного забезпечення для електронної комерції (див. рис. 1.1). Це точна причина, чому ставки настільки високі і чому всі клієнти Magento є великими корпораціями.

Включено все, від маркетингу до сегментації клієнтів. Стандартні функції поставляються разом з відкритим вихідним кодом Magento, тому ви можете працювати над пошуковою оптимізацією за допомогою інструментів для карт Google, API контенту Google і дружніх пошукових систем. [3]

Маркетингові та конверсійні інструменти включають коди купонів, групи клієнтів, пакети продуктів і управління інформаційними бюлетенями. Список дійсно нескінченний, коли справа доходить до функцій, але ось всеосяжна розбивка найбільш важливих областей:

- SEO;
- маркетинг, рекламні акції і конверсії;
- управління сайтом;
- управління каталогом;
- перегляд каталогу;
- перегляд продукту;
- оформлення замовлення, оплата та доставка;

- управління замовленнями;
- поліпшення рівня клієнтського сервісу;
- мобільна комерція;
- стійкі покупки;
- приватні продажі;
- подарункові реєстри;
- збереження кредитних карток.

Built-in Features	Open Source	Magento Commerce
Integrated Checkout, Payment, and Shipping	✓	✓
Mobile Optimized Shopping	✓	✓
Global Selling	✓	✓
Catalog Management	✓	✓
Extended Functionality via App Marketplace	✓	✓
Instant Purchase	✓	✓
Site Search	✓	✓
Business Intelligence Dashboards		✓
B2B Functionality		✓
Advanced Marketing Tools		✓
Content Staging and Preview		✓
Customer Loyalty Tools		✓
WYSIWYG Page Builder		✓
Visual Merchandising		✓
Magento Hosted Cloud		
Optimized Cloud Commerce for Magento		✓
Global Availability		✓
Cloud availability SLA up to 99.99%		✓
Fast Page Loading with Content Delivery Network (CDN)		✓
Performance Monitoring Tools		✓
Enhanced Security with Fastly WAF and DDoS services		✓
PCI Compliant		✓
Easy deployment		✓
Streamlined updates and testing		✓
Support		
Account Manager		✓
Technical Support		✓

Рисунок 1.1 – Опис функціональних можливостей Magento 2 EE та CE

Розробники вважають інтерфейс Magento цілком розумним, тому що він має швидку і просте налаштування самотійного хостингу, і все

об'єднано в навігаційне меню, таке як клієнти, продукти, рекламні акції, продажі та звіти.

Magento пропонує вам кілька варіантів на вибір. Перший називається Magento Open Source (раніше Community Edition). Ця пропозиція з відкритим вихідним кодом який доступний безкоштовно. Тому будь-хто може зайти на сайт Magento і отримати копію програмного забезпечення.

У вас також є можливість підписатися на безкоштовну демонстрацію плану Magento Commerce, але для цього необхідно відправити повідомлення з усією вашою контактною інформацією.

Пакет Magento Commerce – це потужна система для великих і швидкозростаючих підприємств. Навіть несправедливо порівнювати його з такими опціями, як Volusion або Shopify, тому що можливості з Magento безмежні. При цьому Magento пропонує платформу хмарної електронної комерції, оскільки до цього всі вже звикли. Майте на увазі, що ціни на Magento Commerce зазвичай налаштовуються для кожного бізнесу і можуть стати дуже дорогими.

Оскільки Magento вимагає деяких додаткових технічних знань для налаштування, Magento пропонує допомогу з налаштування для великих компаній.

В цілому, може здатися, що ви втрачаєте найпотужнішу платформу електронної комерції в світі, якщо ви не використовуєте Magento. Це одне з кращих рішень для розширення бізнесу в Інтернеті.

Якщо ви працюєте в невеликій бізнесі з невеликим або невизначеним фінансовим потоком, подумайте про використання версії з відкритим вихідним кодом Magento. [5]

Навіть якщо ви користуєтеся Magento Open Source, це не означає, що ви зможете безкоштовно створювати і управляти своїм сайтом. Це платформа з відкритим вихідним кодом, тому вам потрібно буде знайти свій власний хостинг, тему і т.д. Є також деякі інші витрати, такі як доменне ім'я і сертифікат SSL. Потрібно також партнер по обробці платежів.

Тепер ці витрати не сильно відрізняються від витрат WordPress, але ви повинні їх враховувати. Якщо ви подивіться на ціну \$0 і скажете: «Це здорово, звичайно, я використовую безкоштовну платформу для електронної комерції», пам'ятайте, що вона не повністю безкоштовна, і вам все ще потрібен розробник, щоб ваш сайт виглядав добре.

Компанія Magento не продає і не віддає ніяких шаблонів, але спеціальні інструменти дизайну чудові, якщо ви знаєте, що робите (див. рис. 1.2).



Рисунок 1.2 – Приклад комерційної версії теми для Magento 2

Управління запасами Magento дозволяє виконувати різні модифікації, щоб не відставати від того, які товари надходять і виходять з вашого магазину. Наприклад, ви можете перевірити відсталі елементи і змінити номери для мінімального і максимального кількостей.

Продавайте, завантажуйте продукти в своєму магазині, призначайте необмежені атрибути продукту і навіть помірні теги продуктів і огляди, щоб підтримувати повний контроль над своїм власним сайтом. [6]

Ви також можете зв'язатися з різними каналами продажів, такими як eBay і Amazon. Що добре в інтеграції Magento, так це те, що весь ваш

інвентар синхронізується з сайтом. Таким чином, ваші клієнти не будуть купувати товари, яких немає в наявності.

Ще одна річ, яка мені подобається в інвентарі Magento – це автоматизація. Алгоритми використовуються для розрахунку доставки і податків. Ви також можете надати своїм клієнтам кілька унікальних опцій покупки, таких як пошук по сайту і кнопка миттєвої покупки. Глобальні продажі також дають вам чимало переваг, оскільки ви можете підтримувати такі речі, як різні валюти і мови. [7]

Приємна річ про Magento полягає в тому, що вам не потрібно використовувати інші маркетингові інструменти, щоб отримати все, що ви хочете, від маркетингу електронної пошти до оголошень на сайті. SEO автоматично налаштовується, якщо ви не хочете самостійно входити і самі міняти.

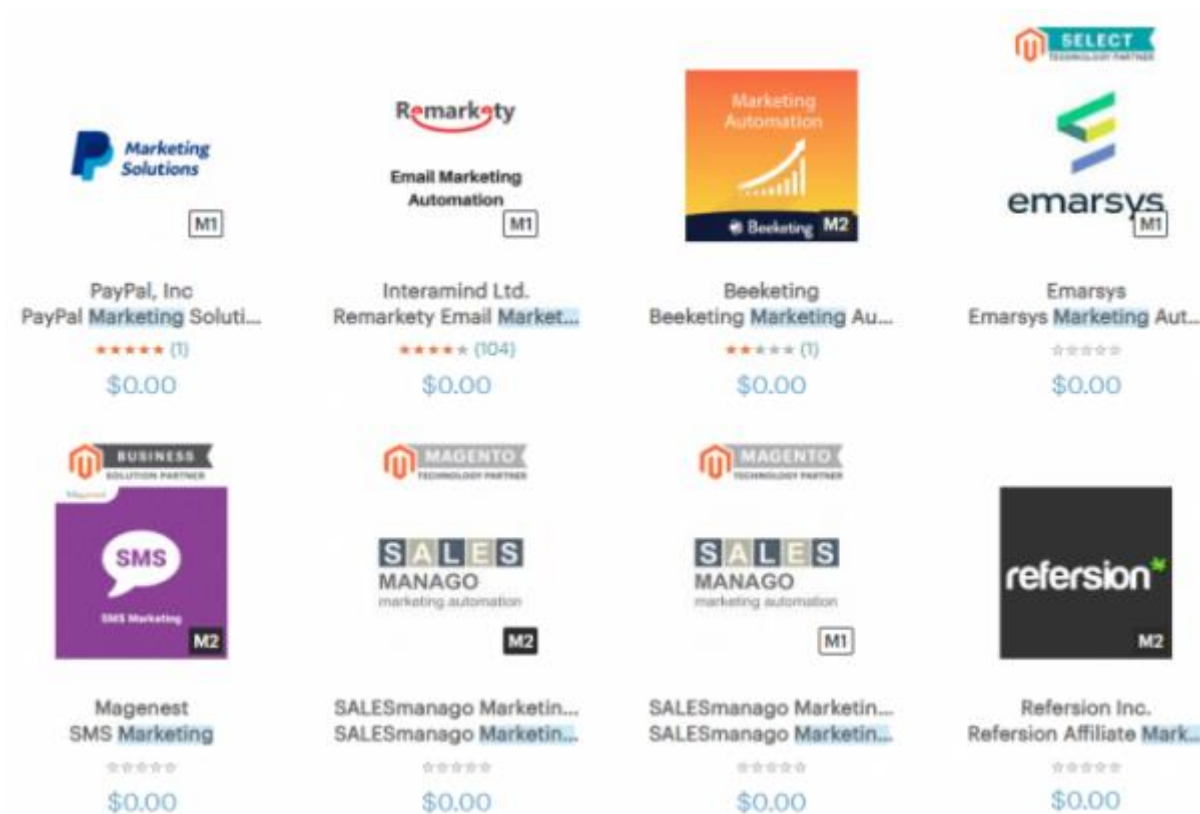


Рисунок 1.3 – Приклад безкоштовних SEO-модулів для Magento 2

Маркетингові переваги включають в себе здатність купувати продукти, пропонувати рекламні ціни і видавати купони. Так само є можливість платформи електронної пошти і багаторівневі ціни для знижки за кількість, змушуючи людей купувати трохи більше, щоб досягти цих знижок.

Деякі з більш просунутих функцій маркетингу як і раніше зарезервовані тільки для тих, хто використовує Magento Commerce. Яскравим прикладом цього можуть служити вбудовані інструменти лояльності клієнтів. Тому, якщо ви працюєте тільки з відкритим вихідним кодом Magento, вам потрібно буде знайти додаток або розробника, щоб створити власну програму лояльності.

Нарешті, мобільна оптимізація і стандартні інструменти SEO включені в обидві версії Magento. Таким чином, ви можете бути спокійні, знаючи, що пошукові системи знаходять ваш сайт, і все добре відображається на невеликих пристроях. [4]

2 ПРОЦЕС НАЛАШТУВАННЯ ПРОСТОГО ПРОТОКОЛУ ПЕРЕСИЛАННЯ ПОШТИ ДЛЯ CMS MAGENTO 2

2.1 Постановка задачі

Метою роботи є розробка модуля для налаштування поштового транспортного протоколу для CMS Magento2. Розробка модуля є, здебільшого, маніпуляцією з стандартними механізмами Magento, які відповідають за відправку електронної пошти, тому етапи розробки було поділено на Frontend частину та Backend частину (див рис. 2.1).

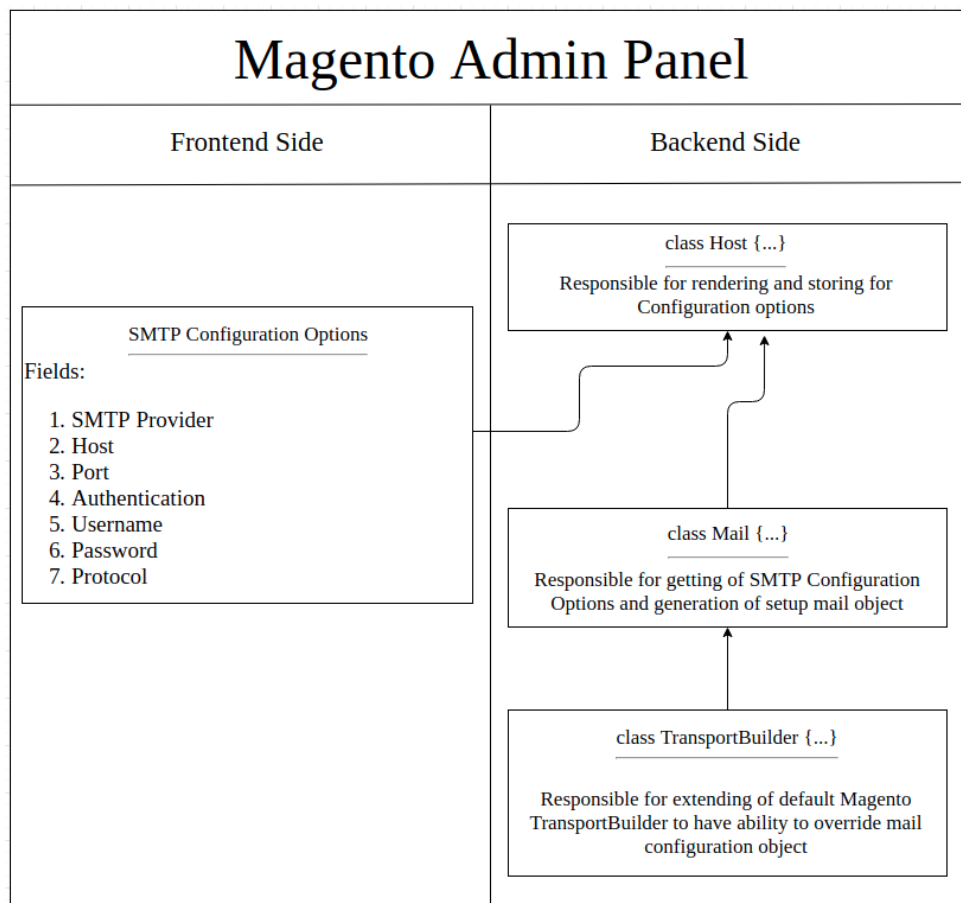


Рисунок 2.1 – Ілюстрація загальної логіки модуля, який розробляється

2.2 Етапи розробки модуля для налаштування простого протоколу пересилання пошти для CMS Magento 2

Для того, щоб отримати бажаний результат, необхідно поступово та сплановано підходити до процесу розробки модуля. Перш за все, необхідно створити користувацький інтерфейс для адміністраторів Magento. Треба передбачити усю необхідну інформацію, яка би потребувалась для налаштування нашого модуля, а саме для налаштування простого протоколу пересилання пошти. Заздалегідь, вивчивши різноманітні ресурси, було сформовано список необхідних атрибутів для налаштування простого протоколу пересилання пошти:

- а) SMTP Provider;
- б) Host;
- в) Port;
- г) Authentication;
- д) Username;
- е) Password;
- ж) Protocol.

За допомогою цього набору атрибутів, будь-який адміністратор Magento буде мати змогу сконфігурувати простий протокол пересилання пошти та цього набору буде достатньо, щоб розробити модуль, у якому backend сторона мала би змогу налаштувати протокол.

Після попереднього етапу, необхідно розробити backend-частину. Для цього, необхідно отримати усі задані атрибути із попереднього етапу та написати декілька моделей, які би мали змогу переписати стандартний функціонал Magento, який пов'язаний з посиленням електронної пошти. Для цього, було знайдено класи, які відповідають за цей функціонал:

- а) `Magento\Framework\Mail\Template\TransportBuilder;`
- б) `Magento\Framework\Mail\TransportInterface.`

Саме їх необхідно буде переписати та сконфігурувати своє налаштування на базі атрибутів, які були створені на попередньому етапі.

Таким чином, була сформована діаграма PHP-класів модуля, який розробляється. Вона дає змогу визначити залежності одних PHP-класів від інших (див. рис.2.2).

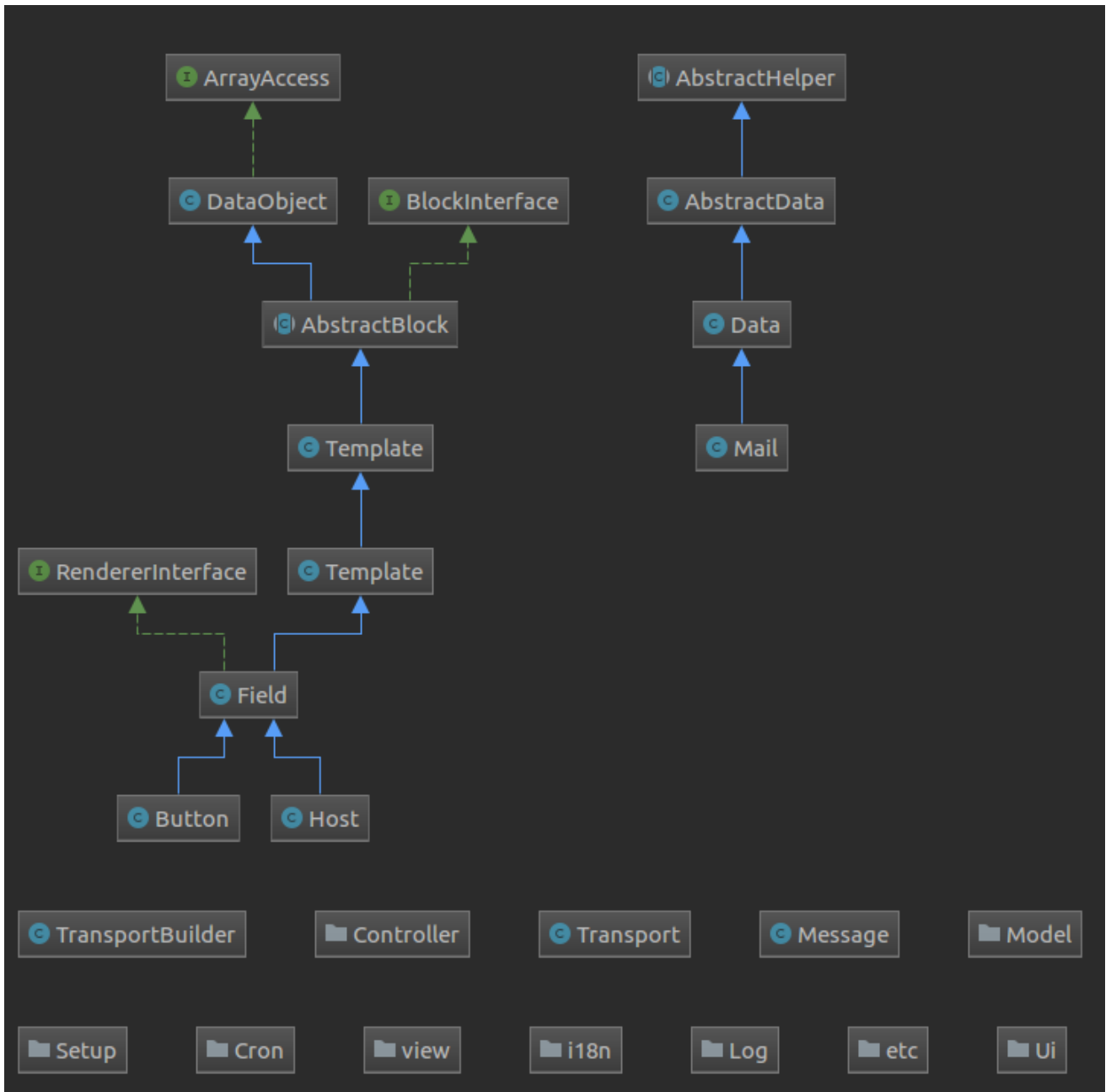


Рисунок 2.2 – Діаграма залежностей PHP-класів

2.3 Перелік технологій для розробки автоматизованого рішення для налаштування простого протоколу пересилання пошти

Останній випуск продукту Magento – Magento 2.3 – містить безліч функціональних виправлень і поліпшень. В якості найкращої практики Magento рекомендує оновлювати установку Magento, щоб ви могли скористатися останніми досягненнями. Ці системні вимоги включають в себе необхідне програмне забезпечення для запуску програми Magento, підтримувані веб-браузери, а також останні вимоги для безпечного з'єднання зі сховищем програмного забезпечення Magento і з PayPal. [8]

Таблиця 2.1 – Системні вимоги для запуску програми Magento

Операційні системи	Linux x86-64
Композитор	Composer потрібно розробникам, які хочуть внести свій вклад в базу коду Magento 2 або розробити розширення для Magento
Веб-сервери	Apache 2.2 или 2.4
	nginx 1.x
PHP версії	Magento 2.3 – 7.1.3+, 7.2.x
	Magento 2.1.2 та вище – 5.6.5, 7.0.4
	Magento 2.1.x – 7.0.2, 7.0.6 до 7.1.0
	Magento 2.1.0 та 2.1.1 – 5.6.x
База даних	MySQL 5.6, 5.7
	Сумісний з MariaDB і Percona
	Magento Enterprise Edition 2.x може використовувати три основні бази даних для забезпечення масштабованості для різних функціональних областей оформлення замовлення, замовлень і даних про продукт.

Продовження таблиці 2.1

SSL	Дійсний сертифікат безпеки потрібно для HTTPS.
	Самозаверяючі SSL-сертифікати не підтримуються.
	Вимоги безпеки транспортного рівня (TLS): – TSL 1.1 або новішої версії; – PayPal і repo.magento.com вимагають TSL 1.1 або більш пізньої версії.
Поштовий сервер	Агент пересилання пошти (MTA) або SMTP-сервер
Зворотний проксі / веб-прискорювач	Лак 4.x или 5.2
Кеш-пам'ять	Redis 3.x
	Memcache 1.4.x
Сховище сесій	memcached остання стабільна версія для зберігання сесій з розширеннями memcache або memcached PHP (остання стабільна версія)
Пошук	Elasticsearch 2.x або 5.2.x
	Apache Solr
Обмін повідомленнями	RabbitMQ (тільки для Enterprise Edition)

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ НАЛАШТУВАННЯ ПРОСТОГО ПРОТОКОЛУ ПЕРЕСИЛАННЯ ПОШТИ ДЛЯ CMS MAGENTO 2

Підготуємо «скелет» модуля, який буде необхідний для того, щоб ідентифікувати наявність нового модуля у системі Magento. Для цього необхідно створити два файли у коді проекту.

Перший файл – registration.php. Він має бути розміщений за таким шляхом app/code/KozakGroup/Smtp/registration.php (рис. 3.1).

```
1 <?php
2
3 use Magento\Framework\Component\ComponentRegistrar;
4
5 ComponentRegistrar::register(
6     type: ComponentRegistrar::MODULE,
7     componentName: 'KozakGroup_Smtp',
8     path: __DIR__
9 );
```

Рисунок 3.1 – Вміст файлу registration.php

Цей файл необхідний для того, щоб повідомити Magento що було зареєстровано новий компонент, а саме – модуль. Це важливо, бо в Magento окрім модулів також існують бібліотеки, теми, мови та налаштування.

Другий файл – module.xml. Він має бути розміщений за таким шляхом app/code/KozakGroup/Smtp/etc/module.xml (рис. 3.2).

```
1 <?xml version="1.0"?>
2
3 <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
5
6     <module name="KozakGroup_Smtp"
7         setup_version="1.0.0"
8         schema_version="1.0.0"/>
9
10 </config>
```

Рисунок 3.2 – Вміст файлу module.xml

Цей файл необхідний для того, щоб дати можливість модулю врегулювати залежність нашого модуля від інших модулів, які також зареєстровані в системі Magento. Також, цей файл дає змогу нашому модулю своєчасно виконувати деякі специфічні файли, які відповідають за вміст та структуру бази даних.

Після того, як ці два файли було вдало створено, необхідно виконати команду в консолі, знаходячись у корені проекту, – «php bin/magento setup:upgrade». Таким чином, ми виконали реконфігурацію усього проекту та зареєстрували наш модуль у системі Magento.

Наступний крок – розробка користувацького інтерфейсу для адміністративної панелі Magento. За допомогою цього інтерфейсу ми зможемо надавати всі необхідні параметри для налаштування Простого Протоколу Пересилання Пошти, які в подальшому будуть використовуватись backend-сторonoю Magento. Достатньо лиш визначити поля в адміністративній панелі Magento 2 та вона самостійно подбає про те, щоб користувач міг зберегти та редагувати дані, які були задані у всіх полях.

В адміністративній панелі буде створено дві вкладки: General Configuration та SMTP Configuration Options.

Перша вкладка буде вміщувати в собі лише одне поле – Enable KozakGroup SMTP. Це поле буде давати можливість адміністратору вмикати/вимикати модуль в загалому. Для цього необхідно створити файл system.xml та розмістити його у тетці – app/code/KozakGroup/Smtp/etc/adminhtml (рис. 3.3).

```
<group id="general" translate="label" type="text" sortOrder="10" showInDefault="1" showInWebsite="1" showInStore="1">
  <label>General Configuration</label>
  <field id="enabled" translate="label" type="select" sortOrder="10" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Enable KozakGroup SMTP</label>
    <source_model>Magento\Config\Model\Config\Source\Yesno</source_model>
    <comment><![CDATA[KozakGroup SMTP is compatible with default Magento SMTP and other email modules.]]></comment>
  </field>
</group>
```

Рисунок 3.3 – Наповнення файлу system.xml для створення вкладки General Configuration

Після того, як було завершено наповнення файлу `system.xml` – необхідно виконати команду «`php bin/magento setup:upgrade`» для того щоб можна було побачити результат (рис. 3.4).



Рисунок 3.4 – Результат роботи коду із `system.xml`

Наступна вкладка – SMTP Configuration Options. Вона має вміщувати перелік таких опцій як:

- а) Host – це поле відповідає за адрес поштового сервісу, який буде відповідати за відправку електронних листів;
- б) Port – це поле, яке буде додано до Host. Існує перелік портів за замовчуванням таких, як 25, 465, or 587. Port 465 (SSL required), Port 587 (TLS required). Детальніше можна дізнатися вже безпосередньо у поштового провайдера;
- в) Protocol – це поле відповідає за протокол передачі електронних листів. Може бути SSL або TLS;
- г) Authentication – це поле відповідає за тип авторизації до поштового сервісу. Може бути Plain, Login або CRAM–MD5;
- д) Username – це поле відповідає за користувацький логін або e-mail;
- е) Password – це поле є паролем до поля Username.

Надалі, щоб створити усі вище-описанні опції треба доповнити вже існуючий файл `system.xml` (рис. 3.5).

Після того, як було завершено наповнення файлу `system.xml` – необхідно виконати команду «`php bin/magento setup:upgrade`» (рис. 3.6).

Після того, як успішно було створено користувацький інтерфейс для адміністративній панелі Magento 2, можна переходити до створення backend-функціоналу, який має безпосереднє відношення до налаштування Простого Протоколу Пересилання Пошти(SMTP).

```

<group id="configuration_option" translate="label" type="text" sortOrder="20" showInDefault="1" showInWebsite="1" showInStore="1">
  <label>SMTP Configuration Options</label>
  <field id="host" translate="label comment" type="text" sortOrder="20" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Host</label>
    <comment>Support Host name and IP Address</comment>
    <frontend_model>KozakGroup\Smtп\Block\Adminhtml\System\Config\Host</frontend_model>
  </field>
  <field id="port" translate="label comment" type="text" sortOrder="30" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Port</label>
    <comment>Default ports: 25, 465, or 587. Port 465 (SSL required), Port 587 (TLS required)</comment>
  </field>
  <field id="protocol" translate="label comment" type="select" sortOrder="35" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Protocol</label>
    <source_model>KozakGroup\Smtп\Model\Config\Source\Protocol</source_model>
    <comment>We recommend use security connections: SSL or TLS. Secure Socket Layer (SSL), Transport Layer Security (TLS) protocols.</comment>
  </field>
  <field id="authentication" translate="label" type="select" sortOrder="40" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Authentication</label>
    <source_model>KozakGroup\Smtп\Model\Config\Source\Authentication</source_model>
  </field>
  <field id="username" translate="label" type="text" sortOrder="60" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Username</label>
  </field>
  <field id="password" translate="label" type="obscure" sortOrder="70" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Password</label>
    <backend_model>Magento\Config\Model\Config\Backend\Encrypted</backend_model>
  </field>
  <field id="return_path_email" translate="label comment" type="text" sortOrder="90" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>Return-Path Email</label>
    <comment>Leave empty to ignore it.</comment>
    <validate>validate-email</validate>
  </field>
</group>

```

Рисунок 3.5 – Наповнення файлу system.xml для створення вкладки SMTP Configuration Options

Рисунок 3.6 – Графічне представлення файлу system.xml в адміністративній панелі Magento 2

Перш за все, необхідно визначити Resource-моделі для полей Host, Port, Protocol. За допомогою цієї моделі ми зможемо запростити налаштування модуля, а саме ми створимо перелік налаштувань для популярних поштових сервісів, таких як Gmail, Mailgun, Mandrill, Elastic Email, Hotmail, Office365 та інші. Для цього необхідно створити Host.php файл та розмістити його в тетці – app/code/KozakGroup/Smtp/Block/Adminhtml/System/Config. В файлі Host.php створюємо функцію «getOptionProvider()» та розмістимо в ній можливі конфігурації для підключення до поштових сервісів (рис. 3.7). Інше наповнення цього файлу буде описано в додатку А.

```

139 public function getOptionProvider()
140 {
141     $options = [
142         'gmail' => [
143             'label' => ('Gmail, GSuite'),
144             'info' => [
145                 'host' => 'smtp.gmail.com',
146                 'port' => '465',
147                 'protocol' => 'ssl'
148             ]
149         ],
150         'amazon' => [
151             'label' => ('Amazon SES'),
152             'info' => [
153                 'us-east-virginia' => [
154                     'label' => ('US East (N. Virginia)'),
155                     'info' => [
156                         'host' => 'email-smtp.us-east-1.amazonaws.com',
157                         'port' => '587',
158                         'protocol' => 'tls'
159                     ]
160                 ],
161                 'us-east-oregon' => [
162                     'label' => ('US West (Oregon)'),
163                     'info' => [
164                         'host' => 'email-smtp.us-west-2.amazonaws.com',
165                         'port' => '587',
166                         'protocol' => 'tls'
167                     ]
168                 ],
169                 'eu-ireland' => [
170                     'label' => ('EU (Ireland)'),
171                     'info' => [
172                         'host' => 'email-smtp.eu-west-1.amazonaws.com',
173                         'port' => '587',
174                         'protocol' => 'tls'
175                     ]
176                 ]
177             ]
178         ],
179         'mailgun' => [
180             'label' => ('Mailgun'),
181             'info' => [
182                 'host' => 'smtp.mailgun.org',
183                 'port' => '587',
184                 'protocol' => 'tls',
185             ]
186         ],

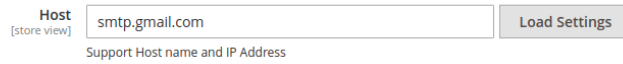
```

Рисунок 3.7 – Приклад конфігураційних опцій для підключення до поштових сервісів

Після того, як було додано опції для підключення до поштових сервісів необхідно виконати команду «php bin/magento setup:upgrade» та перевірити наші зміни у адміністративній панелі Magento 2. Наразі, там з'явилась кнопка

«Load Settings» у вкладці «SMTP Configuration Options» біля поля «Host» (рис. 3.8).

SMTP Configuration Options



Host [store view] Load Settings

Support Host name and IP Address

Рисунок 3.8 – Графічне відтворення кнопки «Load Settings»

Тепер, при натисканні на кнопку «Load Settings», буде з’являтися спливаюче меню з набором опції та можливістю їх вибрати та застосувати (рис. 3.9).

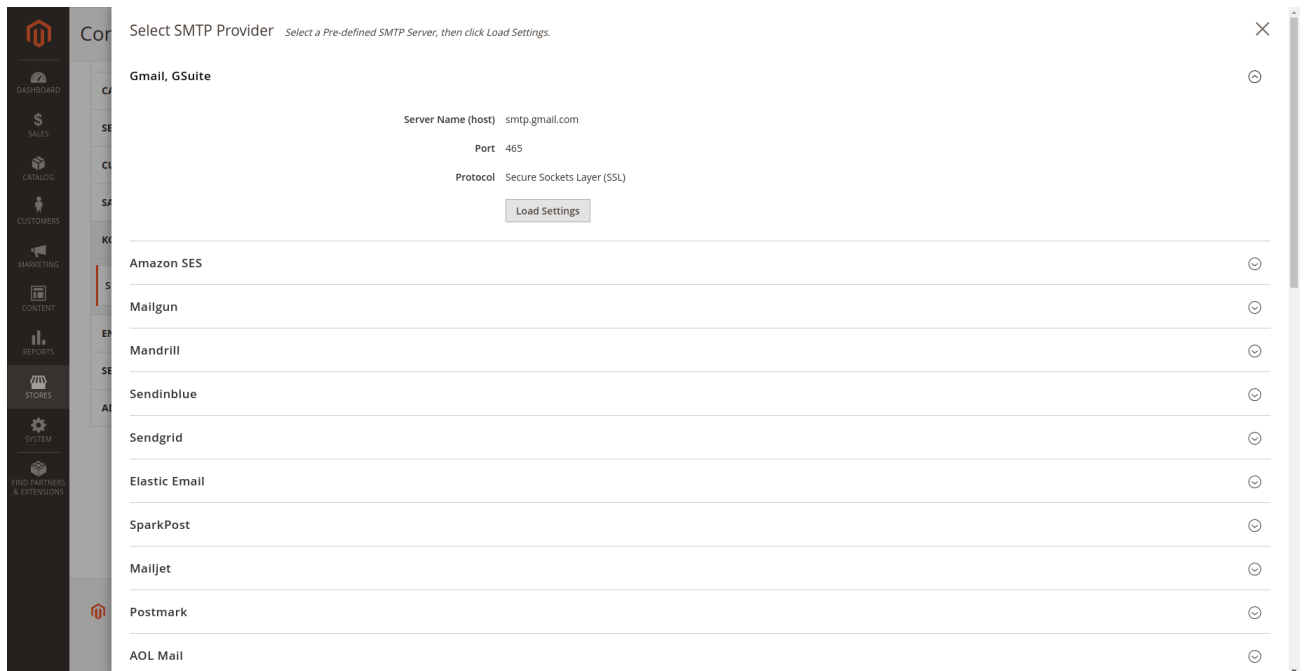


Рисунок 3.9 – Перелік опцій після натискання на кнопку «Load Settings»

На цьому етапі можна переходити для написання класів, які будуть використані для налаштування SMTP на рівні серверу на основі взаємодії користувача з користувацьким інтерфейсом у адміністративній панелі Magento 2. Для цього було досліджено базовий функціонал Magento 2 та знайдено ті класи, які відповідають за відправку електронних листів, а саме:

- a) `Magento\Framework\Mail\Template\TransportBuilder;`

б) Magento\Framework\Mail\TransportInterface.

Ці два класи відповідають за налаштування як наповнення електронного листа, так і за те, які налаштування будуть використанні в момент відправлення листа, а саме Host, Port, Protocol і т.д. Тому було прийняте рішення зробити 2 плагіни та декілька моделей, які б доповнювали функціонал Magento 2. Для того щоб визначити плагін необхідно створити файл di.xml та розмістити його за шляхом – app/code/KozakGroup/Smtp/etc/di.xml (рис. 3.10).

```
<type name="Magento\Framework\Mail\Template\TransportBuilder">
  <plugin name="kozakgroup mail template transport builder"
    type="KozakGroup\Smtp\Mail\Template\TransportBuilder"
    sortOrder="1"
    disabled="false"/>
</type>

<type name="Magento\Framework\Mail\TransportInterface">
  <plugin name="kozakgroup mail transport"
    type="KozakGroup\Smtp\Mail\Transport"
    sortOrder="1"
    disabled="false"/>
</type>
```

Рисунок 3.10 – Вміст файлу di.xml для визначення плагінів

Перед тим як перейти до написання плагінів, необхідно написати модель, яка би допомогла отримати нам усі необхідні конфігурації з адміністративної панелі Magento 2 та давала би змогу сконфігурувати необхідні параметри налаштування SMTP згідно шаблону Magento 2. Потім вже за допомогою плагіну ми будемо викликати нашу модель та підставляти необхідні параметри налаштування SMTP, завдяки яким Magento 2 буде відправляти електронні листи.

Тому, одразу створимо модель Mail.php та розмістимо її у теці – app/code/KozakGroup/Smtp/Mail/Rse/Mail.php (рис. 3.11). Головна функція, що буде розміщена у моделі – «public function getTransport()», буде отримувати дані, які було попередньо заповнені у адміністративній панелі

Magento 2 та приведення до вигляду, який би сприймала Magento 2, а саме створення транспортного об'єкту з конфігураційними параметрами.

```

123  /**
124  * @param $storeId
125  *
126  * @return Zend_Mail_Transport_Smtp | Smtp
127  * @throws Zend_Exception
128  */
129  public function getTransport($storeId)
130  {
131      if ($this->transport === null) {
132          if (!isset($this->smtpOptions[$storeId])) {
133              $configData = $this->smtpHelper->getSmtpConfig('code:', $storeId);
134              $options = [
135                  'host' => isset($configData['host']) ? $configData['host'] : '',
136                  'port' => isset($configData['port']) ? $configData['port'] : '',
137              ];
138              if (isset($configData['authentication']) && $configData['authentication'] !== "") {
139                  $options += [
140                      'auth' => $configData['authentication'],
141                      'username' => isset($configData['username']) ? $configData['username'] : '',
142                      'password' => $this->smtpHelper->getPassword($storeId)
143                  ];
144              }
145              if (isset($configData['protocol']) && $configData['protocol'] !== "") {
146                  $options['ssl'] = $configData['protocol'];
147              }
148              $this->smtpOptions[$storeId] = $options;
149          }
150
151          if (!isset($this->smtpOptions[$storeId]['host']) || !$this->smtpOptions[$storeId]['host']) {
152              throw new Zend_Exception(__('A host is necessary for smtp transport, but none was given'));
153          }
154
155          $options = $this->smtpOptions[$storeId];
156          if (isset($options['auth'])) {
157              $options['connection_class'] = $options['auth'];
158              $options['connection_config'] = [
159                  'username' => $options['username'],
160                  'password' => $options['password']
161              ];
162              unset($options['auth'], $options['username'], $options['password']);
163          }
164          if (isset($options['ssl'])) {
165              $options['connection_config']['ssl'] = $options['ssl'];
166              unset($options['ssl']);
167          }
168          unset($options['type']);
169          $options = new SmtpOptions($options);
170          $this->transport = new Smtp($options);
171      }
172      return $this->transport;
173  }
174

```

Рисунок 3.11 – Приклад основної функції у Mail.php

У попередньому класі був визначений та використаний доступ до класу `KozakGroup\Smtp\Helper\Data`. Цей клас був визначений у змінній `$this->smtpHelper` та його також необхідно створити за шляхом `app/code/KozakGroup/Smtp/Helper/Data.php` (рис. 3.12). Саме за допомогою цього класу ми маємо змогу отримати доступ до «SMTP Configuration Options» значень для яких ми створювали інтерфейс. Він складається з декількох функцій, а саме із «getSmtpConfig(...)» та «getPassword(...)». Здебільшого, вони виконують одну і ту ж функцію – отримання даних із

«SMTP Configuration Options», але друга функція необхідна тому що у ній отримується доступ до поля з паролем, який є зашифрованим.

```

/**
 * @param string $code
 * @param null $storeId
 *
 * @return mixed
 */
public function getSmtпConfig($code = '', $storeId = null)
{
    $code = ($code !== '') ? '/' . $code : '';

    return $this->getModuleConfig( field: self::CONFIG_GROUP_SMTP . $code, $storeId);
}

/**
 * @param null $storeId
 * @param bool $decrypt
 *
 * @return array|mixed|string
 */
public function getPassword($storeId = null, $decrypt = true)
{
    if ($storeId || $storeId = $this->request->getParam( key: 'store')) {
        $password = $this->getSmtпConfig( code: 'password', $storeId);
    } elseif ($websiteCode = $this->request->getParam( key: 'website')) {
        $passwordPath = self::CONFIG_MODULE_PATH . '/' . self::CONFIG_GROUP_SMTP . '/password';
        $password = $this->getConfigValue($passwordPath, $websiteCode, scopeType: ScopeInterface::SCOPE_WEBSITE);
    } else {
        $password = $this->getSmtпConfig( code: 'password');
    }

    if ($decrypt) {
        /** @var EncryptorInterface $encryptor */
        $encryptor = $this->getObject( path: EncryptorInterface::class);

        return $encryptor->decrypt($password);
    }

    return $password;
}

```

Рисунок 3.12 – Приклад основної функції у Data.php

Завдяки Mail-класу, ми маємо функцію «getTransport(...)», яка повертає налаштований транспортний об'єкт, готовий отримати такі параметри електронного листа, як адресат, відправник, повідомлення і т.д. Тому залишилось лише задати ці параметри та здійснити відправку електронного листа. Для цього необхідно створити плагін та мати змогу перехопити функцію, яка відповідає за відправку електронних листів. Як було зазначено до цього, плагін необхідно писати на клас «Magento\Framework\Mail\TransportInterface» та на його основну функцію «sendMessage(...)» (рис. 3.13).

```

/**
 * @param TransportInterface $subject
 * @param Closure $proceed
 *
 * @throws MailException
 * @throws Zend_Exception
 */
public function aroundSendMessage(
    TransportInterface $subject,
    Closure $proceed
) {
    $this->_storeId = $this->registry->registry( key: 'kg_smtp_store_id');
    $message
        = $this->getMessage($subject);
    if ($this->resourceMail->isModuleEnable($this->_storeId) && $message) {
        if ($this->helper->versionCompare('2.2.8')) {
            $message = Message::fromString($message->getRawMessage())->setEncoding( encoding: 'utf-8');
        }
        $message = $this->resourceMail->processMessage($message, $this->_storeId);
        $transport = $this->resourceMail->getTransport($this->_storeId);
        try {
            if (!$this->resourceMail->isDeveloperMode($this->_storeId)) {
                $transport->send($message);
            }
            if ($this->helper->versionCompare('2.2.8')) {
                $messageTmp = $this->getMessage($subject);
                if ($messageTmp && is_object($messageTmp)) {
                    $body = $messageTmp->getBody();
                    if (is_object($body) && $body->isMultiPart()) {
                        $message->setBody($body->getPartContent("0"));
                    }
                }
            }
            $this->emailLog($message);
        } catch (Exception $e) {
            $this->emailLog($message, status: false);
            throw new MailException(new Phrase($e->getMessage()), $e);
        }
    } else {
        $proceed();
    }
}

```

Рисунок 3.13 – Приклад основної функції у Transport.php

Плагін перехоплює виконання основної функції та підставляє до неї транспортний поштовий об'єкт, який був сконфігурований до цього. Також було передбачено можливість того, якщо розроблений модуль буде відключено у налаштуваннях, то цей плагін буде виконувати стандартну логіку відсилки електронного листа, яка базується на налаштуванні «Простого Протоколу Пересилання Пошти» на серверному рівні.

Для перевірки роботи модуля був налаштований модуль за допомогою користувацького інтерфейса та відправлений тестовий електронний лист на власну електронну адресу (рис. 3.14).

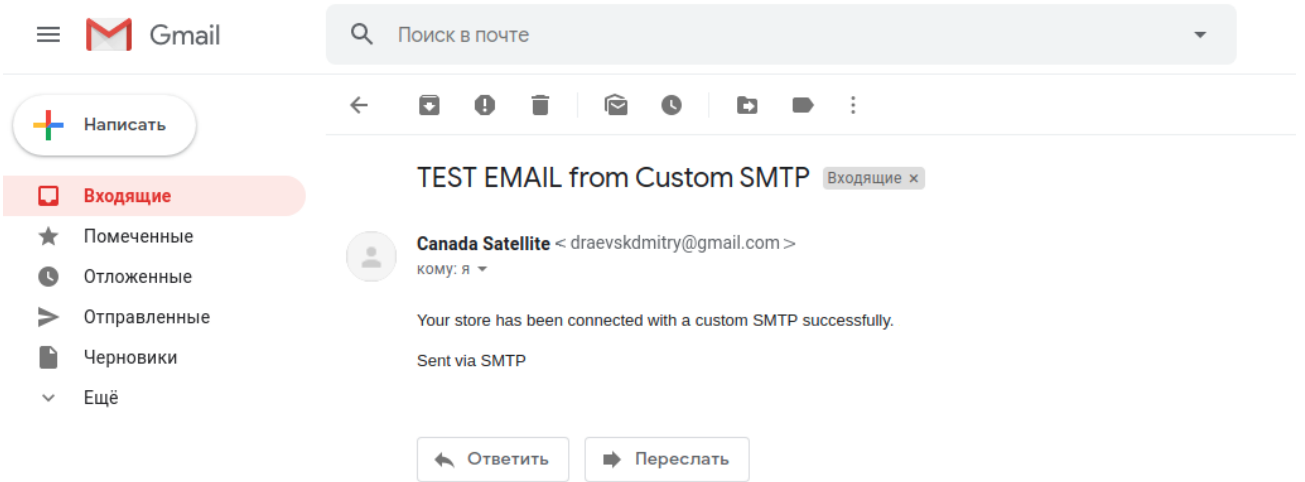


Рисунок 3.14 – Пример работы модуля

ВИСНОВКИ

Запропонований модуль налаштування поштового транспортного протоколу для CMS Magento 2 може бути використаний у комерційній діяльності.

CMS Magento 2 прийнято вважати платформою для створення складних проектів електронної комерції з великим об'ємом фінансування та широким кругом фахівців для налаштування, підтримання та покращення системи. Тому розробка модулів для налаштування різноманітних функціональних особливостей системи є дуже важливим, бо це дає змогу розвиватися не тільки великим та сформованим компаніям, а й новачкам та невеликим проектам, які би хотіли мати можливість стати професіоналами у своїй справі в умовах обмеженості ресурсів.

У модуля налаштування поштового транспортного протоколу для CMS Magento 2 також можуть бути перспективи у покращенні та розширенні існуючого функціоналу, а саме додавання можливості зворотного завантаження електронних листів із поштової системи у систему Magento 2. Таким чином, адміністратору сайту не потрібно було би витрачати час на перевірку електронної пошти сайту, бо усі б листи, що надходять на електронну пошту сайту – дублювались би у адміністративну панель Magento 2.

Таким чином, розглянуто основні алгоритми налаштування поштового транспортного протоколу Postfix у UNIX-системах, розроблено та протестовано модуль налаштування поштового транспортного протоколу для CMS Magento 2.

ПЕРЕЛІК ПОСИЛАНЬ

1. SMTP. Вікіпедія. URL : <https://uk.wikipedia.org/wiki/SMTP> (дата звернення: 12.07.2019)
2. Email Services, Postfix. Ubuntu Documentation. URL : <https://help.ubuntu.com/lts/serverguide/postfix.html> (дата звернення: 18.08.2019)
3. Magento. Вікіпедія. URL : <https://uk.wikipedia.org/wiki/Magento> (дата звернення: 18.08.2019)
4. Allan MacGregor. Magento PHP Developer's Guide. Birmingham. UK : Packt Publishing Ltd, 2015. 256 p.
5. Bret Williams, Jonathan Bownds. Mastering Magento 2 – Second Edition. Birmingham. UK : Packt Publishing Ltd, 2016. 340 p.
6. Fernando J. Miguel. Magento 2 Development Essentials. Birmingham. UK : Packt Publishing Ltd, 2016. 178 p.
7. Branko Ajzele. Getting Started with Magento Module Development. Birmingham. UK : Packt Publishing Ltd, 2013. 128 p.
8. Magento 2.3 Technology stack requirements. Magento DevDocs. URL : <https://devdocs.magento.com/guides/v2.3/install-gde/system-requirements-tech.html> (дата звернення: 04.09.2019)

ДОДАТОК А

Програмний код

```
1) app/code/KozakGroup/Smtp/Helper/Data.php
<?php
namespace KozakGroup\Smtp\Helper;

use Magento\Framework\Encryption\EncryptorInterface;
use Magento\Framework\Exception\LocalizedException;
use Magento\Store\Model\ScopeInterface;

/**
 * Class Data
 * @package KozakGroup\Smtp\Helper
 */
class Data extends Mail
{
    /**
     * Const for storing path to Stores Configuration
     */
    const CONFIG_MODULE_PATH = 'smtp';

    /**
     * Const for storing path to Stores Configuration
     */
    const CONFIG_GROUP_SMTP = 'configuration_option';

    /**
     * @param string $code
     * @param null $storeId
     *
     * @return mixed
     */
    public function getSmtpConfig($code = '', $storeId = null)
    {
```

```

$code = ($code !== '') ? '/' . $code : '';

return $this->getModuleConfig(self::CONFIG_GROUP_SMTP . $code,
$storeId);
}

/**
 * @param null $storeId
 * @param bool $decrypt
 *
 * @return array|mixed|string
 */
public function getPassword($storeId = null, $decrypt = true)
{
    if ($storeId || $storeId = $this->_request->getParam('store')) {
        $password = $this->getSmtplibConfig('password', $storeId);
    } elseif ($websiteCode = $this->_request->getParam('website')) {
        $passwordPath = self::CONFIG_MODULE_PATH . '/' .
self::CONFIG_GROUP_SMTP . '/password';
        $password = $this->getConfigValue($passwordPath, $websiteCode,
ScopeInterface::SCOPE_WEBSITE);
    } else {
        $password = $this->getSmtplibConfig('password');
    }

    if ($decrypt) {
        /** @var EncryptorInterface $encryptor */
        $encryptor = $this->getObject(EncryptorInterface::class);

        return $encryptor->decrypt($password);
    }

    return $password;
}

/**
 * @return int
 * @throws LocalizedException
 */

```

```

public function getScopeId()
{
    $scope = $this->_request->getParam(ScopeInterface::SCOPE_STORE) ?:
$this->storeManager->getStore()->getId();

    if ($website = $this->_request-
>getParam(ScopeInterface::SCOPE_WEBSITE)) {
        $scope = $this->storeManager->getWebsite($website)->getDefaultStore()-
>getId();
    }

    return $scope;
}
}

```

2) app/code/KozakGroup/SmtP/Mail/Rse/Mail.php

```

<?php
namespace KozakGroup\SmtP\Mail\Rse;

use KozakGroup\SmtP\Helper\Data;
use Zend\Mail\Message;
use Zend\Mail\Transport\SmtP;
use Zend\Mail\Transport\SmtPOptions;
use Zend_Exception;
use Zend_Mail_Transport_SmtP;

/**
 * Class Mail
 * @package KozakGroup\SmtP\Application\Rse
 */
class Mail
{
    /**
     * @var Data
     */
    protected $smtPHelper;

    /**

```

```
* @var array Is module enable by store
*/
protected $_moduleEnable = [];

/**
 * @var array is developer mode
 */
protected $_developerMode = [];

/**
 * @var array is enable email log
 */
protected $_emailLog = [];

/**
 * @var string message body email
 */
protected $_message;

/**
 * @var array option by storeid
 */
protected $_smtpOptions = [];

/**
 * @var array
 */
protected $_returnPath = [];

/**
 * @var Zend_Mail_Transport_Smtp
 */
protected $_transport;

/**
 * @var array
 */
protected $_fromByStore = [];
```

```

/**
 * Mail constructor.
 *
 * @param Data $helper
 * @param null $options
 */
public function __construct(Data $helper)
{
    $this->smtpHelper = $helper;
}

/**
 * @param $storeId
 * @param array $options
 *
 * @return $this
 */
public function setSmtOptions($storeId, $options = [])
{
    if (isset($options['return_path'])) {
        $this->_returnPath[$storeId] = $options['return_path'];
        unset($options['return_path']);
    }

    if (isset($options['ignore_log']) && $options['ignore_log']) {
        $this->_emailLog[$storeId] = false;
        unset($options['ignore_log']);
    }

    if (isset($options['force_sent']) && $options['force_sent']) {
        $this->_moduleEnable[$storeId] = true;
        unset($options['force_sent']);
    }

    if (count($options)) {
        $this->_smtpOptions[$storeId] = $options;
    }

    return $this;
}

```



```

}

/**
 * @param $storeId
 *
 * @return Zend_Mail_Transport_Smtp | Smtp
 * @throws Zend_Exception
 */
public function getTransport($storeId)
{
    if ($this->_transport === null) {
        if (!isset($this->_smtpOptions[$storeId])) {
            $configData = $this->smtpHelper->getSmtpConfig('', $storeId);
            $options = [
                'host' => isset($configData['host']) ? $configData['host'] : '',
                'port' => isset($configData['port']) ? $configData['port'] : ''
            ];

            if (isset($configData['authentication']) &&
                $configData['authentication'] !== '') {
                $options += [
                    'auth' => $configData['authentication'],
                    'username' => isset($configData['username']) ?
                $configData['username'] : '',
                    'password' => $this->smtpHelper->getPassword($storeId)
                ];
            }

            if (isset($configData['protocol']) && $configData['protocol'] !== '') {
                $options['ssl'] = $configData['protocol'];
            }

            $this->_smtpOptions[$storeId] = $options;
        }

        if (!isset($this->_smtpOptions[$storeId]['host']) || !$this->
            _smtpOptions[$storeId]['host']) {
            throw new Zend_Exception(__('A host is necessary for smtp transport,
                but none was given'));
        }
    }
}

```

```

    }

    if ($this->smtpHelper->versionCompare('2.2.8')) {
        $options = $this->_smtpOptions[$storeId];
        if (isset($options['auth'])) {
            $options['connection_class'] = $options['auth'];
            $options['connection_config'] = [
                'username' => $options['username'],
                'password' => $options['password']
            ];
            unset($options['auth'], $options['username'], $options['password']);
        }
        if (isset($options['ssl'])) {
            $options['connection_config']['ssl'] = $options['ssl'];
            unset($options['ssl']);
        }
        unset($options['type']);

        $options = new SmtOptions($options);

        $this->_transport = new Smt($options);
    } else {
        $this->_transport = new Zend_Mail_Transport_Smt(
            $this->_smtpOptions[$storeId]['host'],
            $this->_smtpOptions[$storeId]
        );
    }
}

return $this->_transport;
}

/**
 * @param $message
 * @param $storeId
 *
 * @return mixed
 */
public function processMessage($message, $storeId)

```

```

{
    if (!isset($this->_returnPath[$storeId])) {
        $this->_returnPath[$storeId] = $this->smtpHelper-
>getSmtpConfig('return_path_email', $storeId);
    }

    if ($this->_returnPath[$storeId]) {
        if ($this->smtpHelper->versionCompare('2.2.8')) {
            $message->getHeaders()->addHeaders(["Return-Path" => $this-
>_returnPath[$storeId]);
        } elseif (method_exists($message, 'setReturnPath')) {
            $message->setReturnPath($this->_returnPath[$storeId]);
        }
    }

    if (!empty($this->_fromByStore) &&
        ((is_array($message->getHeaders()) && !array_key_exists("From",
$message->getHeaders())) ||
        ($message instanceof Message && !$message->getFrom()->count()))
    ) {
        $message->setFrom($this->_fromByStore['email'], $this-
>_fromByStore['name']);
    }

    return $message;
}

/**
 * @param $email
 * @param $name
 *
 * @return $this
 */
public function setFromByStore($email, $name)
{
    $this->_fromByStore = [
        'email' => $email,
        'name' => $name
    ];
}

```

```

        return $this;
    }
}
3) app/code/KozakGroup/Smtp/Mail/Transport.php
<?php

```

```

namespace KozakGroup\Smtp\Mail;

use Closure;
use Exception;
use Magento\Framework\Exception\MailException;
use Magento\Framework\Mail\TransportInterface;
use Magento\Framework\Phrase;
use Magento\Framework\Registry;
use KozakGroup\Smtp\Helper\Data;
use KozakGroup\Smtp\Mail\Rse\Mail;
use KozakGroup\Smtp\Model\Log;
use KozakGroup\Smtp\Model\LogFactory;
use Psr\Log\LoggerInterface;
use ReflectionClass;
use Zend\Mail\Message;
use Zend_Exception;

/**
 * Class Transport
 * @package KozakGroup\Smtp\Mail
 */
class Transport
{
    /**
     * @var int Store Id
     */
    protected $_storeId;

    /**
     * @var Mail
     */
    protected $resourceMail;

```

```
/**
 * @var LogFactory
 */
protected $logFactory;

/**
 * @var Registry $registry
 */
protected $registry;

/**
 * @var Data
 */
protected $helper;

/**
 * @var LoggerInterface
 */
protected $logger;

/**
 * Transport constructor.
 *
 * @param Mail $resourceMail
 * @param LogFactory $logFactory
 * @param Registry $registry
 * @param Data $helper
 * @param LoggerInterface $logger
 */
public function __construct(
    Mail $resourceMail,
    LogFactory $logFactory,
    Registry $registry,
    Data $helper,
    LoggerInterface $logger
) {
    $this->resourceMail = $resourceMail;
    $this->logFactory = $logFactory;
}
```

```

    $this->registry    = $registry;
    $this->helper      = $helper;
    $this->logger      = $logger;
}

/**
 * @param TransportInterface $subject
 * @param Closure $proceed
 *
 * @throws MailException
 * @throws Zend_Exception
 */
public function aroundSendMessage(
    TransportInterface $subject,
    Closure $proceed
) {
    $this->_storeId = $this->registry->registry('kg_smtp_store_id');
    $message       = $this->getMessage($subject);
    if ($this->resourceMail->isModuleEnable($this->_storeId) && $message) {
        if ($this->helper->versionCompare('2.2.8')) {
            $message = Message::fromString($message->getRawMessage())-
>setEncoding('utf-8');
        }
        $message = $this->resourceMail->processMessage($message, $this-
>_storeId);
        $transport = $this->resourceMail->getTransport($this->_storeId);
        try {
            if (!$this->resourceMail->isDeveloperMode($this->_storeId)) {
                $transport->send($message);
            }
            if ($this->helper->versionCompare('2.2.8')) {
                $messageTmp = $this->getMessage($subject);
                if ($messageTmp && is_object($messageTmp)) {
                    $body = $messageTmp->getBody();
                    if (is_object($body) && $body->isMultiPart()) {
                        $message->setBody($body->getPartContent("0"));
                    }
                }
            }
        }
    }
}

```

```

        $this->emailLog($message);
    } catch (Exception $e) {
        $this->emailLog($message, false);
        throw new MailException(new Phrase($e->getMessage()), $e);
    }
} else {
    $proceed();
}
}

/**
 * @param $transport
 *
 * @return mixed|null
 */
protected function getMessage($transport)
{
    if ($this->helper->versionCompare('2.2.0')) {
        return $transport->getMessage();
    }

    try {
        $reflectionClass = new ReflectionClass($transport);
        $message        = $reflectionClass->getProperty('_message');
    } catch (Exception $e) {
        return null;
    }

    $message->setAccessible(true);

    return $message->getValue($transport);
}
}
4) app/code/KozakGroup/Smtplib/etc/adminhtml/system.xml
<?xml version="1.0"?>

```



```
<source_model>KozakGroup\Smtplib\Model\Config\Source\Protocol</source_model>
```

```
</field>
```

```
<field id="authentication" translate="label" type="select"
sortOrder="40" showInDefault="1" showInWebsite="1" showInStore="1">
```

```
<label>Authentication</label>
```

```
<source_model>KozakGroup\Smtplib\Model\Config\Source\Authentication</source_model>
```

```
</field>
```

```
<field id="username" translate="label" type="text" sortOrder="60"
showInDefault="1" showInWebsite="1" showInStore="1">
```

```
<label>Username</label>
```

```
</field>
```

```
<field id="password" translate="label" type="obscure" sortOrder="70"
showInDefault="1" showInWebsite="1" showInStore="1">
```

```
<label>Password</label>
```

```
<backend_model>Magento\Config\Model\Config\Backend\Encrypted</backend_model>
```

```
</field>
```

```
</group>
```

```
</section>
```

```
</system>
```

```
</config>
```

```
5) app/code/KozakGroup/Smtplib/etc/adminhtml/menu.xml
```

```
<?xml version="1.0"?>
```

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Backend:etc/menu.xsd">
```

```
<menu>
```

```
<add id="KozakGroup_Smtplib::smtp" title="SMTP"
module="KozakGroup_Smtplib" sortOrder="45"
resource="KozakGroup_Smtplib::smtp" parent="Magento_Backend::stores"/>
```

```
<add id="KozakGroup_Smtplib::configuration" title="Configuration"
module="KozakGroup_Smtplib" sortOrder="100"
action="adminhtml/system_config/edit/section/smtp"
```

```
resource="KozakGroup_Smtp::configuration"
parent="KozakGroup_Smtp::smtp"/>
```

```
</menu>
```

```
</config>
```

```
6) app/code/KozakGroup/Smtp/etc/acl.xml
```

```
<?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="KozakGroup_Core::menu">
```

```
<resource id="KozakGroup_Smtp::smtp" title="KozakGroup Smtp"
translate="title" sortOrder="45">
```

```
<resource id="KozakGroup_Smtp::log" title="Emails Log"
translate="title" sortOrder="10"/>
```

```
</resource>
```

```
</resource>
```

```
<resource id="Magento_Backend::stores">
```

```
<resource id="Magento_Backend::stores_settings">
```

```
<resource id="Magento_Config::config">
```

```
<resource id="KozakGroup_Smtp::configuration"
title="KozakGroup Smtp"/>
```

```
</resource>
```

```
</resource>
```

```
</resource>
```

```
</resource>
```

```
</resources>
```

```
</acl>
```

```
</config>
```

```
7) app/code/KozakGroup/Smtp/etc/di.xml
```

```
<?xml version="1.0" ?>
```

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/c
onfig.xsd">
```

```
<type name="Magento\Framework\Mail\Template\TransportBuilder">
```

```
<plugin name="kozakgroup_mail_template_transport_builder"
```

```
type="KozakGroup\Smtp\Mail\Template\TransportBuilder" sortOrder="1"
disabled="false"/>
```

```

    </type>
    <type name="Magento\Framework\Mail\TransportInterface">
        <plugin name="kozakgroup_mail_transport"
type="KozakGroup\Smtp\Mail\Transport" sortOrder="1" disabled="false"/>
    </type>
</config>

```

8) app/code/KozakGroup/Smtp/etc/module.xml

```

<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.x
sd">
    <module name="KozakGroup_Smtp" setup_version="1.0.0"/>
</config>

```

9) app/code/KozakGroup/Smtp/registration.php

```

<?php

```

```

use Magento\Framework\Component\ComponentRegistrar;

```

```

ComponentRegistrar::register(
    ComponentRegistrar::MODULE,
    'KozakGroup_Smtp',
    __DIR__
);

```