

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

Розробка інформаційної системи взаємодії провайдера складських послуг з
контрагентами

Виконав: студент групи 4142

(підпис) *Горун І.П.*

Керівник роботи:

(посада, науковий ступінь вчене звання)
викладач

(підпис) *Книрик К.О.*

м. Миколаїв 2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"
Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.
" ____ " _____ 2024 р.

*ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ*
на здобуття ступеня вищої освіти "бакалавр"

Студенту Горуну Ігорю Петровичу

1. Тема роботи: Розробка інформаційної системи взаємодії провайдера складських послуг з контрагентами

Керівник роботи: Книрик К.О.

Затверджені наказом ректора № 243-уч від "18" березня 2024 року.

2. Термін подання роботи: 20 червня 2024 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів)

5. Перелік презентаційних матеріалів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Книрік К.О., викладач		
2	Книрік К.О., викладач		
3	Книрік К.О., викладач		
4	Книрік К.О., викладач		

7. Дата видачі завдання 05 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та опис об'єкта дослідження	8 березня 2024 р.	
2	Вивчення технологій та архітектурних рішень для використання в проекті	22 березня 2024 р.	
3	Формування вимог та розробка концепції системи	5 квітня 2024 р.	
4	Розробка загальносистемних рішень	12 квітня 2024 р.	
5	Опис рішень щодо інформаційного забезпечення	3 травня 2024 р.	
6	Опис рішень щодо математичного, програмного та технічного забезпечення	17 травня 2024 р.	
7	Опис процесу реалізації системи та створення технічної документації	27 травня 2024	
8	Заключення та висновки	31 травня 2024 р.	
9	Редагування та виправлення помилок	3 червня 2024 р.	
10	Створення презентаційних матеріалів	7 червня 2024 р.	

Студент

(підпис)

Горун І.П.

Керівник роботи

(підпис)

Книрік К.О.

АНОТАЦІЯ

Обсяг пояснювальної записки складає 86 сторінок. У роботі наведено 20 ілюстрацій, 28 таблиць, 5 додатків та 36 джерел у списку літератури.

Об'єктом дослідження роботи є підприємство яке надає послуги зберігання товарів без обмеження строку придатності для третіх осіб – контрагентів. Предметна область роботи – діюча система керування складськими операціями та документообігом підприємства.

Метою роботи є підвищення ефективності поточної системи за рахунок автоматизації деяких регулярних ручних операцій а також створення нової, додаткової інформаційної системи для автоматизації складських операцій при взаємодії з інформаційними системами контрагентів підприємства.

В цьому дослідженні використовувалися методи аналізу вимог працівників підприємства, аналізу можливостей поточної системи керування складськими операціями, проектування розподіленої системи обробки інформації, розробки програмного забезпечення та тестування.

Результати роботи включають розширення функціоналу поточної системи відповідно потребам користувачів, та окрему інформаційну систему для обміну даними з контрагентами підприємства.

При розробці проекту була використана мова програмування Python, базова ORM Odoo, сховище даних Redis, брокер повідомлень Dramatiq та фреймворк FastAPI.

Отримані результати можуть бути застосовані малим бізнесом в сфері надання послуг зберігання товарів третім особам з доволі високою ефективністю процесів складських операцій та обміну даними з контрагентами. Висновки та прогнози щодо розвитку проекту підкреслюють важливість та перспективність подальшої розробки та вдосконалення інформаційної системи.

Робота виконана українською мовою.

Ключові слова: склад, послуги, 3PL, провайдер, зберігання, Odoo, API.

ЗМІСТ

АНОТАЦІЯ	4
ЗМІСТ	5
ВСТУП	6
РОЗДІЛ 1 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЗАЄМОДІЇ ПРОВАЙДЕРА СКЛАДСЬКИХ ПОСЛУГ З КОНТРАГЕНТАМИ	8
1.1 Опис об'єкта дослідження та аналіз предметної області	8
1.2 Формування вимог до системи	17
1.3 Розробка концепції інформаційної системи	19
РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ІС	23
2.1 Загальносистемні рішення	23
2.2 Рішення щодо інформаційного забезпечення	24
2.3 Рішення щодо математичного забезпечення	36
2.4 Рішення щодо програмного забезпечення	40
2.4.1 Рішення щодо застосунку «Інтеграція»	41
2.4.2 Рішення щодо застосунку «Черга»	43
2.4.3 Рішення щодо застосунку «Склад»	44
2.5 Рішення щодо технічного забезпечення	46
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ ІС	48
РОЗДІЛ 4 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ	53
ВИСНОВОК	59
ЛІТЕРАТУРА	60
ДОДАТОК А	63
ДОДАТОК Б	68
ДОДАТОК В	79
ДОДАТОК Г	81
ДОДАТОК Д	83

ВСТУП

Кваліфікаційна робота виконана по темі «Розробка інформаційної системи взаємодії провайдера складських послуг з контрагентами».

У роботі розглядається аналіз процесів місцевого підприємства – постачальника послуг складських операцій з точки зору ефективності та автоматизації поточної системи керування складськими операціями, її можливості та недоліки. Особливим питанням є надання можливості клієнтам підприємства – контрагентам – автоматизувати обмін даними із складом для підвищення швидкості операцій та запобігання помилок яким сприяє обмін даними за консервативними методами - паперовими, файловими різних стандартів і таке інше.

Аналіз послуг складських операцій на сучасному ринку України вказав на затребуваність послуг складських операцій для малого бізнесу та потребу в програмному забезпеченні для невеличких постачальників таких послуг. Наразі майже не існує доступного програмного забезпечення для управління складом як послугою для малого бізнесу, тому запропоновано концепцію системи для місцевого постачальника послуг складських операцій.

Метою цієї роботи є підвищення ефективності роботи місцевого постачальника послуг складських операцій за рахунок автоматизації деяких процесів у рамках існуючої інформаційної системи та створення системи для автоматизованої взаємодії з контрагентами.

У розділі 1 описано об'єкт дослідження – підприємство, яке надає послуги складських операцій, проведено аналіз предметної області, знайдені процеси з низькою ефективністю та можливістю автоматизації, розроблена концепція інформаційної системи.

У розділі 2 надано проект інформаційної системи включаючи загальносистемні рішення щодо обміну даними між підсистемами та зовнішніми системами. Інформаційне забезпечення представлено проектом бази даних та стандартами електронних документів для обміну даними між підсистемами та

зовнішніми системами. Математичне забезпечення представлено алгоритмами задачі по аутентифікації зовнішніх систем та авторизації запитів. Програмне забезпечення представлено рішеннями для реалізації функціональних вимог до проекту. Визначені технічні засоби необхідні для експлуатації системи.

У розділі 3 представлена реалізація проекту, інструкція користувача по оновленням системи, інструкція розробника щодо оновлення та розгортання інтеграційної платформи.

У розділі 4 описана інформація про охорону праці і техніку безпеки, що є важливою складовою будь-якого проекту, зокрема в розробці програмного забезпечення.

Реалізація проекту буде досягнута за рахунок новаторського підходу до використання програмних засобів, в тому числі хмарних, які дозволять забезпечити надійність, швидкість та масштабованість системи обміну даними з контрагентами.

Отже, враховуючи затребуваність автоматизованих систем обліку та обміну даними, розробка цієї інформаційної системи є актуальним та перспективним завданням.

РОЗДІЛ 1 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЗАЄМОДІЇ ПРОВАЙДЕРА СКЛАДСЬКИХ ПОСЛУГ З КОНТРАГЕНТАМИ

1.1 Опис об'єкта дослідження та аналіз предметної області

В останні роки в Україні дуже швидко розвивається малий бізнес – підприємці різних напрямків виробництва – від одягу до меблів, інтернет–торгівлі дрібними побутовими товарами, електронікою, мобільними пристроями та іншим.

Складські операції, зберігання та керування товарами – загальна проблема для більшості таких підприємців – це потребує додаткових приміщень, найманих працівників, інформаційних систем та, як результат, додаткових, доволі великих видатків.

Аутсорсинг складських операцій в багатьох випадках є доволі зручним варіантом для невеличкого бізнесу, він отримує повністю або частково закриті питання складського зберігання, логістики та обліку товарів і може зосередитись на більш важливих задачах та зменшити видатки на утримання складу.

Головна ідея провайдера складських операцій полягає в тому, що компанія–власник не займається торгівельною діяльністю, а лише надає послуги завантаження, зберігання, переміщення, пакування та відвантаження товарів на своєму складі для контрагентів, отримуючи від них оплату за ці послуги.

На українському ринку існує доволі багато великих гравців складських послуг, але майже всі вони спрямовані на великий бізнес. Більшість невеличких бізнесів не цікава з комерційної точки зору великим провайдерам, а маленькому бізнесу не цікаві великі провайдери через повільне виконання операцій та доволі дороге обслуговування. Тому підприємці та малий бізнес шукають невеличких провайдерів складських операцій в своєму місті, або навіть районі міста, щоб мати можливість швидко отримувати товари із складу та виконувати відправлення замовникам без довгого очікування.

Об'єктом дослідження є малий бізнес приватного підприємця під публічною назвою «ITStore» який надає послуги зберігання товарів без строку придатності

– переважно комп'ютерних пристроїв та запчастин, мобільних пристроїв та аксесуарів, мережевих пристроїв та інших дрібних товарів не маючих строку зберігання для інтернет–магазинів, локальних миколаївських торгівельних компаній та виробництв.

ITStore працює згідно свідоцтва про державну реєстрацію фізичної особи–підприємця та свідоцтва платника єдиного податку III групи з підтвердженням видом господарської діяльності згідно КВЕД–2010, Клас 52.10 «Складське господарство», який включає діяльність із зберігання та складування всіх видів товарів [1].

ITStore має приміщення – склад, облаштоване під зберігання та складські операції, розташований неподалік від центру міста, що робить його зручним для фірм та магазинів техніки в місті щодо швидкого доступу до їх товарів.

Склад розділений на п'ять основних зон – завантаження (1), зберігання (2), пакування (4), відвантаження (5) та офісну (3) зони. На Рисунку 1.1 зображена схема розташування зон складу.

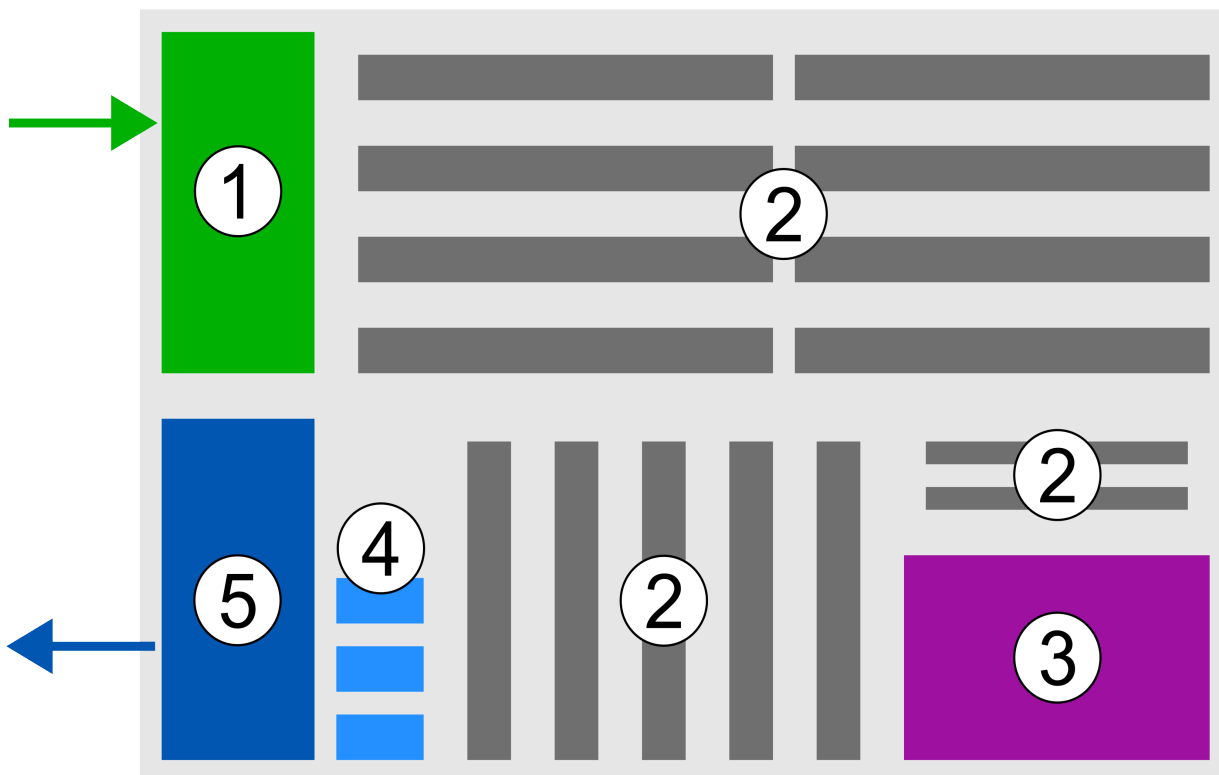


Рисунок 1.1 – Схема розташування зон складу

Зони завантаження та відвантаження це організовані простори з мітками на підлозі для ідентифікації відправлень, підготовлені для прийняття вантажів або їх завантаження в транспорт.

Зона зберігання – найбільша, це облаштовані місця для зберігання товарів – металеві полиці та шафи, які розділені на комірки марковані для ідентифікації належності товарів до контрагентів. На Рисунку 1.2 зображені приклади складських полиць різних розмірів та розміщення пакунків на них.



Рисунок 1.1 – Складські полиці з комірками

Зона пакування – це простір організований таким чином, щоб працівникам було зручно скласти всі зібрані товари, запакувати їх, та відправити у зону відвантаження – там розташовані столи для пакування та полиці для зберігання картонних коробок різних розмірів у які запаковуються товари.

Офісна зона складається з декількох невеличких офісних приміщень для адміністративної діяльності – проведення переговорів з клієнтами, прийняття замовлень, оформлення документів. Також в офісній зоні знаходиться окреме приміщення для серверу, мережевого обладнання та обладнання для відео–нагляду – серверна.

Постійний персонал складу складається з директора, адміністратора, бухгалтера та двох–п'яти операторів, в залежності від завантаженості. Організаційна структура управління складом зображена на Рисунку 1.3.

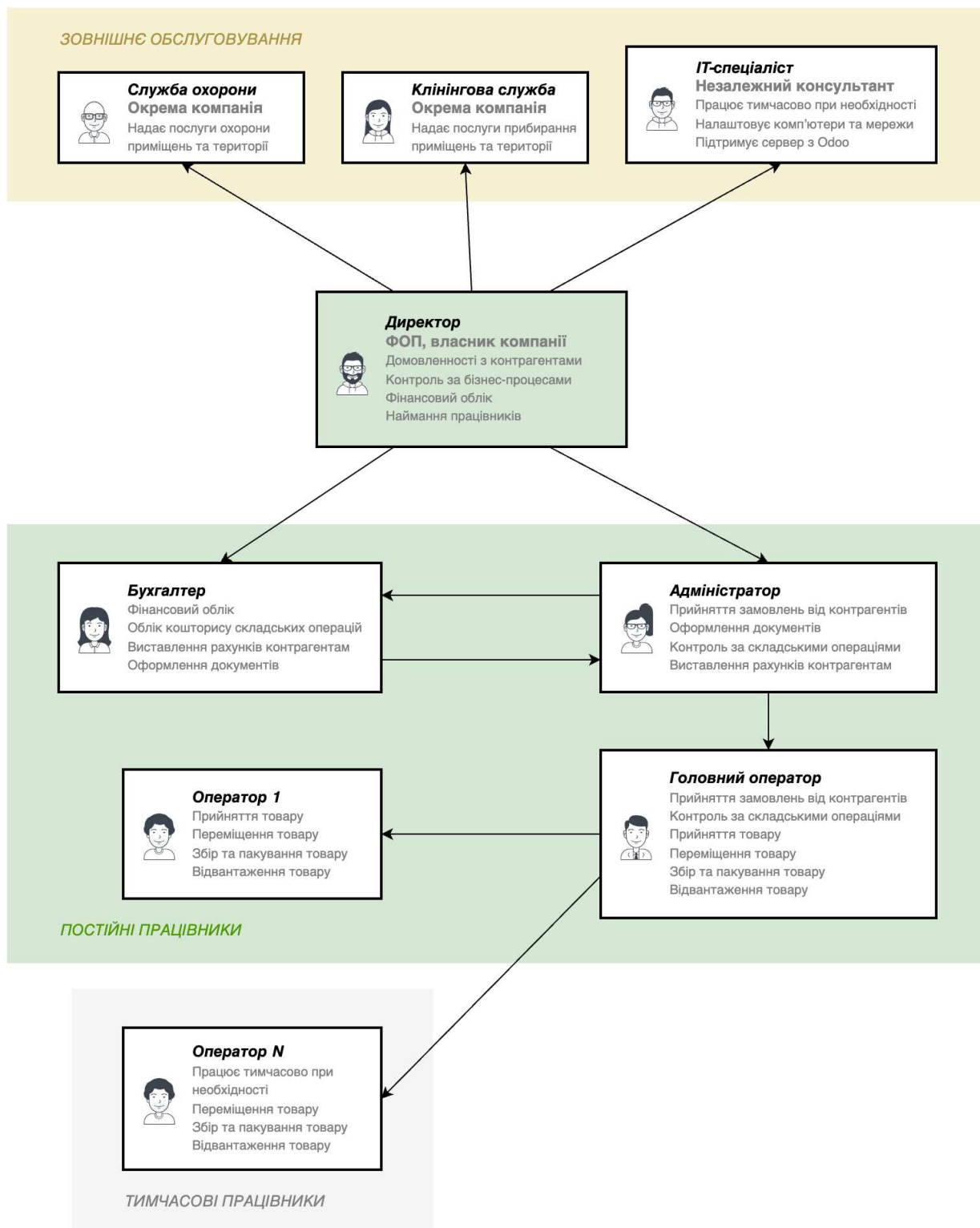


Рисунок 1.3 – Схема організаційної структури управління складом

Функція директора – забезпечення функціонування бізнесу загалом – він приймає рішення щодо прийняття на обслуговування нових контрагентів або відмови від існуючих. Директор укладає договори з контрагентами, погоджує цінову політику щодо послуг зберігання та складських операцій по кожному контрагенту. Контрактує постачальників послуг охорони, прибирання та вивозу бруду, ІТ-послуг. Наймає або контрактує постійних та тимчасових працівників, проводить інструктаж та влаштовує навчальні сесії для працівників.

Бухгалтер веде облік реквізитів контрагентів та кошторису складських операцій, взаємодіє з адміністратором щодо обміну бухгалтерськими документами з контрагентами та сплати за надані послуги. У підприємстві існує дві політики виставлення рахунків:

- одразу після виконання разової операції;
- один раз на місяць по факту всіх виконаних робіт.

Для випадку щомісячної оплати бухгалтер вручну переглядає всі складські операції, які стосуються конкретного контрагента, рахує кількість операцій по видах та вносить до рахунку.

Також бухгалтер має досвід у веденні обліку товарів та складських операцій для часткового заміщення адміністратора у випадку відсутності останнього.

Адміністратор приймає від контрагентів замовлення на завантаження або відвантаження товарів по телефону, електронній пошті або особисто в офісній зоні та друкує замовлення на відповідну складську операцію для подальшого підтвердження представником контрагента. Проводить облік товарів контрагентів, які знаходяться на збереженні у складі та облік і контроль складських операцій – стежить за виконанням замовлень на прийняття, переміщення, підготовку для відвантаження товарів операторами складу, розраховує вартість операцій та зберігання товарів, оформлює та узгоджує документи з контрагентами. Має досвід у первинному бухгалтерському документообігу на випадок необхідності тимчасового заміщення бухгалтера.

Оператори безпосередньо працюють з самим товаром. Отримують замовлення на виконання операцій від адміністратора – завантаження товарів, переміщення по розташуваннях, збір, пакування та відвантаження.

При операції завантаження – приймають товар від контрагента або його представника, вказаного у замовленні, звіряючи прийняту кількість та номенклатуру з тим, що було вказано у замовленні на завантаження, та розкладають товари по розташуваннях складу, які закріплені до контрагента.

При операції переміщення – переміщують вказані товари з одного розташування на інше.

При операції відвантаження – збирають по розташуваннях всі товари згідно замовлення, переміщують у зону пакування, пакують за потребою, та переміщують у зону відвантаження до прибуття кур'єра або служби доставки – будь-якого представника контрагента, вказаного у замовленні на операцію.

На складі завжди працюють як мінімум два оператори, але інколи, при великої запланованої завантаженості по замовленнях, адміністратор або директор залучають ще кількох операторів на підрядну роботу.

Головний оператор має досвід достатній, щоб виконувати частину роботи адміністратора у випадку тимчасової відсутності останнього – прийняття замовлень від контрагентів, ведення обліку товарів та складських операцій, оформлення складських документів. В такому випадку головний оператор також за згодою директора має право долучати додаткових операторів на термін відсутності адміністратора.

Також з ITStore співпрацюють декілька підрядників – служба охорони, клінінгова служба та IT-спеціаліст.

Клінінгова служба займається щоденним прибиранням території, приміщень складу та офісних приміщень після закінчення робочого дня. Раз на тиждень служба забирає весь накоплений мусор – пластик, залишки картонних пакувань та інший бруд.

Служба охорони виконує дві функції – цілодобову фізичну охорону території та приміщень та відео-нагляд за складським приміщенням. Спільно з

ІТ-спеціалістом забезпечують цілодобове функціонування систем відео-нагляду, та мережевий зв'язок охоронних систем з офісом служби охорони.

ІТ-спеціаліст забезпечує функціонування внутрішніх дротової та бездротової мереж для працівників та систем відео-нагляду, інтернет-зв'язку та внутрішнього сервера.

Для ведення обліку товарів та складського документообігу на сервері складу розгорнута багатофункціональна інформаційна система «Odoo», версія спільноти.

Odoo – це інформаційна система з відкритим кодом, яка складається з програм – модулів, які охоплюють всі процеси бізнесу: управління відносинами з клієнтами, торгівлю, виробництво, бухгалтерський облік, управління складом, управління проектами тощо [2]. Odoo має дві версії – для спільноти та для бізнесу. Версія для спільноти – це безкоштовне програмне забезпечення, ліцензоване згідно з GNU LGPLv3 [3]. Версія для бізнесу має власні додаткові модулі та функції. Вихідний код фреймворка та основних модулів планування ресурсів підприємства розроблено бельгійською компанією «Odoo S.A.» [4]

Odoo та її модулі написана на мові Python (версія Odoo 17 потребує Python версії не нижче 3.10) та працює спільно з СУБД PostgreSQL версії не нижче 12 [5].

На сервері складу система розгорнута на базі двох Docker-контейнерів – бази даних PostgreSQL версії 14.4 та веб-сервіс самого Odoo версії 17 [6] [7].

Для працівників складу система доступна через веб-браузер комп'ютера або мобільного телефона лише через дротову або Wi-Fi мережі складу. Доступ до системи з глобальної мережі Інтернет на разі заблокований.

У встановленій на сервері Odoo активовано лише декілька модулів, необхідних для роботи підприємства:

«Контакти» (технічна назва «Contacts»);

«Склад» (технічна назва «Inventory»);

«Виставлення рахунків» (технічна назва «Accounting»);

«Обговорення» (технічна назва «Discussions», встановлюється за замовчуванням).

Модуль «Контакти» відповідає за ведення обліку всіх контрагентів складу – фізичних та юридичних осіб. Для кожного контрагента налаштовуються їх податкові реквізити, фізичні та юридична адреси, банківські рахунки, прайслисти і таке інше.

Модуль «Склад» – основний в рамках цього дослідження – він відповідає за облік товарів та всі складські операції. Має багато налаштувань стосовно керування складом:

- склади та маршрути;
- розташування;
- типи операцій, наприклад надходження, внутрішні;
- категорії товару;
- типи упаковки.

Основні дії доступні в цьому модулі це:

- довідник товарів – створення записів в системі, редагування, налаштування;
- управління замовленнями прийняття товарів на склад.
- управління замовленнями відвантаження замовлених товарів;
- управління замовленнями внутрішніх переміщень товарів між складами та розташуваннями;
- звіти по наявності товарів по різних критеріях.

Модуль «Виставлення рахунків» відповідає за ведення базового бухгалтерського обліку, включаючи:

- виставлення рахунків контрагентам;
- виведення друкованих форм документів;
- можливість надіслати рахунок на електронну адресу контрагента;
- налаштування термінів оплати;
- банківські рахунки підприємства;
- бухгалтерські рахунки;

- звітність.

Модуль «Обговорення» дає можливість користувачам системи:

- обмінюватись повідомленнями;
- додавати підписників-контрагентів до форм документів – рахунків та замовлень на складські операції, та надсилати їм повідомлення.

Для кожного працівника в Odoo заведений користувач із обмеженими правами доступу:

- директор та ІТ-спеціаліст мають повний доступ до всієї системи включно з технічними модулями та глобальними налаштуваннями Odoo;
- бухгалтер та адміністратор мають доступ до модулів «Контакти», «Склад», «Виставлення рахунків» та «Обговорення»;
- оператори мають доступ тільки до модулів «Склад» та «Обговорення».

Провівши аналіз предметної області та отримавши запити від користувачів – працівників складу, знайдено недоліки які знижують загальну ефективність процесів та витрачають час працівників на регулярні однотипні дії.

Модуль «Склад» в Odoo створений для складського обліку з метою самостійної торгівлі, тому має певні недоліки щодо процесу надання послуг зберігання:

- відсутня можливість автоматичного встановлення розташування контрагента при створенні складської операції;
- відсутня можливість вказати представника контрагента при створенні складської операції;
- довідник товарів не має можливості розділення товарів по контрагентах – маємо проблему однакових кодів у різних товарів. Функція вказання приналежності товару існує, але вона призначена для продажу товару контрагента, а не для розподілу товарів по належності до контрагентів [8];
- складність ручного наповнення довідника товарів опрацювання з паперових та навіть електронних носіїв. Odoo має можливість імпорту даних з EXCEL або CSV-файлів, але навіть у цьому випадку процес

займає багато часу на обробку та приведення файлів до потрібного формату [9];

- складність ручного наповнення замовлень складських операцій товарами для опрацювання з паперових та навіть електронних носіїв;
- відсутня можливість для контрагента відправляти замовлення на складські операції з їх інформаційних систем;

Модуль «Виставлення рахунків» має недоліки автоматизації:

- відсутня можливість автоматичного створення рахунку на основі обраної операції;
- відсутня можливість автоматичного створення рахунку на основі всіх операцій вказаного контрагента за період часу.

На підставі проведеного аналізу об'єкту дослідження та виявлених проблем автоматизації всіх вказаних вище процесів маємо сформулювати вимоги для модернізації існуючої системи та створення нової підсистеми інтеграції з контрагентами.

1.2 Формування вимог до системи

На підставі виявлених проблем та запитів модернізована інформаційна система по управлінню складом має складатися з двох підсистем:

- підсистеми управління складом Odoo;
- підсистеми інтеграції з контрагентами.

Визначимо вимоги до підсистеми інтеграції, щоб на їх підставі, та на підставі знайдених проблем автоматизації розробити вимоги до модернізації Odoo.

Підсистема інтеграції має автоматизувати виконання функцій адміністратора та бухгалтера – працівників складу. Має приймати від інформаційних систем контрагентів такі основні запити, та передавати відповідні команди в Odoo:

- запит на прийняття товарів;
- запит на збір та відвантаження товарів;

- запит на виставлення рахунку по вказаній операції;
- запит на виставлення рахунку за вказаний період часу.

Також має по запиту генерувати із Odoo та віддавати інформацію по:

- поточному стану наявності товарів контрагента на складі;
- виставленим рахункам.

Підсистема інтеграції має бути розроблена з використанням сучасних технологій та фреймворків, що забезпечують безпеку та масштабованість програми та, обов'язково, з використання мови програмування Python – тією, якою створена підсистема Odoo.

Схема роботи підсистеми інтеграції має бути надійною – будь-який запит має надійти до Odoo та повернути результат, навіть у разі її недосяжності у поточний час. Обмін даних між усіма кінцевими точками – Odoo, підсистемою інтеграції та інформаційними системами контрагентів має бути безпечним.

На підставі вимог до підсистеми інтеграції, її взаємодії з підсистемою Odoo та знайдених недоліків визначимо вимоги щодо модернізації підсистеми Odoo.

У модуль «Контакти» потрібно ввести додаткові можливості:

- налаштування головного розташування контрагента;
- створення, заміну та видалення ключа доступу контрагента для обміну даними між підсистемою інтеграції та інформаційною системою контрагента.

У модуль «Склад» потрібно ввести додаткові можливості:

- автоматичне встановлення розташування контрагента при створенні замовлення на операцію;
- можливість вказання представника контрагента та його номеру телефона у замовленні на операцію для ідентифікації особи яка отримує або приймає товари;
- розподіл записів у довіднику товарів по належності контрагентам;
- виведення в форму складської операції візуальної інформації про розташування для відвантаження або завантаження;

У модуль «Виставлення рахунків» потрібно ввести додаткові можливості:

- можливість виставлення рахунку по обраним замовленням на складські операції;
- можливість виставлення рахунку по всім операціям обраного контрагента за обраний період часу.

На основі цих вимог розробляється концепція нової інформаційної системи, яка складається з модернізованої підсистеми керування складом та підсистеми інтеграції з контрагентами.

1.3 Розробка концепції інформаційної системи

На основі сформованих вимог розроблено концепцію модернізації поточної системи керування складськими операціями Odoo та концепцію нової підсистеми інтеграції з контрагентами, яка представлена на Рисунку 1.4.

Підсистема інтеграції виконує наступні функціональні задачі:

- авторизація зовнішньої інформаційної системи;
- приймає запити на замовлення операцій завантаження або відвантаження товарів та віддавати унікальний номер замовлення;
- віддає по запиту інформацію про поточний стан замовлення за його унікальним номером;
- приймає запити на скасування замовлення на операцію за його унікальним номером;
- приймає запити на отримання поточного стану наявності товарів на складі;
- приймає запити на виставлення рахунків;
- віддає по запиту інформацію про наявні виставлені рахунки.

Кожен запит від зовнішньої інформаційної системи зберігається у сховищі тимчасових даних для подальшої передачі на обробку в Odoo. Також кожна відповідь від Odoo зберігається у сховищі для подальшої передачі по запиту зовнішньої інформаційної системи.

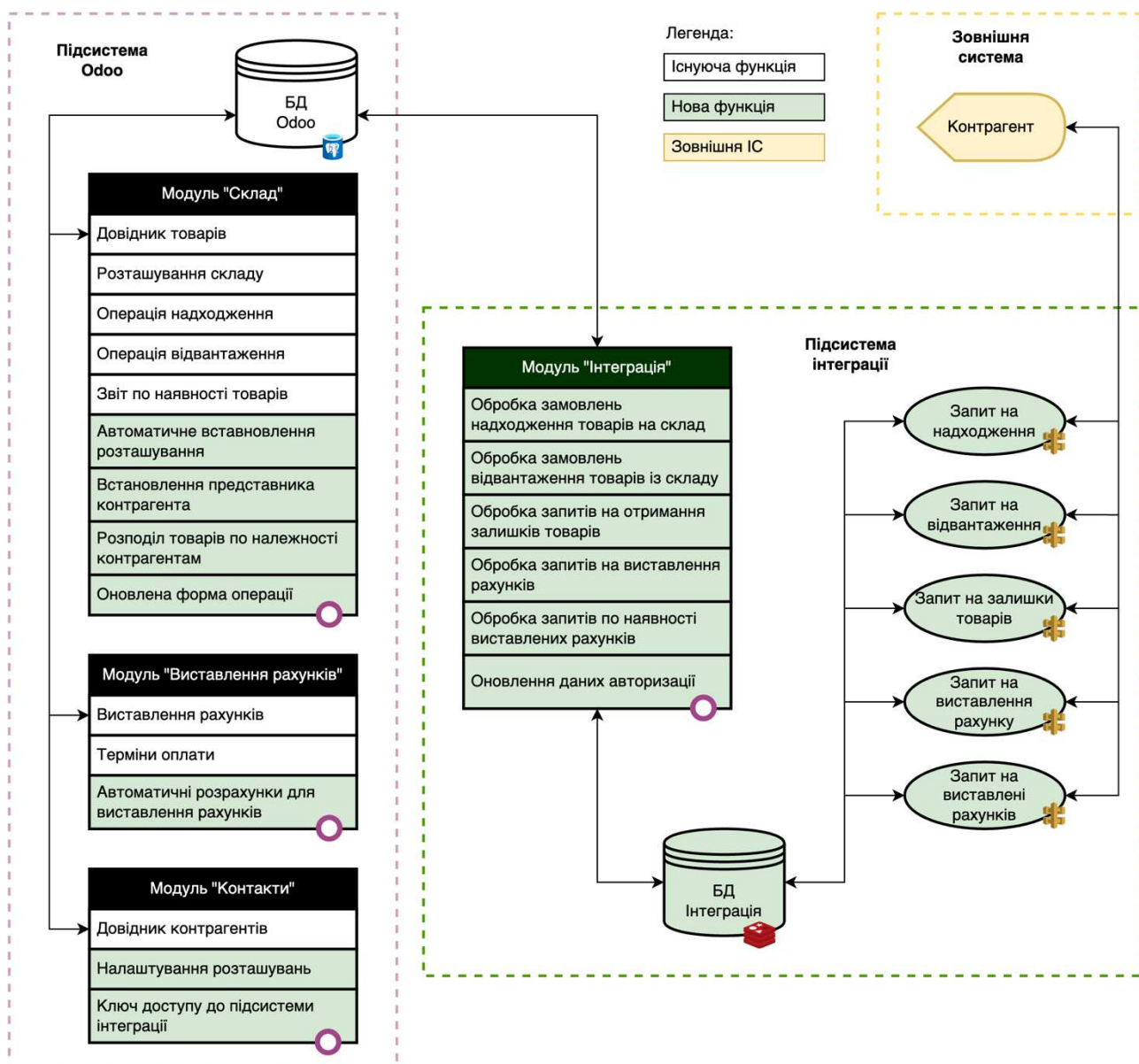


Рисунок 1.4 – Модель концепції інформаційної системи

Для забезпечення взаємодії підсистеми інтеграції з Odoo, в Odoo створюється новий модуль «Інтеграція», а також програмні засоби для виконання завдань з черги [18]. Новий модуль «Інтеграція» виконує наступні функціональні задачі:

- обробка замовлень надходження товарів на склад;
- обробка замовлень відвантаження товарів із складу;
- обробка запитів про наявність товарів на складі;
- обробка запитів на виставлення рахунків;
- обробка запитів про наявність виставлених рахунків;

- оновлення даних для авторизації інформаційних систем контрагентів у сховищі даних «Інтеграція».

Для забезпечення вимог по модернізації підсистеми Odoo створюються нові модулі, які додають в Odoo наступні функціональні задачі:

- налаштування ключів доступу до зовнішніх інформаційних систем;
- безпечний облік різних товарів різних контрагентів;
- автоматизований розрахунок вартості та виставлення рахунків за виконані складські операції за календарний період;
- перехресний зв'язок між розташуваннями та контрагентами;
- підвищення зручності екранних форм складських операцій.

Підсистема інтеграції розробляється відповідно підходу до архітектури мережових протоколів REST з авторизацією запитів JWT згідно стандарту RFC 7519 [11, 12, 13].

Для реалізації архітектури REST обрано FastAPI – швидкий веб-фреймворк для забезпечення асинхронних запитів згідно підходу REST, написаний мовою програмування Python [17].

Для кожного запиту та відповіді для обміну даними між підсистемою інтеграції та зовнішньою інформаційною системою розробляються точки доступу та стандартизовані структури даних у форматі JSON [21].

В якості сховища даних обрано Redis – база зберігання даних у пам'яті. Вона дуже швидка, відмовостійка, має можливість періодичного зберігання даних з оперативної пам'яті на диск та авторизацію запитів [15].

Підсистема інтеграції працює під управлінням операційної системи Linux, дистрибутив Ubuntu версії 22LTS [19].

Для розгортання підсистеми інтеграції необхідний фізичний або віртуальний сервер з відкритими для доступу портами 80 та 443 з глобальної мережі Інтернет, двома обчислювальними ядрами процесора, 4 гігабайтами оперативної пам'яті та мінімум 10 гігабайтами вільного дискового простору. Фізично цей сервер може бути розташований як у приміщенні складу, так і в

обчислювальному центрі обраного постачальника послуг розміщення та оренди серверів.

Для повноцінного та безпечного функціонування підсистеми інтеграції має бути зареєстровано доменне ім'я та випущений довірений сертифікат SSL для забезпечення шифрованого обміну даними між підсистемою інтеграції та зовнішніми системами [20].

Розроблена концепція оновленої інформаційної системи дозволить вирішити існуючі проблеми ефективності та автоматизації процесів складу ITStore, а також вирішить задачу по інтеграції інформаційних систем контрагентів з системою управління складом.

РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ІС

2.1 Загальносистемні рішення

Згідно концепції маємо розподілену систему, підсистеми якої обмінюються даними згідно схеми зображеної на Рисунку 2.1. Обмін даними реалізується за допомогою черги завдань для забезпечення відкладеного виконання запитів, та сховища тимчасових даних – БД «Інтеграція».

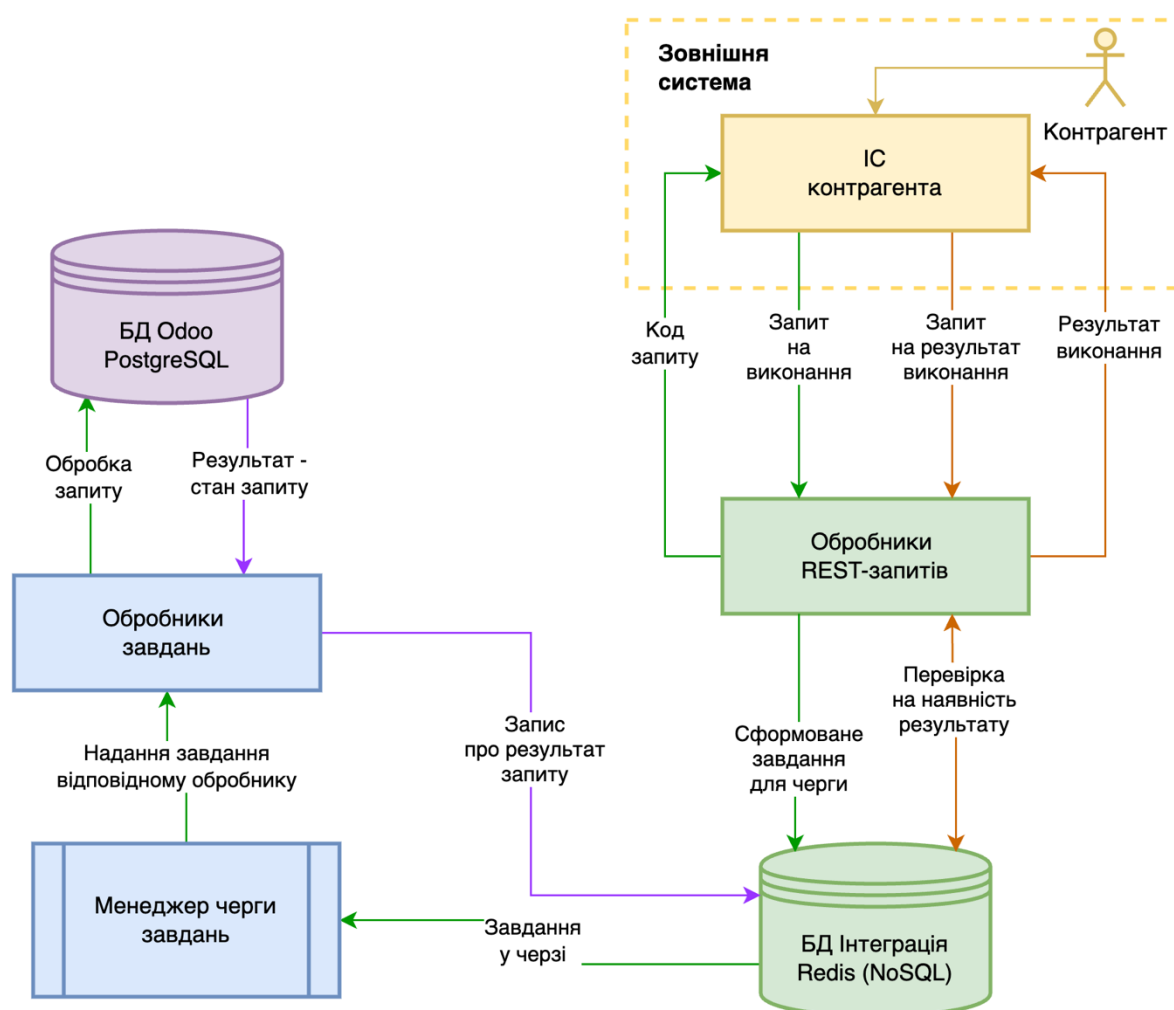


Рисунок 2.1 – Схема обміну даними між підсистемами

Особливість підходу в тому, що зовнішня система не отримує результат одразу, а лише відправляє запит в обробку, отримуючи ідентифікатор завдання по якому зможе з часом отримати результат обробки або стан запиту. Обробник REST-запиту перетворює отримані дані у формат завдання для черги та зберігає

його у сховище даних. Менеджер черги виявляє нове завдання та передає його відповідному обробнику завдань, який виконує всі необхідні операції, спілкуючись з Odoo за допомогою RPC. По закінченню виконання, обробник зберігає результати у сховищі даних. Якщо зовнішня система запитує результат за ідентифікатором, обробник REST-запиту перевіряє наявність результату в сховищі, віддає його зовнішній системі та видаляє із сховища, як вже відпрацьований.

Описаний підхід з використанням черги завдань та тимчасового сховища даних Redis надає доволі високу відмово-стійкість системи щодо збереження даних, отриманих від зовнішніх систем у разі тимчасової відмови підсистеми Odoo або менеджера черг, та збереження результатів виконання у разі тимчасової відмови обробника REST-запитів.

2.2 Рішення щодо інформаційного забезпечення

Згідно розробленої концепції та загальносистемним рішенням у складі інформаційної системи маємо дві бази даних, різних за принципами зберігання даних та призначенням.

Існуюча база даних Odoo працює під управлінням PostgreSQL – об'єктно-реляційній системі керування базами даних, але взаємодія між програмним кодом Odoo та СУБД виконується за допомогою ORM (англ. Object-relational mapping, Об'єктно-реляційна проекція) – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних» [23][24]. Тобто для розширення поточної бази даних, яка вже розгорнута і працює згідно з описом предметної області, відсутня необхідність побудови концептуальної та фізичної моделей, тільки логічної – опис сутностей, які мають бути оновлені або створені.

Обрана для підсистеми інтеграції БД Redis не є реляційною, вона призначена для тимчасового зберігання даних різних видів за принципом ключ-значення – строк, списків та більш складних структур, одна з яких це JSON [25].

Враховуючи те, що між підсистемою інтеграції та системами контрагентів відбувається обмін даними у форматі JSON, дуже вигідно прийняті або готові до відправлення дані зберігати одразу структурованими для відправлення, без необхідності витрачання машинного часу на їх переформатування. До того ж, принцип Redis щодо зберігання даних у оперативній пам'яті забезпечує дуже швидку реакцію на запити даних, а можливість конфігурації для періодичного резервування даних на жорсткий диск та можливість кластеризації БД гарантує збереження даних до моменту їх успішної обробки та відправлення.

Для можливості авторизації зовнішніх інформаційних систем у БД «Інтеграція» мають постійно зберігатися та періодично оновлюватись дані необхідні для авторизації зовнішніх систем та ідентифікації контрагентів для подальшої обробки підсистемою Odoo. Модель «Дані авторизації» показана у вигляді сутності в Таблиці 2.1. Представлення сутності «Дані авторизації» у форматі JSON показано на Рисунку А.1 у Додатку А.

Таблиця 2.1 – Сутність «Дані авторизації»

Зміст поля	Назва поля	Тип	Обов'язкове
Логін	api_login	String	Так
Ключ (пароль)	api_key	String	Так
Контрагент	partner_id	Int	Так

Кожен запит від зовнішньої інформаційної системи та кожна відповідь на запит є електронним документом у форматі JSON, який стандартизується для забезпечення валідності даних та однакової поведінки системи в цілому.

Кожен з цих запитів має включати дані необхідні для подальшого створення відповідних складських документів в підсистемі Odoo:

- замовлення на завантаження;
- замовлення на відвантаження.

Представимо кожний електронний документ запиту у вигляді моделі даних.

Запит на створення замовлення на завантаження товарів на склад показаний у вигляді сутностей в Таблицях 2.2 та 2.3. Представлення запиту «Замовлення на завантаження» у форматі JSON показано на Рисунку А.2 у Додатку А.

Таблиця 2.2 – Сутність «Замовлення на завантаження»

Зміст поля	Назва поля	Тип	Обов'язкове
Номер замовлення	reference	String	Так
Очікувана дата завантаження	inbound_date	Date	Так
Представник	representative_name	String	Ні
Телефон представника	representative_tel	String	Ні
Перелік сутностей «Товар на завантаження»	products	List[Entity]	Так

Таблиця 2.3 – Сутність «Товар на завантаження»

Зміст поля	Назва поля	Тип	Обов'язкове
Артикул	default_code	String	Так
Штрих-код	barcode	String	Ні
Найменування	name	String	Ні
Короткий опис	description	Text	Ні
Кількість	quantity	Integer	Так
Ціна	price	Float	Ні

Запит на створення замовлення на відвантаження товарів із складу має майже таку саму структуру, але не потребує опису товарів, лише їх коди та кількість. Запит показаний у вигляді сутностей в Таблицях 2.4 та 2.5. Представлення запиту «Замовлення на відвантаження» у форматі JSON показано на Рисунку А.3 у Додатку А.

Таблиця 2.4 – Сутність «Замовлення на відвантаження»

Зміст поля	Назва поля	Тип	Обов'язкове
Номер замовлення	reference	String	Так
Очікувана дата відвантаження	outbound_date	Date	Так
Представник (отримувач)	representative_name	String	Ні
Телефон представника	representative_tel	String	Ні

Зміст поля	Назва поля	Тип	Обов'язкове
Пакування	packaging	Boolean	Ні
Перелік сутностей «Товар на відвантаження»	products	List[Entity]	Так

Таблиця 2.5 – Сутність «Товар на відвантаження»

Зміст поля	Назва поля	Тип	Обов'язкове
Артикул	default_code	String	Так
Штрих-код	barcode	String	Ні
Кількість	quantity	Integer	Так

Окремим запитом є запит на виставлення рахунку за послуги складських операцій за обраний календарний період. Цей запит також є джерелом даних для створення документу в підсистемі Odoo – рахунку, але включає лише керуючі поля – календарні дати та додаткове поле. Запит на виставлення рахунку за календарний період показаний у вигляді сутності в Таблиці 2.6. Представлення запиту «Виставлення рахунку» у форматі JSON показано на Рисунку А.4 у Додатку А.

Таблиця 2.6 – Сутність «Запит на виставлення рахунку»

Зміст поля	Назва поля	Тип	Обов'язкове
Початкова дата	date_start	Date	Так
Кінцева дата	date_end	Date	Так
Додати друкований файл у результат	include_invoice_file	Boolean	Ні

Найпростішими запитами є «Запит про залишки» та «Запит про виставлені рахунки» – це електронні документи в яких відсутні будь-які дані, сам факт їх надсилання є підставою для формування завдань на виконання.

Після отримання запитів формуються сутності, описані в Таблицях 2.7 та 2.8 для запису у сховище даних для подальшої обробки чергою завдань згідно документації по менеджеру черг [22].

Таблиця 2.7 – Сутність «Завдання на опрацювання»

Зміст поля	Назва поля	Тип	Обов'язкове
Ідентифікатор завдання	message_id	String	Так
Час поміщення в чергу	message_timestamp	Timestamp	Ні
Назва черги	queue_name	String	Ні
Назва обробника	actor_name	String	Так
Сутність «Аргументи для завдання»	kwargs	Entity	Так

Таблиця 2.8 – Сутність «Аргументи для завдання»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор запиту	uuid	String	Так
Ідентифікатор контрагента	partner_id	Integer	Так
Всі поля із сутності запиту	~	~	Ні

Для кожного запиту, отриманого від зовнішньої інформаційної системи, генерується унікальний ідентифікатор запиту, який записується в поле «uuid». Цей ідентифікатор дозволить у подальшому передати відповідь зовнішній інформаційній системі за її запитом. Поля із сутності запиту відмічені не обов'язковими, тому що у випадку створення завдань на запити «Наявність товарів на складі», «Виставлені рахунки» або на стан відправленого на виконання запиту, дані для обробки відсутні – достатньо лише унікального ідентифікатора запиту та ідентифікатора контрагента.

Представлення сутності «Завдання на опрацювання» у форматі JSON показано на Рисунку А.5 у Додатку А.

Після запису сутності «Завдання на опрацювання» в сховище даних, фактично – поміщення в чергу завдань на обробку, для отриманого запиту у

відповідь формується електронний документ описаний у вигляді сутності у Таблиці 2.9, який відправляється зовнішній інформаційній системі як підтвердження отримання запиту. Представлення сутності «Підтвердження отримання запиту» у форматі JSON показано на Рисунку А.6 у Додатку А.

Таблиця 2.9 – Сутність «Підтвердження отримання запиту»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	uuid	String	Так
Повідомлення	message	String	Так

Для отримання поточного стану складської операції зовнішня інформаційна система надсилає електронний документ «Запит про стан операції», для відміни опрацювання складської операції – електронний документ «Запит на відміну операції», для отримання результатів виконання запитів по наявності товарів або виставлених рахунків – електронний документ «Запит на результат виконання». Усі ці запити мають однакову структуру, описану у Таблиці 2.10.

Таблиця 2.10 – Структура сутностей «Запит про стан операції» та «Запит на результат виконання»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	uuid	String	Так

У відповідь на «Запит про стан операції» після опрацювання підсистемою Odoo, підсистема інтеграції формує електронний документ «Стан складської операції», описаний у вигляді сутності у Таблиці 2.11. Представлення сутності «Стан складської операції» у форматі JSON показано на Рисунку А.7 у Додатку А.

Таблиця 2.11 – Сутність «Стан складської операції»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	uuid	String	Так
Стан операції	state	String	Так

Поле «Стан операції» приймає одне із значень:

- прийнято (accepted);
- створено (draft);
- підтверджено (confirmed);
- прийнято в роботу (assigned);
- завершено (done);
- відмінено (cancel).

У відповідь на «Запит про наявність товарів» після опрацювання підсистемою Odoo, підсистема інтеграції формує електронний документ «Наявність товарів», який вміщує в собі перелік товарів контрагента, кожен з елементів якого описаний у вигляді сутності, як показано у Таблиці 2.12. Представлення сутності «Наявність товарів» у форматі JSON показано на Рисунку А.8 у Додатку А.

Таблиця 2.12 – Одиничний елемент сутності «Наявність товарів»

Зміст поля	Назва поля	Тип	Обов'язкове
Артикул	default_code	String	Так
Штрих-код	barcode	String	Ні
Найменування	name	String	Так
Короткий опис	description	Text	Ні
Кількість	quantity	Integer	Так
Зарезервована кількість	reserved_quantity	Integer	Так
Доступна кількість	available_quantity	Integer	Так

У відповідь на «Запит про виставлені рахунки» після опрацювання підсистемою Odoo, підсистема інтеграції формує електронний документ «Актуальні рахунки», який вміщує в собі перелік рахунків та послуг за які вони були виставлені. Кожен рахунок описаний у вигляді сутностей показаних у

Таблицях 2.13, 2.14 та 2.15. Представлення сутності «Актуальні рахунки» у форматі JSON показано на Рисунку А.9 у Додатку А.

Таблиця 2.13 – Одиничний елемент сутності «Актуальні рахунки»

Зміст поля	Назва поля	Тип	Обов'язкове
Номер рахунку	reference	String	Так
Дата виставлення	issue_date	Date	Так
Термін оплати	deadline_date	Date	Ні
Сутність «Реквізити постачальника»	requisites	Entity	Так
Перелік сутностей «Послуга постачальника»	invoice_lines	List[Entity]	Так
Загальна сума включаючи податки	amount_total	Float	Так
Податки	amount_tax	Float	Ні

Таблиця 2.14 – Сутність «Реквізити постачальника»

Зміст поля	Назва поля	Тип	Обов'язкове
Представник постачальника	signer_name	String	Ні
Посада постачальника	signer_position	String	Ні
Назва компанії	company_name	String	Так
Податковий номер	vat	String	Так
Поштовий індекс	zip	String	Ні
Країна	country	String	Ні
Регіон (область)	region	String	Ні
Місто	city	String	Так
Адреса	street	String	Так
Е-mail	email	String	Ні
Номер телефону	phone	String	Ні
Додаткові реквізити	notes	Text	Ні

Таблиця 2.15 – Сутність «Послуга постачальника»

Зміст поля	Назва поля	Тип	Обов'язкове
Артикул	default_code	String	Так
Найменування послуги	name	String	Так
Кількість	quantity	Int	Так
Ціна	price	Float	Так
Сума	amount	Float	Так
Податки	tax	Float	Ні

Описані вище сутності-запити «Замовлення на завантаження» та «Замовлення на відвантаження» є вхідними документами для подальшого створення записів в базі даних Odoo – в довідник товарів та в складські операції.

Згідно розробленої концепції мають бути модернізовані наступні логічні сутності Odoo:

- складська операція (технічна назва – stock_picking);
- контакт (технічна назва – res_partner);
- товар (технічна назва – product_product).

Скорочений опис сутності «Складська операція» включно з тільки важливими для проектування полями показаний в Таблиці 2.16.

Таблиця 2.16 – Скорочений опис сутності «Складська операція»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	id	Int	Так
Вихідне розташування товарів	location_id	FK	Так
Розташування призначення	location_dest_id	FK	Так
Контрагент	partner_id	FK	Так
Належність товарів контрагенту	owner_id	FK	Ні
Номер	name	String	Авто

Зміст поля	Назва поля	Тип	Обов'язкове
Номер у системі контрагента	origin	String	Hi
Стан	state	String	Авто
Очікувана дата	scheduled_date	Datetime	Так
Дата створення	create_date	Datetime	Так

Згідно до вимог, сутність «Складська операція» має бути модернізована додаванням нових полів, які вказані у Таблиці 2.17.

Таблиця 2.17 – Поля додані до сутності «Складська операція»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор завдання	api_uuid	String	Hi
Представник контрагента	partner_representative_name	String	Hi
Номер телефону представника	partner_representative_tel	String	Hi

Скорочений опис сутності «Контакт» включно з тільки важливими для проектування полями показаний в Таблиці 2.18.

Таблиця 2.18 – Скорочений опис сутності «Контакт»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	id	Int	Так
Ідентифікатор компанія або фізична особа	is_company	Boolean	Так
Ім'я фізичної особи	name	String	Hi
Назва компанії контрагента	company_name	String	Hi
Податковий номер	vat	String	Hi
Категорія (індустрія)	industry_id	FK	Hi
Поштовий індекс	zip	String	Hi
Країна	country_id	FK	Hi

Зміст поля	Назва поля	Тип	Обов'язкове
Регіон (область)	state_id	FK	Hi
Місто	city	String	Hi
Адреса	street	String	Hi
E-mail	email	String	Hi
Номер телефону	phone	String	Hi
Активований	active	Boolean	Так
Дата створення	create_date	Datetime	Так

Згідно до вимог, сутність «Контакт» має бути модернізована додаванням нових полів, які вказані у Таблиці 2.19.

Таблиця 2.19 – Поля додані до сутності «Контакт»

Зміст поля	Назва поля	Тип	Обов'язкове
Логін авторизації запитів підсистеми інтеграції	api_login	String	Hi
Ключ (пароль) авторизації запитів підсистеми інтеграції	api_key	String	Hi
Головне розташування контрагента	main_location_id	FK	Hi

Скорочений опис сутності «Товар» включно з тільки важливими для проектування полями показаний в Таблиці 2.20.

Таблиця 2.20 – Скорочений опис сутності «Товар»

Зміст поля	Назва поля	Тип	Обов'язкове
Унікальний ідентифікатор	id	Int	Так
Артикул	default_code	String	Так
Штрих-код	barcode	String	Hi
Назва	name	String	Так
Опис	description	Text	Hi
Об'єм	volume	Float	Hi
Вага	weight	Float	Hi

Зміст поля	Назва поля	Тип	Обов'язкове
Активний	active	Boolean	Так
Дата створення	create_date	Datetime	Так

Згідно аналізу предметної області існує проблема приналежності записів в довіднику товарів контрагентам, до того ж фізична модель включає в собі унікальний ключ для поля «Штрих-код», що створює проблему одночасного зберігання в довіднику товарів записів з однаковим значенням поля «Штрих-код». Тому для вирішення цих проблем маємо прибрати унікальний ключ з цього поля, додати поле приналежності запису про товар до контрагента – «Власник», та створити новий комбінований ключ базуючись одразу на трьох полях – «Артикул», «Штрих-код» та «Власник». Додаткові поля модернізації сутності «Товар» вказані у Таблиці 2.21.

Таблиця 2.21 – Поля оновлені та додані до сутності «Товар»

Зміст поля	Назва поля	Тип	Обов'язкове	
Артикул	default_code	String	Так	Комбінований унікальний ключ
Штрих-код	barcode	String	Ні	
Власник	owner_id	FK	Так	

Діаграму частини логічної моделі БД Odoo для описаних вище існуючих та оновлених сутностей приведено на Рисунку 2.2.

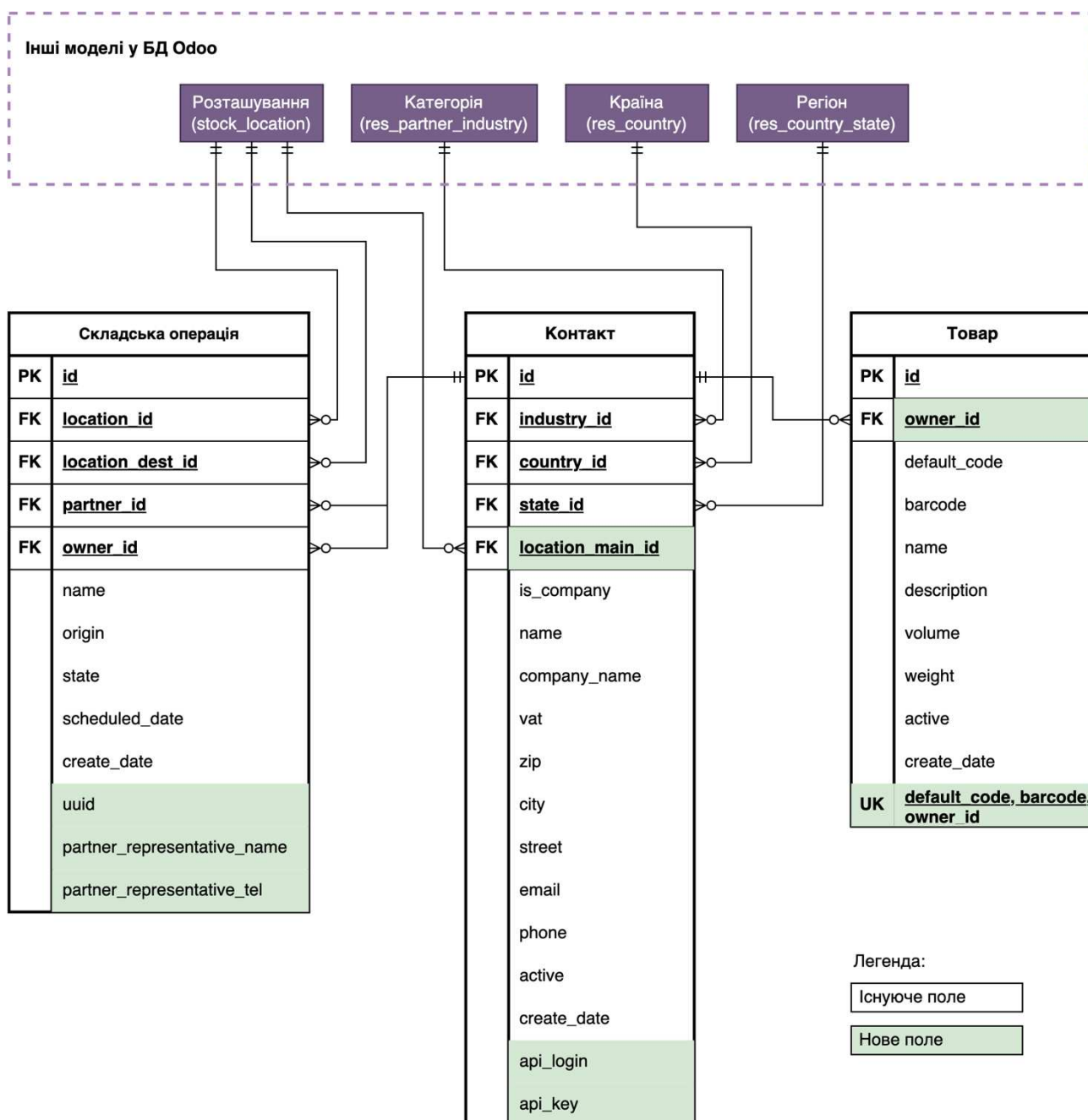


Рисунок 2.2 – Діаграма логічної моделі оновлених сутностей в БД Odoo

2.3 Рішення щодо математичного забезпечення

Згідно концепції підсистеми інтеграції кожен запит від зовнішньої системи має бути авторизованим – по схемі веб-токенів JSON (JWT) описаної в RFC 7519 [13]. Це відкритий стандарт, який визначає компактний і автономний спосіб безпечної передачі інформації між сторонами у вигляді об'єкта JSON. Веб-токени JSON - це спосіб безпечної передачі інформації між сторонами. Оскільки JWT

можна підписувати - наприклад, за допомогою секретного ключа - можна бути впевненими, що відправники є тими, за кого себе видають. Крім того, оскільки підпис обчислюється за допомогою заголовка і корисного навантаження, можна перевірити, що вміст не був підроблений [12].

Для виконання будь-якого запиту до підсистеми інтеграції зовнішня система запрошує веб-токен, передаючи при цьому системі інтеграції дані для авторизації – логін та ключ.

Приймаючи ці дані, підсистема інтеграції порівнює їх з даними авторизації, які зберігаються у БД «Інтеграція», та, якщо вони вірні, формує веб-токен та відправляє його зовнішній інформаційній системі у відповідь.

Веб-токен має обмежений час дії, встановлений у 5 хвилин – після закінчення терміну дії, він більше не може бути використаний для запитів та має бути поновлений.

Кожен веб-токен складається з трьох частин:

- заголовок;
- корисне навантаження;
- підпис.

Заголовок та корисне навантаження – це звичайні JSON-документи – у заголовку вказуються тип та алгоритм підписання токена, у корисному навантаженні – дані, на основі яких і будується токен. Структура заголовка в форматі JSON показана на Рисунок 2.4.



Рисунок 2.4 – Структура заголовка токена

Заголовок має лише два параметри – тип токена, який завжди має значення “JWT”, та алгоритм, який використовується для підписання токена, в нашому випадку – HMAC SHA256 – код автентифікації повідомлень на основі хешування з ключем [27].

Корисне навантаження складається з трьох параметрів – логіну зовнішньої інформаційної системи, часу створення токена та часу закінчення строку дії токена. Обидві часові мітки мають формат «Час Unix» – кількість секунд, які минули від початку епохи Unix – з першого січня, 1970-того року. Структура корисного навантаження в форматі JSON показана на Рисунку 2.5.

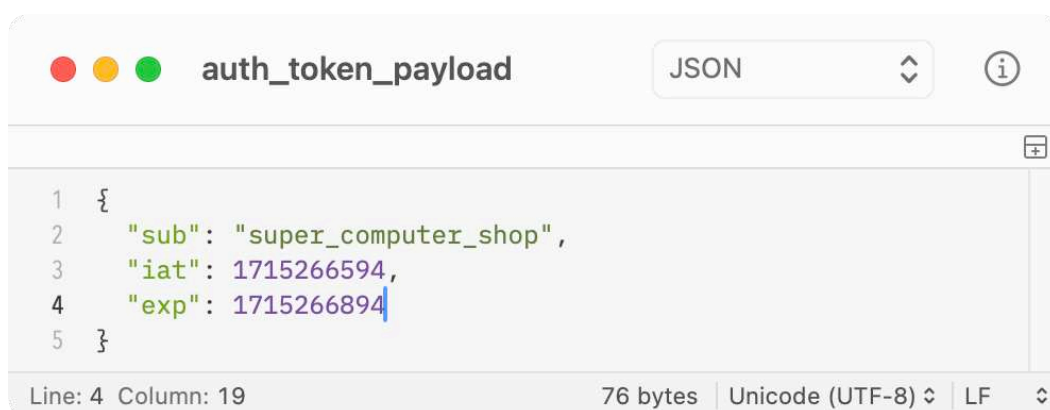


Рисунок 2.5 – Структура корисного навантаження

JSON-об’єкти заголовку та корисного навантаження перетворюються у строки за алгоритмом Base64URL [26].

Далі згідно алгоритмом JWT отримані строки мають бути об’єднані в одну через крапку. Отриманий результат для вказаних вище JSON-об’єктів заголовка та корисного навантаження показаний на Рисунку 2.6.



Рисунок 2.6 – Закодовані заголовок та корисне навантаження

Наступним етапом створення токена є його підписання – створення криптографічного підпису на основі заголовка та корисного навантаження. Підпис створюється за формулою:

$$\text{Sign} = \text{HMACSHA256}(\text{base64UrlEncode}(\text{header}) + "." + \text{base64UrlEncode}(\text{payload}), \text{secret})$$

Де:

- header – заголовок у форматі JSON, закодований у строку алгоритмом Base64;
- payload – корисне навантаження у форматі JSON, закодоване у строку алгоритмом Base64;
- secret – секретний ключ, рядок з набору символів, який відомий тільки системі інтеграції та призначений для підвищення безпеки підпису;
- HMACSHA256 – функція створення хеш-коду на основі переданих у функцію даних, вказаних вище.

Після отримання підпису, він через крапку додається до попереднього результату – скомбінованих заголовка та корисного навантаження – і це дає фінальний результат – веб-токен, повний вигляд якого показаний на Рисунку 2.7.



Рисунок 2.7 – Сформований токен

На час життя отриманого токена зовнішня система може використовувати його в заголовку REST-запиту для підтвердження того, що запит є авторизованим.

2.4 Рішення щодо програмного забезпечення

Згідно розробленої концепції оновлена інформаційна система складається із декількох компонентів:

- обробники REST-запитів;
- база тимчасових даних – Redis;
- менеджер черг завдань;
- обробники завдань;
- моніторинг черг;
- ПЗ Odoo (існує, потребує модернізації);
- база даних Odoo – PostgreSQL.

На підставі того, що основний компонент системи – Odoo, розроблений на мові програмування Python, згідно вимог до проекту, всі інші нові компоненти – обробники REST-запитів, менеджер черг та обробники завдань, також мають бути розроблені тією самою мовою програмування.

З точки зору програмної архітектури описані вище компоненти зібрані у три окремих програмних застосунки, які схематично зображені на Рисунку 2.8:

- «Інтеграція» – обробники REST-запитів на базі FastAPI та база тимчасових даних Redis, технічна назва – `3pl_api`;
- «Черга» – менеджер черг Dramatiq, обробники завдань та моніторинг черг, технічна назва – `3pl_queue`;
- «Склад» – Odoo та його СУБД PostgreSQL, технічна назва `3pl_stock`.

Для створення всіх необхідних обробників REST-запитів обрано веб-фреймворк FastAPI – швидкий в розробці та гнучкий фреймворк, написаний мовою програмування Python. В якості зовнішнього веб-сервера, який відповідає за прийом REST-запитів з шифруванням даних за допомогою SSL-сертифікатів, використовується NGINX [31].

Для реалізації підходу обміну даними через черги завдань обрано менеджер черг Dramatiq – легкий та функціональний, написаний на мові програмування Python, маючий можливість створення завдань різними способами та сумісний з обраною базою даних Redis [16].

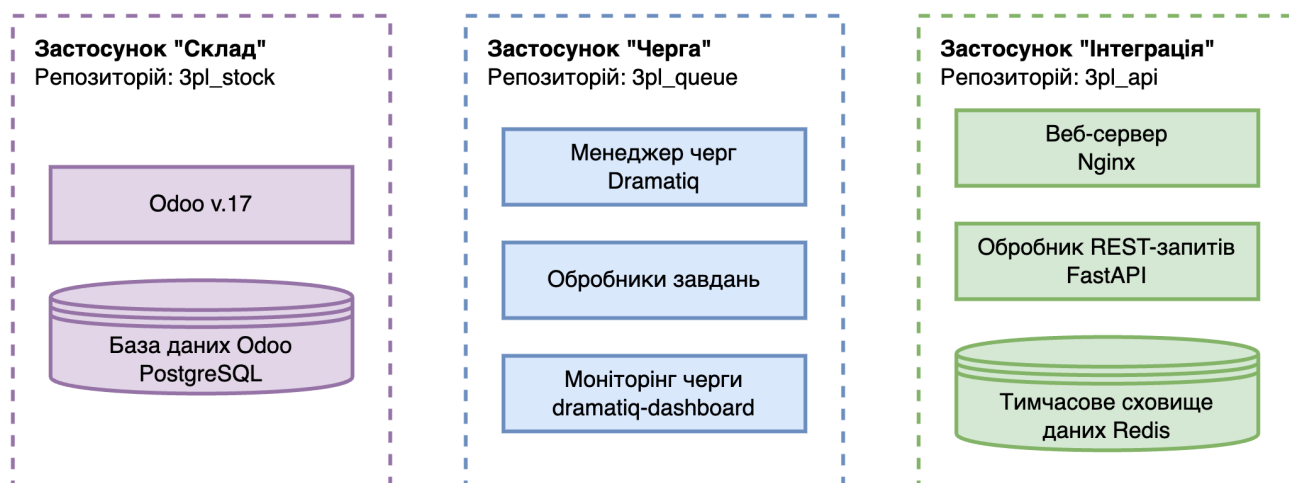


Рисунок 2.8 – Компоненти застосунків

Обробники завдань черги розробляються як окремі Python-програми, згідно документації до менеджера черг Dramatiq.

Для розробки та тестування додатків да їх функціональних можливостей використовується інтегроване середовище розробки «PyCharm» [32].

2.4.1 Рішення щодо застосунку «Інтеграція»

Для створення та функціонування застосунку розгортається віртуальне Python-середовище, що забезпечує встановлення тільки тих модулів Python, які необхідні для конкретного додатку, згідно документації по Python [30]. Перелік необхідних модулів та їх версії показано на Рисунку Б.1 у Додатку Б.

Згідно рішення щодо інформаційного забезпечення має бути реалізоване підключення до БД Redis для доступу до даних необхідних для автентифікації зовнішніх систем та авторизація запитів згідно алгоритму JWT, описаному в розділі 2.3. Частини кодів підключення до бази даних та запиту до неї щодо даних авторизації показано на Рисунках Б.2 та Б.3 у Додатку Б.

Для кожного запиту, описаного в концепції системи, має бути створена «кінцева точка» – веб-адреса та одночасно функція доступу до даних або прийняття даних від зовнішньої системи. Згідно описаних в розділі 2.2 запитів, сформовані кінцеві точки, описані в Таблиці 2.22.

Таблиця 2.22 – Кінцеві точки REST-запитів

Назва функції	Відносна веб-адреса	Опис
auth_token	/auth/token	Запит на отримання токена авторизації
create_inbound	/transfer/inbound	Створення замовлення на завантаження
create_outbound	/transfer/outbound	Створення замовлення на відвантаження
transfer_state	/transfer/state	Запит на стан замовлення
issue_invoice	/invoice/issue	Запит на виставлення рахунків за вказаний період
actual_invoices	/invoice/actual	Запит на створення звіту по актуальних виставлених рахунках
invoice_state	/invoice/state	Стан запиту по рахункам
build_stock	/stock/build	Запит на створення звіту по залишках на складі
actual_stock	/stock/state	Запит на отримання результатів звіту по залишках

Всі вказані вище функції розроблені та проведено тестування за допомогою модуля «HTTP Client» у середовищі розробки PyCharm. Результати тестування функції «Запит на отримання токена авторизації» та частина коду розробленого тесту показано на Рисунках Б.4 та Б.5 у Додатку Б.

REST-фреймворк FastAPI перевіряє кожен запит від зовнішньої системи на відповідність типів даних та JSON-структури. Для цього кожен очікуваний запит до кінцевої точки описується як Python-клас з вказанням типів кожного атрибуту класу. Код класів на прикладі запиту на створення замовлення на завантаження, включно з окремою функцією перевірки формату та відповідності дати, вказано на Рисунку Б.6 у Додатку Б.

Результати тестування перевірки запиту на відповідність на прикладі запиту на створення замовлення на завантаження та код тесту показані на Рисунках Б.7 та Б.8 у Додатку Б.

Отримані дані структуруються та зберігаються у тимчасовому сховище Redis у вигляді завдання на виконання для менеджера черг Dramatiq. Код для поміщення завдання в чергу показаний на Рисунку Б.9 у Додатку Б.

2.4.2 Рішення щодо застосунку «Черга»

Для створення та функціонування застосунку розгортається віртуальне Python-середовище, перелік необхідних модулів та їх версії показано на Рисунку Б.10 у Додатку Б.

Для обробки запитів розробляються невеличкі функції – обробники завдань черги, кожен з яких виконує лише функцію шлюзу між застосунками «Інтеграція» та «Склад» – він лише приймає дані з черги, віддає Odoo на обробку, очікує та приймає результат виконання та повертає його в тимчасове сховище даних Redis для подальшого опрацювання на стороні застосунку «Інтеграція» – передачі зовнішній інформаційній системі по її запиту. Згідно документації по менеджеру черг функція має таку саму назву, як вказано в полі «actor_name» при створенні задачі на виконання (см. Рисунок А.5 у Додатку А).

Згідно концепції розроблені функції – обробників запитів, кожна з яких приймає параметри відповідно запитів. У Таблиці 2.23 вказано перелік функцій обробників, назва черги та їх призначення.

Таблиця 2.23 – Перелік функцій-обробників завдань черг

Назва функції	Назва черги	Опис
create_inbound_order	inbound	Передає в Odoo на виконання запит на створення замовлення на завантаження товарів, повертає в Redis стан запиту.
create_outbound_order	outbound	Передає в Odoo на виконання запит на створення замовлення на відвантаження товарів, повертає в Redis стан запиту.
create_invoice	invoicing	Передає в Odoo на виконання запит на виставлення рахунку за вказаний період, повертає в Redis стан запиту.

Назва функції	Назва черги	Опис
actual_invoices	invoicing	Передає в Odoo на виконання запит на побудову звіту про актуальні виставлені рахунки, повертає в Redis результат виконання.
build_stock_report	stock	Передає в Odoo на виконання запит на побудову звіту про наявність товарів на складі, повертає в Redis результат виконання.
update_credentials	default	Оновлює дані необхідні для автентифікації зовнішніх інформаційних систем

Дані, які приймають ці функції, та результати, які вони повертають у БД Redis повністю відповідають інформаційним моделям описаним в розділі 2.2. Передача в Odoo на виконання виконується через протокол RPC за допомогою модуля Python «odoo-rpc-client». Код функції на прикладі «create_inbound_order» показаний на Рисунку Б.11 у Додатку Б.

Для візуального нагляду за роботою обробників встановлюється модуль Python «dramatiq-dashboard», та запускається згідно його документації [33]. Зовнішній вигляд вікна моніторингу та окремої задачі показані на Рисунках Б.12 та Б.13 у Додатку Б.

Тестування функцій відбувається шляхом одночасного запуску застосунків «Черга» та «Інтеграція» та виконання автоматичних тестів, описаних в розділі 2.4.1. Журнал виконання завдання обробником черги – функцією «create_inbound_order» – показано на Рисунку Б.14 у Додатку Б.

2.4.3 Рішення щодо застосунку «Склад»

Для створення та функціонування застосунку розгортається віртуальне середовище з використанням Docker – інструментарію для управління ізольованими Linux-контейнерами [6]. На базі Docker конфігуруються два сервіси:

- Для самого ПЗ Odoo на основі офіційного Docker-контейнеру версії 17;
- Для СУБД PostgreSQL на основі офіційного Docker-контейнеру версії 14.

Згідно документації по Odoo створюється новий модуль, який має включати в собі всі необхідні оновлення до підсистеми. Модуль описується маніфестом, код якого показаний на Рисунку Б.15 у Додатку Б.

Згідно проекту рішень щодо інформаційного забезпечення та документації по Odoo ORM оновлюються логічні моделі даних в описаних сутностях «Складська операція», «Контакт» та «Товар». Частини кодів на прикладах моделей «Складська операція» та «Контакт» показані на Рисунках Б.16 та Б.17 у Додатку Б.

Для виконання запитів із обробників черги повідомлень, в Odoo реалізуються функції відповідні до функцій обробників, вказаних у Таблиці 2.23. Код функції Odoo на прикладі створення замовлення на завантаження товарів на склад показаний на Рисунку Б.18 у Додатку Б.

Згідно вимог та концепції оновлюються екранні форми інтерфейсу користувача Odoo. Коди оновлення екранних форм на сутностей «Складська операція», «Контакт» та «Товар» показано на Рисунках Б.19-21 у Додатку Б.

Згідно концепції та рішень щодо інформаційного забезпечення автентифікація зовнішніх систем має бути миттєвою, але дані контрагентів зберігаються в БД Odoo, а обробники REST-запитів застосунку «Інтеграція» не мають до неї прямого доступу, тому сховище тимчасових даних Redis має вмішувати дані, необхідні для автентифікації. Для оновлення даних авторизації також буде використаний підхід з використанням черги завдань. Для цього буде використано програму Cron з стандартного пакету ОС Linux – створення завдання на оновлення даних авторизації в БД «Інтеграція» кожні чотири години. Схему оновлення даних авторизації показано на Рисунку 2.9.

Для реалізації цієї схеми на стороні Odoo створюється функція «list_api_credentials», яка збирає із БД Odoo необхідні для авторизації зовнішніх інформаційних систем дані за структурою описаною в розділі 2.2 та повертає їх обробнику завдання для подальшого оновлення даних авторизації у БД Redis.



Рисунок 2.9 – Схема оновлення даних авторизації зовнішніх систем

Всі програмні рішення реалізовані та протестовані як автоматичними тестами – для кінцевих точок REST-запитів та обробників завдань черги, так і вручну – для екранних форм в Odoo. Результати роботи всіх компонентів та їх функцій відповідають вимогам проекту.

2.5 Рішення щодо технічного забезпечення

Підсистема Odoo та її база даних PostgreSQL вже розгорнуті у приміщенні складу на сервері під управлінням операційної системи Linux Ubuntu версії 22. Сервер має 4-ядерний процесор та 16 ГБ оперативної пам'яті, що достатньо для Odoo, її бази даних та для роботи обробників завдань черги Dramatiq. Також рекомендовано встановлення системи безперебійного живлення сервера та мережевих комунікаторів на випадок знеструмлення приміщення на короткий термін – до 5 хвилин.

Для розгортання підсистеми інтеграції рекомендовано хмарну платформу «Amazon Web Services» та її сервіс оренди віртуальних серверів «EC2» (Elastic Compute Cloud). Рекомендований EC2 сервер типу M4.LARGE – він забезпечує баланс пам'яті, обчислювальних та мережевих ресурсів і є достатнім вибором для підсистеми інтеграції. Сервер M4.LARGE включає в собі два ядра

віртуального процесора на базі Intel Xeon Scalable Processor, 8 ГБ оперативної пам'яті та стандартні 30 ГБ дискового простору [28].

В якості альтернативи рекомендований хмарний сервіс HETZNER та сервер типу CX21. Сервер має два хмарних обчислювальних ядра на базі Intel Xeon Gold або AMD EPYC 7002, 4 ГБ оперативної пам'яті та 40 ГБ дискового простору [29]. На відмінність від платформи AWS платформа HETZNER має простіші можливості налаштування. Також всі сервери сервісу HETZNER на відміну від AWS EC2 одразу включають публічну IPv4 та IPv6 адреси.

У Таблиці 2.22 наведено порівняльні дані по параметрах та приблизних цінах за кожен рік користування рекомендованими серверами.

Таблиця 2.22 – параметри та приблизна вартість оренди віртуальних серверів

Тип серверу	Сервіс	Кількість ядер	Оперативна пам'ять, ГБ	Дисковий простір, ГБ	Орієнтовна ціна, грн./рік
M4.LARGE	AWS EC2	2	8	30	2 931.6
CX21	HETZNER	2	4	40	3 033.6

Також для функціонування застосунку «Інтеграція» реєструється доменне ім'я у будь якого українського реєстратора. Доменне ім'я налаштовується на IP-адресу отриману після придбання обраного серверу. Орієнтовна вартість доменного імені в зоні MK.UA у національного реєстратора доменних імен «Imena.UA» строком на 1 рік – 444 грн [34].

Розробка технічного проекту інформаційної системи виконана у повному обсязі. Визначені загальносистемні рішення, рішення щодо інформаційного, математичного, програмного та технічного забезпечення системи. Виконані та описані розробка необхідних застосунків та їх тестування.

На підставі технічного проекту всі складові інформаційної системи можуть бути встановлені та введені в експлуатацію.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ ІС

Для успішного впровадження інформаційної системи, для кожного із її застосунків, розроблені інструкції для системних адміністраторів щодо розгортання та підтримки, та інструкції користувачів підсистеми Odoo щодо оновлених функцій. Загальні інструкції користувача Odoo наведені на офіційному веб-сайті Odoo [35].

Інструкцію по розгортанню та налаштуванню застосунку «Інтеграція», порядок доступу до документації по REST-запитам, а також посилання на репозиторій з кодом програм, наведено у Додатку В.

Інструкцію по розгортанню та налаштуванню застосунку «Черга» та посилання на репозиторій з кодом програм, наведено у Додатку Г.

Інструкцію користувача Odoo щодо оновлень в рамках застосунку «Склад», а також посилання на репозиторій з кодом нового модуля Odoo, наведено у Додатку Д.

Проведено опробування кожного з застосунків та інформаційної системи в цілому, шляхом розгортання всіх компонентів та виконання модульного тестування застосунку «Інтеграція» та ручного тестування оновленої підсистеми Odoo.

Опробування всіх компонентів інформаційної системи та схеми обміну даними проміж ними показало позитивні результати та високу відмово-стійкість інформаційної системи в цілому.

Реалізацію проекту виконано в повному обсязі з врахуванням вимог бізнесу, згідно розробленої концепції та рішень. Проект розроблено з використанням сучасних ідей щодо архітектури програмного забезпечення та з використанням найсучасніших версій допоміжного програмного забезпечення.

HTTP-запит аутентифікації та отриманий у відповідь токен авторизації показані на Рисунках 3.1 та 3.2.

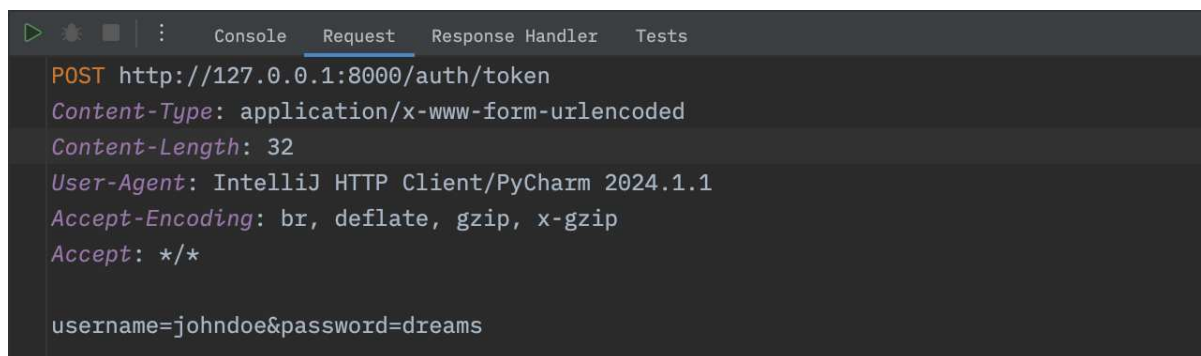


Рисунок 3.1 – Запит на отримання токenu

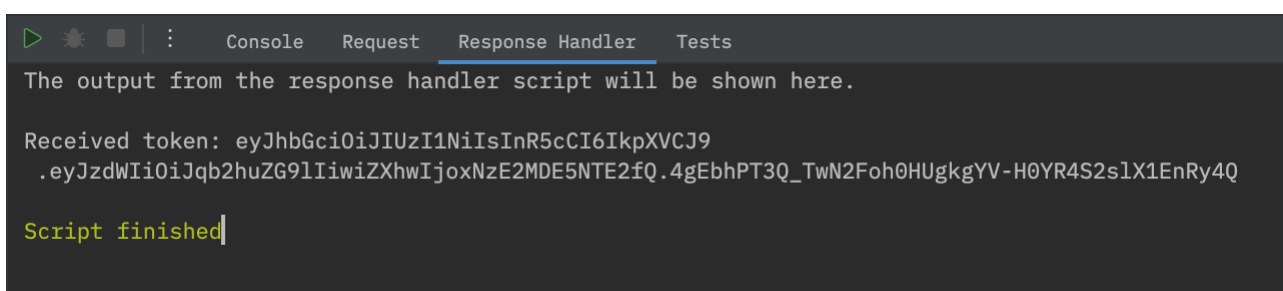


Рисунок 3.2 – Токен авторизації відправлений у відповідь

HTTP-запит на створення замовлення завантаження товарів на склад, відповідь з підтвердженням про прийняття запиту в обробку та журнал менеджера черги щодо виконаної обробки показані на Рисунках 3.3, 3.4 та 3.5.

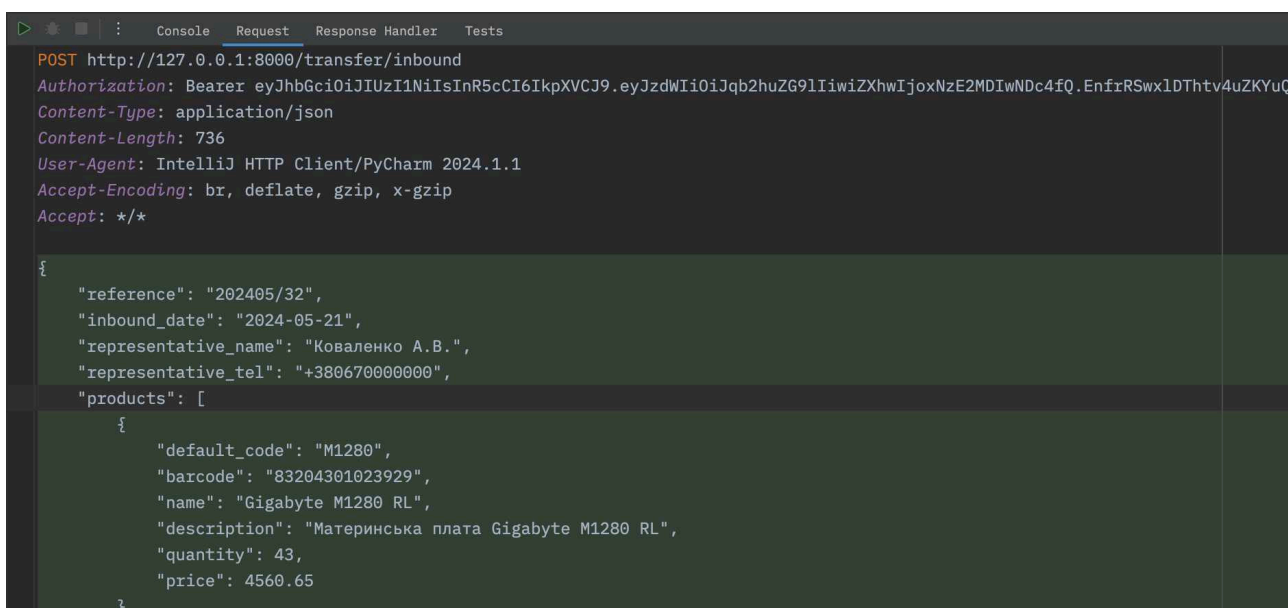
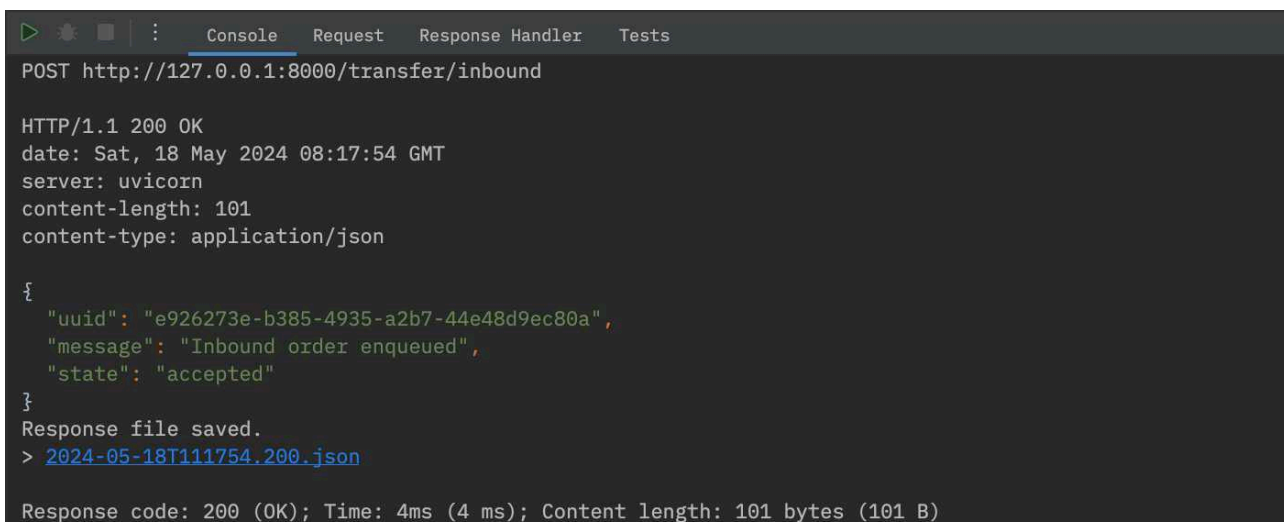


Рисунок 3.3 – Запит на створення замовлення на завантаження



```

POST http://127.0.0.1:8000/transfer/inbound

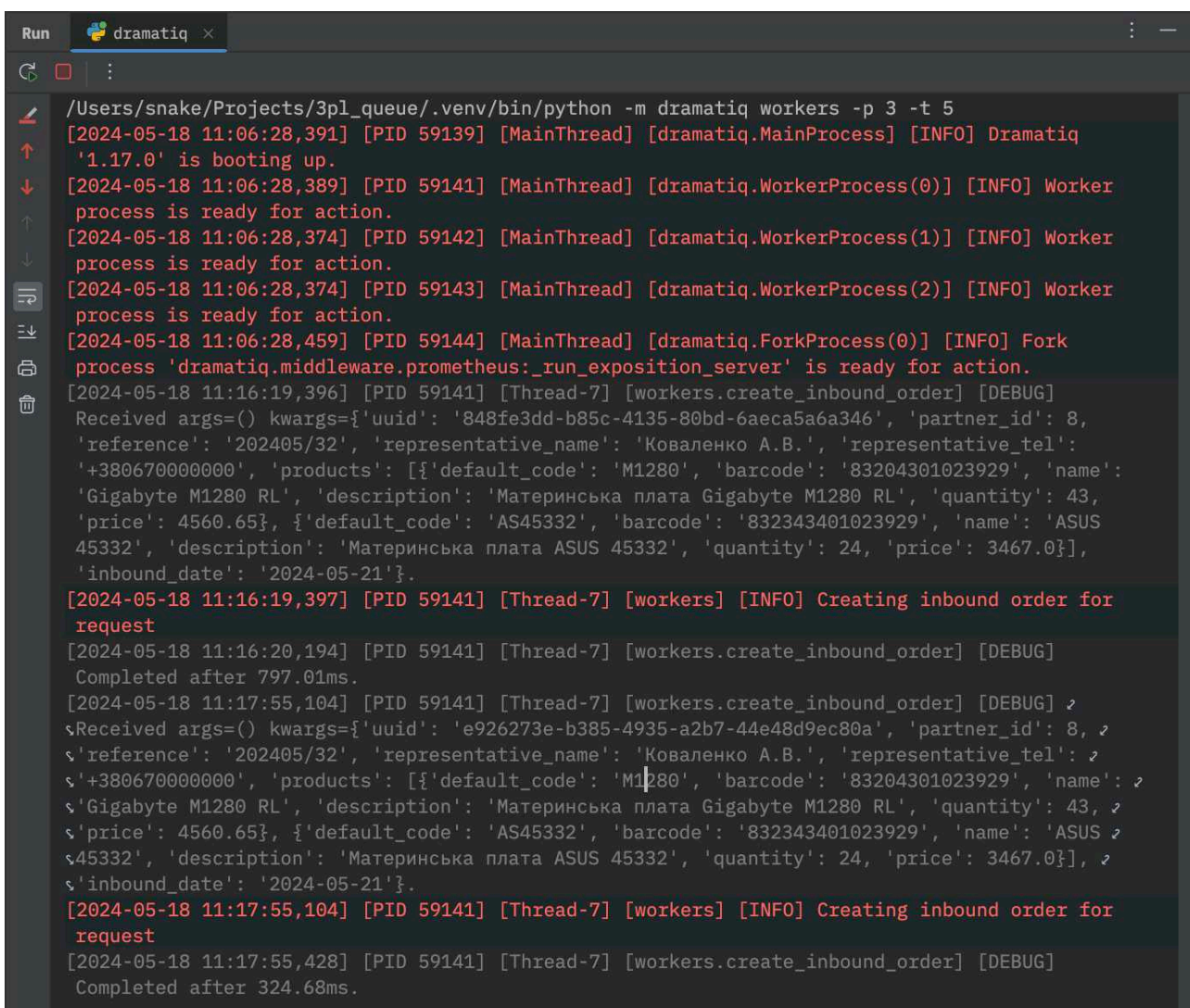
HTTP/1.1 200 OK
date: Sat, 18 May 2024 08:17:54 GMT
server: uvicorn
content-length: 101
content-type: application/json

{
  "uuid": "e926273e-b385-4935-a2b7-44e48d9ec80a",
  "message": "Inbound order enqueued",
  "state": "accepted"
}
Response file saved.
> 2024-05-18T111754.200.json

Response code: 200 (OK); Time: 4ms (4 ms); Content length: 101 bytes (101 B)

```

Рисунок 3.4 – відповідь з підтвердженням про взяття запиту на обробку



```

Run dramatiq x
/Users/snake/Projects/3pl_queue/.venv/bin/python -m dramatiq workers -p 3 -t 5
[2024-05-18 11:06:28,391] [PID 59139] [MainThread] [dramatiq.MainProcess] [INFO] Dramatiq
'1.17.0' is booting up.
[2024-05-18 11:06:28,389] [PID 59141] [MainThread] [dramatiq.WorkerProcess(0)] [INFO] Worker
process is ready for action.
[2024-05-18 11:06:28,374] [PID 59142] [MainThread] [dramatiq.WorkerProcess(1)] [INFO] Worker
process is ready for action.
[2024-05-18 11:06:28,374] [PID 59143] [MainThread] [dramatiq.WorkerProcess(2)] [INFO] Worker
process is ready for action.
[2024-05-18 11:06:28,459] [PID 59144] [MainThread] [dramatiq.ForkProcess(0)] [INFO] Fork
process 'dramatiq.middleware.prometheus:run_exposition_server' is ready for action.
[2024-05-18 11:16:19,396] [PID 59141] [Thread-7] [workers.create_inbound_order] [DEBUG]
Received args=() kwargs={'uuid': '848fe3dd-b85c-4135-80bd-6aeca5a6a346', 'partner_id': 8,
'reference': '202405/32', 'representative_name': 'Коваленко А.В.', 'representative_tel':
'+380670000000', 'products': [{'default_code': 'M1280', 'barcode': '83204301023929', 'name':
'Gigabyte M1280 RL', 'description': 'Материнська плата Gigabyte M1280 RL', 'quantity': 43,
'price': 4560.65}, {'default_code': 'AS45332', 'barcode': '832343401023929', 'name': 'ASUS
45332', 'description': 'Материнська плата ASUS 45332', 'quantity': 24, 'price': 3467.0}],
'inbound_date': '2024-05-21'}.
[2024-05-18 11:16:19,397] [PID 59141] [Thread-7] [workers] [INFO] Creating inbound order for
request
[2024-05-18 11:16:20,194] [PID 59141] [Thread-7] [workers.create_inbound_order] [DEBUG]
Completed after 797.01ms.
[2024-05-18 11:17:55,104] [PID 59141] [Thread-7] [workers.create_inbound_order] [DEBUG]
Received args=() kwargs={'uuid': 'e926273e-b385-4935-a2b7-44e48d9ec80a', 'partner_id': 8,
'reference': '202405/32', 'representative_name': 'Коваленко А.В.', 'representative_tel':
'+380670000000', 'products': [{'default_code': 'M1280', 'barcode': '83204301023929', 'name':
'Gigabyte M1280 RL', 'description': 'Материнська плата Gigabyte M1280 RL', 'quantity': 43,
'price': 4560.65}, {'default_code': 'AS45332', 'barcode': '832343401023929', 'name': 'ASUS
45332', 'description': 'Материнська плата ASUS 45332', 'quantity': 24, 'price': 3467.0}],
'inbound_date': '2024-05-21'}.
[2024-05-18 11:17:55,104] [PID 59141] [Thread-7] [workers] [INFO] Creating inbound order for
request
[2024-05-18 11:17:55,428] [PID 59141] [Thread-7] [workers.create_inbound_order] [DEBUG]
Completed after 324.68ms.

```

Рисунок 3.5 – Виконання обробником черги функції «create_inbound_order»

Успішно створені в підсистемі Odoo товари, передані через запит та нове замовлення на завантаження цих товарів, включно з оновленими полями «Представник контрагента», «Контактні дані представника» та «Унікальний ідентифікатор завдання» показані на Рисунках 3.6 та 3.7.

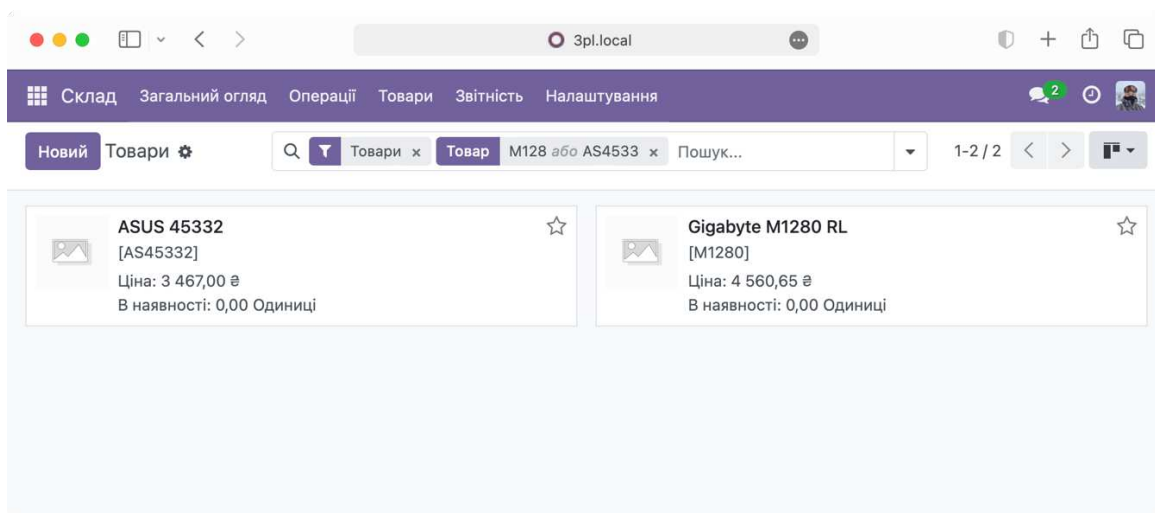


Рисунок 3.6 – Нові товари створені внаслідок обробки запиту

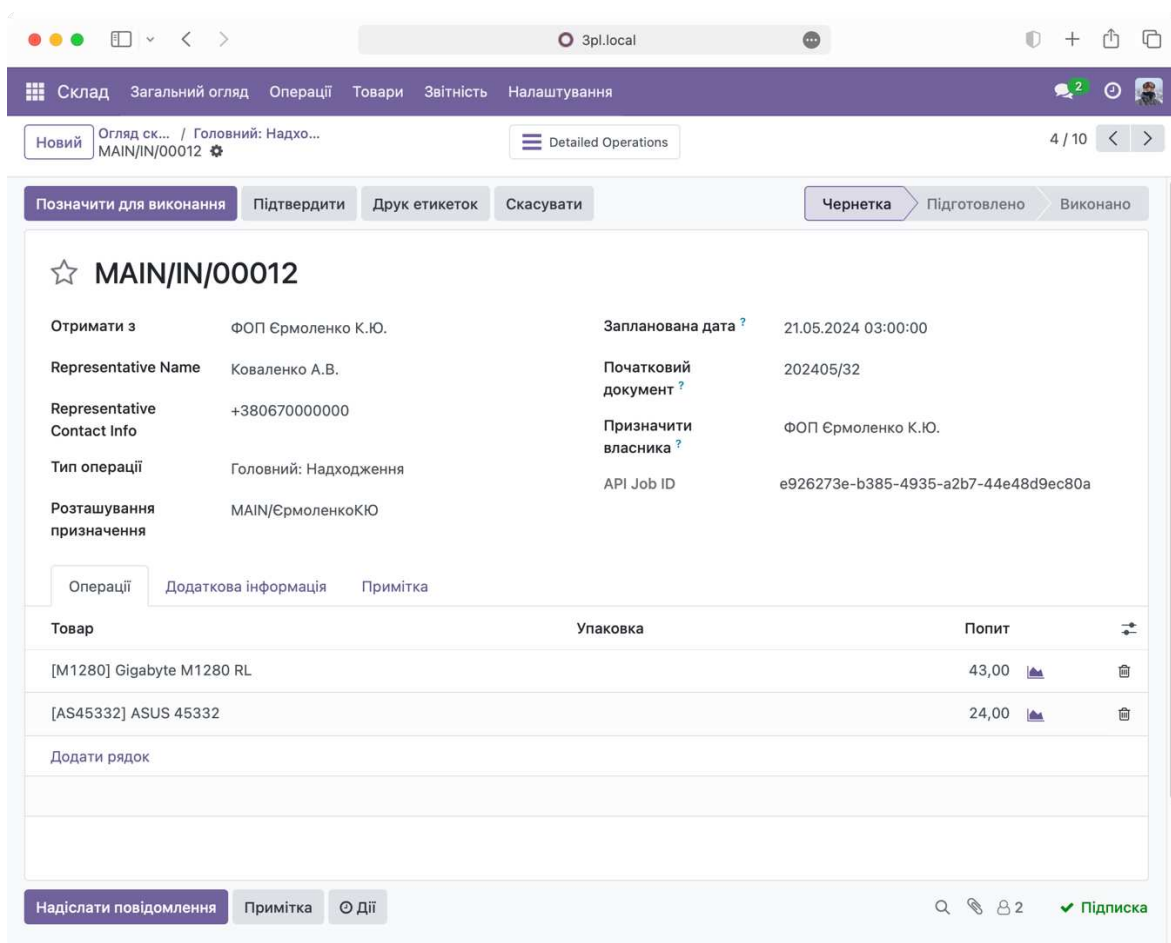


Рисунок 3.7 – Нове замовлення на завантаження створене в Odoo

Оновлена форма даних про контрагента з даними про головну локацію та налаштуваннями аутентифікації показано на Рисунку 3.8. Оновлена форма товару з вказанням власника показано на Рисунку 3.9.

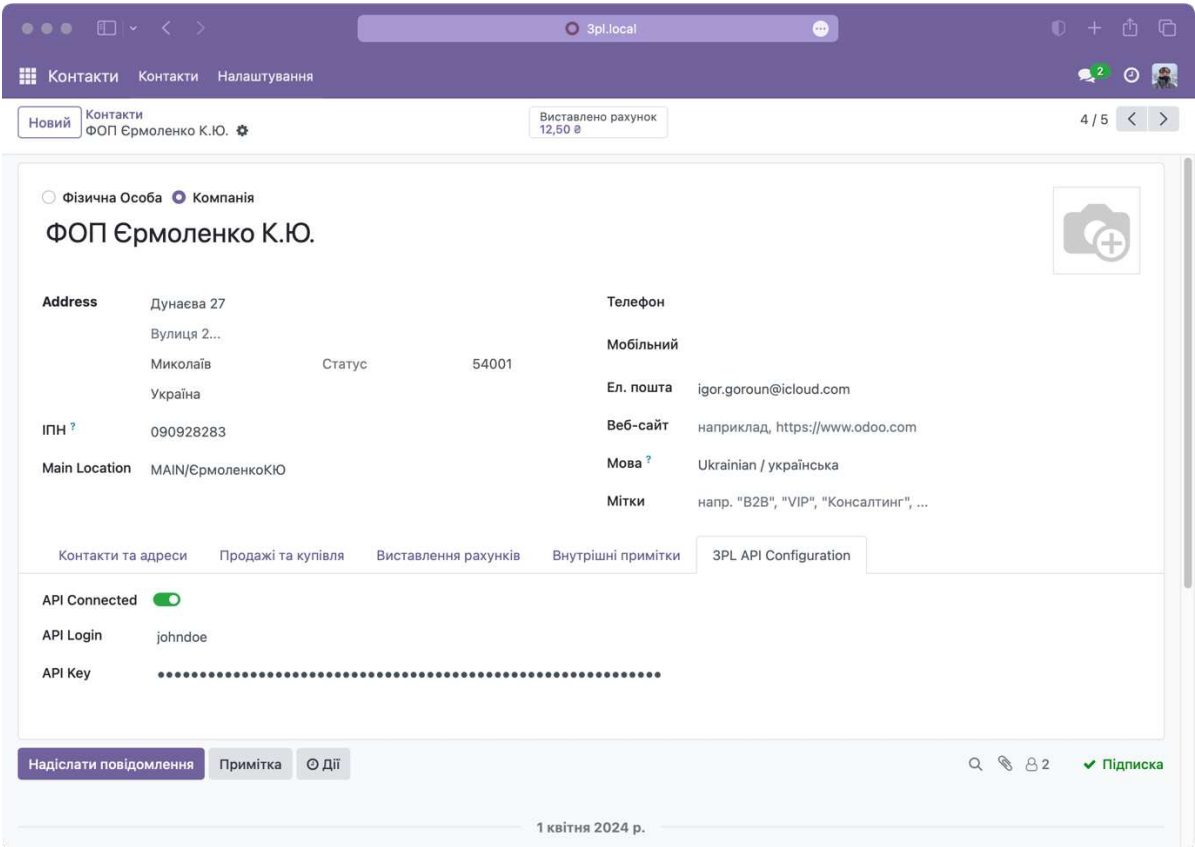


Рисунок 3.8 – Оновлені налаштування контрагента в модулі «Контакти»

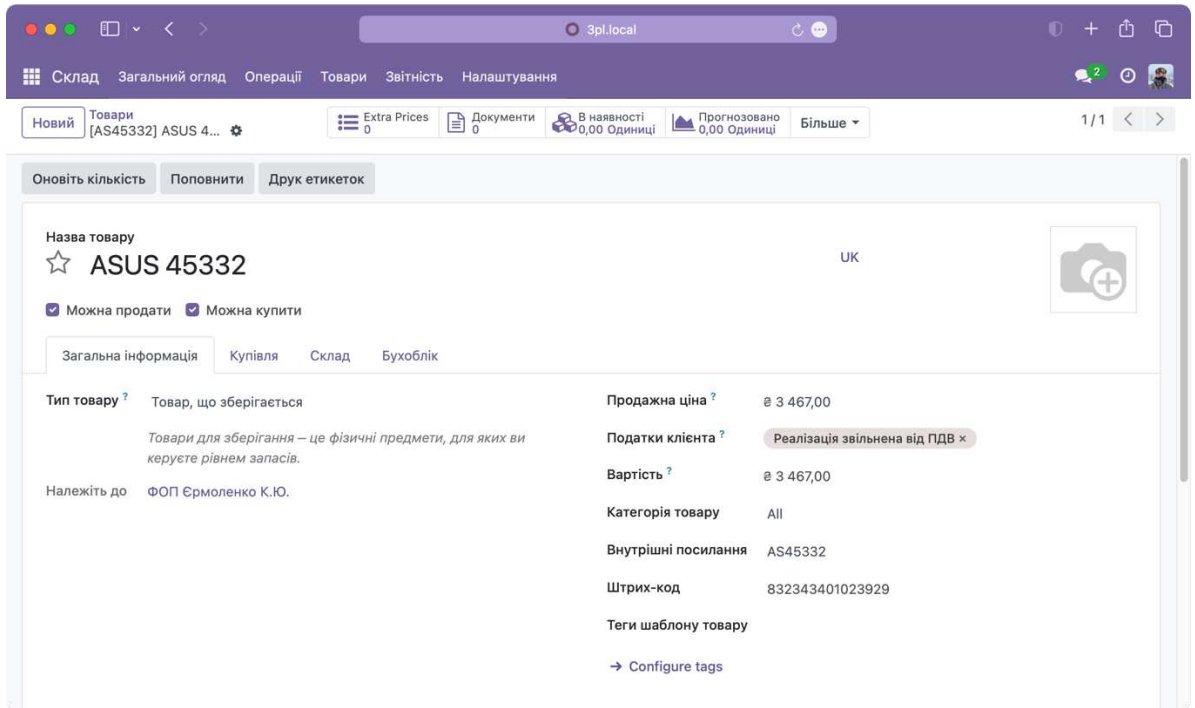


Рисунок 3.9 – Оновлена форма товару

РОЗДІЛ 4 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ

Вимоги безпеки та захисту здоров'я працівників при роботі з екранними пристроями затверджено наказом Міністерства соціальної політики від 14.02.2018 № 207. Ці вимоги поширюються на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми та видів діяльності та встановлюють мінімальні вимоги безпеки та захисту здоров'я при здійсненні роботи, пов'язаної з використанням екранних пристроїв незалежно від їх типу та моделі.

Вимоги не поширюються на:

- Робочі місця здобувачів освіти у комп'ютерних класах (кабінетах, аудиторіях тощо) навчальних закладів;
- Робочі місця пілотів, водіїв або операторів транспортних засобів, обладнані екранними пристроями у системах обробки даних на борту засобів сполучення, екранні пристрої у складі машин та обладнання, що переміщуються у процесі роботи;
- Робочі місця працівників, які займаються обслуговуванням, ремонтом та налагодженням екранних пристроїв;
- Портативні системи обробки даних, якщо вони не завжди використовуються на робочому місці;
- Обчислювальні машини (калькулятори), касові апарати та пристрої з невеликими пристроями індикації даних або результатів вимірювання;
- Цифрові друкарські машини, які обладнані візуальними дисплейними терміналами (дисплейні друкарські машини);
- Планшети, смартфони, мобільні телефони.

Роботодавець повинен проінформувати працівників під розписку про умови праці та наявність на їх робочих місцях небезпечних та шкідливих виробничих факторів (фізичних, хімічних, біологічних, психофізіологічних), які виникають при роботі з екранними пристроями та ще не усунуті, а також про можливі наслідки їх впливу на здоров'я працівників.

Роботодавець повинен забезпечити навчання та перевірку знань працівників з питань охорони праці та безпечного використання екранних пристроїв до початку роботи з ними, а також у випадках модифікації та організації роботи обладнання.

Роботодавець повинен вжити відповідних заходів, щоб забезпечити відповідність робочого місця працівника цим вимогам.

Під час облаштування робочого місця працівника з екранними пристроями необхідно вибирати таке обладнання, яке не створює надмірного шуму і не виділяє надмірного тепла. Рівні шуму на робочих місцях осіб, які працюють з екранними пристроями, повинні відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених ухвалою Головного державного санітарного лікаря України від 01.12.1999 № 37.

Роботодавець повинен за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку відповідно до Державних санітарних правил та норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря від 10.12.1998 № 7.

Роботодавець повинен забезпечити власним коштом проведення медичних оглядів працівників відповідно до вимог Порядку проведення медичних оглядів працівників певних категорій, затвердженим наказом Міністерства охорони здоров'я України від 21.05.2007 № 246.

За результатами цих оглядів роботодавець у разі потреби повинен забезпечити виконання відповідних оздоровчих заходів.

Роботодавець зобов'язаний за необхідності проводити лабораторні дослідження умов праці працівників з метою виявлення шкідливих та небезпечних факторів виробничого середовища, тяжкості та напруженості трудового процесу (зокрема щодо виявлення ризиків, пов'язаних із погіршенням зору, порушенням фізичного стану, стресом) та здійснювати заходи щодо усунення виявлених ризиків .

Робочі місця працівників з екранними пристроями мають бути спроектовані і мати такі розміри, щоб працівники мали простір зміни робочого становища і рухів.

Для забезпечення безпеки та захисту здоров'я працівників все випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня (вплив на людину факторів середовища - шуму, вібрації, забруднювачів, температури тощо, що не викликає соматичних або психічних розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій з точки зору безпеки та охорони здоров'я працівників.

Організація робочого місця працівника з екранними пристроями повинна забезпечувати відповідність всіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характером виконуваних робіт.

Висвітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном та навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.

Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями повинен підтримуватися на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 0.12.1999 №42 (далі - ДСП 3.3.6.042-99).

Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість при розміщенні екрана, клавіатури, документів та відповідного обладнання.

Робоче крісло має бути стійким і дозволяти працівникові з екранними пристроями легко рухатися та займати зручне положення.

Сидіння має регулюватися по висоті, спинка сидіння як по висоті, так і по нахилу. Слід передбачати підніжку для тих, кому це потрібно для зручності.

Мінімальні вимоги безпеки під час роботи з екранними пристроями:

- Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень;
- Після завершення роботи екранні пристрої слід відключати від електричної мережі;
- У разі виникнення аварійної ситуації необхідно негайно відключити пристрій від електричної мережі.

Не допускається:

- Виконувати технічне обслуговування, ремонт та налагодження екранних пристроїв безпосередньо на робочому місці працівника під час роботи з екранними пристроями;
- Відключати захисні пристрої, самовільно проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;
- Працювати з екранними пристроями, де під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності.

При виконанні робіт операторського типу, пов'язаних з нервово-емоційною напругою, у приміщеннях при роботі з екранними пристроями, на пультах та постах управління технологічними процесами та в інших приміщеннях повинні дотримуватися оптимальних умов мікроклімату відповідно до вимог ГСН 3.3.6.042 -99.

Мінімальні вимоги безпеки до екранних пристроїв:

- Екранні пристрої не повинні бути джерелом ризику для працівників;
- Все випромінювання, крім видимої частини електромагнітного спектра, має бути зведено до незначного рівня з погляду безпеки та охорони здоров'я працівників;
- Символи на екранних пристроях повинні бути чіткими, відповідного розміру. Між символами та рядками символів має бути належна відстань;
- Зображення на екрані має бути стабільним, без миготінь чи інших видів нестабільності;

- Яскравість та/або контрастність символів повинні легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов;
- Вибираючи екрани, слід віддавати перевагу таким екранам, які легко та вільно обертаються та нахиляються відповідно до потреби працівника;
- При необхідності може використовуватися окрема підставка або регульований стіл для розміщення екрана;
- Екран не повинен відблискувати або відображати світло, щоб не викликати дискомфорту у працівника під час роботи з екранними пристроями;
- Вибираючи клавіатуру, слід віддавати перевагу такій клавіатурі, яка відкидається і є автономною (відокремленою від екрану), щоб працівник міг вибрати зручну робочу позу та уникнути втоми рук (кисті та верхньої частини руки);
- Поверхня клавіатури має бути матовою, щоб уникнути відображення. Розташування клавіш і клавіші повинні полегшувати роботу з клавіатурою. Позначення клавіш має бути досить контрастним та розбірливим;
- Обладнання, що входить до робочої станції, не повинно виділяти надлишкового тепла, що може заподіяти незручності працівникам при роботі з екранними пристроями;
- При розробці, виборі, замовленні та модифікації програмного забезпечення, а також при розробці задач, що передбачають використання обладнання з екранними пристроями, роботодавець повинен керуватися таким програмним забезпеченням, яке відповідає розв'язуванню завданням і просто у використанні, а де необхідно адаптованим до рівня знань та досвіду працівника.

Відповідно до статті 43 Закону України «Про охорону праці» за порушення законодавства про охорону праці та невиконання приписів (розпоряджень) посадових осіб органів виконавчої влади з нагляду за охороною праці юридичні

та фізичні особи, які згідно із законодавством використовують найману працю, залучаються органами виконавчої влади з нагляду за охороною праці до сплати штрафу у порядку, встановленому законом.

Сплата штрафу не звільняє юридичну чи фізичну особу, яка відповідно до законодавства використовує найману працю, від усунення виявлених порушень у визначені терміни.

Максимальний розмір штрафу не може перевищувати п'яти відсотків середньомісячного фонду заробітної плати за попередній рік юридичної або фізичної особи, яка відповідно до законодавства використовує найману працю.

Несплата або неповна сплата юридичними або фізичними особами, які відповідно до законодавства використовують найману працю, штрафу тягне за собою нарахування пені на несплачену суму штрафу (його частини) з розрахунку 120% річних облікової ставки Національного банку України, що діяла в період такої несплати, за кожен день прострочення.

Інформація про охорону праці підготовлена на основі матеріалу [36].

ВИСНОВОК

У кваліфікаційній роботі розглянуто розробку інформаційної системи взаємодії провайдера складських послуг з контрагентами.

Виконаний опис об'єкту дослідження і аналіз предметної області, сформовані вимоги, на основі яких розроблена концепція оновленої інформаційної системи.

Розроблений технічний проект системи. Представлені загальносистемні рішення щодо архітектури обміну даними між компонентами системи. Спроектвані рішення по модернізації бази даних та принципів зберігання тимчасових даних. Описані алгоритми роботи проекту. Визначені програмні і технічні засоби реалізації і експлуатації проекту.

Розроблені інструкційні матеріали та проведена апробація проекту, яка показала ефективну роботу усіх програмних компонентів інформаційної системи.

На завершення роботи представлені правила охорони праці під час роботи за комп'ютером.

Актуальність та новизна проекту полягає у підвищенні ефективності роботи постачальника складських послуг за рахунок автоматизації обміну електронними документами між складом та контрагентами.

ЛІТЕРАТУРА

1. Класифікація видів економічної діяльності. – [Електронний ресурс] – https://kved.ukrstat.gov.ua/KVED2010/52/KVED10_52_10.html
2. About Us | Odoo. – [Електронний ресурс] – https://www.odoo.com/uk_UA/page/about-us
3. GNU LESSER GENERAL PUBLIC LICENSE. – [Електронний ресурс] – <https://www.gnu.org/licenses/lgpl-3.0.en.html>
4. Odoo – Wikipedia. – [Електронний ресурс] – <https://en.wikipedia.org/wiki/Odoo>
5. Source install – Odoo 17. – [Електронний ресурс] – https://www.odoo.com/documentation/17.0/administration/on_premise/source.html#prepare
6. Use containers to Build, Share and Run your applications. – [Електронний ресурс] – <https://www.docker.com/resources/what-container/>
7. PostgreSQL 14.11 Documentation – [Електронний ресурс] – <https://www.postgresql.org/docs/14/index.html>
8. Consignment: buy and sell stock without owning it – Odoo 17. – [Електронний ресурс] – https://www.odoo.com/documentation/17.0/applications/inventory_and_mrp/inventory/warehouses_storage/advanced_operations_warehouse/owned_stock.html
9. Export and import data – Odoo 17. – [Електронний ресурс] – https://www.odoo.com/documentation/17.0/applications/essentials/export_import_data.html
10. HTTPS - Wikipedia. – [Електронний ресурс] – <https://en.wikipedia.org/wiki/HTTPS>
11. The Six Guiding Principles of REST. – [Електронний ресурс] – <https://restfulapi.net/>
12. JSON Web Token. – [Електронний ресурс] – <https://jwt.io/>
13. RFC 7519 – JSON Web Token (JWT). – [Електронний ресурс] – <https://datatracker.ietf.org/doc/html/rfc7519>

14. External API – Odoo 17. – [Електронний ресурс] – https://www.odoo.com/documentation/17.0/developer/reference/external_api.html
15. Redis – The in-memory data store. – [Електронний ресурс] – <https://redis.io/>
16. Dramatiq: background tasks. – [Електронний ресурс] – <https://dramatiq.io/>
17. FastAPI. – [Електронний ресурс] – <https://fastapi.tiangolo.com/>
18. Chapter 3: A New Application - Odoo 17. – [Електронний ресурс] – https://www.odoo.com/documentation/17.0/developer/tutorials/getting_started/03_new_wapp.html
19. The Ubuntu lifecycle and release cadence – [Електронний ресурс] – <https://ubuntu.com/about/release-cycle>
20. Trusted and untrusted certificates – [Електронний ресурс] – <https://www.sslmarket.com/ssl/trusted-and-untrusted-certificate>
21. ECMA-404: The JSON data interchange syntax – [Електронний ресурс] – <https://ecma-international.org/publications-and-standards/standards/ecma-404/>
22. Advanced Topics - Dramatiq 1.15.0 documentation – [Електронний ресурс] – <https://dramatiq.io/advanced.html#messages>
23. Об'єктно-реляційне відображення - Вікіпедія – [Електронний ресурс] – https://uk.wikipedia.org/wiki/Об'єктно-реляційне_відображення
24. ORM API – Odoo 17.0 documentation – [Електронний ресурс] – <https://www.odoo.com/documentation/17.0/developer/reference/backend/orm.html>
25. Understand Redis data types – [Електронний ресурс] – <https://redis.io/docs/latest/develop/data-types/>
26. Base64 Standards – [Електронний ресурс] – <https://base64.guru/standards/base64url>
27. HMAC – Вікіпедія – [Електронний ресурс] – <https://uk.wikipedia.org/wiki/HMAC>
28. Compute – Amazon EC2 Instance Types – [Електронний ресурс] – <https://aws.amazon.com/ec2/instance-types/>
29. Hetzner Online GmbH – [Електронний ресурс] – <https://www.hetzner.com/cloud/>

30. Creation of virtual environments – [Електронний ресурс] – <https://docs.python.org/3/library/venv.html>
31. Advanced Load Balancer, Web-server & Reverse Proxy – [Електронний ресурс] – <https://www.nginx.com>
32. The Python IDE for data science and web development – [Електронний ресурс] – <https://www.jetbrains.com/pycharm/>
33. Simple Dashboard for Dramatiq – [Електронний ресурс] – https://github.com/Bogdanp/dramatiq_dashboard
34. Вартість доменів – [Електронний ресурс] – <https://www.imena.ua/ua/domains/prices>
35. User Docs – Odoo 17.0 documentation – [Електронний ресурс] – <https://www.odoo.com/documentation/17.0/applications.html>
36. Вступили в силу нові вимоги до роботи за комп'ютером – [Електронний ресурс] – <https://medoc.ua/blog/vstupili-v-silu-novi-vimogi-do-roboti-za-kompjuterom>

ДОДАТОК А

JSON-представлення сутностей даних для обміну між системами

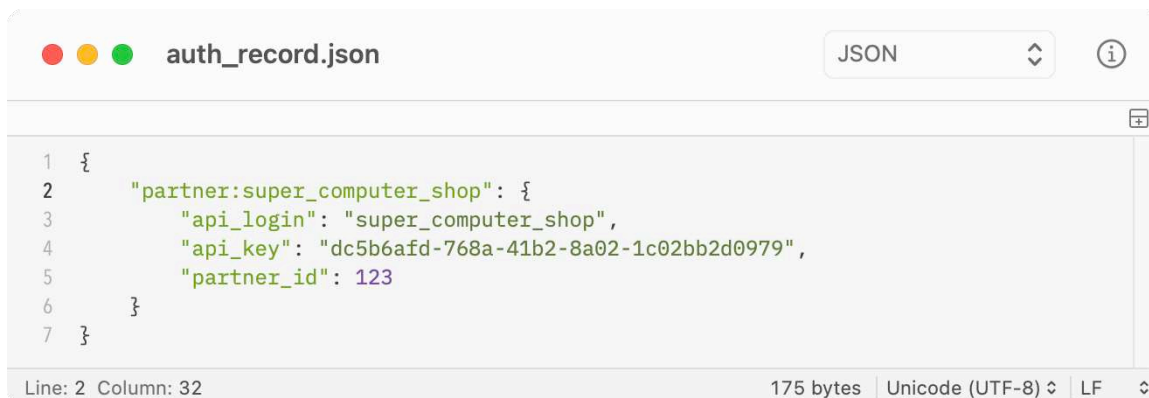


Рисунок А.1 – Представлення сутності «Дані авторизації» у форматі JSON

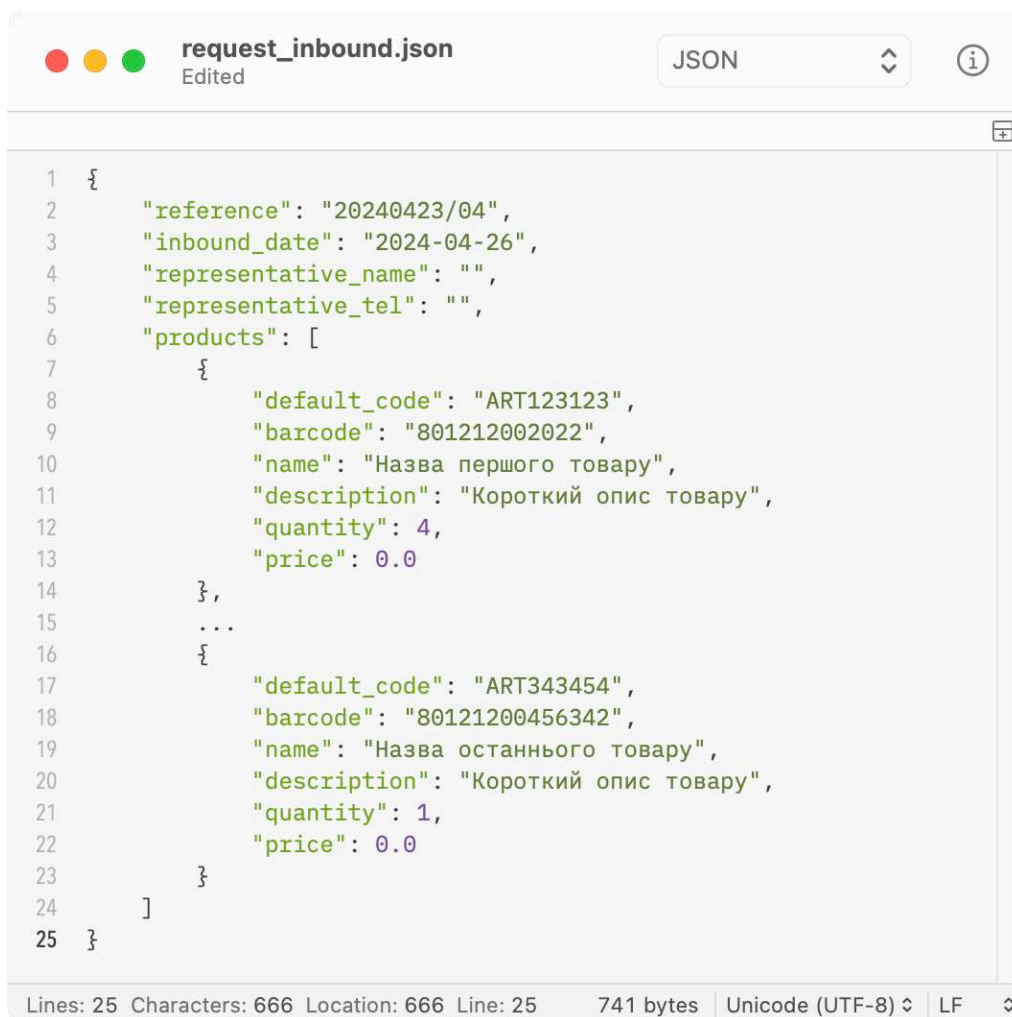


Рисунок А.2 – Представлення електронного документу «Замовлення на завантаження» у форматі JSON

```

1  {
2    "reference": "20240423/06",
3    "outbound_date": "2024-04-25",
4    "representative_name": "ПІБ представника",
5    "representative_tel": "+380501239845",
6    "packaging": true,
7    "products": [
8      {
9        "default_code": "ART123123",
10       "quantity": 2,
11     },
12     ...
13     {
14       "default_code": "ART343454",
15       "quantity": 1,
16     }
17   ]
18 }

```

Line: 3 Column: 32 411 bytes Unicode (UTF-8) LF

Рисунок А.3 – Представлення електронного документу «Замовлення на відвантаження» у форматі JSON

```

1  {
2    "date_start": "2024-03-01",
3    "date_end": "2024-03-31",
4    "include_invoice_file": true
5  }

```

Line: 5 Column: 1 301 bytes Unicode (UTF-8) LF

Рисунок А.4 – Представлення електронного документу «Виставлення рахунку» у форматі JSON

```

1  {
2    "queue_name": "default",
3    "actor_name": "create_inbound_order",
4    "kwargs": {
5      "uuid": "dc5b6afd-768a-41b2-8a02-1c02bb2d0979",
6      "partner_id": 342,
7      "reference": "20240423/04",
8      "inbound_date": "2024-04-26",
9      "representative_name": "",
10     "representative_tel": "",
11     "products": [
12       {
13         "default_code": "ART123123",
14         "barcode": "801212002022",
15         "name": "Назва першого товару",
16         "description": "Короткий опис товару",
17         "quantity": 4,
18         "price": 0.0
19       },
20       ...
21       {
22         "default_code": "ART343454",
23         "barcode": "80121200456342",
24         "name": "Назва останнього товару",
25         "description": "Короткий опис товару",
26         "quantity": 1,
27         "price": 0.0
28       }
29     ]
30   },
31   "message_id": "ea2c942a-f6d8-46ca-ae7b-0e2cd250d3cb",
32   "message_timestamp": 0
33 }

```

Line: 6 Column: 25 1 kB Unicode (UTF-8) LF

Рисунок А.5 – Представлення електронного документу «Завдання на опрацювання» у форматі JSON

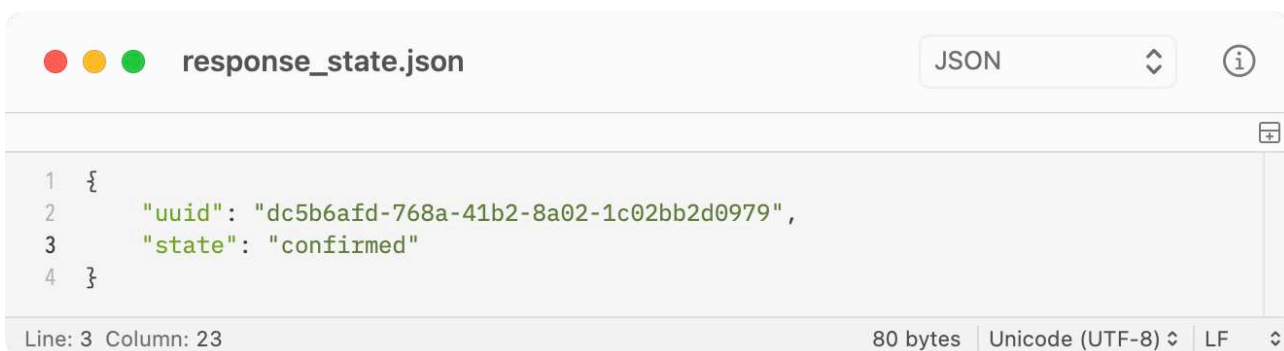
```

1  {
2    "uuid": "dc5b6afd-768a-41b2-8a02-1c02bb2d0979",
3    "message": "Запит прийнято"
4  }

```

Line: 4 Column: 1 100 bytes Unicode (UTF-8) LF

Рисунок А.6 – Представлення електронного документу «Відповідь на запит» у форматі JSON



The screenshot shows a code editor window titled 'response_state.json'. The JSON content is as follows:

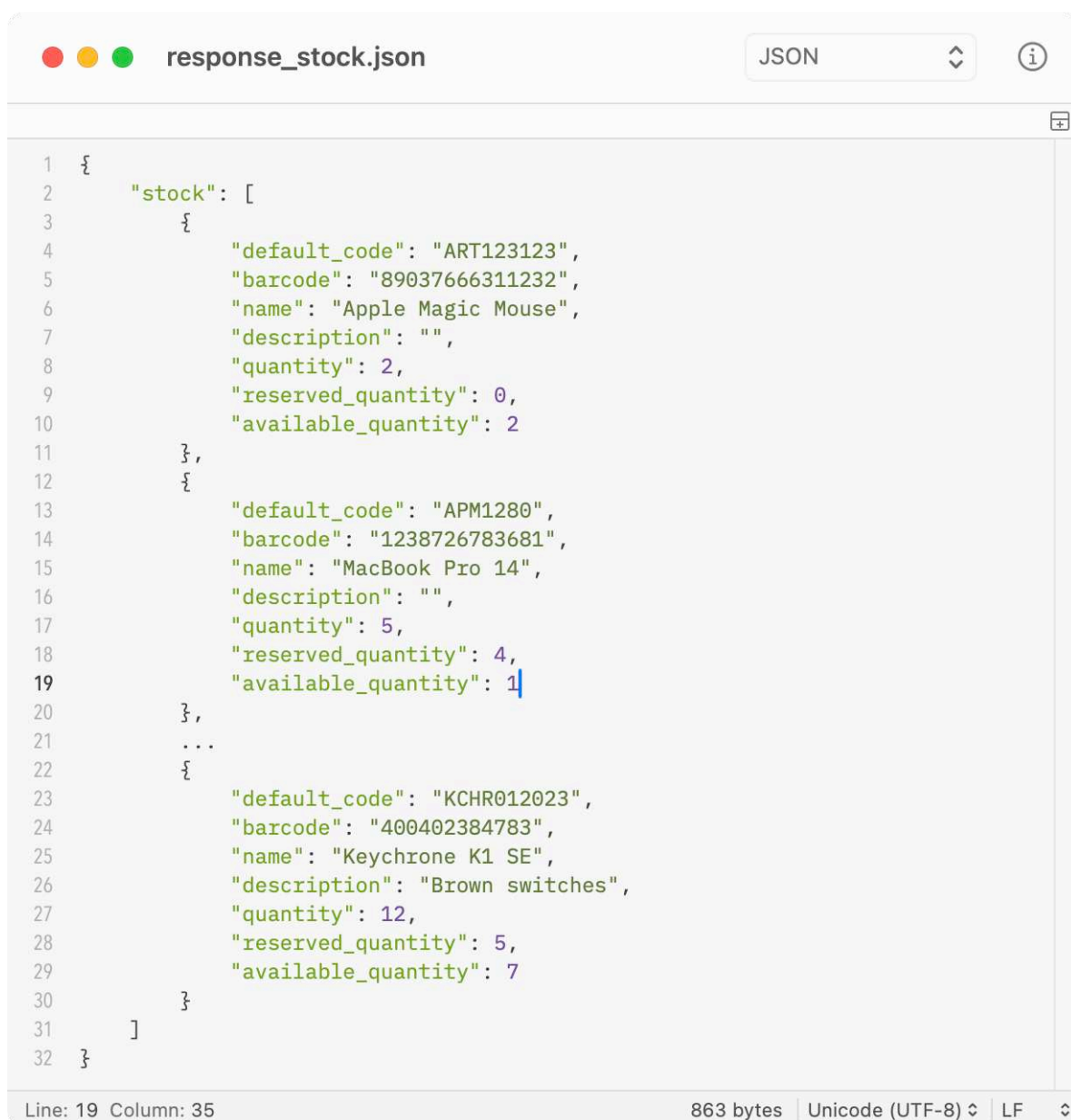
```

1 {
2   "uuid": "dc5b6afd-768a-41b2-8a02-1c02bb2d0979",
3   "state": "confirmed"
4 }

```

The status bar at the bottom indicates 'Line: 3 Column: 23', '80 bytes', 'Unicode (UTF-8)', and 'LF'.

Рисунок А.7 – Представлення сутності «Стан складської операції» у форматі JSON



The screenshot shows a code editor window titled 'response_stock.json'. The JSON content is as follows:

```

1 {
2   "stock": [
3     {
4       "default_code": "ART123123",
5       "barcode": "89037666311232",
6       "name": "Apple Magic Mouse",
7       "description": "",
8       "quantity": 2,
9       "reserved_quantity": 0,
10      "available_quantity": 2
11    },
12    {
13      "default_code": "APM1280",
14      "barcode": "1238726783681",
15      "name": "MacBook Pro 14",
16      "description": "",
17      "quantity": 5,
18      "reserved_quantity": 4,
19      "available_quantity": 1
20    },
21    ...
22  ]
23 }

```

The status bar at the bottom indicates 'Line: 19 Column: 35', '863 bytes', 'Unicode (UTF-8)', and 'LF'.

Рисунок А.8 – Представлення сутності «Наявність товарів» у форматі JSON

```

1  {
2    "invoices": [
3      {
4        "reference": "INV/2024/00234",
5        "issue_date": "2024-04-21",
6        "deadline_date": "2024-04-30",
7        "amount_total": 2500.0,
8        "amount_tax": 0.0,
9        "requisites": {
10         "signer_name": "Горун І.П.",
11         "signer_position": "Директор",
12         "company_name": "ФОП Горун І.П.",
13         "vat": "2944013075",
14         "zip": "54001",
15         "country": "Україна",
16         "state": "Миколаївська область",
17         "city": "Миколаїв",
18         "street": "вул. Схована, 12",
19         "email": "igor.goroun@gmail.com",
20         "phone": "+380971234567",
21         "notes": ""
22       },
23       "invoice_lines": [
24         {
25           "default_code": "INBOUND",
26           "name": "Прийняття товару",
27           "quantity": 467,
28           "price": 2.5,
29           "amount": 1167.5,
30           "tax": 0.0
31         },
32         ...
33         {
34           "default_code": "OUTBOUND",
35           "name": "Відвантаження товару",
36           "quantity": 490,
37           "price": 2.5,
38           "amount": 1225.0,
39           "tax": 0.0
40         }
41       ]
42     },
43     ...
44   ]
45 }

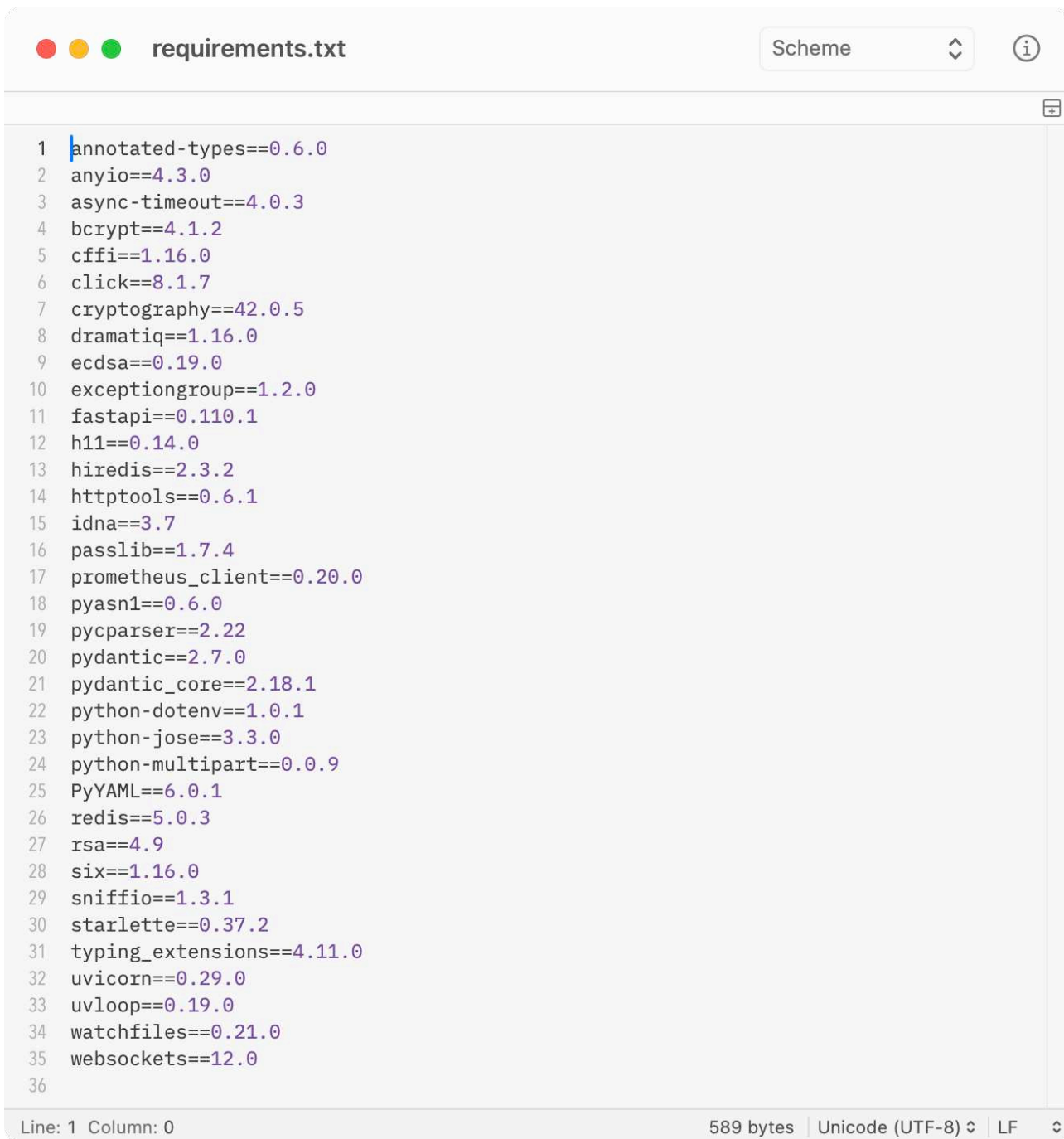
```

Line: 32 Column: 19 2 kB Unicode (UTF-8) LF

Рисунок А.9 – Представлення сутності «Актуальні рахунки» у форматі JSON

ДОДАТОК Б

Програмні рішення щодо застосунків «Інтеграція», «Склад» та «Черга»



```
requirements.txt
Scheme ⓘ
1 annotated-types==0.6.0
2 anyio==4.3.0
3 async-timeout==4.0.3
4 bcrypt==4.1.2
5 cffi==1.16.0
6 click==8.1.7
7 cryptography==42.0.5
8 dramatiq==1.16.0
9 ecdsa==0.19.0
10 exceptiongroup==1.2.0
11 fastapi==0.110.1
12 h11==0.14.0
13 hiredis==2.3.2
14 httptools==0.6.1
15 idna==3.7
16 passlib==1.7.4
17 prometheus_client==0.20.0
18 pyasn1==0.6.0
19 pycparser==2.22
20 pydantic==2.7.0
21 pydantic_core==2.18.1
22 python-dotenv==1.0.1
23 python-jose==3.3.0
24 python-multipart==0.0.9
25 PyYAML==6.0.1
26 redis==5.0.3
27 rsa==4.9
28 six==1.16.0
29 sniffio==1.3.1
30 starlette==0.37.2
31 typing_extensions==4.11.0
32 uvicorn==0.29.0
33 uvloop==0.19.0
34 watchfiles==0.21.0
35 websockets==12.0
36
```

Line: 1 Column: 0 589 bytes | Unicode (UTF-8) | LF

Рисунок Б.1 – Перелік модулів Python-середовища для додатку «Інтеграція»


```

15 cache_db = Redis(
16     host=os.getenv("REDIS_HOST"),
17     port=6379,
18     username=os.getenv("REDIS_USER"),
19     password=os.getenv("REDIS_PASSWORD"),
20     decode_responses=True
21 )

```

Рисунок Б.2 – Підключення до БД Redis

```

2 usages new *
74 def get_user(api_login: str) -> RedisUser | bool:
75     user_data = cache_db.get(f"{PARTNER_LOGIN_KEY}:{api_login}")
76     user = json.loads(user_data) if user_data else None
77     if user:
78         return RedisUser(**user)
79     return False

```

Рисунок Б.3 – Отримання даних автентифікації з БД

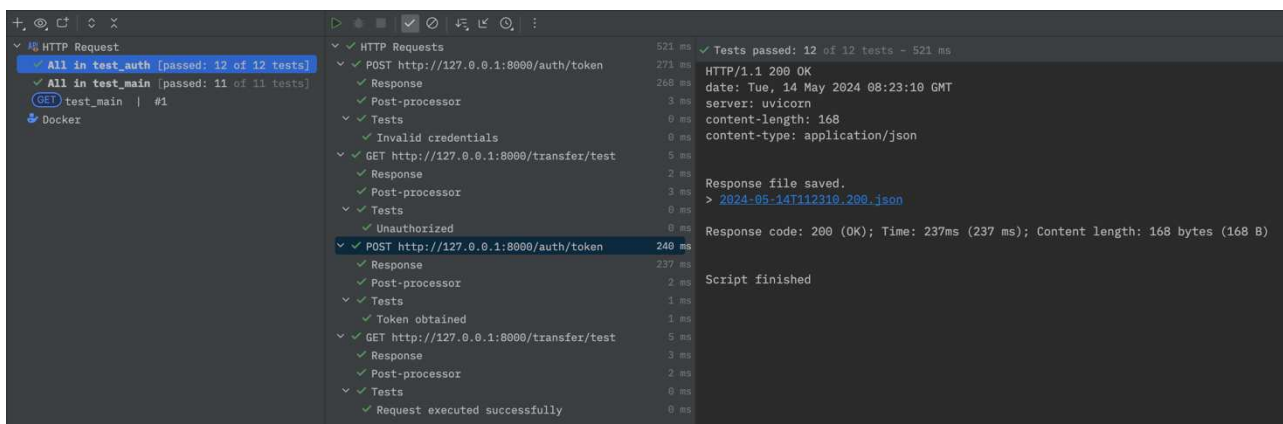


Рисунок Б.4 – Результати тестування функції отримання веб-токену

```

4      ### Запитуємо токен - негативний тест
5      POST http://127.0.0.1:8000/auth/token
6      Content-Type: application/x-www-form-urlencoded
7
8      username = johndoe &
9      password = wrongpassw
10
11      > {%
12          client.test("Invalid credentials", function () {
13              client.assert(response.status === 401, "Should be 401 - unauthorized");
14          });
15      %}
16      ###
17
18      ### Тестовий запит БЕЗ ТОКЕНА на кінцеву точку яка потребує авторизації
19      GET http://127.0.0.1:8000/transfer/test
20      Content-Type: application/json
21
22      > {%
23          client.test("Unauthorized", function () {
24              client.assert(response.status === 401, "Should be 401 - unauthorized");
25          });
26      %}
27      ###
28      ⚠
29      ### Запитуємо токен - позитивний тест
30      POST http://127.0.0.1:8000/auth/token
31      Content-Type: application/x-www-form-urlencoded
32
33      username = johndoe &
34      password = dreams
35
36      > {%
37          client.test("Token obtained", function () {
38              client.assert(response.status === 200, "Response status is not 200");
39          });
40          client.global.set("auth_token", response.body.access_token);
41      %}
42      ###
43
44      ### Тестовий запит з токеном на кінцеву точку яка потребує авторизації
45      GET http://127.0.0.1:8000/transfer/test
46      Authorization: Bearer {{auth_token}}
47      Content-Type: application/json
48
49      > {%
50          client.test("Request executed successfully", function () {
51              client.assert(response.status === 200, "Response status is not 200");
52          });
53      %}

```

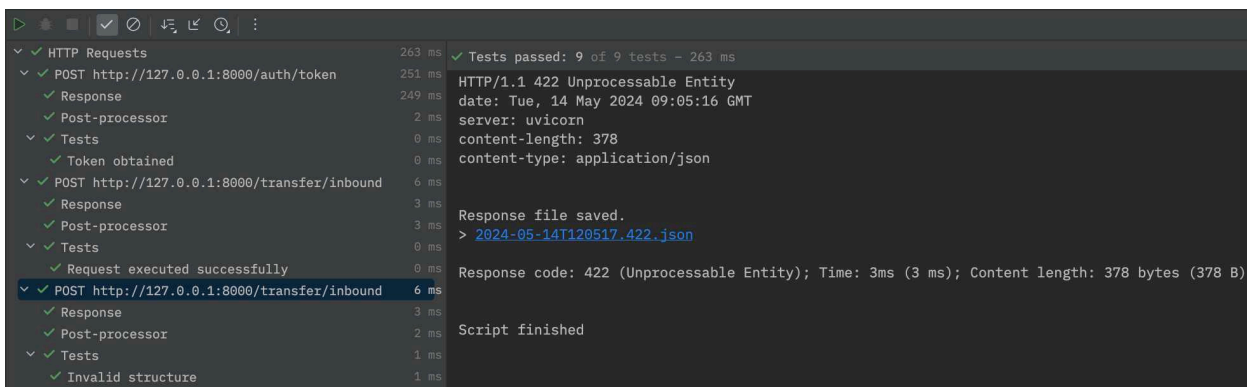
Рисунок Б.5 – Код тестів на отримання токenu

```

1 usage new *
40 class Transfer(BaseModel):
41     channel: Literal['inbound', 'outbound']
42     reference: str
43     representative_name: str = None
44     representative_tel: str = None
45     products: List[Any]
46
47     new *
48     @field_validator('field': "inbound_date", *fields: "outbound_date", mode='before', check_fields=False)
49     @classmethod
50     def string_to_date(cls, v: Any) -> Any:
51         if isinstance(v, str):
52             return datetime.strptime(v, __format: "%Y-%m-%d").date()
53         raise ValueError('Incorrect date format')
54
55 1 usage new *
56 class InboundProduct(BaseModel):
57     default_code: str
58     barcode: str = None
59     name: str
60     description: str = None
61     quantity: int
62     price: float
63
64 1 usage new *
65 class InboundTransfer(Transfer):
66     channel: str = 'inbound'
67     inbound_date: date
68     products: List[InboundProduct]

```

Рисунок Б.6 – Код класів для запиту на створення замовлення на завантаження



```

> 2024-05-14T120517.422.json
Response code: 422 (Unprocessable Entity); Time: 3ms (3 ms); Content length: 378 bytes (378 B)
Script finished

```

Рисунок Б.7 – Результат тестування перевірки даних при отриманні запиту

```

19  ### Запит на створення замовлення на завантаження - позитивний
20  POST http://127.0.0.1:8000/transfer/inbound
21  Authorization: Bearer {{auth_token}}
22  Content-Type: application/json
23
24  {
25    "reference": "202405/04",
26    "inbound_date": "2024-05-25",
27    "representative_name": "",
28    "representative_tel": "",
29    "products": [
30      {
31        "default_code": "ART343454",
32        "barcode": "80121200456342",
33        "name": "Назва останнього товару",
34        "description": "Короткий опис товару",
35        "quantity": 1,
36        "price": 32.65
37      }
38    ]
39  }
40
41  > {%
42    client.test("Request executed successfully", function () {
43      client.assert(response.status === 200, "Response status is not 200");
44    });
45  %}
46  ###
47
48  ### Запит на створення замовлення на завантаження - з невалідною структурою
49  POST http://127.0.0.1:8000/transfer/inbound
50  Authorization: Bearer {{auth_token}}
51  Content-Type: application/json
52
53  {
54    "reference": "202405/04",
55    "inbound_dates": "2024-05-25"
56  }
57
58  > {%
59    client.test("Invalid structure", function () {
60      client.assert(response.status === 422, "Should be 422 - Unprocessable Entity");
61    });
62  %}

```

Рисунок Б.8 – Код тесту для перевірки даних при отриманні запиту

```

28 2 usages new *
29 class DramatiqJobOptions(BaseModel):
30     redis_message_id: UUID4 = Field(default_factory=uuid4)
31
32     new *
33     @field_serializer('redis_message_id')
34     def serialize_redis_message_id(self, v: UUID4):
35         return str(v)
36
37 1 usage new *
38 class DramatiqJob(BaseModel):
39     queue_name: str
40     actor_name: str
41     args: tuple = ()
42     kwargs: dict
43     options: DramatiqJobOptions = Field(default_factory=DramatiqJobOptions)
44     message_id: UUID4 = Field(default_factory=uuid4)
45     message_timestamp: float = Field(default_factory=time)
46
47     new *
48     @field_serializer('message_id')
49     def serialize_redis_message_id(self, v: UUID4):
50         return str(v)
51
52     new *
53     @field_serializer('message_timestamp')
54     def serialize_ts(self, v: float):
55         return int(v)
56
57 2 usages new *
58 def push_job(payload: dict, actor_name: str, queue_name: str = 'default'):
59     job = DramatiqJob(
60         queue_name=queue_name,
61         actor_name=actor_name,
62         kwargs=payload
63     )
64     logger.warning(f"Pushing job {job.dict()}")
65     cache_db.hset( name: f'dramatiq:{queue_name}.msgs', str(job.options.redis_message_id), json.dumps(job.dict()))
66     cache_db.rpush( name: f'dramatiq:{queue_name}', *values: str(job.options.redis_message_id))

```

Рисунок Б.9 – Функції поміщення завдання в чергу

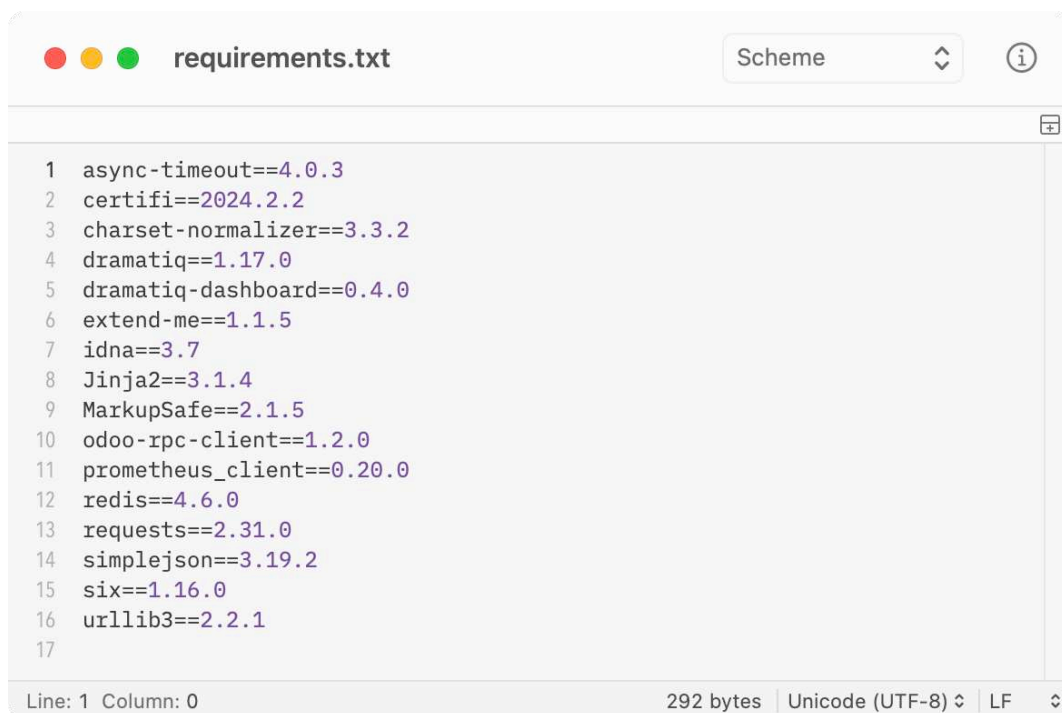


Рисунок Б.10 – Перелік модулів Python-середовища для додатку «Черга»

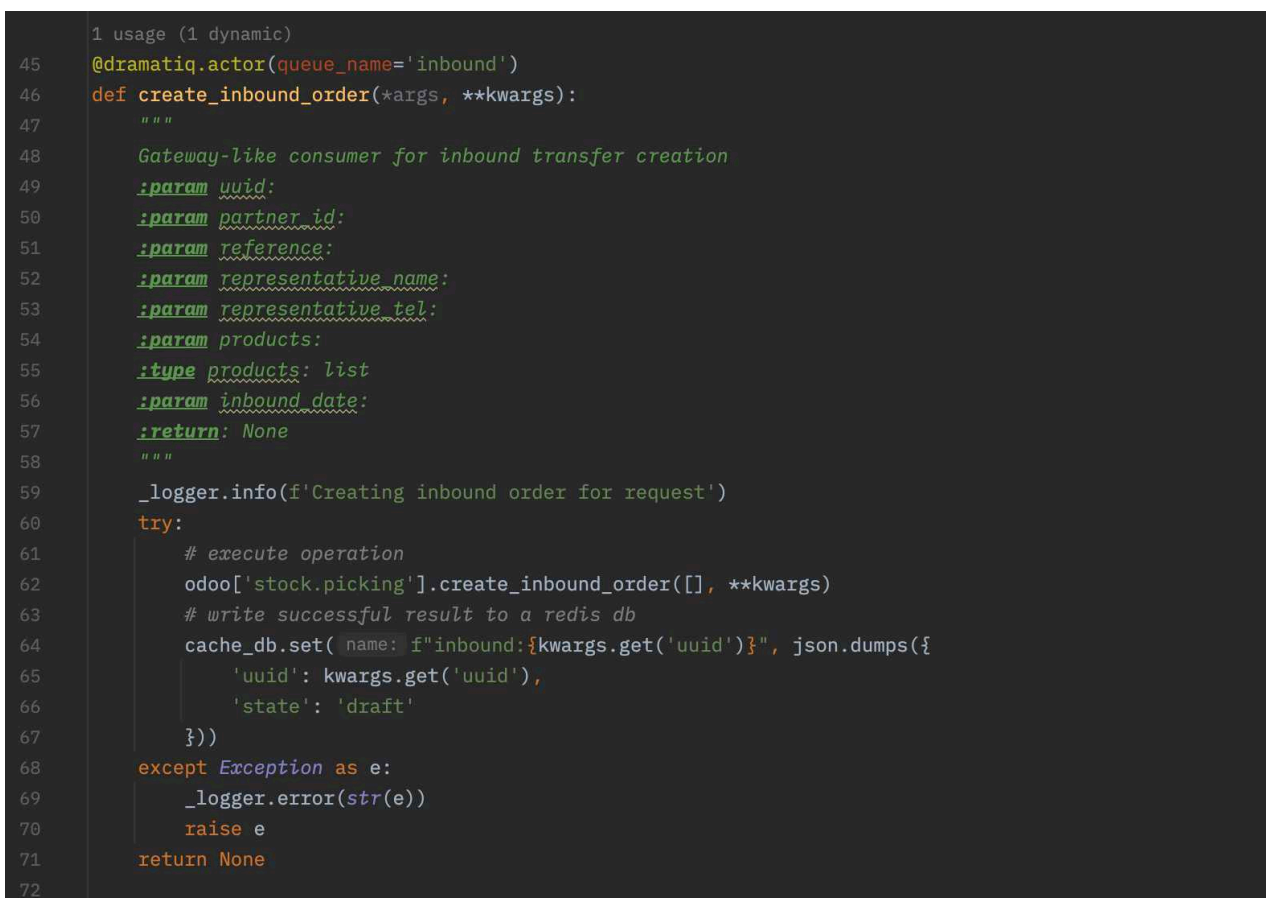
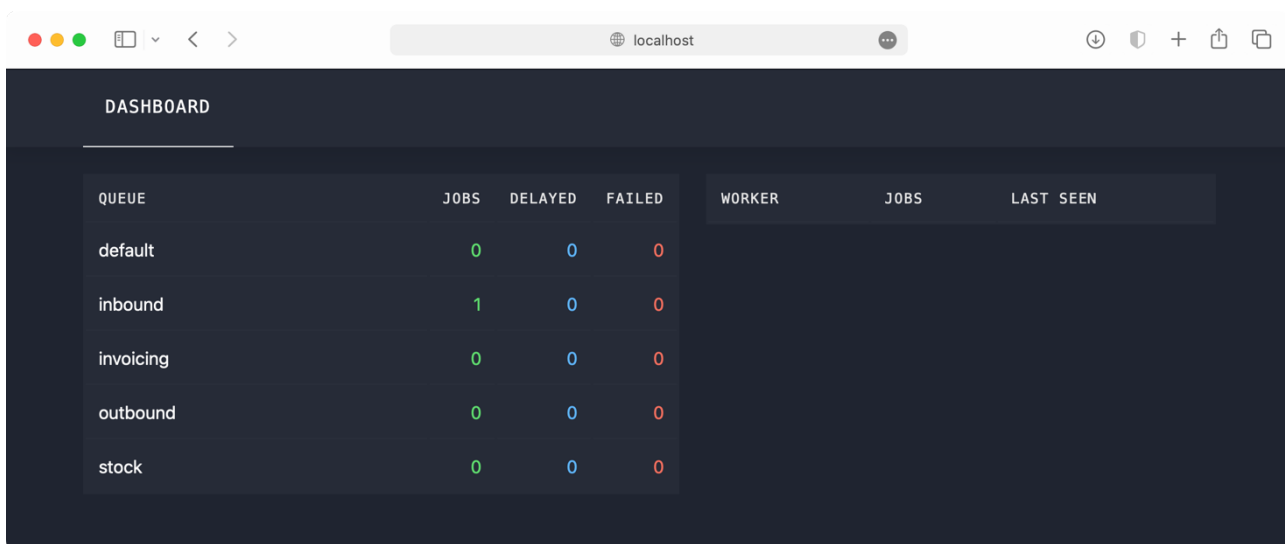


Рисунок Б.11 – Частина коду виконання обробником завдання з черги

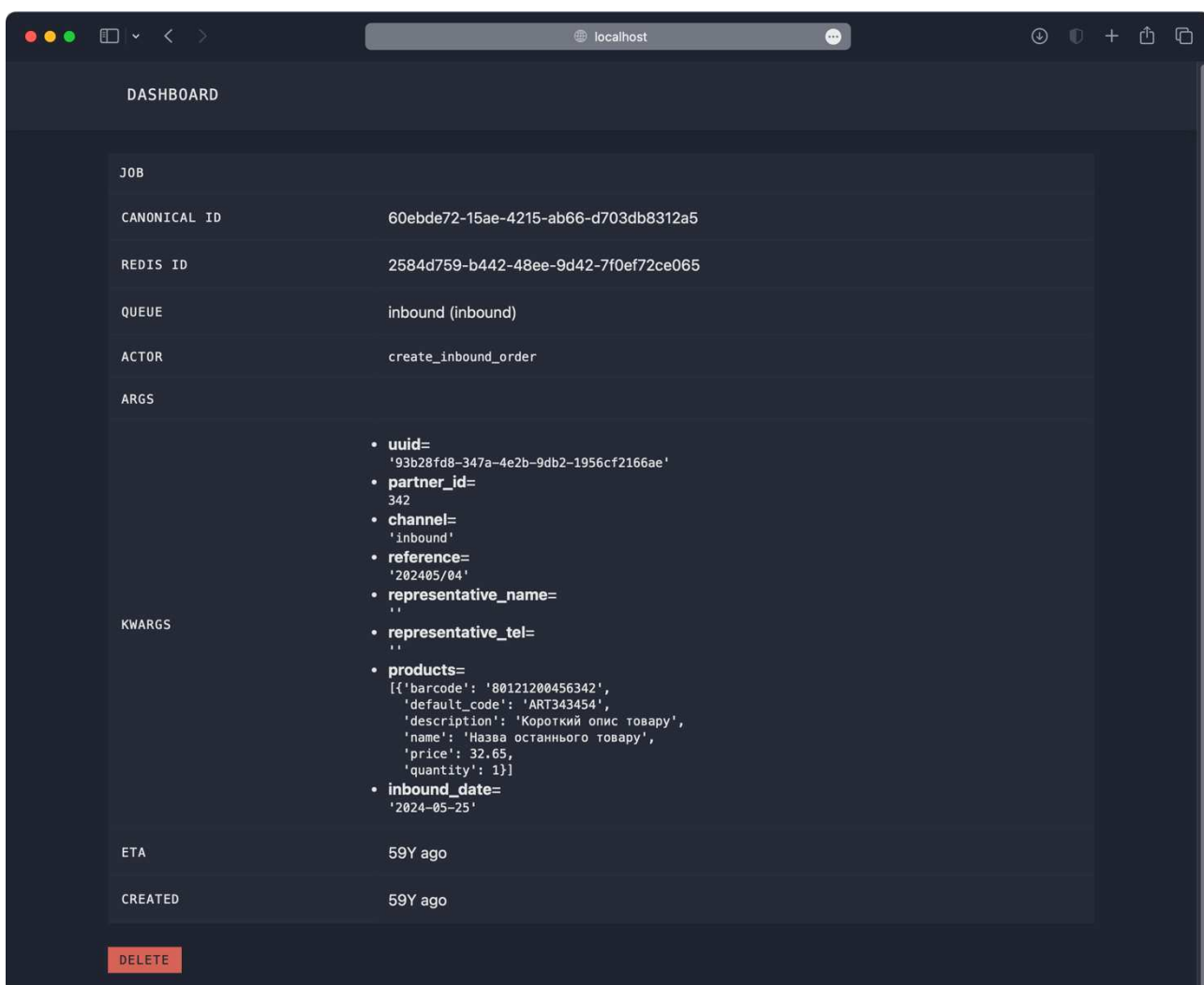


The screenshot shows the Dramatiq dashboard with a table of queues and their job status. The browser address bar shows 'localhost'.

QUEUE	JOB	DELAYED	FAILED
default	0	0	0
inbound	1	0	0
invoicing	0	0	0
outbound	0	0	0
stock	0	0	0

Below the queue table, there is a section for 'WORKER' with columns 'JOBS' and 'LAST SEEN'.

Рисунок Б.12 – Вікно моніторингу черг Dramatiq



The screenshot shows the Dramatiq dashboard with details of a specific job. The browser address bar shows 'localhost'.

JOB	
CANONICAL ID	60ebde72-15ae-4215-ab66-d703db8312a5
REDIS ID	2584d759-b442-48ee-9d42-7f0ef72ce065
QUEUE	inbound (inbound)
ACTOR	create_inbound_order
ARGS	
KWARGS	• uuid= '93b28fd8-347a-4e2b-9db2-1956cf2166ae' • partner_id= 342 • channel= 'inbound' • reference= '202405/04' • representative_name= '' • representative_tel= '' • products= [{'barcode': '80121200456342', 'default_code': 'ART343454', 'description': 'Короткий опис товару', 'name': 'Назва останнього товару', 'price': 32.65, 'quantity': 1}] • inbound_date= '2024-05-25'
ETA	59Y ago
CREATED	59Y ago

At the bottom of the job details section, there is a red button labeled 'DELETE'.

Рисунок Б.13 – Склад окремої задачі в моніторингу Dramatiq


```

Run dramatiq x
/Users/snake/Projects/3pl_queue/.venv/bin/python -m dramatiq workers -p 3 -t 5
[2024-05-14 17:59:36,680] [PID 40568] [MainThread] [dramatiq.MainProcess] [INFO] Dramatiq '1.17.0' is booting up.
[2024-05-14 17:59:36,678] [PID 40570] [MainThread] [dramatiq.WorkerProcess(0)] [INFO] Worker process is ready for action.
[2024-05-14 17:59:36,665] [PID 40571] [MainThread] [dramatiq.WorkerProcess(1)] [INFO] Worker process is ready for action.
[2024-05-14 17:59:36,665] [PID 40572] [MainThread] [dramatiq.WorkerProcess(2)] [INFO] Worker process is ready for action.
[2024-05-14 17:59:36,748] [PID 40573] [MainThread] [dramatiq.ForkProcess(0)] [INFO] Fork process 'dramatiq.middleware
.prometheus:run_exposition_server' is ready for action.
[2024-05-14 17:59:50,360] [PID 40572] [Thread-6] [workers.create_inbound_order] [DEBUG] Received args=() kwargs={'uuid':
'9c939e49-e473-4165-97b7-9fb6129d8859', 'partner_id': 342, 'channel': 'inbound', 'reference': '202405/04',
'representative_name': '', 'representative_tel': '', 'products': [{'default_code': 'ART343454', 'barcode':
'80121200456342', 'name': 'Назва останнього товару', 'description': 'Короткий опис товару', 'quantity': 1, 'price':
32.65}], 'inbound_date': '2024-05-25'}.
[2024-05-14 17:59:50,360] [PID 40572] [Thread-6] [workers] [INFO] Creating inbound order for request
[2024-05-14 17:59:50,362] [PID 40572] [Thread-6] [workers.create_inbound_order] [DEBUG] Completed after 1.84ms.

```

Рисунок Б.14 – Журнал виконання завдання обробником

```

1  {
2      'name': "Third-party logistics stock management",
3      'application': True,
4      'version': '17.0.0.1',
5      'depends': ['stock'],
6      'author': "Ihor Horun",
7      'category': 'Inventory/Inventory',
8      'license': 'LGPL-3',
9      'description': """
10         Third-party logistics stock management.
11         Qualification work of Ihor Horun, student of group number 4142
12     """,
13     'data': [
14         'views/res_partner_views.xml',
15         'views/stock_picking_views.xml',
16     ]
17 }

```

Рисунок Б.15 – Маніфест нового модуля для Odoo

```

1  from odoo import models, fields, api
2  from datetime import date, datetime
3  import logging
4
5  _logger = logging.getLogger(__name__)
6
7
8  class StockPicking(models.Model):
9      _inherit = 'stock.picking'
10
11     api_uuid = fields.Char(string='API Job ID', readonly=True, )
12     partner_representative_name = fields.Char(string='Representative Name', readonly=False, )
13     partner_representative_tel = fields.Char(string='Representative Contact Info', readonly=False, )
14

```

Рисунок Б.16 – Оновлення логічної моделі «Складська операція»


```

1 import logging
2 from odoo import models, fields, api
3
4
5 _logger = logging.getLogger(__name__)
6
7 class ResPartner(models.Model):
8     _inherit = 'res.partner'
9
10     main_location_id = fields.Many2one(comodel_name='stock.location', string='Main Location', required=False, ondelete='set null', )
11     api_connected = fields.Boolean(string='API Connected', default=False, )
12     api_login = fields.Char(string='API Login', required=False, )
13     api_key = fields.Char(string='API Key', required=False, )
14
15

```

Рисунок Б.17 – Оновлення логічної моделі «Контакт»

```

8 class StockPicking(models.Model):
9     _inherit = 'stock.picking'
10
11     api_uuid = fields.Char(string='API Job ID', readonly=True, )
12     partner_representative_name = fields.Char(string='Representative Name', readonly=False, )
13     partner_representative_tel = fields.Char(string='Representative Contact Info', readonly=False, )
14
15     def create_inbound_order(self, *args, uuid: str = None, partner_id: int = None, reference: str = None,
16                             representative_name: str = None, representative_tel: str = None,
17                             products: list = None, inbound_date: str = None):
18
19         # format date
20         if inbound_date is not None:
21             inbound_date = datetime.strptime(inbound_date, '%Y-%m-%d')
22         else:
23             inbound_date = date.today()
24         # check products existed in request
25         if products is None or not products:
26             raise ValueError('Products list is empty')
27         # get partner and warehouse/locations objects
28         partner = self.env['res.partner'].browse(partner_id)
29         warehouse = self.env['stock.warehouse'].search(domain=[('code', '=', 'MAIN')], limit=1)
30         src_location = self.env.ref('stock.stock_location_suppliers')
31         dest_location = partner.main_location_id or warehouse.in_type_id.default_location_dest_id
32         # create transfer
33         inbound_transfer = self.create([
34             {
35                 'api_uuid': uuid,
36                 'note': "API-created transfer with job uuid: {}".format(uuid),
37                 'partner_id': partner.id,
38                 'owner_id': partner.id,
39                 'partner_representative_name': representative_name,
40                 'partner_representative_tel': representative_tel,
41                 'origin': reference,
42                 'scheduled_date': inbound_date,
43                 'picking_type_id': warehouse.in_type_id.id,
44                 'location_id': src_location.id,
45                 'location_dest_id': dest_location.id,
46                 'move_ids': self.prepare_transfer_moves(products, partner.id, src_location.id, dest_location.id)
47             }
48         ])
49         _logger.info(f'Inbound order created: {inbound_transfer}')
50         return True

```

Рисунок Б.18 – Функція створення замовлення на завантаження в Odoo

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <odoo>
3   <record id="view_3pl_stock_picking_form" model="ir.ui.view">
4     <field name="name">view.3pl.stock.picking.form</field>
5     <field name="model">stock.picking</field>
6     <field name="inherit_id" ref="stock.view_picking_form" />
7     <field name="arch" type="xml">
8       <field name="partner_id" position="after">
9         <field name="partner_representative_name" />
10        <field name="partner_representative_tel" />
11      </field>
12      <field name="owner_id" position="after">
13        <field name="api_uid"/>
14      </field>
15    </field>
16  </record>
17 </odoo>

```

Рисунок Б.19 – Код оновлення екранної форми моделі «Складська операція»

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <odoo>
3   <data>
4     <record id="view_3pl_stock_partner_form" model="ir.ui.view">
5       <field name="name">view.3pl.stock.res.partner.form</field>
6       <field name="model">res.partner</field>
7       <field name="inherit_id" ref="base.view_partner_form" />
8       <field name="priority" eval="1" />
9       <field name="arch" type="xml">
10        <xpath expr="//notebook" position="inside">
11          <page name="3pl_api" string="3PL API Configuration">
12            <group>
13              <field name="api_connected" widget="boolean_toggle"/>
14              <field name="api_login" invisible="not api_connected"/>
15              <field name="api_key" password="True" invisible="not api_connected"/>
16            </group>
17          </page>
18        </xpath>
19        <field name="vat" position="after">
20          <field name="main_location_id" />
21        </field>
22      </field>
23    </record>
24  </data>
25 </odoo>

```

Рисунок Б.20 – Код оновлення екранної форми моделі «Контакт»

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <odoo>
3   <record id="view_3pl_stock_product_form" model="ir.ui.view">
4     <field name="name">view.3pl.stock.product.form</field>
5     <field name="model">product.template</field>
6     <field name="inherit_id" ref="product.product_template_form_view" />
7     <field name="arch" type="xml">
8       <xpath expr="//group[@name='group_general']" position="inside">
9         <field name="owner_id" />
10      </xpath>
11    </field>
12  </record>
13 </odoo>

```

Рисунок Б.21 – Код оновлення екранної форми моделі «Товар»

ДОДАТОК В

Інструкція для системного адміністратора по розгортанню та налаштуванню застосунку «Інтеграція»

Повний код застосунку «Інтеграція» знаходиться у репозиторії за посиланням: https://github.com/igorgoroun/3pl_api.

Крок 1. Встановіть базу даних Redis згідно документації доступною за посиланням – https://redis.io/docs/latest/operate/oss_and_stack/install/install-redis/. Налаштуйте доступ до БД тільки для авторизованого користувача згідно документації по правам доступу до Redis доступною за посиланням – https://redis.io/docs/latest/operate/oss_and_stack/management/security/.

Крок 2. Завантажте файли застосунку в окрему директорію на сервері, створіть нове віртуальне середовище Python та встановіть все необхідні модулі:

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

Крок 3. Скопіюйте файл із змінними середовища та вкажіть в ньому дані доступу до БД Redis, налаштовані у Кроці 1:

```
cp .env.default .env
vi .env
```

Файл має 4 параметри налаштування:

```
REDIS_HOST=localhost
REDIS_PORT=6379
REDIS_USER=YOUR_REDIS_USER
REDIS_PASSWORD=YOUR_REDIS_PASS
```

Крок 4. Для запуску FastAPI виконайте наступну команду, :

```
python -m uvicorn api.main:logistics_api
```

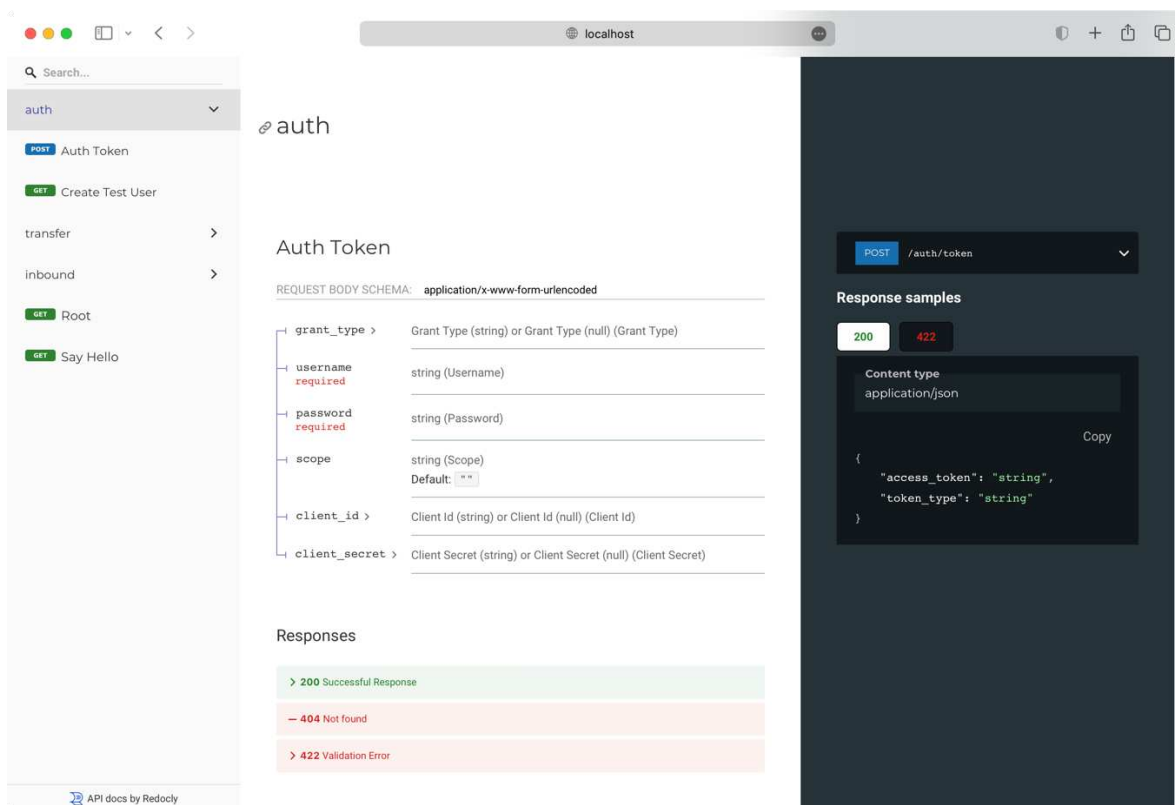
Або дотримуйтесь вказівок щодо налаштування сервісу FastAPI за допомогою системи управління Linux-сервісами SYSTEMD згідно інструкції за посиланням <https://www.dermitch.de/post/systemd-python-daemons-part-1/> , або за допомогою Docker згідно документації доступної за посиланням <https://fastapi.tiangolo.com/ru/deployment/docker/> .

Крок 5. Встановіть та налаштуйте веб-сервер NGINX згідно його документації по встановленню та налаштування, доступної за посиланням <https://nginx.org/en/docs/install.html>.

Крок 6. Перевірте доступність REST-запитів, та доступність документації по запитам. При успішному встановленню та налаштуванню застосунку, згідно документації по FastAPI, документація по запитам має бути доступна за посиланням:

АДРЕСА_СЕРВЕРУ/redoc

Документація має виглядати як показано на рисунку нижче:



ДОДАТОК Г

Інструкція для системного адміністратора по розгортанню та налаштуванню застосунку «Черга»

Повний код застосунку «Інтеграція» знаходиться у репозиторії за посиланням: https://github.com/igorgoroun/3pl_queue.

Крок 1. Завантажте файли застосунку в окрему директорію на сервері, створіть нове віртуальне середовище Python та встановіть все необхідні модулі:

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

Крок 2. Скопіюйте файл із змінними середовища та вкажіть в ньому дані доступу до БД Redis, налаштовані у Кроці 3 Додатку В, дані доступу до підсистеми Odoo та дані, необхідні для запуску моніторингу черг:

```
cp .env.default .env
vi .env
```

Файл має 11 параметрів налаштування:

```
# Odoo
ODOO_RPC_HOST=localhost
ODOO_RPC_PORT=8069
ODOO_RPC_DB=dbname
ODOO_RPC_USER=ODOO_LOGIN
ODOO_RPC_PASS=ODOO_PASS
# Redis
REDIS_HOST=localhost
REDIS_PORT=6379
REDIS_USER=REDIS_LOGIN
REDIS_PASSWORD=REDIS_PASS
# Dashboard
DASH_BINDING_PORT=8080
DASH_BINDING_HOST=127.0.0.1
```

Крок 3. Запустіть менеджер черг Dramatiq, для запуску виконайте наступну команду під активованим віртуальним середовищем Python:

```
dramatiq workers
```

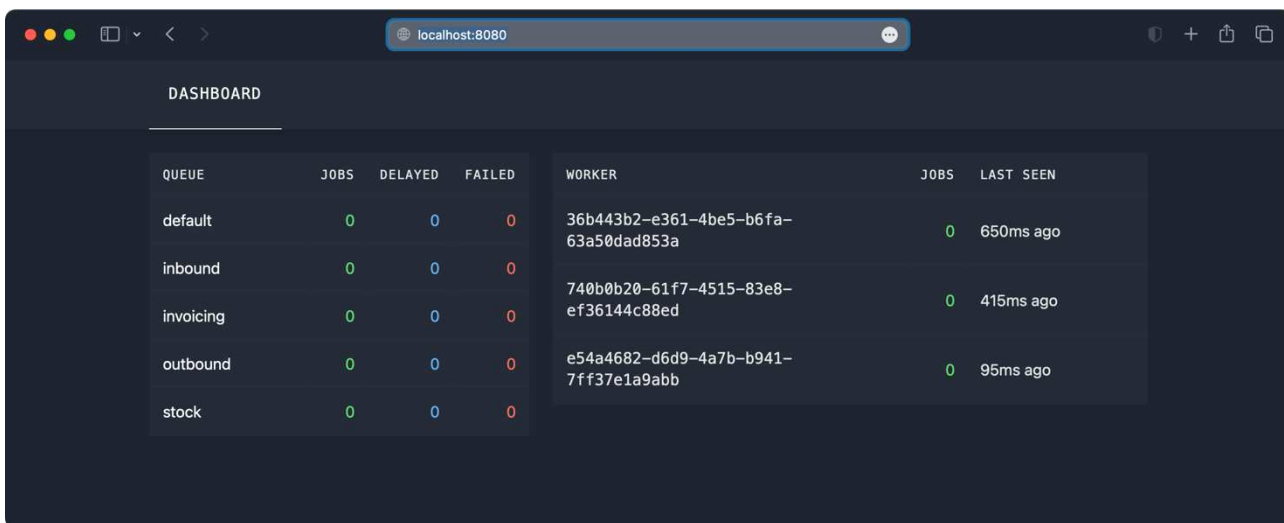
При успішному запуску, менеджер черг напише в журналі інформацію про активні процеси, як показано на рисунку:

```
(.venv) snake@~/P/3pl_queue $ dramatiq workers
[2024-05-18 14:50:15,207] [PID 61298] [MainThread] [dramatiq.MainProcess] [INFO] Dramatiq '1.17.0' is booting up.
[2024-05-18 14:50:15,182] [PID 61300] [MainThread] [dramatiq.WorkerProcess(0)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61301] [MainThread] [dramatiq.WorkerProcess(1)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61302] [MainThread] [dramatiq.WorkerProcess(2)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61303] [MainThread] [dramatiq.WorkerProcess(3)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61304] [MainThread] [dramatiq.WorkerProcess(4)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61305] [MainThread] [dramatiq.WorkerProcess(5)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61306] [MainThread] [dramatiq.WorkerProcess(6)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61307] [MainThread] [dramatiq.WorkerProcess(7)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61308] [MainThread] [dramatiq.WorkerProcess(8)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,182] [PID 61309] [MainThread] [dramatiq.WorkerProcess(9)] [INFO] Worker process is ready for action.
[2024-05-18 14:50:15,270] [PID 61310] [MainThread] [dramatiq.ForkProcess(0)] [INFO] Fork process 'dramatiq.middleware.prometheu
is ready for action.
```

Крок 4. Запустіть моніторинг черг, для запуску виконайте наступну команду під активованим віртуальним середовищем Python:

python dashboard.py

Моніторинг має бути доступний за налаштованими адресом та портом, як вказано в Кроці 2:



ДОДАТОК Д

Інструкція адміністратора та користувача Odoo щодо оновлень в екранних формах модулів «Склад» та «Контакти» та функціях системи.

Повний код оновленого модуля для Odoo знаходиться у репозиторії за посиланням: https://github.com/igorgoroun/3pl_stock.

В екранній формі налаштувань контрагента додано нові поля для вказання головного розташування товарів контрагента, та поля для вказання даних для аутентифікації інформаційної системи контрагента для взаємодії через підсистему інтеграції:

Скріншот екранної форми налаштувань контрагента в Odoo. Форма називається «ФОП Єрмоленко К.Ю.».

Вибрано тип контрагента: **Компанія**.

Основні дані:

- Address:** Дунаєва 27, Миколаїв, Україна, Статус: 54001
- ІПН:** 090928283
- Main Location:** MAIN/ЄрмоленкоКЮ
- Телефон:** (поле порожнє)
- Мобільний:** (поле порожнє)
- Ел. пошта:** igor.goroun@icloud.com
- Веб-сайт:** наприклад, https://www.odoo.com
- Мова:** Ukrainian / українська
- Мітки:** напр. "B2B", "VIP", "Консалтинг", ...

Вкладка «3PL API Configuration»:

- API Connected:** ☒
- API Login:** johndoe
- API Key:** [маскований]

Нижче форми є кнопки: «Надіслати повідомлення», «Примітка», «Дії».

Дата: 1 квітня 2024 р.

Користувач: Igor Horun (Бухгалтер (Продавець))

На екранну форму товару додано інформацію про належність до контрагента, значення цього поля встановлюється автоматично у випадку створення через підсистему інтеграції:

Назва товару
☆ Gigabyte M1280 RL UK

☒ Можна продати ☒ Можна купити

Загальна інформація Купівля Склад Бухоблік

Тип товару? Товар, що зберігається
Товари для зберігання – це фізичні предмети, для яких ви керуєте рівнем запасів.

Належить до ФОП Єрмоленко К.Ю.

Продажна ціна? € 4 560,65

Податки клієнта? Реалізація звільнена від ПДВ x

Вартість? € 4 560,65

Категорія товару All

Внутрішні посилання M1280

Штрих-код 83204301023929

Теги шаблону товару
→ Configure tags

ВНУТРІШНІ ПРИМІТКИ

Материнська плата Gigabyte M1280 RL UK

Надіслати повідомлення Примітка Дії

Також додано нове пошукове поле по товарах – контрагент:

Новий Товари

Пошук: Товари x Контрагент ФОП Єрмоленко К.Ю. x Пошук...

Назва товару	Внутрішні посилання	Теги шаблону товару	Продажна ціна	Вартість	В наявності	Прогнозований
ASUS 45332	AS45332		3 467,00 €	3 467,00 €	0,00	0,00
Gigabyte M1280 RL	M1280		4 560,65 €	4 560,65 €	0,00	0,00
Назва останнього товару	ART343454		32,65 €	32,65 €	0,00	0,00

На екранну форму складської операції додано поле з вказанням унікального ідентифікатора завдання, якщо операція була створена через підсистему інтеграції запитом від контрагента. Також при створенні операції завантаження вручну, система автоматично підставляє розташування призначення, прив'язане до контрагента та власника товарів, які мають бути отримані.

☆ MAIN/IN/00012

Отримати з: ФОП Єрмоленко К.Ю. Запланована дата: 21.05.2024 03:00:00

Representative Name: Коваленко А.В. Початковий документ: 202405/32

Representative Contact Info: +380670000000 Призначити власника: ФОП Єрмоленко К.Ю.

Тип операції: Головний: Надходження API Job ID: e926273e-b385-4935-a2b7-44e48d9ec80a

Розташування призначення: MAIN/ЄрмоленкоКЮ

Операції Додаткова інформація Примітка

Товар	Попит
[M1280] Gigabyte M1280 RL	43,00
[AS45332] ASUS 45332	24,00

Додати рядок

Надіслати повідомлення Примітка Дії

На екранну форму переліку складських операцій додано дію «Виставити рахунок»:

Новий Надходження 3 вибраний Відмінити резервування Перевірте наявність Друк Дії 1-4 / 4

	Референс	Від	До	Контакт	Запланована дата	Статус
☒	☆ MAIN/OUT/00002	MAIN/КарбовськийЮВ	Partners/Customers	ФОП Карбовський Ю.В.		Виконано
☒	☆ MAIN/OUT/00003	MAIN/КарбовськийЮВ	Partners/Customers	ФОП Карбовський Ю.В.		Виконано
☒	☆ MAIN/IN/00004	Partners/Vendors	MAIN/Store	ФОП Карбовський Ю.В.		Виконано
☐	☆ MAIN/OUT/00005	MAIN/Store	Partners/Customers	ФОП Карбовський Ю.В.		Скасовано

- Експорт
- Дублювати
- Видалити
- Надіслати електронний лист
- Підтвердити
- Відмінити резервування
- Виставити рахунок

Після виконання дії «Виставити рахунок» система автоматично на підставі обраних складських операцій розрахує кількість наданих послуг, та створить новий рахунок, доступний в модулі «Виставлення рахунків»:

Виставлення рахунків Клієнти Постачальники Звітність Налаштування

Новий Рахунки /2

1/1

Надіслати та друкувати Зареєструвати платіж Попередній перегляд Сторно Зробити чернеткою Чернетка Опубліковано

Ви маєте залишкові кредити для цього покупця. Ви можете розділити їх, щоб позначити цей рахунок як сплачений.

Рахунок клієнта
/2

Клієнт: ФОП Карбовський Ю.В.
вулиця Садова, будинок 4
Миколаїв 54055
Україна – 34234214

Дата рахунку: 18.05.2024

Референс оплати: /2

Терміни оплати: Кінець поточного місяця

Адреса доставки: ФОП Карбовський Ю.В.

Валюта: UAH

Рядки рахунку Інша інформація

Товар	Мітка	Кількість	Ціна	Податки	Без ПДВ
[MOVEITEM] Переміщення одиниці това	[MOVEITEM] Переміщення одиниці товару	356,00	2,00	Реалізація звільнена від ПДВ	712,00 ₴
[MOVEPARCEL] Переміщення парсела	[MOVEPARCEL] Переміщення парсела	54,00	10,00	Реалізація звільнена від ПДВ	540,00 ₴

Терміни та умови

Сума без податків: **1 252,00 ₴**
ПДВ 20%: 0,00 ₴
Разом: **1 252,00 ₴**