МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп’ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
_____ Ігор ШЕЛЕХОВ
(підпис)
_____ грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп’ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна технологія автоматизованого тестування бізнес-процесів на основі Camunda та jMeter»
здобувача групи ІН.м - 26 Олексіча Дмитра Володимировича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Дмитро ОЛЕКСІЧ
(підпис)

Керівник,
старший викладач,
кандидат технічних наук Олег БЕРЕСТ

_____
(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри
_____ Ігор ШЕЛЕХОВ
(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-26 Олексіча Дмитра Володимировича

1. Тема роботи: «Інформаційна технологія автоматизованого тестування бізнес-процесів на основі Camunda та jMeter»
затверджую наказом по СумДУ від *«06» грудня 2023 року № 1412-VI*
2. Термін здачі здобувачем кваліфікаційної роботи *до 16 грудня 2023 року*
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження. 2) Аналіз об'єкту дослідження. 3) Розробка інформаційного та програмного забезпечення для тестування бізнес-процесів. 4) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «____» _____ 20 ___ р.
Завдання прийняв до виконання _____ Керівник _____
(підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	*Аналіз проблеми предметної області, постановка й формування завдань дослідження*		
2	*Аналіз об'єкту дослідження*		
3	*Розробка інформаційного та програмного забезпечення для тестування бізнес-процесів*		
4	*Аналіз отриманих результатів*		
5	*Оформлення пояснювальної записки до кваліфікаційної роботи*		

Здобувач вищої освіти _____ Керівник _____
(підпис) (підпис)

АНОТАЦІЯ

Записка: 71 стр., 33 рис., 4 таблиці, 1 додаток, 43 використаних джерел.

Обґрунтування актуальності теми роботи – автоматизація бізнес-процесів є невід’ємною складовою підвищення ефективності, а якісне тестування спрямоване на дотримання стандартів якості, що є запорукою успішного функціонування та розвитку бізнесу.

Об’єкт дослідження — інформаційна технологія оркестрації бізнес-процесів соціальної мережі.

Мета роботи — створення інформаційної технології автоматизованого тестування бізнес-процесів на основі Camunda та JMeter.

Методи дослідження — методи побудови інформаційних систем.

Результати — було спроектовано та реалізовано інформаційну систему автоматизованого тестування бізнес-процесів на основі Camunda та JMeter. Під час виконання завдання було проаналізовано підходи до моделювання бізнес-процесів та засади використання BPMN при вирішенні даної задачі. Розглянуто прототипування бізнес-процесів за допомогою Camunda та його роль у процесі розробки під час донесення технічних задач до команди розробки. Досліджено технічні аспекти оркестрації бізнес-процесів соціальної мережі, налаштування CI/CD та місце автоматизованого тестування у цьому процесі. На базі результатів дослідження також розроблено сервіс з оркестрації бізнес-процесів в системі на базі інтеграції коду SpringBoot та графічних нотацій Camunda, а також автоматизовані тести для тестування роботи даного сервісу за допомогою jMeter.

Інформатика, бізнес-процеси, BPMN, Camunda, jMeter

Зміст

Вступ

Обґрунтування вибору теми роботи. Автоматизація бізнес-процесів дозволяє підприємствам оптимізувати та прискорювати виконання завдань, що призводить до підвищення загальної ефективності операцій. Автоматизованим системам властива менша кількість помилок, які можуть виникнути при ручному виконанні завдань, що є критичним для точності та надійності бізнес-процесів. Такі системи легше адаптуються до змін у бізнес-процесах, забезпечуючи гнучкість та швидкість реакції на нові вимоги ринку чи внутрішні зміни. Таким чином,

Актуальність. Зростання популярності систем управління бізнес-процесами (BPM) та автоматизації бізнес-процесів свідчать про важливість цього напрямку для підприємств. Розгортання автоматизованих систем дозволяє оптимізувати роботу, зменшувати помилки та швидше адаптуватися до змін у бізнес-середовищі.

Зважаючи на те, що окрестрація автоматизованих бізнес-процесів є важливою складовою у забезпеченні якості продукту, тестування такої оркестрації є не менш важливою, адже останнє є невід'ємною частиною у забезпеченні якості. Тестування дозволяє перевірити, чи працюють бізнес-процеси відповідно до очікувань і чи вони надійні в умовах реального використання, а також виявити потенційні помилки та недоліки у роботі системи, забезпечуючи їх виправлення на ранніх етапах розробки. Важливо також приділяти достатньо уваги автоматизації тесутвання, адже це прискорює цикл розробки та допомагає впроваджувати нові функції та зміни швидше, що спрощує завдання супроводу, оскільки швидко виявляє та вирішує проблеми без значного втручання користувача.

Таким чином, автоматизоване тестування оркестрації бізнес-процесів є ключовим елементом забезпечення високої якості та надійності роботи

системи, що є критичним для успішного функціонування та розвитку бізнесу

Об’єкт дослідження. Процес оркестрації бізнес-процесів соціальної мережі.

Предмет дослідження. Інформаційна технологія автоматзації тестування бізнес-процесів на основі Camunda та jMeter.

Гіпотеза. Оркестрація процесів дозволяє створювати єдиний шар інтеграції для всіх рішень, що спрощує управління та підтримку інфраструктури. Засоби оркестрації забезпечують можливості контролю та моніторингу виконання бізнес-процесів, що є ключовим для ефективного управління та вчасного реагування на зміни.

Новизна. Формування технічних завдань для реалізації рішень з автоматизації бізнес-процесів є доволі неординарною задачею. Поєднання нотацій для аналітики бізнес-процесів із програмним кодом забезпечує зручний інструмент для взаємодії бізнес-аналітиків та розробників, покращуючи розуміння та комунікацію між цими групами. Нотації дозволяють стандартизувати процеси, що призводить до підвищення якості автоматизації та легку інтеграцію автоматизованих систем.

Структура. Дане робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв’язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1 ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

1.1 Підходи до моделювання бізнес-процесів

Сучасні підприємства використовують процесний підхід до управління, що вимагає використання спеціальних технологій для побудови бізнес-систем. Це означає, що процеси повинні бути чітко визначені та керовані, щоб забезпечити оптимальну продуктивність та досягти цілей підприємства [1].

Оцінка та моделювання бізнес-систем є важливою частиною ефективного управління бізнес-процесами. Вони дозволяють визначити конкретні процеси, необхідні для досягнення цілей підприємства, та забезпечити їх ефективність. Моделювання бізнес-процесів створює можливість попередньо проаналізувати їхню ефективність та визначити шляхи оптимізації [2].

Завдяки моделюванню бізнес-процесів, підприємства можуть більш точно визначити потреби в ресурсах, часі та фінансах. Це сприяє більш точному плануванню та керуванню діяльністю підприємства [3-5].

Цифрові технології та програмне забезпечення для моделювання бізнес-процесів допомагають створити віртуальну модель підприємства, на якій можна проводити різноманітні експерименти та визначати найкращі рішення [4-8]. Це полегшує прийняття обґрунтованих рішень та забезпечує конкурентні переваги на ринку.

Моделювання бізнес-процесів стає ключовим інструментом для розробки та вдосконалення стратегії підприємства. Воно допомагає адаптувати процеси до змінних ринкових умов та збільшити ефективність діяльності, забезпечуючи стійку конкурентоспроможність.

Розглянемо різні підходи щодо розуміння бізнес-процесів, наведені в табл. 1.1.

Таблиця 1.1 — Підходи до визначення поняття «бізнес-процеси»

Автори	Визначення
1	2
Біннер Х.	Система взаємопов’язаних дій, кінцевими результатами здійснення яких є виробництво продуктів/послуг, що становлять цінність для зовнішніх і внутрішніх споживачів
Дейвенпорт Т., Шорт Дж.	Структурована безліч вимірюваних дій, що спроектовані для виробництва специфічної послуги або продукту для конкретного споживача або ринку. Включає в себе роботи, завдання впорядковані в просторі та часі з наявністю визначених ««входів» та «виходів»
Чорнобай Л. І., Дума О. І.	Система безперервних, пов’язаних між собою, впорядкованих та керованих дій, яка є елементом механізму формування доданої вартості в процесі перетворення ресурсів підприємства, спрямована на забезпечення високої продуктивності та ефективності організації загалом та забезпечення реалізації споживчої цінності для цільового ринку, застосовуючи бізнес-модель підприємства
ISO 9001	Стійка, цілеспрямована сукупність взаємопов’язаних видів діяльності, яка за визначеною технологією трансформує вхід у вихід

Продовження табл. 1.1

1	2
Жмиков Я.	це логічна послідовність дій людини чи групи людей в компанії або комерційній структурі
Крижановський Є. М., Ящолт А.Р., Жуков С.О., Козачко О. М.	будь-яка діяльність, що має вхідний продукт, додає вартість до нього, та забезпечує вихідний продукт для внутрішнього або зовнішнього споживача

Джерело: складено та доповнено автором на основі [5; 9-15]

За наведеними визначеннями вище, бізнес-процес можна розглядати як систему дій і операцій, які мають структурований характер і спрямовані на виробництво продуктів або послуг, які є цінністю для зовнішніх і внутрішніх споживачів. Основні характеристики бізнес-процесу включають [16-18]:

1. ***Система дій***: бізнес-процес включає в себе послідовність дій і операцій, які відбуваються в певному порядку.
2. ***Вхід та вихід***: У бізнес-процесі є чітко визначені "входи" і "виходи", що вказують на те, які ресурси або дані використовуються та які результати виробляються в процесі.
3. ***Трансформація ресурсів***: бізнес-процес трансформує вхідні ресурси чи дані в вихідні продукти чи послуги.
4. ***Мета***: Головною метою бізнес-процесу є створення цінності для споживачів, які можуть бути як зовнішніми клієнтами, так і внутрішніми відділами чи стейкхолдерами.
5. ***Ефективність і продуктивність***: бізнес-процес спрямований на досягнення високої продуктивності та ефективності, а також на забезпечення високої якості продуктів чи послуг за рахунок: зменшення помилок та ризику; покращення операційних процесів; підвищення інтеграції та якості керування процесами.

6. ***Логічна послідовність***: бізнес-процес включає логічну послідовність дій, які можуть бути виконані людьми або групами людей в компанії або організації.
7. ***Додана вартість***: бізнес-процес додає вартість до вхідних продуктів чи послуг і забезпечує вихідний продукт для споживача.

Зрозуміння сутності бізнес-процесу важливе для ефективного управління організацією, оптимізації процесів і досягнення бажаних результатів. бізнес-процеси можуть бути моделювані, аналізовані і автоматизовані для поліпшення продуктивності та якості виробництва або надання послуг.

Розглядаючи бізнес-процеси, необхідно звернути увагу на потребу у прототипуванні бізнес-процесів, яка полягає в кількох ключових аспектах [4; 8; 14; 16; 19]:

- ***Візуалізація***: Прототипування дозволяє створити графічне представлення бізнес-процесу, що робить його легко зрозумілим для всіх стейкхолдерів, незалежно від їхнього рівня технічної експертизи. Графічні моделі набагато доступніші, ніж технічна документація.
- ***Тестування***: Прототипи дозволяють виконувати тестування бізнес-процесу в контрольованому середовищі. Це допомагає виявляти помилки та неефективності ще до впровадження процесу в реальному середовищі, що може зекономити час і ресурси.
- ***Залучення стейкхолдерів***: Прототипи є ефективним інструментом для залучення різних стейкхолдерів, таких як керівництво, клієнти та робочі групи, до процесу розробки. Вони можуть бачити, як буде виглядати процес і надавати свої коментарі та пропозиції.
- ***Оптимізація***: Прототипування дозволяє ідентифікувати можливості для оптимізації бізнес-процесу, включаючи скорочення часу

виконання завдань, зменшення надмірних кроків та раціоналізацію використання ресурсів.

- ***Покращення розуміння***: Прототипи сприяють кращому розумінню бізнес-процесу серед учасників проекту. Вони допомагають уникнути невідповідностей між очікуваннями та реальними результатами.
- ***Підготовка документації***: Графічні моделі, створені під час прототипування, можуть служити основою для документації процесу, що допомагає у збереженні інформації та забезпеченні зрозумілості.

Усі ці аспекти допомагають покращити ефективність, продуктивність та якість бізнес-процесів і забезпечують їхню відповідність бізнес-потребам та стратегії організації. Прототипування є важливою частиною процесу управління бізнес-процесами і розробки нових процесів.

Існує кілька підходів до моделювання бізнес-процесів, і вибір конкретного підходу залежить від потреб організації та мети моделювання. Ось декілька загальних підходів [4; 8; 14; 16; 19]:

- *Бізнес-процеси як системи*: Цей підхід розглядає бізнес-процеси як системи, де кожен процес має вхідні та вихідні дані, ресурси та операторів. Модель описує, як ці компоненти взаємодіють для досягнення певних результатів.
- *Подійно-орієнтоване моделювання*: Цей підхід розглядає бізнес-процеси через призму подій і реакцій на ці події. Модель включає в себе події, стани і дії, що відбуваються під час процесу.
- *Потокові діаграми (Flowcharts)*: Це графічний підхід до моделювання бізнес-процесів. Використовуються стрілки, прямокутники, ромби тощо для відображення послідовності дій у процесі.

- *BPMN (Business Process Model and Notation)*: Ця стандартна нотація спеціально розроблена для моделювання бізнес-процесів. Вона включає в себе символи для представлення задач, подій, вороття, рішень і багато іншого.
- *UML (Unified Modeling Language)*: Хоча UML в основному використовується для моделювання програмних систем, його можна також застосовувати для бізнес-процесів, використовуючи діаграми активностей і послідовностей.
- *Процеси, орієнтовані на взаємодію*: Цей підхід розглядає процеси як послідовність взаємодій між стейкхолдерами та системами. Модель визначає, як відбувається ця взаємодія.
- *Математичне моделювання*: Деякі бізнес-процеси можуть бути модельовані за допомогою математичних рівнянь і алгоритмів для аналізу та оптимізації продуктивності.
- *Децентралізоване моделювання*: В цьому підході бізнес-процеси моделюються та оптимізуються на рівні окремих підрозділів чи відділів організації.

Вибір конкретного підходу залежить від контексту і мети моделювання бізнес-процесу. Організації можуть комбінувати різні підходи для кращого розуміння, оптимізації та управління своїми процесами.

1.2 Система графічних нотацій та моделей (BPMN)

Моделювання бізнес-процесів — це важливий етап у розробці та удосконаленні бізнес-процесів [8]. Моделювання допомагає визначити реальні вимоги клієнта та стейкхолдерів до бізнес-процесу, що дозволяє уникнути непорозумінь та забезпечити, що розроблений процес відповідає потребам користувачів [16]. Прототипи допомагають візуалізувати бізнес-

процес та зробити його більш зрозумілим для всіх учасників проекту. Вони дозволяють стейкхолдерам побачити, як процес буде функціонувати в реальному житті [19]. Прототипи дозволяють швидко виявити потенційні проблеми та недоліки в процесі. Це допомагає уникнути серйозних помилок та витрат у майбутньому. Вони створюють спільну мову між різними стейкхолдерами проекту, що полегшує співпрацю та розв'язання конфліктів [19].

Створення моделей та прототипів дозволяє розробникам отримувати зворотний зв'язок від клієнта та вдосконалювати процес до того, як він буде повністю розроблений. Учасники розробки можуть проводити тестування та валідацію ідей щодо бізнес-процесу, не витрачаючи багато ресурсів. Це може скоротити час розробки та зменшити ризик невдач. Безумовною перевагою ще є той факт, що прототипи можуть бути легко змінені та адаптовані на ранніх етапах розробки, що дозволяє вносити корективи до процесу на ранніх етапах, коли це ще недорого.

Узагальнюючи наведені аргументи, ми можемо впевнено стверджувати, що моделювання та прототипування бізнес-процесів сприяє покращенню розробки та управлінню процесами, забезпечуючи візуалізацію, тестування та оптимізацію процесу на ранніх етапах проекту.

На сьогодні досить поширена практика прототипування бізнес-процесів за допомогою нотацій, таких як Business Process Model and Notation (BPMN) або Unified Modeling Language (UML), що є важливим етапом при розробці та вдосконаленні бізнес-процесів. BPMN — це стандарт моделювання процесів, який має в основі послідовність подій та дій, які мають початок та кінець [10]. Цей стандарт був розроблений для того, щоб стандартизувати способи представлення бізнес-процесів та зрозуміти їхню структуру та функціональність. BPMN надає способи графічного зображення бізнес-процесів за допомогою символів та нотацій, що є загальними для всіх

учасників проекту [19]. Графічні моделі дозволяють легко розуміти та сприяють візуалізації кожного кроку процесу, що, в свою чергу, дозволяє ідентифікувати можливі проблеми, помилки та неефективності до впровадження процесу в реальному середовищі.

Наочне представлення також сприяє легкості розуміння не тільки спеціалістами, але й стейкхолдерами, які можуть внести свої пропозиції та коментарі щодо процесу на ранніх етапах розробки. Це дозволяє розширити коло осіб, які беруть участь у обговоренні та долучити учасників з різними доменними знаннями, виявляючи шляхи для оптимізації, включаючи скорочення часу виконання завдань, зменшення надмірних кроків та оптимізацію ресурсів. Моделі, створені за допомогою нотацій можуть бути використані як проектна документація, скорочуючи текстовий опис та підвищуючи розуміння, особливо у мультинаціональних та мультимовних командах, що допомагає забезпечити зрозумілість та легкість впровадження.

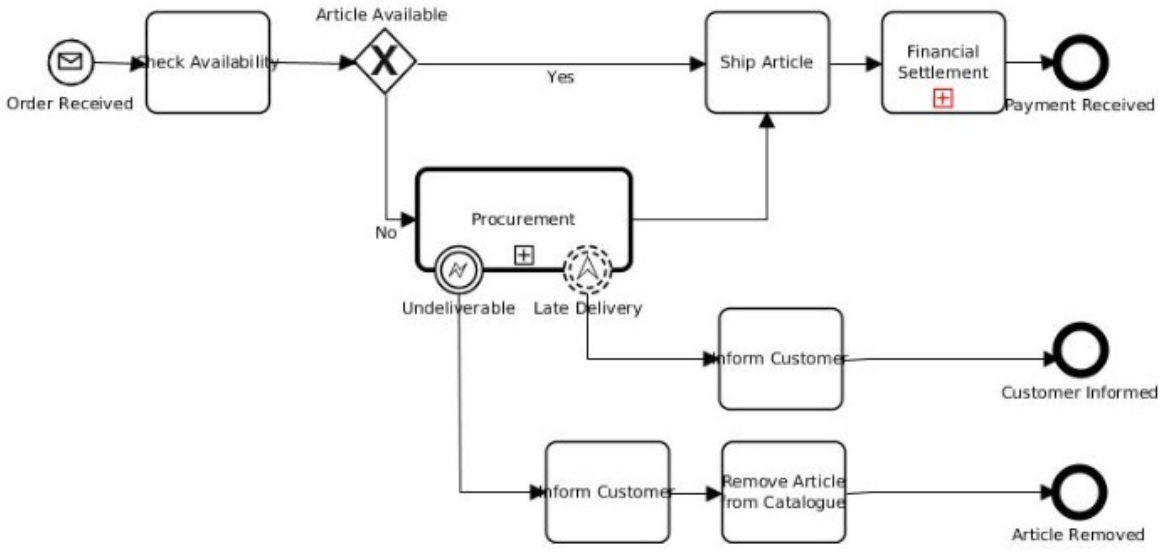


Рисунок 1.1 — Приклад графічної діаграми, створеної за допомогою BPMN [7]

Основними елементами BPMN є [7, 16, 19]:

Завдання (Task): Завдання визначають конкретні дії або кроки, які виконуються в межах бізнес-процесу. Вони представлені у вигляді прямокутників.

Подія (Event): Події вказують на зміни у стані бізнес-процесу. Вони можуть бути початковими, кінцевими, проміжними та сповільнюючими подіями.

Потік повідомлень (Message Flow): Цей елемент вказує на обмін повідомленнями між елементами процесу.

Вихідний потік (Sequence Flow): Вихідні потоки визначають послідовність виконання завдань та подій в бізнес-процесі.

Брама (Gateway): Брами визначають рішення або гілки в бізнес-процесі, в залежності від певних умов.

Пул — це контейнер, який представляє учасників або ролі в бізнес-процесі

Діаграма активностей — це графічне представлення процесу у вигляді діаграми.

BPMN дозволяє створювати ясні, зрозумілі та однозначні діаграми бізнес-процесів, які можуть бути легко сприйняті та інтерпретовані різними стейкхолдерами в організації. Цей стандарт є популярним інструментом для аналізу, прототипування та автоматизації бізнес-процесів в організаціях різних галузей.

Важливо зазначити, що UML і BPMN є двома різними нотаціями, які використовуються для графічного відображення різних аспектів програмних систем і бізнес-процесів відповідно. Хоча вони використовуються для різних цілей, є деякі схожості між ними [7; 19; 20]:

- UML та BPMN надають графічну нотацію для представлення моделей. Вони використовують діаграми та символи для візуального представлення інформації;
- обидві нотації мають різні типи елементів, такі як об'єкти, діаграми, лінії зв'язку тощо, для представлення конкретних аспектів системи чи бізнес-процесу;
- UML та BPMN можуть бути використані для спілкування з кінцевими користувачами і стейкхолдерами, які можуть не бути технічно освіченими;
- є стандартами, розробленими і підтримуваними відповідними організаціями (Object Management Group для UML і Object Management Group спільно з Business Process Management Initiative для BPMN).

Однак важливо відзначити, що UML і BPMN призначені для різних цілей і мають різні області застосування. UML використовується для моделювання програмних систем і структури даних, тоді як BPMN спрямований на моделювання бізнес-процесів і їх автоматизацію. Тому, хоча є схожості в їхній графічній нотації та способах представлення інформації, вони мають різні фокуси і використовуються в різних контекстах.

Найбільш наближеною за виглядом та суттю до BPMN (Business Process Model and Notation) діаграми є UML Activity діаграма. Обидві діаграми використовуються для моделювання процесів, послідовності дій та керування потоком робіт. Однак їхні основні використання і контексти відрізняються.

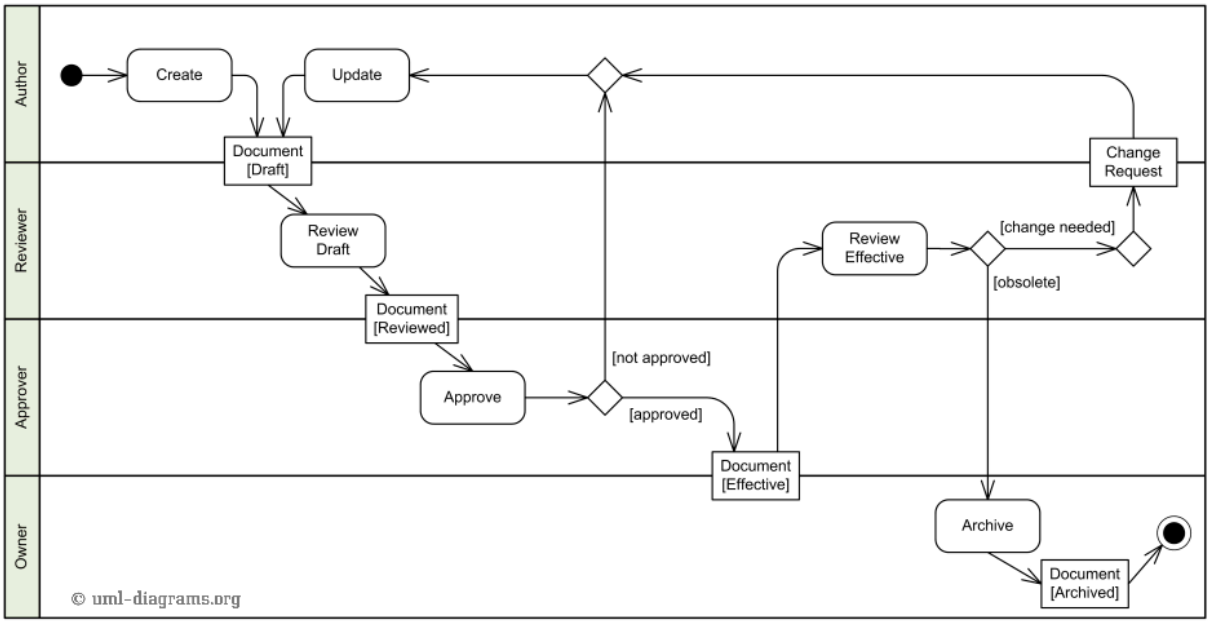


Рисунок 1.2 — Приклад UML Activity діаграми [20]

Характеристику спільних та відмінних рис, характерних для UML Activity діаграмами та діаграмам, побудованих за допомогою BPMN можна побачити у таблиці 1.2.

Таблиця 1.2 — Характеристика спільних та відмінних рис UML Activity діаграмам та BPMN-моделей

	Моделювання послідовностей дій, які відбуваються в рамках певного процесу	Використання символів та нотацій для представлення дій, рішень, об'єктів і потоків даних	Використання візуальних елементів для зрозумілого представлення процесів і послідовностей	Фокус на послідовних діях, вирішенні задач і обробці даних в рамках програмного коду	Використання для комунікації між бізнес-аналітиками, менеджерами і технічними командами
UML Activity діаграма	+	+	+	+	-
BPMN-моделі	+	+	+	-	+

Джерело: складено та доповнено автором на основі [13; 21]

Отже, хоча обидві нотації можуть використовуватися для моделювання процесів, вони мають різні контексти і застосування. UML Activity діаграми частіше використовуються в технічних аспектах розробки програмних систем, тоді як BPMN використовується для моделювання бізнес-процесів та організаційних аспектів.

Безпосередньої реалізації для моделювання бізнес-процесами BPMN знайшла у програмному забезпеченні Camunda. **Camunda** — це відкрите програмне забезпечення, яке надає інструменти для моделювання, автоматизації та управління бізнес-процесами. Це має велике значення для підприємств та організацій у багатьох галузях з наступних причин [10; 13; 22; 23]:

- *Управління бізнес-процесами:* Camunda дозволяє підприємствам моделювати і визначати бізнес-процеси, які вони використовують в своєму діловому середовищі. Це допомагає організаціям зрозуміти та оптимізувати, як вони функціонують.
- *Автоматизація процесів:* Camunda дозволяє автоматизувати бізнес-процеси, визначені в молулях BPMN (Business Process Model and Notation). Це забезпечує послідовне виконання завдань та операцій, зменшуючи ручну роботу та помилки.
- *Моніторинг та аналітика:* Camunda надає засоби для моніторингу виконання бізнес-процесів. Це допомагає виявляти затримки, проблеми та можливості для покращення. Аналітика надає інформацію про продуктивність та використання ресурсів.
- *Розподілена архітектура:* Camunda розроблений з урахуванням розподіленої архітектури, що дозволяє обробляти великий обсяг бізнес-процесів в розподілених середовищах.
- *Інтеграція:* Camunda легко інтегрується з іншими системами і додатками, що дозволяє використовувати його як основу для

автоматизації бізнес-процесів у різних галузях, включаючи фінанси, охорону здоров'я, виробництво, роздрібну торгівлю та інші.

- *Відкрите програмне забезпечення:* Camunda є відкритим програмним забезпеченням з відкритим кодом, що дозволяє користувачам безкоштовно використовувати та модифікувати його. Велика спільнота користувачів сприяє розвитку та підтримці цієї платформи.

Узагальнюючи, Camunda є потужним інструментом для управління бізнес-процесами, що допомагає підприємствам оптимізувати, автоматизувати та контролювати їхні процеси, забезпечуючи більшу ефективність та якість роботи.

Поряд з Camunda існує багато інших інструментів для моделювання та автоматизації бізнес-процесів, серед яких можна навести декілька популярних аналогів

Таблиця 1.3 — Характеристика окремих компаній на ринку послуг моделювання бізнес-процесів

Назва компанії	Загальна характеристика продукту	Умови надання послуг	Специфіка
1	2	3	4
UiPath	Пропонує платформу для роботизації процесів (RPA), яка дозволяє автоматизувати рутинні завдання.	Існує комбінація безкоштовних та платних версій. Деталі можуть залежати від типу користувача (індивідуальний, підприємство).	Відомий своєю потужною RPA-технологією, здатною обробляти різноманітні завдання автоматизації процесів.
Automation Anywhere	Надає платформу для роботизації бізнес-процесів та задач RPA.	Має різні плани та моделі ціноутворення, включаючи платні версії для корпоративних користувачів.	Відзначається гнучкістю та широким функціоналом для автоматизації завдань на всіх рівнях.

Продовження табл. 1.3

1	2	3	4
Nintex	Пропонує рішення для автоматизації бізнес-процесів та управління робочими потоками.	Має платні плани, враховуючи потреби корпоративних клієнтів.	Використовується для автоматизації робочих процесів на платформі Microsoft SharePoint.
SAP NetWeaver BPM	Це частина SAP NetWeaver, яка забезпечує можливості управління бізнес-процесами.	Зазвичай, це платне рішення для клієнтів SAP.	Інтегрована з рішеннями SAP, надає можливості автоматизації та оптимізації бізнес-процесів.
Appian	Пропонує платформу для низькокодової розробки та управління бізнес-процесами.	Зазвичай, платне рішення для корпоративних клієнтів.	Славиться своєю можливістю швидко розробляти та впроваджувати бізнес-процеси.
Microsoft Power Automate	Дозволяє створювати автоматизовані робочі потоки між різними додатками та сервісами.	Має безкоштовні та платні плани, інтегровані з іншими сервісами Microsoft.	Ідеальний варіант для користувачів, які вже використовують екосистему Microsoft та шукають прості рішення для автоматизації.
Facebook Business Manager	Надає інструменти для управління бізнес-сторінками та оголошеннями на Facebook.	Безкоштовно для користувачів, але з можливістю платного розширення функціоналу.	Фокусується на соціальному маркетингу та взаємодії з аудиторією через платформу Facebook.
Camunda	Пропонує відкрите програмне забезпечення для автоматизації бізнес-процесів та управління робочими потоками.	Відкритий і безкоштовний, але існують комерційні плани для підприємств з додатковими функціями та підтримкою.	Спеціалізується на автоматизації бізнес-процесів та використовується як вбудована частина багатьох систем.
Innovation Group	Фокусується на розробці програмних рішень для страхових та фінансових секторів.	Зазвичай платне рішення згідно з обсягом та потребами клієнта.	Спеціалізується на інноваціях у страхуванні та фінансах, включаючи автоматизацію бізнес-процесів.

Продовження табл. 1.3

1	2	3	4
IBM Business Process Manager	Це рішення для автоматизації, моделювання та оптимізації бізнес-процесів.	Зазвичай платне рішення, орієнтоване на великі підприємства.	Інтегроване з різними сервісами та продуктами IBM, спрямоване на складні корпоративні потреби.
Azure Automation	Частина хмарної платформи Microsoft Azure для автоматизації операцій.	Платні плани з урахуванням обсягу та використання ресурсів в хмарі.	Фокусується на автоматизації та управлінні хмарними ресурсами.

Джерело: складено та доповнено автором на основі [22-29]

Розподіл долі на ринку послуг з моделювання бізнес-процесів компаній з таблиці 1.3 наведений на рисунку 1.3 нижче

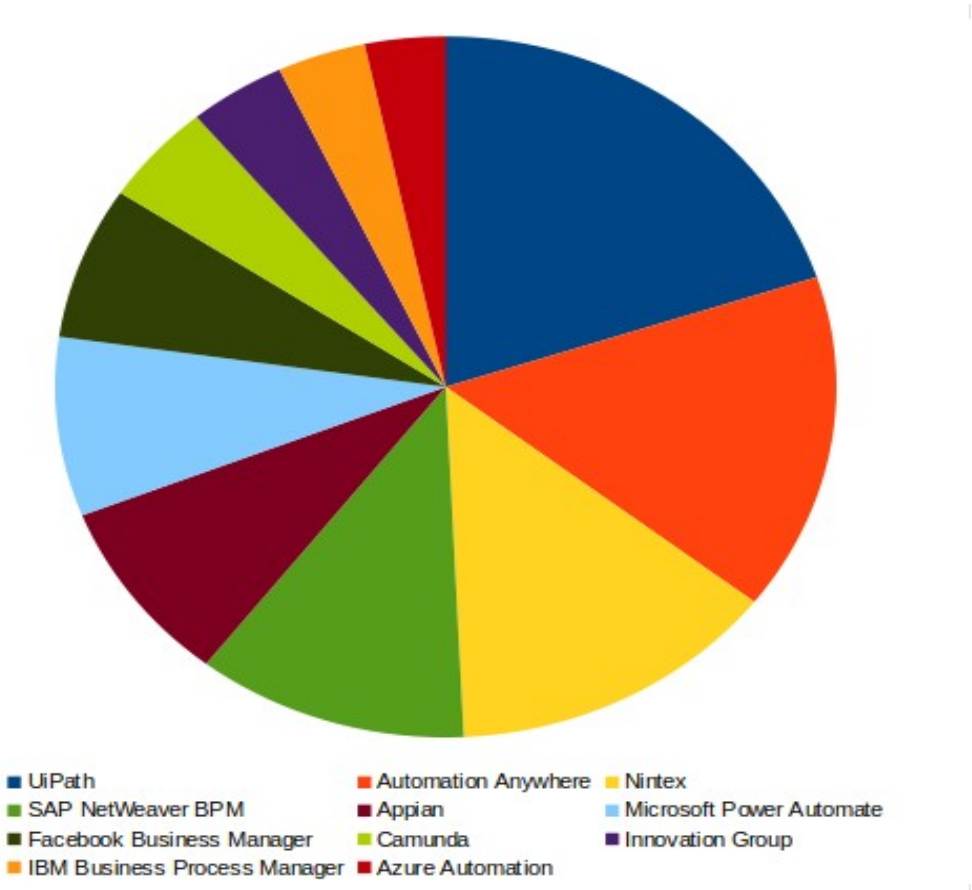


Рисунок. 1.3 — Порівняння долі окремих компаній на ринку послуг моделювання бізнес-процесів [22, 23, 29]

Ці інструменти можуть варіюватися за функціональністю, масштабом, вартістю та спрямованістю, тому вибір конкретного інструмента буде залежати від потреб вашого проекту та бізнес-вимог.

1.3 Постановка задачі дослідження

Довівши значущість бізнес-процесів та проаналізувавши підходи до побудови графічних нотацій за допомогою BPMN, переходимо до постановки задачі для побудови інформаційної технології автоматизації тестування бізнес-процесів.

В якості інструменту оркестрації бізнес-процесів ми пропонуємо обрати Camunda, незважаючи на його ненайбільшу частку на ринку послуг даного типу (3,13% [23]), безперечними перевагами є висока простота використання та потужні можливості для моделювання та виконання процесів. Використання BPMN надає не лише зручний інтерфейс для моделювання процесів, але і стандартизує їх, що спрощує розуміння та співпрацю між командами, а вбудований двигун забезпечує їх ефективне виконання. Вона є високоефективним інструментом для оркестрації та управління процесами. Поряд з цим, незважаючи на досить потужний інструментарій, якій йде “з коробки” Camunda не має вбудованих засобів для автоматизованого тестування.

jMeter — це добре відомий та широко використовуваний інструмент для автоматизованого тестування великої кількості протоколів, включаючи HTTP, HTTPS, JDBC, FTP та інші [33]. Його висока ефективність, гнучкість та можливість використання для різних видів тестувань роблять його популярним серед розробників.

Сполучення Camunda з jMeter визначається не лише їхньою високою функціональністю, а й сучасністю використання в індустрії. Camunda, як

інструмент для оркестрації, дозволяє підприємствам створювати гнучкі та високопродуктивні бізнес-процеси. jMeter, у свою чергу, надає потужні інструменти для автоматизованого тестування, забезпечуючи високу якість та надійність системи.

Обрана комбінація інструментів дозволяє підтримувати високий рівень якості та ефективності бізнес-процесів, забезпечуючи їхню адаптивність до змін у середовищі та стрімке впровадження нових функцій. Такий підхід відповідає вимогам сучасного бізнесу до надійності та швидкості виконання завдань.

Сучасне бізнес-середовище вимагає від підприємств швидкої адаптації до змін, інноваційних рішень та високої продуктивності. Використання інструментів, які забезпечують автоматизацію та оптимізацію бізнес-процесів, стає критичним для забезпечення конкурентоспроможності підприємства на ринку. Обрана комбінація Camunda та jMeter відповідає сучасним вимогам та дозволяє ефективно управляти бізнес-процесами, забезпечуючи їхню якість та стабільність у реальних умовах використання.

Розробка інформаційної технології автоматизації тестування бізнес-процесів на основі Camunda та jMeter є доволі цікавою задачею як з теоретичної точки зору, так має і значний прикладний аспект, враховуючи викладені вище аргументи щодо важливості оркестрації бізнес-процесів. Ця розробка не лише покликана підвищити ефективність виконання завдань, але і забезпечити стабільність та відповідність бізнес-процесів стратегічним цілям підприємства в умовах постійних змін та викликів на ринку.

2 ПРОТОТИПУВАННЯ БІЗНЕС-ПРОЦЕСІВ ЗА ДОПОМОГОЮ CAMUNDA

2.1 Практичні засади об’єкту дослідження

Об’єктом дослідження був обраний застосунок-соціальна мережа, створений на базі фреймворку SpringBoot на платформі Maven. При написанні проекту використана мова програмування Java 17ої версії. Інформація щодо деяких модулів, використаних в проектів надані в таблиці нижче.

Таблиця 2.1 — Модулі, використані при написанні об’єкту дослідження

Назва модулю	Опис
1	2
spring-boot-starter-actuator	надає готові до використання функції для моніторингу та управління додатком. Він містить ендпоінти для перевірки стану, метрик, конфігурації логування та багато іншого.
spring-boot-starter-data-jpa	забезпечує підтримку доступу до даних за допомогою Java Persistence API (JPA). Він містить залежності для роботи з базами даних за допомогою JPA, таких як Hibernate як реалізація JPA за замовчуванням.
spring-boot-starter-web	надає фреймворк веб-додатків для створення веб-орієнтованих застосунків. Він включає залежності для створення RESTful-сервісів, обробки HTTP-запитів і відповідей, а також налаштування веб-пов'язаних функцій.
spring-boot-starter-thymeleaf	дозволяє використовувати Thymeleaf як шаблонний двигун для рендерингу HTML-сторінок на серверній стороні. Він включає залежності для інтеграції Thymeleaf з Spring Boot.
spring-boot-starter-test	надає залежності для тестування додатку. Він включає бібліотеки, такі як JUnit, Mockito та Spring Test, для написання модульних тестів, інтеграційних тестів та виконання тестових перевірок.
spring-boot-starter-security	надає функціональність забезпечення безпеки додатку. Він включає залежності для реалізації аутентифікації, авторизації та захисту веб-точок доступу.

Продовження табл. 2.1

thymeleaf-extras-springsecurity	забезпечує інтеграцію між Thymeleaf та Spring Security. Він включає додаткові функції і утиліти для роботи з завданнями, пов'язаними з безпекою, в шаблонах Thymeleaf.
org.camunda.bpm.springboot	представляє собою стартовий пакет (starter) для інтеграції Camunda BPM з проектами Spring Boot. Цей стартовий пакет містить необхідні залежності та конфігурації для зручної і безпроблемної інтеграції двигуна бізнес-процесів Camunda BPM з додатками, побудованими на платформі Spring Boot.
spring-boot-starter-validation	надає підтримку валідації в додатку. Він включає залежності для використання анотацій валідації Bean Validation (JSR 380) для перевірки вхідних даних.
spring-boot-devtools	надає функції для розробки, які покращують продуктивність. Він включає автоматичне перезапуск додатку, живе оновлення статичних ресурсів та інші інструменти для швидшої розробки.
rometools	бібліотека для роботи з RSS- та Atom-стрічками. Вона надає підтримку для парсингу, маніпулювання та генерації даних стрічок.
spring-boot-properties-migrator	допомагає мігрувати застарілі властивості конфігурації в Spring Boot додатках. Він надає утиліти для ідентифікації та оновлення застарілих властивостей під час оновлення додатка.
lombok	бібліотека, яка зменшує кількість рутинного коду в класах Java. Вона надає анотації для автоматичної генерації методів доступу (getters/setters), конструкторів та інших часто використовуваних методів, що покращує читабельність та підтримку коду

Джерело: складено та доповнено автором на основі [30]

Проект реалізований за допомогою мікросервісної архітектури на базі “клієнт-серверного” підходу. Серверна частина хоститься на хмарних серверах Amazon, AWS, а клієнтська частина доступна через браузер або мобільний застосунок. Детальна схема наведена на рисунку нижче.

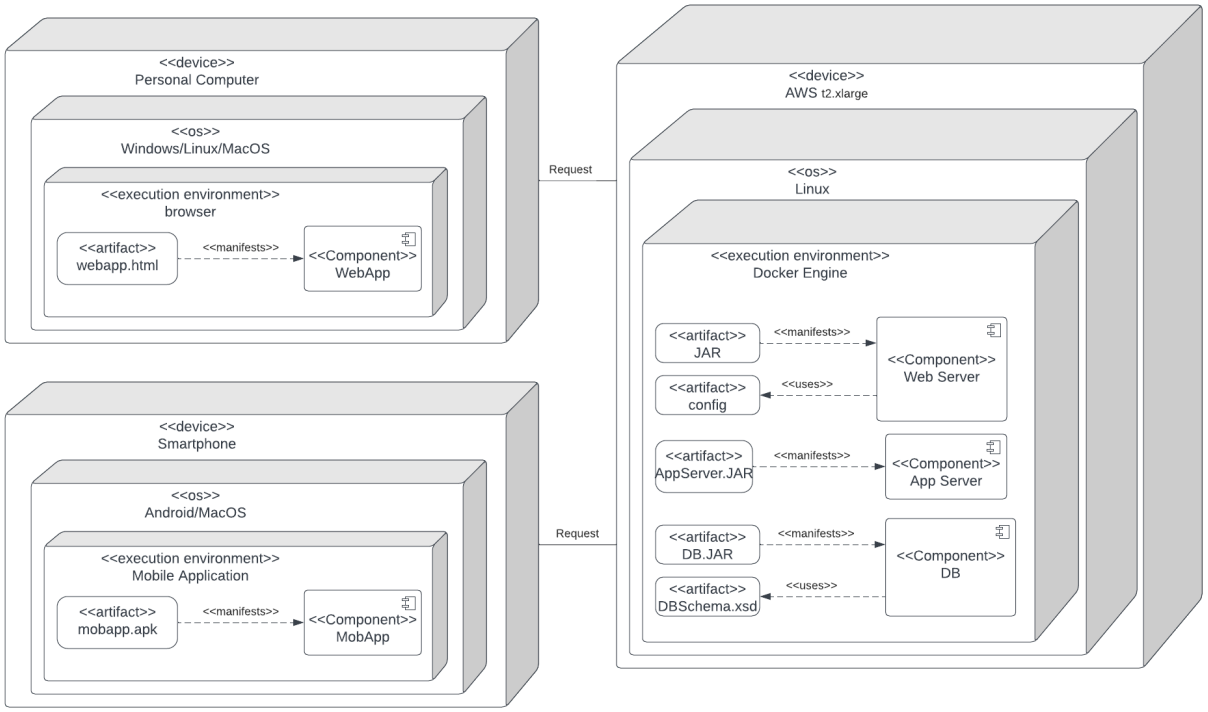


Рисунок 2.1 — Діаграма деплою об’єкта дослідження

В проекті обрана мікросервісна архітектура і кожен з компонентів працює окремо, маючи власну базу та спілкуючись з іншими сервісами за допомогою відповідних ендпоінтів. Компонента складова представлена на рисунку нижче.

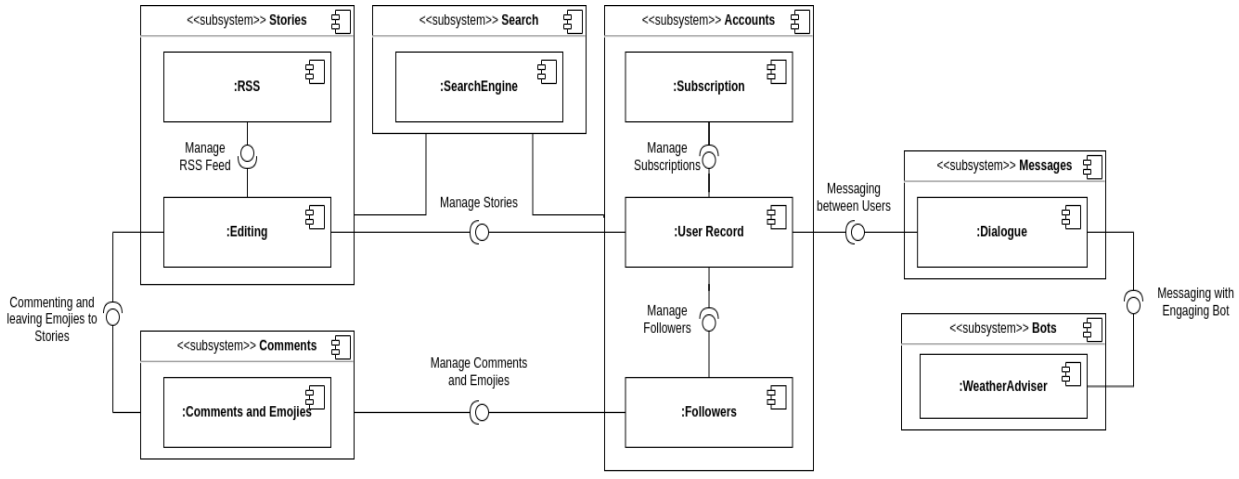


Рисунок 2.2 — Компонентна діаграма об’єкта дослідження

Мікросервіс Stories відповідає за створення дописів (компонент — Editing) та наповнення стрічки дописів (RSS). Мікросервіс Accounts відповідає за записи користувачів (User Record), їх типи підписки (Subscription), а також характер взаємодії між користувачами, який регулюється компонентом Followers. В залежності від того чи є користувачі друзями/послідовниками один одного вони мають змогу переглядати та коментувати дописи один одного як у стрічці, так і на сторінках відповідних користувачів. За функціонал коментування в текстовому режимі або картинками-емоджі відповідає компонент Comments and Emojies, який є складовою підсистеми Comments.

Пошуковий сервіс Search використовує запропонований AWS пошукове ядро ElasticSearch та надає можливість пошуку дописів та користувачів. Дописи шукаються за темою та вмістом, користувачі — за емейлом, логіном або ім’ям. В системі доступний лише текстовий пошук.

Поряд з описаними сервісами, в системі також функціонує сервіс приватних повідомлень — Messages, який надає можливість обмінюватися повідомленнями у реальному часі із іншими користувачами. На даний момент також впроваджується функціонал активізації спілкування та заохочення відвідування мережі за рахунок ботів, одним з яких є WeatherAdviser. Даний бот на базі даних про погоду у регіоні користувача через сервіс приватних повідомлень надає поради, на кшталт “Сьогодні обіцяють дощ, не забудьте взяти парасольку”. Детальніше про механізм його роботи буде описано нижче.

2.2 Версіювання та CI/CD на об’єкті дослідження

При створенні даного програмного продукту використана система контролю версій — GitHub та стратегія гілкування для релізів (Release

Branching Strategy). Завдяки цьому забезпечується структурований підхід для управління кількома версіями програмного забезпечення, що гарантує гладке та контрольоване впровадження нових функцій, виправлень помилок та оновлень.

Відмінною рисою даної стратегії є створення відокремленої гілки, відома як гілка релізу (release branch), для підготовки та стабілізації програмного забезпечення для релізу. Гілка релізу, як правило, створюється з основної гілки розробки (наприклад, гілки "master") в певний момент, зазвичай, коли програмне забезпечення досягає стабільного стану і готове до випуску [31].

За допомогою стратегії гілкування для релізів команда може зберігати чітку відмежованість між незавершеними розробками та процесом релізу. Вона забезпечує контрольоване середовище для завершення та впровадження релізів програмного забезпечення, забезпечуючи при цьому безперебійну продовження основних розробницьких робіт.

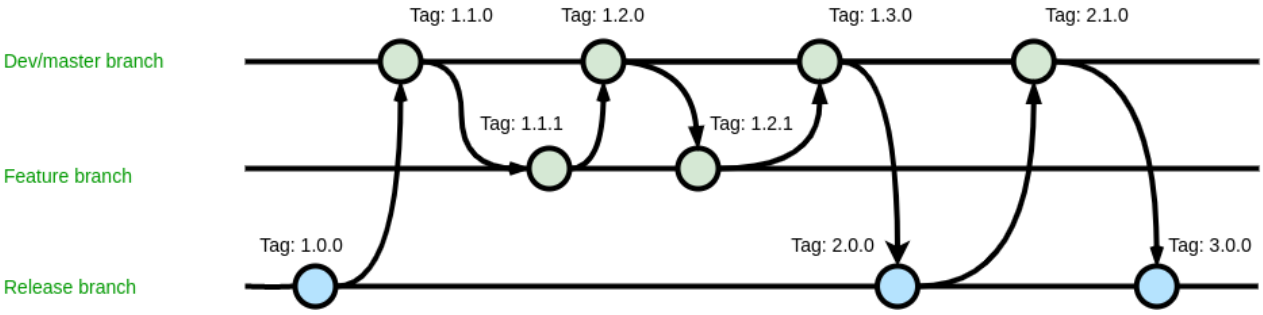


Рисунок 2.3 — Стратегія гілкування на об’єкті дослідження

Для збірки проекту використовується функціонал Github Actions — це функція платформи GitHub, яка дозволяє автоматизувати робочий процес, пов'язаний з розробкою програмного забезпечення. GitHub Actions надає гнучкість та потужність для автоматизації різних робочих процесів в гітхаб-репозиторіях.

Таблиця 2.2 — Основні можливості GitHub Actions

Можливості	Опис
Continuous Integration (CI)	дозволяє встановлювати та налаштовувати CI-потоки для вашого проекту. При кожному злитті коду (pull request) або коміті в гілку автоматично виконуються тести, збірка проекту та інші перевірки
Continuous Deployment (CD)	використовується для автоматичного розгортання вашого проекту після успішної збірки, що спрощує реліз нових версій програмного забезпечення;
Спільна робота між командами	дозволяє налаштовувати робочі процеси, що залежать від подій, таких як створення нового злиття коду, коментарі, створення релізу, тощо.
Наявність готових дій	має багато готових стандартних кроків для збирання, тестування, розгортання тощо, або існує можливість написати свої власні дії, які можна використовувати у робочих процесах
Графік робочих процесів	визначати коли і як часто повинні виконуватися налаштування GitHub Actions
Розширені можливості для CI/CD	має декілька можливостей для конфігурації CI/CD, таких як сховище секретів, можливість використання власних середовищ виконання (self-hosted runners), інтеграцію з Docker та інше.

Джерело: складено та доповнено автором на основі [32]

Безпосередні кроки, які виконуються при збірці наведені на рисунку ниже.

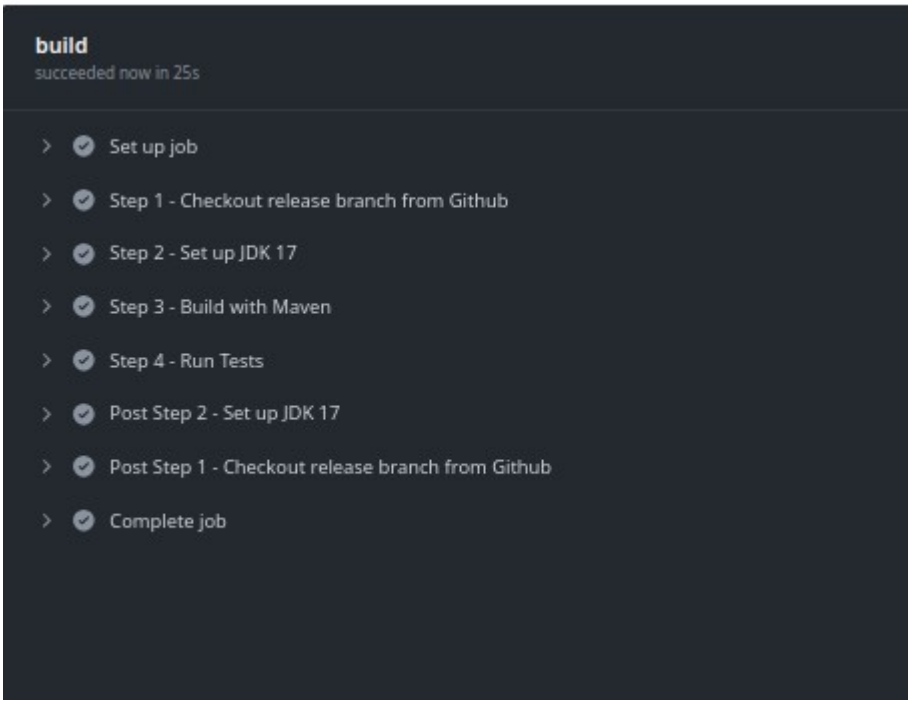


Рисунок 2.4 — Кроки збірки проекту за допомогою Github Actions

Під час збірки виконується не лише безпосередня збірка проєкту, а й супровідні кроки: налаштування середовища, збірка, тестування.

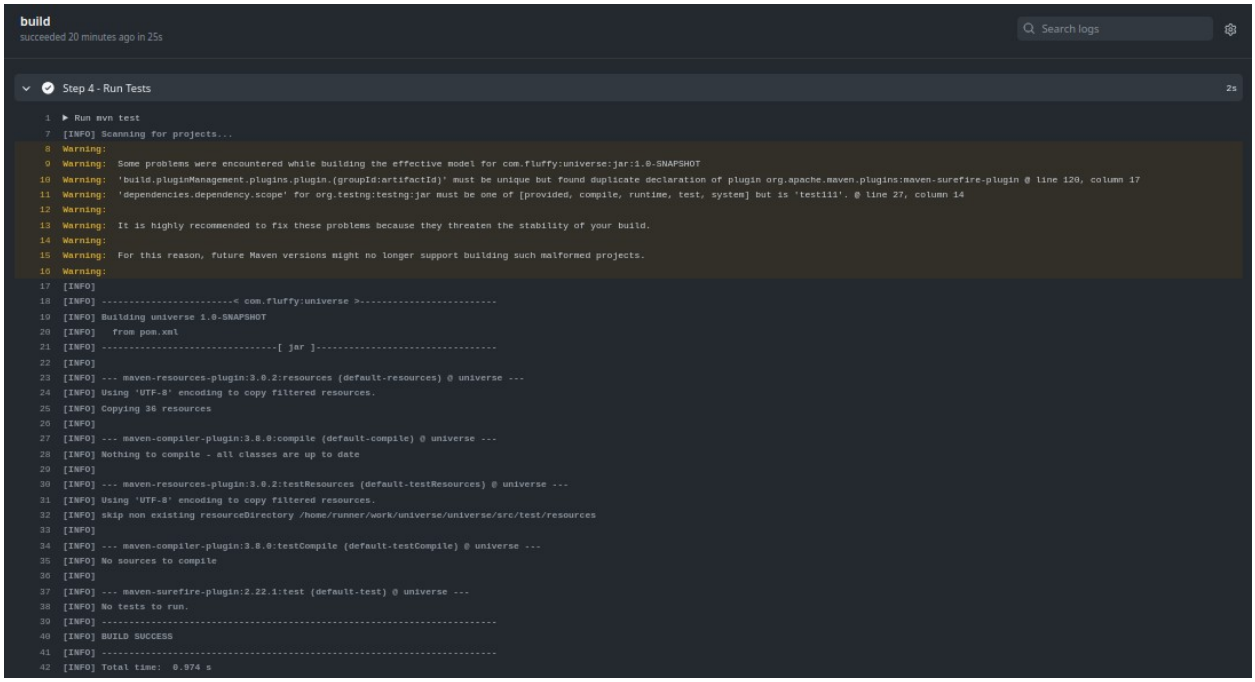


Рисунок 2.5 — Запуск тестів при збірці проекту

Під час збірки за допомогою Github Actions автоматично запускаються тести проекту, які включають як юніт-тести, так і автоматизовані end-to-end тести та вантажні тести

2.3 Засади тестування на об’єкті дослідження за допомогою jMeter

Досить багато уваги команда приділяє написанню вантажних тестів для jMeter, які також автоматично запускаються при збірці проекту. Це важливо, тому що на даний момент розробка бекенду випереджає фронтендну частину, але ми маємо бути певні, що впроваджений функціонал стабільний і здатен витримати істотне навантаження.

jMeter — це інструмент для навантажувального тестування та вимірювання продуктивності програмного забезпечення [33]. Він розроблений для тестування веб-додатків, але також підтримує інші протоколи, такі як FTP, JDBC, SOAP, тощо [34]. jMeter дає можливість створювати та виконувати тести, моделювати реальні навантаження на сервер та аналізувати результати.

Вантажне тестування за допомогою Apache jMeter - це процес симуляції одночасного доступу багатьох користувачів до веб-сайту чи веб-додатку для оцінки його продуктивності та витривалості під навантаженням. Механізм проведення вантажного тестування за допомогою jMeter на проекті налаштований наступним чином:

1. Залежності jMeter включені до файлу конфігурації CI/CD. Як було показано на рисунку вище, під час зборки тести запускаються автоматично;
2. Test Plan — це вихідний компонент jMeter, в якому будується тестовий сценарій. Він є контейнером для всіх інших компонентів jMeter, таких як Thread Group, Samplers, Listeners, Assertions та інші. В Test Plan

визначається налаштування тесту, такі як кількість користувачів, час завершення, глобальні налаштування проксі, заголовки, додаткові параметри тощо [35].

3. Thread Group — це компонент, який визначає групу потоків (користувачів), які будуть виконувати тест. Кожен потік виконується окремо, що дозволяє створювати навантаження на сервер [33].

   У Thread Group ви можете налаштувати наступні параметри [34]:

   - Number of Threads (Users): Кількість потоків (користувачів), які будуть виконувати тест.
   - Ramp-Up Period: Час, протягом якого потоки будуть стартувати. Наприклад, якщо ви вказали 100 потоків і Ramp-Up Period 10 секунд, то новий потік буде стартувати кожну 100 мс протягом 10 секунд.
   - Loop Count: Кількість разів, які поток повторить весь сценарій.
   - Scheduler: Дозволяє запускати тест за заданим графіком, вказуючи час початку та тривалість.

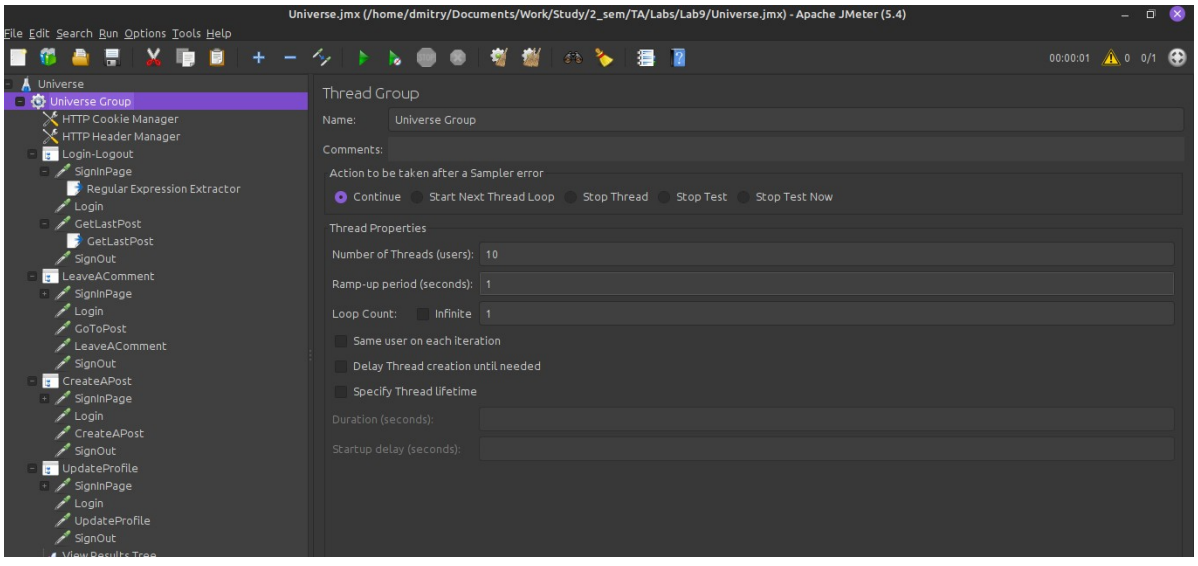


Рисунок 2.6 — Демонстрація Thread Group тестів у jMeter

4. HTTP Cookie Manager: Це компонент, який дозволяє управляти HTTP-кукі на протязі тестування. Він автоматично додає, оновлює та видаляє

куки для кожного HTTP-запиту, забезпечуючи правильне управління сеансами і підтримку стану [33].

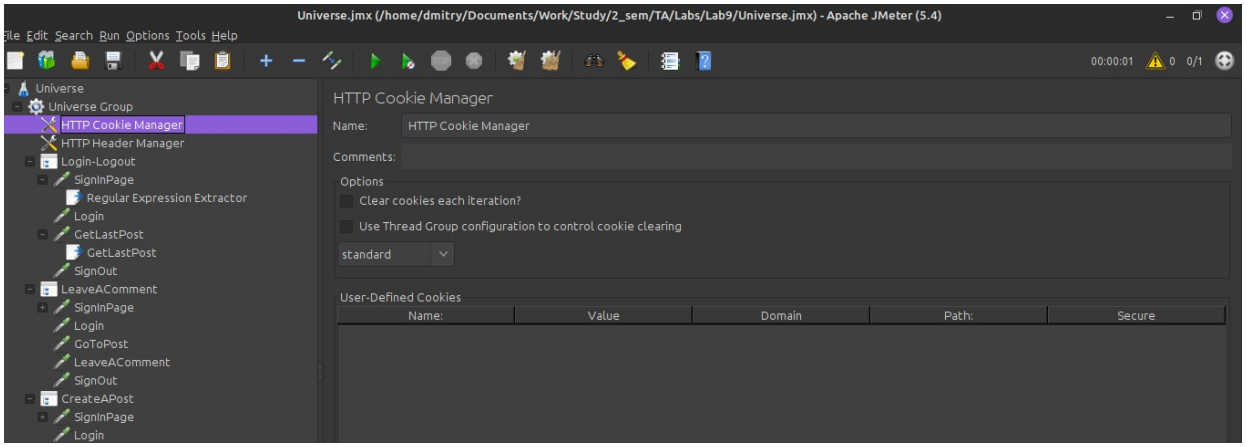


Рисунок 2.7 — Демонстрація налаштувань HTTP Cookie Manager’а тестів у jMeter

5. HTTP Header Manager: Компонент, який дозволяє додавати HTTP-заголовки до кожного HTTP-запиту. Ви можете використовувати його для встановлення заголовків, таких як User-Agent, Content-Type, Authorization тощо [33].

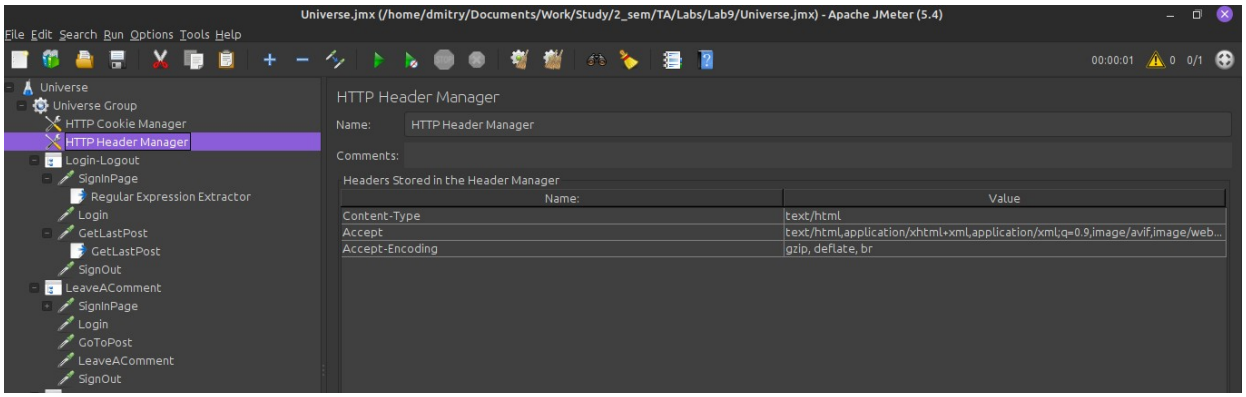


Рисунок 2.8 — Демонстрація налаштувань HTTP Header Manager’а тестів у jMeter

6. HTTP Request: Цей компонент дозволяє виконувати HTTP-запити до веб-сервера [35].

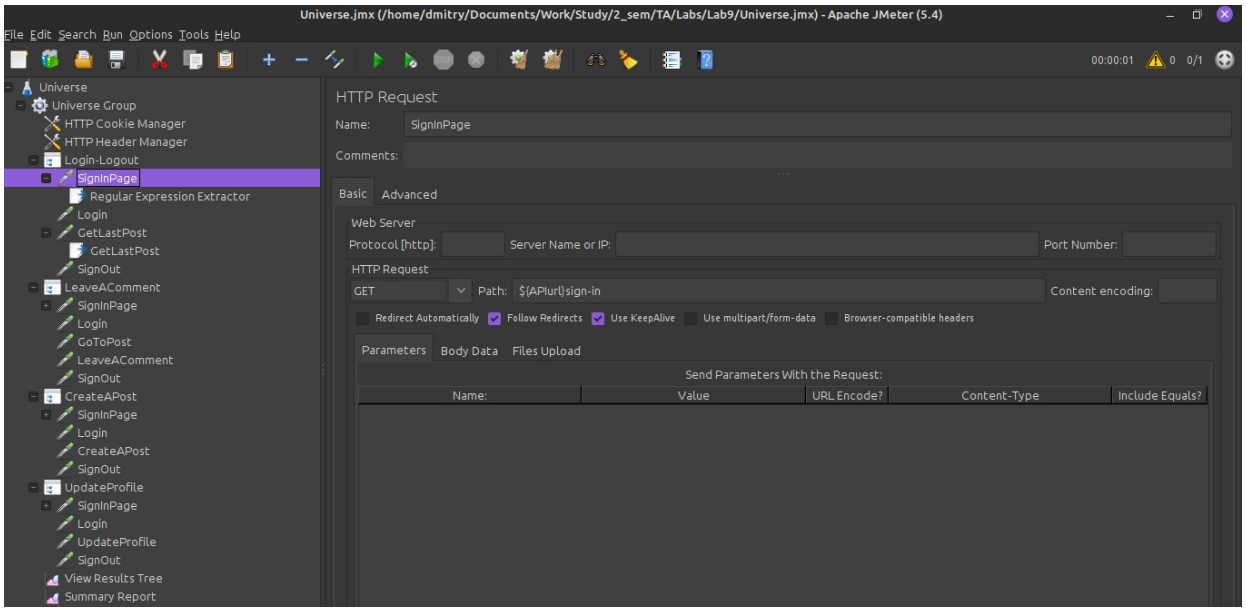


Рисунок 2.9 — Приклад HTTP Request’а тестів у jMeter

7. Regular Expression Extractor: Це компонент, який дозволяє витягувати дані з відповіді сервера за допомогою регулярних виразів, можна використовувати його для виділення конкретних значень з тексту відповіді і використовувати їх в подальших запитах або для аналізу результатів [34].

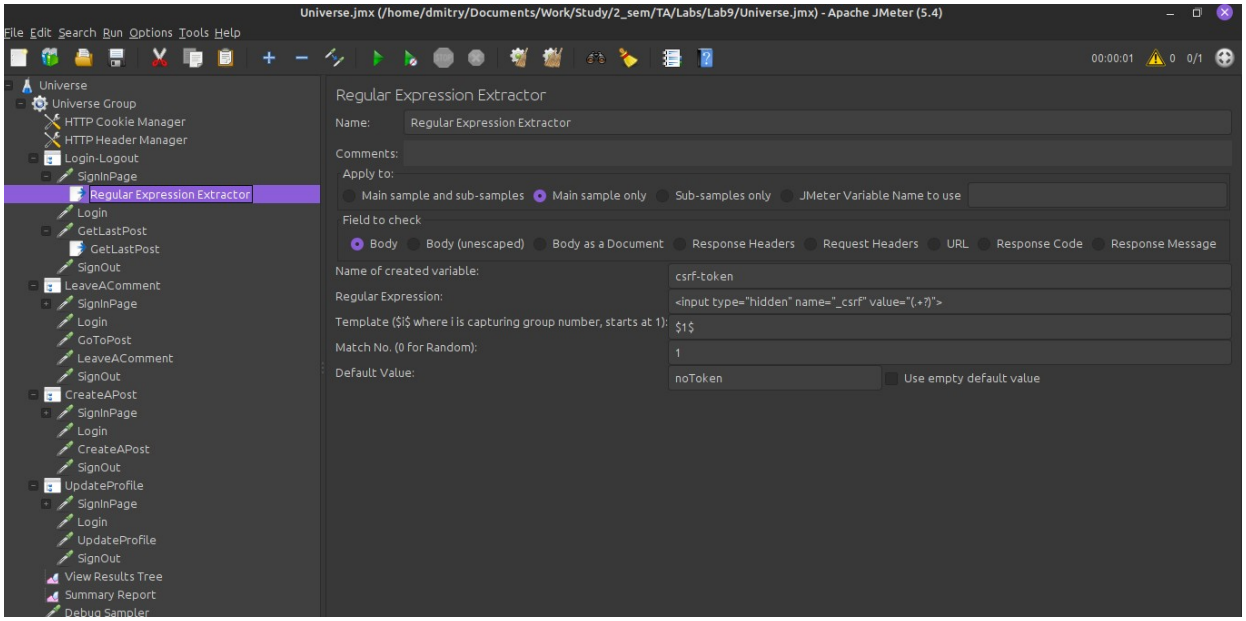


Рисунок 2.10 — Приклад Regular Expression Extractor’а тестів у jMeter

8. View Results Tree: Цей компонент дозволяє переглядати деталі відповіді сервера для кожного запиту: заголовки, тіло відповіді, помилки, час відповіді тощо [35]. Він допомагає аналізувати результати тестування та виявляти проблеми.

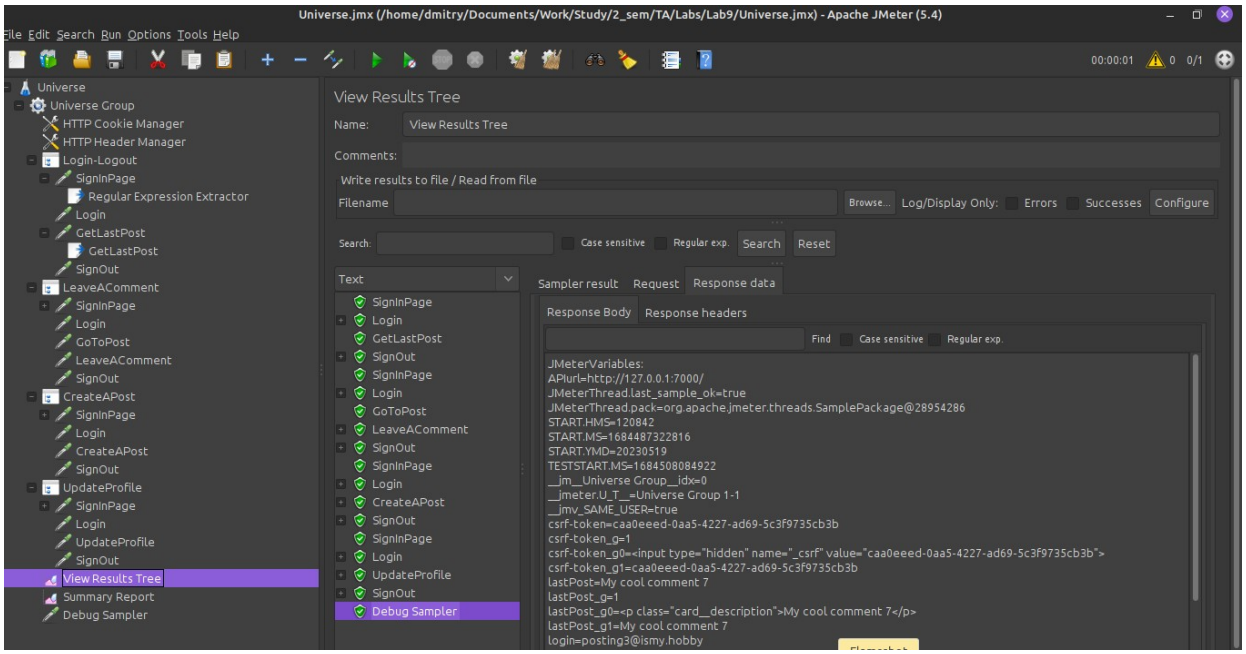
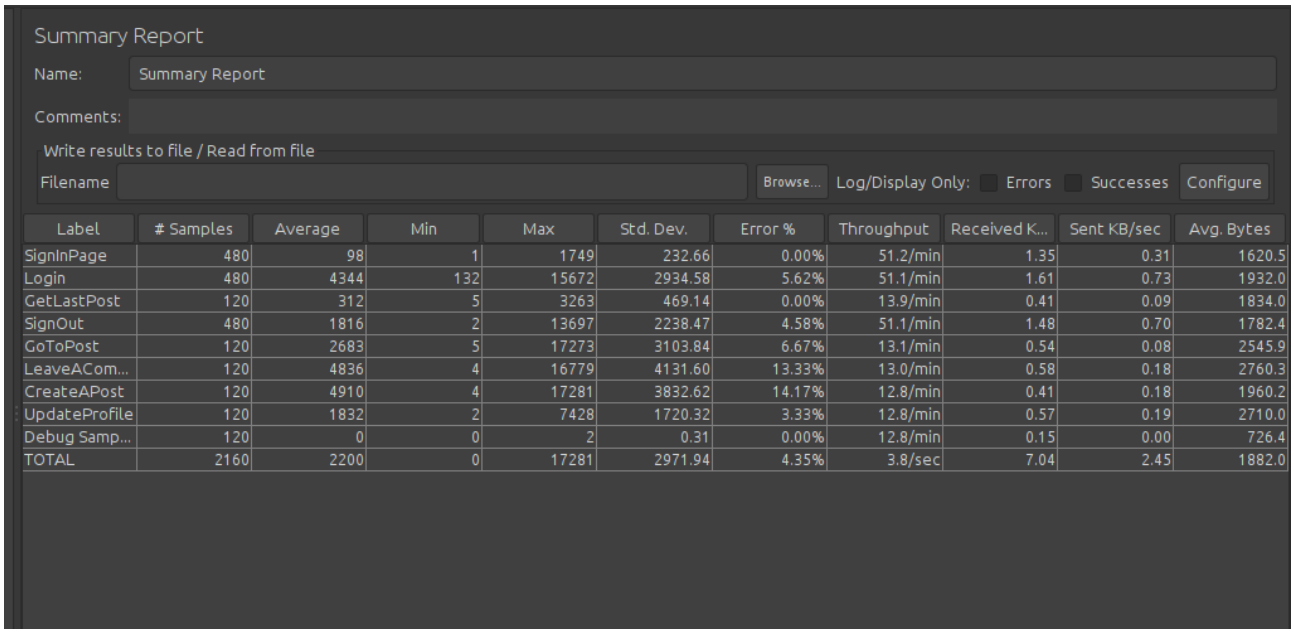


Рисунок 2.11 — Приклад View Results Tree тестів у jMeter

9. Summary Report: Компонент, який надає підсумкову інформацію про продуктивність тесту, включає загальну кількість запитів, середній час відповіді, мінімальний та максимальний час відповіді тощо [33]. Він допомагає отримати загальний огляд про продуктивність системи під навантаженням.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
SignInPage	480	98	1	1749	232.66	0.00%	51.2/min	1.35	0.31	1620.5
Login	480	4344	132	15672	2934.58	5.62%	51.1/min	1.61	0.73	1932.0
GetLastPost	120	312	5	3263	469.14	0.00%	13.9/min	0.41	0.09	1834.0
SignOut	480	1816	2	13697	2238.47	4.58%	51.1/min	1.48	0.70	1782.4
GoToPost	120	2683	5	17273	3103.84	6.67%	13.1/min	0.54	0.08	2545.9
LeaveACom...	120	4836	4	16779	4131.60	13.33%	13.0/min	0.58	0.18	2760.3
CreateAPost	120	4910	4	17281	3832.62	14.17%	12.8/min	0.41	0.18	1960.2
UpdateProfile	120	1832	2	7428	1720.32	3.33%	12.8/min	0.57	0.19	2710.0
Debug Samp...	120	0	0	2	0.31	0.00%	12.8/min	0.15	0.00	726.4
TOTAL	2160	2200	0	17281	2971.94	4.35%	3.8/sec	7.04	2.45	1882.0

Рисунок 2.12 — Приклад Summary Report тестів у jMeter

10. Debug Sampler: Цей компонент дозволяє додавати спеціальний запит для відлагодження тестів. Він використовується для перевірки значень змінних, логування повідомлень або інших дій під час виконання тестів [35].

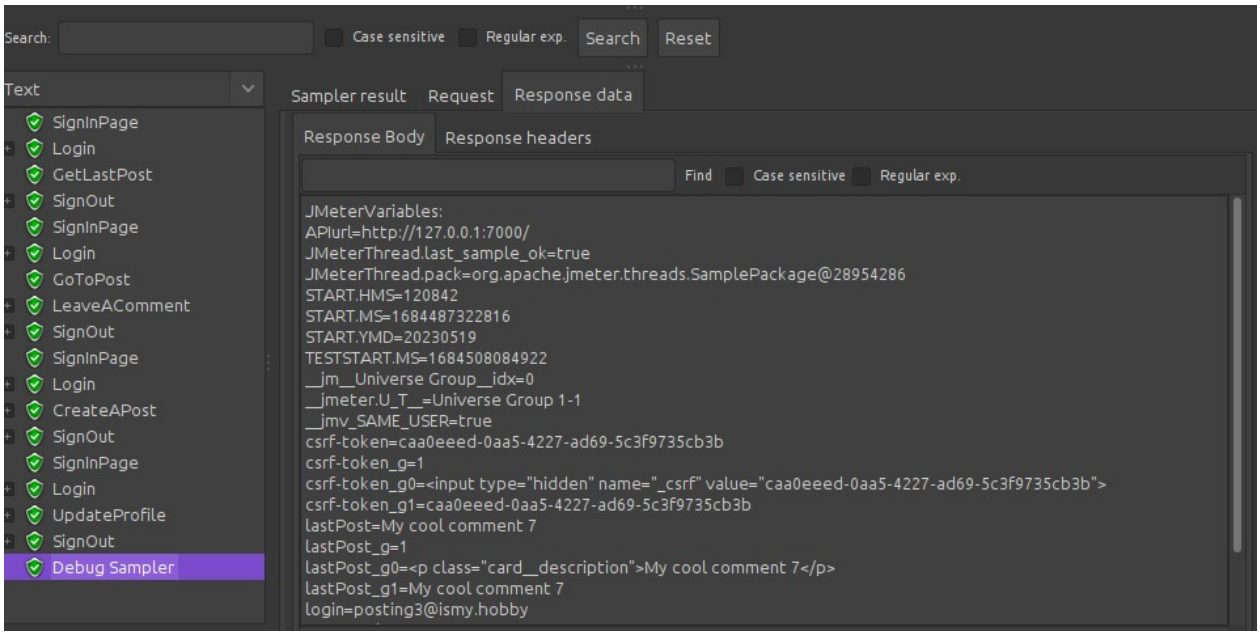


Рисунок 2.13 — Приклад роботи Debug Sample у jMeter

Сценарії можуть використовуватися не лише для вантажного, а й стрес-тестування, метою якого є виявлення меж стійкості системи.

3 ІНТЕГРАЦІЯ JMETER З CAMUNDA

3.1 Оркестрація процесів в системі за допомогою Camunda

На проекті також широко використовується BPMN, а саме Camunda для документування логіки роботи окремих сервісів і команда добре знайома з нею. Загальна структура Camunda представлена на рисунку нижче.

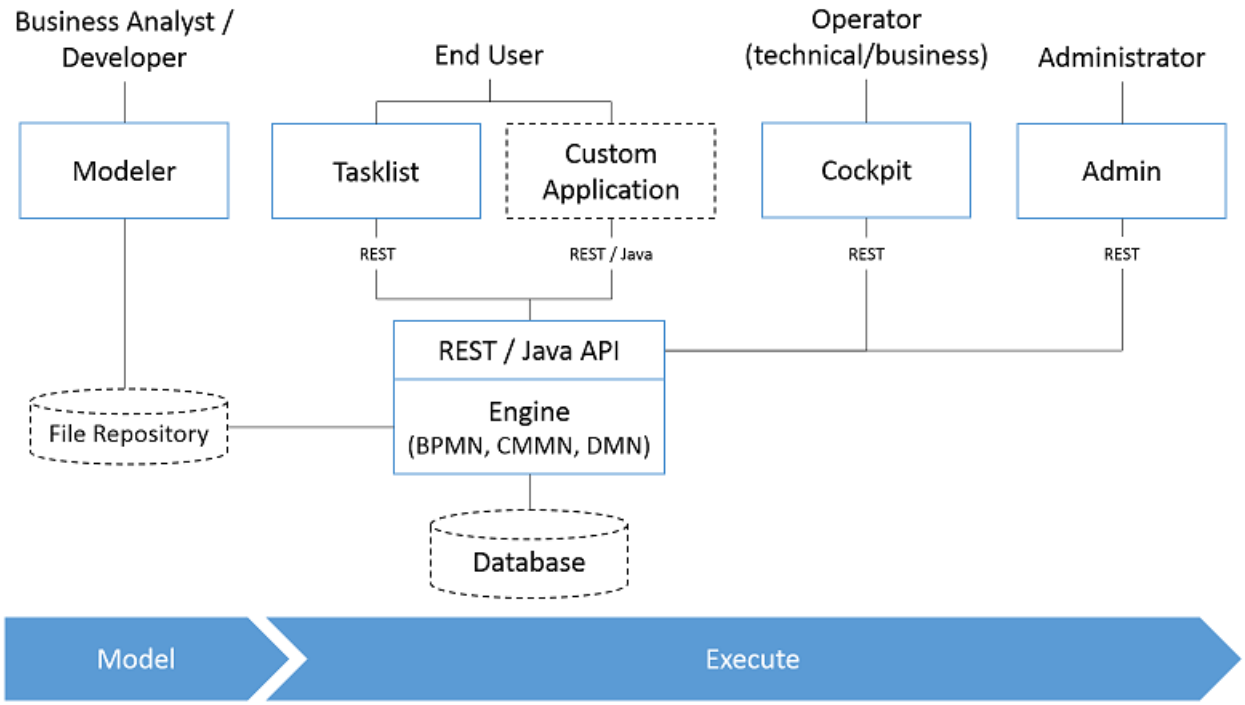
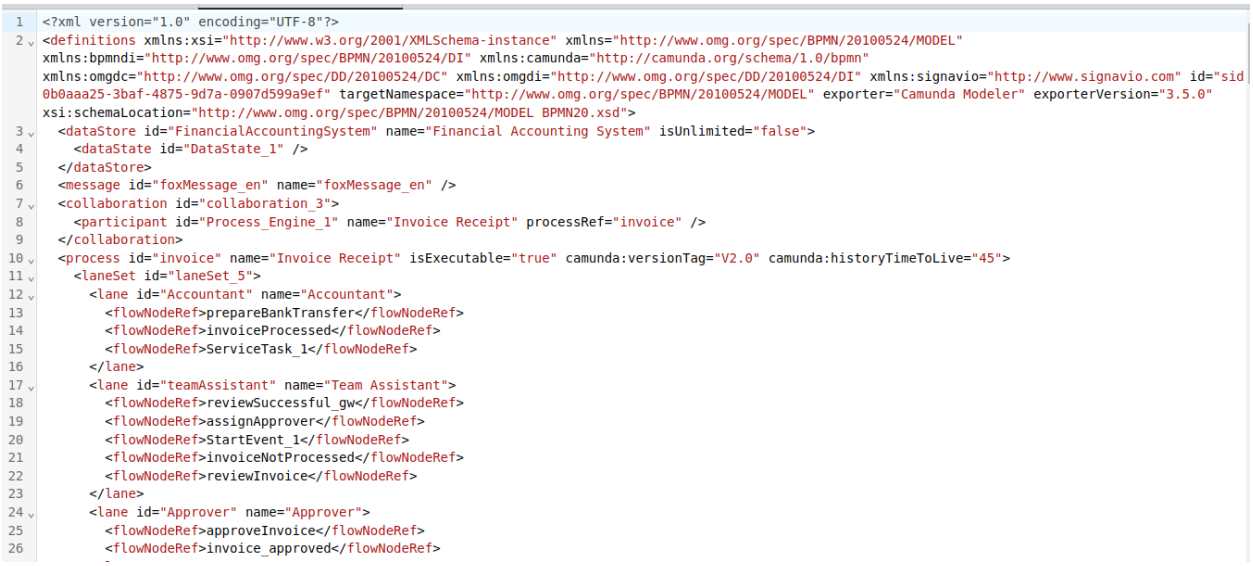


Рисунок 3.1 — Сервісна діаграма Camunda [36]

Бізнес-аналітики та проєкті оунери використовуються Modeler для опису логіки та моделювання процесів. Після узгодження ці діаграми передаються у тікетах команді розробки. Важливим моментом є той факт, що дані діаграми можуть бути представлені не лише як графічне відображення, а мають також текстове відображення, яке зберігається у xml-форматі. Завдяки цьому вони можуть розглядатися як програмні інструкції.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:camunda="http://camunda.org/schema/1.0/bpmn"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC" xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI" xmlns:signavio="http://www.signavio.com" id="sid
0b0aaa25-3baf-4875-9d7a-0907d599a9ef" targetNamespace="http://www.omg.org/spec/BPMN/20100524/MODEL" exporter="Camunda Modeler" exporterVersion="3.5.0"
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd">
  <dataStore id="FinancialAccountingSystem" name="Financial Accounting System" isUnlimited="false">
    <dataState id="DataState_1" />
  </dataStore>
  <message id="foxMessage_en" name="foxMessage_en" />
  <collaboration id="collaboration_3">
    <participant id="Process_Engine_1" name="Invoice Receipt" processRef="invoice" />
  </collaboration>
  <process id="invoice" name="Invoice Receipt" isExecutable="true" camunda:versionTag="V2.0" camunda:historyTimeToLive="45">
    <laneSet id="laneSet_5">
      <lane id="Accountant" name="Accountant">
        <flowNodeRef>prepareBankTransfer</flowNodeRef>
        <flowNodeRef>invoiceProcessed</flowNodeRef>
        <flowNodeRef>ServiceTask_1</flowNodeRef>
      </lane>
      <lane id="teamAssistant" name="Team Assistant">
        <flowNodeRef>reviewSuccessful_gw</flowNodeRef>
        <flowNodeRef>assignApprover</flowNodeRef>
        <flowNodeRef>StartEvent_1</flowNodeRef>
        <flowNodeRef>invoiceNotProcessed</flowNodeRef>
        <flowNodeRef>reviewInvoice</flowNodeRef>
      </lane>
      <lane id="Approver" name="Approver">
        <flowNodeRef>approveInvoice</flowNodeRef>
        <flowNodeRef>invoice_approved</flowNodeRef>
```

Рисунок 3.2 — XML відображення нотації у Camunda Modeler’і

Camunda дозволяє запускати представлені на діаграмах сценарії як процеси за допомогою Tasklist’а, а Cockpit дозволяє відстежувати їх виконання, збираючи дані про ефективність, навантаження та швидкість проходження токенів.

Важливо зазначити, що дані діаграми мають окремий репозиторій на рівні з кодом та також підпадають під правила версіювання на рівні з кодом проекту.

Щодо самого ядра Camunda, то тут необхідно зазначити, що двигун процесів Camunda, на базі свого відкритого API Process Engine Public, працює як сервісно-орієнтований інтерфейс програмування додатків, розроблений для взаємодії з процес-двигуном з використанням Java-додатків. API забезпечує чітке розподілення обов'язків у межах процес-двигуна, розділяючи завдання, такі як Репозиторій процесів, Взаємодія з процесами в режимі реального часу та Управління завданнями, на окремі служби [37].

Публічний API використовує патерн доступу типу команди: потоки, які входять у процес-двигун, направляються через Перехоплювач команд, який використовується для налаштування контексту потоку, такого як транзакції.

BPMN 2.0 Core Engine є основою процес-двигуна. Він включає легкий виконавчий двигун для графових структур (PVM - Віртуальна машина процесів), парсер BPMN 2.0, відповідальний за перетворення XML-файлів BPMN 2.0 у об'єкти Java, та набір реалізацій поведінки BPMN. Ці реалізації надають логіку для конструкцій BPMN 2.0, таких як ворота або сервісні завдання.

Для обробки асинхронних фонових завдань, таких як таймери чи асинхронні продовження в процесі, Camunda використовує Job Executor. Цей компонент вирішально важливий для ефективної обробки завдань, які відбуваються поза поточним ходом процесу.

Для забезпечення постійності стану екземпляра процесу процес-двигун включає окремий шар зберігання. Цей шар відповідає за зберігання необхідної інформації в реляційній базі даних. Для ефективного відображення об'єктів в реляційну базу даних використовується маппінговий двигун MyBatis, що поліпшує загальну ефективність управління даними.

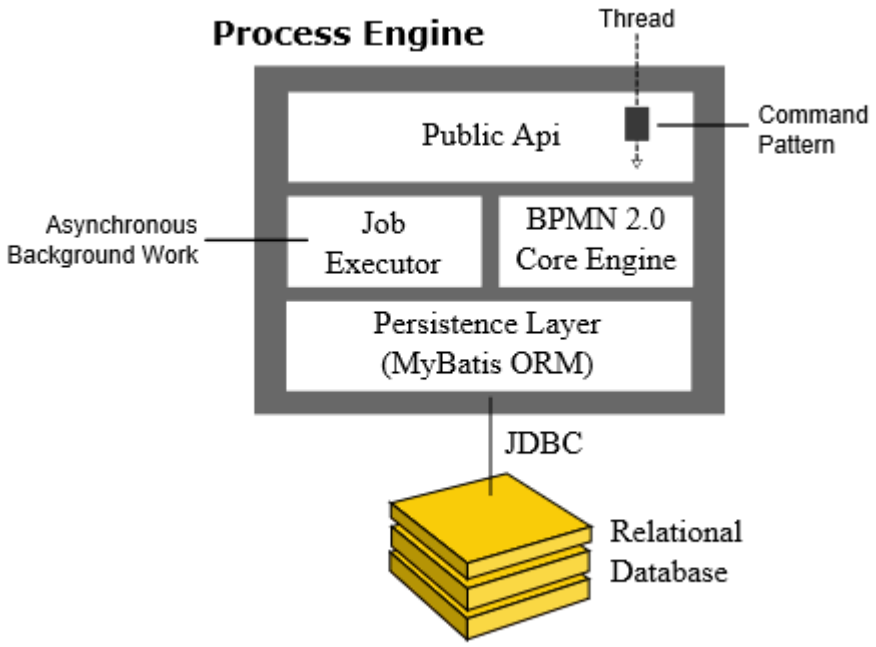


Рисунок 3.3 — Архітектура процесного ядра платформи Camunda 7[37]

Керуючись зазначеними перевагами, було прийнято рішення спробувати піти далі з інтеграцією Camunda у проєкті і організувати не лише графічну оркестрацію бізнес-процесів, а й впровадити її до системи на рівні з іншими мікросервісами для їх оркестрації.

Це стало можливе за рахунок впровадження архітектури, орієнтованої на події (event-driven architecture) на боці Camunda, яка детально описана у статті [38]. Згідно даного підходу централізований потік робочих процесів, орієнтований на події, обробляється за допомогою брокера повідомлень. Детальніше даний підхід наведений на рисунку нижче.

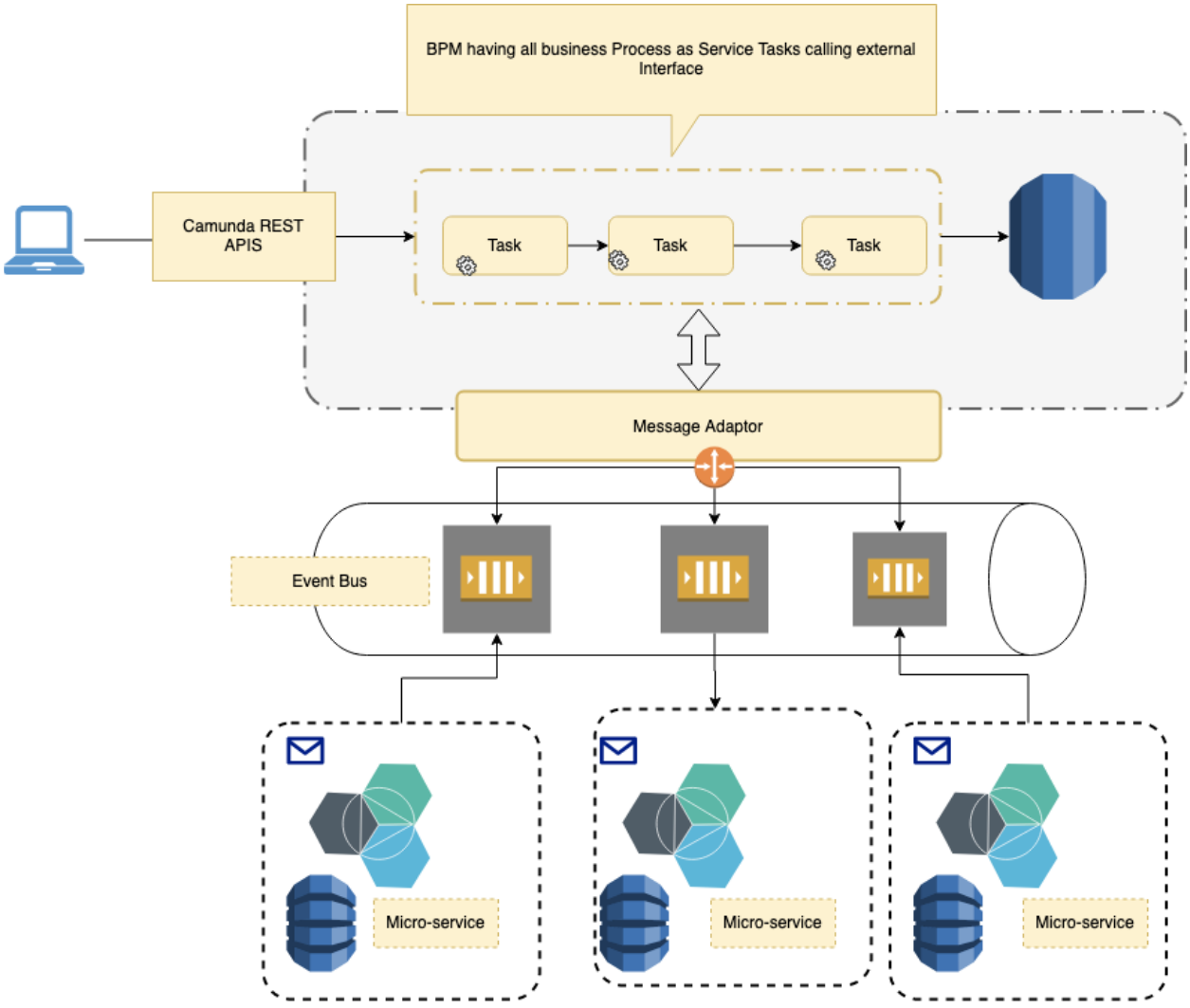


Рисунок 3.4 — Приклад використання Camunda в подія-орієнтованій архітектурі за допомогою BPMN [38]

Кожен бізнес-процес матиме адаптер брокера повідомлень для відправлення повідомлення через чергу (SQS або Kafka), і асинхронно після виконання завдання потік буде знати стан цього запиту. Оркестрація, орієнтована на події, з використанням BPMN (мови моделювання бізнес-процесів) може бути спрощена за допомогою ZeeBee — двигуна для оркестрації мікросервісів, який дозволяє визначати, оркеструвати та контролювати бізнес-процеси через мікросервіси [39].

У цій архітектурі ми взаємодіємо з Camunda за допомогою інтерфейсу REST для запуску бізнес-процесу, визначеного в робочому потоці, який може бути створений за допомогою Camunda Modeler. Важливо відзначити, що нами був написаний власний адаптер сервісу на мові Java, який може взаємодіяти з системою подій.

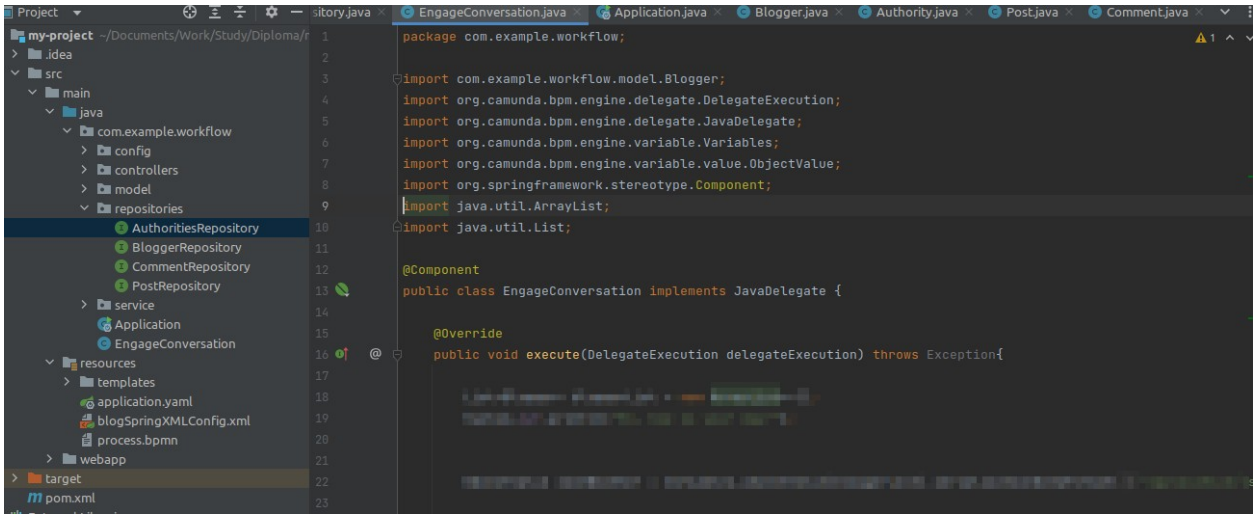


Рисунок 3.5 — Адаптер сервісу для роботи з Camunda

На жаль, ми не можемо розкрити його вміст через NDA, проте важливо відзначити специфіку його роботи, а саме:

- адаптер-клас має імплементувати JavaDelegate з пакету org.camunda.bpm.engine.delegate;
- в адаптер-класі має бути переписаний метод execute.

Проте це не все, що необхідно зробити для того, щоб код на базі Spring Boot міг ефективно взаємодіяти з Camunda. У структурі проекту, серед

ресурсів — /resources, має бути файл з налаштуваннями bpmn-нотацій process.bpmn, якій буде містити налаштування, які будуть зв’язувати біни та класи Spring Boot із елементами діаграми Camunda.

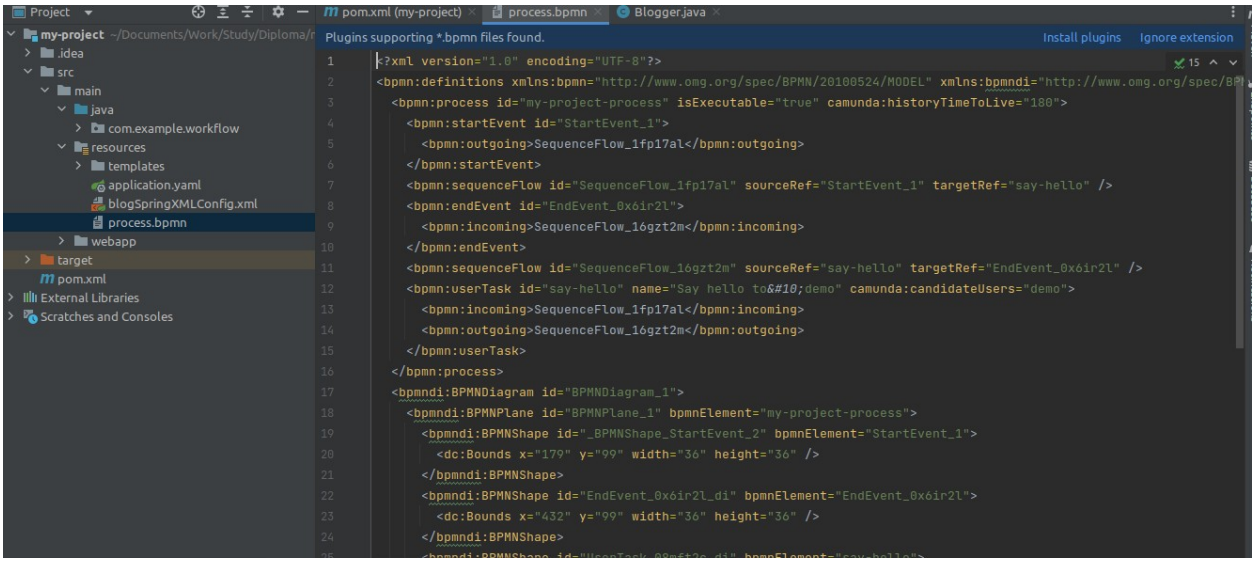


Рисунок 3.6 — Вміст process.bpmn

Строка:

```
<bpmn:process id="my-project-process" isExecutable="true" camunda:historyTimeToLive="180">
```

безпосередньо зв’язує код з діаграмою з Camunda-Modeler’а і надає можливість керувати процесами, які ми плануємо оркеструвати за допомогою двигуна Camunda. Також важливо вказати залежність у pom.xml:

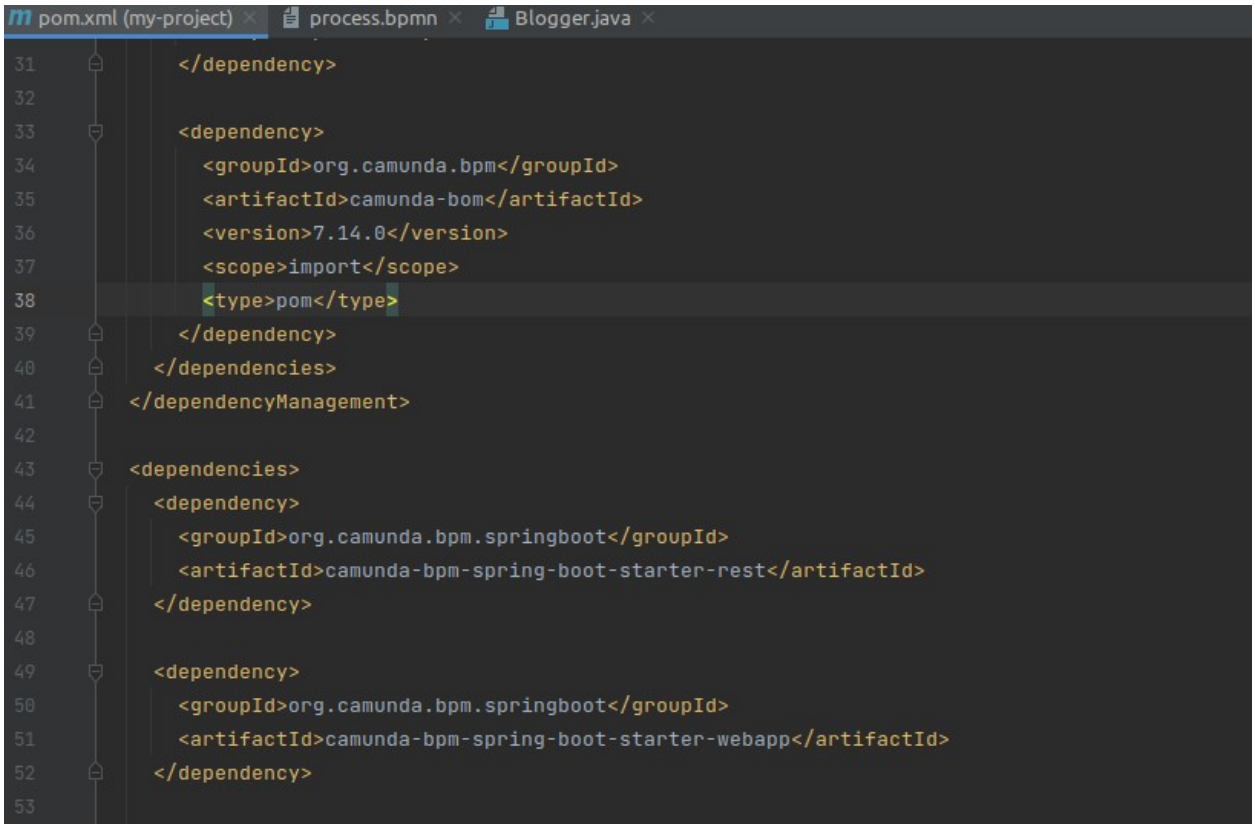


Рисунок 3.7 — Вказання залежностей у pom.xml для інтеграції проекту з Camunda

Згідно [40, 41] Camunda підтримує більше 30 типів подій, як дозволяють планувати їх за часом, подією, умовами, сигналами, помилками, скасуванням, тощо. Різноманіття дій які можуть виконуватися дійсно вражає і безсумно позитивним боком такої імплементації є ще той факт, що Camunda підтримує не лише вбудовану (Embedded), а й винесену (Standalone) реалізацію [37], що надає можливість для горизонтального масштабування за умов зростання навантаження.

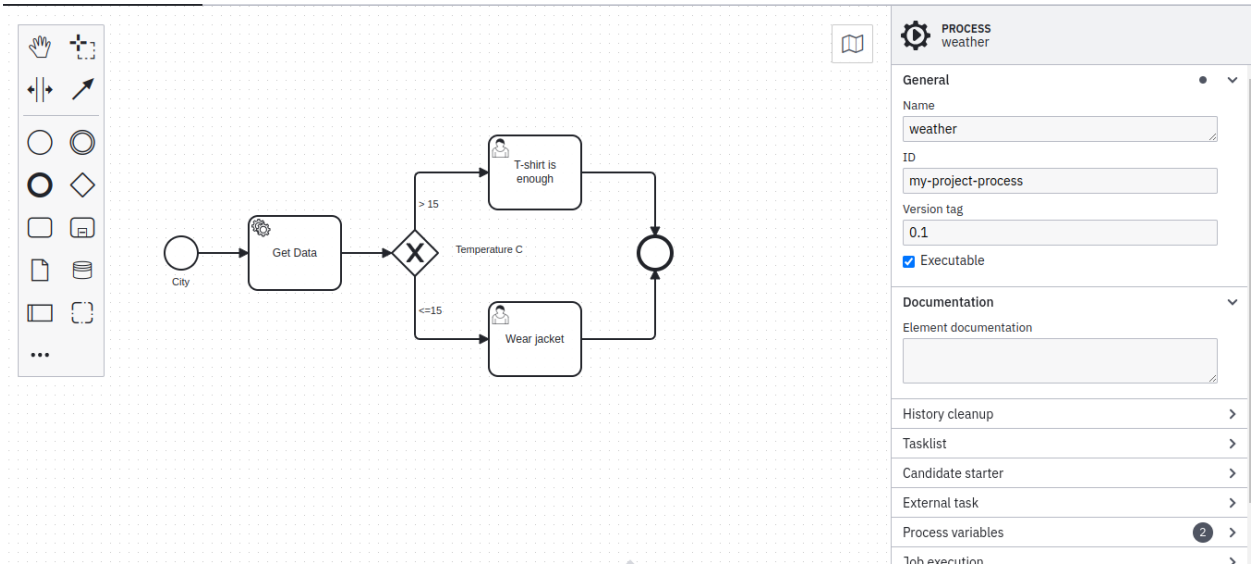


Рисунок 3.8 — Діаграма в Camunda-Modeler, яка зв’язана зі Spring Boot проектом за допомогою process.bpmn

Проте така імплементація має і низку недоліків [42], зокрема:

- Масштабування:

➢ Вертикальне масштабування: Одна з проблем може виникнути при спробі масштабувати систему вертикально (збільшення потужності окремого сервера). Camunda може потребувати значних ресурсів, і це може стати обмеженням для вертикального масштабування.

➢ Горизонтальне масштабування: Деякі складні сценарії масштабування горизонтально можуть заблокувати системи через тривалий час виконання, особливо коли необхідно розподіляти токени між різними екземплярами Camunda.

- Паралельна обробка процесів:

➢ Загальна токенізація: Якщо у вас є велика кількість екземплярів процесів, які використовують однакові та ресурсозалежні сервіси, може виникнути конфлікт за ресурси, оскільки токени будуть конкурувати між собою.

➢ Синхронізація та стан: У складних паралельних структурах може бути важко керувати станом та взаємодією між різними токенами, особливо якщо є необхідність в синхронізації їхньої роботи.

- Управління ресурсами: При великому обсязі та інтенсивності виконання процесів може виникнути проблема з управлінням ресурсами на сервері, де запущений Camunda Engine.
- Системні вимоги: Обробка великих обсягів даних або великої кількості інстансів процесів може призвести до підвищеного використання пам'яті та обчислювальних ресурсів.
- Підтримка асинхронних операцій: Деякі сценарії використання, зокрема ті, що потребують обробки великого обсягу асинхронних завдань, можуть вимагати додаткових оптимізацій та підходів.

Таким чином, за допомогою Camunda-Modeler, орієнтованого на події, ми можемо вирішити проблеми, що виникають у традиційному підході без використання BPM, а саме:

1. Явно визначати та моделювати робочі процеси, що охоплюють кілька мікросервісів.

2. Отримувати детальну інформацію про продуктивність робочого процесу та розуміти місця, де виникають проблеми.

3. Оркеструвати мікросервіси, що реалізують визначений робочий процес, для забезпечення виконання всіх екземплярів робочого процесу згідно з планом, навіть якщо виникають проблеми.

Проте, не можна ігнорувати потенційні проблеми, щоб за рахунок правильного використання відповідного інструменту істотно покращити та підвищити ефективність системи, а не завести себе у глухий кут.

3.2 Тестування бізнес-процесів за допомогою jMeter

У попередньому розділі ми звертали увагу, що команда досить багато уваги приділяє тестуванню якості коду і кожне зливання коду до гілки релізу призводить не лише до збірки процесу, а й автоматичного запуску тестів, які написані в тому числі і за допомогою jMeter. Такі тести є доволі гнучкими у налаштуванні та, одночасно з цим, дозволяють проводити навантажне тестування та стрес-тестування, визначаючи границі можливостей системи.

Після впровадження BPMN-сценаріїв до оркестрації процесів в системі на рівні коду за допомогою Camunda перед нами постало питання як саме ми можемо вимірювати на аналізувати ефективність роботи такого підходу.

За допомогою вбудованого Tasklist ми має можливість відслідковувати та керувати процесами, запускаючи їх, передаючи дані у відповідності до потреби або скасовуючи їх. Можливості Tasklist'а наведені на рисунку нижче.

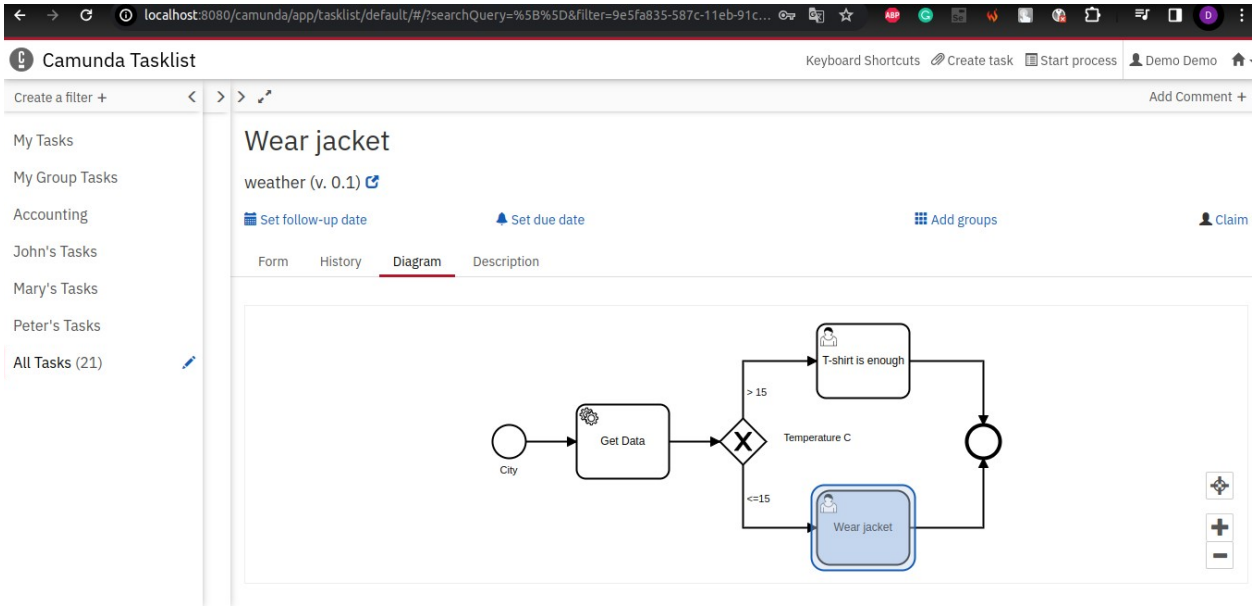


Рисунок 3.9 — Приклад роботи Tasklist'а на проектів

Наступним вбудованим інструментом для аналізу продуктивності роботи системи є Cockpit, де ми маємо змогу детальніше проаналізувати окремі складові процесу. Тут ми можемо побачити які вбудовані інстанси і в

якій кількості задіяні в роботі певного процесу, чи траплялися помилки при виконанні певного сценарію, які роботи задіяні при певному процесі.

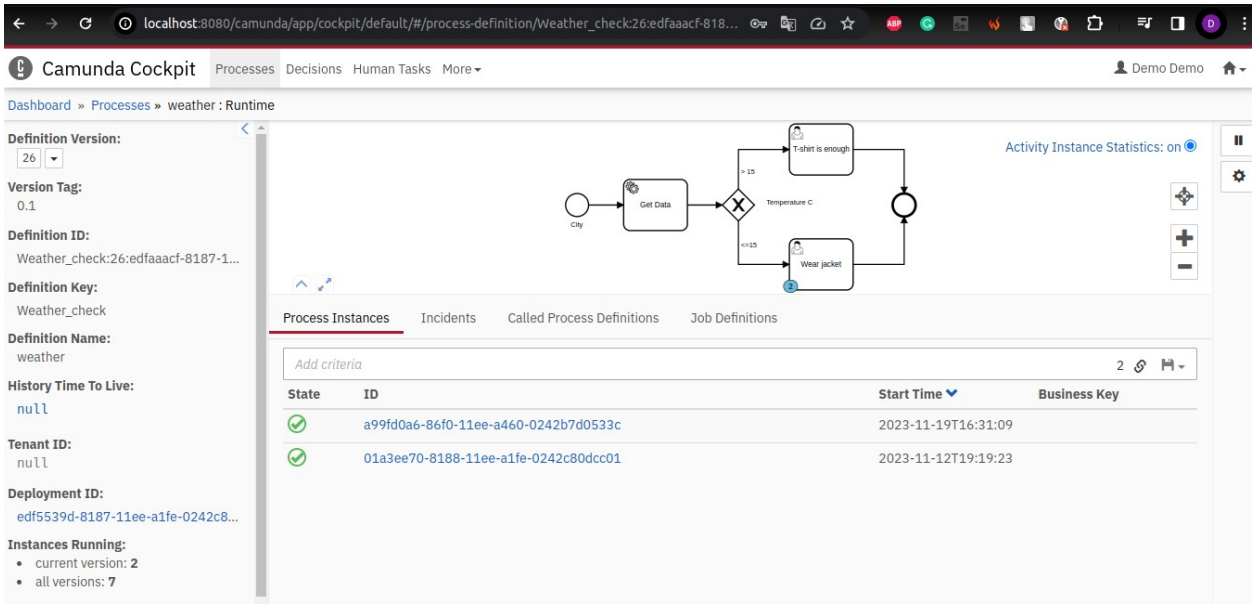


Рисунок 3.10 — Приклад використання Cockpit’а для аналізу продуктивності процесу

Окрім цього, можна побачити які значення змінних та задачі викликаються в процесі і які параметри передаються на етапах. Приклад для нашої задачі наведений нижче.

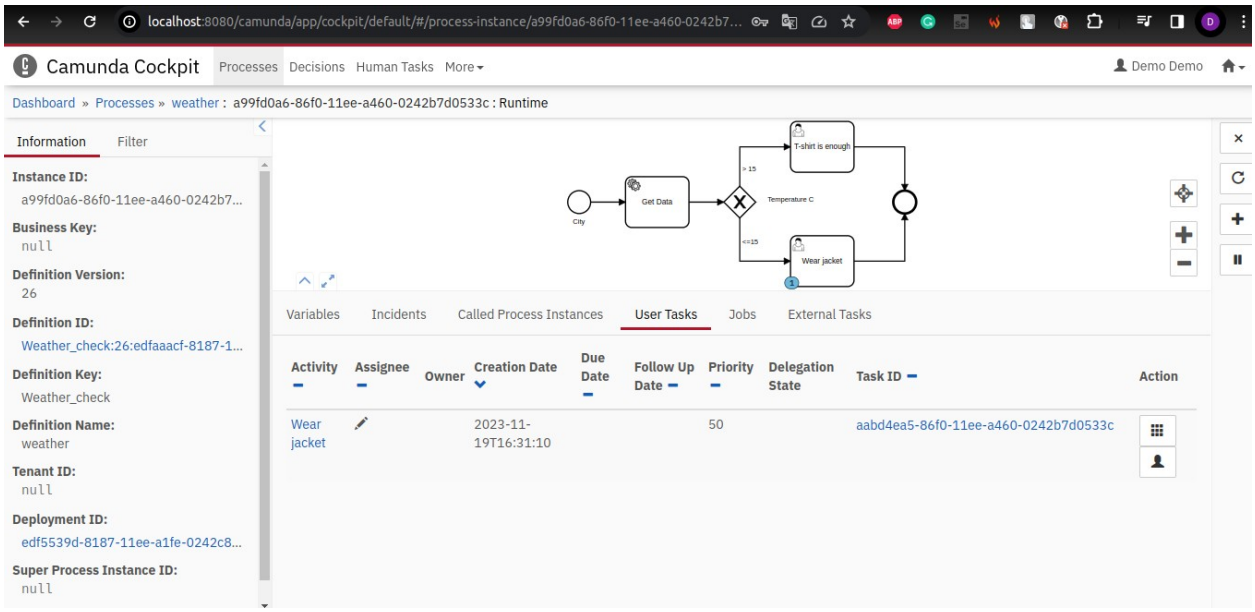


Рисунок 3.11 — Аналіз користувацьких задач за допомогою Cockpit’у

Проте, Tasklist та Cockpit дають лише спрощене уявлення про певний процес з позиції тестування. Адже змушують користувача вводити дані вручну, що за умов необхідності великої кількості тестів не досить зручно. Саме тому для автоматизації тестів нами був обраний jMeter.

jMeter не має можливості спостерігати за роботою інстансів у Cockpit’і, проте дозволяє запускати велику кількість тестів без необхідності керування кожним з них вручну. Однак для роботи з ним ми маємо використовувати АПІ-запити і Camunda надає таку можливість.

Починаючи з версії Camunda BPM 7.0, було введено REST API для взаємодії з двигуном процесів Camunda. Таким чином, з версії 7.0 і пізніших, Camunda підтримує REST API [43].

За допомогою налаштувань плану тестування у jMeter, ми маємо змогу передавати повторювано використовувані змінні

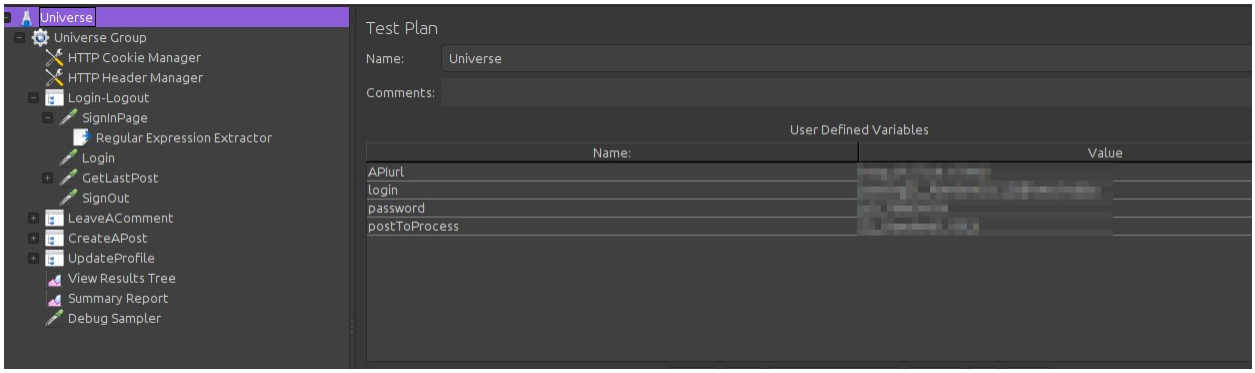


Рисунок 3.12 — Демонстрація налаштувань плану тестування у jMeter

За допомогою налаштувань потоку ми маємо можливість передавати значення щодо кількості потоків, періоду згортання тестування та кількості циклів тестування та багато інших налаштувань, які покликані зробити процес тестування зручнішим. За допомогою простих контролерів ми можемо групувати тести за ознаками у групи, що дозволяє керувати ними більш вибірково. Що стосується самих запитів, то завдяки змінним плану

тестування ми маємо змогу їх повторного використання навіть при викликах через REST API.

В процесі тестування нами широко використовувалися на рівні з http-запитами пре- та пост-процесори. Адже досить часто потрібно зберегти певні дані, наприклад, токен чи значення певної змінної для подальшого тестування. Приклад використання такого пост-процесора наведений нижче.

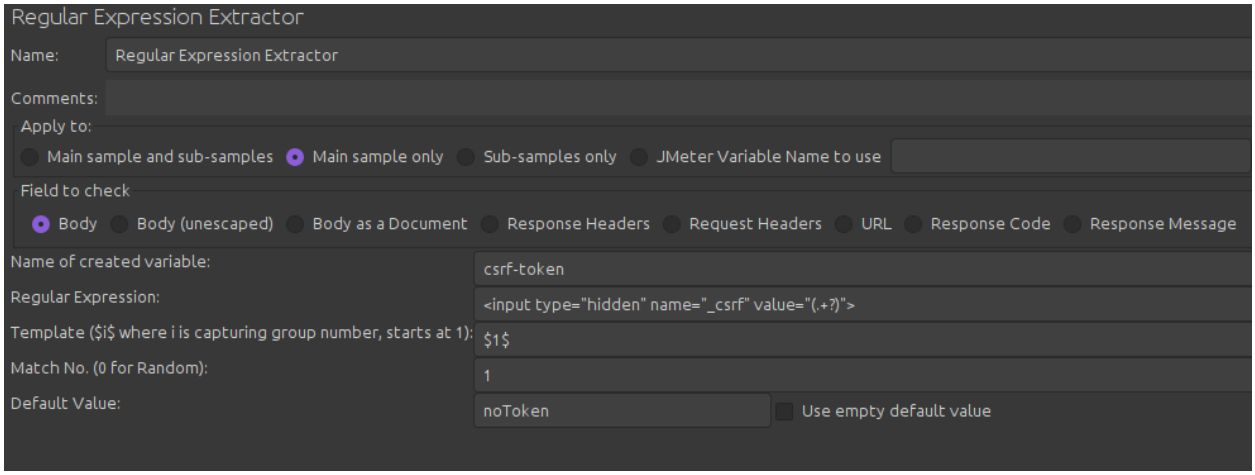


Рисунок 3.13 — Приклад використання пост-процесора для отримання csrf-токена

Проте різноманіття пост-процесорів є доволі значним і вони дозволяють працювати не лише з відповідями у html-форматі, а також json, xml, тощо.

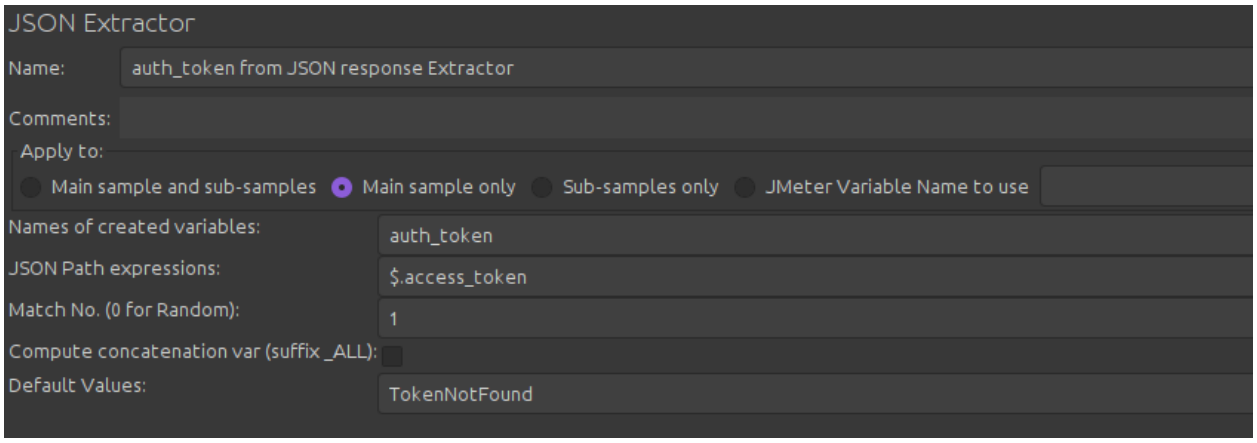


Рисунок 3.14 — Приклад використання пост-процесора для отримання даних із відповіді у форматі json

Також важливо зазначити, що окремі параметри можуть передаватися у самому запиті, що значно покращує гнучкість таких запитів та тестів.

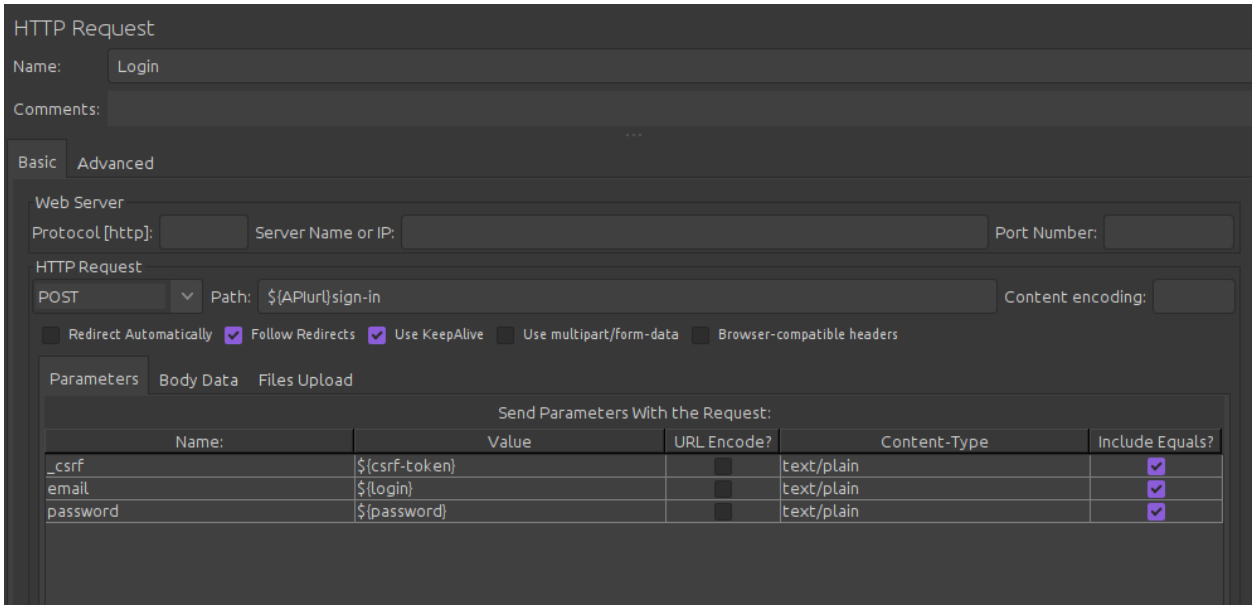


Рисунок 3.15 — Приклад використання параметрів запиту

При роботі з Camunda ми використовуємо апі, надане безпосередньо нею, передаючи відповідні параметри як зазначено у документації [43]. Перелік апі, яке надає Camunda є доволі різноманітним і дозволяє виконувати всі сценарії не лише графічно, а й суто за його викликами. Одночасно з цим, підтримується доступ до всіх подій та змінних на всіх етапах бізнес-процесів. Отже, ми можемо як розпочати процес, так і передати відповідні дані, імітуючи поведінку користувача, запустити повну подію та ознайомитися з результатами її виконання.

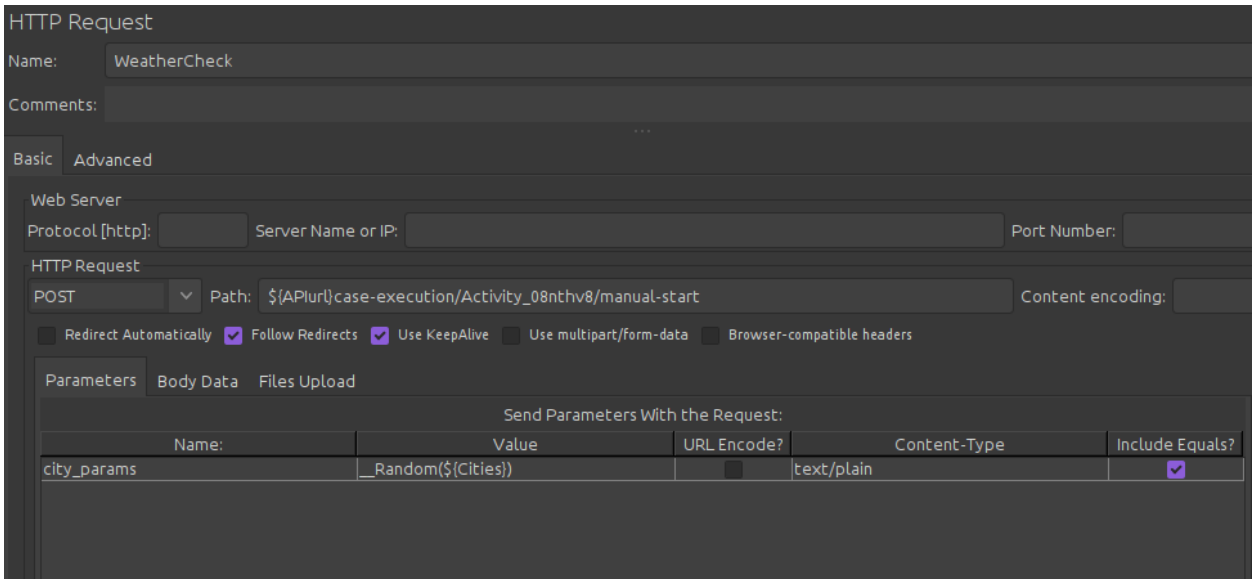


Рисунок 3.16 — Приклад запиту до процесу у Camunda

В свою чергу, jMeter має можливості для управління запитами та обробки результатів їх виконання. Так, наприклад, в залежності від потреб довільні значення можуть обиратися з елементу конфігурації User Defined Variables.

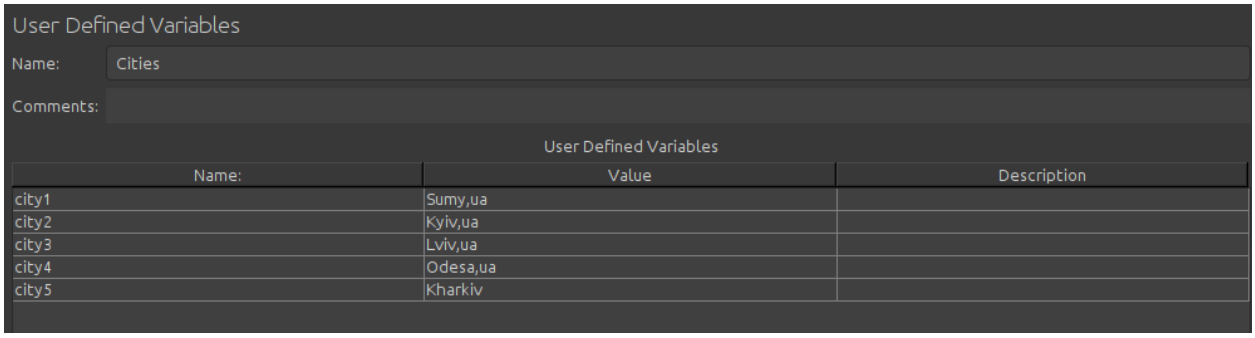


Рисунок 3.17 — Приклад User Defined Variables

Таким чином, jMeter повністю задовольняє нашу потребу у тестуванні сервіса з оркестрації процесів, розробленого на базі standalone інстансу Camunda. Проте він також може бути використаний для тестування самих діаграм у Camunda, розширюючи можливості Tasklist і Cockpit’a. jMeter надає широкий спектр можливостей як з позиції широти сценаріїв, які можна

реалізувати за його допомоги, так і навантаження, яке можна створити за його допомогою, а отже є чудовим інструментом тестування. В свою чергу Camunda також надає всі можливості для проведення такого тестування і їх поєднання є цілком вдалим та доповнюючим.

Висновки

В результаті виконання кваліфікаційної роботи магістра була розроблена інформаційна система автоматизованого тестування бізнес-процесів у вигляді сервісу на базі Camunda, а також підготовані автоматизовані тести для вантажного тестування за допомогою jMeter.

В результаті підготовки дипломного проекту була доведена актуальність предметної області дослідження.

У проектній частині кваліфікаційної роботи магістра було розглянуто існуючі підходи до розуміння бізнес-процесів та проведений аналіз підходів до їх документування. Розглянуто місце Camunda як інструменту з оркестрації бізнес-процесів у документуванні та прототипуванні бізнес-процесів за допомогою графічних нотацій на базі BPMN. Проведений аналіз засад розробки бази дослідження та запропоновані підходи щодо задоволення поточних потреб у розробці програмного забезпечення для оркестрації процесів бази дослідження.

У практичній частині розроблено на базі Spring Boot та Camunda сервіс, якій дозволяє оркеструвати процеси, побудовані у Modeler Camunda, що істотно пришвидшило процес розробки за рахунок скорочення часу від опису задачі до її програмної реалізації. Для даного сервісу розроблені також автоматизовані тести на базі jMeter, які можуть виконуватися як вбудований крок CI/CD при виконанні збірки проекту засобами Github Actions.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Устинова О. Нова нормальність вимагає від бізнес-лідерства швидкої реакції та критичного мислення. KPMG Review Magazine. 2020. № 8. С. 15–19.
2. Чому ваша компанія має діджиталізуватись і 4 приклади того, як це робити. URL: https://eba. com.ua/chomu-vasha-kompaniya-maye-didzhytalizuvatys-ta-4-pryklady-togo-yak-tse-robyty/ (дата звернення 23.10.2023).
3. A Quick Start Guide To Process Management for the Enterprise. agilepoint. 2022. URL: https://www.agilepoint. com/insights/business-process-management/a-quick-start-guide-to-business-process-management-for-theenterprise (дата звернення: 28.10.2023).
4. Business Process Management. Automation Anywhere. 2022. URL: https://www.automationanywhere.com/ rpa/business-process-management (дата звернення: 11.11.2023).
5. Orekhova T., Tertychnyi Ya. Current trends of electronic commercial development in the world markets. Business Management. 2019. Vol. 4. P. 39-51.
6. Digitalization in finance and accounting. URL: https://www.pwc.de/de/im-fokus/digitaleabschlusspruefung/ pwc-digitalisation-in-finance-2018.pdf (дата звернення: 29.10.2023).
7. Document Management Process. URL: https://www.uml-diagrams.org/document-management-uml-activity-diagram-example.html (дата звернення 03.11.2023)
8. Harris E. What Problem does Business Process Management Solve? URL: https://www.processexcellencenetwork. com/business-process-management-

bpm/ articles/what-problem-does-business-processmanagement (дата звернення 03.11.2023)

9. Khadzhynov I., Kovalska L., Taranych A., Boyko A., Samiilenko H. Determining impact of digitalization on the potential-forming space of regions in the context of sustainable development. Amazonia Investiga. 2022. Volume 11. Issue 50. P. 272–281.
10. Zhmykov Y. Нотація моделювання бізнес-процесів 2.0+. 2022. URL: https://ux.pub/zhmikhov/notatsiia-modieliuvannia-biznies-protsiesiv-20-3nfp#:~:text=%D0%91%D1%96%D0%B7%D0%BD%D0%B5%D1%81%2D%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%20(business%20process),%D1%8F%D0%BA%D1%96%20%D0%BC%D0%B0%D1%8E%D1%82%D1%8C%20%D0%BF%D0%BE%D1%87%D0%B0%D1%82%D0%BE%D0%BA%20%D1%82%D0%B0%20%D0%BA%D1%96%D0%BD%D0%B5%D1%86%D1%8C. (дата звернення: 23.10.2023)
11. ДСТУ ISO 9001:2015 «Системи управління якістю. Вимоги» [Електронний ресурс]. - Режим доступу: http://khoda.gov.ua/image/catalog/files/%209001.pdf (дата звернення 21.10.2023).
12. Кріско Ю. Що таке автоматизація бізнес процесів: визначення, приклади та сервіси. Snowio. 2022. URL: https://snov.io/glossary/ua/business-process-automation-ua/ (дата звернення: 22.10.2023).
13. Діджиталізація аутсорсингу: як по-новому вирішувати звичні завдання. URL: https:// mind.ua/openmind/ 20188945-didzhitalizaciya-autsorsingu-yak-po-novomu-virishuvatizvichni-zavdannya (дата звернення 21.10.2023).

14. Тур О.В., Матусевич А.С. Управління бізнес-процесами на підприємстві. Ефективна економіка. 2018. № 6. URL: http://www.economy.nayka.com.ua/pdf/6_2018/59.pdf (дата звернення 23.10.2023)
15. Що таке автоматизація бізнес-процесів? from ua. 2022. URL: https://from-ua.info/shcho-takeavtomatyzatsiia-biznes-protsesiv/ (дата звернення: 10.11.2023).
16. Lawton G., Tucci L. Business Process Automation. TechTarget. 2022. URL: https://www.techtarget.com/ searchcio/definition/business-process-automation (дата звернення: 20.10.2023).
17. Король С. Я., Клочко А. О. Цифрові технології в обліку й аудиті. Держава та регіони. Серія: Економіка та підприємництво. 2020. № 1 (112). С. 170–176. DOI: https://doi.org/10.32840/1814-1161/2020-1-29
18. Лігоненко Л., Хріпко А., Доманський А. Зміст та механізм формування стратегії діджиталізації в бізнес-організаціях. Інтернаука. Економічні науки. 2018. Випуск № 22 (62). 2 т. С. 21-24.
19. Моделювання бізнес-процесів та управління IT-проектами : навчальний посібник [Електронний ресурс] / Є. М. Крижановський, А.Р. Ящолт, С.О. Жуков, О. М. Козачко – Вінниця : ВНТУ, 2018. – (PDF, 91 с.) URL: https://ecopy.posibnyky.vntu.edu.ua/txt/2018/Kryzanovsk_yascholt_modelyuvanna_np_p024.pdf (дата звернення: 23.10.2023).
20. Order Fulfillment and Procurement. URL: https://cloud.trisotech.com/bpmnquickguide/bpmn-quick-guide/bpmn-examples.html (дата звернення 03.11.2023)
21. Соловйова І. Коли компанії потрібна автоматизація бізнес процесів? Економічна правда. 2021. URL: https://www.epravda.com.ua/columns/2021/10/27/679100/ (дата звернення: 25.10.2023).

22. Top Competitors and Alternatives of Camunda. URL: https://6sense.com/tech/workflow-automation/camunda-market-share (дата звернення 03.11.2023)
23. Top competitors of Camunda. URL: https://www.datanyze.com/market-share/business-process-management--452/camunda-market-share (дата звернення 03.11.2023)
24. Jalli A. 5 Best Business Automation Software of 2023 (Complete Guide). codingem.com. 2022. URL: https://www.codingem.com/best-business-automation-software/ (дата звернення: 02.11.2023).
25. Microsoft Annual Report 2021. Microsoft. 2022. URL: https://www.microsoft.com/investor/reports/ar21/ index.html (дата звернення: 20.10.2023).
26. RPA – роботизована автоматизація процесів. 2019. URL: https://coprime.com.ua/ru/blog/ 2019/07/31/rpa (дата звернення: 23.10.2023).
27. RPA_Robotic process automation. 2019. URL: http:// www.tadviser.ru/index.php (дата звернення: 23.10.2023).
28. Steward J. How much Does Marketing Automation Software Cost? Comparison of Pricing Plans. Finance Online. 2022. URL: https://financesonline.com/how-much-does-marketing-automation-software-cost-comparison-of-pricing-plans/ (дата звернення: 12.11.2023).
29. Вишневський Ю. Топ-25 лідерів діджиталізації. URL: https://www.dsnews.ua/ukr/future/top-25- liderov-didzhitalizatsii-01062020220000 (дата звернення: 23.10.2023).

30. Group: SpringFramework Boot. URL: https://mvnrepository.com/artifact/org.springframework.boot (дата звернення: 10.11.2023).
31. GIT Branching Strategies. URL: https://medium.com/cloudnloud/git-branching-strategies-a6eafe4541ae (дата звернення: 10.11.2023).
32. GitHub Actions documentation. URL: https://docs.github.com/en/actions (дата звернення: 10.11.2023).
33. jMeter Introduction. URL: https://www.mygreatlearning.com/jMeter/tutorials/jMeter-introduction (дата звернення: 10.11.2023).
34. What can I do with it? URL: https://jMeter.apache.org/ (дата звернення: 10.11.2023).
35. jMeter Tutorial: Getting Started With the Basics. URL: https://www.blazemeter.com/blog/jMeter-tutorial (дата звернення: 10.11.2023).
36. Camunda Platform 7: Introduction. URL: https://docs.camunda.org/manual/7.20/introduction/ (дата звернення: 10.11.2023).
37. Architecture Overview. URL: https://docs.camunda.org/manual/7.20/introduction/architecture/ (дата звернення: 10.11.2023).
38. Event-driven architecture using camunda as Microservice workflow orchestration. URL: https://tkssharma.com/Event-driven-architecture-using-camunda-Microservice-workflow-orchestration/ (дата звернення: 10.11.2023).

39. Zeebe URL: https://docs.camunda.io/docs/components/zeebe/zeebe-overview/ (дата звернення: 10.11.2023)
40. All BPMN 2.0 Symbols explained with examples. URL: https://camunda.com/bpmn/reference/ (дата звернення: 10.11.2023)
41. BPMN 2.0 Implementation Reference URL: https://docs.camunda.org/manual/7.20/reference/bpmn20/ (дата звернення: 10.11.2023)
42. Why process orchestration needs advanced workflow patterns. URL: https://camunda.com/blog/2022/07/why-process-orchestration-needs-advanced-workflow-patterns/ (дата звернення: 10.11.2023)
43. OpenAPI URL: https://docs.camunda.org/manual/7.14/reference/rest/openapi/ (дата звернення: 10.11.2023)

Додаток А

Лістинг програмного коду основних модулів

фрагмент pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example.workflow</groupId>
  <artifactId>my-project</artifactId>
  <version>1.0.0-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.0.2</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
```

```
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-dependencies</artifactId>
      <version>3.1.1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
    <dependency>
      <groupId>org.camunda.bpm</groupId>
      <artifactId>camunda-bom</artifactId>
      <version>7.14.0</version>
      <scope>import</scope>
      <type>pom</type>
    </dependency>
  </dependencies>
</dependencyManagement>
<dependencies>
  <dependency>
    <groupId>org.camunda.bpm.springboot</groupId>
    <artifactId>camunda-bpm-spring-boot-starter-rest</artifactId>
  </dependency>
  <dependency>
    <groupId>org.camunda.bpm.springboot</groupId>
    <artifactId>camunda-bpm-spring-boot-starter-webapp</artifactId>
  </dependency>
  <dependency>
    <groupId>org.camunda.bpm</groupId>
    <artifactId>camunda-engine-plugin-spin</artifactId>
  </dependency>
```

```
    <dependency>
      <groupId>org.camunda.spin</groupId>
      <artifactId>camunda-spin-dataformat-all</artifactId>
    </dependency>
  </dependencies>
</project>
```

process.bpmn

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions
xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
xmlns:camunda="http://camunda.org/schema/1.0/bpmn" id="Definitions_0fr9mxs"
targetNamespace="http://bpmn.io/schema/bpmn">
              <bpmn:process id="my-project-process" isExecutable="true"
camunda:historyTimeToLive="180">
    <bpmn:startEvent id="StartEvent_1">
      <bpmn:outgoing>SequenceFlow_1fp17al</bpmn:outgoing>
    </bpmn:startEvent>
      <bpmn:sequenceFlow id="SequenceFlow_1fp17al" sourceRef="StartEvent_1"
targetRef="say-hello" />
    <bpmn:endEvent id="EndEvent_0x6ir2l">
      <bpmn:incoming>SequenceFlow_16gzt2m</bpmn:incoming>
    </bpmn:endEvent>
```

<bpmn:sequenceFlow id="SequenceFlow_16gzt2m" sourceRef="say-hello" targetRef="EndEvent_0x6ir2l" />

<bpmn:userTask id="say-hello" name="Say hello to
demo" camunda:candidateUsers="demo">

<bpmn:incoming>SequenceFlow_1fp17al</bpmn:incoming>

<bpmn:outgoing>SequenceFlow_16gzt2m</bpmn:outgoing>

</bpmn:userTask>

</bpmn:process>

<bpmndi:BPMNDiagram id="BPMNDiagram_1">

<bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="my-project-process">

<bpmndi:BPMNShape id="_BPMNShape_StartEvent_2" bpmnElement="StartEvent_1">

<dc:Bounds x="179" y="99" width="36" height="36" />

</bpmndi:BPMNShape>

<bpmndi:BPMNShape id="EndEvent_0x6ir2l_di" bpmnElement="EndEvent_0x6ir2l">

<dc:Bounds x="432" y="99" width="36" height="36" />

</bpmndi:BPMNShape>

<bpmndi:BPMNShape id="UserTask_08mft2c_di" bpmnElement="say-hello">

<dc:Bounds x="270" y="77" width="100" height="80" />

</bpmndi:BPMNShape>

<bpmndi:BPMNEdge id="SequenceFlow_1fp17al_di" bpmnElement="SequenceFlow_1fp17al">

<di:waypoint x="215" y="117" />

<di:waypoint x="270" y="117" />

</bpmndi:BPMNEdge>

<bpmndi:BPMNEdge id="SequenceFlow_16gzt2m_di" bpmnElement="SequenceFlow_16gzt2m">
<di:waypoint x="370" y="117" />
<di:waypoint x="432" y="117" />
</bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</bpmn:definitions>

weather.bpmn

<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions
xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:camunda="http://camunda.org/schema/1.0/bpmn"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI" id="Definitions_1gpeisp" targetNamespace="http://bpmn.io/schema/bpmn" exporter="Camunda Modeler" exporterVersion="5.16.0">
<bpmn:process id="my-project-process" name="weather" isExecutable="true" camunda:versionTag="0.1">
<bpmn:startEvent id="Name_City" name="City">
<bpmn:extensionElements>
<camunda:formData>
<camunda:formField id="city_params" label="City" type="string" defaultValue="Sumy,ua" />

```
      </camunda:formData>
    </bpmn:extensionElements>
    <bpmn:outgoing>Flow_0hgnvx8</bpmn:outgoing>
  </bpmn:startEvent>
        <bpmn:sequenceFlow id="Flow_0hgnvx8" sourceRef="Name_City"
targetRef="Activity_08nthv8" />
  <bpmn:serviceTask id="Activity_08nthv8" name="Get Data">
    <bpmn:extensionElements>
      <camunda:connector>
        <camunda:inputOutput>
                                                    <camunda:inputParameter
name="url">http://api.openweathermap.org/data/2.5/weather?q=$
{city_params}&amp;amp;appid=8a1c6703023c2366117dbba551cf9ed7
</camunda:inputParameter>
          <camunda:inputParameter name="headers">
            <camunda:map>
              <camunda:entry key="Content-type">application/json</camunda:entry>
            </camunda:map>
          </camunda:inputParameter>
                                                    <camunda:inputParameter
name="method">GET</camunda:inputParameter>
          <camunda:outputParameter name="response">
                                                        <camunda:script
scriptFormat="JavaScript">S(response);</camunda:script>
          </camunda:outputParameter>
          <camunda:outputParameter name="statusCode">
                                                        <camunda:script
scriptFormat="JavaScript">statusCode;</camunda:script>
```

```
        </camunda:outputParameter>
      </camunda:inputOutput>
      <camunda:connectorId>http-connector</camunda:connectorId>
    </camunda:connector>
  </bpmn:extensionElements>
  <bpmn:incoming>Flow_0hgnvx8</bpmn:incoming>
  <bpmn:outgoing>Flow_1sqwjgl</bpmn:outgoing>
</bpmn:serviceTask>
<bpmn:exclusiveGateway id="Gateway_1cg0cyz" name="Temperature C">
  <bpmn:incoming>Flow_1sqwjgl</bpmn:incoming>
  <bpmn:outgoing>Flow_01w2cwb</bpmn:outgoing>
  <bpmn:outgoing>Flow_189045s</bpmn:outgoing>
</bpmn:exclusiveGateway>
    <bpmn:sequenceFlow id="Flow_1sqwjgl" sourceRef="Activity_08nthv8"
targetRef="Gateway_1cg0cyz" />
          <bpmn:sequenceFlow id="Flow_01w2cwb" name="&#62; 15"
sourceRef="Gateway_1cg0cyz" targetRef="Activity_06lcdld">
           <bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">${response.prop("main").prop("temp").numberValue() - 273.15 &gt; 15}</bpmn:conditionExpression>
</bpmn:sequenceFlow>
           <bpmn:sequenceFlow id="Flow_189045s" name="&#60;=15"
sourceRef="Gateway_1cg0cyz" targetRef="Activity_098gk8v">
          <bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">${response.prop("main").prop("temp").numberValue() - 273.15 &lt;= 15}</bpmn:conditionExpression>
</bpmn:sequenceFlow>
<bpmn:endEvent id="Event_03ydsug">
```

```
    <bpmn:incoming>Flow_1gubyv9</bpmn:incoming>
    <bpmn:incoming>Flow_0eqp3i0</bpmn:incoming>
  </bpmn:endEvent>
      <bpmn:sequenceFlow id="Flow_1gubyv9" sourceRef="Activity_06lcdld" targetRef="Event_03ydsug" />
      <bpmn:sequenceFlow id="Flow_0eqp3i0" sourceRef="Activity_098gk8v" targetRef="Event_03ydsug" />
  <bpmn:userTask id="Activity_06lcdld" name="T-shirt is enough">
    <bpmn:extensionElements>
      <camunda:formData>
                <camunda:formField id="temp" label="Temp" type="string" defaultValue="$
{response.prop(&#34;main&#34;).prop(&#34;temp&#34;).numberValue() - 273.15}" />
      </camunda:formData>
    </bpmn:extensionElements>
    <bpmn:incoming>Flow_01w2cwb</bpmn:incoming>
    <bpmn:outgoing>Flow_1gubyv9</bpmn:outgoing>
  </bpmn:userTask>
  <bpmn:userTask id="Activity_098gk8v" name="Wear jacket">
    <bpmn:extensionElements>
      <camunda:formData>
                <camunda:formField id="temp" label="Temp" type="string" defaultValue="$
{response.prop(&#34;main&#34;).prop(&#34;temp&#34;).numberValue() - 273.15}" />
      </camunda:formData>
    </bpmn:extensionElements>
```

```
    <bpmn:incoming>Flow_189045s</bpmn:incoming>
    <bpmn:outgoing>Flow_0eqp3i0</bpmn:outgoing>
   </bpmn:userTask>
  </bpmn:process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_1">
        <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="my-project-process">
                    <bpmndi:BPMNShape id="_BPMNShape_StartEvent_2" bpmnElement="Name_City">
      <dc:Bounds x="179" y="159" width="36" height="36" />
      <bpmndi:BPMNLabel>
       <dc:Bounds x="187" y="202" width="20" height="14" />
      </bpmndi:BPMNLabel>
    </bpmndi:BPMNShape>
                        <bpmndi:BPMNShape id="Activity_196ca0j_di" bpmnElement="Activity_08nthv8">
      <dc:Bounds x="270" y="137" width="100" height="80" />
    </bpmndi:BPMNShape>
                        <bpmndi:BPMNShape id="Gateway_1cg0cyz_di" bpmnElement="Gateway_1cg0cyz" isMarkerVisible="true">
      <dc:Bounds x="425" y="152" width="50" height="50" />
      <bpmndi:BPMNLabel>
       <dc:Bounds x="494" y="167" width="73" height="14" />
      </bpmndi:BPMNLabel>
    </bpmndi:BPMNShape>
                        <bpmndi:BPMNShape id="Event_03ydsug_di" bpmnElement="Event_03ydsug">
      <dc:Bounds x="692" y="159" width="36" height="36" />
```

```
    </bpmndi:BPMNShape>
                        <bpmndi:BPMNShape id="Activity_1buhgmd_di"
bpmnElement="Activity_06lcdld">
      <dc:Bounds x="530" y="50" width="100" height="80" />
    </bpmndi:BPMNShape>
                        <bpmndi:BPMNShape id="Activity_09emckf_di"
bpmnElement="Activity_098gk8v">
      <dc:Bounds x="530" y="230" width="100" height="80" />
    </bpmndi:BPMNShape>
                          <bpmndi:BPMNEdge id="Flow_0hgnvx8_di"
bpmnElement="Flow_0hgnvx8">
      <di:waypoint x="215" y="177" />
      <di:waypoint x="270" y="177" />
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge id="Flow_1sqwjgl_di" bpmnElement="Flow_1sqwjgl">
      <di:waypoint x="370" y="177" />
      <di:waypoint x="425" y="177" />
    </bpmndi:BPMNEdge>
                          <bpmndi:BPMNEdge id="Flow_01w2cwb_di"
bpmnElement="Flow_01w2cwb">
      <di:waypoint x="450" y="152" />
      <di:waypoint x="450" y="90" />
      <di:waypoint x="530" y="90" />
      <bpmndi:BPMNLabel>
        <dc:Bounds x="454" y="118" width="22" height="14" />
      </bpmndi:BPMNLabel>
    </bpmndi:BPMNEdge>
```

```
                                    <bpmndi:BPMNEdge id="Flow_189045s_di"
bpmnElement="Flow_189045s">
      <di:waypoint x="450" y="202" />
      <di:waypoint x="450" y="270" />
      <di:waypoint x="530" y="270" />
      <bpmndi:BPMNLabel>
        <dc:Bounds x="454" y="233" width="26" height="14" />
      </bpmndi:BPMNLabel>
    </bpmndi:BPMNEdge>
                                    <bpmndi:BPMNEdge id="Flow_1gubyv9_di"
bpmnElement="Flow_1gubyv9">
      <di:waypoint x="630" y="90" />
      <di:waypoint x="710" y="90" />
      <di:waypoint x="710" y="159" />
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge id="Flow_0eqp3i0_di" bpmnElement="Flow_0eqp3i0">
      <di:waypoint x="630" y="270" />
      <di:waypoint x="710" y="270" />
      <di:waypoint x="710" y="195" />
    </bpmndi:BPMNEdge>
  </bpmndi:BPMNPlane>
 </bpmndi:BPMNDiagram>
</bpmn:definitions>
```