

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

Водзинський Євген Юрійович

**Програмне забезпечення системи тестування текстових
полів web-застосунків**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітній ступінь: бакалавр

Науковий керівник:

Бісюк Віктор Анатолійович _____

викладач

(підпис)

(дата)

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

(_____) **Смірнов О.А.**
(підпис) ПІБ

_____ 2021 р.

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітньо-кваліфікаційний рівень бакалавр
спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
О.А.Смірнов
“ ” 20 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Водзинського Євгена Юрійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи тестування
Текстових полів web-застосунків

керівник роботи Бісюк Віктор Анатолійович, викладач

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

затверджені наказом закладу вищої освіти від “ ” 2020 року №

2. Строк подання студентом роботи 22.05.2021 р.

3. Мета та завдання випускної кваліфікаційної роботи розробка програмного
забезпечення системи тестування текстових полів web-застосунків

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

1. Призначення та область використання.

2.Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи головного модулю 1 аркуш

Блок-схема алгоритму роботи тестової процедури 1 аркуш

АНОТАЦІЯ

Водзинський Є.Ю. Програмне забезпечення системи тестування текстових полів web-застосунків. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький 2021.

В представленій кваліфікаційній бакалаврській роботі розроблено програмне забезпечення системи тестування текстових полів web-застосунків.

В роботі досліджено і описано різні методи і способи тестування web-застосунків.

Проаналізовано ряд останніх інтернет та літературних джерел з даної тематики.

Для досягнення мети кваліфікаційної бакалаврської роботи виконано наступні завдання:

- переглянуто і проаналізовано аналогічні існуючі системи.
- обрано оптимальні засоби розробки програмного забезпечення.
- запропоновано структурну, функціональну схеми та діаграму процесів.
- розроблено програмне забезпечення системи кібербезпеки для виявлення вразливостей web-застосунків.

Розроблено алгоритми роботи системи які відображені на відповідних блок-схемах.

Код програми написано на мові Java з використанням бібліотек Selenium WebDriver.

Ключові слова: web-застосунки, якість продукту, вразливості системи.

ANNOTATION

**Vodzynski Y.Y. Software for testing text fields of web-applications.
123 Computer Engineering. Central Ukrainian National Technical University.
Kropyvnytskyi 2021.**

In the presented qualifying bachelor's thesis the software for testing text fields of web-applications is developed.

The paper explores and describes various methods and ways to testing web-applications.

A number of recent Internet and literature sources on this topic have been analyzed.

To achieve the goal of qualifying bachelor's work, the following tasks were performed:

- similar existing systems are reviewed and analyzed.
- the optimal means of software development are selected.
- the structural, functional scheme and the diagram of processes are offered.
- developed cybersecurity software to detect vulnerabilities in web applications.

Algorithms of system operation which are reflected on the corresponding block diagrams are developed.

The program code is written in Java with uses Selenium WebDriver.

Keywords: web-applications, product quality, system vulnerabilities.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	4
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	11
2.3 Розгорнута постановка завдання	16
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	18
3.1 Опис особливостей тестування web-застосунків.....	18
3.2 Способи тестування текстових полів	23
3.3 Розробка структурної схеми	25
3.4 Розробка функціональної схеми	26
3.5 Розробка діаграми взаємодії процесів.....	27
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ	29
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	29
4.2 Захист розробленого програмного забезпечення	38
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	40
6 ОСНОВНІ ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43

					КБР-123.21.0023.00.00.ПЗ			
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>	Програмне забезпечення системи тестування текстових полів web-застосунків	<i>Лім.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розробив</i>	Водзинський Є.Ю					<i>Б</i>	<i>1</i>	
<i>Перевірів</i>	Бісюк В.А.							
<i>Н. Контр.</i>	Гермак В.С.							
<i>Зате.</i>	Смірнов О.А.							
						ЦНТУ КІ-18-3ск		

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І
ТЕРМІНІВ**

ІС- інформаційні системи

ІІ – інформаційне повідомлення

ПК – персональний комп'ютер

ОС - операційна система

QA – (quality assurance) забезпечення якості

QC – (quality control) контроль якості

НСД – несанкціонований доступ

КБР-123.21.0023.00.00.ПЗ

Арк.

2

Вим.	Арк.	№ документа	Підпис	Дата
------	------	-------------	--------	------

Саме тому, виявлення вразливостей безпеки web-застосунку не менш важлива, ніж реалізація його основних функцій. Виявлення вразливостей на етапі проектування та розробки web-застосунку дозволяє значно полегшити роботу над його системою захисту.

Крім навмисного НСД можливий також вплив людського фактору. Тобто ненавмисне втручання або помилкове внесення некоректних даних через текстові поля Web-застосунку. Це може значно вплинути на систему, якщо не передбачено верифікацію даних або навіть привести до її краху.

КБР-123.21.0023.00.00.ПЗ

Арк.

4

Вим.	Арк.	№ документа	Підпис	Дата

- Персоналізовані веб-системи різного спрямування, які надають комплекс послуг свої користувачам

Створення якісного сайту має починатися з максимально чіткого визначення мети, завдань сайту і детального проектування. Протягом усього життєвого циклу необхідно постійно аналізувати ситуацію, вивчати існуючі технології, вислуховувати думки користувачів і, при необхідності, впроваджувати і вносити корективи.

Майже кожен web-застосунок має ряд діалогових форм з текстовими полями для введення інформації у текстовому вигляді. Наприклад форма реєстрації користувача або форма для проведення онлайн оплати.

Текстове поле в додатку здається такою звичайною справою, однак це одна з найбільш важливих речей, яку необхідно завжди тестувати. Чому? Тому що текстові поля дають доступ до серверної частини web-застосунків і його бази даних.

Валідація текстового поля - це те, що запобігає появі в базі некоректних або шкідливих даних. Ці дані можуть викликати різноманітні проблеми для користувачів і розробників. Валідація також запобігає атакам міжсайтового скриптинга і SQL-ін'єкціям.

Попередня валідація даних може виконуватись на стороні клієнта (front-end validation), обов'язково повинна бути реалізована валідація на стороні сервера та при запитах запису в БД.

Тестування та валідація текстових полів – рекомендовано завжди автоматизувати, тому що потрібно проводити невелику кількість операцій, але з достатньо великою кількістю тестових даних.

Отже програмне забезпечення бакалаврської роботи призначене для контролю та підвищення якості різного роду web-застосунків та систем – в навчальних закладах, на підприємствах, в офісах, банківських мережах та інш.

Приведені дослідження та аналіз літературних джерел прискорить засвоєння та закріплення навчального матеріалу з вказаної тематики. Крім того

					КБР-123.21.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

надасть можливість розроблювати більш оптимальні алгоритми програм при розробці власних програмних систем відповідного призначення.

Використані в ході роботи над проектом алгоритми та прийоми програмування, а також супроводжений докладними коментарями програмний код і пояснювальна записка будуть корисні починаючим програмістам і спеціалістам в сфері QA і можуть бути використані при підготовці студентів та фахівців з напрямку комп'ютерна інженерія.

КБР-123.21.0023.00.00.ПЗ

Арк.

7

Вим.	Арк.	№ документа	Підпис	Дата
------	------	-------------	--------	------

- розробити структурну схему;
- розробити функціональну схему системи;
- проаналізувати засоби та способи тестування текстових полів web-застосунків;
- розробити алгоритми роботи програмного забезпечення;
- розробити програмне забезпечення, яке б реалізувало наступні функції і мало такі властивості:
 - а) виконує тестування текстових полів web-додатку в автоматичному режимі на валідні та невалідні значення;
 - б) виводить інформацію про статус пройдених тестів;
 - в) програмне забезпечення повинно мати інтуїтивно зрозумілий інтерфейс і супроводжуватись технічним описом (довідкою) ;
 - г) програмне забезпечення повинно бути розроблено на мові програмування Java, використовувати всі можливості сучасних середовищ розробки;
 - ж) працювати на сучасних апаратних платформах під керуванням ОС сімейства Windows та Linux.

Вим.	Арк.	№ документа	Підпис	Дата

Це далеко не єдиний критерій, пов'язаний з сервером, який характеризує якість сайту. Частково в рішенні даного питання Вам може допомогти хороший хостинг.

Швидкість завантаження. На мій погляд, це друге, що «бачить» користувач. На сайт користувач може потрапляти з різних джерел (пошукових систем, закладок в браузері, посилань з інших сайтів, посилань в поштових листах і т. Д.). Незалежно від джерела переходу, час, після якого користувач підтвердить своє бажання перейти на Ваш сайт (натиснувши на посилання або натиснувши на клавішу Enter в адресному рядку, куди він ввів URL Вашого ресурсу), до появи перших елементів сторінки повинно бути мінімальним. Сервер повинен максимально швидко зреагувати на запит і видати відповідь.

Безумовно, кожна сторінка сайту повинна повністю завантажуватися максимально швидко (наприклад, до 2 секунд). Але її початкова поява (початок відтворення або рендеринг) має бути практично моментальним. Щоб користувач бачив, що сторінка є. Якщо сайт вантажиться вкрай повільно, то користувач може просто не дочекатися і піти. Швидкість завантаження важлива для користувачів і пошукових систем. Тому, досягнувши мінімального часу завантаження сторінки, Ви підвищите загальну зручність користування сайту (для відвідувачів) і можете знизити показник відмов (не всі чекатимуть, коли ж завантажиться Ваша сторінка).

Google і, можливо, інші пошукові системи, враховують при ранжируванні швидкість завантаження сайту як один з показників. До того ж менший показник відмов може служити позитивним сигналом для пошукових систем і додасть Вашій сторінці і сайту в цілому перевагу при ранжируванні в пошуковій видачі. Частково в рішенні даного питання Вам може допомогти якісний хостинг і оптимізація сайту.

Безпека сайту - сервер і сайт повинні бути максимально захищені від несанкціонованого доступу зломисників і не містити шкідливого коду. До

					КБР-123.21.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		19

пошукової оптимізації та популяризації інтернет-проекту в цілому. Вважаю, що якість сайту дуже сильно залежить від цінності його вмісту.

Корисний вміст дуже важливий параметр «якості» сайту, але він не піддається тестуванню, це суб'єктивна оцінка користувачів. Тому під час роботи над моїм програмним забезпеченням цей аспект не враховувався.

Зручний інтерфейс. Під інтерфейсом будемо розуміти не тільки дизайн в сенсі графічного (естетичного) оформлення, а й навігацію, пошук, кнопки соціальних мереж, коментарі і т. д. Для спрощення віднесемо сюди і структуру сайту, оформлення вмісту, структуру URL адрес, пр.

Під зручним інтерфейсом я розумію те, що всі елементи знаходяться в таких місцях, які зручні цільовим користувачеві, цими елементами легко користуватися, вони не викликають двозначності і служать корисним доповненням до вмісту сайту.

Зручний інтерфейс - це інтерфейс, використання якого не викликає ускладнень у більшості цільових користувачів сайту на різних пристроях і в різних браузерях.

Чому це важливо? Після того, як користувачу відобразилася сторінка з вмістом, він приступає до його вивчення. Наприклад, це «типова» сторінка сайту з текстом, зображенням і відео. Текст повинен легко читатися, зображення і відео повинні гармонійно його доповнювати, збільшення зображень, запуск відео і завантаження файлів повинні бути очевидними і не викликати сумнівів і труднощів. Крім цього, переглянувши вміст на одній сторінці, користувач, можливо, захоче його прокоментувати, задати питання автору, додати в обране, поділитися з друзями по електронній пошті або в соціальних сервісах, роздрукувати і т. д. У всьому цьому інтерфейс сайту повинен допомагати максимально швидко і легко.

Що дає зручний інтерфейс? Він підвищує задоволеність Ваших користувачів, їх лояльність, зменшення показника відмов, збільшення часу перебування на сайті, збільшення глибини перегляду і подібне. Хороші

					КБР-123.21.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		22

- введення декількох пробілів;
- заповніть поле максимальною можливою кількістю символів
- введіть всі спецсимволи клавіатури. Якщо ви отримаєте помилку, спробуйте розібратися, який символ або символи її викликають.

- введіть символи, що не належать до ASCII, і емоджі. Якщо ви отримаєте помилку, спробуйте розібратися, який символ або символи її викликають.

- спробуйте міжсайтовий скриптинг - введіть такий скрипт: `<script>alert ("I hacked this!") </ Script>`. Якщо введенні скрипта при з'явиться цей надпис у спливаючому вікні - значить, поле вразливе для XSS-атаки.

- спробуйте ввести SQL-ін'єкцію, наприклад, `FOO '); DROP TABLE USERS;` (Не робіть цього на базі даних сайту який вже працює на продакшен!).

Наступні перевірки доцільно проводити якщо, ми щось знаємо про те, що має бути введено в це поле, і які обмеження для даних:

- спробуйте ввести значення з типом даних, відмінним від очікуваного - наприклад, якщо поле очікує введення вартості, спробуйте ввести текст або дату.

- якщо поле очікує рядок, спробуйте ввести рядок на 1 символ коротше, ніж потрібно, на 1 символ довше, ніж потрібно, мінімально можливу кількість символів, максимальну їх кількість, і кількість, яка перевищує максимум.

- якщо поле очікує числа, спробуйте ввести максимум, мінімум, значення вище максимуму і нижче мінімуму, і значення, яке вдвічі перевищує максимум.

- якщо поле очікує цілого числа, спробуйте ввести десятковий дріб.

- якщо поле очікує числа з плаваючою точкою, спробуйте ввести значення з двома комами і значення, що починається з коми.

- якщо поле очікує вартості, спробуйте ввести значення з більш ніж двома знаками після коми.

					<i>КБР-123.21.0023.00.00.ПЗ</i>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		24

- в) Модуль підключення та запуску тестових модулів;
- г) Модуль тестування форми логіну та реєстрації користувача;
- д) Модуль тестування полів форми створення заказу;
- е) Модуль тестування полів форми проведення оплати;
- ж) Модуль зберігання результатів тестів.

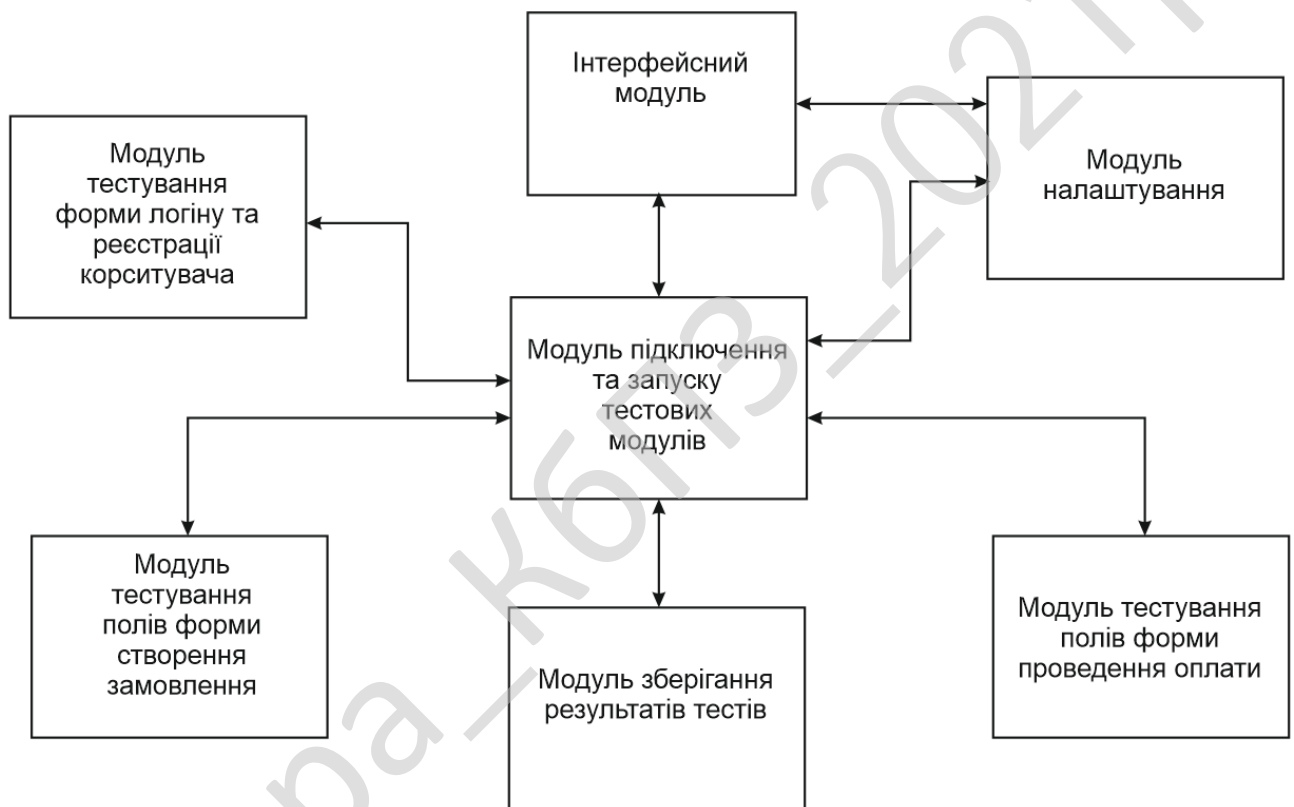


Рисунок 3.1 - Структурна схема роботи системи

3.4 Розробка функціональної схеми

Функціональну схему розробленого програмного забезпечення та системних складових відображено на рис 3.2.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ

До початку безпосереднього написання програмного коду, було розроблено блок-схеми алгоритмів, які представлено нижче.

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

На рисунку 4.1 відображено загальний алгоритм програми. Після перевірки наявності всіх тестових модулів, виконується перевірка наявності всіх тестових даних,

підключення всіх наявних тестів виконується їх поетапний запуск. Після виконання кожного тесту інформація про виконання зберігається в лог-файл. У випадку, якщо тест не пройдено (через тайм-аут, відсутність зв'язку з сервером, некоректні вхідні дані) тест переривається і в лог-файл зберігається інформація про помилку.

На блок-схемах алгоритмів підпрограм (рис. 4.2 та 4.3) відображено процес виконання перевірки стійкості нового паролю та системи автентифікації та збереження сесії користувача.

Відповідно до вимог безпеки новий пароль перевіряється на кількість символів (не менше 9), наявність великих символів, цифр та спец-символів, така висока складність паролю обумовлена наявністю в web-застосунку виконанням операцій онлайн-оплат.

Перевірка системи автентифікації та збереження сесії користувача необхідна для виключення ситуацій, коли користувач не виконавши операцію виходу з облікового запису може залишити свій обліковий запис вразливим.

					КБР-123.21.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		29

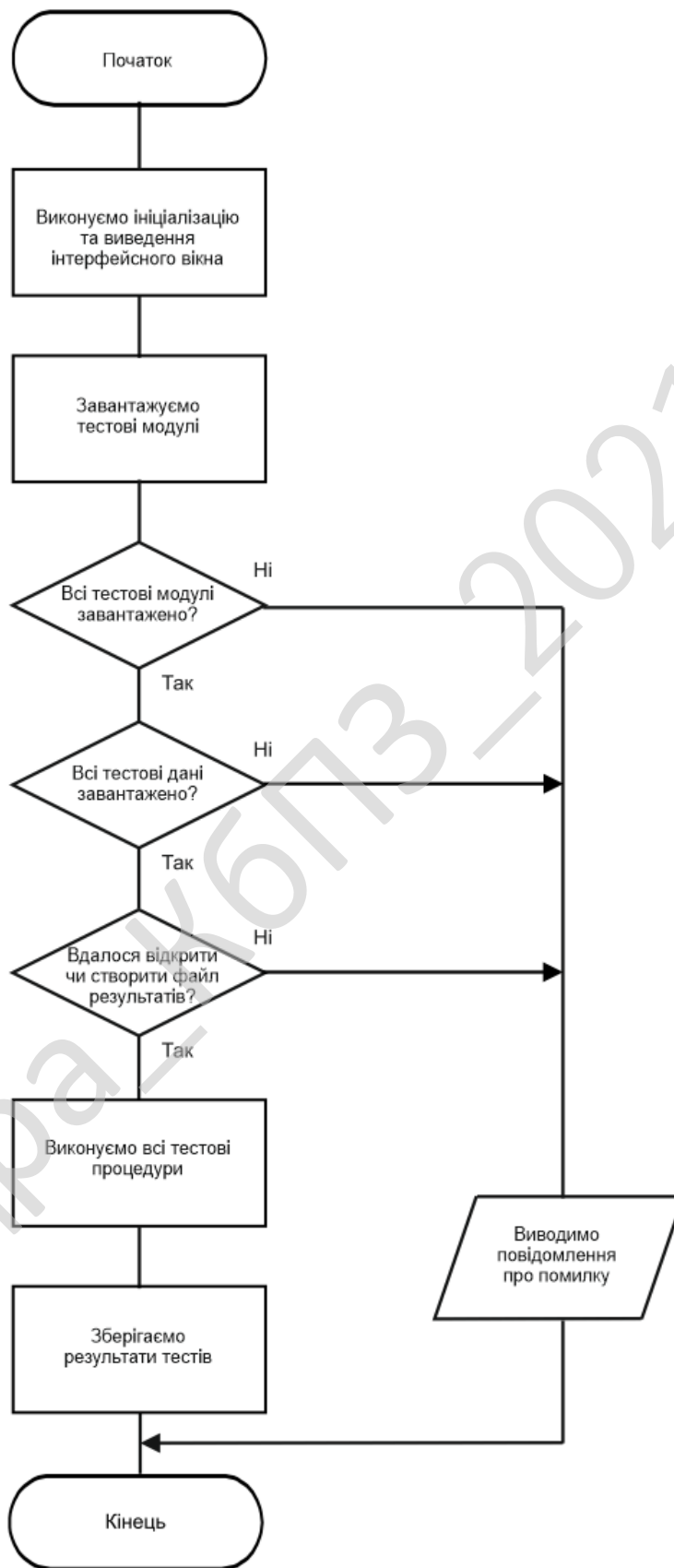


Рисунок 4.1 – Блок-схема алгоритму роботи головного модулю

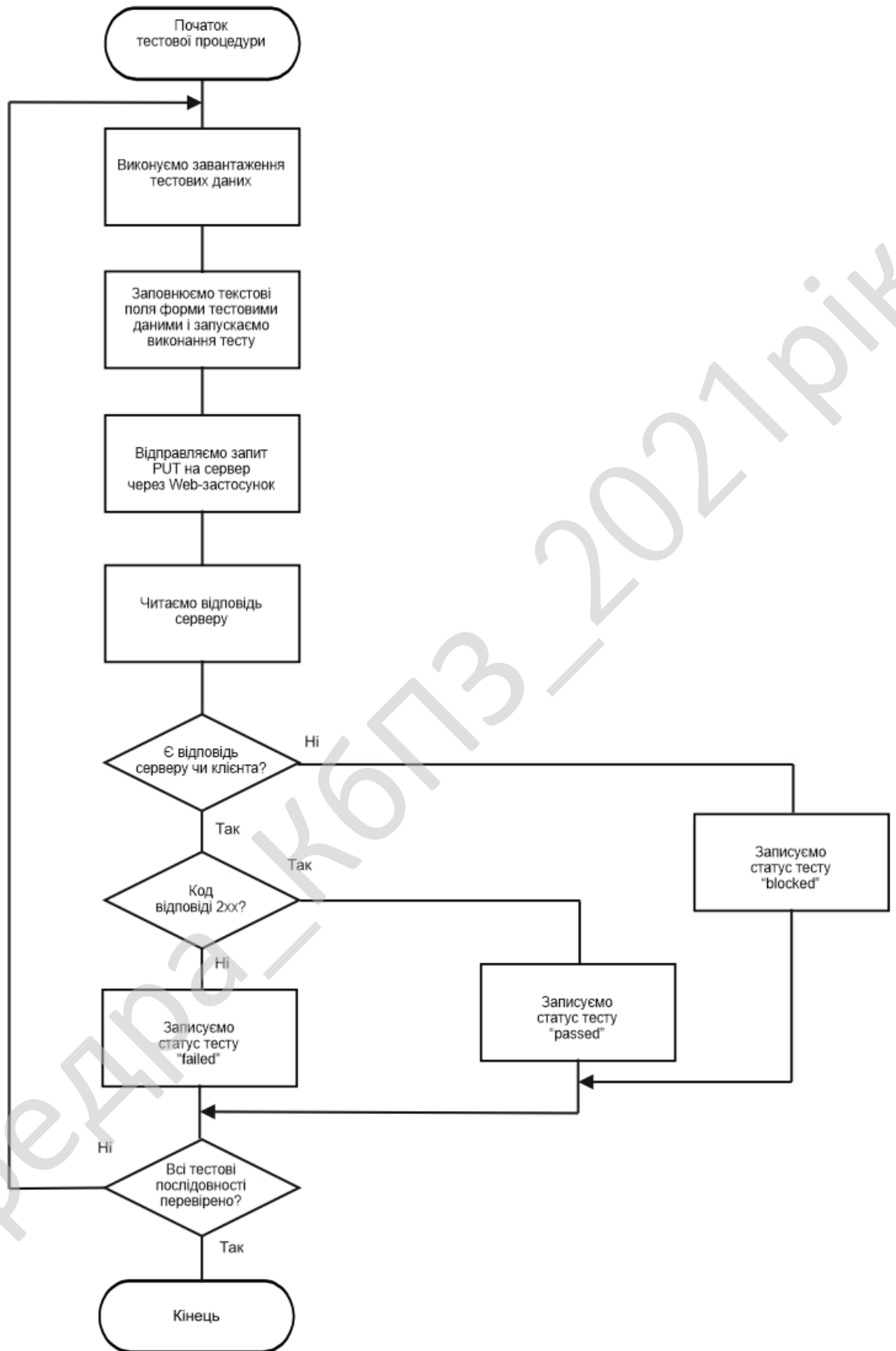


Рисунок 4.2 – Блок-схема алгоритму роботи тестової процедури

Вим.	Арк.	№ документа	Підпис	Дата

КБР-123.21.0023.00.00.ПЗ

Арк.

31


```

LandingPage landingPage = new LandingPage();
landingPage.checkLandingHeader(OrderCreation.name,
OrderCreation.address, OrderCreation.city, OrderCreation.phone);

OrderFormTestData OrderTestData = new OrderFormTestData(
    OrderCreation.ProductType + "", "1",
    new DateWorker().getTodayPlusDates(1) + "",
    new DateWorker().getTodayPlusDates(2) + "",
    "test name", "test@4fnerore.com", "4395494554", "test addr"
);

```

Завантажуємо форму замовлення, заповнюємо поля тестовими даними і відправляємо на обробку у Web-додаток:

```

form.getProductType().select(OrderTestData.ProductType);
form.getGuestsCount().select(OrderTestData.guests);
form.getFrom().select(OrderTestData.From);
form.getTo().select(OrderTestData.To);

form.getPlanInfo().getSedanPlan().check("Sail included", price);
form.getPlanInfo().getNext()
    .check(price)
    .click();

form.getUserInfo().fill(
    OrderTestData.username, OrderTestData.email,
    OrderTestData.phone, OrderTestData.address
);
form.getBuyResult()
    .check("Sail included", price, 1, "Bonus are included")
    .getBuyNow().click();

PaymentPage paymentPage = new PaymentPage();
paymentPage.getReservationDetails().check(
    OrderTestData.username, OrderTestData.phone,
OrderTestData.address,
    OrderTestData.From, OrderTestData.To, OrderTestData.guests
);
paymentPage.getPaymentDetails()
    .check(price, true)
    .getPurchase().click();

VisaPaymentPopup popup = VisaPaymentPopup.render()
    .fill(OrderTestData.email, "4242 4242 4242 4242", "022022",
"222")
    .check(
    OrderCreation.name + "", OrderTestData.email + "",
    "4242 4242 4242 4242", "02 / 2022", "222"
    );
popup.getPayWithProductd().click();

SuccessPaymentPage successPaymentPage = SuccessPaymentPage.render()
    .check(
    OrderTestData.username + "",
    OrderTestData.From + "",
    OrderTestData.To + "",
    OrderTestData.ProductType.replace(" (Disel)", "") + "",

```

КБР-123.21.0023.00.00.ПЗ

Арк.

34

Вим.	Арк.	№ документа	Підпис	Дата


```

        getPaymentSystem().select(VisaData.PAYMENT_SYSTEM);
        getTciProductOrderalOrderCommission().setValue(((VisaData)
payment).tciProductOrderalOrderCommission);
        getSendInvitationEmail().setFlag(isSendInvitationEmail);
    } else if (payment instanceof VisaData) {
        getPaymentSystem().select(VisaData.PAYMENT_SYSTEM);
        getTciProductOrderalOrderCommission().setValue(((VisaData)
payment).tciProductOrderalOrderCommission);
        if (((VisaData) payment).legalInformationJpData != null) {
            getNext().click();
            renderLegalInformationJa();
            getLegalInformationJa().fill(
                ((VisaData) payment).legalInformationJpData.seller,
                ((VisaData)
payment).legalInformationJpData.representative,
                ((VisaData) payment).legalInformationJpData.address,
                ((VisaData)
payment).legalInformationJpData.sellingPrice,
                ((VisaData)
payment).legalInformationJpData.additionalSurcharges,
                ((VisaData)
payment).legalInformationJpData.paymentPeriodAndMethod,
                ((VisaData)
payment).legalInformationJpData.deliveryTime,
                ((VisaData) payment).legalInformationJpData.returns,
                ((VisaData)
payment).legalInformationJpData.cancellation,
                ((VisaData)
payment).legalInformationJpData.phoneNumber,
                ((VisaData) payment).legalInformationJpData.email
            );
            if (((VisaData) payment).legalInformationEnData != null) {
                getNext().click();
                renderLegalInformationEn();
                getLegalInformationEn().fill(
                    ((VisaData) payment).legalInformationEnData.seller,
                    ((VisaData)
payment).legalInformationEnData.representative,
                    ((VisaData)
payment).legalInformationEnData.address,
                    ((VisaData)
payment).legalInformationEnData.sellingPrice,
                    ((VisaData)
payment).legalInformationEnData.additionalSurcharges,
                    ((VisaData)
payment).legalInformationEnData.paymentPeriodAndMethod,
                    ((VisaData)
payment).legalInformationEnData.deliveryTime,
                    ((VisaData)
payment).legalInformationEnData.returns,
                    ((VisaData)
payment).legalInformationEnData.cancellation
                );
                getSendInvitationEmail().setFlag(isSendInvitationEmail);
            }
        }
    }
    return this;
}

```

КБР-123.21.0023.00.00.ПЗ

Арк.

37

Вим.	Арк.	№ документа	Підпис	Дата
------	------	-------------	--------	------

Якщо пристрій може виконувати довільні програми, то це - комп'ютер загального призначення, і його власник повинен контролювати, що цей комп'ютер робить. Коли пристрій не дає користувачу робити це, ми називаємо це тівоізацією.

Закони, що забороняють вільні програми. Такі законодавчі акти, як Закон про авторське право цифрового тисячоліття і Директива про авторське право Європейського союзу, роблять злочином написання і обмін програмами, які можуть зламати цифрове управління обмеженнями. Ці закони не повинні обмежувати прав, які вам надає GPL.

Дискримінаційні патентні угоди. Нещодавно компанія Microsoft почала розповідати людям, що вона не збирається переслідувати користувачів вільних програм за порушення патентів - до тих пір, поки ви отримуєте ці програми від постачальника, який платить Microsoft за цей привілей. В кінцевому рахунку Microsoft намагається збирати відрахування за користування вільними програмами, що обмежує свободу користувачів. Ні в якій компанії не повинно бути можливості це робити.

У версію 3 внесено також деяке число поліпшень для того, щоб всім було легше розуміти ліцензію і користуватися нею. Але при всіх цих змінах GPLv3 не є радикально новою ліцензією; навпаки, це результат еволюції попередньої версії.

Тут виводиться інформація про наявні тестові модулі, а також поточні результати їх завантаження.

Через кнопки екранної форми користувач може додавати нові тести в систему та запускати вже підключені.

Інформація про результати виконання тестів додається в текстовий файл testresult.log в кореновому каталозі проекту, якщо файлу не знайдено, то створюється новий.

Site "prestashop.qatestlab.com.ua"	FF 66	GC 74	Opera 60
Check the registration on the site			
test1	Passed	Passed	Passed
test2	Passed	Passed	Passed
test3	Passed	Passed	Passed
test4	Passed	Passed	Passed
test5	Passed	Passed	Passed
test6	Failed	Failed	Failed
test7	Failed	Failed	Failed
test8	Failed	Failed	Failed
test9	Failed	Failed	Failed
test10	Failed	Failed	Failed
test11	Failed	Failed	Failed

Рисунок 5.2 – Скріншот testresult.log файлу.

Розроблена система досить проста і зручна у використанні, але для створення нових тестових модулів, необхідне знання Rest API, Java та Selenium.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення кваліфікаційної бакалаврської роботи реалізовано відповідно завданню.

Практичні випробування показали повну працездатність системи та відповідність технічному завданню.

Створена система реалізована на мові Java в середовищі IntelliJ IDEA.

Реалізована підтримка ОС Windows та Linux останніх версій, а також найбільш поширених web-браузерів Mozilla FireFox, Opera, Google Chrom, Microsoft Edge.

Програмне забезпечення може бути використане як у комерційних проектах, так і з метою навчання студентів зі спеціальностей Комп'ютерна інженерія та Кібербезпека.

Оформлення документації пояснювальної записки виконувалось згідно вимог викладених в методичних рекомендацій та консультацій керівника кваліфікаційної бакалаврської роботи. В основних розділах описано засоби та інструменти, які були використані в ході роботи та власне процес розробки, алгоритми функціонування системи і програмна реалізація.

					КБР-123.21.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		42

35. Корольов, А. Л. Комп'ютерне моделювання. Лабораторний практикум [Електронний ресурс] / А. Л. Корольов. - М.: БИНОМ. Лабораторія знань, 2012. - 296 с. : Ил. - (Педагогічну освіту). - ISBN 978-5-9963-0270-3.

36. Баранова, Е. К. Основи інформатики та захисту інформації [Електронний ресурс]: Навч. посібник / Є. К. Баранова. - М.: РІОР: ИНФРА-М, 2013. - 183 с.

37. Гагаріна Л. Р./ Технологія розробки програмного забезпечення./ Гагаріна Л. Р, Кокорева Е. В., Віснадул Б. Д. - М.: ІД. «Форум»: ИНФРА -М, 2011. – 400 с.

38. Голіцина О. Л. / Програмування на мовах високого рівня. Голіцина О. Л., Попов И. І /М.: ФОРУМ, 2011. - 496 с

39. Покришень Д.А. / Технологічні принципи проектування інформаційних систем / Д.А. Покришень // Сборник научных трудов SWorld. – Выпуск 1. Том 17. – Иваново: МАРКОВА АД, 2014. – С. 37-44

40. Саммервил И. / Инженерия программного обеспечения / И. Саммервил. - М.: Вильямс, 2002. - 720 с.

41. Spillner A. Software Testing Foundations / A. Spillner, T. Linz, H. Schafer. - Dpunkt: Verlag, 2007. - 277 p.

42. Rajlich W. A stage Model for the Software Life Cycle / W. Rajlich, K. Bennett // Computer. - 2000. - July. - P. 77 - 70.

43. Bosch J. Design and use of software architectures / J. Bosch. -Addison Wesley, 2000. - 325 p.

44. Glass R.L. Extrime programming the good, the bad. and the bottom line / R.L. Glass // IEEE. Software. - 2001, - Vol.18, n.7. - P.11 -P. 112,- 111 - 112.

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					КБР-123.21.0023.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив		Водзинський Є.Ю			Програмне забезпечення системи тестування текстових полів web-застосунків	Літ.	Аркуш	Аркушів
Перевірів		Бісюк В.А.				Б	1	6
Н. Контр.		Гермак В.С.				ЦНТУ КІ-18-3ск		
Затв.		Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення системи тестування текстових полів web-застосунків.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (наказ № _____ від ____ грудня 2020 р.).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є підвищення контроль та підвищення якості сайтів на основі впровадження нових інформаційних технологій і застосування сучасних засобів програмування. А саме розробка програмного забезпечення системи тестування текстових полів web-застосунків.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					КБР-123.21.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- модульну архітектуру з можливістю редагування та модифікації;
- можливість завдання і редагування початкових параметрів процесу;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версії драйверів та операційної системи (підтримка ОС Windows версії 8 і вище).

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Компонент повинен використати існуючі угоди по стандартним викликам процедур, функцій, засобів і форм, визначених технічною документацією на середовище розробки.

					КБР-123.21.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 18-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ЕОМ типу IBM PC, працювати в ОС Windows 10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 8/10.

5.8.1 Обладнання

Комп'ютер Intel® Core2/8 Mb/2 GHz/RAM 8 Gb або більш потужні.

5.8.2 Мова програмування

Середовище Java.

Бібліотеки - Selenium WebDriver.

					КБР-123.21.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема роботи системи – 1 аркуш.
- Функціональна схема роботи системи – 1 аркуш.
- Діаграма взаємодії процесів – 1 аркуш.
- Блок-схема алгоритму роботи головного модулю – 1 аркуш.
- Блок-схема алгоритму тестової процедури – 1 аркуш.
- Пояснювальна записка – 45 арк.

					КБР-123.21.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

8.3 Розробка функціональних схем, блок-схем алгоритмів роботи програмного забезпечення компоненту.

8.4 Побудова схем взаємодії структур даних.

8.5 Створення прототипу компоненту. Створення програмного продукту.

8.6 Відлагодження компоненту, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

9.2 Подання кваліфікаційної бакалаврської роботи на захист .06.2021 р.

					КБР-123.21.0023.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник кваліфікаційної бакалаврської роботи
_____ В.А. Бісюк

*Програмне забезпечення системи тестування текстових
полів web-застосунків*

Лістинг програми

Код документу 12

Носій: DVD-диск

Загальна кількість аркушів: 32

Літера: РП

```

//Модуль для вибору і завантаження тестових модулів та формування тестових наборів
package project;

import org.junit.platform.runner.JUnitPlatform;
import org.junit.platform.suite.api.SelectClasses;
import org.junit.runner.RunWith;

@RunWith(JUnitPlatform.class)
>SelectClasses({
    TestSuiteLoginCheck.class,
    TestSuiteOrderingCheck.class,
    TestSuitePaymentCheck.class
})
public class RunAllTestSuits {}

package project;

import core.control.ProductsManager;
import core.control.Watcher;
import core.Products.Chrome;
import core.tools.DateWorker;
import org.junit.jupiter.api.*;
import org.junit.jupiter.api.extension.ExtendWith;
import project.management.LandingPage;
import project.management.ManagementLoginPage;
import project.management.products.OrderFormTestData;
import project.management.products.OrderCreation;
import project.management.Product.ProductsPage;
import project.management.payment.*;

import static com.codeborne.selenium.Condition.*;

mainURL=("http://prestashop.qatestlab.com.ua");

@DisplayName("TEST SUITE 'Create Order'")
@ExtendWith(Watcher.class)
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class TestSuiteLoginCheck {

    @BeforeEach
    public void beforeExecutionTests() {
        ProductsManager.launch(new Chrome().setup());
    }

    @AfterAll
    public static void afterExecutionTests() {
        ProductsManager.quitAll();
    }

    // Модулі тестування полів форми створення замовлення

    @Order(1)
    @DisplayName("TEST CASE: Create Order (Visa, Without Source)")
    @Test
    public void LoginCheckVisaWithoutSource() {
        ManagementLoginPage.open()
            .login("testProductOrder123+1211197@gmail.com",
System.getenv("TciProductOrderalOrderTestPass"));
        ProductsPage ProductsPage = ProductsPage.open().renderProductList();

```

```

OrderCreation OrderCreation = new OrderCreation(
"http://prestashop.gatestlab.com.ua/OrderForm/active_storage/blobs/JfcmFpbHMiOnsibW
Vz" +
"c2FnZSI6IkJBaHBBaElHIiwizXhwIjpudWxsLCJwdXIiOiJibG9iX2lkIn19--
c648464d7f193d26cb0ebd91" +
"91809cd93a1433a7/Product_exterior.ua"
);
OrderCreation OrderCreationD = new OrderCreation(
    "Queen (ua) [auto] (Disel)",
    "60,000", null
);
OrderCreation OrderCreation = new OrderCreation(
    "Product (Name) [Color] (Size)", null
);

String price = OrderCreation.priceWithSail;

ProductsPage.getProductItem(0)
    .check(
        OrderCreation.name, OrderCreation.zipCode,
OrderCreation.city, OrderCreation.address,
        OrderCreation.interiorImageSrc
    ).getPreview().click();

LandingPage landingPage = new LandingPage();
landingPage.checkLandingHeader(OrderCreation.name, OrderCreation.address,
OrderCreation.city, OrderCreation.phone);

OrderFormTestData OrderTestData = new OrderFormTestData(
    OrderCreation.ProductType + "", "1",
    new DateWorker().getTodayPlusDates(1) + "",
    new DateWorker().getTodayPlusDates(2) + "",
    "test name", "test@4fnerore.com", "4395494554", "test addr"
);

LandingPage.OrderForm form = landingPage.getOrderForm();

form.getErrorMessage().shouldBe(not(visible));

form.getProductType().select(OrderTestData.ProductType);
form.getGuestsCount().select(OrderTestData.guests);
form.getFrom().select(OrderTestData.From);
form.getTo().select(OrderTestData.To);

form.getPlanInfo().getSedanPlan().check("Sail included", price);
form.getPlanInfo().getNext()
    .check(price)
    .click();

form.getUserInfo().fill(
    OrderTestData.username, OrderTestData.email,
    OrderTestData.phone, OrderTestData.address
);
form.getBuyResult()
    .check("Sail included", price, 1, "Bonus are included")
    .getBuyNow().click();

PaymentPage paymentPage = new PaymentPage();
paymentPage.getReservationDetails().check(
    OrderTestData.username, OrderTestData.phone, OrderTestData.address,

```

```

        OrderTestData.From, OrderTestData.To, OrderTestData.guests
    );
    paymentPage.getPaymentDetails()
        .check(price, true)
        .getPurchase().click();

    VisaPaymentPopup popup = VisaPaymentPopup.render()
        .fill(OrderTestData.email, "4242 4242 4242 4242", "022022", "222")
        .check(
            OrderCreation.name + "", OrderTestData.email + "",
            "4242 4242 4242 4242", "02 / 2022", "222"
        );
    popup.getPayWithProductd().click();

    SuccessPaymentPage successPaymentPage = SuccessPaymentPage.render()
        .check(
            OrderTestData.username + "",
            OrderTestData.From + "",
            OrderTestData.To + "",
            OrderTestData.ProductType.replace(" (Disel)", "") + "",
            "Sail included",
            OrderTestData.guests + "",
            price + "",
            OrderTestData.email + ""
        );
}

@Order(2)
@DisplayName("TEST CASE: Create Order (VISA, Without Source)")
@Test
public void LoginCheckVisaWithoutSource() {
    ManagementLoginPage.open()
        .login("testProductOrder123+1211195@gmail.com",
System.getenv("TciProductOrderalOrderTestPass"));

    OrderCreation OrderCreation = new OrderCreation(
        "TCIPProduct (ua) [auto]",
        "LA City, LA-to 105-0001 US (ua)",
        "LA", "105-0001",
        "000 435 3443", "https://www.google.com",

"https://beta.t.tciProductOrderalengines.us/rails/active_storage/blobs/eyJfcmFpbHMi
OnsibWVzc2FnZ" +
        "SI6IkJBaHBBdGdGIiwizXhwIjpuYWxsLCJwJwXIIoiJibG9iX2lkIn19--
6eaa74650b0990171ffa114f456278688" +
        "6f70cad/exterior.useg"
    );
    OrderCreation OrderCreation = new OrderCreation(
        "Classic Product (ua) [auto] (No Disel)",
        "63,600", "59,200" //price x2 (2 nights)
    );
    String price = OrderCreation.priceWithoutSail;

    ProductsPage ProductsPage = new ProductsPage().renderProductList();
    ProductsPage.getProductItem(0)
        .check(
            OrderCreation.name, OrderCreation.zipCode,
OrderCreation.city, OrderCreation.address,
            OrderCreation.interiorImageSrc
        ).getPreview().click();

    LandingPage landingPage = new LandingPage();

```

```

landingPage.checkLandingHeader(OrderCreation.name, OrderCreation.address,
OrderCreation.city, OrderCreation.phone);

OrderFormTestData OrderTestData = new OrderFormTestData(
    OrderCreation.ProductType + "", "3",
    new DateWorker().getTodayPlusDates(1) + "",
    new DateWorker().getTodayPlusDates(3) + "",
    "test name", "test@4fnerore.com", "4395494554", "test addr"
);

LandingPage.OrderForm form = landingPage.getOrderForm();

form.getErrorMessage().shouldBe(not(visible));

form.getProductType().select(OrderTestData.ProductType);
form.getGuestsCount().select(OrderTestData.guests);
form.getFrom().select(OrderTestData.From);
form.getTo().select(OrderTestData.To);

form.getPlanInfo().getPlanList().getSailIncluded().checkPrice(OrderCreation.priceWithSail);
form.getPlanInfo().getPlanList().getWithoutSail()
    .checkPrice(price)
    .click();
form.getPlanInfo().getPriceAlert().shouldHave(text("All prices are tax
included"));
form.getPlanInfo().getNext().click();

form.getUserInfo().fill(
    OrderTestData.username, OrderTestData.email,
    OrderTestData.phone, OrderTestData.address
);
form.getBuyResult()
    .check("Without Sail", price, 2, "Bonus are included")
    .getBuyNow().click();

PaymentPage paymentPage = new PaymentPage();
paymentPage.getReservationDetails().check(
    OrderTestData.username, OrderTestData.phone, OrderTestData.address,
    OrderTestData.From, OrderTestData.To, OrderTestData.guests
);
paymentPage.getPaymentDetails()
    .check(price, true)
    .getPurchase().click();

VisaPaymentPage VisaPaymentPage = VisaPaymentPage.render();
VisaPaymentPage.getApplicationInformation().check(OrderTestData.username,
OrderTestData.email);
VisaPaymentPage.getNameOfProduct().shouldHave(text(OrderCreation.name));
VisaPaymentPage.getPlanDetails().check(
    OrderCreation.ProductType.replace(" (No Diesel)", ""), "Without
Sail"
);
VisaPaymentPage.getGuestDetails().getEea().widget().selectOption("Yes");
VisaPaymentPage.getGuestDetails().check(
    OrderTestData.username + "",
    OrderTestData.address + "",
    OrderTestData.phone + "",
    OrderTestData.email + "",
    "Yes"
);
VisaPaymentPage.getOrderDetails().check(
    OrderTestData.From + "",

```

```

        OrderTestData.To + "",
        2, "15:00"
    );
    VisaPaymentPage.getTourPrice(); //todo
    VisaPaymentPage.getPaymentSection().fill("4111111111111111", "09", "2021",
"946");
    VisaPaymentPage.getAttentionInformation(); //todo
}

@Order(3)
@DisplayName("TEST CASE: Create Order (VISA, Without Source)")
@Test
public void LoginCheckVISAWithoutSource() {
    ManagementLoginPage.open()
        .login("testProductOrder123+1211193@gmail.com",
System.getenv("TciProductOrderalOrderTestPass"));

    OrderCreation OrderCreation = new OrderCreation(
"https://prestashop.t.tciProductOrderalengines.us/rails/active_storage/blobs/" +
"eyJfcmFpbHMiOnsibWVzc2FnZSI6IkJBaHBBdDhGIiwizXhwIjpudWxsLCJwdXIiOiJibG9iX2lkIn19-"
+
        "-68aca82c9a0e9a929fe3bc3cb795f55417cd3af8/exterior2.usg"
    );
    OrderCreation OrderCreation = new OrderCreation(
        "Classic roop (ua) (Diesel)",
        "2,150", null
    );
    String price = OrderCreation.priceWithSail;

    ProductsPage ProductsPage = new ProductsPage().renderProductList();
    ProductsPage.getProductItem(0)
        .check(
            OrderCreation.name, OrderCreation.zipCode,
OrderCreation.city, OrderCreation.address,
            OrderCreation.interiorImageSrc
        ).getPreview().click();

    LandingPage landingPage = new LandingPage();
    landingPage.checkLandingHeader(OrderCreation.name, OrderCreation.address,
OrderCreation.city, OrderCreation.phone);

    OrderFormTestData OrderTestData = new OrderFormTestData(
        OrderCreation.ProductType + "", "1",
        new DateWorker().getTodayPlusDates(1) + "",
        new DateWorker().getTodayPlusDates(2) + "",
        "test name", "test@4fnerore.com", "4395494554", "test addr"
    );
    LandingPage.OrderForm form = landingPage.getOrderForm();

    form.getErrorMessage().shouldBe(not(visible));

    form.getProductType().select(OrderTestData.ProductType);
    form.getGuestsCount().select(OrderTestData.guests);
    form.getFrom().select(OrderTestData.From);
    form.getTo().select(OrderTestData.To);

    form.getPlanInfo().getSedanPlan().check("Without Sail", price);
    form.getPlanInfo().getPriceAlert().shouldHave(text("All prices are tax
included"));
    form.getPlanInfo().getNext()

```

```

        .check(price)
        .click();

form.getUserInfo().fill(
    OrderTestData.username, OrderTestData.email,
    OrderTestData.phone, OrderTestData.address
);
form.getBuyResult()
    .check("Without Sail", price, 1, "Bonus are included")
    .getBuyNow().click();

PaymentPage paymentPage = new PaymentPage();
paymentPage.getReservationDetails().check(
    OrderTestData.username, OrderTestData.phone, OrderTestData.address,
    OrderTestData.From, OrderTestData.To, OrderTestData.guests
);
paymentPage.getPaymentDetails()
    .check(price, true)
    .getPurchase().click();
VisaPaymentPopUp Visa = VisaPaymentPopUp.render();
Visa
    .fill("test", "1234 3245 1111 2435",
        "March", "2022", "123")
    .check("test", "1234 3245 1111 2435",
        "March", "2022", "123", price)
    .getPayWithProductd().click();

SuccessPaymentPage successPaymentPage = SuccessPaymentPage.render();
successPaymentPage.check(
    OrderTestData.username + "",
    OrderTestData.From + "",
    OrderTestData.To + "",
    OrderTestData.ProductType.replace(" (Disel)", "") + "",
    "Without Sail",
    OrderTestData.guests + "",
    price + "",
    OrderTestData.email + ""
);
}

@Order(4)
@DisplayName("TEST CASE: Create Order (VISA, TCI)")
@Test
public void LoginCheckVisaTCI() {
    ManagementLoginPage.open()
        .login("testlogin83+0400@gmail.com ",
System.getenv("TciProductOrderalOrderOlgaPass"));
    //todo test
}

@Order(5)
@DisplayName("TEST CASE: Create Order (Visa, TCI)")
@Test
public void LoginCheckVisaTCI() {
    ManagementLoginPage.open()
        .login("testProductOrder123+1510191@gmail.com",
System.getenv("TciProductOrderalOrderTestPass"));
    //todo test
}

@Order(6)

```

```

    @DisplayName("TEST CASE: Create Order (VISA, TCI)")
    @Test
    public void LoginCheckVisaTCI() {
        ManagementLoginPage.open()
            .login("testlogin83+04@gmail.com ",
System.getenv("TciProductOrderalOrderOlgaPass"));
        //todo test
    }
}

```

```
package project;
```

```

import com.codeborne.selenium.Selenide;
import core.control.ProductsManager;
import core.control.Watcher;
import core.control.application_tools.LocalStorage;
import core.Products.Chrome;
import org.junit.jupiter.api.*;
import org.junit.jupiter.api.extension.ExtendWith;
import project.management.ManagementLoginPage;
import project.management.ManagementSidebar;
import project.management.Product.ConnectToChannelManagerPopup;
import project.management.Product.ProductNewPage;

```

```

import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Random;

```

```

@DisplayName("TEST SUITE 'Create Product'")
@ExtendWith(Watcher.class)
public class TestSuitePaymentCheck {

    @BeforeAll
    public static void beforeExecutionTests() {
        ProductsManager.launch(new Chrome().setup());
    }

    @AfterEach
    public void afterEachExecutionTest() {
        LocalStorage.getStorage().setItem("management", "");
    }

    @AfterAll
    public static void afterExecutionTests() {
        ProductsManager.quitAll();
    }

    @DisplayName("TEST CASE: Create new Order")
    @Test
    public void createNewProduct() {
        ManagementLoginPage.open()
            .login("testlogin83+0400@gmail.com",
System.getenv("TciProductOrderalOrderOlgaPass"));
        new ManagementSidebar().getAddProduct().click();

        ConnectToChannelManagerPopup
            .render()
            .fill("B3892756",
System.getenv("TCIApiKeyPass")).getConfirm().click();
        Selenide.sleep(3000);
    }
}

```

```

String code = " #" + String.format("%06d", new Random().nextInt(999999));

ProductNewPage ProductNewPage = new ProductNewPage();
ProductNewPage.RegisterProduct registerProduct =
ProductNewPage.getRegisterProduct();
registerProduct.fill(
    "P48909",
    new File("src/main/resources/" + "images/logos/Product_logo.useg"),
    new File("src/main/resources/" +
"images/exterior/exterior_Product.usg"),
    "0100973", "T-shirt",
    "4-chōme-6 T-shirt (ua)",
    "4-chōme-6 T-shirt (ua)",
    "5495959444",
    "Privacy policy (ua)", "Privacy policy (ua)"
);
registerProduct.getSaveAndAddCancellationPolicy().click();

ProductNewPage.CancellationSettings cancellationSettings =
ProductNewPage.getCancellationSettings();
cancellationSettings.fillEarlyCancellationFeeSettings(
    "3", "5", "10"
);
cancellationSettings.getSaveAndAddCancellationPolicy().click();

ProductNewPage.AddProduct addProduct = ProductNewPage.getAddProduct();
addProduct.fill(
    "Double (ua)" + code,
    "Double (ua)" + code,
    false,
    "Double",
    "4",
    "2",
    true,
    "17074713",
    false,
    null,
    new ArrayList<>(Arrays.asList("TV", "Comb")), // these amenities
will be exclude
    "Product description (ua)",
    "Product description (ua)",
    new File("src/main/resources/" +
"images/vip_Product/vip_Product_1.usg"),
    new File("src/main/resources/" +
"images/vip_Product/vip_Product_2.usg")
);
addProduct.getComplete().click();

//todo check toast message
//todo check that Product is successfully created
}
}

@BeforeEach
public void beforeEachExecutionTest() {
    ProductsManager.launch(new Chrome().setup());
    this.code = " #" + String.format("%06d", new Random().nextInt(999999));
    this.email = MailController.EMAIL
        .replace("@", code.replace(" #", "+") + "@");
}

@AfterEach

```

```

public void afterEachExecutionTest() {
    MailController.getUnreadMails(); // marks all mails as read
}

@AfterAll
public static void afterExecutionTests() {
    ProductsManager.quitAll();
}

@Test
public void createNewClientVisaTCI() {
    AdminLoginPage.open()
        .login("default-admin@mail.com", "default");

    new AdminSidebar().getAddNewClient().click();
    AddNewClientPopup popup = AddNewClientPopup.render();
    popup.fill(
        "Devbug" + code,
        "test_" + code.replace(" ", ""),
        email + "",
        "4395995495",
        "test google (ua)" + code,
        "test google (ua)" + code,
        "TL TCI",
        popup.fillVisaData(true),
        true
    ).getAddNewClient().click();

    SingUpNewClientEmail mail = new SingUpNewClientEmail(Objects
        .requireNonNull(MailController.waitNewMail(120)));
    mail.check(email, "test_" + code.replace(" ", ""));

    CompleteSingUpPage completeSingUpPage =
    CompleteSingUpPage.open(mail.getInviteLink());
    String password = System.getenv("TciProductOrderalOrderTestPass");
    completeSingUpPage
        .fill(password + "_Te$1", password + "_Te$1")
        .getSetPassword().click();
    ConnectVisaPage.render();
    new PageHeader().getLogout().click();
}

@Test
public void createNewClientVisaWithoutSource() {
    AdminLoginPage.open()
        .login("default-admin@mail.com", "default");

    new AdminSidebar().getAddNewClient().click();
    AddNewClientPopup popup = AddNewClientPopup.render();
    popup.fill(
        "Devbug2" + code,
        "test2_" + code.replace(" ", ""),
        email + "",
        "95495423949",
        "test google 2 (ua)" + code,
        "test google 2 (ua)" + code,
        "Without source",
        popup.fillVisaData("10"),
    );
}

```

```

        true
    ).getAddNewClient().click();

    SingUpNewClientEmail mail = new SingUpNewClientEmail(Objects
        .requireNonNull(MailController.waitNewMail(120)));
    mail.check(email, "test2_" + code.replace(" ", ""));

    CompleteSingUpPage completeSingUpPage =
CompleteSingUpPage.open(mail.getInviteLink());
    String password = System.getenv("TciProductOrderalOrderTestPass");
    completeSingUpPage
        .fill(password + "_Te$1", password + "_Te$1")
        .getSetPassword().click();
    new ProductsPage();
    new PageHeader().getLogout().click();
}

@DisplayName("TEST CASE: Create new client (VISA, TCI)")
@Order(3)
@Test
public void createNewClientVisaTCI() {
    AdminLoginPage.open()
        .login("default-admin@mail.com", "default");

    new AdminSidebar().getAddNewClient().click();
    AddNewClientPopup popup = AddNewClientPopup.render();
    popup.fill(
        "Devbug3" + code,
        "test3_" + code.replace(" ", ""),
        email + "",
        "59955404230",
        "test google 3 (ua)" + code,
        "test google 3 (ua)" + code,
        "TL TCI",
        popup.fillVisaData(
            "6",
            popup.fillLegalInformationJpData(
                "test seller (ua)",
                "test representative (ua)",
                "test address (ua)",
                "59955404230",
                email + "",
                "21000",
                "test surcharges (ua)",
                "test payment method (ua)",
                "3",
                "7",
                "5"
            ),
            popup.fillLegalInformationEnData(
                "test seller (ua)",
                "test representative (ua)",
                "test address (ua)",
                "21000",
                "test surcharges (ua)",
                "test payment method (ua)",
                "3",
                "7",
                "5"
            )
        ),
        true
    ).getAddNewClient().click();
}

```

```

SingUpNewClientEmail mail = new SingUpNewClientEmail(Objects
    .requireNonNull(MailController.waitNewMail(120)));
mail.check(email, "test2_" + code.replace(" ", ""));

CompleteSingUpPage completeSingUpPage =
CompleteSingUpPage.open(mail.getInviteLink());
String password = System.getenv("TciProductOrderalOrderTestPass");
completeSingUpPage
    .fill(password + "_Te$1", password + "_Te$1")
    .getSetPassword().click();
new ProductsPage();
new PageHeader().getLogout().click();
}
}

// Модуль тестування текстових полів логіну користувача та полів реєстрації нового
// користувача в системі

package project;

import core.control.Delay;
import core.main_web_elements.Button;
import core.main_web_elements.Input;
import core.main_web_elements.WebElement;
import io.qameta.allure.Step;
import org.openqa.selenium.By;

import static com.codeborne.selenide.Condition.visible;

public class LoginForm extends WebElement {
    private Input email;
    private Input password;
    private Button submit;
    private Button forgotPassword;
    private WebElement rememberMe;

    public LoginForm() {
        super("//div[@class='md-title' and text()='Login']/..");
        this.email = new
Input(this.widget().$(By.xpath("form//input[@type='email']")));
        this.password = new
Input(this.widget().$(By.xpath("form//input[@type='password']"))) {
            @Step("☒ set value [*****]")
            @Override
            public WebElement setValue(String text) {
                return super.setValue(text);
            }
        };
        this.submit = new
Button(this.widget().$(By.xpath("form//button[@type='submit']")));
        this.forgotPassword = new
Button(this.widget().$(By.xpath("div//a[text()='Forgot password?']")));
        this.rememberMe = new WebElement(this.widget().$(By.xpath(""))); //todo
checkbox
    }

    @DisplayName("TEST SUITE 'Create new User'")
    @ExtendWith(Watcher.class)
    @TestMethodOrder(MethodOrderer.OrderAnnotation.class)
    public class TestSuiteOrderingCheck {
        private String code;

```

```

private String email;

public void render() {
    this.widget().waitUntil(visible, Delay.STANDARD.getTime());
    this.email.shouldBe(visible);
    this.password.shouldBe(visible);
    this.submit.shouldBe(visible);
    this.forgotPassword.shouldBe(visible);
}

@Step("email")
public Input getEmail() {
    return email;
}

@Step("password")
public Input getPassword() {
    return password;
}

@Step("login (button)")
public Button getSubmit() {
    return submit;
}

@Step("forgot password")
public Button getForgotPassword() {
    return forgotPassword;
}

@Step("remember me")
public WebElement getRememberMe() {
    return rememberMe;
}
}

// Модуль перевірки логіну з правами менеджера

package project.management;

import com.codeborne.selenide.Selenide;
import core.control.Delay;
import core.control.Domen;
import core.main_web_elements.WebElement;
import io.qameta.allure.Step;
import project.admin.AdminLoginPage;
import project.management.Product.ProductsPage;

import static com.codeborne.selenide.Condition.visible;

public class ManagementLoginPage extends AdminLoginPage {
    private WebElement logo;

    @Step("open [Management Login page]")
    public static ManagementLoginPage open() {
        return Selenide.open(ManagementLoginPage.url(), ManagementLoginPage.class);
    }

    public static String url() {
        return Domen.MANAGEMENT.getUrl().concat("login");
    }
}

```

```

    }

    @Override
    public void render() {
        super.render();
        this.logo = new WebElement("//img[@alt='TciProductOrderal Order']");
        logo.widget().shouldBe(visible);
    }

    @Step("logo")
    public WebElement getLogo() {
        return logo;
    }

    @Override
    public void login(String email, String password) {
        getLoginForm().getEmail().setValue(email);
        getLoginForm().getPassword().setValue(password);
        getLoginForm().getSubmit().click();
        new ProductsPage().getTitle().widget().waitUntil(visible,
Delay.STANDARD.getTime());
        Selenide.sleep(Delay.LOW_1000.getTime());
    }
}

package project.management;

import com.codeborne.selenide.Selenide;
import com.codeborne.selenide.SelenideElement;
import core.control.Delay;
import core.main_web_elements.*;
import io.qameta.allure.Step;
import org.openqa.selenium.By;

import static com.codeborne.selenide.Condition.*;
import static com.codeborne.selenide.Selenide.$;

public class LandingPage {

    private WebElement OrderTitle = new WebElement("//div[@class='Order-
title']/h2");
    private WebElement address = new WebElement("//div[@class='Order-
Product']/div[1]");
    private WebElement phone = new WebElement("//div[@class='Order-
Product']/div[2]");
    private WebElement website = new WebElement("//div[@class='Order-
Product']/div[3]");
    private OrderForm OrderForm;

    public LandingPage() {
        render();
    }

    public void render() {
        OrderTitle.waitUntil(visible, Delay.LONG.getTime());
        this.OrderForm = new OrderForm();
    }
}

```

```

@Step("Order title")
public WebElement getOrderTitle() {
    return OrderTitle;
}

@Step("Address")
public WebElement getAddress() {
    return address;
}

@Step("Phone")
public WebElement getPhone() {
    return phone;
}

@Step("Website")
public WebElement getWebsite() {
    return website;
}

public OrderForm getOrderForm() {
    return OrderForm;
}

public void checkLandingHeader(String title, String address, String city,
String phone) {
    getOrderTitle().shouldHave(text(title));
    getAddress().shouldHave(text(address + ", " + city));
    getPhone().shouldHave(text(phone));
}

public class OrderForm extends WebElement {
    private Selector ProductType;
    private Selector guestsCount;
    private DatePicker From;
    private DatePicker To;
    private UserInfo userInfo;
    private PlanInfo planInfo;
    private BuyResult BuyResult;
    private WebElement errorMessage;

    public OrderForm() {
        super("//div[@class='Order-form']/form");
        this.ProductType = new Selector("//label[text()='Product
type:']/following-sibling::div");
        this.guestsCount = new Selector("//label[text()='Guests']/following-
sibling::div");
        this.From = new DatePicker("//label[text()='Sail request']/preceding-
sibling::input");
        this.To = new DatePicker("//label[text()='Bonus request']/preceding-
sibling::input") {

            @Override
            public void select(String date) {
                super.select(date);
                Selenium.sleep(Delay.LOW_500.getTime());
                if (errorMessage.widget().is(not(visible)) &&
                    errorMessage.widget().is(not(text("Requested period is
not available to Buy")))) {

```

```

        OrderForm.this.planInfo = new PlanInfo();
    }
}
};
this.errorMessage = new WebElement("//p[@class='md-error']");
render();
}

public void render() {
    this.widget().waitUntil(visible, Delay.STANDARD.getTime());
    if (errorMessage.widget().is(not(visible)) &&
        errorMessage.widget().is(not(text("Sorry, there are no Products
matching your criteria. " +
        "Try to change date range or number of guests")))) {
        this.planInfo = new PlanInfo();
    }
    this.userInfo = new UserInfo();
    this.BuyResult = new BuyResult();
}

@Step("Product type")
public Selector getProductType() {
    return ProductType;
}

@Step("Guests count")
public Selector getGuestsCount() {
    return guestsCount;
}

@Step("Sail request")
public DatePicker getFrom() {
    return From;
}

@Step("Bonus request")
public DatePicker getTo() {
    return To;
}

public UserInfo getUserInfo() {
    return userInfo;
}

@Step("Error message")
public WebElement getErrorMessage() {
    return errorMessage;
}

public PlanInfo getPlanInfo() {
    return planInfo;
}

public BuyResult getBuyResult() {
    return BuyResult;
}

public class PlanInfo extends WebElement {
    private SedanPlan sedanPlan;
    private PlanList planList;
}

```

```

private ButtonNext next;
private WebElement priceAlert;

public PlanInfo() {
    super("//div[contains(@class, 'confirm-button-container')]");
    this.priceAlert = new
WebElement(this.widget().$(By.xpath("div[@class='price-alert']")));
    this.next = new ButtonNext(this.widget().$(By.xpath("button"))) {

        @Override
        public void click() {
            super.click();
            OrderForm.this.planInfo.widget().waitUntil(hidden,
Delay.LOW_2000.getTime());
            OrderForm.this.userInfo = new UserInfo();
            OrderForm.this.BuyResult = new BuyResult();
        }
    };
    renderPlanSection();
}

private void renderPlanSection() {
    Selenide.sleep(Delay.LOW_500.getTime());
    $(By.xpath("//div[@class='spinner']")).waitUntil(hidden,
Delay.STANDARD.getTime());
    this.singlePlan = new SinglePlan();
    this.planList = new PlanList();
}

@Step("Single Plan")
public SinglePlan getSinglePlan() {
    return singlePlan;
}

@Step("Both Plan list")
public PlanList getPlanList() {
    return planList;
}

@Step("Next")
public ButtonNext getNext() {
    return next;
}

@Step("Price alert (tax)")
public WebElement getPriceAlert() {
    return priceAlert;
}

public class ButtonNext extends Button {

    private ButtonNext(SelenideElement element) {
        super(element);
    }

    @Step("price")
    public ButtonNext check(String price) {

```

```

        getNext().shouldHave(text("Next " + price));

        return this;
    }
}

public class SinglePlan extends WebElement {
    private WebElement planType;
    private WebElement price;

    private SinglePlan() {
        super("//div[contains(@class, 'plan-list')]");
        this.planType = new
WebElement(this.widget().$(By.xpath("div/div[@class='single-plan']")));
        this.price = new
WebElement(this.widget().$(By.xpath("div/div[@class='single-price']")));
    }

    @Step("plan type")
    public WebElement getPlanType() {
        return planType;
    }

    @Step("price")
    public WebElement getPrice() {
        return price;
    }

    public SinglePlan check(String planType, String price) {
        getPlanType().shouldHave(text(planType));
        getPrice().shouldHave(text(price));
        return this;
    }
}

public class PlanList extends WebElement {
    private BuyItemButton withoutSail;
    private BuyItemButton SailIncluded;

    private PlanList() {
        super("//div[contains(@class, 'plan-list')]/div[@class='both-
plan']");
        this.withoutSail = new BuyItemButton(this.widget()
        .$(By.xpath("div[contains(text(), 'Without Sail')]")));
        this.SailIncluded = new BuyItemButton(this.widget()
        .$(By.xpath("div[contains(text(), 'Sail
included')]")));
    }

    @Step("Without Sail (button)")
    public BuyItemButton getWithoutSail() {
        return withoutSail;
    }

    @Step("Sail Included (button)")

```

```

public BuyItemButton getSailIncluded() {
    return SailIncluded;
}

public class BuyItemButton extends WebElement {
    private WebElement price;

    private BuyItemButton(SelenideElement element) {
        super(element);
        this.price = new
WebElement(this.widget().$(By.xpath("span")));
    }

    @Step("price")
    public WebElement getPrice() {
        return price;
    }

    public BuyItemButton checkPrice(String price) {
        getPrice().shouldHave(text(price));
        return this;
    }
}

}

}

public class UserInfo extends WebElement {
    private Input name;
    private Input email;
    private Input phoneNumber;
    private Input address;

    private UserInfo() {
        super("//div[@class='user-info']");
        this.name = new
Input(this.widget().$(By.xpath("div//label[text()='Name']/following-
sibling::input")));
        this.email = new
Input(this.widget().$(By.xpath("div//label[text()='Email']/following-
sibling::input")));
        this.phoneNumber = new
Input(this.widget().$(By.xpath("div//label[text()='Phone number']/following-
sibling::input")));
        this.address = new
Input(this.widget().$(By.xpath("div//label[text()='Address']/following-
sibling::input")));
    }

    @Step("Name")
    public Input getName() {
        return name;
    }
    @Step("Email")
    public Input getEmail() {
        return email;
    }
}

```

```

@Step("Phone number")
public Input getPhoneNumber() {
    return phoneNumber;
}

@Step("Address")
public Input getAddress() {
    return address;
}

public void fill(String name, String email, String phoneNumber, String
address) {
    getName().setValue(name);
    getEmail().setValue(email);
    getPhoneNumber().setValue(phoneNumber);
    getAddress().setValue(address);
}

}

public class BuyResult extends WebElement {
    private WebElement nights;
    private WebElement price;
    private WebElement planType;
    private WebElement taxIndicator;
    private Button cancel;
    private Button BuyNow;

    private BuyResult() {
        super($(By.xpath("//div[@class='Order-form-footer between Order-
info-final']")));
        this.nights = new
WebElement(this.widget().$(By.xpath("div//div[@class='Order-nights']")));
        this.price = new
WebElement(this.widget().$(By.xpath("div//div[@class='price']")));
        this.planType = new
WebElement(this.widget().$(By.xpath("div//div[@class='plan-type']")));
        this.taxIndicator = new
WebElement(this.widget().$(By.xpath("div//div[@class='tax-indicator']")));
        this.cancel = new Button(this.widget().$(By.xpath("following-
sibling::div//div[text()='Back']/../.."))); {

            @Override
            public void click() {
                super.click();
                OrderForm.this.userInfo.widget().waitUntil(hidden,
Delay.LOW_2000.getTime());
                OrderForm.this.BuyResult.widget().shouldBe(hidden);
                OrderForm.this.planInfo = new PlanInfo();
            }
        };
        this.BuyNow = new Button(this.widget().$(By.xpath("following-
sibling::div//div[text()='Buy now']/../..")));
    }

    @Step("Amount ")
    public WebElement getNights() {
        return nights;
    }
}

```

```

    @Step("Size")
    public WebElement getTaxIndicator() {
        return taxIndicator;
    }

    @Step("Price")
    public WebElement getPrice() {
        return price;
    }

    @Step("Delivery type")
    public WebElement getPlanType() {
        return planType;
    }

    @Step("Back")
    public Button getCancel() {
        return cancel;
    }

    @Step("Buy Now")
    public Button getBuyNow() {
        return BuyNow;
    }

    public BuyResult check(String planType, String price, int nights,
String taxIndicator) {
        getPlanType().shouldHave(text(planType));
        getPrice().shouldHave(text(price));
        if (nights == 1) {
            getNights().shouldHave(exactText(nights + " night"));
        } else {
            getNights().shouldHave(exactText(nights + " nights"));
        }
        getTaxIndicator().shouldHave(text(taxIndicator));
        return this;
    }
}
}
}

//Модуль тестування входу з логіном адміністратора
package project.admin;

import com.codeborne.selenide.Selenide;
import core.control.Domen;
import io.qameta.allure.Step;
import project.LoginForm;

import static core.control.application_tools.LocalStorage.*;

public class AdminLoginPage {
    private LoginForm loginForm;

    public AdminLoginPage() {
        render();
    }
}

```

```
@Step("open [Admin Login page]")
public static AdminLoginPage open() {
    return Selenide.open(AdminLoginPage.url(), AdminLoginPage.class);
}

public static String url() {
    return Domen.ADMIN.getUrl().concat("login");
}

public void render() {
    if (getStorage().getItem("locale") == null) {
        getStorage().setItem("locale", "en");
        Selenide.refresh();
    }
    this.loginForm = new LoginForm();
    loginForm.render();
}

public LoginForm getLoginForm() {
    return loginForm;
}

public void login(String email, String password) {
    getLoginForm().getEmail().setValue(email);
    getLoginForm().getPassword().setValue(password);
    getLoginForm().getSubmit().click();
    new ClientsPage();
}
}
```

```

package project.admin;

import core.control.Delay;
import core.main_web_elements.*;
import io.qameta.allure.Step;

import static com.codeborne.selenide.Condition.visible;
import static com.codeborne.selenide.Selenide.$x;

public class AddNewClientPopup extends WebElement {
    private WebElement title;
    private Button btnClose;
    private Input companyName;
    private Input username;
    private Input email;
    private Input phoneNumber;
    private Input googleDisplayNameEn;
    private Input googleDisplayNameJp;
    private Selector source;
    private Selector paymentSystem;
    private Checkbox sendInvitationEmail;
    private Button addNewClient;

    private Checkbox USProductAssociationMemberProducts;
    private Input tciProductOrderalOrderCommission;
    private Button next;
    private LegalInformationJa legalInformationJa;
    private LegalInformationEn legalInformationEn;

    public static AddNewClientPopup render() {
        $x("//span[text()='Add New Client']/../..").waitUntil(visible,
Delay.MEDIUM.getTime());
        return new AddNewClientPopup();
    }

    private AddNewClientPopup() {
        super("//span[text()='Add New Client']/../..");
        this.title = new WebElement("//span[text()='Add New Client']");
        this.btnClose = new Button(this.widget().$x("span/a"));
        this.companyName = new Input(this.widget().$x("div//label[text()='Company
Name']/following-sibling::input"));
        this.username = new
Input(this.widget().$x("div//label[text()='Username']/following-sibling::input"));
        this.email = new
Input(this.widget().$x("div//label[text()='Email']/following-sibling::input"));
        this.phoneNumber = new Input(this.widget().$x("div//label[text()='Phone
number']/following-sibling::input"));
        this.googleDisplayNameEn = new Input(this.widget()
.$x("div//label[text()='Google display name (ua)']/following-
sibling::input"));
        this.googleDisplayNameJp = new Input(this.widget()
.$x("div//label[text()='Google display name (ua)']/following-
sibling::input"));
        this.source = new
Selector(this.widget().$x("div//label[text()='Source']/following-sibling::div"));
        this.paymentSystem = new
Selector(this.widget().$x("div//label[text()='Payment system']/following-
sibling::div")) {

            @Override

```

```

        public void select(String option) {
            super.select(option);
            AddNewClientPopup.this.renderPaymentSystem();
        }
    };
    renderPaymentSystem();
    this.sendInvitationEmail = new
Checkbox(this.widget().$x("//div/label[text()='Send invitation email']/.."));
    this.addNewClient = new
Button(this.widget().$x("div//button//div[text()='Add New Client']/../.."));
}

private void renderPaymentSystem() {
    switch (paymentSystem.getInput().widget().getValue()) {
        case VisaData.PAYMENT_SYSTEM :
            AddNewClientPopup.this.USProductAssociationMemberProducts = new
Checkbox(
                "//label[text()='US Product Association member Products
(Visa fee 2.5%)]/.."");
                break;
        case VisaData.PAYMENT_SYSTEM :
            AddNewClientPopup.this.tciProductOrderalOrderCommission = new
Input(AddNewClientPopup.this.widget()
                .$x("div//label[text()='TciProductOrderal Order
Commission']/following-sibling::input"));
                break;
        case VisaData.PAYMENT_SYSTEM :
            AddNewClientPopup.this.tciProductOrderalOrderCommission = new
Input(AddNewClientPopup.this.widget()
                .$x("div//label[text()='TciProductOrderal Order
Commission']/following-sibling::input"));
            AddNewClientPopup.this.next = new
Button(AddNewClientPopup.this.widget()
                .$x("div//button//div[text()='Next']/../.."));
                break;
        default:
            break;
    }
}

public void renderLegalInformationJa() {
    this.legalInformationJa = new LegalInformationJa();
}

public void renderLegalInformationEn() {
    this.legalInformationEn = new LegalInformationEn();
}

@Step("Title")
public WebElement getTitle() {
    return title;
}

@Step("X (close)")
public Button getBtnClose() {
    return btnClose;
}

@Step("Company name")
public Input getCompanyName() {
    return companyName;
}

```

```
@Step("Username")
public Input getUsername() {
    return username;
}

@Step("Email")
public Input getEmail() {
    return email;
}

@Step("Phone Number")
public Input getPhoneNumber() {
    return phoneNumber;
}

@Step("Google Display Name (ua)")
public Input getGoogleDisplayNameEn() {
    return googleDisplayNameEn;
}

@Step("GoogleDisplayName (ua)")
public Input getGoogleDisplayNameJp() {
    return googleDisplayNameJp;
}

@Step("Source")
public Selector getSource() {
    return source;
}

@Step("Payment system")
public Selector getPaymentSystem() {
    return paymentSystem;
}

@Step("Send Invitation Email")
public Checkbox getSendInvitationEmail() {
    return sendInvitationEmail;
}

@Step("Add New Client")
public Button getAddNewClient() {
    return addNewClient;
}

@Step("US Product Association Member Products")
public Checkbox getUSProductAssociationMemberProducts() {
    return USProductAssociationMemberProducts;
}

@Step("TciProductOrderal Order Commission")
public Input getTciProductOrderalOrderCommission() {
    return tciProductOrderalOrderCommission;
}

@Step("Next")
public Button getNext() {
    return next;
}

@Step("Legal Information (ua)")
public LegalInformationJa getLegalInformationJa() {
    return legalInformationJa;
}
```

```
}
```

```
@Step("Legal Information (ua)")
public LegalInformationEn getLegalInformationEn() {
    return legalInformationEn;
}
```

```
//Модуль тестування полів форми оплати
```

```
public AddNewClientPopup fill(String companyName, String username, String
email, String phoneNumber,
                                String googleDisplayNameEn, String
googleDisplayNameJp, String source,
                                Payment payment, boolean isSendInvitationEmail) {

    getCompanyName().setValue(companyName);
    getUsername().setValue(username);
    getEmail().setValue(email);
    getPhoneNumber().setValue(phoneNumber);
    getGoogleDisplayNameEn().setValue(googleDisplayNameEn);
    getGoogleDisplayNameJp().setValue(googleDisplayNameJp);
    getSource().select(source);

    if (payment instanceof VisaData) {
        getPaymentSystem().select(VisaData.PAYMENT_SYSTEM);
        getUSProductAssociationMemberProducts().setFlag(((VisaData)
payment).isUSProductAssociationMemberProducts());
        getSendInvitationEmail().setFlag(isSendInvitationEmail);
    } else if (payment instanceof VisaData) {
        getPaymentSystem().select(VisaData.PAYMENT_SYSTEM);
        getTciProductOrderalOrderCommission().setValue(((VisaData)
payment).tciProductOrderalOrderCommission());
        getSendInvitationEmail().setFlag(isSendInvitationEmail);
    } else if (payment instanceof VisaData) {
        getPaymentSystem().select(VisaData.PAYMENT_SYSTEM);
        getTciProductOrderalOrderCommission().setValue(((VisaData)
payment).tciProductOrderalOrderCommission());
        if (((VisaData) payment).legalInformationJpData != null) {
            getNext().click();
            renderLegalInformationJa();
            getLegalInformationJa().fill(
                ((VisaData) payment).legalInformationJpData.seller,
                ((VisaData) payment).legalInformationJpData.representative,
                ((VisaData) payment).legalInformationJpData.address,
                ((VisaData) payment).legalInformationJpData.sellingPrice,
                ((VisaData)
payment).legalInformationJpData.additionalSurcharges,
                ((VisaData)
payment).legalInformationJpData.paymentPeriodAndMethod,
                ((VisaData) payment).legalInformationJpData.deliveryTime,
                ((VisaData) payment).legalInformationJpData.returns,
                ((VisaData) payment).legalInformationJpData.cancellation,
                ((VisaData) payment).legalInformationJpData.phoneNumber,
                ((VisaData) payment).legalInformationJpData.email
            );
        }
        if (((VisaData) payment).legalInformationEnData != null) {
            getNext().click();
            renderLegalInformationEn();
            getLegalInformationEn().fill(
                ((VisaData) payment).legalInformationEnData.seller,
                ((VisaData)
payment).legalInformationEnData.representative,
                ((VisaData) payment).legalInformationEnData.address,
```

```

        ((VisaData)
payment).legalInformationEnData.sellingPrice,
        ((VisaData)
payment).legalInformationEnData.additionalSurcharges,
        ((VisaData)
payment).legalInformationEnData.paymentPeriodAndMethod,
        ((VisaData)
payment).legalInformationEnData.deliveryTime,
        ((VisaData) payment).legalInformationEnData.returns,
        ((VisaData)
payment).legalInformationEnData.cancellation
    );
    getSendInvitationEmail().setFlag(isSendInvitationEmail);
}
}
}
return this;
}

public VisaData fillVisaData(boolean isUSProductAssociationMemberProducts) {
    return new VisaData(isUSProductAssociationMemberProducts);
}

public VisaData fillVisaData(String tciProductOrderalOrderCommission) {
    return new VisaData(tciProductOrderalOrderCommission);
}

public VisaData fillVisaData(String tciProductOrderalOrderCommission,
LegalInformationJpData legalInformationJpData,
        LegalInformationEnData legalInformationEnData) {
    return new VisaData(tciProductOrderalOrderCommission,
legalInformationJpData, legalInformationEnData);
}

public LegalInformationJpData fillLegalInformationJpData(String seller, String
representative, String address,
        String phoneNumber,
String email, String sellingPrice,
        String
additionalSurcharges, String paymentPeriodAndMethod,
        String deliveryTime,
String returns, String cancellation) {
    return new LegalInformationJpData(seller, representative, address,
phoneNumber, email, sellingPrice,
        additionalSurcharges, paymentPeriodAndMethod, deliveryTime,
returns, cancellation);
}

public LegalInformationEnData fillLegalInformationEnData(String seller, String
representative, String address,
        String sellingPrice,
String additionalSurcharges,
        String
paymentPeriodAndMethod, String deliveryTime,
        String returns, String
cancellation) {
    return new LegalInformationEnData(seller, representative, address,
sellingPrice,
        additionalSurcharges, paymentPeriodAndMethod, deliveryTime,
returns, cancellation);
}

```

```
}

```

```
public class LegalInformationEn extends LegalInformation {

```

```
    private LegalInformationEn() {
        super("//span[text()='Legal Information (ua)']/../..");
        AddNewClientPopup.this.sendInvitationEmail = new Checkbox(this.widget()
            .$x("//div/label[text()='Send invitation email']/.."));
        AddNewClientPopup.this.addNewClient = new Button(this.widget()
            .$x("div//button//div[text()='Add New Client']/../.."));
    }

```

```
@Override

```

```
public Button getBack() {
    return new Button(super.getBack().widget()) {

```

```
        @Override

```

```
        public void click() {

```

```
            super.click();

```

```
            AddNewClientPopup.this.renderLegalInformationJa();

```

```
        }

```

```
    };

```

```
    }

```

```
}

```

```
public class LegalInformationJa extends LegalInformation {

```

```
    private Input phoneNumber;

```

```
    private Input email;

```

```
    private LegalInformationJa() {

```

```
        super("//span[text()='Legal Information (ua)']/../..");

```

```
        this.phoneNumber = new Input(this.widget().$x("div//label[text()='Phone number']/following-sibling::input"));

```

```
        this.email = new

```

```
Input(this.widget().$x("div//label[text()='Email']/following-sibling::input"));

```

```
        AddNewClientPopup.this.next = new

```

```
Button(this.widget().$x("div//button//div[text()='Next']/../.."));

```

```
    }

```

```
@Step("Phone Number")

```

```
public Input getPhoneNumber() {

```

```
    return phoneNumber;

```

```
    }

```

```
@Step("Email")

```

```
public Input getEmail() {

```

```
    return email;

```

```
    }

```

```
@Override

```

```
public Button getBack() {

```

```
    return new Button(super.getBack().widget()) {

```

```
        @Override

```

```
        public void click() {

```

```
            super.click();

```

```

        AddNewClientPopup.render();
    }
};
}

    public void fill(String seller, String representative, String address,
String sellingPrice,
                    String additionalSurcharges, String
paymentPeriodAndMethod, String deliveryTime,
                    String returns, String cancellation, String phoneNumber,
String email) {

        super.fill(seller, representative, address, sellingPrice,
additionalSurcharges, paymentPeriodAndMethod,
                    deliveryTime, returns, cancellation);
        getPhoneNumber().setValue(phoneNumber);
        getEmail().setValue(email);
    }
}

public static class LegalInformation extends WebElement {
    private Input seller;
    private Input representative;
    private Input address;
    private Input sellingPrice;
    private Input additionalSurcharges;
    private Input paymentPeriodAndMethod;
    private Input deliveryTime;
    private Input returns;
    private Input cancellation;
    private Button back;

    public LegalInformation(String xpath) {
        super(xpath);
        this.seller = new
Input(this.widget().$x("div//label[text()='Seller']/following-sibling::input"));
        this.representative = new
Input(this.widget().$x("div//label[text()='Representative']/following-
sibling::input"));
        this.address = new
Input(this.widget().$x("div//label[text()='Address']/following-sibling::input"));
        this.sellingPrice = new
Input(this.widget().$x("div//label[text()='Selling price']/following-
sibling::input"));
        this.additionalSurcharges = new
Input(this.widget().$x("div//label[text()='Additional Surcharges']/following-
sibling::input"));
        this.paymentPeriodAndMethod = new
Input(this.widget().$x("div//label[text()='Payment Period and Method']/following-
sibling::textarea"));
        this.deliveryTime = new
Input(this.widget().$x("div//label[text()='Delivery Time']/following-
sibling::input"));
        this.returns = new
Input(this.widget().$x("div//label[text()='Returns']/following-sibling::input"));
        this.cancellation = new
Input(this.widget().$x("div//label[text()='Cancellation']/following-
sibling::textarea"));
        this.back = new
Button(this.widget().$x("div//button//div[text()='Back']/../.."));

```

```

    }

    @Step("Seller")
    public Input getSeller() {
        return seller;
    }

    @Step("Representative")
    public Input getRepresentative() {
        return representative;
    }

    @Step("Address")
    public Input getAddress() {
        return address;
    }

    @Step("Selling Price")
    public Input getSellingPrice() {
        return sellingPrice;
    }

    @Step("Additional Surcharges")
    public Input getAdditionalSurcharges() {
        return additionalSurcharges;
    }

    @Step("Payment Period And Method")
    public Input getPaymentPeriodAndMethod() {
        return paymentPeriodAndMethod;
    }

    @Step("Delivery Time")
    public Input getDeliveryTime() {
        return deliveryTime;
    }

    @Step("Returns")
    public Input getReturns() {
        return returns;
    }

    @Step("Cancellation")
    public Input getCancellation() {
        return cancellation;
    }

    @Step("Back")
    public Button getBack() {
        return back;
    }

    public void fill(String seller, String representative, String address,
String sellingPrice,
                    String additionalSurcharges, String
paymentPeriodAndMethod, String deliveryTime,
                    String returns, String cancellation) {

        getSeller().setValue(seller);
        getRepresentative().setValue(representative);
        getAddress().setValue(address);
        getSellingPrice().setValue(sellingPrice);

```

```

        getAdditionalSurcharges().setValue(additionalSurcharges);
        getPaymentPeriodAndMethod().setValue(paymentPeriodAndMethod);
        getDeliveryTime().setValue(deliveryTime);
        getReturns().setValue(returns);
        getCancellation().setValue(cancellation);
    }
}

public interface Payment {}

private static class VisaData implements Payment {
    static final String PAYMENT_SYSTEM = "Visa";
    boolean isUSProductAssociationMemberProducts;

    private VisaData(boolean isUSProductAssociationMemberProducts) {
        this.isUSProductAssociationMemberProducts =
isUSProductAssociationMemberProducts;
    }
}

private static class VisaData implements Payment {
    static final String PAYMENT_SYSTEM = "VISA";
    String tciProductOrderalOrderCommission;

    private VisaData(String tciProductOrderalOrderCommission) {
        this.tciProductOrderalOrderCommission =
tciProductOrderalOrderCommission;
    }
}

private static class VisaData implements Payment {
    static final String PAYMENT_SYSTEM = "VISA";
    String tciProductOrderalOrderCommission;
    LegalInformationJpData legalInformationJpData;
    LegalInformationEnData legalInformationEnData;

    public VisaData(String tciProductOrderalOrderCommission,
LegalInformationJpData legalInformationJpData,
        LegalInformationEnData legalInformationEnData) {

        this.tciProductOrderalOrderCommission =
tciProductOrderalOrderCommission;
        this.legalInformationJpData = legalInformationJpData;
        this.legalInformationEnData = legalInformationEnData;
    }
}

private static class LegalInformationEnData {
    String seller;
    String representative;
    String address;
    String sellingPrice;
    String additionalSurcharges;
}

```

```
String paymentPeriodAndMethod;
String deliveryTime;
String returns;
String cancellation;

    public LegalInformationEnData(String seller, String representative, String
address, String sellingPrice,
                                String additionalSurcharges, String
paymentPeriodAndMethod, String deliveryTime,
                                String returns, String cancellation) {
        this.seller = seller;
        this.representative = representative;
        this.address = address;
        this.sellingPrice = sellingPrice;
        this.additionalSurcharges = additionalSurcharges;
        this.paymentPeriodAndMethod = paymentPeriodAndMethod;
        this.deliveryTime = deliveryTime;
        this.returns = returns;
        this.cancellation = cancellation;
    }
}

private static class LegalInformationJpData extends LegalInformationEnData {
    String phoneNumber;
    String email;

    public LegalInformationJpData(String seller, String representative, String
address, String phoneNumber,
                                String email, String sellingPrice, String
additionalSurcharges,
                                String paymentPeriodAndMethod, String
deliveryTime, String returns,
                                String cancellation) {

        super(seller, representative, address, sellingPrice,
additionalSurcharges, paymentPeriodAndMethod,
              deliveryTime, returns, cancellation);
        this.phoneNumber = phoneNumber;
        this.email = email;
    }
}
}
```