

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри

\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

Система автоматизації роботи торговельних представників "Мобільні  
продажі"

---

---

---

Освітньо-кваліфікаційний рівень “бакалавр”  
Напрямок підготовки 6.050102 – “комп’ютерна інженерія”

Керівник проекту:

\_\_\_\_\_ (підпис)

Скарга-Бандурова І.С.

\_\_\_\_\_ (ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_ (підпис)

Критська Я.О.

\_\_\_\_\_ (ініціали, прізвище)

Студент:

\_\_\_\_\_ (підпис)

Приймак С.О.

\_\_\_\_\_ (ініціали, прізвище)

Група:

КІ-14з

---

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний  
рівень бакалавр  
Напрямок підготовки 6.050102 – “комп'ютерна інженерія”  
(шифр і назва)  
Спеціальність \_\_\_\_\_  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
I.C. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Приймаку Сергію Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Система автоматизації роботи торговельних  
представників "Мобільні продажі"

керівник проекту  
(роботи) Скарга-Бандурова І.С., д.т.н., проф.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "15" 05 2018 р. № \_\_\_\_\_

2. Термін подання студентом роботи 17.06.2018

3. Вихідні дані до  
роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз систем мобільних продажів та формулювання технічного завдання; проектування системи; інтерфейс користувача; охорона праці та безпека в надзвичайних ситуаціях

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

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	ст. викл. Критська Я.О.		

7. Дата видачі завдання 15.05.2018

Керівник

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Ознайомлення з предметною галуззю	03.05 – 15.05	
2	Аналіз задачі та постановка проблеми	16.05 – 18.05	
3	Розгляд існуючих рішень щодо вирішення задачі.	19.05 – 25.05	
4	Розробка інтерфейсу.	26.05 – 03.06	
5	Розробка програмного коду і тестування	04.06 – 10.06	
6	Розробка розділу «Охорона праці та безпека в надзвичайних ситуаціях»	04.06 – 08.06	
7	Оформлення пояснювальної записки	09.06 – 15.06	

Студент

\_\_\_\_\_ ( підпис )

Керівник

\_\_\_\_\_ ( підпис )

Приймак С.О.

\_\_\_\_\_ (прізвище та ініціали)

Скарга-Бандурова І.С.

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Дипломна робота бакалавра складається з вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

Об'єкт дослідження: побудова клієнт-серверної системи для автоматизації роботи торговельних представників за допомогою мобільних пристроїв.

Мета: розробка системи для збору заявок торговельними агентами та автоматизованої передачі даних для подальшої обробки операторами.

В процесі розробки коротко розглядаються та порівнюються існуючі рішення, описується архітектура та взаємодія із системою.

Робота містить 101 сторінку, 23 рисунки, 2 додатки, 38 джерел.

Ключові слова: SFA, мобільні продажі, Android, iOS

## ЗМІСТ

РЕФЕРАТ .....	3
<b>1 АНАЛІЗ СИСТЕМ МОБІЛЬНИХ ПРОДАЖІВ ТА ФОРМУЛЮВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ .....</b>	<b>11</b>
1.1 Огляд предметної області .....	11
1.2 Огляд існуючих рішень .....	16
1.2.1 SalesWorks Enterprise .....	16
1.2.2 1С: Мобільна торгівля .....	17
1.2.3 Microsoft Dynamics 365 for Sales .....	18
1.2.4 Salesforce .....	19
1.2.5 Zoho CRM.....	19
1.3 Формулювання завдання роботи.....	20
1.3.1 Назва розробки .....	20
1.3.2 Призначення розробки.....	21
1.3.3 Вимоги до функціональних характеристик .....	21
1.3.4 Вимоги до інтерфейсу.....	22
1.3.5 Вимоги до програмного забезпечення .....	23
1.3.6 Вимоги до апаратного забезпечення .....	24
1.4 Висновок до розділу 1 .....	24
<b>2 ПРОЕКТУВАННЯ СИСТЕМИ.....</b>	<b>25</b>
2.1 Архітектура системи .....	25
2.1.1 Компоненти системи.....	25
2.1.2 Основні дані системи.....	27
2.2 Взаємодія клієнтів з системою .....	32
2.2.1 Реєстрація користувача та додавання торгівельної організації.....	32
2.2.2 Додавання користувачів до організації.....	33
2.2.3 Додавання пристроїв до організації .....	33
2.2.4 Додавання користувача на мобільному пристрої .....	34

	5
2.3 Прикладний програмний інтерфейс системи.....	35
2.3.1 Автентифікація .....	36
2.3.2 Користувачі.....	39
2.3.3 Користувачі організації .....	41
2.3.4 Товари.....	43
2.3.5 Ціни.....	44
2.3.6 Адреси .....	46
2.3.7 Замовники .....	48
2.3.8 Заявки .....	49
2.3.9 Синхронізація заявок на мобільному пристрої .....	54
2.4 Середовище та інструменти розробки.....	54
2.4.1 Мови програмування та розмітки.....	54
2.4.2 Каркаси додатків .....	57
2.4.3 Сервер Web-додатків .....	58
2.4.4 База даних .....	59
2.4.5 Основні інструменти та IDE.....	60
Висновки до розділу 2 .....	62
3 ІНТЕРФЕЙС КОРИСТУВАЧА.....	63
3.1 Інтерфейс користувача мобільного додатку .....	63
3.1.1 Додавання користувача на мобільному пристрої .....	63
3.1.2 Головне меню та головний екран мобільного додатку .....	64
3.1.3 Створення нової та зміна існуючої заявки.....	65
3.1.4 Вибір замовника та додавання товару.....	66
3.2 Інтерфейс користувача веб-додатку .....	67
3.2.1 Реєстрація користувача та додавання торгівельної організації.....	67
3.2.2 Додавання користувачів до організації.....	71
3.2.3 Перегляд списку заявок .....	72
3.2.4 Створення нової заявки .....	73
Висновок до розділу 3 .....	74

4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	75
4.1	Аналіз стану умов праці.....	75
4.2	Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером.....	76
4.3	Заходи з охорони праці .....	77
4.3.1	Загальні заходи безпеки.....	77
4.3.2	Електробезпека .....	79
4.3.3	Розрахунок захисного заземлення .....	81
4.4	Заходи, що забезпечують виробничу санітарію та гігієну праці .....	83
4.4.1	Мікроклімат .....	83
4.4.2	Освітлення.....	83
4.5	Рекомендації щодо пожежної безпеки .....	86
	Висновки до розділу 4 .....	88
	ВИСНОВКИ.....	89
	СПИСОК ЛІТЕРАТУРИ .....	90
	ДОДАТОК А ЛІСТИНГ НАБОРУ ТЕСТІВ ДЛЯ ПІДСИСТЕМИ ЗАЯВОК .....	94
	ДОДАТОК Б ПРЕЗЕНТАЦІЯ .....	97

## ВСТУП

Завдяки поширенню мобільних пристроїв та постійного підвищення мобільними операторами якості покриття з доступом до мережі Internet, все більше як великих, так і дрібних компаній намагаються зробити свій бізнес доступним в мережі Internet, або оптимізувати роботу бізнесу завдяки мобільним технологіям. Однією із галузей де це дало можливість стрімко покращити ефективність, стала сфера дистрибуції товарів.

Рано чи пізно торговельні компанії стикаються з необхідністю формування дистрибутивної мережі для ефективного продажу товару. Це передбачає створення різних категорій “польових” співробітників - торгових представників, мерчендайзерів, супервайзерів, які за родом своєї діяльності більшу частину робочого часу повинні перебувати поза офісом (в торгових точках, на маршруті), але в той же час залишатися на зв'язку з офісом - скидати замовлення, отримувати нові завдання і т.п. Організація ефективної роботи таких працівників потребують значних зусиль по забезпеченню актуальною інформацією як мобільних працівників, так і працівників у офісі. Зазвичай торговельні організації автоматизують тільки бізнес-процеси прямих продажів і мерчендайзингу, так як на цих ділянках автоматизація на основі мобільних пристроїв дає найбільший вигравш в ефективності, і це якраз є та діяльність, в якій можна повністю розкрити потенціал сучасних мобільних технологій.

Клієнтська частина системи мобільної торгівлі базується на мобільних пристроях - смартфонах і планшетах. Наразі, завдяки тому що сучасні смартфони надають усі необхідні функції, мобільна частина не потребує інших спеціалізованих пристроїв. Якщо раніше для роботи торговельного представника організація купувала, або орендувала КПК чи комунікатор, а для синхронізації даних з офісною базою даних працівнику треба було з'явитися в офіс щоб під'єднати КПК до офісної мережі тому що невелика частина КПК мали GPRS-модем, то тепер в цьому вже немає потреби. Зараз



майже будь-який недорогий смартфон або планшет на базі операційної системи Android обладнано GPS-навігатором, модемом та/або модулем WiFi. Все частіше організації дозволяють, або навіть вимагають, щоб працівники використовували їх власні мобільні пристрої для роботи. Це, хоча і дещо збільшує ризики, зв'язані з безпекою ведення бізнесу, але значно знижує вартість впровадження мобільного продажу та обслуговування парку мобільних пристроїв для цього.

Обов'язковий компонент будь-якої системи мобільної торгівлі - автоматизована служба зв'язку та синхронізації даних, що працює як в автоматичному режимі, так і за окремим запитом користувача (наприклад, оперативне оновлення інформації про складські запаси для формування замовлення на доставку певної групи товарів або вивантаження даних в кінці робочого дня в офісну облікову систему).

При впровадженні систем по автоматизації торгівлі (SFA), компанія-замовник бере на себе весь вантаж по налаштуванню системи. Найчастіше цей процес виявиться набагато ефективніше, якщо його здійснити силами фахівців постачальника системи. Якщо забути про конкретні речі, таких як адаптація під індивідуальні бізнес-процеси, то впровадження складається з одних і тих же етапів. Перший - попереднє обстеження бізнес-процесів підприємства (на місці або віддалено), за результатами якого йому можуть бути надані рекомендації, які ділянки діяльності слід автоматизувати. Далі - розробка проекту інтеграції системи мобільної торгівлі в інфраструктуру компанії. Тут крім плану технічного переозброєння будуть враховані всі вже діючі у замовника системи та запропонований варіант їх модернізації і використання в новій автоматизованій системі. Третій етап - впровадження на місці. Сюди входять як установка і настройка техніки, так і навчання користувачів і запуск рішення в промислову експлуатацію. І четвертий етап - передача системи на підтримку замовнику, якщо здійснювати допомогу користувачам компанія-замовник буде самостійно. Така схема добре зарекомендувала себе для великих дистриб'юторів, але не завжди прийнятна

для дрібних торговельних організацій. Більшість систем на ринку пропонують повну автоматизацію всіх аспектів мобільних продажів. Але не завжди всі ці можливості потрібні. Наприклад для дрібних торговельні організації, які тільки починають роботу в сфері мобільної дистрибуції і прагнуть автоматизувати, наприклад, тільки збір заявок, такі рішення будуть занадто дорогими.

Таким чином, метою даного дипломного проекту є розробка системи мобільного продажу для дрібних торгових організацій для збору заявок та автоматизованої їх передачі для подальшої обробки.

**Перелік умовних позначок та скорочень**

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
B2B	Business-to-Business
B2C	Business-to-Customer
CRM	Customer Relationship Management
CRUD	Create/Retrieve/Update/Delete
CSS	Cascading Style Sheets
DSA	Digital Signature Algorithm
ERP	Enterprise Resource Planning
IANA	Internet Assigned Numbers Authority
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
LTS	Long Term Support
MVT	Model View Template
REST	Representational state transfer
SASS	Syntactically awesome style sheets
SDK	Software Development Toolkit
SMTP	Simple Mail Transfer Protocol
SFA	Sales Force Automation
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

# 1 АНАЛІЗ СИСТЕМ МОБІЛЬНИХ ПРОДАЖІВ ТА ФОРМУЛЮВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ

## 1.1 Огляд предметної області

Мобільні продажі або мобільна торгівля - підхід до організації бізнес-процесу роботи з торговими точками, що має на увазі високий ступінь автоматизації і активне використання смартфонів, планшетів і аналогічних мобільних пристроїв. На практиці так само часто називають саме програмно-апаратне рішення для організації мобільного торгівлі.

Існують відмінності, що відокремлюють рішення для мобільної торгівлі в окремий клас. Рішення для мобільних продажів призначені, в першу чергу, для автоматизації роботи мерчандайзерів та торговельних представників, тобто експлуатуються переважно в сегменті гуртової торгівлі моделі business-to-business (B2B). Коли ж мова заходить, наприклад, про надання можливості споживачу з зробити замовлення через мобільний пристрій, то мова йде вже не про мобільний продаж, а, скоріше, про мобільні покупки, тобто про сегмент роздрібною торгівлі моделі business-to-customer (B2C).

Термін “мобільний продаж” (мобільна торгівля) є усталеним і включає слово “мобільний” з історичних причин, оскільки в момент виникнення концепції мобільних продажів були широко поширені і використовувалися саме мобільні стільникові телефони, а не КПК або смартфони. Тому термін “мобільний продаж” іноді викликає помилкові асоціації з продажами мобільних телефонів або навіть з оперативністю і “мобільністю” самих продавців.

В даний час в системах мобільних продажів використовуються в основному стандартні комерційні смартфони і планшетні комп'ютери з інсталюваним додатковим ПО, що дозволяє отримувати дані по залишкам

товару на складі продавця, актуальними цінами, знижкам, список клієнтів, дебіторську заборгованість і відправляти сформовані замовлення через мобільну мережу безпосередньо в програму ведення бухгалтерського або управлінського обліку.

Програмні рішення класу “мобільні продажі” і “мобільна торгівля” для автоматизації роботи торгових представників, мерчандайзерів, експедиторів та інших категорій “роз’їзних” співробітників досить поширені. Цей напрямок активно розвивається і пов’язаний з іншими аналогічними широко відомими рішеннями по автоматизації торгівлі, такими як CRM. Розробкою зайняті провідні компанії-вендори, такі як Microsoft (на платформі Microsoft Dynamics) і багато інших. Ряд лідируючих на ринку вітчизняних і зарубіжних фірм займається активною розробкою програмних рішень для мобільних продажів. Функціональність доступних платформ для мобільних продажів швидко розширюється за рахунок нових доступних користувачам можливостей (статистика, передача візуальної інформації, анкетування, моніторинг, збір маркетингової інформації, навігація і т.п.).

При сучасному веденні бізнесу дистрибуції товарів, просто наявність в мобільному пристрої даних з центральної облікової системи, вже недостатньо щоб мати додаткові переваги над конкурентами і якісно підвищити рівень дистрибуції. Зараз все більше дистриб’юторів пред’являють додаткові вимоги до спеціалізованого програмного забезпечення для мобільних співробітників: розрахунок рекомендованого замовлення, інформування торгового агента про статус виконання плану продажів, пропозиції до вибору рекомендованих позицій товару в залежності від статусу торгової точки, підрахунок суми замовлень торгової точки і видача результату для надання додаткової знижки, надання мобільному агенту послідовність дій необхідних для виконання в точці продажу та ін. Вся ця додаткова функціональність дозволяє мінімізувати час витрачений мобільним співробітником у клієнта, в порівнянні з часом при роботі з паперовими носіями, а також без труднощів працювати з величезним

масивом інформації необхідної для успішних продажів. Додаткові можливості програмного забезпечення дозволяють перевести кількісну дистрибуцію в площину якісної дистрибуції, тим самим підвищивши конкурентність компанії дистриб'ютора і прив'язавши кінцевого клієнта (торгову точку) ні до мобільного співробітника, а до компанії яку він представляє. Це дозволяє компанії дистриб'ютору, не порушувати роботу між нею і торговою точкою при заміні співробітника.

Основні переваги мобільних продажів:

- оперативність, можливість працювати в реальному часу, скорочення числа візитів до головного офісу;
- скорочення часу на створення заявок в торговій точці, за рахунок зручної роботи з великою кількістю товарів;
- збільшення кількості відвідуваних точок в день (за рахунок економії часу на прийом заявки і повного звільнення від диктування замовлення по телефону);
- отримання всіх даних з головного офісу для ведення успішних продажів (залишки, ціни, історія продажів по торговій точці, дебіторська заборгованість, рекомендовані замовлення, план продажів, інформації про клієнта, знижки клієнта та ін.);
- усунення в замовленні порожніх позицій товарів (тобто позицій товару якого на даний момент немає на складі), а заміна його на схожі що є в наявності;
- підвищення персональної відповідальності торгових представників і мерчандайзерів, оперативний контроль за їх роботою, зниження помилок і підвищення продуктивності праці;
- можливість відстежувати переміщення співробітників на карті (при наявності GPS-сенсора в мобільному пристрої і підтримки даної функції програмою);
- можливість використовувати звичайні недорогі смартфони або планшети на базі ОС Android;

- мінімізація паперового документообігу і скорочення витрат на операторів з прийому замовлень по телефону
- оптимізація складських залишків дистриб'юторських компаній.

### Життєвий цикл заявки

Життєвий цикл заявки складається з наступних етапів (рис.1.1).

1. Створення чернетки заявки
2. Передача заявки для обробки оператором
3. Обробка заявки
4. Виконання заявки
5. Заявку виконано або скасовано

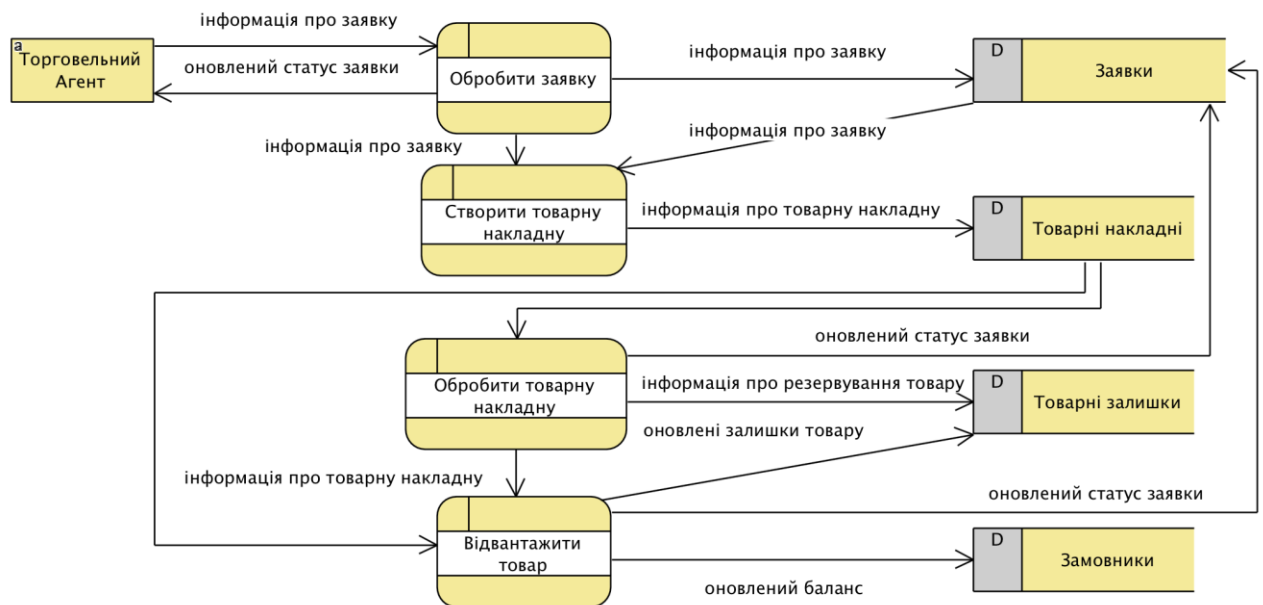


Рисунок 1.1 – Діаграма обміну даних для життєвого циклу заявки

На першому етапі торговельний представник під час відвідування торговельної точки створює на своєму пристрої чернетку заявки. Чернетка може бути майже повністю або цілком готовою заявкою з тією лише різницею, що торговельний представник ще може зробити зміни до заявки або скасувати її. Чернетки ніколи не покидають мобільного пристрою торговельного представника.

Коли торговельний представник впевнений, що заявка в повному обсязі містить всю правдиву та необхідну інформацію, він помічає заявку як

таку що готова до обробки. При наступному сеансі синхронізації даних такі заявки відправляються на сервер для подальшої обробки операторами. Заявки, що очікують обробки більше не можуть бути змінені торговельним представником, навіть якщо вони ще не були відправлені на сервер. При передачі заявок на сервері створюються відповідні записи у базі даних і заявкам присвоюється унікальні ідентифікатори, які передаються на пристрій торговельного представника для подальшого відстеження стану заявок.

Після того, як заявку було передано на сервер, оператор перевіряє інформацію в ній. Якщо торговельна організація вимагає від замовника передплати, а на балансі замовника недостатньо коштів, то оператор переводить заявку у стан “очікується оплата”. Якщо коштів на балансі замовника достатньо, або можлива післяплата, та у оператора нема зауважень, то баланс замовника корегується, заявка передається на виконання, а її стан змінюється на “очікується відправлення”. Крім того коли заявка є у стані “готова до обробки”, то оператор може скасувати заявку, якщо на те є причини.

Після того як заявку було передано на виконання, оператор складу перевіряє наявність усіх товарних позицій, та якщо заявку можливо виконати, створює на підставі заявки товарну накладну, та відвантажує товар замовнику. При цьому заявка переводиться у стан “очікується отримання”. Якщо деяких позицій немає в наявності, то оператор складу в свою чергу може замовити ці позиції у постачальників, та повернутися до обробки заявки пізніше.

Після того, як товар було відвантажено та представник замовника підтвердив отримання товару підписавши необхідні документи, та отриманий товар було повністю сплачено раніше, то заявка переводиться у стан “виконано”. Якщо ж замовник ще має сплатити за отриманий товар, то заявка переводиться у стан “очікується оплата”.



Після того як товар було отримано замовником та за нього було повністю сплачено, або заявка було скасовано - життєвий цикл заявки завершується і жодні зміни до неї не припустимі.

## **1.2 Огляд існуючих рішень**

Розглянемо найбільш відомі рішення з тих, що представлені на ринку в даній області, а саме – SalesWorks Enterprise та 1С: Мобільна торгівля.

### **1.2.1 SalesWorks Enterprise**

SalesWorks - це трирівнева система (виробник / імпортер - дистриб'ютор - торговий агент). Є стандартом де-факто систем управління продажами в Україні. Це корпоративне рішення для великих торгово-виробничих компаній, регіональних та національних мереж розподілу та імпортерів. Система спрямована на автоматизацію бухгалтерського обліку та аналізу даних про торговельну діяльність дистриб'юторів (філій), а також всіх основних видів роздрібного розподілу (пресування, фургони та мерчендайзинг).

Кожному рівню відповідає окремий модуль системи:

- Виробник: Центральний модуль, центральна БД (ЦБД)
- Дистриб'ютор: Настільний модуль, настільна БД (НБД)
- Торговий представник: Мобільний модуль, мобільна БД (МБД)
- Основні характеристики та переваги:
  - обробка запису клієнта;
  - історія замовлень, обсяг, асортимент товарів, замовлених кожним клієнтом;
  - моніторинг маршрутів;
  - розробка та оптимізація маршрутів, що проводяться торговими агентами, включаючи планування частоти відвідувань клієнтів, термін служби, послідовність руху та інші правила;
  - ведення обліку агентів;

- використання інструментів керування з відвідуванням торгових точок торгового агента;
- формування планів продажів для кожного агента у фінансовій та фізичній формі;
- регулювання маршрутів подорожей;
- обсяг замовлень;
- своєчасний обмін інформацією про стан запасів;
- бронювання товарів на складі в режимі реального часу;
- взяття GPS-координат відвідування торгової точки;
- "зняття" залишків;
- проведення опитування.

Це забезпечує єдині корпоративні стандарти та правила ведення бізнесу у всій мережі дистрибуції.

Недоліки: рішення переважно орієнтоване на великих корпоративних замовників, що мають широку мережу дистриб'юторів. Досить коштовне рішення, що вимагає адаптації бізнес-процесів, закупівлі обладнання і програмного забезпечення для впровадження у дистриб'юторів.

### **1.2.2 1С: Мобільна торгівля**

Продукт "1С: Мобільна торгівля" призначається для автоматизації мобільної торгівлі за допомогою мобільних пристроїв (смартфонів, планшетних комп'ютерів) під управлінням операційних систем Android і iOS. Продукт дозволяє автоматизувати різні види діяльності торгових представників оптових і дистриб'юторських компаній, що використовують облікові системи на платформі "1С: Підприємство 8", такі як "1С: Управління торгівлею ред. 11.2", "1С: ERP Управління підприємством 2".

"1С: Мобільна торгівля" не вимагає установки додаткових модулів. Для обміну даними використовується стандартна обробка "Налаштування синхронізації пристроїв з встановленим мобільним додатком" 1С: Замовлення".

Переваги: зріле рішення, активно розвивається, практично безшовна інтеграція з існуючою інфраструктурою 1С, відносно недорога вартість впровадження при використанні інших продуктів 1С для обліку та управління торгівлею.

Основні характеристики та переваги:

- робота по маршруту;
- збір та обробка замовлень на поставку товарів;
- робота з договорами;
- можливість роботи с декількома складами;
- облік касових операцій;
- робота з типами цін і зі знижками;
- текстовий пошук товарів та клієнтів;
- створення клієнтів і товарів;
- фільтрація списку документів за різними критеріями;
- перегляд в мобільному пристрої розташування торгових точок на географічних картах;
- можливість вибіркового обміну даними.

Недоліки: офіційно не представлений в Україні, для повноцінної роботи вимагає інтеграції з іншими продуктами 1С.

### **1.2.3 Microsoft Dynamics 365 for Sales**

Microsoft Dynamics 365 - це продуктова лінійка програм корпоративного планування (ERP) та управління клієнтськими відносинами (CRM), оголошена корпорацією Майкрософт в рамках лінійки продуктів Microsoft Dynamic.[11] Dynamics 365 реалізується у двох виданнях: Business Edition для малих та середніх підприємств (МСБ або SMB) та Enterprise Edition для середніх та великих організацій

Недоліки: для повноцінної роботи мобільного додатку потрібне постійне з'єднання з мережею Internet, висока ціна, інтерфейс користувача

тільки англійською мовою; систему спроектовано під робочі процеси відмінні від прийнятих в Україні.

#### **1.2.4 Salesforce**

Salesforce - це основний продукт, який пропонує платформа американської компанії Salesforce.com, Inc[28]. Він надає торговельним компаніям інтерфейс для управління справами та управління завданнями, а також для автоматичної маршрутизації та ескалації важливих подій. Портал клієнтів Salesforce надає своїм клієнтам можливість відстежувати власні ситуаційні задачі, надає аналітичні інструменти та інші послуги пов'язаних з керуванням прав і контрактів.

Недоліки: для повноцінної роботи мобільного додатку потрібне постійне з'єднання з мережею Internet, висока ціна, інтерфейс користувача тільки англійською мовою; систему спроектовано під робочі процеси відмінні від прийнятих в Україні.

#### **1.2.5 Zoho CRM**

Zoho CRM — це система управління відносинами з покупцями (англ. customer relationship management). Відтак, тут є базові функції CRM, з поставками, інвентаризацією, і певними можливостями виставлення рахунків, що більше лежить в області ERP (англ. Enterprise Resource Planning). Zoho CRM є вільною, якщо ви вкладаєтеся в обмеження трьох користувачів, і коштує \$12 на місяць за кожного додаткового користувача. Також існує Enterprise Version (версія масштабу підприємства) Zoho CRM з місячною ціною \$25 за кожного користувача.[38]

Недоліки: офіційно не представлений в Україні, інтерфейс користувача тільки англійською мовою; систему спроектовано під робочі процеси відмінні від прийнятих в Україні.

По результатам проведеного аналізу (табл. 1.1) можна зробити висновок, про необхідність розробки системи, оскільки жодна з розглянутих систем не відповідає у повній мірі вимогам впровадження на дрібних торговельних організаціях в Україні.

Таблиця 1.1 – Порівняння існуючих рішень

Вимоги	Sales Works	1С:Мобіль на торгівля	Microsoft Dynamics 365 for Sales	Sales Force	Zoho CRM
Підтримка української або російської мов	так	так	ні	ні	ні
Адаптованість під процеси прийняті в Україні	так	так	ні	ні	ні
Зберігання координат торговельної точки	так	ні	ні	ні	ні
Можливість роботи без підключення до мережі Internet	так	так	ні	ні	ні
Можливість розгорнути систему власними силами	ні	так	ні	ні	ні
Орієнтованість на малий бізнес	ні	так	ні	ні	ні

### 1.3 Формулювання завдання роботи

#### 1.3.1 Назва розробки

Система автоматизації роботи торговельних представників "Мобільні продажі".

### **1.3.2 Призначення розробки**

Призначенням розробки є створення додатку для мобільних пристроїв на базі ОС Android та web-додатку для забезпечення роботи мобільного додатку.

Систему має бути поділено на три модулі – модуль Web-API та адміністрування, мобільний модуль для збору заявок торговельними представниками та веб-модуль керування заявками.

### **1.3.3 Вимоги до функціональних характеристик**

Система має реалізувати основні сценарії (рис. 1.2) пов'язані із створенням та обробкою заявок торговельним агентом та оператором.

У мобільному модулі повинна бути реалізована можливість автентифікації торговельного агента за допомогою адреси електронної пошти, створювати заявки, переглядати список товарів та торговельних точок. При створенні заявок мають бути збережені географічні координати місця де востаннє було змінено заявку. Також мобільний додаток повинен періодично та автоматично синхронізувати дані з сервером, тобто завантажувати оновлені дані з сервера та відправлятися на сервер створені заявки для подальшої обробки. Обсяг даних, що передаються має бути мінімізовано.

У настільному модулі має бути можливість автентифікації користувачів за допомогою електронної адреси, створювати та змінювати заявки, переглядати список товарів та торговельних точок.

Модуль адміністрування має надавати можливість автентифікації адміністраторів системи, керувати обліковими записами користувачів системи, переглядати та змінювати дані будь-яких користувачів.

Модуль Web-API має надавати всі необхідні дані для роботи мобільного та настільного модулів.

Усі дані системи повинні зберігатися в структурованому вигляді під управлінням реляційної СУБД.

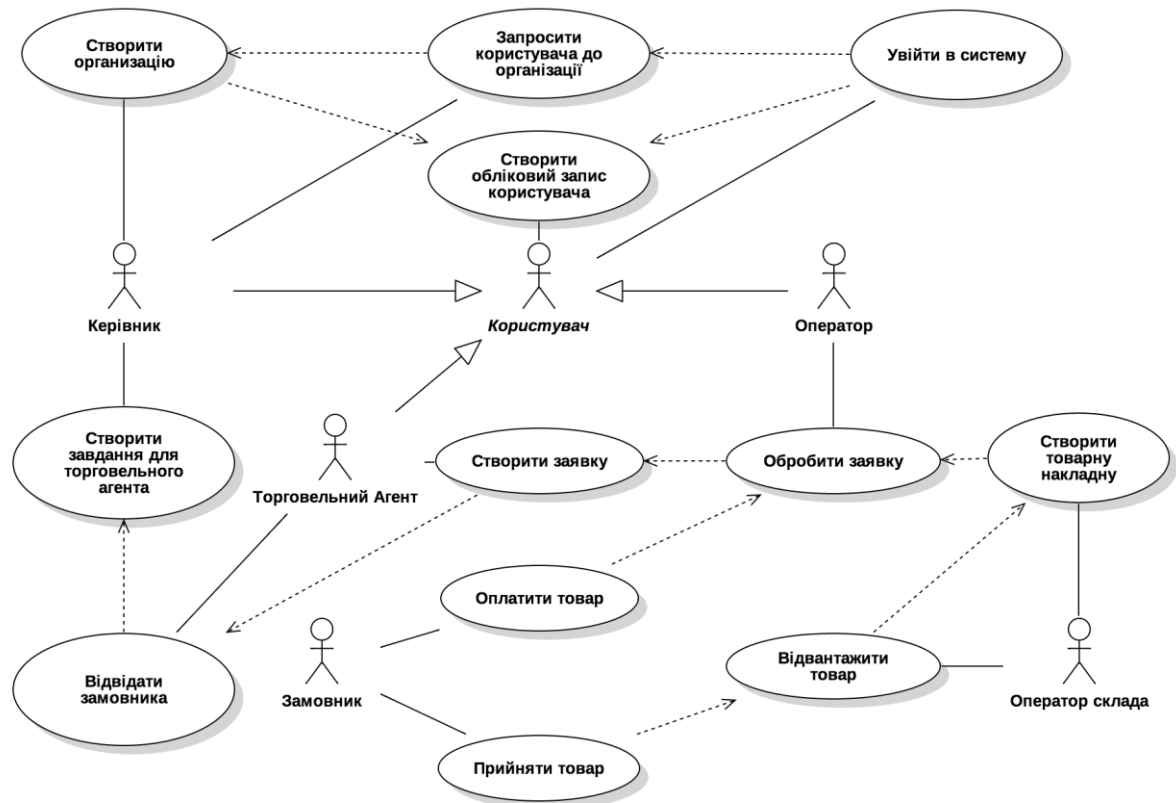


Рисунок 1.2 – Діаграма основних прецедентів системи

Систему має бути розроблено таким чином щоб уможливити подальше розширення функцій та спростити внесення змін у існуючі функції.

Для забезпечення безпеки доступ до системи повинен здійснюватися через захищений протокол HTTPS.

### 1.3.4 Вимоги до інтерфейсу

Інтерфейс користувача системи повинен забезпечувати інтуїтивно зрозуміле уявлення про структуру, розміщену на ньому інформацію, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи і функціональні кнопки повинні забезпечувати однозначне розуміння, умовні позначення повинні відповідати загальноприйнятим.

Інтерфейс користувача повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок.

Графічні елементи навігації повинні бути забезпечені альтернативним підписом. Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсах і відображати відповідну інформацію. Для навігації повинна використовуватися система контент-меню.

Меню має представляти собою текстовий блок (список гіперпосилань) в лівій колонці або у верхній частині сторінки (залежно від розроблюваного дизайну).

Для розділів, що містять підрозділи, має бути передбачено випадаюче підменю.

При виборі будь-якого з пунктів меню користувачем повинна завантажуватися відповідна йому інформаційна сторінка.

Сторінки всіх розділів ресурсу повинні формуватися програмним шляхом на підставі інформації з бази даних на сервері.

Для користувачів має бути реалізована контактна форма для зв'язку з власником ресурсу..

### **1.3.5 Вимоги до програмного забезпечення**

Настільний модуль системи повинен бути розрахований для роботи в сучасних версіях наступних браузерів: Chrome, Firefox та Edge під керівництвом ОС Windows, Mac OS та Linux (крім Edge). Мобільний модуль має працювати на мобільних пристроях під керівництвом ОС Android версії не нижче ніж 5.0 Lolіpor. Модуль Web-API та адміністрування повинен бути розрахований для роботи під керуванням операційної системи Ubuntu Linux версії 14.04 або пізнішої LTS-версії.



### 1.3.6 Вимоги до апаратного забезпечення

Мінімальні системні вимоги для запуску програмного забезпечення на Web-сервері:

- процесор - з тактовою частотою не менш ніж 2ГГц та кількістю ядер не менш ніж 2;
- оперативна пам'ять – не менш ніж 512МБ;
- дискова пам'ять – не менш ніж 20ГБ.

Мінімальні системні вимоги для запуску програмного забезпечення на мобільному пристрої:

- процесор ARM Cortex-A8 з частотою не менш ніж 600 МГц;
- оперативна пам'ять - не менше 1ГБ;
- наявність модулю GPS або тотожного модулю для гео-локації;
- наявність модему;
- розмір екрану – не менш ніж 4.7”.

### Висновки до розділу 1

У розділі було розглянуто предметну область та порівняно декілька існуючих рішень. Крім того було сформовано призначення системи та вимоги до її роботи.

## 2 ПРОЕКТУВАННЯ СИСТЕМИ

### 2.1 Архітектура системи

#### 2.1.1 Компоненти системи

Система складається з чотирьох основних компонентів (рис.2.1) – серверу додатка, мобільного клієнта, веб-клієнта та сервера бази даних. Центральну роль відіграє сервер додатку, який поєднує в собі сервер автентифікації та авторизації і сервер ресурсів. Для певних задач серверу додатку є потреба відправляти листи електронної пошти до адміністраторів або користувачів за допомогою поштового серверу, який виконує роль ретранслятора запитів на відправку листів до наступного поштового серверу.

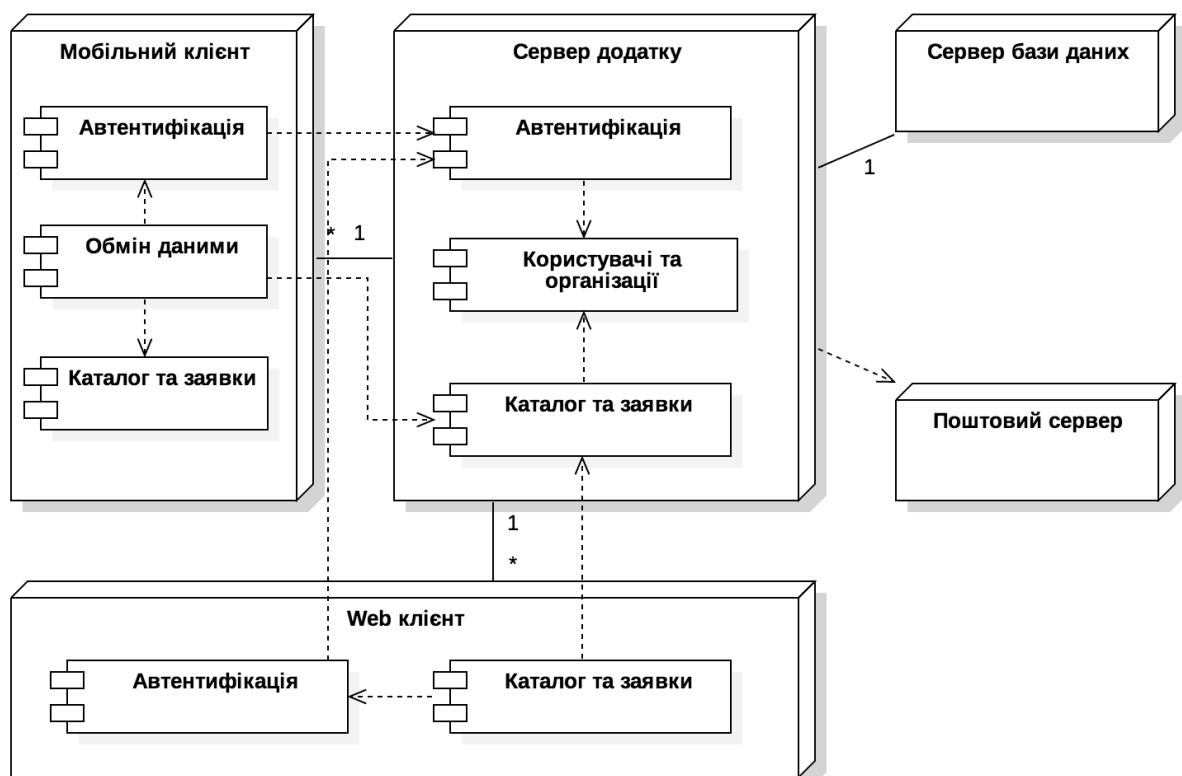


Рисунок 2.1 – Схема розгортання системи

Сервер додатку відповідає за обробку інформації та передачу її між клієнтами та сервером бази даних і містить підсистему автентифікації, підсистему керування користувачами та організаціями і підсистему каталогу та роботи з заявками і клієнтами. Основною є підсистема керування користувачами та організаціями, яка відповідає за створення, зміну і збереження таких даних як обліковий запис користувача, профілі торгівельної організації та її учасників, запрошення користувачів до системи тощо. Підсистема автентифікації залежить від підсистеми керування користувачами та відповідає за їх ідентифікацію і видачу повноважень на роботу з системою. Підсистема каталогу та заявок також залежить від підсистеми керування користувачами, але в більшій мірі відносно до профілів торгівельних організацій та їх учасників. Основна задача цієї підсистеми – це створення, зміна та збереження даних про товари і ціни, замовників і їх заявки. Крім функцій обробки та зберігання інформації всі підсистеми серверу додатку здатні через API забезпечувати доступ к цим даним користувачам системи за допомогою клієнтських додатків. Одним з таких додатків є веб-клієнт.

Веб клієнт відповідає за передачу запитів користувача серверу додатку ті відображення результатів у браузері і містить підсистему автентифікації та керування каталогом і заказами. На відміну від схожих підсистем серверу додатку ці підсистеми не є самостійними, а радше виконують роль посередника між користувачем та відповідними підсистемами на сервері, які транслюють дії користувача у запити до сервера.

Мобільний клієнт виконує ті ж функції що й мобільний додаток з тією лише різницею, що для забезпечення автономності роботи йому потрібна актуальна копія даних з серверу додатків, яка час від часу треба синхронізувати із сервером. За синхронізацію даних відповідає підсистема обміну даних. Таким чином робота підсистеми каталогу та заявок мобільного додатку транслюються вже не у запити до сервера, а до бази даних на пристрої, що містить копію даних актуальних на певний момент часу.

## 2.1.2 Основні дані системи

На рисунку 2.2 представлено фізичну схему фрагмента бази даних, що відповідає за зберігання облікових записів користувачів і профілів організацій та пов'язаних з ними об'єктів.

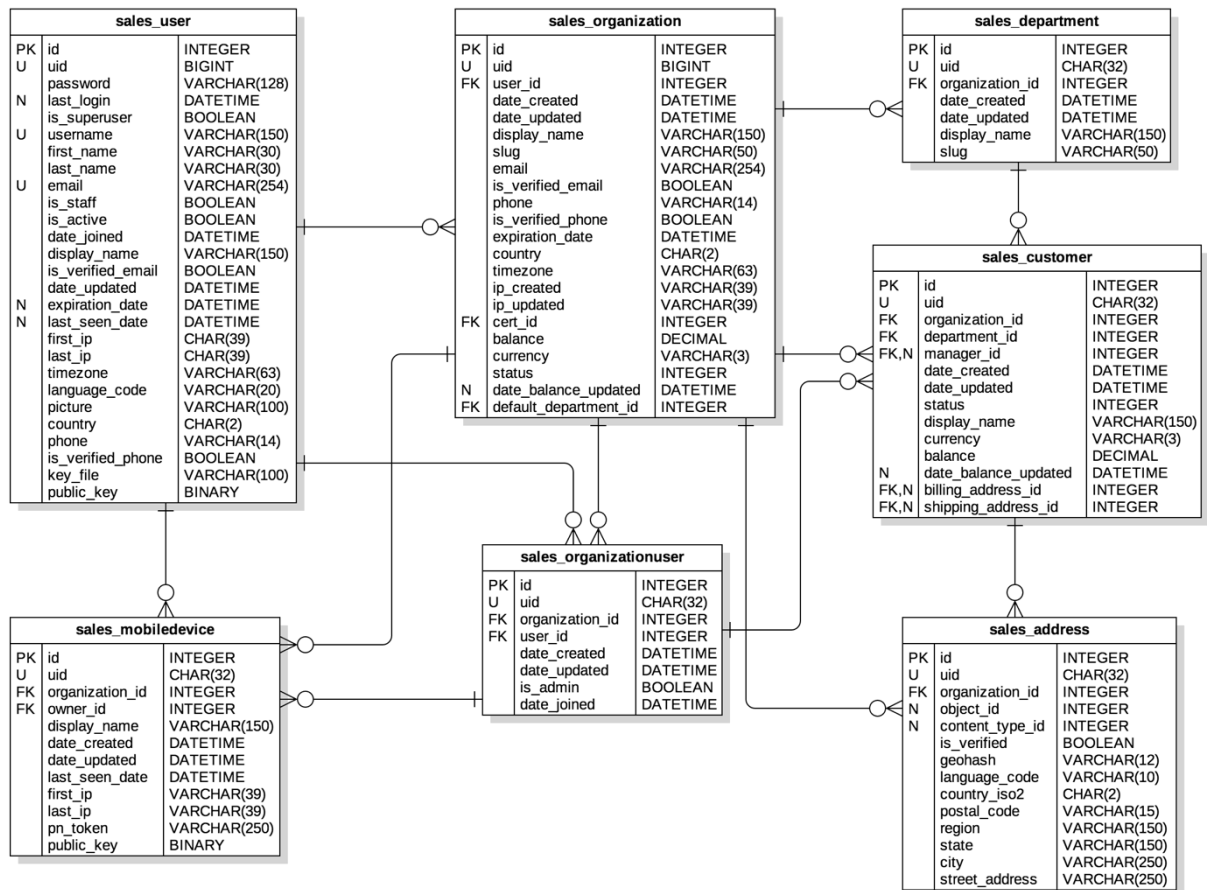


Рисунок 2.2 – Схема бази даних серверу додатка для основних об'єктів підсистеми автентифікації та керування організаціями

Таблиця `sales_user` – основна таблиця підсистеми. В цій таблиці зберігається вся інформація про обліковий запис користувача системи, а саме первинний ключ, унікальний ідентифікатор, адресу електронної пошти, ім'я та дані про пароль, дату реєстрації, криптографічні ключі та інші відомості з профілю.

Таблиця `sales_mobiledevice` містить дані про мобільні пристрої організації такі як посилання на організацію, поточного власника пристрою, назву, дату створення та оновлення запису тощо.

Таблиця `sales_organization` містить основні дані про торгову організацію, такі як посилання на користувача, що її зареєстрував, дату створення, назва організації, адресу електронної пошти та телефон, країну, валюту та часовий пояс, стан (активна чи неактивна) тощо.

Таблиця `sales_organizationuser` є таблицею, що забезпечує відношення М:М між таблицями `sales_organization` та `sales_user`.

Таблиця `sales_deptatment` містить дані про відділи організації (назва відділу та дата створення запису).

Таблиця `sales_customer` містить дані про замовників організації, такі як назва та унікальний ідентифікатор, дату створення та оновлення запису, стан (активний чи неактивний), поточний баланс замовника та посилання на користувача організації, що відповідає за роботу з замовником, а також посилання на юридичний адрес та адресу доставки.

Таблиця `sales_address` містить адресну книгу організації, в якій збережено дані про поштові адреси та їх географічні координати у вигляді хеш-коду[15].

Основні класи підсистеми каталогу і заявок та їх відношення зображено на рис. 2.3. Головним класом підсистеми, що володіє рештою об'єктів підсистеми є `Organization`. Він містить необхідні дані про торговельну організацію такі як найменування (`display_name`), унікальний ідентифікатор (`uid`), посилання на користувача, що зареєстрував організацію (`user`), адресу електронної пошти організації (`email`), чи та адреса є перевіреною (`is_verified_email`), телефон організації та ознаку чи він дійсно належить організації (`is_verified_phone`), двозначний код країни за стандартом ISO 3166-1 alpha-2 (`country_iso2`), часовий пояс організації згідно з базою даних часових поясів IANA [33], посилання на відділ за замовченням (`default_department`) тощо.

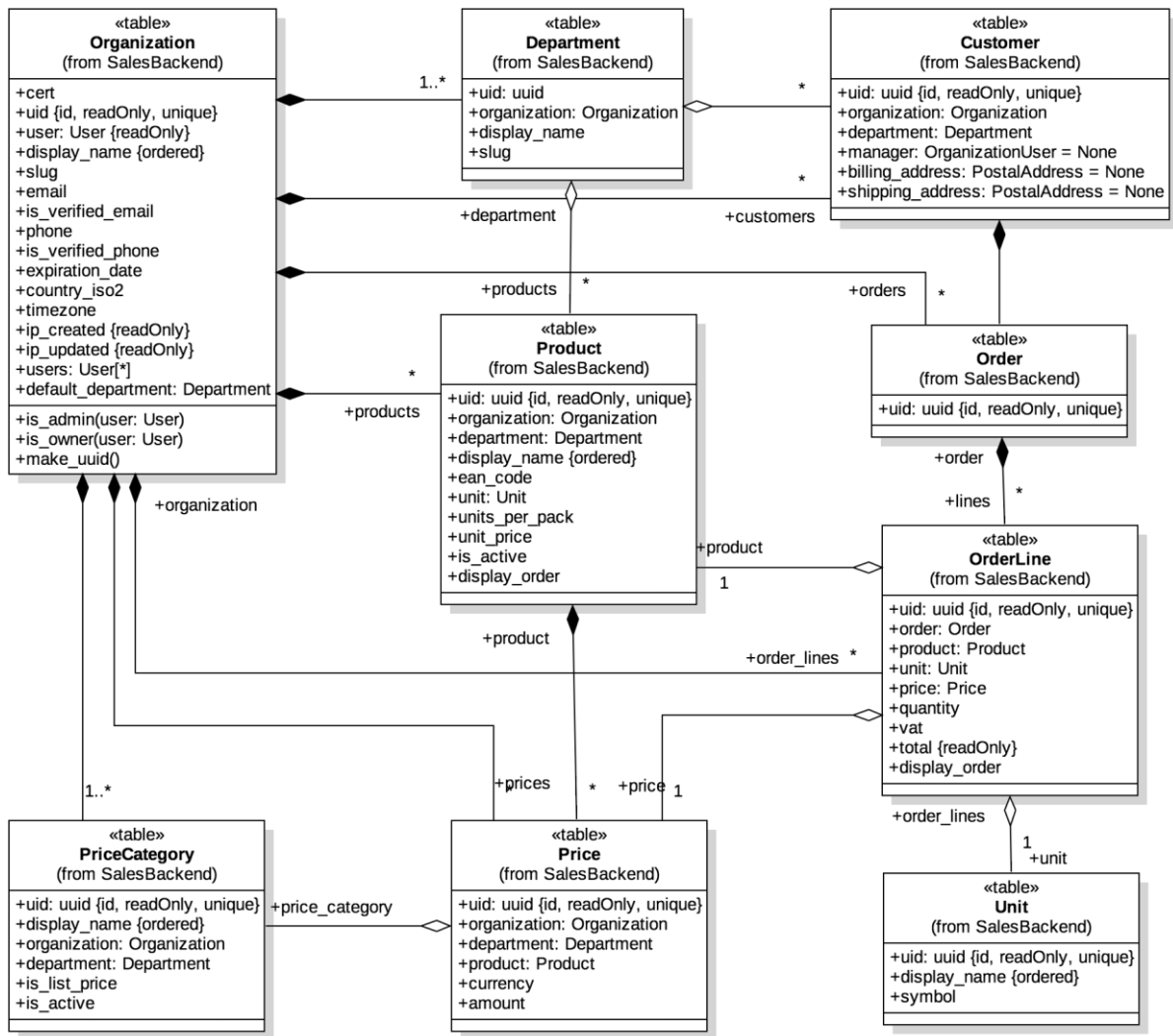


Рисунок 2.3 – Діаграма класів підсистеми керування заявками

PriceCategory – це клас, що відповідає за роботу з категоріями цін (оптова, дрібно-оптова, роздрібна тощо). Кожна організація має мати щонайменше одну категорію цін, одна з яких стає категорією за замовченням. Категорію за замовченням автоматично буде створено при реєстрації організації. Цей клас містить наступні дані: найменування категорії (`display_name`), посилання на організацію (`organization`) та відділ (`department`), чи є категорія цін активною (`is_active`) та чи відображається вона в прайс-листах (`is_list_price`). Категорія цін може бути специфічною до відділу чи загальною до всієї організації. В останньому випадку посилання на відділ приймає пусте значення (`department = null`).

Department – клас, що відповідає за відділи організації. Містить унікальний ідентифікатор (uid), посилання на організацію (organization) і назву відділу для відображення користувачу (display\_name) та у адресній строці (slug).

Клас Product відповідає за роботу з товаром і містить наступні дані: унікальний ідентифікатор (uid), посилання на організацію (organization) та відділ (department), найменування для відображення користувачу (display\_name), універсальний код продукту EAN чи UPC [12], посилання на одиницю виміру (unit), кількість одиниць в упаковці (units\_per\_pack), ціна одиниці товару (unit\_price) чи є товар активним (is\_active) та його порядок відносно інших товарів при зображенні у списках (display\_order). Кожна організація може мати будь-яку кількість товарів. Товар може бути специфічний до відділу або загальним до всієї організації. В останньому випадку посилання на відділ приймає пусте значення (department = null).

Клас Price відповідає за роботу з цінами і містить такі дані як унікальний ідентифікатор (uid), посилання на організацію (organization) та відділ (department), посилання на категорію цін (price\_category) та товар (product), трьох-символьний код валюти згідно з ISO 4217[19] та вартість товару. Кожна організація може мати будь-яку кількість цін.

Customer – це клас для роботи з замовниками. Він містить такі поля: унікальний ідентифікатор (uid), посилання на організацію (organization) та відділ (department), найменування для відображення користувачу (display\_name), необов'язкове посилання на учасника організації, який відповідає за роботу з замовником, посилання на юридичну адресу (billing\_address) та адресу доставки (delivery\_address). Замовник може бути належити до відділу або бути загальним до всієї організації. В останньому випадку посилання на відділ приймає пусте значення (department = null). Організація та відділ може мати будь-яку кількість замовників.

Order – це клас заголовка заявки. Він містить посилання на замовника (customer), організацію (organization) та відділ (department), посилання на

учасника організації, що створив заявку (owner), поточний стан заявки (status), вартість товару в заявці (amount), посилання на категорію цін (price\_category) та коментар (comment). Можливі значення стану заявки можна побачити на рис. 2.4.

OrderLine – це клас, що містить дані по кожній позиції замовлення, такі як: унікальний ідентифікатор (uid), посилання на заявку (order), посилання на товар (product), кількість товару (quantity), ціну товару (price), суму податку на додану вартість (vat), вартість товару (amount), порядок відображення відносно інших позицій (display\_order).

Unit – клас для одиниць виміру. Містить наступні дані: унікальний ідентифікатор (uid), посилання на організацію (organization), найменування для відображення користувачу (display\_name) та символ або скорочену назву одиниці виміру (symbol).

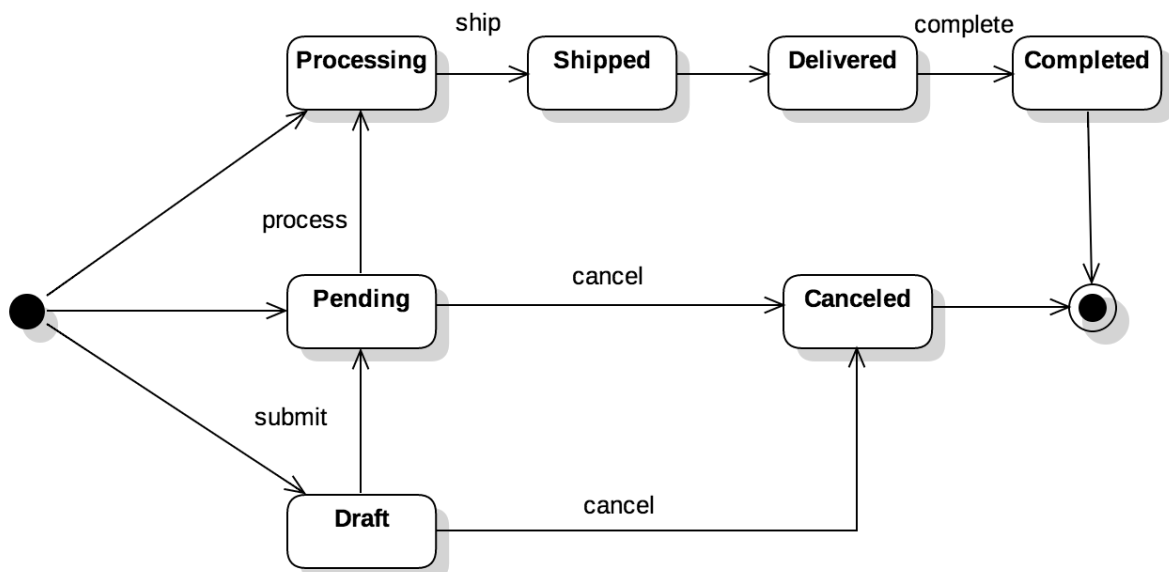


Рисунок 2.4 – Діаграма станів заявки

Кожна заявка може бути в одному з наступних станів:

- draft            чернетка заявки
- pending        нова заявка, очікується обробка



– processing заявка прийнята оператором для обробки. Подальші зміни торговельним агентом не припустимі.

– shipped заявку оброблено, на її підставі створено товарну накладну, товар відправлено замовнику

– delivered товар доставлено замовнику, очікується підтвердження від замовника

– complete замовником прийняв товар, заявку виконано. Подальші зміни не припустимі.

– canceled заявку скасовано. Подальші зміни не припустимі.

## 2.2 Взаємодія клієнтів з системою

### 2.2.1 Реєстрація користувача та додавання торговельної організації

Процес створення облікового запису користувача та організації зображено на рис. 2.5.

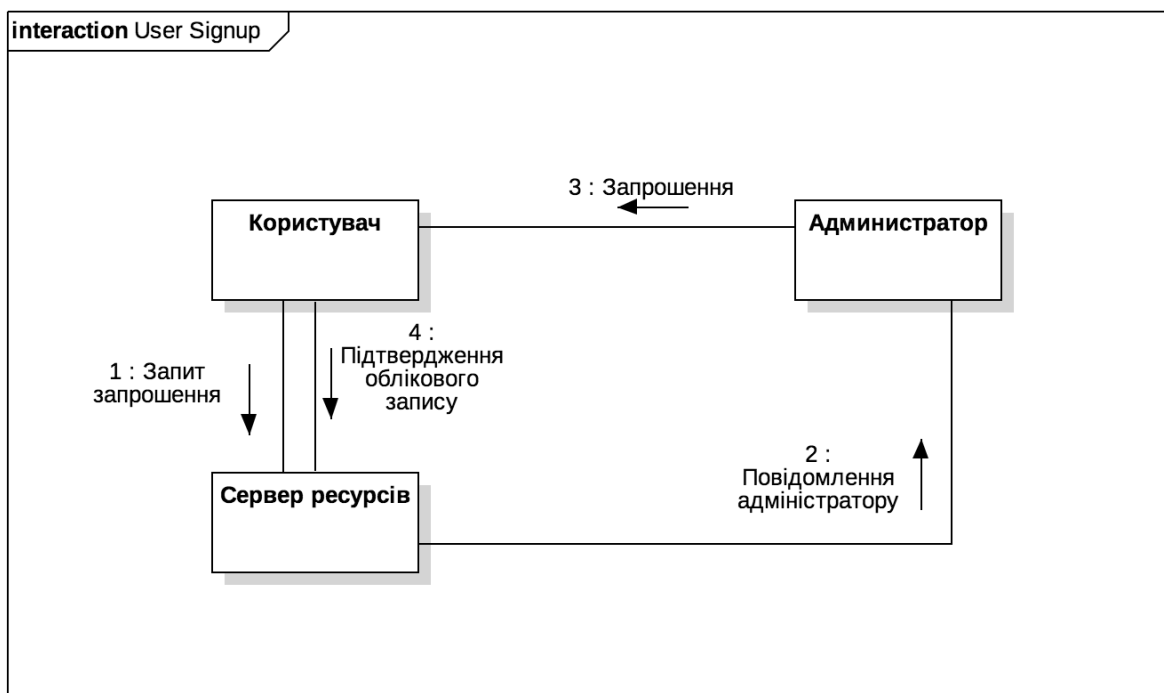


Рисунок 2.5 – Процес реєстрації користувача

Для створення нового облікового запису користувачу треба відправити запит на запрошення через форму на сайті. У запиті вказується ім'я та прізвище користувача, назву організації та інші контактні дані. Після створення запиту на адресу електронної пошти адміністратора сайту буде відправлено листа з повідомленням про новий запит на запрошення. Адміністратор сайту переглядає запит та приймає рішення чи схвалити запрошення чи скасувати запит. В обох випадках користувачу буде відправлено листа на адресу вказану в запиті.

Якщо адміністратор схвалив запит, то користувач отримає листа з посиланням на запрошення, де він зможе закінчити реєстрацію. Одночасно із створенням облікового запису користувача буде створено і організацію.

### **2.2.2 Додавання користувачів до організації**

Після того як було створено організацію, користувач, що її зареєстрував стає адміністратором цієї організації. Адміністратор має право додавати учасників до організації. Для цього треба на сторінці списку користувачів організації створити нове запрошення з адресом електронної пошти нового користувача. На цю адресу буде відправлене повідомлення з посиланням на форму реєстрації. На цьому етапі користувач може відмовитися від запрошення, або прийняти його. Якщо користувач прийняв запрошення а вказану адресу електронної пошти ще не зареєстровано в системі, то йому спочатку буде запропоновано створити обліковий запис. Автентифікованого користувача з існуючим обліковим записом буде автоматично додано до організації.

### **2.2.3 Додавання пристроїв до організації**

Для того щоб існуючому користувачу почати роботу із системою на мобільному пристрої, достатньо встановити додаток, та пройти

автентифікацію. Жодних спеціальних дій для активації пристрою в профілі організації на разі не потребується.

#### **2.2.4 Додавання користувача на мобільному пристрої**

Після того як користувач вперше запустив додаток на мобільному пристрої для пристрою створюється пара криптографічних ключів. Далі користувач повинен ввести дійсний адрес електронної пошти та пароль (крок 1 на рис. 2.6). Якщо введені дані відповідають існуючому активному обліковому запису, то сервер поверне дані, з якими надалі буде можливо відправляти запити до сервера від імені цього користувача. На цьому етапі автентифікації закінчено і починається ініціалізація пристрою для нового облікового запису (крок 3).

По перше у каталозі даних додатку створюється файл бази даних для цього облікового запису, по друге робиться запит на сервер для отримання списку організацій доступних користувачеві. Після цього запускається процес синхронізації. Перед початком першої синхронізації для кожної організації сервером випускається сертифікат для закритого ключа даного пристрою, яким надалі підписуються всі запити до сервера. Це дозволяє серверу впевнитись що дані було передано без пошкоджень та у разі необхідності припинити роботу з даним пристроєм відкликавши сертифікат пристрою для організації. Останнє може бути корисним якщо мобільний пристрій було втрачено або скомпрометовано. Після генерації сертифікату на сервер робиться серія запитів для отримання всіх даних необхідних для роботи додатку. Після того як було отримано дані для всіх організацій ініціалізація пристрою вважається завершеною.

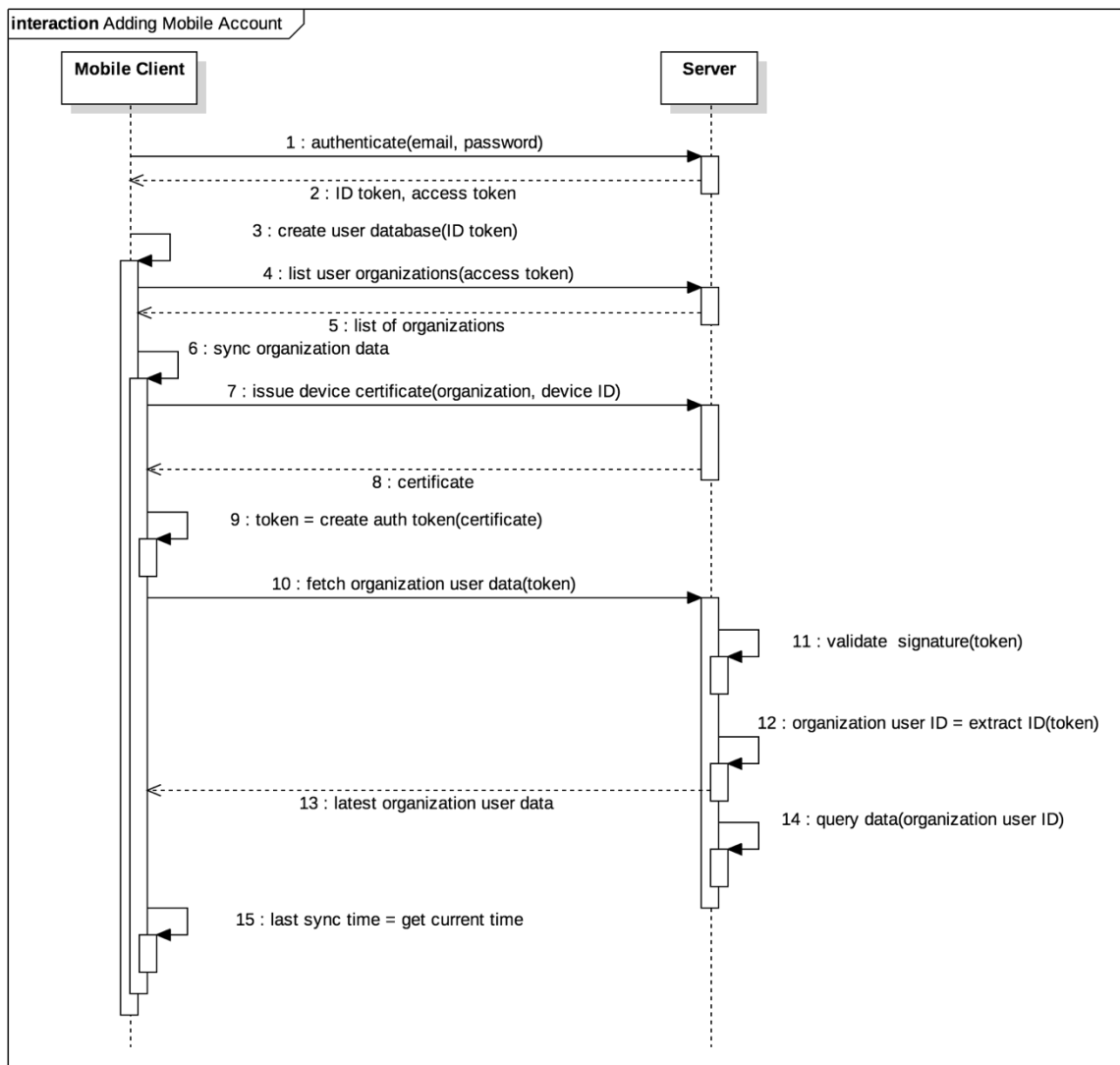


Рисунок 2.6 – Схема послідовності дій додавання облікового запису на мобільному пристрої

### 2.3 Прикладний програмний інтерфейс системи

Усі дані, що необхідні для роботи системи зберігаються та обробляються на сервері. Доступ к цим даним клієнтам організовано у вигляді ресурсів REST (Representational State Transfer)[25]. REST - архітектурний стиль, який визначає набір обмежень та властивостей на основі HTTP. Веб-сервіси, які відповідають архітектурному стилю REST або RESTful web services, забезпечують сумісність між комп'ютерними системами в Інтернеті. REST-сумісні веб-сервери дозволяють запитувачим

системам отримувати доступ до текстових уявлень веб-ресурсів і керувати ними, використовуючи єдиний та попередньо визначений набір операцій.

"Веб-ресурси" вперше були визначені в World Wide Web як документи або файли, визначені їх URL-адресами. Однак сьогодні вони мають набагато більш загальне і абстрактне визначення, яке охоплює всі речі або об'єкти, які можуть бути ідентифіковані, названі, адресовані або оброблені будь-яким способом в Інтернеті. У RESTful веб-службі запити, зроблені на URI ресурсу, викликають відповідь, яка може бути в HTML, XML, JSON або іншому форматі. Відповідь може підтверджувати, що деякі зміни були внесені в збережений ресурс, також відповідь може надавати гіпертекстові посилання на інші пов'язані ресурси або збірки ресурсів. При використуванні протоколу HTTP доступні операції GET, POST, PUT, DELETE та інші попередньо визначені методи CRUD HTTP.

Термін «репрезентативний стан» був введений та визначений Рой Філдінг у докторській дисертації в 2000 році [13]. Дисертація Філдінга пояснює принципи REST, які були відомі як "об'єктна модель HTTP", починаючи з 1994 р., і використовувалися при розробці стандартів HTTP 1.1 та уніфікованих ідентифікаторів ресурсів (URI) [26]. Термін призначений для виявлення того, як веде себе добре продумана веб-програма: це мережа веб-ресурсів (віртуальна машина-станція), де користувач проходить через додаток, вибираючи посилання і такі операції, як GET або DELETE (переходи стану), внаслідок чого наступний ресурс (що представляє наступний стан програми) передається користувачеві для їх використання.

Далі наведені основні важливі ресурси прикладного інтерфейсу додатку.

### **2.3.1 Автентифікація**

Автентифікацію користувачів реалізовано з використання стандарту OAuth [27].

OAuth - це відкритий стандарт для делегування доступу, який зазвичай використовується як спосіб для користувачів Інтернету надавати веб-сайтам або додаткам доступ до своєї інформації на інших веб-сайтах, але не надаючи їм паролів [37]. Цей механізм використовується такими компаніями, як Amazon, Google, Facebook, Microsoft і Twitter, щоб дозволити користувачам ділитися інформацією про свої рахунки із сторонніми програмами або веб-сайтами. OAuth надає клієнтам безпечний делегований доступ до ресурсів сервера від імені власника ресурсу. Стандарт визначає процес власників ресурсів для авторизації незалежного доступу до своїх серверних ресурсів без обміну їх обліковими даними. Спеціально розроблений для роботи з протоколом передачі по протоколу Hypertext Transfer (HTTP), OAuth, по суті, дозволяє отримати токени доступу (access token) стороннім клієнтам сервером авторизації із затвердженням власника ресурсу (рис. 2.7). Тоді третя сторона використовує токен доступу для звернення до захищених ресурсів, розміщених на сервері ресурсів. Токени доступу - це рядок, що представляє дозвіл, виданий клієнту. Рядок зазвичай непрозорий для клієнта. Токени мають області та тривалість доступу, надані власником ресурсу. Після того, як токен доступу було отримано він додається в полі "Authorization" до кожного запиту до захищеного ресурсу.

Разом з токеном доступу клієнту може видаватися токен для оновлення (refresh token). Токени для оновлення є обліковими даними, які використовуються для отримання токенів доступу. Токени для оновлення випускаються клієнту сервером авторизації і використовуються для отримання нового токена доступу, коли поточний токен доступу стає недійсним або термін його дії закінчується, або для отримання додаткових токенів доступу з ідентичною або більш вузькою областю дії (токени доступу можуть мати менший термін служби та менше дозволів, ніж санкціонований власником ресурсу).

В даному проекті в ролі клієнтів виступають мобільний та веб-додатки, а у ролі серверу авторизації та серверу ресурсів – сервер додатку.

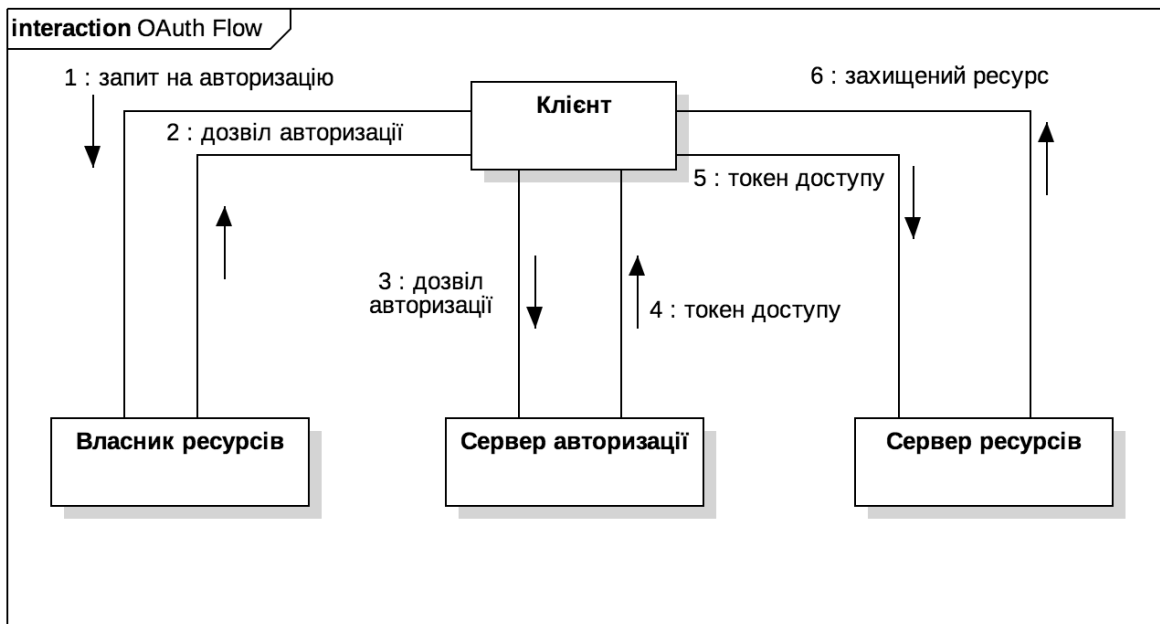


Рисунок 2.7 – Діаграма взаємодії при авторизації клієнта за протоколом OAuth

### Формат запиту

```

POST /o/token/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
  
```

### Опис полів тіла запиту

– `grant_type` – тип запиту на авторизацію. Тип даних: рядок. Для автентифікації користувача значення має бути “password”, а для оновлення токена доступу за допомогою токена відновлення - “refresh\_token”.

– `client_id` – ідентифікатор клієнта. Тип даних: рядок.

– `username` – адреса електронної пошти користувача. Тип даних: рядок.

– `password` – пароль користувача. Тип даних: рядок.

### Опис полів відповіді

– `expires_in` – термін дії токена доступу в секундах. Тип даних: ціле число.

- `scope` - область дії токєну. Тип даних: рядок.
- `access_token` - токен доступу. Тип даних: рядок
- `refresh_token` - токен відновлення. Тип даних: рядок
- `token_type` - тип токєну. Тип даних: рядок. Завжди має значення “Bearer”.

### Приклад запиту

```
POST /o/token/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=password&client_id=fErlusVewzIGtvC5bDdx0g3jI&username=alice%40example.com&password=p4$sw0Rd
```

### Приклад відповіді

```
{
  "expires_in": 3600,
  "scope": "write read",
  "access_token": "mPss4XqtTJYeoPwFhlraRixraBGAYy",
  "refresh_token": "PvMDDWF8wo0oCvhnMkbDxc4ZRL0blX",
  "token_type": "Bearer"
}
```

## 2.3.2 Користувачі

Цей ресурс дозволяє отримати дані поточного автєнтифікованого користувача системи.

### Формат запиту

```
GET /api/users/me HTTP/1.1
Authorization: Bearer mPss4XqtTJYeoPwFhlraRixraBGAYy
Cache-Control: no-cache
```

### Опис полів відповіді

- `id` - унікальний ідєнтифікатор користувача системи. Тип даних: рядок.
- `displayName` - ім'я користувача для відображення в інтерфейсі. Тип даних: рядок.



- `firstName` ім'я користувача. Тип даних: рядок.
- `lastName` прізвище користувача. Тип даних: рядок.
- `fullName` повне ім'я користувача. Тип даних: рядок.
- `email` адреса електронної пошти користувача. Тип даних: рядок.
- `isVerifiedEmail` ознака чи користувач контролює дану адресу електронної пошти. Тип даних: булеве.
- `phone` номер телефону користувача. Тип даних: рядок.
- `isVerifiedPhone` ознака чи користувач контролює даний номер телефону. Тип даних: булеве.
- `countryIso2` двозначний код країни користувача за стандартом ISO 3166-1 alpha-2. Тип даних: рядок.
- `timezone` часовий пояс користувача згідно з базою даних часових поясів IANA [33]. Тип даних: рядок.
- `languageCode` код назви мови користувача згідно з стандартом ISO 639-2. Тип даних: рядок.
- `pictureUrl` посилання на зображення профілю користувача відносно корню сервера ресурсів. Тип даних: рядок.
- `publicKey` публічний ключ RSA користувача у форматі DER закодований Base64 [9]. Тип даних: рядок.

#### Приклад відповіді

```
{
  "id": "2N21DM09QP",
  "displayName": "Alice Dixon",
  "firstName": "Alice",
  "lastName": "Dixon",
  "fullName": "Alice Dixon",
  "email": "alice@example.com",
  "isVerifiedEmail": true,
  "phone": "+3805554443333",
  "isVerifiedPhone": false,
  "countryIso2": "FR",
```

```

    "timezone": "Europe/Paris",
    "languageCode": "en",
    "pictureUrl": "2N/21/DM09QP/profile.jpg",
    "publicKey":
"MIIBCgKCAQEAAxAcQ2ewlRLhj9KzG3TwEJ93Zt1Ayf+pkixhH7V261xN/JVLwc/MUu5uctccksYE/M5veJv6KV
VMw00i432tOwdozZ9fEx5UPGBbkqtnM1L8UupgbRbxLmzbX79MG8++EYTTThVbY8EUFvGoJf680N7NQ2VpgVS5j
uDXrJSv4ftS8T+DkrKKv7KoBGAKIze5E208Hmatf+mECwwn1NoCt33cOEJQL7Poc3afvi8FyGF26K5sLRfC8Z9
GRsrLh0xNOE4+v5TCXYQJkonKmhB/kDze7zmJ2jPLGWQWRAtIeT741CnDPSTGGVy2CJx8DIInCipyTh/vQCiIFX
EX1UGNKuhawIDAQAB"
  }

```

### 2.3.3 Користувачі організації

Цей ресурс дозволяє отримати список користувачів організації, до якої належить поточний автентифікований користувач системи. Треба відзначити, що користувач організації не є тотожним до облікового запису користувача системи, незважаючи на те що вони можуть відповідати одній людині. Для кожної організації користувач системи має окремий публічний ключ та може мати адресу електронної пошти, що відрізняється від такого, що вказано у його обліковому запису в системі. Це надає змогу одному користувачу систему приймати участь у декількох організаціях одночасно та виконувати в них різні ролі маючи тільки один обліковий запис.

Список користувачів поділено на сторінки. Посилання на наступну та попередню сторінку надані у відповіді сервера

#### Формат запити

```

GET /api/people HTTP/1.1
Authorization: Bearer JCZoSF1paw5B2D7vNt1lIF6TzOH2T2
Cache-Control: no-cache

```

#### Опис полів відповіді

- count            загальна кількість користувачів у відповіді. Тип даних: ціле число
- next             адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінки. Тип даних: рядок або null.

- previous адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінки. Тип даних: рядок або null.
- results список користувачів
- results[].id унікальний ідентифікатор користувача організації. Тип даних: рядок.
- results[].email адреса електронної пошти користувача організації.
- results[].organizationId унікальний ідентифікатор організації, до якої належить користувач. Тип даних: рядок.
- results[].isAdmin відзначає чи є користувач адміністратором організації. Тип даних: булеве.
- publicKey публічний ключ RSA користувача у форматі DER закодований Base64 [9]. Тип даних: рядок.

### Приклад відповіді

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "92872aac-5130-0ab5-cb61-91cecea6019d",
      "displayName": "Alice Dixon",
      "email": "alice@example.com",
      "organizationId": "4JGWNARM9G",
      "isAdmin": true,
```

```

    "publicKey":
      "MIIBCgKCAQEAo/nsCWhr50mAIj+3B6FxuavFw5x0Ndw0hgsjt3Hb3719t6Fkd+VyIpFJUiOjKtzQvcquAeOft
      5hfapIQFoXzBKFDwsbC5aC1ldXE2BRmCd09s5B1CEgVIjRYAszbozVMPSlmKs4t71IcB4yYqsauDT6BExp76cI
      YM+1H6R+kAQIDAQAB"
    }
  ]
}

```

### 2.3.4 Товари

Цей ресурс дозволяє отримати список товарів організації, до якої належить поточний автентифікований користувач. Кожен товар належить організації і не може бути використаний за її межами. Список товарів поділено на сторінки. Посилання на наступну та попередню сторінку надані у відповіді сервера.

#### Формат запити

```

GET /api/products HTTP/1.1
Authorization: Bearer JCZoSFlpaw5B2D7vNt1ifI6TzOH2T2
Cache-Control: no-cache

```

#### Опис полів відповіді

- `count` загальна кількість товарів у відповіді. Тип даних: ціле число
- `next` адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінку. Тип даних: рядок або null.
- `previous` адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінку. Тип даних: рядок або null.
- `results` список товарів
- `results[].id` унікальний ідентифікатор товару. Тип даних: рядок.
- `results[].displayName` найменування товару для відображення у інтерфейсі користувача. Тип даних: рядок.
- `results[].organizationId` унікальний ідентифікатор організації, до якої належить товар. Тип даних: рядок.

– results[].departmentId унікальний ідентифікатор відділу організації, до якої належить товар. Може має значення null, якщо товар не належить окремому відділу. Тип даних: рядок.

– results[].eanCode універсальний код продукту EAN чи UPC [12]. Тип даних: рядок.

– results[].unitsPerPack кількість одиниць в упаковці. Тип даних: ціле число.

– results[].status поточний статус товару. Може приймати значення “active” або “inactive”. Тип даних: рядок.

– results[].displayOrder Порядковий номер для упорядкування відносно інших товарів в інтерфейсі користувача.

### Приклад відповіді

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "92872aac-5130-1a1e-d1f8-14d76490afaa",
      "displayName": "Allina Liquoroso",
      "organizationId": "4JGWNARM9G",
      "departmentId": "92872aac-5130-cf96-dd63-03ff8d54f8c5",
      "eanCode": "",
      "unitsPerPack": 1,
      "status": 1,
      "displayOrder": 0
    }
  ]
}
```

### 2.3.5 Ціни

Цей ресурс дозволяє отримати список цін на товари організації, до якої належить поточний автентифікований користувач. Список цін поділено на сторінки. Посилання на наступну та попередню сторінку надані у відповіді сервера.

Кожен об'єкт ціни належить до організації і не може бути використаний за її межами. Крім того, якщо організацію поділено на відділи, то можливо що один із відділів буде мати ціну на товар дійсну виключно для цього відділу. Якщо приналежності ціни до відділу не вказано, то ціна доступна для всіх відділів організації.

### Формат запиту

```
GET /api/prices HTTP/1.1
Authorization: Bearer JCZoSF1paw5B2D7vNt1lIFt6TzOH2T2
Cache-Control: no-cache
```

### Опис полів відповіді

- count загальна кількість цін у відповіді. Тип даних: ціле число
- next адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінки. Тип даних: рядок або null.
- previous адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінки. Тип даних: рядок або null.
- results список цін
- results[].id унікальний ідентифікатор ціни. Тип даних: рядок.
- results[].organizationId унікальний ідентифікатор організації, до якої належить ціна. Тип даних: рядок.
- results[].departmentId унікальний ідентифікатор відділу організації, до якої належить ціна. Може має значення null, якщо ціна не належить окремому відділу. Тип даних: рядок.
- results[].productId унікальний ідентифікатор товару організації, до якої належить ціна Тип даних: рядок.
- results[].currency Трьох-символьний код валюти згідно з ISO 4217[19]. Тип даних: рядок.
- results[].amount абсолютне значення вартості товару. Тип даних: число з фіксованою точкою, рядок.

### Приклад відповіді

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "92872aac-5130-b853-7ecd-ea8de7d5ecf0",
      "organizationId": "4JGWNARM9G",
      "departmentId": null,
      "categoryId": "92872aac-5130-29c4-187d-02436de26811",
      "productId": "92872aac-5130-1a1e-d1f8-14d76490afaa",
      "currency": "UAH",
      "amount": "866.090",
    }
  ]
}
```

### 2.3.6 Адреси

Цей ресурс дозволяє отримати адресну книгу організації, до якої належить поточний автентифікований користувач. Будь-який об'єкт адреси не може бути використаний за межами організації, до якій він належить. Список адресів поділено на сторінки. Посилання на наступну та попередню сторінку надані у відповіді сервера.

#### Формат запити

```
GET /api/addresses HTTP/1.1
Authorization: Bearer JCZoSF1paw5B2D7vNt1ifI6TzOH2T2
Cache-Control: no-cache
```

#### Опис полів відповіді

- **count**            загальна кількість адресів у відповіді. Тип даних: ціле число
- **next**            адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінки. Тип даних: рядок або null.
- **previous**        адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінки. Тип даних: рядок або null.

- results список цін
- results[].id унікальний ідентифікатор адреси. Тип даних: рядок.
- results[].organizationId унікальний ідентифікатор організації, до якої належить ціна. Тип даних: рядок.
  - results[].streetAddress назва вілиці та ідентифікатор будівлі та квартири чи офісу. Тип даних: рядок.
  - results[].city назва міста або іншого населеного пункту. Тип даних: рядок.
  - results[].state назва штату або області. Тип даних: рядок.
  - results[].region назва району, якщо доступно. Тип даних: рядок.
  - results[].postalCode поштовий індекс. Тип даних: рядок.
  - results[].countryIso2 двозначний код країни за стандартом ISO 3166-1 alpha-2. Тип даних: рядок.
  - results[].isVerified Ознака чи було цю адресу перевірено. Тип даних: булеве.
  - results[].languageCode код назви мови адреси згідно зі стандартом ISO 639-2. Тип даних: рядок



## Приклад відповіді

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "92872aac-5130-f649-debf-f9e922e0eac6",
      "streetAddress": "Petpi Lane",
      "city": "Hojezfi",
      "state": "NM",
      "region": "",
      "postalCode": "10292",
      "countryIso2": "US",
      "organizationId": "4JGWNARM9G",
      "isVerified": false,
      "languageCode": "en-us",
    },
  ]
}
```

### 2.3.7 Замовники

Цей ресурс дозволяє отримати список замовників організації, до якої належить поточний автентифікований користувач системи. Будь-який об'єкт цього ресурсу не може бути використаний за межами організації, якій він належить. Список замовників поділено на сторінки. Посилання на наступну та попередню сторінку надані у відповіді сервера.

#### Формат запити

```
GET /api/customers HTTP/1.1
Authorization: Bearer JCZoSF1paw5B2D7vNt1lIF6TzOH2T2
Cache-Control: no-cache
```

#### Опис полів відповіді

- **count** загальна кількість користувачів у відповіді. Тип даних: ціле число
- **next** адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінки. Тип даних: рядок або null.

- previous адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінки. Тип даних: рядок або null.
- results список користувачів
- results[].id унікальний ідентифікатор замовника організації. Тип даних: рядок.
- results[].displayName Найменування замовника для відображення у інтерфейсі користувача. Тип даних: рядок.
- results[].organizationId унікальний ідентифікатор організації, до якої належить замовник. Тип даних: рядок.
- results[].departmentId унікальний ідентифікатор відділу організації, до якої належить замовник. Може має значення null, якщо замовник не належить окремому відділу. Тип даних: рядок.
- results[].managerId унікальний ідентифікатор користувача організації, відповідного за роботу с замовником. Тип даних: рядок.
- results[].currency Трьох-символьний код валюти згідно з ISO 4217[19]. Тип даних: рядок.
- results[].balance абсолютне значення балансу замовника. Тип даних: число з фіксованою точкою, рядок.
- results[].dateBalanceUpdated дата та час останньої зміни балансу у форматі ISO 8601[20]. Тип даних: дата, рядок.
- results[].billingAddressId унікальний ідентифікатор юридичної адреси замовника, відповідного за роботу с замовником. Тип даних: рядок
- results[].shippingAddressId унікальний ідентифікатор адреси замовника для доставки. Тип даних: рядок

### 2.3.8 Заявки

Цей ресурс дозволяє отримувати список заявок організації, до якої належить поточний автентифікований користувач системи, а також

створювати нові та змінити існуючі заявки. Будь-який об'єкт цього ресурсу не може бути використаний за межами організації, якій він належить.

### Формат запиту на отримання даних

```
GET /api/orders HTTP/1.1
Authorization: Bearer JCZoSF1paw5B2D7vNt1ifI6TzOH2T2
Cache-Control: no-cache
```

### Опис полів відповіді

- count            загальна кількість заявок у відповіді. Тип даних: ціле число
- next            адреса наступної сторінки відповіді або “null”, якщо відповідь не має наступної сторінки. Тип даних: рядок або null.
- previous        адреса наступної сторінки відповіді або “null”, якщо відповідь не має попередньої сторінки. Тип даних: рядок або null.
- results         список користувачів
- results[].id    унікальний ідентифікатор заявки. Тип даних: рядок.
- results[].organizationId    унікальний ідентифікатор організації, до якої належить заявка. Тип даних: рядок.
- results[].departmentId    унікальний ідентифікатор відділу, до якого належить заявка. Тип даних: рядок.
- results[].customerId    унікальний ідентифікатор замовника. Тип даних: рядок.
- results[].ownerId        унікальний ідентифікатор користувача організації, що створив заявку. Тип даних: рядок.
- results[].currency        Трьох-символьний код валюти згідно з ISO 4217 [19]. Тип даних: рядок.
- results[].amount         сумарна вартість товару в заявці. Тип даних: число з фіксованою точкою, рядок.
- results[].status         поточний стан заявки. Може приймати такі значення: “draft”, “pending”, “processing”, “shipped”, “complete”, “canceled”. Тип даних: рядок.

- results[].comment необов'язковий коментар до заявки. Тип даних: рядок.
- results[].dateCreated дата та час створення заявки у форматі ISO 8601 [20]. Тип даних: дата, рядок.
- results[].dateUpdated дата та час останньої зміни заявки у форматі ISO 8601 [20]. Тип даних: дата, рядок.
- results[].location географічні координати місця, де було створено заявку, у вигляді хеш-коду [15].
- results[].lines[].product унікальний ідентифікатор товару. Тип даних: рядок.
- results[].lines[].quantity кількість товару. Тип даних: число з фіксованою точкою, рядок.
- results[].lines[].price ціна товару. Тип даних: число з фіксованою точкою, рядок.
- results[].lines[].amount сумарна вартість товару. Тип даних: число з фіксованою точкою, рядок.

## Приклад відповіді

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "92872aac-5130-4eee-e5c0-342df40a5885",
      "organizationId": "4JGWNARM9G",
      "departmentId": null,
      "customerId": "92872aac-5130-dff6-08b1-3e25d42b8379",
      "ownerId": "92872aac-5130-0ab5-cb61-91cecea6019d",
      "currency": "UAH",
      "amount": "3673.390",
      "status": "pending",
      "comment": null,
      "dateCreated": "2018-06-11T19:44:18.224574+02:00",
      "dateUpdated": "2018-06-11T19:44:18.323425+02:00",
      "location": "vxeg7v7",
      "lines": [
        {
          "product": "92872aac-5130-b750-0ed8-fb99d950992d",
          "quantity": "1.000",
          "price": "901.730",
          "amount": "901.730"
        }
      ]
    }
  ]
}
```

## Формат запиту на створення

```
POST /api/orders HTTP/1.1
Content-Type: application/json
Authorization: Bearer JCZoSF1paw5B2D7vNt1lIF6TzOH2T2
Cache-Control: no-cache
```

## Приклад запиту

```
{
  "currency": "UAH",
  "categoryId": "92872aac-5130-29c4-187d-02436de26811",
  "customerId": "92872aac-5130-dff6-08b1-3e25d42b8379",
  "organizationId": "4JGWNARM9G",
  "location": "",
  "comment": "",
  "lines": [
    {
      "product": "92872aac-5130-4908-0527-94ecc934da9e",
      "quantity": 42
    }
  ],
  "status": "pending"
}
```

## Приклад відповіді

```
{
  "id": "92872aac-5130-8579-7bb9-00b4e36ba374",
  "organizationId": "4JGWNARM9G",
  "departmentId": null,
  "customerId": "92872aac-5130-dff6-08b1-3e25d42b8379",
  "ownerId": "92872aac-5130-0ab5-cb61-91cecea6019d",
  "categoryId": "92872aac-5130-29c4-187d-02436de26811",
  "currency": "UAH",
  "amount": "29802.780",
  "status": "pending",
  "comment": "",
  "dateCreated": "2018-06-12T17:05:41.898549+02:00",
  "dateUpdated": "2018-06-12T17:05:41.986202+02:00",
  "location": "",
  "lines": [
    {
      "product": "92872aac-5130-4908-0527-94ecc934da9e",
      "quantity": "42.000",
      "price": "709.590",
      "amount": 29802.78
    }
  ],
}
```

### **2.3.9 Синхронізація заявок на мобільному пристрої**

Під час синхронізації (рис. 2.8) з мобільного пристрою на сервер по одній відправляються всі заявки, що були створені або змінені з минулої синхронізації, крім заявок що мають статус “draft”. Передача заявок на сервер по черзі дозволяє підвищити шанси успішної передачі при слабкому каналі зв'язку. Крім того, інакше треба було б або окремо групувати нові та змінені заявки, і однаково робити кілька більших запитів, або додавати підтримку формування одного пакету зі всіма змінами. З точки зору мінімізації кількості з'єднань та впливу на батарею це, можливо, було б кращім рішенням, але це значно ускладнило б реалізацію, та значно збільшило б ризик помилок передачі даних через обриви та тайм-аути.

Після того як заявки було передано мобільний клієнт запитує у серверу ресурсів дані, що були змінені після минулої успішної синхронізації. Коли всі дані було успішно прийнято процес синхронізації завершується..

## **2.4 Середовище та інструменти розробки**

### **2.4.1 Мови програмування та розмітки**

Під час роботи над проектом біло використано декілька мов програмування та розмітки. Сервер ресурсів було розроблено на мові програмування Python, інтерфейс користувача серверної частини та веб-клієнта було реалізовано за допомогою мови розмітки HTML та CSS. Програмну частину веб-клієнта та значну частину мобільного клієнта було розроблено на мові програмування Dart.

Python - інтерпретована мова програмування високого рівня для загального програмування. Створений Guido van Rossum і вперше випущений в 1991 році, Python має філософію дизайну, яка підкреслює високу читабельність коду. Мова Python має динамічний тип системи та автоматичне керування пам'яттю.

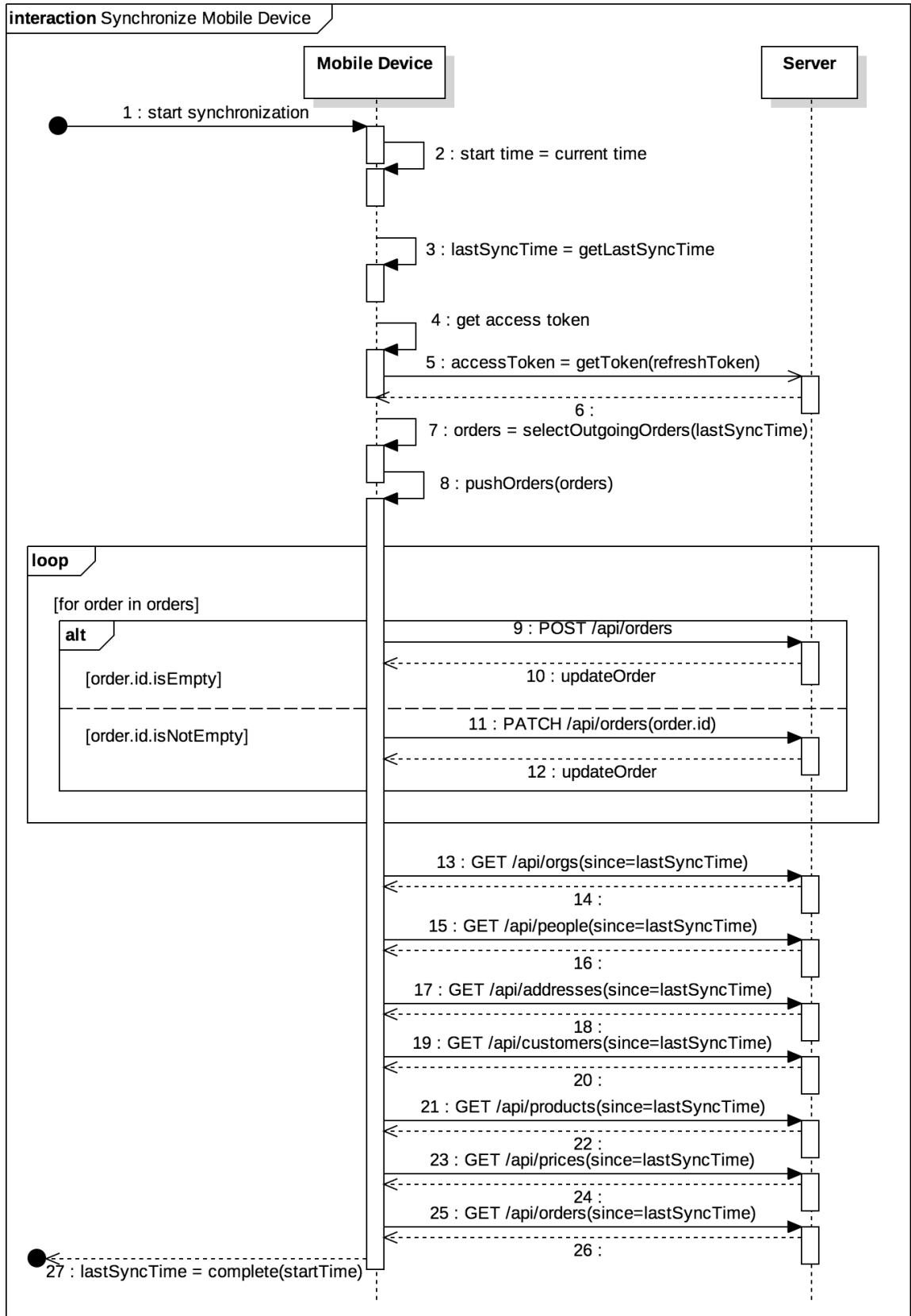


Рисунок 2.8 – Діаграма послідовності операції при синхронізації мобільного пристрою



Віна підтримує декілька парадигм програмування, включаючи об'єктно-орієнтовану, імперативну, функціональну та процедурну, і має велику і всеосяжну стандартну бібліотеку. Інтерпретатори Python доступні для багатьох операційних систем. CPython, еталонна реалізація Python, є програмним забезпеченням із відкритим кодом і має спільну модель розвитку. CPython керує неприбуткова організація Python Software Foundation.

Dart - це мова програмування загального призначення, яку спочатку було розроблено компанією Google, а пізніше затверджений компанією Ecma (ECMA-408) [10]. Вона використовується для створення веб-серверів та мобільних додатків та пристроїв Internet of Things (IoT). Це програмне забезпечення з відкритим кодом під дозволеною ліцензією на безкоштовне програмне забезпечення (модифікована ліцензія BSD). Dart є об'єктно-орієнтованою мовою з одиничним успадкуванням використовуючи синтаксис подібний до мови C. Dart підтримує інтерфейси, змішувачі, абстрактні класи і статичні типи.

Мова розмітки гіпертексту HTML - це стандартна мова розмітки для створення веб-сторінок та веб-програм [18]. Використовуючи каскадні таблиці стилів (CSS) та JavaScript, віна утворює базу для технологій World Wide Web. Веб-браузери отримують HTML-документи з веб-сервера або з локального сховища та перетворюють документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та включає інструкції що впливають на зовнішній вид документа. HTML-елементи є будівельними блоками HTML-сторінок. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані в відтворену сторінку. HTML забезпечує засіб для створення структурованих документів шляхом позначення структурної семантики тексту, такого як заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML позначені тегами, написаними за допомогою кутових дужок. Теги, такі як `<img />` та `<input />`, безпосередньо виводять

вміст на сторінку. Інші теги, такі як `<p>` оточують і надають інформацію про текст документа, і можуть містити інші теги як під-елементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки. HTML може вставляти програми, написані мовою сценаріїв, такими як JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компоновання вмісту.

SASS - це мова сценаріїв препроцесора, яка інтерпретується або компілюється в каскадні таблиці стилів (CSS). SassScript - це сама мова скриптів. SASS складається з двох синтаксисів. Оригінальний синтаксис подібний до мови Haml. Для відокремлення правил використовується відступ, та символів нового рядка для відокремлення кодів блоків. Новіший синтаксис "SCSS" використовує форматування блоку, подібне до CSS. Він використовує фігурні дужки для позначення блоків коду та точок з комою для розділення рядків усередині блоку. CSS3 складається із серії селекторів та псевдо-селекторів, які відповідають правилам групи, які застосовуються до них. Sass розширює CSS, надаючи кілька механізмів, доступних більш традиційним мовам програмування, зокрема об'єктно-орієнтованим мовам, але які не доступні самому CSS3. При інтерпретації SassScript він створює блоки правил CSS для різних селекторів, як це визначено файлом Sass. Перекладач Sass перетворює SassScript на CSS. Крім того, Sass може відстежувати зміни в файлі .sass або .scss і автоматично переводити його на вихідний файл .css.[29]

#### **2.4.2 Каркаси додатків**

Для реалізації мобільного клієнта було обрано Flutter, тому що цей каркас додатків дозволяє швидко розробляти ефективні мобільні додатки одночасно для Android та iOS. Для веб-клієнта було обрано Angular Dart 4 через те що Angular є популярним каркасом для розробки динамічних веб-додатків а використання мови Dart дало змогу повторно використовувати деякі модулі для обох клієнтських додатків, значно зменшивши стек

технологій. Сервер ресурсів було повністю реалізовано за допомогою Django 2 та Django Rest Framework.

Flutter - це SDK для розробників мобільних додатків із відкритим кодом, створений Google. Він використовується для розробки додатків для Android та iOS. Двигун Flutter, написаний в основному на C++, забезпечує підтримку рендерингу на низькому рівні за допомогою графічної бібліотеки Google Skia. Окрім того, він взаємодіє з SDK, що надаються Android та iOS. Основа бібліотеки, написана на мові Dart, надає базові класи та функції, такі як API для спілкування з движком, які використовуються для побудови програм, що використовують Flutter [14]. Зовнішній вид інтерфейсу користувача Flutter передбачає збирання та/або створення різних віджетів. Віджет у Flutter являє собою незмінний опис частини користувальницького інтерфейсу; вся графіка, включаючи текст, фігури та анімації створюються за допомогою віджетів. Більш складні віджети можуть бути створені шляхом об'єднання багатьох простих віджетів [32].

Django - це безкоштовна веб-система з відкритим кодом, написана на Python, яка йде за архітектурною схемою model-view-template (MVT). Система підтримується Django Software Foundation (DSF), незалежною некомерційною організацією. Основна мета Django - полегшити створення складних веб-сайтів, керованих базами даних. Django наголошує повторне використання і можливість "підключення" компонентів, написання менше коду та його низьку зв'язаність, швидку розробку і принцип "не повторюйте себе". Python використовується у всьому, навіть для файлів налаштувань та моделей даних. Django також забезпечує необов'язковий адміністративний інтерфейс створення, читання, оновлення та видалення об'єктів моделі, який генерується динамічно через інтроспекцію.

### **2.4.3 Сервер Web-додатків**

Для розгортання веб-додатків на сервері було використано Nginx, Gunicorn та Supervisor.

Nginx (стилізований як as NGINX, NGiIX чи nginx) є веб-сервером, який також може використовуватися як зворотний проксі-сервер, балансувач навантаження, поштовий проксі-сервер та HTTP кеш-пам'ять. Програмне забезпечення було створено Ігорем Сисоєвим і вперше було випущено в 2004 році [22]. Компанія з такою ж назвою була створена в 2011 році для надання підтримки та платного програмного забезпечення Nginx plus[34].

Nginx - це безкоштовне програмне забезпечення з відкритим кодом, яке випускається відповідно до умов ліцензії BSD. Значна частина веб-серверів використовує NGINX, часто як балансування навантаження.

Gunicorn "Зелений Єдиноріг" [17] - це HTTP-сервер інтерфейсу Gateway Interface (WSGI) для Python. Це модель pre-fork worker, що була перенесена з проекту Unicorn Ruby. Сервер Gunicorn в цілому сумісний з низкою каркасів веб-додатків, просто реалізований, невибагливий до ресурсів сервера і досить швидкий.

Supervisord - це система клієнт-сервер, яка дозволяє користувачам здійснювати моніторинг та керування кількома процесами в UNIX-подібних операційних системах. Він виконує деякі задачі таких програм, як launchd, daemontools та runit. Але на відміну від деяких з цих програм, він не призначений для запуску ініціалізації як "process id 1".

#### **2.4.4 База даних**

Для зберігання даних на сервері було застосовано СУБД PostgreSQL, а на мобільному пристрої – вбудовану базу даних SQLite 3.

PostgreSQL, або часто просто Postgres, являє собою об'єктно-реляційну систему управління базами даних (ORDBMS) з акцентом на розширюваність і відповідність стандартам [7]. В якості сервера баз даних основними функціями є безпечне зберігання даних і повернення цих даних у відповідь на запити інших програмних додатків. Він може працювати з робочими навантаженнями, починаючи від невеликих одномашинних прикладних програм до великих програм для роботи з Інтернетом (або для

зберігання даних) з багатьма одночасними користувачами; на MacOS-сервері PostgreSQL є базовою базою даних за замовчуванням [36], вона також доступна для Microsoft Windows і Linux (постачається у більшості дистрибутивів).

PostgreSQL є ACID-сумісним і транзакційним[24]. PostgreSQL має оновлювані перегляди та матеріалізовані перегляди, тригери, зовнішні ключі [7].

PostgreSQL розроблена PostgreSQL Global Development Group, різноманітною групою багатьох компаній та індивідуальних учасників [36]. Це вільне та відкрите джерело, випущене за умовами ліцензії PostgreSQL, дозволена ліцензія на програмне забезпечення.

SQLite - реляційна система керування базами даних, що міститься в бібліотеці програмування C. На відміну від багатьох інших систем управління базами даних, SQLite не є двигуном бази даних клієнт-сервер. SQLite відповідає ACID [23] і реалізує більшість стандарту SQL за допомогою динамічного та слабо набраного синтаксису SQL, який не гарантує цілісності домену [31].

SQLite - це популярний вибір як програмне забезпечення вбудованої бази даних для зберігання локального/клієнтського середовища в прикладному програмному забезпеченні, такому як веб-браузери. Це, мабуть, найпоширеніший двигун бази даних, який сьогодні використовується багатьма широко розповсюдженими браузерами, операційними системами та вбудованими системами (наприклад, мобільними телефонами) [21]. SQLite має прив'язки до багатьох мовах програмування.

#### **2.4.5 Основні інструменти та IDE**

Для редагування текстів проекту було використано Android Studio та PyCharm, а для зберігання бази вихідного коду - Git.

Git - це система керування версіями для відстеження змін у комп'ютерних файлах та координації роботи над цими файлами серед

декількох людей. Він використовується, перш за все, для управління вихідними кодами при розробці програмного забезпечення [16], але його можна використовувати для відстеження змін у будь-якому наборі файлів. В якості розподіленої системи контролю версій вона спрямована на швидкість, [30] цілісність даних та підтримку розподілених, нелінійних робочих процесів. Git був створений Лінусом Торвальдсом у 2005 році для розробки ядра Linux, а інші розробники ядра сприяли його початкового розвитку [35]. Його нинішній супровідник з 2005 року - Юніо Хамано. Як і в більшості інших систем керування розподіленою версією, і на відміну від більшості серверів-клієнт-серверів, кожен каталог Git на кожному комп'ютері є повноцінним сховищем із повними можливостями відстеження історії та повної версії, незалежно від доступу до мережі або центрального сервера. Git - це безкоштовне програмне забезпечення з відкритим кодом, яке розповсюджується відповідно до умов загальної ліцензії GNU версії 2.

Android Studio - це офіційне інтегроване середовище розробки (IDE) для операційної системи Google Android, побудоване на основі програмного забезпечення JetBrains IntelliJ IDEA та розроблене спеціально для розробки Android [8]. Він доступний для завантаження в операційних системах на базі Windows, MacOS та Linu. Це заміна для Eclipse Android Development Tools (ADT) як основна IDE для розробки власних додатків для Android.

PyCharm (рис. 2.9) - це інтегроване середовище розробки (IDE), яке використовується в комп'ютерному програмуванні, зокрема для мови Python. Розроблений чеською компанією JetBrains. Він забезпечує аналіз коду, графічний відладчик, інтегровану систему виконання unit-тестів, інтеграцію з системами керування версіями (VCS) та підтримує розробку веб-сторінок за допомогою Django.

PyCharm є крос-платформению програмою, з версіями для Windows, macOS та Linux. Існує безкоштовна редакція Community Edition випущена під ліцензією Apache, а також більш розвинута версія - Professional Edition під власною ліцензією.

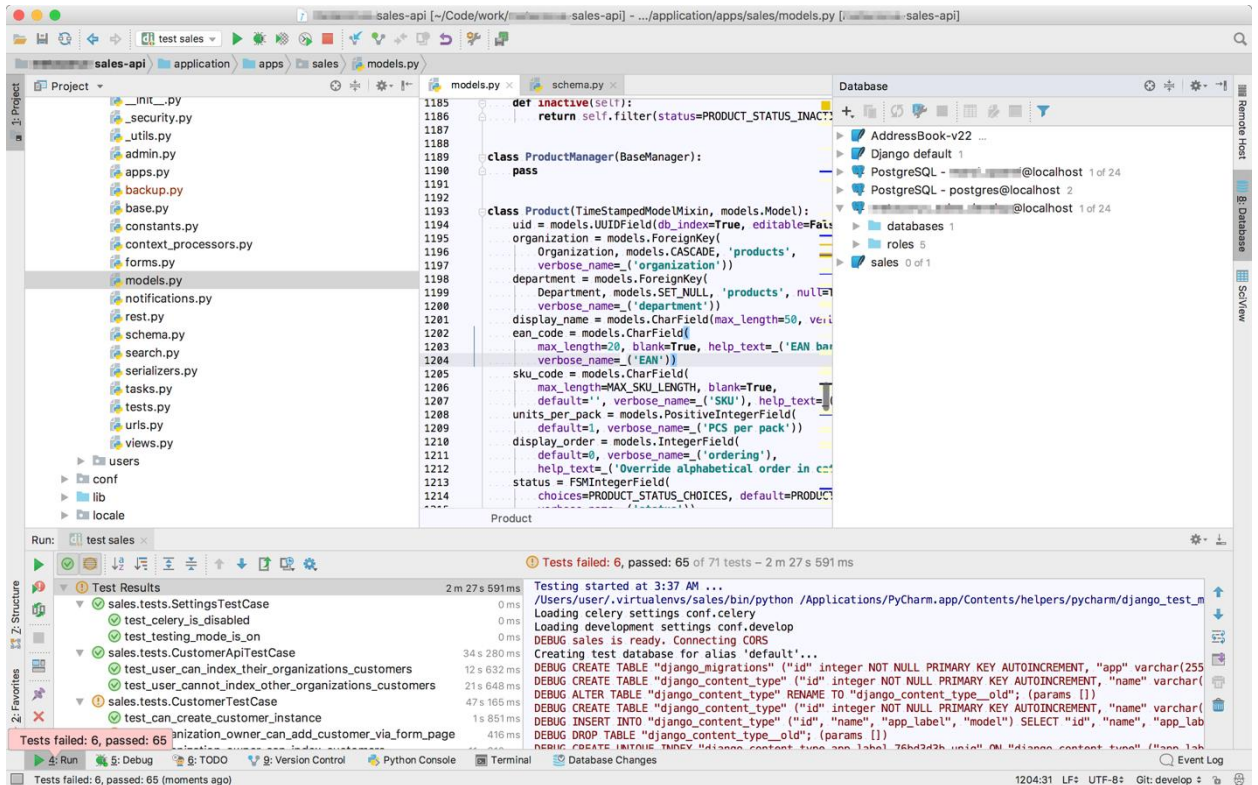


Рисунок 2.9 – Знімок екрану PyCharm з відкритим проектом

## Висновки до розділу 2

У розділі було розглянуто архітектуру системи, описані основні моменти взаємодії користувачів та клієнтських додатків із системою. Також було сформовано та описано прикладний інтерфейс системи, задокументовано формат запитів та відповідей сервера ресурсів, наведені примірники даних. Коротко описано основні інструменти та середовище розробки системи.

## 3 ІНТЕРФЕЙС КОРИСТУВАЧА

### 3.1 Інтерфейс користувача мобільного додатку

#### 3.1.1 Додавання користувача на мобільному пристрої

Перший екран, який бачить користувач вперше запустивши додаток на мобільному пристрої є екран автентифікації (рис. 3.1). Для того щоб почати користуватися додатком користувач має ввести адресу електронної пошти та пароль свого облікового запису у поля “Email” та “пароль” відповідно, та натиснути кнопку “Увійти”. Якщо введені дані не відповідають даним активного користувача, то користувачеві буде виведено повідомлення про помилку.

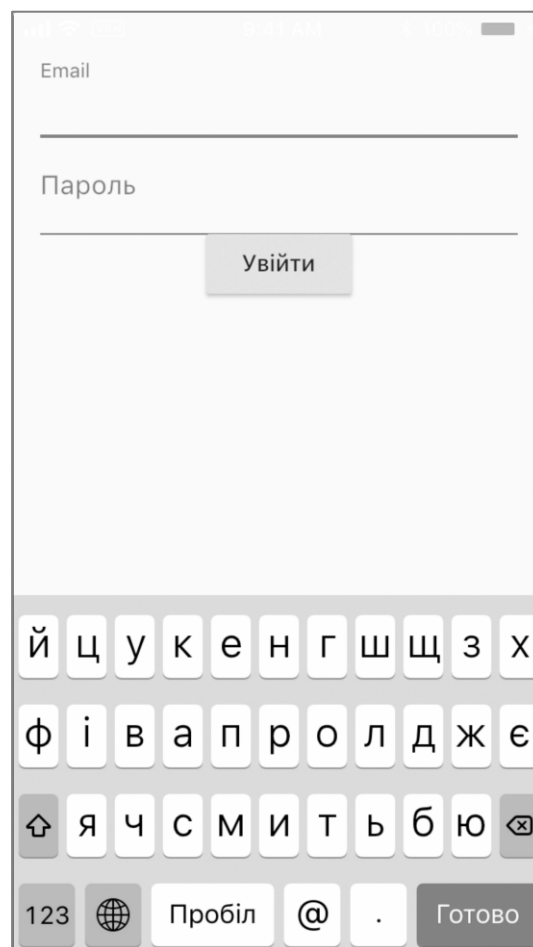


Рисунок 3.1 – Екран автентифікації користувача



### 3.1.2 Головне меню та головний екран мобільного додатку

Після того як користувач успішно пройшов автентифікацію він побачить головний екран додатку (рис. 3.2). На головному екрані зображено список заявок що було створено користувачем, в кожному рядку якого можна побачити назву замовника та його адресу доставки, а також вартість товару в заявці. Якщо користувач натисне на заявці, то він зможе переглянути її подробиці або змінити її, якщо це дозволяється станом заявки.

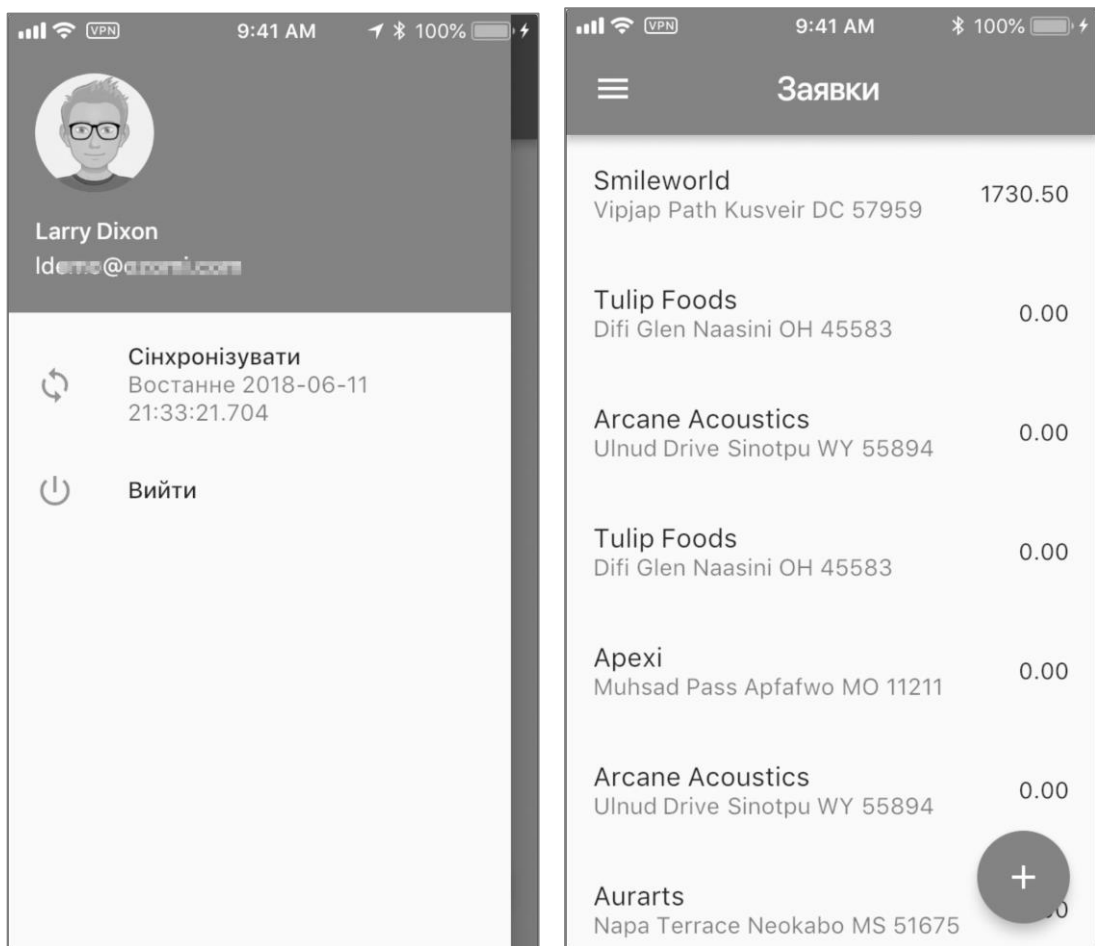


Рисунок 3.2 – Головне меню та головний екран додатку

Зліва вгорі екрану знаходиться кнопка для відкриття головного меню додатку, де можна побачити короткі відомості про обліковий запис поточного користувача, а саме: зображення профіля, ім'я та прізвище та його

адресу електронної пошти. Крім того в головному меню знаходиться пункти меню для старту синхронізації “Синхронізувати” та виходу із системи “Вийти”. В пункті меню синхронізації також можна побачити час останньої успішної синхронізації.

Також на головному екрані користувачеві доступна кнопка для створення нової заявки.

### 3.1.3 Створення нової та зміна існуючої заявки

Для створення нової або коригування існуючої заявки користувачу відкривається екран, де він може вибрати замовника, ввести примітку до заявки та додати товарні позиції (рис. 3.3).

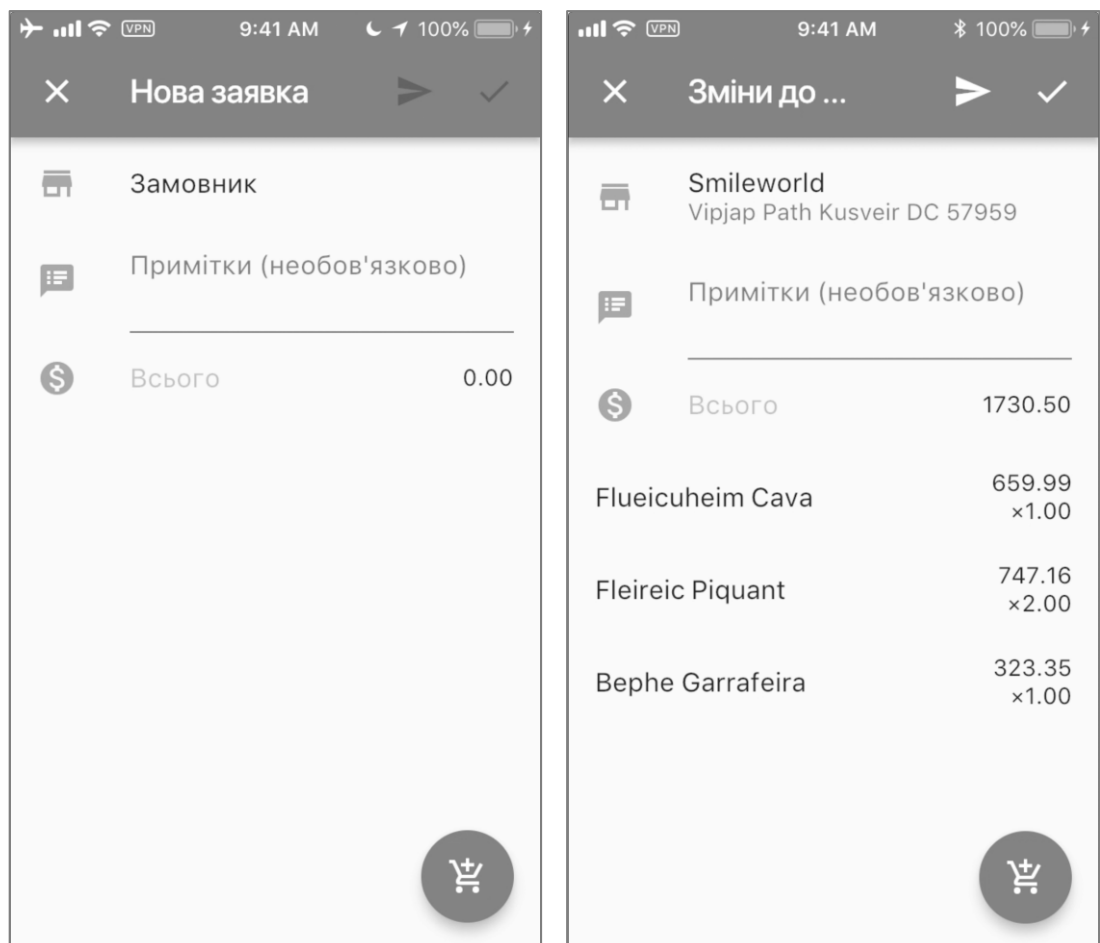


Рисунок 3.3 – Екран редагування заявки

Вгорі екрану наведено його заголовок (“Нова заявка” або “Зміни до заявки”), та кнопки для закриття форми, збереження нової заявки, або збереження заявки у чернетках. Одразу під заголовком наведені поля для вводу замовника, та примітки до заявки. Далі іде інформаційна стрічка із сумою вартості товару в заявці, під якою розміщено список товарних позицій. Внизу екрана користувачеві доступна кнопка для додавання товару в заявку.

### 3.1.4 Вибір замовника та додавання товару

Для того щоб надати можливість користувачеві вибирати замовника або товар із списку, було розроблено відповідні екрани (рис. 3.4).

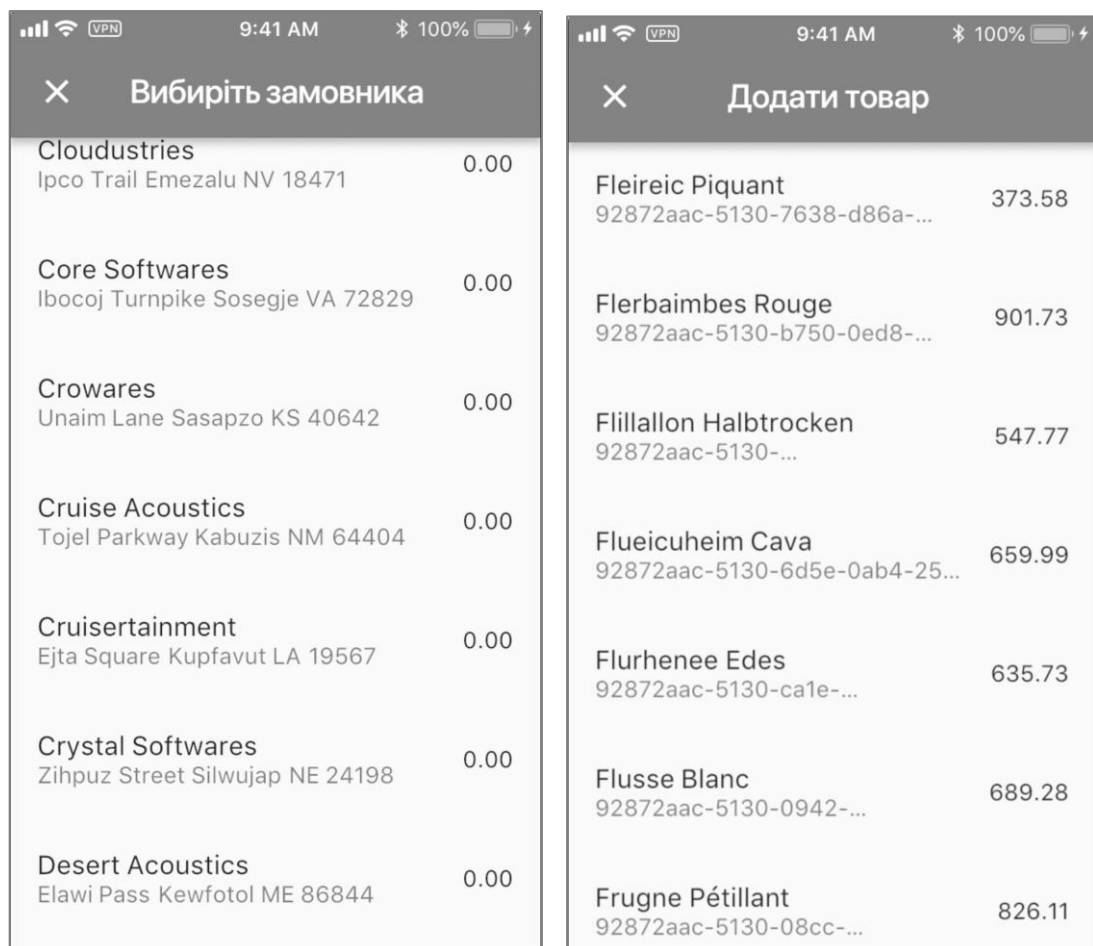


Рисунок 3.4 – Екрани вибору замовника та товару

На екрані вибору замовника наведено список замовників, доступних користувачеві, в кожному рядку якого відображено найменування та адресу доставки замовника, та його баланс у системі. Екран товару відображає список найменувань та цін доступних товарів.

## **3.2 Інтерфейс користувача веб-додатку**

### **3.2.1 Реєстрація користувача та додавання торгівельної організації**

На рис. 3.5 зображено копію екрану із сторінкою автентифікації користувача. Це сторінка, яку користувач бачить при першому відвіданні. На сторінці відображено форму для вводу електронної пошти та паролю для входу на сайт додатку. Крім того користувачу доступні посилання на сторінки реєстрації "sign up" (рис. 3.6) та відновлення паролю "forgot password".

Сторінка реєстрації – це те місце, де користувач може створити запит на запрошення до системи. Для цього йому потрібно заповнити відповідну форму ввівши такі дані як своє ім'я та прізвище, адресу електронної пошти, найменування торгівельної організації, країну та часовий пояс організація, та номер телефону для зв'язку. На підставі цих даних адміністратор системи буде приймати рішення чи затвердити або відхилити заявку на запрошення.

Після того, як адміністратор системи затвердить заявку, користувач зможе завершити створення свого облікового запису на цій сторінці, перейшовши за унікальним посиланням на запрошення із електронного листа, та почати користуватися системою.

Після того яка користувач заповнив форму та створив запит на запрошення до системи, адміністратор системи отримає листа із повідомленням, де буде також пряме посилання на сторінку перегляду запрошення.

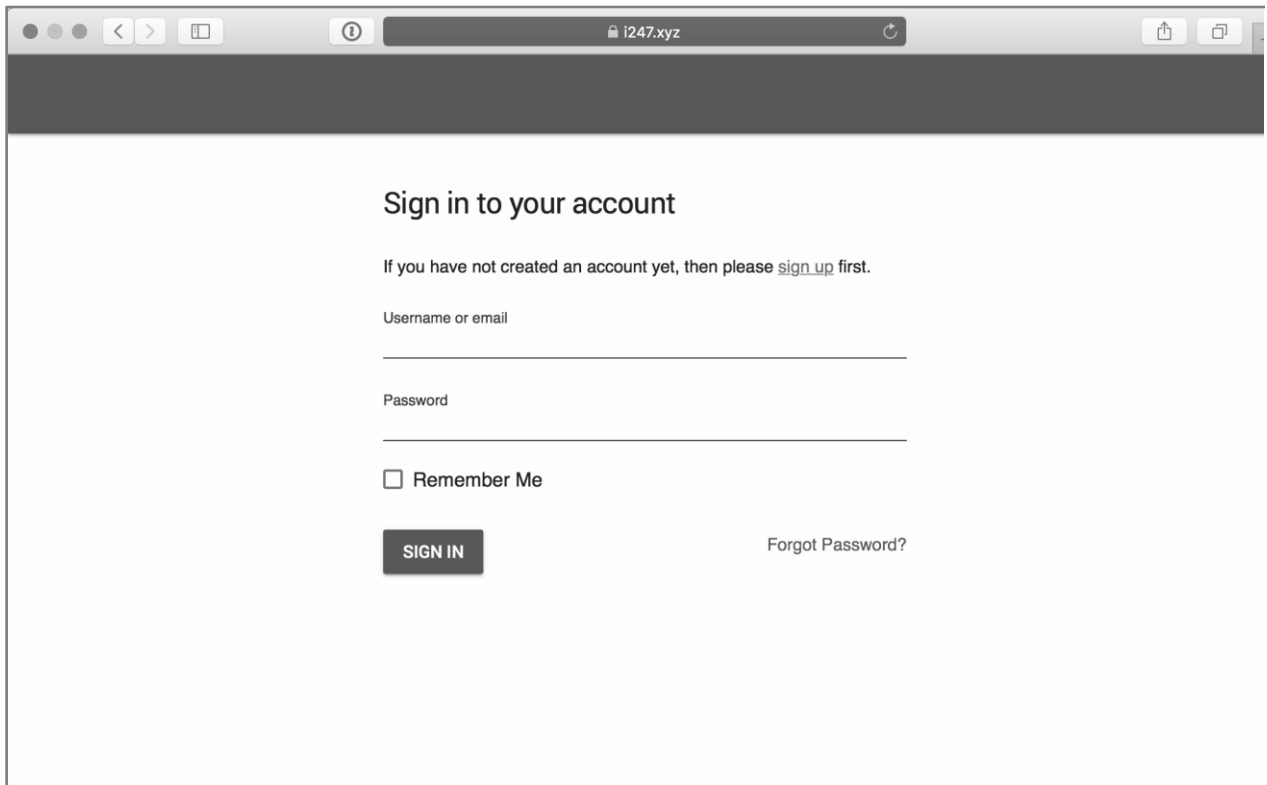
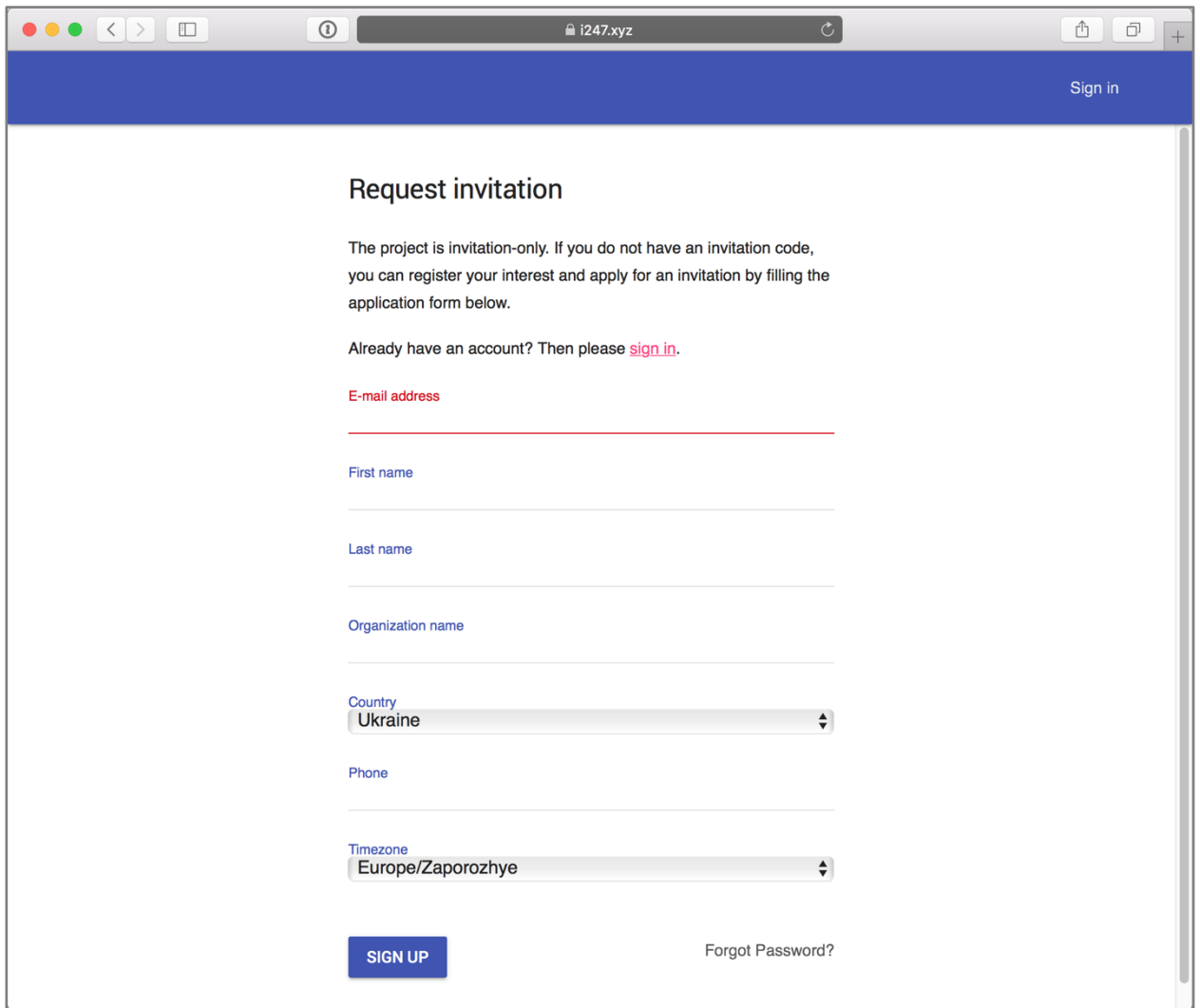


Рисунок 3.5 – Сторінка автентифікації користувача

На рис. 3.7 зображено копію екрану із сторінкою перегляду запиту на запрошення. Ця сторінка доступна тільки адміністраторам системи і дає їм змогу затвердити або скасувати запит на запрошення, що був створений користувачем на сторінці реєстрації (рис. 3.6). На сторінці відображені контактні відомості користувача, що зробив запит, найменування і часовий пояс торгівельної організації, та дані про те коли було створено запит і його термін дії.

Крім того, адміністратор системи має змогу побачити всі запити на запрошення та самі запрошення, що очікують дії користувача або адміністратора.



The screenshot shows a web browser window with the address bar displaying 'i247.xyz'. The page has a blue header with a 'Sign in' link. The main content area is titled 'Request invitation' and contains the following text: 'The project is invitation-only. If you do not have an invitation code, you can register your interest and apply for an invitation by filling the application form below.' Below this, it says 'Already have an account? Then please [sign in](#).' The registration form includes the following fields: 'E-mail address' (with a red underline), 'First name', 'Last name', 'Organization name', 'Country' (a dropdown menu with 'Ukraine' selected), 'Phone', and 'Timezone' (a dropdown menu with 'Europe/Zaporozhye' selected). At the bottom of the form, there is a blue 'SIGN UP' button and a 'Forgot Password?' link.

Рисунок 3.6 – Сторінка реєстрації користувача

На рис. 3.8 зображено копію екрану сторінки, де можна побачити цю інформацію у вигляді таблиці. Кожен рядок таблиці відповідає одному запиту або запрошенню та дає можливість побачити основну інформацію про нього, таку як ім'я та адресу електронної пошти користувача, що створив запис, назву організації що буде створено, дату створення запису і термін його дії (якщо доступний) та поточний стан запису.

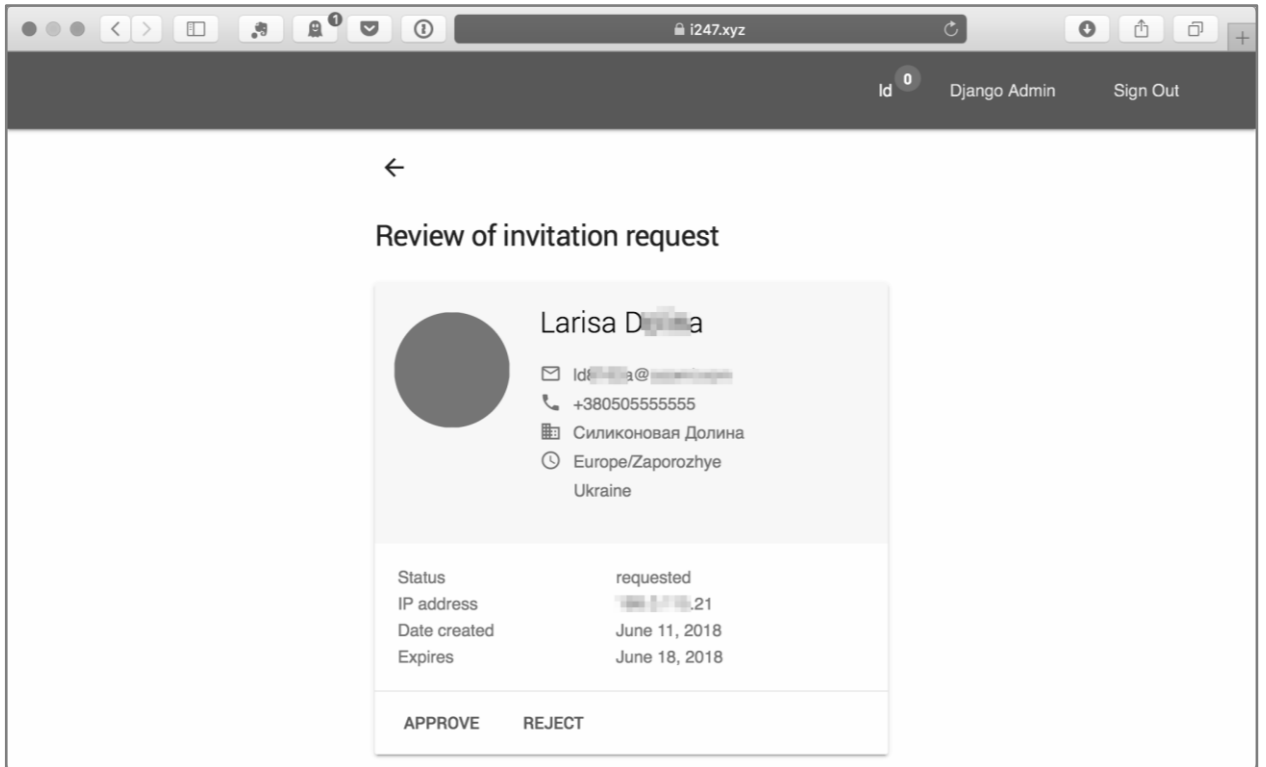


Рисунок 3.7 – Сторінка перегляду запрошення

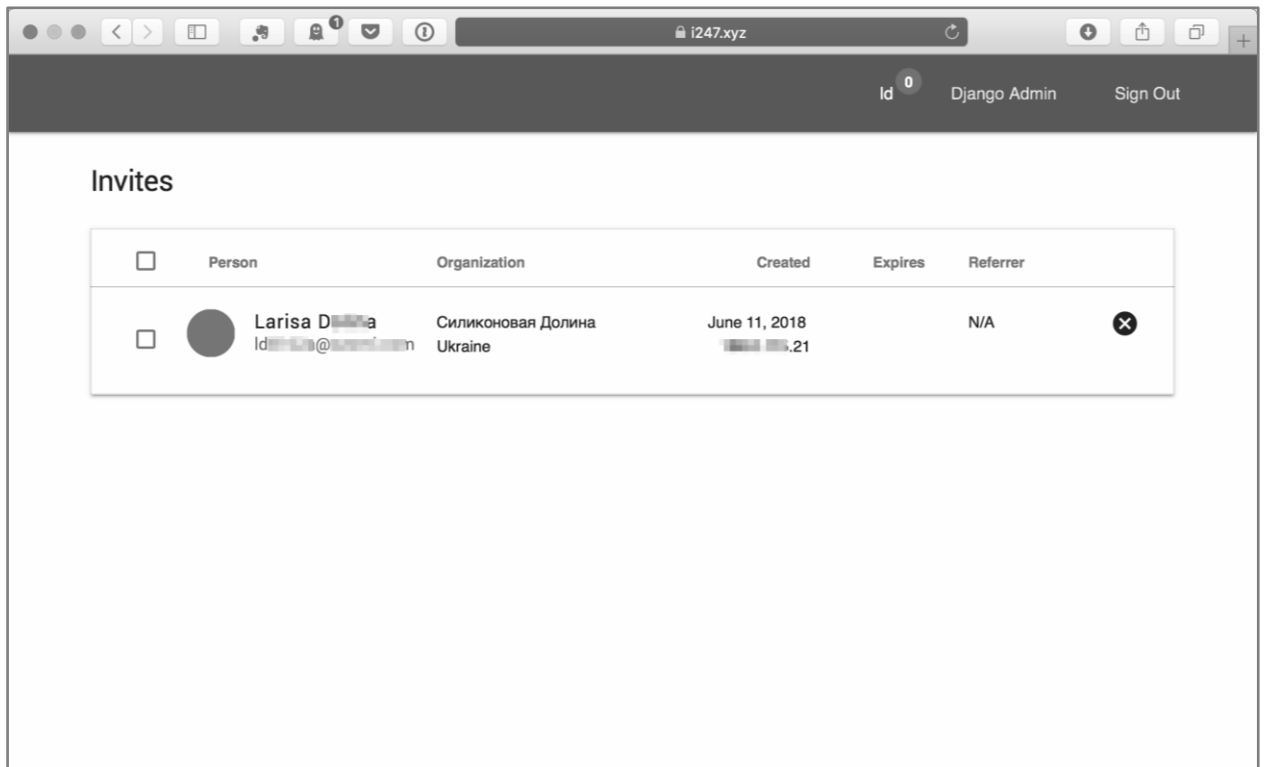


Рисунок 3.8 – Сторінка списку активних запрошень

### 3.2.2 Додавання користувачів до організації

Коли користувач скористається запрошенням до системи та створить обліковий запис та профіль організації, він зможе додавати користувачів до цієї організації. Для цього йому потрібно на сторінці перегляду учасників організації (рис. 3.9) натиснути на кнопку із символом плюсу, після чого він зможе запросити нового користувача, ввівши його адресу електронної пошти (рис. 3.10). Всі створені запрошення та їх стан буде відображено у списку очікування запрошень “Pending Invitations”. По мірі того як користувачі будуть реєструватися в системі за цими запрошеннями, вони будуть зникати із списку очікування і з’являтися у списку учасників організації “Users”.

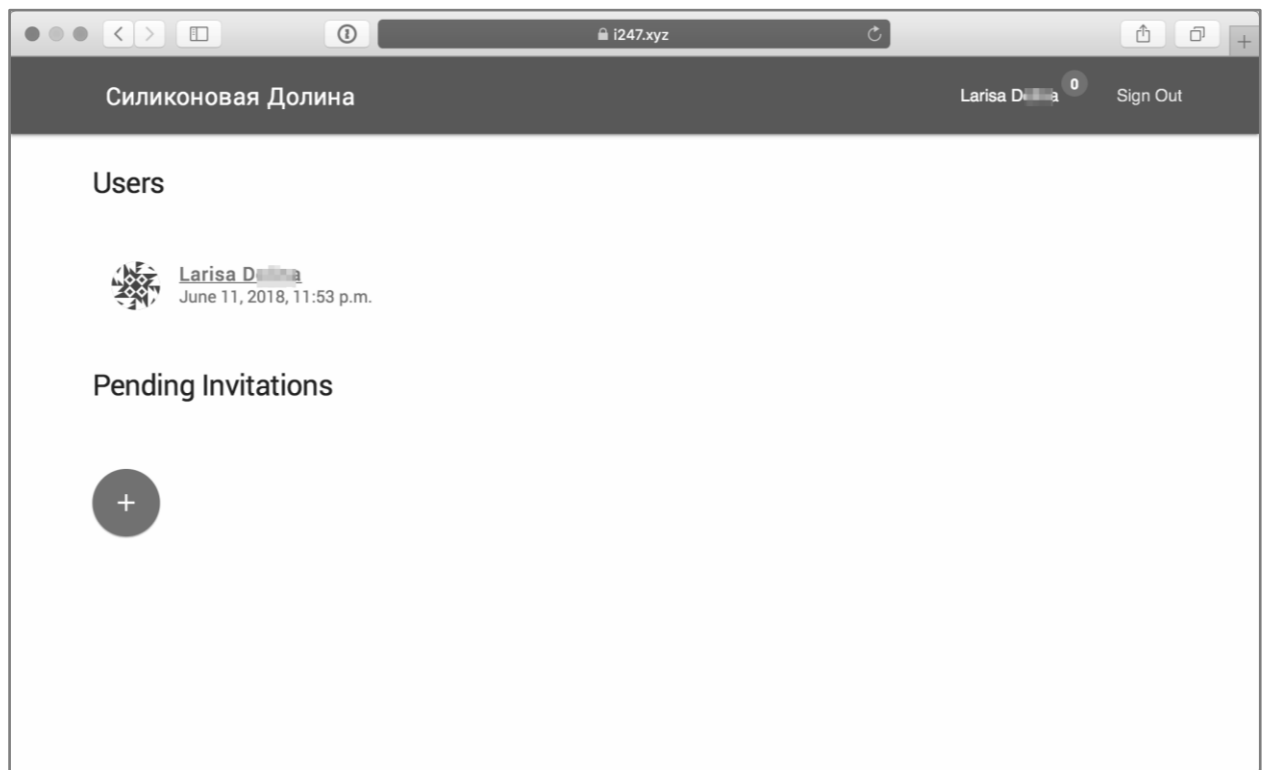


Рисунок 3.9 – Сторінка списку користувачів організації



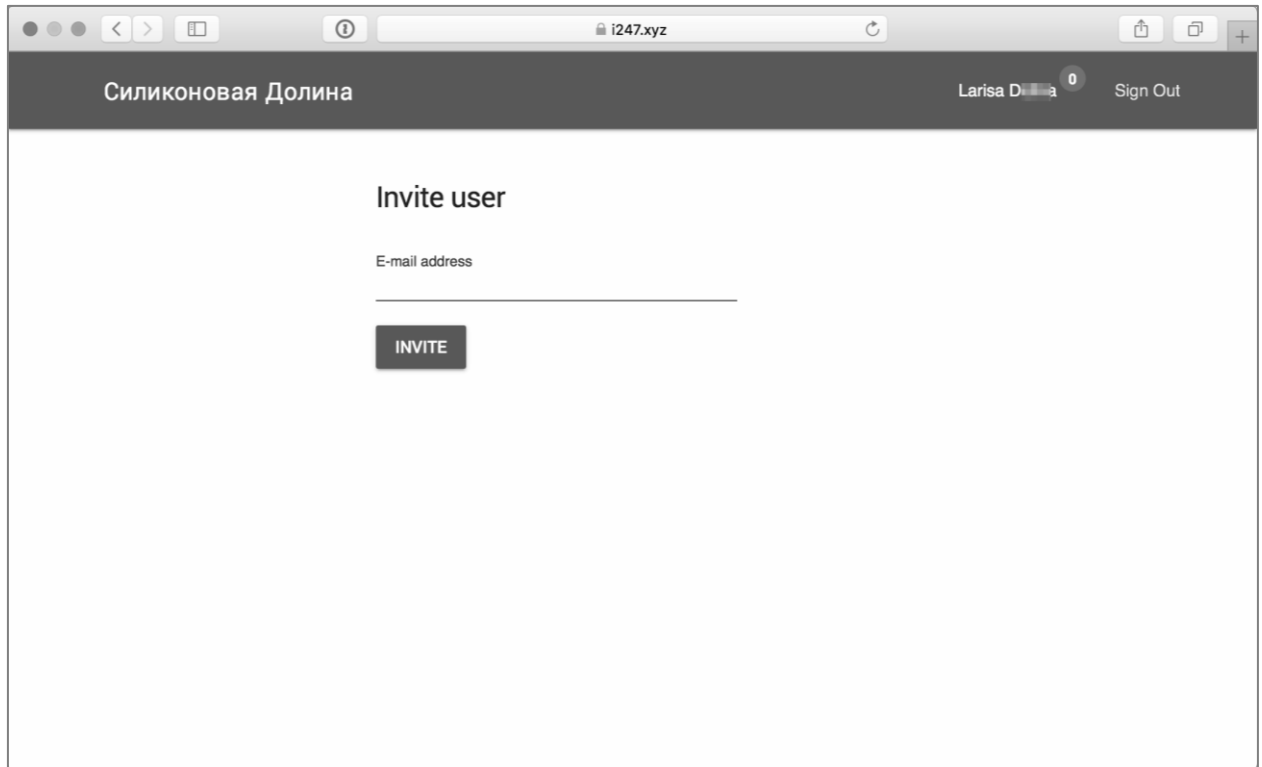


Рисунок 3.10 – Сторінка запрошення користувача до організації

### 3.2.3 Перегляд списку заявок

Будь-який учасник організації може побачити список заявок, що був створений у системі. Це можна зробити на сторінці “Заявки” (рис. 3.11). Заявки виведено у вигляді таблиці та згруповано по даті створення. Кожен рядок таблиці відображає основні відомості про заявку, такі як найменування замовника та адресу доставки, вартість товару у заявці, стан заявки, ім’я учасника організації, що створив заявку (торговельний агент або інший користувач), а також дату створення заявки та ознаку чи доступні координати місця де її було створено.

Користувач також має можливість подивитись більш детальну інформацію по заявці натиснувши на її заголовок. При цьому рядок із заявкою розсунеться і користувачеві стане доступна така інформація як повний коментар до заявці (якщо він доступний), посилання на місце створення заявки на карті Google Maps, та список позицій заявки, в кожному

рядку якого відображено найменування товару, його кількість та ціну, а також суму по даному рядку.

**Заявки**

Jun 11, 2018 Всього: 503

**Арехі**  
Muhsad Pass €3,673.39 Нова Larry Dixon 8:44 PM

Google Maps

#	Товар	Кількість	Ціна	Сума
1	Flerbaimbes Rouge	x 1	€901.73	€901.73
2	Frugne Pétillant	x 1	€826.11	€826.11
3	Fiillallon Halbtrocken	x 1	€547.77	€547.77
4	Banett	x 2	€698.89	€1,397.78

ПОЧАТИ ОБРОБКУ

---

<b>Smileworld</b> Virjap Path	€0.00	Нова	Larry Dixon	11:30 AM	⌵
<b>Tulip Foods</b> Difi Glen	€0.00	Нова	Larry Dixon	11:30 AM	⌵
<b>Arcane Acoustics</b> Mead Drive	€0.00	Нова	Larry Dixon	11:30 AM	⌵

Рисунок 3.11 – Сторінка із списком заявок

### 3.2.4 Створення нової заявки

Крім перегляду існуючих заявок на сторінці “Заявки” користувач має можливість створити нову заявку. Для цього йому потрібно натиснути на кнопку із символом плюсу внизу екрану і заповнити форму заявки (рис. 3.12). В цій формі потрібно вибрати замовника із списку що випадає, ввести коментар якщо це потрібно, та додати товарні позиції до заявки, натиснувши на кнопку із символом кошика. Для кожної позиції треба вибрати товар із списку, що випадає та ввести його кількість. У користувача є можливість зберегти нову заявку натиснувши на кнопку “Створити” або додати її у чернетки кнопкою “В чернетки”. Якщо користувач не бажає

зберігати введену інформацію він може просто закрити форму, натиснувши на відповідну кнопку.

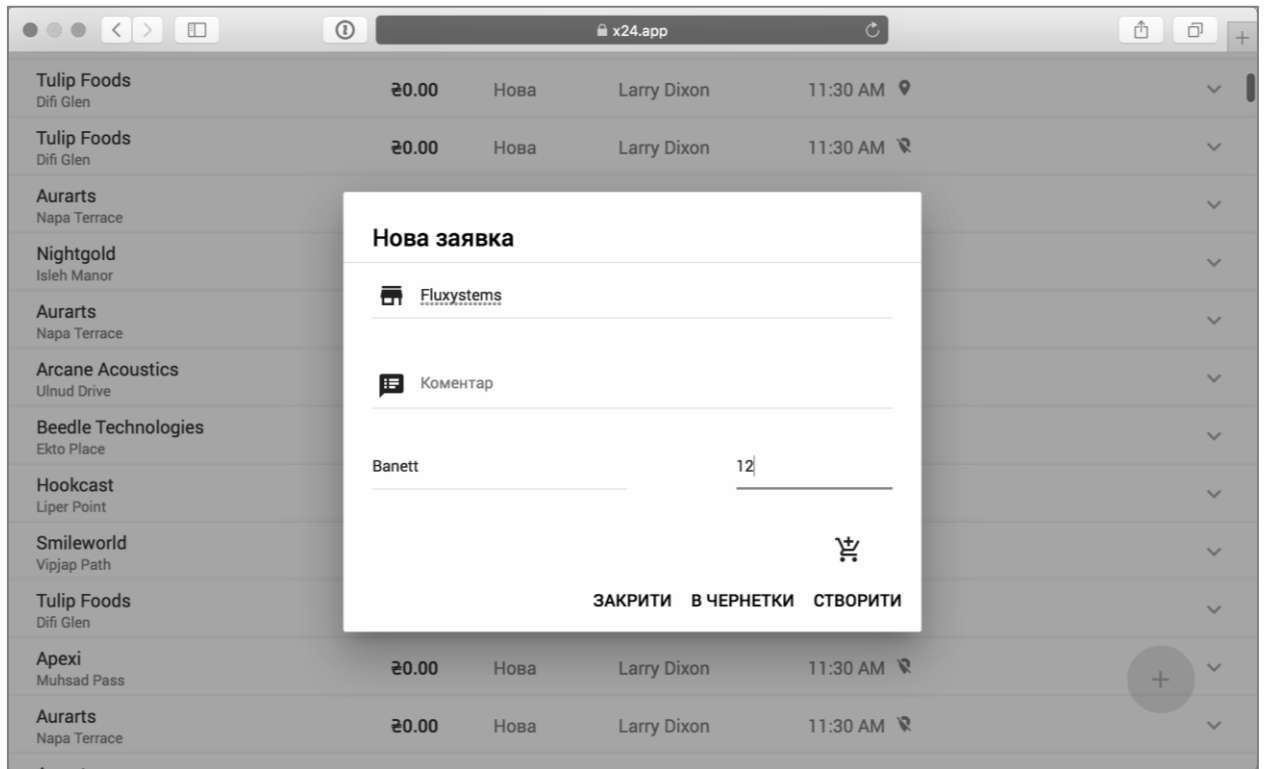


Рисунок 3.12 – Форма створення заявки

### Висновок до розділу 3

У розділі було наведено результати розробки інтерфейсу користувача, наведені зразки. Також було представлено короткий опис призначення та функціональних можливостей екранних форм.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

В даному розділі проведено аналіз потенційних небезпечних і шкідливих виробничих факторів, причин пожеж. Розглянуто заходи, які дозволяють забезпечити гігієну праці та виробничу санітарію. На підставі аналізу розроблено заходи з техніки безпеки і рекомендації з пожежної профілактики. І оскільки завданням на дипломне проектування є програмне забезпечення, то аналіз потенційно небезпечних і шкідливих виробничих факторів виконується для персонального комп'ютера, на якому передбачається реалізація та розробка системи автоматизації збору інформації.

### **4.1 Аналіз стану умов праці**

Для створення системи автоматизації збору інформації достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

Оформлення дипломного проекту з розробки системи автоматизації збору інформації за фізичним навантаженням відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово-скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявні психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- статичного;
- динамічного;

б) нервово-психічного перевантаження:

- розумового перенапруження;
- монотонності праці;
- перенапруження аналізаторів;
- емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи на ведені ДСанПіН 3.3.2.007-98[4]. Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви: - для розробників програм тривалістю 15 хвилин через кожну годину роботи.

#### **4.2 Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером**

Основними характеристиками персонального комп'ютера є наступні:

- 1) робоча напруга  $U = + 220 \pm 5\%$ ;
- 2) робочий струм  $I = 2A$ ;
- 3) споживана потужність  $P = 350 \text{ Вт}$ .

Роботу користувача розробленої підсистеми слід віднести до категорії Ia (легкі фізичні роботи). До даної категорії відносяться всі види діяльності, які виконуються сидячи, з періодичним ходінням, і не потребують фізичного напруження НПАОП 0.00. 1.28 [6]

При експлуатації даного програмного продукту існують такі небезпечні і шкідливі виробничі фактори:

1) фізичні:

- 1) підвищений рівень напруги електричної мережі, замикання якої може статися через тіло людини;
- 2) підвищена або знижена вологість повітря;
- 3) підвищена або знижена рухомість повітря;
- 4) підвищений рівень статичної електрики;
- 5) підвищена напруженість електричного поля;
- 6) відсутність або нестача природного світла;
- 7) знижена освітленість робочої зони;
- 8) підвищений рівень шуму на робочому місці;
- 9) підвищений рівень електромагнітного випромінювання;
- 10) знижена контрастність;

2) психофізіологічні;

3) фізичні перевантаження:

- 1) статичні;
- 2) динамічні;

4) нервово-психічні перевантаження:

- 1) розумове перенапруження;
- 2) монотонність праці;
- 3) перенапруження аналізаторів;
- 4) емоційні перевантаження.

### **4.3 Заходи з охорони праці**

#### **4.3.1 Загальні заходи безпеки**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання приклад деяких заходів безпеки:

1. Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:

а) якщо об'єм приміщення 20 м<sup>3</sup>, то потрібно подати не менш як 30 м<sup>3</sup>/год повітря;

б) якщо об'єм приміщення у межах від 20 до 40 м<sup>3</sup>, то потрібно подати не менш як 20 м<sup>3</sup>/год повітря;

в) якщо об'єм приміщення становить понад 40 м<sup>3</sup>, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

- зниження рівня шуму та вібрації:

а) у джерелі виникнення, шляхом застосування раціональних конструкцій, нових матеріалів і технологічних процесів;

б) звукоізолювання устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;

в) використання засобів індивідуального захисту.

2. Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також

мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;

- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

- не тягнути за мережевий кабель, щоб витягти вилку з розетки;

- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

- не залишати включені електроприлади без нагляду;

- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

#### **4.3.2 Електробезпека**

Основним небезпечним фактором при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявності електричної напруги на обладнанні.

Проходячи через тіло людини, електричний струм чинить на нього складний вплив, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної



(роздратування і збудження нервових волокон та інших органів тканин організму) дій.

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- 1) значення сили струму;
- 2) електричного опору тіла людини і тривалості протікання через нього струму;
- 3) типу і частоти струму;
- 4) індивідуальних властивостей людини і навколишнього середовища.

Приміщення для ЕОМ відноситься до приміщень без підвищеної небезпеки, тобто в приміщення, в яких відсутні умови, що створюють підвищену або особливу небезпеку. Небезпека ураження електричним струмом існує всюди, де використовуються електроустановки, тому приміщення без підвищеної небезпеки не можна назвати безпечними.

Електробезпека забезпечується:

- 1) відповідною конструкцією електроустановок;
- 2) застосуванням технічних способів і засобів захисту;
- 3) організаційними і технічними заходами.

Конструкція електроустановок відповідає умовам їх експлуатації та забезпечує захист персоналу від дотику до струмоведучих частин.

Основними технічними способами і засобами захисту від ураження електричним струмом, що використовуються окремо або в поєднанні один з одним, є:

- 1) захисне заземлення;
- 2) занулення;
- 3) вирівнювання потенціалів;
- 4) мале напруга;
- 5) електричне поділ мереж;
- 6) захисне відключення;
- 7) ізоляція струмоведучих частин;

- 8) компенсація струмів замикання на землю;
- 9) захисні пристрої;
- 10) попереджувальна сигналізація, блокування, знаки безпеки;
- 11) ізолюючі захисні та запобіжні пристосування.

### 4.3.3 Розрахунок захисного заземлення

Основними технічними способами і засобами захисту від ураження електричним струмом, що передбачаються в даному дипломному проєкті, є:

- 1) захисне заземлення,
- 2) занулення,
- 3) захисне відключення,
- 4) ізоляція струмоведучих частин.

Завдання захисного заземлення - усунення небезпеки ураження струмом у випадку дотику до корпусу та інших струмоведучих металевих частин електроустановок, які опинилися під напругою.

Розрахунок заземлюючого контуру виконується виходячи з умови:

$$R_{3y} = \frac{R_3 \cdot R_{\Pi}}{R_{\Pi} \cdot n \cdot \eta_3 + R_3 \cdot \eta_{\Pi}} \leq 4 \text{ Ом} \quad (4.1)$$

де  $R_3$  - опір заземлювача (стержня, труби, куточка і т.д.), Ом;

$R_{\Pi}$  - опір лінії, що з'єднує заземлювачі, Ом;

$n$  - кількість заземлювачів;

$\eta_3$  і  $\eta_{\Pi}$  - коефіцієнти екранування відповідно заземлювача і з'єднує смуги ( $\eta_3 = 0.2 \div 0.9$ ;  $\eta_{\Pi} = 0.1 \div 0.7$ ).

Опір заземлювача розраховується за формулою 4.2

$$R_3 = \frac{\rho}{2 \cdot \pi \cdot l} \left( \ln \frac{2 \cdot l}{d} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.2)$$

де  $\rho$  - питомий опір ґрунту (взяти з довідкової літератури);

$l$  - довжина заземлювача (для труб 2-3м, для стрижнів до 10м), м;

$d$  - діаметр заземлювача (для стрижнів 0.01 – 0.03м, для труб 0.03 – 0.05м);

$t$  - відстань від середини забитого в ґрунт заземлювача до рівня землі (необхідно враховувати, що відстань від верхнього кінця заземлювача до поверхні землі має бути не менше 0.5), м.

Розрахуємо опір заземлювача:

$$R_3 = \frac{60}{2 \cdot \pi \cdot 3} \left( \ln \frac{2 \cdot 3}{0.03} + \frac{1}{2} \cdot \ln \frac{4 \cdot 1 + 3}{4 \cdot 1 - 3} \right) = 19.96 \quad (4.3)$$

Опір лінії, що з'єднує заземлювачі розраховується за формулою 4.4

$$R_{\Pi} = \frac{\rho}{2 \cdot \pi \cdot l} \cdot \ln \frac{2 \cdot L^2}{b \cdot t'}, \quad (4.4)$$

де  $L$  - довжина лінії, що з'єднує заземлювачі (при контурному заземленні вона приблизно дорівнює периметру виробничої будівлі), м;

$b$  - ширина смуги (0.03 - при прокладанні всередині будівлі і 0.05 - при прокладанні поза будівлею), м;

$t$  - глибина заземлення від рівня землі (0.5 м).

Розрахуємо опір лінії, що з'єднує заземлювачі

$$R_{\Pi} = \frac{60}{2 \cdot \pi \cdot 3} \cdot \ln \frac{2 \cdot 50^2}{0.03 \cdot 0.5} = 14.37 \quad (4.5)$$

Необхідна кількість заземлювачів, розраховується за формулою 4.6

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3}, \quad (4.6)$$

де 4 - допустимий загальний опір;

2 - коефіцієнт сезонності.

Розрахуємо необхідну кількість заземлювачів,

$$n = \frac{2 \cdot 19.9}{4 \cdot 0.5} = 19.9 \approx 20 \quad (4.7)$$

Округлимо результат в більшу сторону і отримуємо необхідну кількість заземлювачів - 20. Маючи всі необхідні дані розрахуємо опір заземлюючого контуру.

$$R_{\text{зз}} = \frac{19.96 \cdot 14.37}{14.37 \cdot 20 \cdot 0.5 + 19.96 \cdot 0.4} = 1.89 \leq 4 \text{ Ом} \quad (4.8)$$

Опір заземлюючого контуру 1.89 Ом, що відповідає умові  $R_{\text{зз}} < 4 \text{ Ом}$ .

#### 4.4 Заходи, що забезпечують виробничу санітарію та гігієну праці

##### 4.4.1 Мікроклімат

Важкість праці характеризує сукупну дію всіх елементів, складових умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому представленні поняття тяжкості праці однаково застосовні як до розумової, так і до фізичної праці. Згідно [1] тяжкість роботи персоналу, який обслуговує ЕОМ, відноситься до легкої категорії 1б (роботи, виконувані сидячи, не вимагаючи систематичного фізичного напруження і перенесення важких предметів) [1]. Загальні санітарно-гігієнічні вимоги до повітря робочої зони. Оптимальні норми мікроклімату в робочій зоні, що забезпечуються для робіт легкої категорії 1б приведені в таблиці 4.1.

Таблиця 4.1 - Оптимальні норми мікроклімату

Період рок	Температура, °С	Відносна вологість, %	Швидкість вітру, м/с, не більше
Холодний	21 - 23	60 – 40	0.1
Теплий	22 - 24	60 - 40	0.2

##### 4.4.2 Освітлення

Світло є природною умовою існування людини . Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Гарне

освітлення діє тонізує, створює гарний настрій, поліпшує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла у світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряд точності (середня точність). При цьому нормована освітленість на робочому місці ( $E_n$ ) дорівнює 200 лк. Джерелом природного освітлення є сонячне світло. У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2015 [3]

Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє [3]

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \cdot S_n \quad (4.9)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$ .

$$S_n = a \times b = 4 \times 5 = 20 \text{ м}^2,$$

$$S = 1/8 \times 20 = 2.5 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S = 2.5 \text{ м}^2$ . Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-

прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 4м, ширина 4м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80Вт) з світловим потоком 5400лм кожна. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників  $n$  визначається по формулі (4.10):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M} \quad (4.10)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 20$  м<sup>2</sup>;

$Z$  – поправочний коефіцієнт світильника ( $Z = 1.15$  для ламп розжарювання та ДРЛ;

$Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1.1;

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1.5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0.575;

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.10), отримуємо:

$$n = \frac{300 \times 20 \times 1.1 \times 1.5}{5400 \times 0.575 \times 2} \approx 1.6 \quad (4.11)$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160Вт, напругою 220В.

#### 4.5 Рекомендації щодо пожежної безпеки

Виникнення пожежі можливо, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Для оцінки пожежної небезпеки слід проаналізувати ймовірність взаємодії цих трьох чинників.

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С;

2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, те

3) склотекстоліт ДЦ - матеріал друкованих плат, трудногорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

4) пластикат кабельний No.489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;

5) деревина - будівельний і оздоблювальний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання

255 °С, температура самозаймання 399 °С.

Згідно НАПБ Б. 03.002-2007 таке приміщення належить до категорії "В" (пожежонебезпечної) [5].

Простору всередині приміщень в межах яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали у відповідності з ПУЕ відносяться до пожежонебезпечної зони класу П-Па.

Потенційними джерелами запалювання можуть бути:

- 1) іскри і дуги короткого замикання;
- 2) електрична іскра при замиканні і розмиканні ланцюгів;
- 3) перегіви від тривалого перевантаження;
- 4) відкритий вогонь і продукти горіння;
- 5) наявність речовин, нагрітих вище температури самозаймання;
- 6) розряд на статичну електрику.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки обладнання;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згоряння, що виділяються під час пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор та ін.

При горінні пластмас, крім звичних продуктів згоряння, виділяються різні продукти термічного розкладання: хлорангідридні кислоти; формальдегіди; хлористий водень; фосген; синильна кислота; аміак; фенол; ацетон; стирол. Пожежо- вибухонебезпечність речовин і матеріалів. Номенклатура показників і методи їх визначення ГОСТ 12.1.044 - 89 ЕСБТ[2].

Для захисту персоналу від впливу небезпечних і шкідливих факторів пожежі проектом передбачається застосування промислового протигаза фільтруючого з коробкою марки В (жовтий).

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем вентиляції та кондиціонування, розвиненою системою електроживлення ЕОМ. Небезпека загорання в ЕОМ пов'язана з великою кількістю щільно розташованих на платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80 ... 100 °С), що може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання.

Пожежна безпека при застосуванні ЕОМ забезпечується:



- 1) системою запобігання пожежі:
- 2) системою протипожежного захисту:
- 3) організаційно-технічними заходами.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі та важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворенню (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонами відповідно до ПУЕ;
- 2) застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалювання;
- 3) виключення можливості появи іскрового розряду в займистою середовищі з енергією, яка дорівнює і вище мінімальної енергії запалювання.

#### **Висновки до розділу 4**

У розділі "Охорона праці" виконаний аналіз потенційних небезпек при роботі із засобами обчислювальної техніки, на підставі якого розроблено заходи з техніки безпеки, заходи, що забезпечують виробничу санітарію та гігієну праці, розрахунки природного та штучного освітлень, рекомендації з пожежної профілактики, які підтверджені відповідними розрахунками.

## ВИСНОВКИ

Метою даної дипломної роботи було створення системи автоматизації роботи торговельних представників.

У першому розділі проекту були поставлені цілі і завдання, необхідні для реалізації проекту, де ми з'ясували наскільки актуальна тема проекту. Були розглянуті існуючі аналоги, аналіз яких допоміг у визначенні напрямку розробки прототипу і дизайну.

У другому розділі було описано архітектуру системи, її компоненти та взаємодію клієнтських додатків із системою. Задokumentували програмний прикладний інтерфейс системи та середовище її розробки та виконання.

У третьому розділі було наведено зразки зовнішнього вигляду додатків системи, коротко описано взаємодію із нею користувача.

Таким чином у результаті проекту було створено базову систему автоматизації збору роботи торговельних представників.

На жаль на даному етапі система охоплює лише частину запланованого функціоналу. Через брак часу було реалізовано тільки ручний режим синхронізації даних мобільного додатку та лише частково було перекладено інтерфейс користувача з англійської мови. Доробка цих та додавання нових функцій системи вже після запуску системи в експлуатації.

## СПИСОК ЛІТЕРАТУРИ

1. ГОСТ 12.1.005-88 "ССБТ Загальні санітарно-гігієнічні вимоги до повітря робочої зони".
2. ГОСТ 12.1.044-89 "ССБТ. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения."
3. ДБН В.2.5-28:2015 "Природне і штучне освітлення."
4. ДСанПіН 3.3.2.007-98 "Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин."
5. НАПБ Б.03.002-2007 "Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою."
6. НПАОП 0.00-1.28-10 "Правила охорони праці під час експлуатації електронно- обчислювальних машин."
7. About PostgreSQL [Електронний ресурс]. – Точка доступу: URL: <https://www.postgresql.org/about/>
8. Android Studio [Електронний ресурс]. – Точка доступу: URL: [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)
9. Base64 [Електронний ресурс]. – Точка доступу: <https://en.wikipedia.org/wiki/Base64>
10. Dart (programming language) [Електронний ресурс]. – Точка доступу: [https://en.wikipedia.org/wiki/Dart\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language))
11. Dynamics 365 - Wikipedia." [https://en.wikipedia.org/wiki/Dynamics\\_365](https://en.wikipedia.org/wiki/Dynamics_365). Accessed 17 May. 2018
12. EAN/UPC barcodes [Електронний ресурс]. – Точка доступу: <https://www.gs1.org/standards/barcodes/ean-upc>

13. Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). University of California, Irvine. This chapter introduced the Representational State Transfer (REST) architectural style for distributed hypermedia systems. REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. [Электронный ресурс]. – Точка доступа: URL: [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
14. Flutter (software) [Электронный ресурс]. – Точка доступа: [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))
15. Geohash [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/Geohash>
16. Git [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/Git>
17. Unicorn [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/Unicorn>
18. HTML [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/HTML>
19. ISO 4217 [Электронный ресурс]. – Точка доступа: URL: [https://en.wikipedia.org/wiki/ISO\\_4217](https://en.wikipedia.org/wiki/ISO_4217)
20. ISO 8601 [Электронный ресурс]. – Точка доступа: URL: [https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601)
21. Most Widely Deployed SQL Database Estimates". SQLite.org. Retrieved May 11, 2011. [Электронный ресурс]. – Точка доступа: URL: <https://sqlite.org/mostdeployed.html>
22. Nginx [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/Nginx>

23. Owens, Michael (2006). "Chapter 4: SQL". In Gilmore, Jason; Thomas, Keir. *The Definitive Guide to SQLite*. D. Richard Hipp (foreword), Preston Hagar (technical reviewer). Apress. p. 133. ISBN 978-1-59059-673-9. Retrieved 30 December 2014. [Электронный ресурс]. – Точка доступа: URL: <https://books.google.com/books?id=VsZ5bUh0XAkC&pg=PA133>
24. PostgreSQL [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/PostgreSQL>
25. Representational state transfer [Электронный ресурс]. – Точка доступа: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)
26. RFC 1945 [Электронный ресурс]. – Точка доступа: URL: <https://tools.ietf.org/html/rfc1945>
27. RFC 6749 - The OAuth 2.0 Authorization Framework". Retrieved 2016-05-15. [Электронный ресурс]. – Точка доступа: URL: <http://tools.ietf.org/html/rfc6749>
28. Salesforce.com - Wikipedia. Retrieved May 9, 2018, from <https://en.wikipedia.org/wiki/Salesforce.com>
29. Sass (stylesheet language) [Электронный ресурс]. – Точка доступа: URL: [https://en.wikipedia.org/wiki/Sass\\_\(stylesheet\\_language\)](https://en.wikipedia.org/wiki/Sass_(stylesheet_language))
30. Scopatz, Anthony; Huff, Kathryn D. (2015). *Effective Computation in Physics*. O'Reilly Media, Inc. p. 351. ISBN 9781491901595. Archived from the original on 7 May 2016. Retrieved 20 April 2016.
31. SQLite [Электронный ресурс]. – Точка доступа: URL: <https://en.wikipedia.org/wiki/SQLite>
32. Technical Overview - Flutter". flutter.io. Retrieved 2017-12-13. [Электронный ресурс]. – Точка доступа: <https://flutter.io/technical-overview/>
33. Time Zone Database [Электронный ресурс]. – Точка доступа: URL: <https://www.iana.org/time-zones>
34. Tony Mobily (5 January 2012). "Interview with Igor Sysoev, author of Apache's competitor NGINX". *Free Software Magazine*. Retrieved 18

- October 2013. [Електронний ресурс]. – Точка доступу: URL: [http://www.freesoftwaremagazine.com/articles/interview\\_igor\\_sysoev\\_auth\\_or\\_apaches\\_competitor\\_nginx](http://www.freesoftwaremagazine.com/articles/interview_igor_sysoev_auth_or_apaches_competitor_nginx)
35. Torvalds, Linus (2005-04-07). "Re: Kernel SCM saga." linux-kernel (Mailing list). "So I'm writing some scripts to try to track things a whole lot faster." [Електронний ресурс]. – Точка доступу: URL: <https://marc.info/?l=linux-kernel&m=111288700902396>
36. "What is PostgreSQL?". PostgreSQL 9.3.0 Documentation. PostgreSQL Global Development Group. [Електронний ресурс]. – Точка доступу: URL: <https://www.postgresql.org/docs/current/static/intro-what-is.html> Retrieved 2013-09-20.
37. Whitson Gordon. "Understanding OAuth: What Happens When You Log Into a Site with Google, Twitter, or Facebook". Retrieved 2016-05-15. [Електронний ресурс]. – Точка доступу: URL: <http://lifelifehacker.com/5918086/understanding-oauth-what-happens-when-you-log-into-a-site-with-google-twitter-or-facebook>
38. Zoho — Вікіпедія. Retrieved May 9, 2018, from <https://uk.wikipedia.org/wiki/Zoho>

## ДОДАТОК А ЛІСТИНГ НАБОРУ ТЕСТІВ ДЛЯ ПІДСИСТЕМИ ЗАЯВОК

```

1. class OrderTestCase(BaseApiTestCase):
2.     def test_can_create_order_object(self):
3.         order = self.assertIsNotNone(any_order())
4.         # any_order_line(order=order)
5.
6.     def test_user_can_index_organization_orders(self):
7.         org_owner = any_user()
8.         expiration_date = test_subscription_end()
9.         org = models.Organization.objects\
10.             .create_for_user(org_owner, any_string(), expiration_date)
11.         org_user = org.organization_users.create(user=any_user())
12.         expected_orders = [any_order(organization=org) for x in range(5)]
13.
14.         self.client.force_login(org_user.user)
15.         path = reverse('order-list')
16.         response = self.client.get(path)
17.         self.assertEqual(response.status_code, status.HTTP_200_OK)
18.         self.assertIn('results', response.data)
19.         products = response.data['results']
20.         products_ids = {o['id'] for o in products}
21.         self.assertEqual({str(o.uid) for o in expected_orders}, products_ids)
22.
23.     def test_user_can_create_order(self):
24.         org_owner = any_user()
25.         expiration_date = test_subscription_end()
26.         org = models.Organization.objects\
27.             .create_for_user(org_owner, any_string(), expiration_date)
28.         customer = org.customers.create(display_name=any_string())
29.         org_user = org.organization_users.create(user=any_user())
30.         category = org.price_categories\
31.             .create(display_name=any_string(), is_active=True)
32.         products = []
33.         for i in range(5):
34.             p = any_product(
35.                 organization=org, status=constants.PRODUCT_STATUS_ACTIVE)
36.             products.append(p)
37.             any_price(price_category=category, product=p, organization=org)
38.         data = {
39.             'organization_id': org.uid,
40.             'owner_id': org_user.uid,
41.             'customer_id': customer.uid,
42.             'category_id': category.uid,

```

```

43.         'location': 'u4pruydqqvj',
44.         'status': dict(
45.             constants.ORDER_STATUS_NAMES)[constants.ORDER_STATUS_PROCESSING],
46.         'lines': [dict(product=p.uid, quantity=1) for p in products]
47.     }
48.
49.     self.client.force_login(org_user.user)
50.     path = reverse('order-list')
51.     response = self.client.post(path, data)
52.     self.assertEqual(response.status_code, status.HTTP_201_CREATED)
53.     self.assertIn('status', response.data)
54.     self.assertEqual(data['status'], response.data['status'])
55.     self.assertIn('location', response.data)
56.     self.assertEqual(data['location'], response.data['location'])
57.
58.     def test_order_owner_has_change_order_permission(self):
59.         org_owner = any_user()
60.         expiration_date = test_subscription_end()
61.         org = models.Organization.objects\
62.             .create_for_user(org_owner, any_string(), expiration_date)
63.         customer = org.customers.create(display_name=any_string())
64.         org_user = org.organization_users.create(user=any_user())
65.         category = org.price_categories\
66.             .create(display_name=any_string(), is_active=True)
67.         order = any_order(
68.             organization=org, owner=org_user, customer=customer,
69.             status=constants.ORDER_STATUS_DRAFT, price_category=category)
70.         self.assertTrue(order.owner.user.has_perm(fpn('change_order')))
71.         self.assertTrue(order.owner.user.has_perm(fpn('change_order'), order))
72.
73.     def test_user_can_update_order_status(self):
74.         org_owner = any_user()
75.         expiration_date = test_subscription_end()
76.         org = models.Organization.objects\
77.             .create_for_user(org_owner, any_string(), expiration_date)
78.         customer = org.customers.create(display_name=any_string())
79.         org_user = org.organization_users.create(user=any_user())
80.         category = org.price_categories\
81.             .create(display_name=any_string(), is_active=True)
82.         order = any_order(
83.             organization=org, owner=org_user, customer=customer,
84.             status=constants.ORDER_STATUS_DRAFT, price_category=category)
85.         data = {
86.             'status': dict(
87.                 constants.ORDER_STATUS_NAMES)[constants.ORDER_STATUS_PENDING],
88.         }
89.

```



```

90.         self.client.force_login(org_user.user)
91.         path = reverse('order-detail', args=[order.uid])
92.         response = self.client.patch(path, data)
93.         self.assertEqual(response.status_code, status.HTTP_200_OK)
94.         self.assertIn('status', response.data)
95.         self.assertEqual(data['status'], response.data['status'])
96.
97.     def test_user_can_update_order_with_lines(self):
98.         org_owner = any_user()
99.         expiration_date = test_subscription_end()
100.        org = models.Organization.objects\
101.            .create_for_user(org_owner, any_string(), expiration_date)
102.        customer = org.customers.create(display_name=any_string())
103.        org_user = org.organization_users.create(user=any_user())
104.        category = org.price_categories\
105.            .create(display_name=any_string(), is_active=True)
106.        order = any_order(
107.            organization=org, owner=org_user, customer=customer,
108.            status=constants.ORDER_STATUS_DRAFT, price_category=category)
109.        products = []
110.        for i in range(5):
111.            p = any_product(
112.                organization=org, status=constants.PRODUCT_STATUS_ACTIVE)
113.            products.append(p)
114.            any_price(price_category=category, product=p, organization=org)
115.        data = {
116.            'lines': [dict(product=p.uid, quantity=1) for p in products]
117.        }
118.
119.        self.client.force_login(org_user.user)
120.        path = reverse('order-detail', args=[order.uid])
121.        response = self.client.patch(path, data)
122.        self.assertEqual(response.status_code, status.HTTP_200_OK)
123.

```

## ДОДАТОК Б ПРЕЗЕНТАЦІЯ

Міністерство освіти і науки України  
Східноукраїнський національний університет ім. В.Дала

Система автоматизації роботи  
торгівельних представників

# “Мобільні продажі”

Студент групи КІ-14з  
Приймак С.О.

Керівник проекту  
Скарга-Бандурова І.С.

## Мета роботи

Збір заявок торговельними агентами за допомогою мобільного додатку

Автоматизована передача заявок в офіс

## Що саме покращуємо

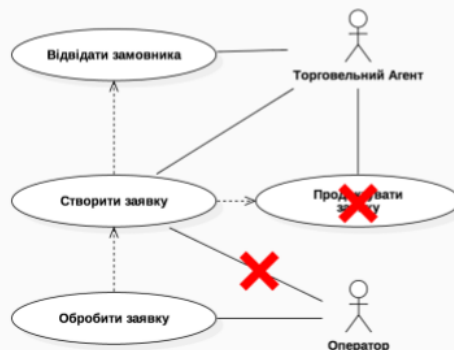
1. Агент відвідує замовника та записує замовлення "на папірець"
2. Агент диктує замовлення оператору по телефону
3. Оператор створює і обробляє заявку



## Що саме покращуємо

1. Агент відвідує замовника та створює заявку у додатку
2. При можливості дані синхронізуються із сервером
3. Оператор обробляє заявку

Агент не має більше записувати та диктувати замовлення. Оператор більше не повинен створювати заявки.



## Існуючі рішення



Microsoft  
Dynamics 365



## Існуючі рішення



Microsoft  
Dynamics 365



## Структура системи

Сервер додатку:

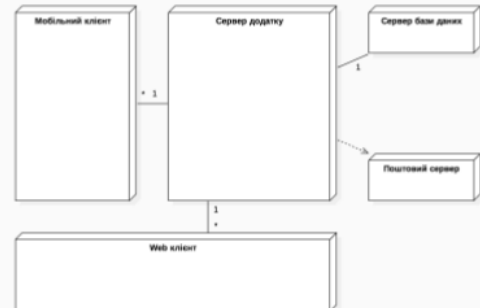
- Прийом, обробка та збереження даних системи

Мобільний клієнт:

- Автономність
- Створення заявок та обмін із сервером

Web клієнт:

- Створення та обробка заявок



## Стек технологій

Сервер додатку:

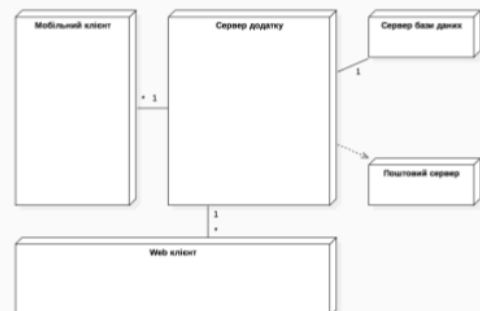
- Nginx, Linux, PostgreSQL, Django, Python

Мобільний клієнт:

- Android/iOS, SQLite, Flutter, Dart

Web клієнт:

- Chrome/Firefox/Safari, Angular, Dart



## Підсистеми

### Автентифікація

- перевірка повноважень користувача

### Користувачі та організації

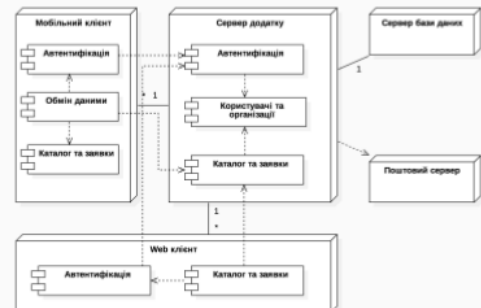
- керування обліковими записами

### Каталог та замки

- керування каталогом товарів та замовленнями

### Обмін даними

- синхронізація даних із сервером



## Що далі?

- Автоматична синхронізація
- Сповіщення користувачів про зміну даних
- Покращення інтерфейсу користувача
- Списки завдань
- Підтримка маршрутів
- Інтеграція з обліковими системами (1С тощо)
- ...