

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Випускна робота

на тему:

**«Мобільний додаток на базі ОС Android
«Доставка води» »**

Завідувач випускаючої кафедри

Довбиш А.С.

Керівник роботи

Тиркусова Н.В

Студента групи ІН-71/2

Сігало А.І.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 г.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІН-71 спеціальності “Комп'ютерні науки” денної форми навчання Сітало Алли Іванівни

Тема: "Мобільний додаток на базі ОС Android «Доставка води»"

Затверджена наказом по СумДУ

№ _____ от _____ 2021 г.

Зміст пояснювальної записки: 1) огляд методів розробки додатку; 2) аналізування та огляд продуктів-аналогів; 3) постановка задачі; 4) проектування мобільного додатку; 5) реалізація та перевірка результату роботи.

Дата видачі завдання “ _____ ” _____ 2021 г.

Керівник випускної роботи _____ Тиркусова Н.В.

Завдання прийняв до виконання _____ Сітало А.

РЕФЕРАТ

Записка: 55 стор., 17 рис., 1 табл., 1 додаток, 18 джерел.

Об'єкт дослідження — процес взаємодії мобільного додатку та центральної інформаційної бази, забезпечення роботи системи в режимі користувача: реалізація механізмів авторизації, перегляду доступних товарів, створення замовлень та їх моніторинг.

Мета роботи — розробка мобільного додатку на основі операційної системи Android для інтернет-служби, що спеціалізується на постачанні питної води.

Методи дослідження — створення програмного коду та інтерфейсу за допомогою середовища 1С Підприємства, методи об'єктно-орієнтованого програмування, HTTP методи для взаємодії з центральною інформаційною базою.

Результати — розроблено мобільний додаток на операційній системі Android, який дозволяє користувачеві замовляти товари та послуги, а саме воду та її доставку, забезпечена взаємодія мобільного з центральною базою УНФ (управління невеликою фірмою). Розроблений застосунок реалізовано за допомогою програмного середовища 1С.

МОБІЛЬНИЙ ДОДАТОК, ОПЕРАЦІЙНА СИСТЕМА ANDROID,
HTTP МЕТОДИ, 1С, ДОСТАВКА ВОДИ, JSON, ПРОГРАМНЕ
СЕРЕДОВИЩЕ, WEB-SERBER

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1 Огляд проблемної області	6
1.2 Огляд подібних рішень.....	7
1.3 Постановка задачі.....	9
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ	11
2.1 Методи для взаємодії мобільного та бази	11
2.2 Публікація бази за допомогою ІІS.....	11
2.3 Передача даних в форматі JSON	12
2.4 Проектування мобільного додатку.....	13
2.5 Проектування мобільного додатку.....	14
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	19
3.1 Розробка моделі інформаційної системи.....	19
3.2 Програмна реалізація.....	20
ВИСНОВКИ.....	33
СПИСОК ЛІТЕРАТУРИ.....	34
ДОДАТОК А.....	36

ВСТУП

На сьогоднішній день засоби мобільного зв'язку вже не є чимось надзвичайним. Вони на разі виконують не тільки свою основну функцію, котра полягає в тому, щоб бути засобом спілкування, а й захопили набагато більший спектр функцій. Нещодавно в нашому житті з'явилися смартфони, котрі мають широке призначення, виконуючи функцію, як плеєра, так і фотоапарата, можливість доступу до Інтернету та багато іншого. Правду кажучи, звичайний, нічим не примітний смартфон, став мовби малою копією комп'ютера, котрий з кожним днем стає все більш невід'ємною частиною нашого життя.

Нині все більше та більше смартфонів, планшетів та інших видів пристроїв, котрі допомагають нам як у повсякденному житті, так і в роботі, навчанні, вони випускаються на базі ОС (операційна система) Android.

Операційна система Android підтримує різноманітні прибори та пристрої великої кількості виробників. Також особливістю даної ОС є доступність для розробників, засоби для розробки на даній платформі є цілком безкоштовні, на відміну від операційної системи, на якій працює iPhone, та не потребують значного базового капіталу для розробки. Також операційна система Android надає можливість використовувати велику кількість різних бібліотек для роботи з зовнішніми ресурсами, такими як Google Map API, Yandex MapKit та багато інших. А ось, наприклад, в Windows Phone Mobile такої можливості не має.

В рамках даної дипломної роботи буде описано створення мобільного додатку на базі ОС Android, орієнтованого, зокрема, на доставку води. Описані вище переваги визначають таку велику кількість пристроїв саме на базі ОС Android, що є ознакою масового використання та популярності. Таким чином, додатки стають все більш оснащеними новим функціоналом, котрий робить замовлення і доставку води максимально зручними для користувача.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Огляд проблемної області

Мобільний додаток (застосунок) – це програмне забезпечення, головним призначенням якого є робота на планшетах, телефонах та інших портативних пристроях. В основному додатки встановлюються на сам пристрій або завантажуються і встановлюються через онлайн магазини мобільних додатків. Основними прикладами таких є Google Play, для власників Android, App Store для IOS, Windows Phone Store та багато інших, до того ж завантажувати мобільні додатки можна як безкоштовно, так і за окрему плату.

Спершу додатки розглядалися як інструмент, що контролював та оперував загальними потоками інформації, такі як календар, контакти, погода, пошта та інше. Проте згодом поступове розширення його функціоналу, завдяки попиту споживачів, призвело до появи нових функціональних можливостей, а разом із ними і інструментів для їх реалізації для розробників програмного забезпечення. [1]

Для того, щоб створити додаток на операційній системі Android необхідно використовувати відповідне інтегроване середовище розробки. Розглянемо три варіанти найбільш популярних середовищ, таких як, IntelliJ IDEA, Android Studio та 1С платформа для розробки мобільних додатків. [2]

IntelliJ IDEA – це середовище компанії JetBrains, дає змогу програмувати на декількох мовах, має досить широкі можливості, автоматичний підбір значень. Серед недоліків можна відмітити те, що версія є не безкоштовною для використання.

Офіційним середовищем для розробки під Android вважається Android Studio, котрий створений на основі попереднього середовища та допрацьований певними нововведеннями й завернутий в більш зрозумілий та приємний інтерфейс. Характерною рисою є те, що розробка можлива лише для однієї операційної системи. [3]

1С Платформа Підприємство – це технологія, що дозволяє вести розробку мобільних додатків під управлінням різних операційних систем на мові програмування 1С. Для отримання дистрибутивів використовується спеціальне прикладне рішення – Складальник мобільних додатків.

Для поставленої задачі нам найбільш доречно буде використати розробку на платформі 1С. Так як більшість підприємств веде облік в програмних продуктах 1С, в мобільному додатку можна легко створити структуру документів, подібну структурі бази центральної інформаційної бази 1С. [4,5]

Під час розробки програмного забезпечення на базі Android необхідно враховувати існуючі можливості та обмеження. Одним з обмежень є, наприклад, те що, мобільні телефони мають не надто потужний процесор та акумулятор в порівнянні з комп'ютером. Проте в них значно краще розвинуті інші функціональні аспекти – камера з більш кращою якістю фотографій та відеозапис, а також можливість використання геолокації для визначення місцезнаходження користувача. Розробникам мобільних додатків необхідно враховувати розміри екрану, які мають широкий спектр, конфігурацію обладнання, різні технічні характеристики та швидко адаптацію під зміни в кожній платформі. [6]

Для тестування мобільних додатків використовують емулятори, а уже після проходження всіх тестових вимог перевіряються у середовищі подальшого використання. За допомогою емулятора розробник може відтворити роботу будь-якого пристрою, фізичного доступу до котрого він не має. [7]

1.2 Огляд подібних рішень

В процесі дослідження було проведено пошук та аналіз товарів-аналогів та виявлено декілька мобільних додатків, котрі розраховані на замовлення та доставку води, такі як «My water» та «Ера водоля». Було здійснено огляд цих додатків для аналізування та виявлення слабких та сильних сторін для

подальшого використання цих даних при розробці продукту «Доставки ВОДИ».

«My water» - це сервіс, який пропонує доставку питної води всього декількох брендів. Інтерфейс мобільного додатку досить зрозумілий та привабливий для користувача, виконаний згідно всіх вимог сучасних трендів, але даний додаток здебільшого орієнтований на Київ, та інші великі міста. При використанні часто вимагає обрати місце знаходження та з'являється надокучлива реклама. Інтерфейс додатку «My water» продемонстрований на Рисунку 1.1.

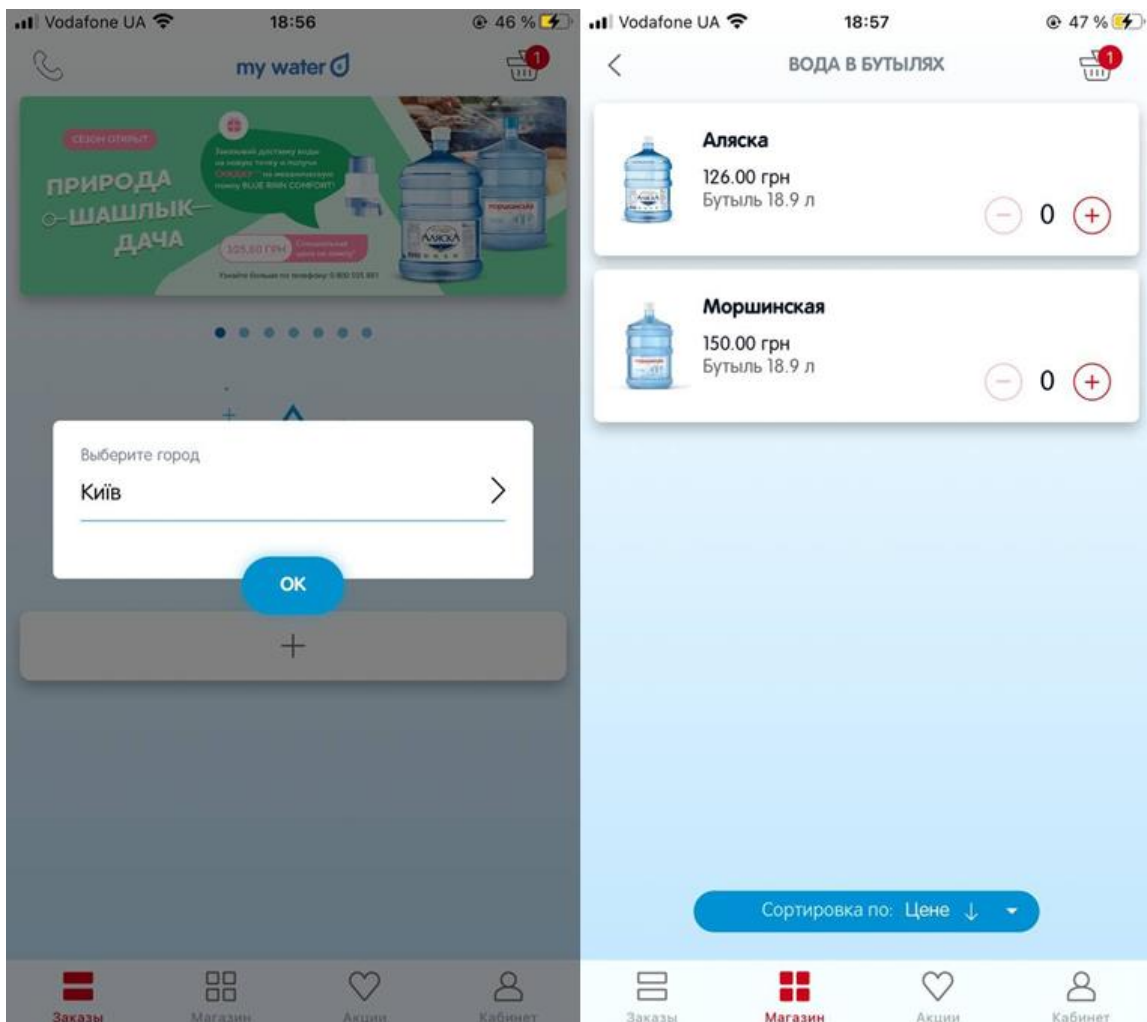


Рисунок 1.1 – Інтерфейс додатку «My water»

«Ера водоля» - це мобільний додаток для замовлення води в місті Суми, розрахований на наше місто та область, є більш географічно орієнтований в порівняльній з першим. Інтерфейс додатку не зовсім зручний зі сторони UX, асортимент товарів також не дуже широкий. Зовнішній вид мобільного додатку «Ера водоля» можна переглянути на Рисунку 1.2.

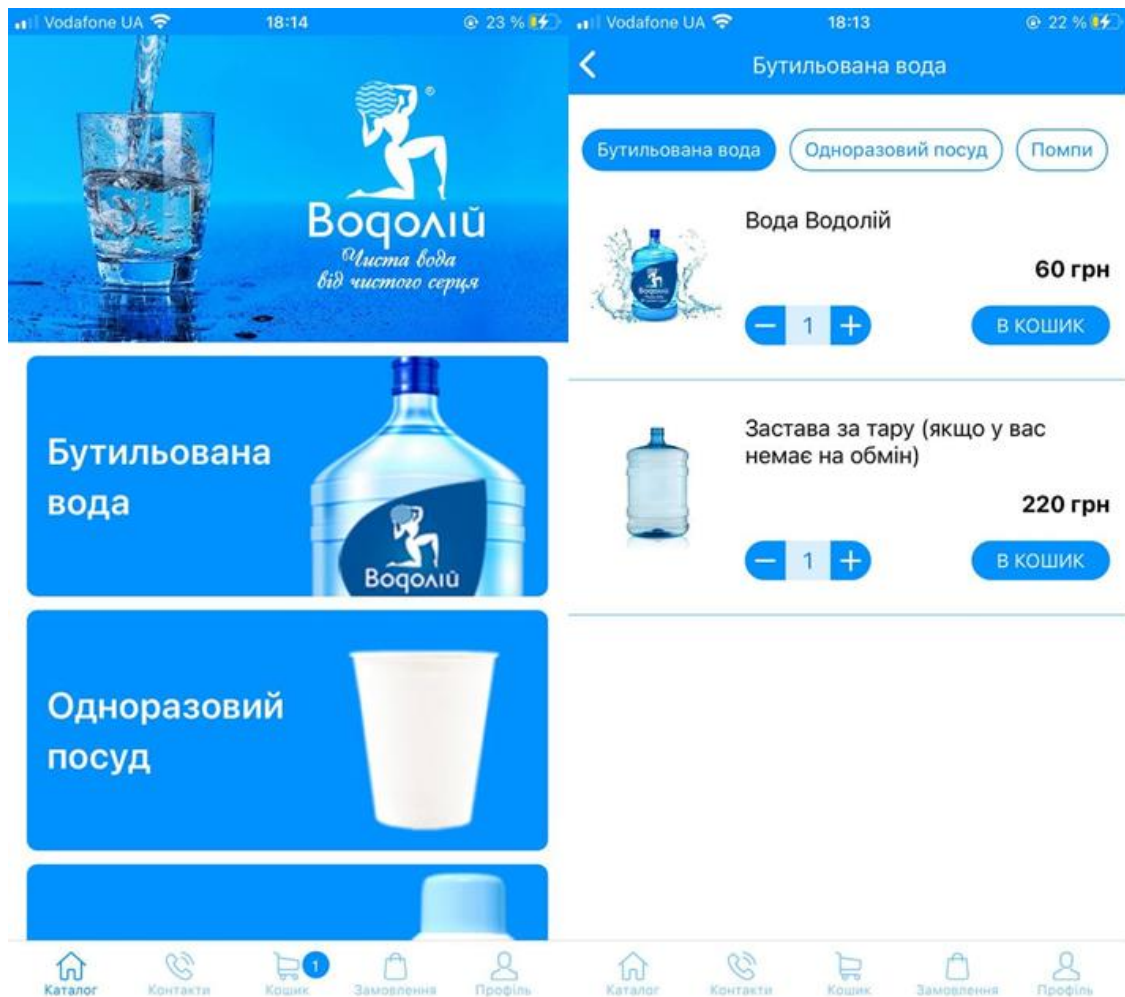


Рисунок 1.2 – Інтерфейс «Ера водоля»

1.3 Постановка задачі

Завданням дипломної роботи є розробка мобільного додатку «Доставка води», функціонал котрого дозволить здійснювати замовлення води у замовника за допомогою даної системи.

Даний додаток містить в собі такі розділи меню:

- Вітрина додатку

- Реєстрація та авторизація
- Заовлення (створення нового заовлення, список заовлень та шаблонів)
- Контакти

На вітрині відображається перелік товарів з фото та ціною, де користувач може обрати потрібний йому товар та обрати його кількість за допомогою іконок «+» щоб збільшити, та «-» для зменшення кількості на одиницю.

Реєстрація користувача здійснюється за номером телефону, котрий вказується у відповідному полі, після натискання кнопки реєстрації даний номер надсилається на обробку в базу для подальшої реєстрації.

Авторизація здійснюється після отримання користувачем його особистого логіну та паролю, вказавши ці дані у відповідні поля, користувач отримує повноцінний доступ до мобільного додатку, де може здійснювати покупку товарів. Авторизуватися можливо з різними ролями, які визначаються при реєстрації, згідно яких формується список контрагентів на ім'я котрих можливо здійснювати заовлення.

Для здійснення заовлень необхідно обрати перелік товарів, вказавши їх кількість, обрати адрес для доставки води, спосіб оплати, інтервал часу в котрий необхідно виконати доставку, дату та додатковий коментар до заовлення. Існує можливість переглядати раніше створені заовлення та заовлення збережені як шаблон.

Для зв'язку з call-центром та технічною підтримкою є відповідний розділ контакти, де вказані дані для дзвінка, та банківські реквізити для оплати заовлення. Даний розділ є доступний навіть для не авторизованого користувача для можливості зв'язку з службою по доставці води, що є досить зручно.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Методи для взаємодії мобільного та бази

Під час виконання даної роботи було використано середовище для розробки мобільного додатку на платформі Android. При його розробці для взаємодії мобільного з центральною базою використовувалися http методи, для передачі даних з мобільного додатку до центральної бази, та навпаки. Основними методами були POST та GET методи.

HTTP – це протокол передачі даних, який використовують в комп'ютерних мережах. Назва від англійського скорочення назви Hyper Text Transfer Protocol, протокол передачі гіпертекстових документів. Тип HTTP-запиту, який надсилається вказує серверу на те, яку саме дію ми хочемо виконати з ресурсом.

При відправці методу GET здійснюється запит інформації із зазначеної бази і не впливає на вміст ресурсу. Запит цього методу доступний для кешування та додавання в закладки.

При відправці методу POST здійснюється відправка даних, яка в свою чергу уже може впливати на вміст бази до котрої був відправлений запит, даний метод не має обмеження в обсязі. Різницею між методами GET та POST є те що в другому, не можливе кешування, додавання в закладки, тобто вони не залишають слідів в браузері. [8]

Кожен запит чи відповідь має такі три головні частини:

- стартовий рядок;
- заголовки;
- тіло повідомлення, в ньому описуються дані запиту, запитаний ресурс

чи опис проблеми, коли запит не можливо виконати.

2.2 Публікація бази за допомогою IIS

Для розміщення центральної бази в Інтернеті, було використано сервер розробки компанії Microsoft - Internet Information Services.

IIS складається з групи інтернет-серверів, в склад якої входить і веб-сервер та FTP-сервер. IIS містить додатки для побудови і управління веб-сайтами, машину для пошуку і підтримки розробників веб-додатків, у котрих є доступ до баз даних.

Компанія Microsoft створила IIS для розробки веб-сервісів з використанням архітектури облікових записів користувачів, що надається доменом чи службою Active Directory. Завдяки даному механізму можна використовувати бази даних користувачів котрі вже існують, а не створювати нові.

Переваги такого механізму вагомі, особливо якщо уявити собі потреби великої організації з величезними базами даних користувачів. Тим паче, можливість тісної інтеграції нового сервісу з доменом робить використання IIS ще більш вигідним. [9,10]

2.3 Передача даних в форматі JSON

Для передачі даних між центральною базою та мобільним додатком було обрано формат JSON.

JSON (JavaScript Object Notation) – це формат для зберігання та обміну інформацією, доступною для читання людиною. Файл містить тільки текст і використовує розширення .json.

Існує два основних елементів об'єкта JSON: ключі та значення.

Ключ повинен бути рядком. Він містить послідовність символів, які укладені в лапки.

Значення є допустимим типом даних JSON. Вони можуть бути в декількох форматах, таких як масив, об'єкт, число, рядки, логічного значення або значення null.

Об'єкт JSON повинен починатися і закінчуватися фігурними дужками `{}`. Всередині може бути дві або більше пар ключів / значень з комою для їх розділення. А за кожним ключем повинна бути двокрапка, щоб відрізнити його від значення. [11].

2.4 Проектування мобільного додатку

Розроблений мобільний додаток має широкий функціонал, варіанти використання котрого представлено в діаграмі прецедентів на рисунку 2.1, де відображається зв'язок між акторами та прецедентами.

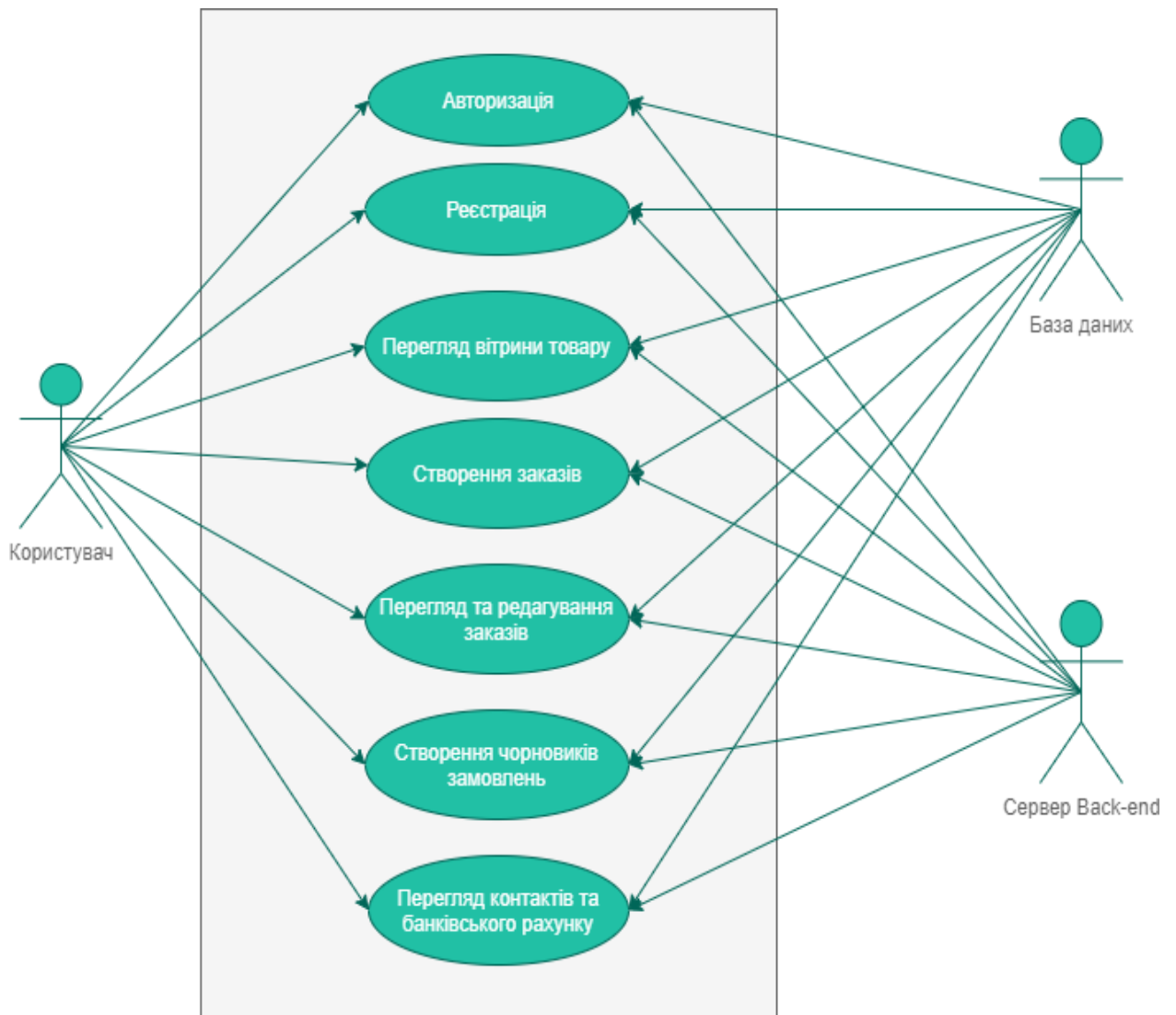


Рисунок 2.1 – Use-case діаграма

Основними варіантами використання є авторизація, реєстрація, перегляд вітрини з товарами, замовлення товарів здійснюючи створення заказів, їх перегляд та редагування, створення шаблонів та перегляд номерів телефонів технічної підтримки та банківських реквізитів для сплати замовлень.

2.5 Проектування мобільного додатку

Під час розробки мобільного додатку використовується інформаційна база 1С УНФ, в ній було визначено сутності та атрибути та зв'язки між ними.

Виділено такі основні сутності для правильної роботи системи:

- Користувач
- Партнери
- Замовлення
- Адреси
- Види оплати
- Інтервали часу для доставки
- Географія
- Номенклатура
- Ціни на товар

У таблиці 2.1 описані основні сутності та атрибути, які використовувалися для реалізації в мобільному додатку.

Кожен з яких є важливою складовою в реалізації мобільного додатку. Сутність користувача є основною для реалізації авторизації та реєстрації, дані логіна та паролю є атрибутами саме неї. Також у користувача існує певна роль, якщо він клієнт то може зробити замовлення тільки на себе, як фізичну особу. Другим варіантом ролі є агент – це користувач який має дещо ширші права, в центральній базі під кожного агента визначений список партнерів, доступ до яких він має, а саме може переглядати, редагувати та створювати замовлення на певного контрагента.

Сутність замовлення описує структуру котра потім буде передаватися в форматі json серверу, де в центральній базі УНФ буде сформований документ «Заказ», з котрим надалі будуть працювати логіст. Тобто це один з головніших документів для роботи з клієнтом, він містить в собі інформацію про адресу доставки, клієнта, спосіб оплати, це може бути готівка або безготівковий розрахунок, дату доставки, інтервал часу в котрий необхідно доставити товар

клієнту, номенклатуру товару (переважно це вода), її кількість, та ціна до сплати. [11,12]

Таблиця 2.1 – Опис сутностей та атрибутів

Сутність	Опис сутності	Атрибут	Опис атрибута
Users	Користувачі мобільного додатку	name	Ім'я користувача
		surname	Прізвище користувача
		role	Роль (клієнт, агент)
		login	Логін для авторизації в мобільному додатку
		password	Пароль для авторизації в мобільному додатку
		email	Електронна пошта користувача
		user_id	Ідентифікатор користувача
Partners	Контрагент, партнер, замовник товару	partner_id	Ідентифікатор контрагента
		user_id	Ідентифікатор користувача
		address_id	Ідентифікатор адреси
Orders	Замовлення	order_id	Ідентифікатор замовлення
		address_id	Ідентифікатор адреси
		nomenclature_id	Ідентифікатор номенклатури
		interval_time_id	Ідентифікатор інтервала часу
		type_payment_id	Ідентифікатор виду оплати

Таблиця 2.1 – Продовження таблиці опису сутностей та атрибутів

		amount_nomencl	Кількість товару
		partner_id	Ідентифікатор контрагента
Address	Адреса доставки товару	address_id	Ідентифікатор адреси
		country_id	Ідентифікатор країни
		city	Місто
		street	Вулиця
		building	Будівля
Countries	Країна доставки товару, географія	country_id	Ідентифікатор країни
		country_name	Найменування країни
Type_Payment	Вид оплати замовлення	type_payment_id	Ідентифікатор виду оплати
		payment_name	Найменування виду
Interval_time	Інтервали часу для доставки товару	interval_time_id	Ідентифікатор інтервала часу
		interval_name	Найменування інтервалу
Nomenclature	Номенклатура товару	nomenclature_id	Ідентифікатор номенклатури
		nomenclature_name	Найменування номенклатури
		price_id	Ідентифікатор ціни
		nomenclature_photo	Фото номенклатури
Price	Ціна за товар	price_id	Ідентифікатор ціни
		price_name	Найменування ціни
		price_course	Курс валюти
		nomenclature_id	Ідентифікатор номенклатури
		price_data	Дата з якої прийнята ціна

Описані сутності та атрибути, є основними, в центральній базі кожна сутність має більш розгорнуту структуру з ширшим списком атрибутів та складнішими зв'язками.

Було розроблено ER-діаграму, в якій відображається зв'язок між сутностями, це наведено на рисунку 2.2.

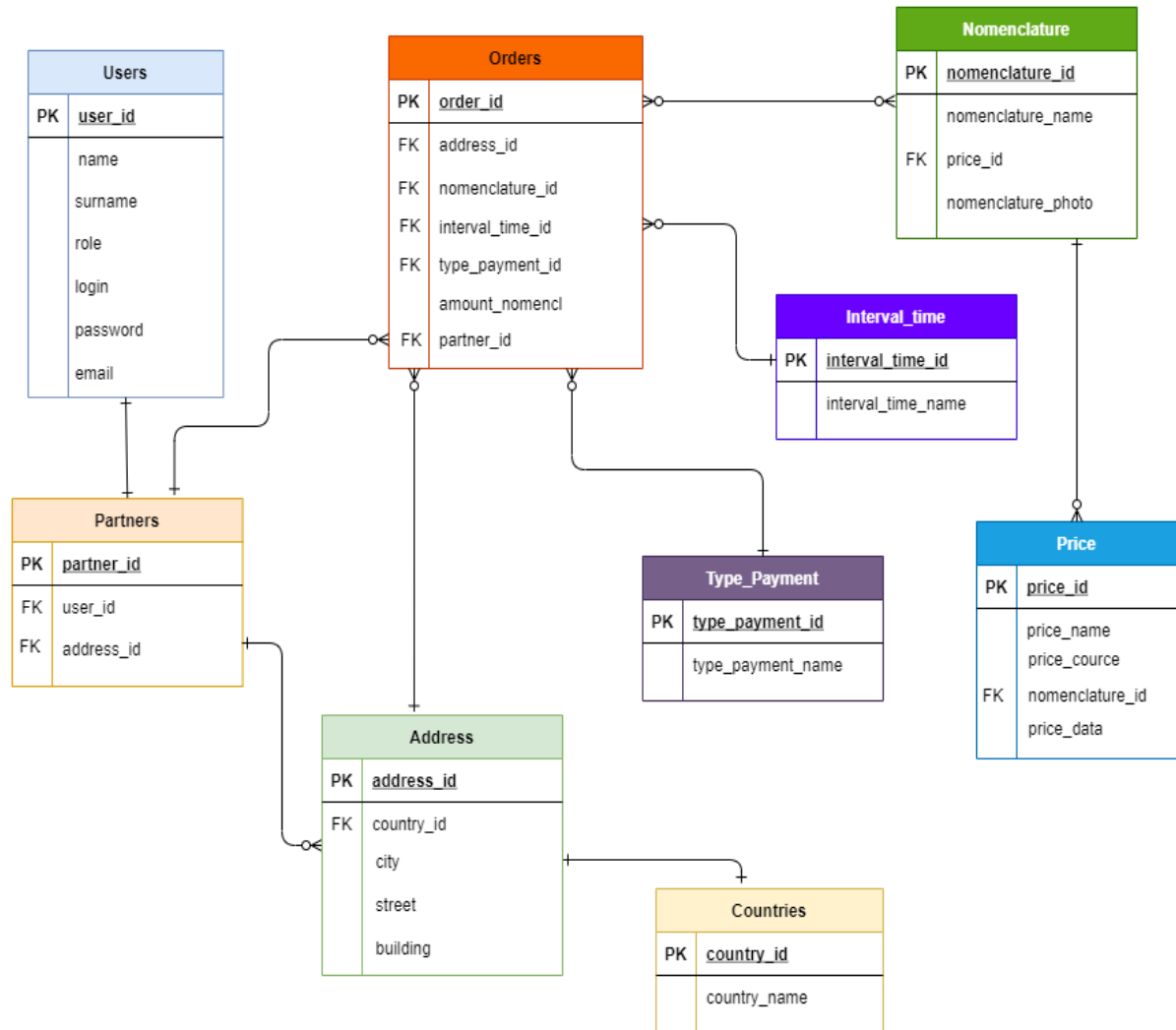


Рисунок 2.2 – ER-діаграма

Між сутностями Users та Partners існує зв'язок один до одного, що характеризується тим що у одного користувача може бути лише один контрагент. Для сутності Partners та Address обрано зв'язок один до багатьох, тобто у одного контрагента може бути одна адреса, а певна адреса може мати безліч контрагентів.

Між сутністю Orders та сутностями Partners, Address, Type_Payment та Interval_Time установлений зв'язок багато до одного, тобто може в базі

створюватися безліч заказів по певному контрагенту, з певним адресом, типом оплати, інтервалом часу в котрий буде здійснена доставка та номенклатурою, але в самому заказі ці дані відповідають одному значенню. Також з сутність Orders та Nomenclature існує зв'язок багато до багатьох, в заказі може існувати певна кількість номенклатури, так само, як і для деякої номенклатури може існувати певна кількість замовлень.

Між сутностями Address та Country обраний зв'язок один до одного, тобто для конкретної адреси є одна географія, а для однієї географії може існувати лише одна адреса. Географія мається на увазі певна точка на карті, котра має координати широти та довготи та вказує на певну адресу на карті та є унікальним значенням, використовуються при автоматичному визначені поточної адреси користувача при оформленні замовлення в мобільному додатку.

Відношення між Nomenclature та Price характеризується як один до багатьох, для однієї номенклатури може існувати декілька цін з різними датами та курсом валют, а одна ціна має лише одну номенклатуру, товар. Різні ціни характеризуються видом замовлення, це може бути роздрібна торгівля та оптова з урахуванням поточної дати та курс валют на цю дату.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Розробка моделі інформаційної системи

В ході роботи було розроблено додаток «Доставка води», який взаємодіє з центральною інформаційною базою УНФ, яка в даному випадку виступає сервером, де зберігаються та обробляються дані. Їх взаємодія налагоджена за допомогою веб-серверу Internet Information Services, який підтримує протоколи HTTP, завдяки яким мобільний додаток обмінюється даними з сервером. Архітектура роботи зображено на рисунку 3.1.



Рисунок 3.1 – Схематична архітектура роботи системи

Центральною базою системи виступає типова конфігурація УНФ 1.6, на основі якої була допрацьована база за допомогою високо-рівневої мови програмування 1С, з використанням набору базових класів, таких як константи, довідник, документ і т.д. На підставі типових класів в конфігураторі було створено підкласи, що мають свої набори. За допомогою модулів менеджерів та модулів об'єктів було розширено функціонал підкласу. [13, 15]

Мобільний додаток реалізовано для платформи Android, яка являється найбільш популярною мобільною операційною системою в світі, на мові програмування 1С.

Взаємодія центральної бази з мобільним додатком реалізовано за допомогою HTTP-запитів, запит та відповіді на запити надсилаються в форматі JSON. Приклад структури в форматі json буде наведено далі. [13]

В системі 1С УНФ 1.6 зберігаються всі дані по користувачеві, товару, замовленнях. Обліковий запис користувача мобільного додатку знаходиться в довіднику Контрагенти, в котрому допрацьована структура з логіном, паролем, генерацією паролю, та ознакою використання в мобільному додатку.

Дані про товар знаходяться в довіднику Номенклатура, який також був змінений структурно згідно технічного завдання.

Мобільний додаток за допомогою HTTP-запитів отримує та відправляє дані до центральної бази. [14, 15]

3.2 Програмна реалізація

3.2.1 Вітрина додатку

Користувач запускає Мобільний додаток.

Мобільний додаток відкриває форму Вітрини та надсилає http запит до бази 1С УНФ 1.6 для отримання інформації про зміст каталогу товарів на Вітрині. Формується запит по інформаційній базі, в якому обираються всі товари та інформація по ним, якщо встановлена ознака загрузки в мобільний додаток, яка міститься в картці кожної номенклатури.

На рисунку 3.2 продемонстровано частина коду при надсиланні запиту для відображення товарів на Вітрині.

```

□ функция ПолучитьКаталогТоваров () Экспорт

    АpiKey = ПолучитьЗначениеКонстанты("ES_ApiKey");
    СтрокаПосле = "";
    Соединение = ПолучитьСоединениеСервера(СтрокаПосле);
    Запрос = Новый HTTPЗапрос;

    АдресРесурса = СтрокаПосле + "/hs/es-mobile/GetProductCatalog?apikey=" + АpiKey;
    HTTPЗапрос = Новый HTTPЗапрос(АдресРесурса);

    Попытка
        Ответ = Соединение.Получить(HTTPЗапрос);
    Исключение
        ТекстОтвета = "Ошибка при подключении";
        СтрОтвета = Новый Структура;
        СтрОтвета.Вставить("Результат", Ложь);
        СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);
        Возврат СтрОтвета;
    КонецПопытки;

    Если Ответ.КодСостояния <> 200 Тогда
        ТекстОтвета = "Ошибка при подключении";
        СтрОтвета = Новый Структура;
        СтрОтвета.Вставить("Результат", Ложь);
        СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);

        Возврат СтрОтвета;
    Иначе
        Ответ = Ответ.ПолучитьТелоКакСтроку();
        ЧтениеJSON = Новый ЧтениеJSON;
        ЧтениеJSON.УстановитьСтроку(Ответ);
        Данные = ПрочитатьJSON(ЧтениеJSON);
        ЧтениеJSON.Закреть();
    Иначе

```

Рисунок 3.2 – Фрагмент коду функції отримання каталогу вітрини

1С УНФ 1.6 відповідає на запит і повертає структуру в форматі json:

```

{"МассивВитрины":[
  {"КартинкаТовара":"Фото1",
  "НаименованиеТовара":"Иволжанская"},{
  "КартинкаТовара":"Фото2",
  "НаименованиеТовара":"Артезианская"}
  ...
]}

```

Список товару визначається в 1С УНФ 1.6, ознакою в картці товару.

Мобільний додаток обробляє відповідь центральної бази і заповнює динамічну форму даними. При виводі на формі Вітрини кожен елемент блока з товаром додається програмно копіюванням основного, з подальшим доданням його нижче по списку. Програмне заповнення даної форми наведено на рисунку 3.3, де показаний фрагмент коду, повний код процедури можна переглянути в Додатку А.

```

        НаСервере
        Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

            Данные = ES_ОбщийМодульСервер.ПолучитьКаталогТоваров();

            Если Данные.Результат Тогда
                МассивВитрины = Данные.Данные.МассивВитрины;

                ДобавляемыеРеквизиты = Новый Массив;
                Для Счетчик = 0 По МассивВитрины.Количество() Цикл

                    мРеквизит = Новый РеквизитФормы("Картинка_" + Счетчик, Новый ОписаниеТипов("Строка"));
                    ДобавляемыеРеквизиты.Добавить(мРеквизит);

                КонечЦикла;

                ИзменитьРеквизиты(ДобавляемыеРеквизиты);

                Для Каждого Позиция Из МассивВитрины Цикл
                    СоздатьКопиюЭлемента(Элементы.ОписаниеВоды_, МассивВитрины.Найти(Позиция), Элементы.ВитринаВоды);
                    Элементы["ОписаниеВоды_" + МассивВитрины.Найти(Позиция)].Видимость = Истина;
                    Элементы["Название_" + МассивВитрины.Найти(Позиция)].Заголовок = Позиция.НаименованиеТовара;
                    Элементы["Цена_" + МассивВитрины.Найти(Позиция)].Заголовок = Позиция.ЦенаТовара;

                    мДвоичныеДанные = Вазе64Значение(Позиция.КартинкаТовара);
                    Имяфайла = КаталогВременныхФайлов() + ПолучитьИмяВременногоФайла(".jpg");
                    мДвоичныеДанные.Записать(Имяфайла);

                    ЭтаФорма["Картинка_" + МассивВитрины.Найти(Позиция)] = ПоместитьВоВременноеХранилище(Новый Картинка(Имяфайла));

                    УдалитьФайлы(Имяфайла);
                КонечЦикла;
    
```

Рисунок 3.3 – Фрагмент коду процедури заповнення форми вітрини

Фотографія товару приходить в форматі двійкових даних, при відображенні на вітрині відбувається перетворення в картинку, найменування товару та ціна також заповнюються після перетворення даних. На рисунку 3.4 можемо переглянути вигляд інтерфейсу форми Вітрини, вигляд продемонстрований на різних пристроях.

Для тестування використовувалися телефони з різним розширенням екрану та різними версіями Android, також інтерфейс був перевірений в емуляторі. [16, 17]

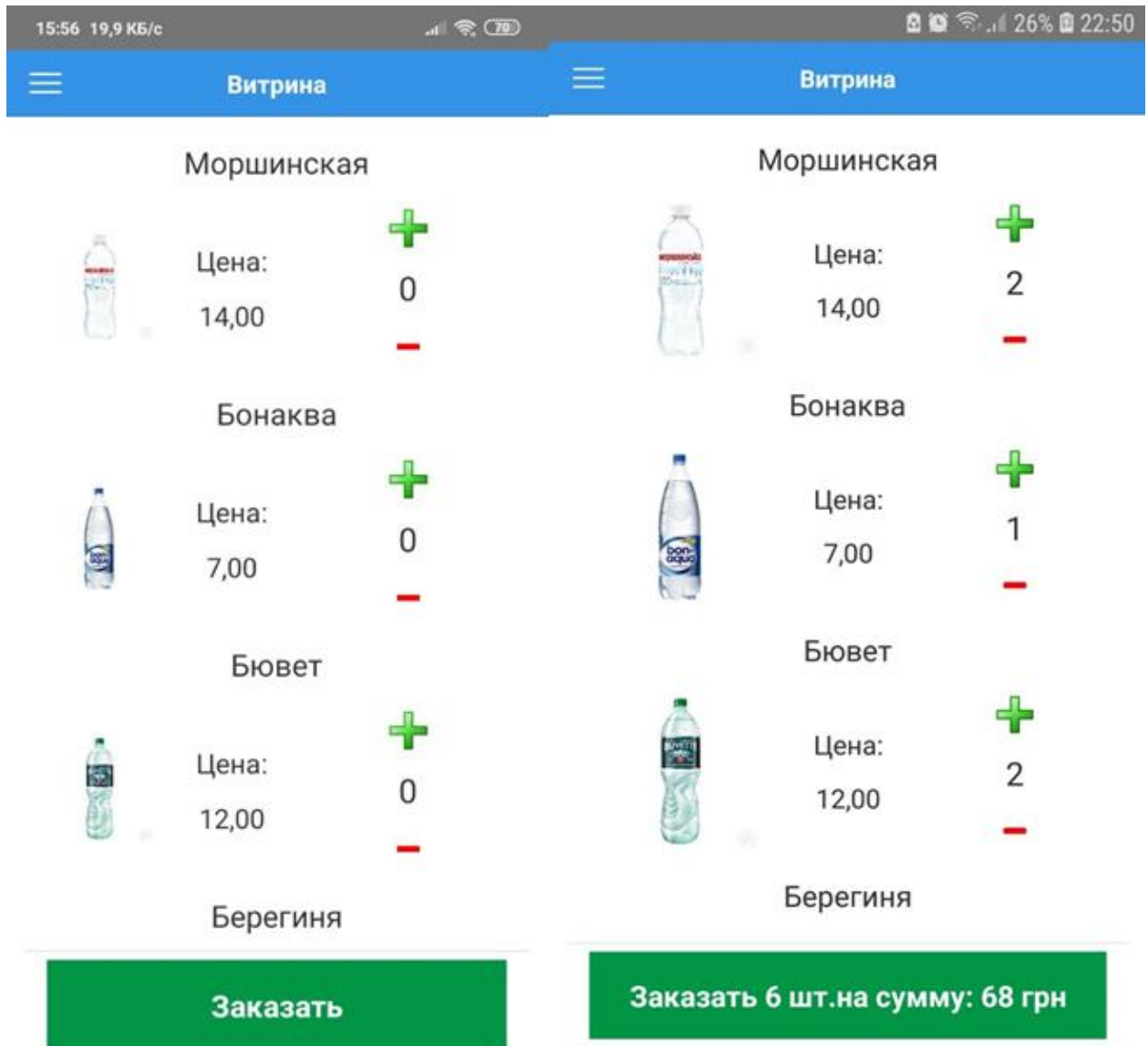


Рисунок 3.4 – Вітрина мобільного додатку

3.2.2 Реєстрація та авторизація користувача в мобільному додатку

Під час реєстрації та авторизації користувача в мобільному додатку виконується певний алгоритм роботи програми. Під час авторизації вводяться дані логіну та паролю та обробляються, а за рахунок відповіді при їх обробці виконуються різні дії програми, при успішній авторизації відкривається наступне вікно для роботи з замовленнями, при не правильних або некоректно введених даних відбувається відмова від авторизації і повернення на попередній крок з введенням даних.

При реєстрації користувача відкривається вікно для заповнення даних, а саме мобільного телефону, після чого дані обробляються, на даний номер йде виклик, менеджер виконує реєстрацію користувача в базі. Результатом успішного виконання є отриманий логіну та паролю і повернення на форму авторизацію, при відмові – очищається значення номеру для повторного вводу. Алгоритм наведений на Рисунку 3.5.

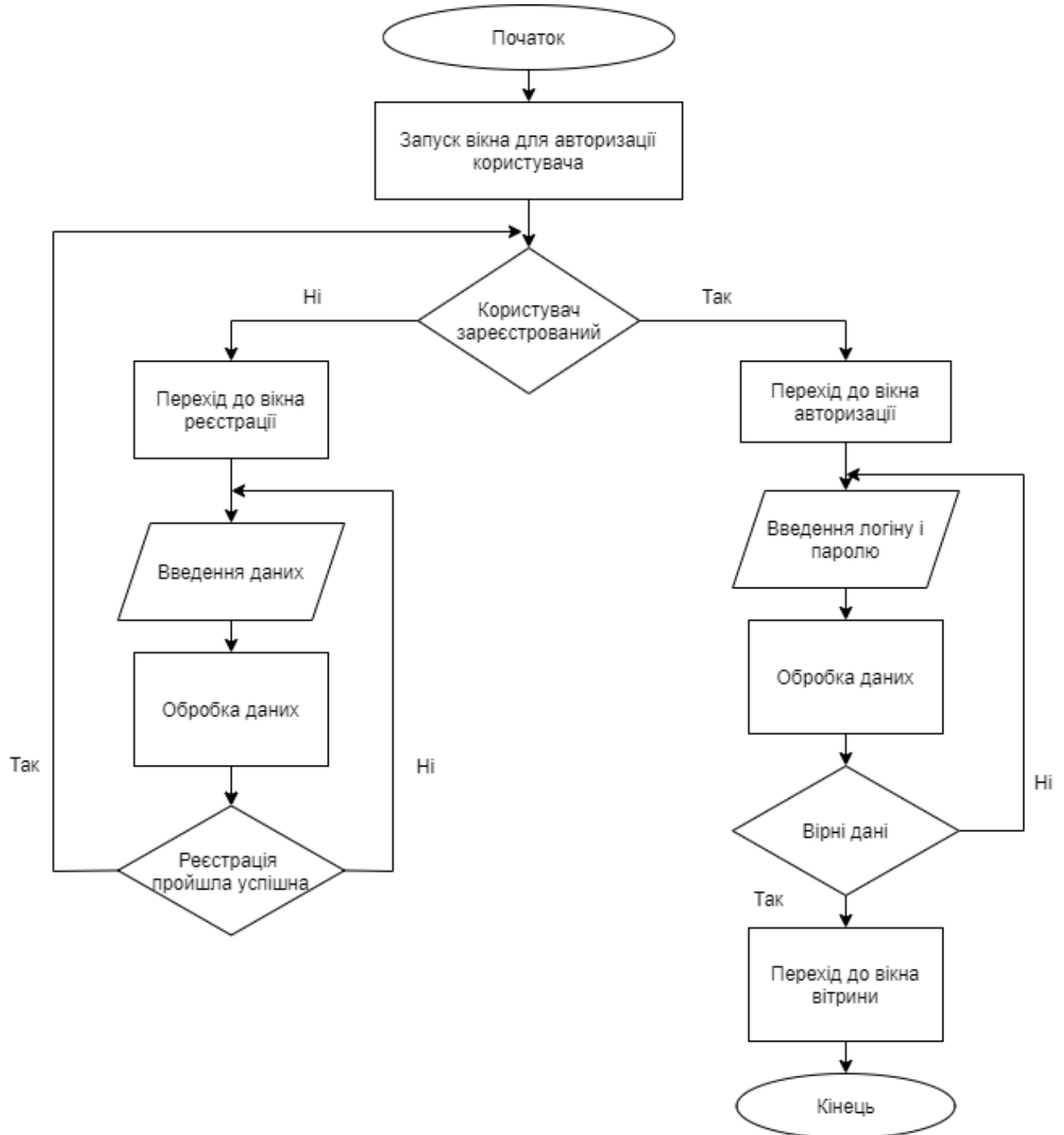


Рисунок 3.5 – Блок-схема алгоритму реєстрації та авторизації

Мобільний додаток відкриває форму Авторизації з полями для введення Логіна і Пароля і командами для Авторизації і Реєстрації. При необхідності мобільний додаток може запам'ятати пароль користувача, якщо він встановив відповідну ознаку на формі. На рисунку 3.6 зображена форму авторизації та реєстрації.

The image displays two side-by-side screenshots of the Aqua Trade mobile application interface. Both screens feature a blue header with a hamburger menu icon and the text 'Авторизация'. Below the header is a navigation bar with 'Авторизация' and 'Регистрация' options. The left screen shows the login form with fields for 'Логин:' (containing '000000003') and 'Пароль:' (containing '...'). A checked checkbox labeled 'Запомнить' is visible, and a green 'Войти' button is at the bottom. The right screen shows the registration form with a 'Номер телефона:' field (containing '0950124110') and a green 'Подтвердить' button.

Рисунок 3.6 – Форма авторизації та реєстрації в мобільному додатку

Користувач вводить облікові дані (логін і пароль), отримані при реєстрації і виконує команду «Увійти».

На рисунку 3.7 показаний фрагмент коду при авторизації:

```

&НаСервере
□ Процедура ВойтиНаСервере (Авторизован)

    Ответ = ES_ОбщийМодульСервер.ВыполнитьАвторизацию(ЛогинАвторизация, ПарольАвторизация);

    Если Ответ.Результат Тогда
        Если Ответ.ТипПользователя = 0 Тогда
            ES_ОбщийМодульСервер.УстановитьЗначениеКонстанты("ES_ТипПользователя",
            ES_ОбщийМодульСервер.ПолучитьЗначениеПеречисления("ES_ТипПользователя", "Агент"));
        Иначе
            ES_ОбщийМодульСервер.УстановитьЗначениеКонстанты("ES_ТипПользователя",
            ES_ОбщийМодульСервер.ПолучитьЗначениеПеречисления("ES_ТипПользователя", "Клиент"));
        КонецЕсли;

        Если Запомнить Тогда
            Константы.ES_Логин.Установить(ЛогинАвторизация);
            Константы.ES_Пароль.Установить(ПарольАвторизация);
        Иначе
            Константы.ES_Логин.Установить(ЛогинАвторизация);
            Константы.ES_Пароль.Установить("");
        КонецЕсли;

        ОбработатьСтруктуруОтветаЗаполнитьСправочники(Ответ.Контрагент);

        Авторизован = Истина;
    Иначе
        Сообщить(Ответ.ТекстОшибки);
        Авторизован = Ложь;
    КонецЕсли;
КонецПроцедуры

```

Рисунок 3.7 – Фрагмент коду авторизації

Мобільний додаток виконує запит в 1С УНФ 1.6 і передає введені облікові дані.

1С УНФ 1.6 звіряє введені облікові дані з облікового запису користувача і формує відповідь про успішну авторизацію або помилку.

Мобільний додаток отримує відповідь від 1С УНФ 1.6, виконує вхід в мобільний додаток і показує список замовлень або показує користувачеві інформаційне вікно про помилку.

Агент в результаті успішної авторизації бачить форму списку з Замовленнями своїх клієнтів.

Клієнт в результаті успішної авторизації бачить форму списку з власними Замовленнями. [18]

3.2.3 Завантаження замовлень

Користувач після успішної Авторизації в мобільному додатку бачить форму списку замовлень. Для заповнення списку замовленнями виконує команду «Завантажити замовлення» в командній панелі.

Мобільний додаток здійснює запит в 1С УНФ 1.6 на отримання списку замовлень по даному користувачеві мобільного додаток.

1С УНФ 1.6 здійснює вибірку Замовлення з датою доставки на поточну дату.

Структура замовлення в мобільному застосунку представлена на рисунку 3.8

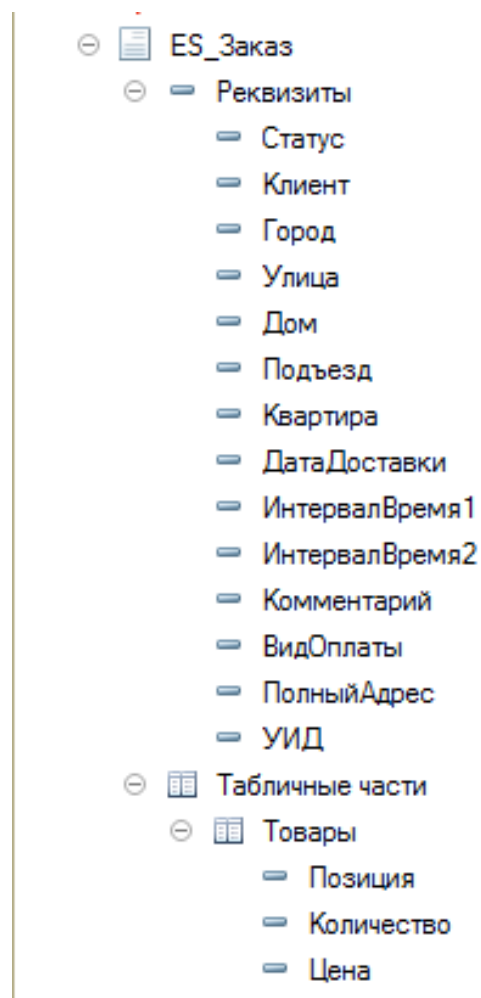


Рисунок 3.8 - Структура замовлення

Для Агента вибираються Замовлення клієнтів, які за ним закріплені.

Для Клієнта вибираються Замовлення тільки по даному клієнту.

1С УНФ 1.6 формує відповідь, повертаючи список замовлень або повідомлення про помилку.

Мобільний додаток отримує відповідь від 1С УНФ 1.6, заповнює список замовлень або показує користувачеві інформаційне вікно про помилку. Процедура формування списку заказів продемонстрована на рисунку 3.9. Реєстрація та авторизація користувача в мобільному.

```

Процедура СоздатьЗаказыПоСписку (СтруктураОтвета) Экспорт
    Если СтруктураОтвета.Получить ("Результат") Тогда
        МассивЗаказов = СтруктураОтвета.Получить ("Заавки");
        Для Каждого Позиция из МассивЗаказов Цикл
            Если Документы.ES_Заказ.НайтиПоРеквизиту ("УИД", Позиция.Получить ("УИД")).Пустая () Тогда
                ДокументЗаказа = Документы.ES_Заказ.СоздатьДокумент ();
                ДокументЗаказа.УИД = Позиция.Получить ("УИД");
            Иначе
                ДокументЗаказа = Документы.ES_Заказ.НайтиПоРеквизиту ("УИД", Позиция.Получить ("УИД")).ПолучитьОбъект ();
                ДокументЗаказа.ЭтотОбъект.Товары.Очистить ();
            КонецЕсли;
            ДокументЗаказа.Дата = ПривестиКТипуДата (СтрЗаменить (Позиция.Получить ("Дата"), "-", "."));
            ДокументЗаказа.Статус = ПолучитьЗначениеПеречисления ("ES_СтатусыЗаказа", Позиция.Получить ("Статус"));
            ДокументЗаказа.Клиент = Справочники.Контрагенты.НайтиПоНаименованию (Позиция.Получить ("Клиент"));
            ДокументЗаказа.ДатаДоставки = ПривестиКТипуДата (СтрЗаменить (Позиция.Получить ("ДатаДоставки"), "-", "."));
            ДокументЗаказа.ИнтервалВремя1 = ПолучитьЗначениеПеречисления ("ES_Интервалы", СтрокаПеречисления (Позиция.Получить ("ИнтервалВремя1")));
            ДокументЗаказа.ИнтервалВремя2 = ПолучитьЗначениеПеречисления ("ES_Интервалы", СтрокаПеречисления (Позиция.Получить ("ИнтервалВремя2")));
            ДокументЗаказа.Комментарий = Позиция.Получить ("Комментарий");
            Если Позиция.Получить ("ВидОплаты") = 1 Тогда
                ДокументЗаказа.ВидОплаты = "Наличные";
            ИначеЕсли Позиция.Получить ("ВидОплаты") = 2 Тогда
                ДокументЗаказа.ВидОплаты = "КартаПриват";
            ИначеЕсли Позиция.Получить ("ВидОплаты") = 3 Тогда
                ДокументЗаказа.ВидОплаты = "Безналичные";
            КонецЕсли;
            ДокументЗаказа.ПолныйАдрес = Позиция.Получить ("ПолныйАдрес");
            Товары = Позиция.Получить ("Товары");
            Для каждого Эл Из Товары Цикл
                НоваяСтрока = ДокументЗаказа.ЭтотОбъект.Товары.Добавить ();
                НоваяСтрока.Позиция = Справочники.Номенклатура.НайтиПоНаименованию (Эл.Получить ("Позиция"));
                НоваяСтрока.Количество = Эл.Получить ("Количество");
                НоваяСтрока.Цена = Эл.Получить ("Цена");
            КонецЦикла;
    КонецПроцедуры

```

Рисунок 3.9 – Процедура коду заповнення списку замовлень

Відбір Замовлення на формі списку здійснюється за періодом. Для Агента відкритий ширший функціонал, в котрому також можливий відбір по клієнтам.

На формі після вибору фільтру відображаються замовлення, котрі задовольняють дані відбору, які уже були створенні в базі УНФ по авторизованому користувачеві.

Також в списку замовлень будуть показані шаблони, це замовлення які були збережені користувачем для подальшого використання. Даний механізм існує для зручності повторення подібних запитів.

Форма списку замовлень клієнта в Мобільному додатку зображено на рисунку 3.10 на різних пристроях від різних виробників, а саме Samsung та Huawei.

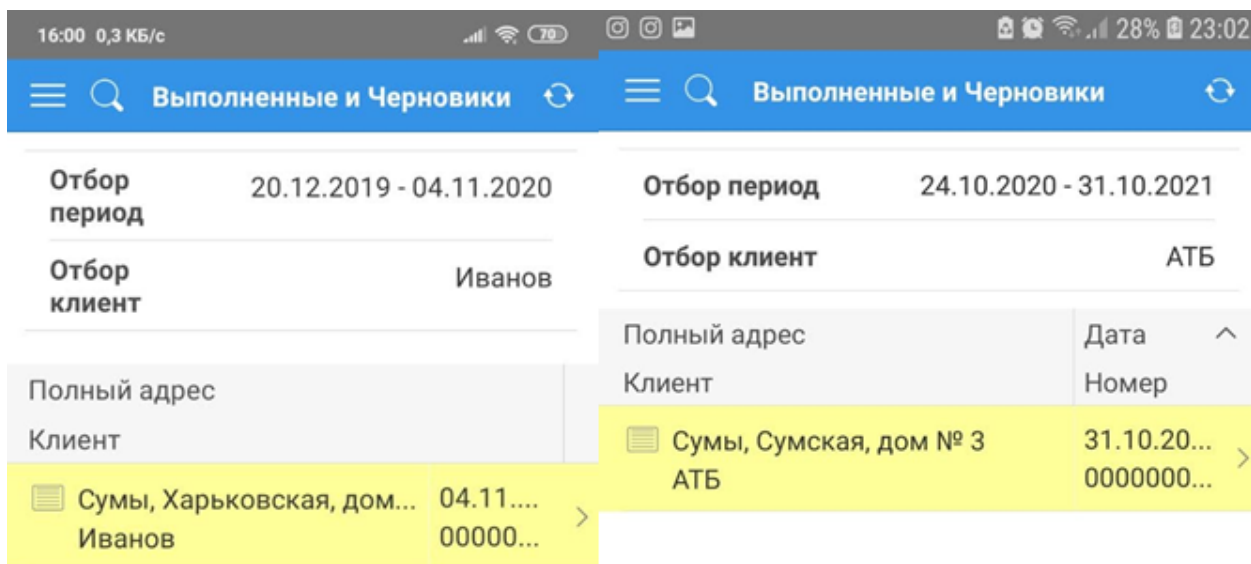


Рисунок 3.10 – Форма списку заказов

Користувач додатку обирає товар, який йому необхідний і натискає команду «Создать заказ».

Мобільний додаток перевіряє авторизацію користувача, якщо вона успішна – відкривається форма нового замовлення з заповненою інформацією по товарам, в іншому випадку просить Користувача мобільного додатку зареєструватися в базі.

3.2.4 Створення замовлення

Користувач проходить Авторизацію в мобільному додатку і виконує команду «Создать заказ» на формі списку замовлень.

Мобільний додаток відкриває форму Заовлення. Зовнішній вигляд форми зображено на рисунку 3.11.

Создать новый (создание) * Создать новый (создание)

31.10.2020 07.06.2021

Клиент Нажмите для выбора Клиент Нажмите для выбора

г. Сумы ул. Комсомольская д. 10 п. 1 этаж 3 кв. 134 Адрес Нажмите для выб...

Дата доставки Дата доставки

Интервалы 12:00 - 15:00 17:00 - 20:00 Интервалы

Наличные Безналичные Наличные Безналичные

Добавить Еще > Добавить Еще >

Позиция	Количес...	Цена
Моршинская	2	14,00 >
Бонаква	1	7,00 >
Бювет	2	12,00 >
Миргородская	1	9,00 >
Общая сумма:		68,00
Комментарий		

Общая сумма: Комментарий

Отправить

Черновик Отменить

Рисунок 3.11 – Форма Заовлення


Користувач заповнює поля: Адреса, Дата доставки, Інтервали доставки, Коментар, вибирає Вид оплати, додає позиції заовлення і їх кількість в табличну частину заовлення. Введення адреси здійснюється двома способами:

1. Виконати команду «Визначити місце розташування» – заповнює адресу поточним місцезнаходженням користувача мобільного додаток.

2. Виконати команду «Ввести» – викликає діалогове вікно для введення адреси (вулиця, будинок, під'їзд, квартира). Форма вводу адреси продемонстрована на рисунку 3.12.

3. Виконати команду «Вибрати зі списку» – викликає вікно вибору зі списку. У ньому визначено список адрес доступних для доставки. Цей список завантажується з картки клієнта основної бази 1С УНФ 1.6.

Закреть Ввод адреса



Город

Улица

Дом

Подъезд Этаж Квартира

0 0

Ок Отмена

Рисунок 3.12 – Форма введення адреси

3.2.5 Контакти мобільного додатку

Користувач проходить Авторизацію в мобільному додатку і в командному інтерфейсі обирає вкладку «Контакти».

Мобільний додаток відкриває форму з параметрами мобільного додаток. Форма параметрів в Мобільному додатку зображена на рисунку 3.13.

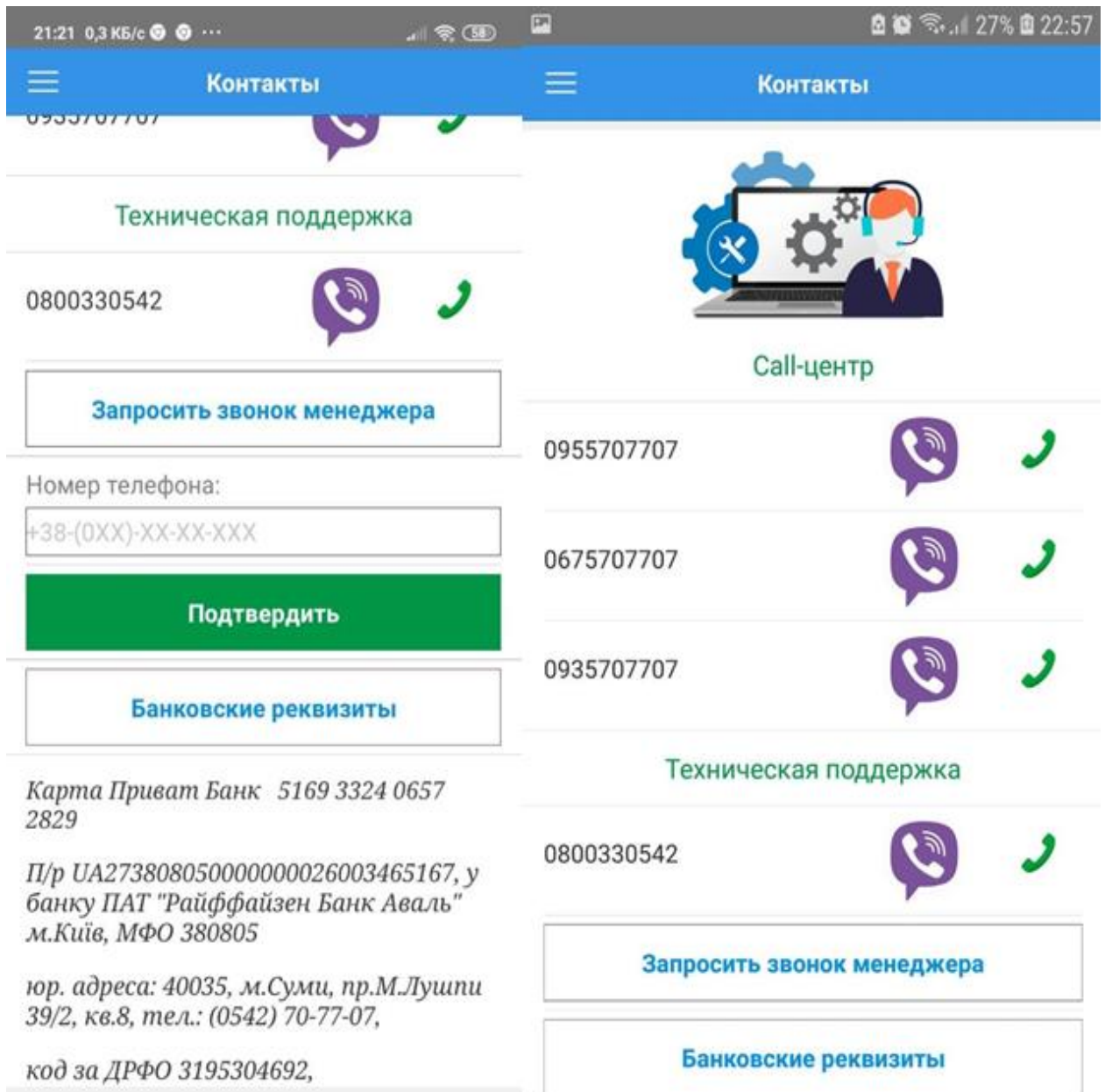


Рисунок 3.13 – Форма Контактів

ВИСНОВКИ

У ході виконання роботи було здійснено аналіз сучасного стану в розробці мобільних додатків на операційній платформі Android, розглянуто основні інструменти розробки та середовища, виконаний огляд та аналіз продуктів-аналогів.

В процесі вибору механізмів реалізації та середовища було розглянуто та проаналізовано найпопулярніші з них та обрано платформу 1С для мобільних додатків, дане середовище найбільш задовольняє вимоги поставленої задачі та є найбільш відповідним до центральної інформаційної бази 1С.

Результатом роботи є створений мобільний додаток на операційній системі Android «Доставка води» за допомогою мови програмування 1С. Він забезпечує роботу системи в режимі користувача: реалізовано механізми авторизації та реєстрації, перегляд доступних товарів, створення та моніторинг замовлень. Додаток має зручний та сучасний інтерфейс, який задовільняє вимоги поставленої задачі, допрацьовано модулі центральної бази даних, їх структура для взаємодії з мобільним, обмін між ними налагоджено по протоколу HTTP за допомогою веб-серверу.

Даний продукт організовує зв'язок з користувачем та організацією, яка надає послуги по продажу та доставці води. Зареєструвавшись в мобільному додатку можна здійснювати замовлення, переглядати уже створені замовлення та створювати шаблони замовлень. Дана робота є актуальною, особливо в карантинний період, коли потрібно мінімізувати контакт з людьми, то правильним рішенням є саме система, яка дозволить безконтактно замовити товар і отримати його в зазначений час в потрібному користувачеві місці доставки. При розробці були використані набуті знання та навички та вдосконалено їх.

СПИСОК ЛІТЕРАТУРИ

1. Мобільний додаток [Електронний ресурс]: – Що таке мобільний додаток або застосунок – Режим доступа: https://uk.wikipedia.org/wiki/Мобільний_застосунок
2. Прасти, Н. Блокчейн. Розробка додатків. / Н. Прасти. - СПб.: ВHV, 2019. - 176 с.
3. Эспозито, Д. Разработка современных веб-приложений: анализ предметных областей и технологий / Д. Эспозито. - М.: Вильямс И.Д., 2017. - 464 с.
4. Хорев, П.Б. Объектно-ориентированное программирование: Учебное пособие / П.Б. Хорев. - М.: Академия, 2018. - 384 с
5. Бобошко, Д.Д. 1С: Предприятие 8.0. Программирование в примерах / Д.Д. Бобошко. - М.: Кудиц-Пресс, 2007. - 384 с.
6. Розробка мобільного [Електронний ресурс]: – <https://knowledge.allbest.ru/programming/3c0b65625b3bd69a4d53a89521316c370.html>
7. Дэрси, Л. Разработка приложений для Android-устройств. Т. 1: Базовые принципы / Л. Дэрси, Ш. Кондер. - М.: Лори, 2018. - 402 с.
8. Методи запроса [Електронний ресурс]: – <https://ru.w3docs.com/uchebnik-html/metody-http-zaprosa.html>
9. Foundations for the study of software architecture // Software Engineering Notes. — ACM SIGSOFT[en], 2017. — Vol. 17, iss. 4 (6 June). — P. 40-52. Цитовано 2017-01-08.
10. Інформація про Android Studio. [Електронний ресурс]. – Режим доступа: https://uk.wikipedia.org/wiki/Android_Studio
11. Методологія розробки додатків. [Електронний ресурс]. – Режим доступа: <http://android.mobile-review.com/articles/22580/>
12. Архітектура ОС Android-додатків. [Електронний ресурс]. – Режим доступа: <http://habrahabr.ru/post/140655/>

13. Android Studio Development [Электронный ресурс] // Android Platform & Tools Survey. – 2017. – Режим доступа до ресурсу: <https://developer.android.com/>
14. Sierra K. Head First / К. Sierra, В. Bates., 2019. – O'Reilly, – 254 с.
15. Майер Р. Android 2: Программирование для планшетных компьютеров и смартфонов / Ретро Майер ; пер. с англ.– М. : Эксмо, 2018. – 672 с
16. Популярні середовища розробки [Электронный ресурс] – Режим доступа: <http://bit.ly/2InsdIW>
17. A good index of development of network technologies: 2017-2020, available at: <http://www.cisco.com/web/UA/about/news/2016/02/0222j.html> (rus).
18. Arovina, M, "Prospects of Application of Mobile Apps in the Regional Information Market of Ukraine, 2019 – 217 с.

ДОДАТОК А

Функція отримання зв'язку с сервером:

```

Функція ПолучитьСоединениеСервера(СтрокаПосле) Экспорт
    АдресСервера = ПолучитьЗначениеКонстанты("ES_АдресСервера");
    Пользователь = ПолучитьЗначениеКонстанты("ES_ПользовательИБ");
    Пароль = ПолучитьЗначениеКонстанты("ES_ПарольИБ");
    Если СтрНайти(АдресСервера,"https") = 1 Тогда
        ЗащищенноеСоединение = Истина;
        АдресСер = СтрЗаменить(АдресСервера,"https://","");
    Иначе
        ЗащищенноеСоединение = Ложь;
        АдресСер = СтрЗаменить(АдресСервера,"http://","");
    КонецЕсли;
    АдресСер = Сокрлп(АдресСер);
    ПозицияСлеша = Найти(АдресСер,"/");
    СтрокаДо = Лев(АдресСер,ПозицияСлеша-1);
    СтрокаПосле = Сред(АдресСер,ПозицияСлеша+1,СтрДлина(АдресСер));
    АдресСервера = СтрокаДо
    ПозицияПорта = Найти(АдресСервера,":");
    Если НЕ ЗащищенноеСоединение Тогда
        Если ПозицияПорта > 0 Тогда
            Порт = Прав(АдресСервера,СтрДлина(АдресСервера)-ПозицияПорта);
            АдресСервера = СтрЗаменить(АдресСервера,":"+Порт,"");
            Если Порт = "" или Порт = "00" Тогда
                Порт = 80;
            КонецЕсли;
        Иначе
            Порт = 80;
        КонецЕсли;
    Иначе
        Если ПозицияПорта > 0 Тогда
            Порт = Прав(АдресСервера,СтрДлина(АдресСервера)-ПозицияПорта);
            АдресСервера = СтрЗаменить(АдресСервера,":"+Порт,"");
            Если Порт = "" или Порт = "00" Тогда
                Порт = 443;
            КонецЕсли;
        Иначе
            Порт = 443;
        КонецЕсли;
    КонецЕсли;
    Если ЗащищенноеСоединение Тогда
        Соединение = Новый HTTPСоединение(АдресСервера, Число(Порт),
        Пользователь, Пароль, , , Новый ЗащищенноеСоединениеOpenSSL());
    Иначе
        Соединение = Новый HTTPСоединение(АдресСервера, Число(Порт),
        Пользователь, Пароль);
    КонецЕсли;
    Возврат Соединение;
КонецФункции

```

Функція Авторизації:

```

Функция ВыполнитьАвторизацию(Логин, Пароль) Экспорт
  АpiKey = ПолучитьЗначениеКонстанты("ES_ApiKey");
  СтрокаПосле = "";
  Соединение = ПолучитьСоединениеСервера(СтрокаПосле);
  Запрос = Новый HTTPЗапрос;

  АдресРесурса = СтрокаПосле+"/hs/es-mobile/Authorization?apikey="+ApiKey;

  HTTPЗапрос = Новый HTTPЗапрос(АдресРесурса);

  СтрАвторизация= Новый Структура;
  СтрАвторизация.Вставить("Логин", Логин);
  СтрАвторизация.Вставить("Пароль", Пароль);
  ОбработкаJSON = Обработки.JSON_and_UnJSON.Создать();
  СтрСтрока = ОбработкаJSON.JSON(СтрАвторизация);
  HTTPЗапрос.УстановитьТелоИзСтроки(СтрСтрока);
  Попытка
    Ответ = Соединение.ОтправитьДляОбработки(HTTPЗапрос);
  Исключение
    ТекстОтвета = "Ошибка при подключении";
    СтрОтвета = Новый Структура;
    СтрОтвета.Вставить("Результат", Ложь);
    СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);
    Возврат СтрОтвета;
  КонецПопытки;
  Если Ответ.КодСостояния <> 200 Тогда
    ТекстОтвета = "Ошибка при подключении";
    СтрОтвета = Новый Структура;
    СтрОтвета.Вставить("Результат", Ложь);
    СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);
    Возврат СтрОтвета;
  Иначе
    Ответ = Ответ.ПолучитьТелоКакСтроку();
    ЧтениеJSON = Новый ЧтениеJSON;
    ЧтениеJSON.УстановитьСтроку(Ответ);
    Данные = ПрочитатьJSON(ЧтениеJSON);
    ЧтениеJSON.Закреть();
    Возврат Данные;
  КонецЕсли;
КонецФункции

```

Процедура створення замовлень:

```

Процедура СоздатьЗаказыПоСписку (СтруктураОтвета) Экспорт
  Если СтруктураОтвета.Получить("Результат") Тогда
    МассивЗаказов = СтруктураОтвета.Получить("Заявки");
    Для Каждого Позиция из МассивЗаказов Цикл
      Если Документы.ES_Заказ.НайтиПоРеквизиту("УИД",
        Позиция.Получить("УИД")).Пустая() Тогда
        ДокументЗаказа = Документы.ES_Заказ.СоздатьДокумент();
        ДокументЗаказа.УИД = Позиция.Получить("УИД");
        Иначе
          ДокументЗаказа = Документы.ES_Заказ.НайтиПоРеквизиту("УИД",
            Позиция.Получить("УИД")).ПолучитьОбъект();
          ДокументЗаказа.ЭтотОбъект.Товары.Очистить();

```

```

        КонецЕсли;
        ДокументЗаказа.Дата = ПривестиКТипуДата(СтрЗаменить(
Позиция.Получить("Дата"), "-", "."));
        ДокументЗаказа.Статус = ПолучитьЗначениеПеречисления(
"ES_СтатусыЗаказа", Позиция.Получить("Статус"));
        ДокументЗаказа.Клиент = Справочники.Контрагенты.НайтиПоНаименованию(
Позиция.Получить("Клиент"));
        ДокументЗаказа.ДатаДоставки = ПривестиКТипуДата(СтрЗаменить(
Позиция.
Получить("ДатаДоставки"), "-", "."));
        ДокументЗаказа.ИнтервалВремя1 = ПолучитьЗначениеПеречисления(
"ES_Интервалы", СтрокаПеречисление(Позиция.Получить("ИнтервалВремя1"))); //
        ДокументЗаказа.ИнтервалВремя2 = ПолучитьЗначениеПеречисления(
"ES_Интервалы", СтрокаПеречисление(Позиция.Получить("ИнтервалВремя2"))); //
        ДокументЗаказа.Комментарий = Позиция.Получить("Комментарий");
        Если Позиция.Получить("ВидОплаты") = 1 Тогда
        ДокументЗаказа.ВидОплаты = "Наличные";
        ИначеЕсли Позиция.Получить("ВидОплаты") = 2 Тогда
        ДокументЗаказа.ВидОплаты = "КартаПриват";
        ИначеЕсли Позиция.Получить("ВидОплаты") = 3 Тогда
        ДокументЗаказа.ВидОплаты = "Безналичные";
        КонецЕсли;
        ДокументЗаказа.ПолныйАдрес = Позиция.Получить("ПолныйАдрес");
        Товары = Позиция.Получить("Товары");
        Для каждого Эл Из Товары Цикл
        НоваяСтрока = ДокументЗаказа.ЭтотОбъект.Товары.Добавить();
        НоваяСтрока.Позиция = Справочники.Номенклатура.НайтиПоНаименованию(
Эл.Получить("Позиция"));
        НоваяСтрока.Количество = Эл.Получить("Количество");
        НоваяСтрока.Цена = Эл.Получить("Цена");
        КонецЦикла;
        ДокументЗаказа.Записать(РежимЗаписиДокумента.Запись);
        КонецЦикла
    Иначе
        ТекстОтвета = "Ошибка при подключении";
        СтрОтвета = Новый Структура;
        СтрОтвета.Вставить("Результат", Ложь);
        СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);
    КонецЕсли;
КонецПроцедуры

```

Функція відправки замовлення на сервер:

```

Функция ОтправитЗаказ(Ссылка, Отправка) Экспорт
    СтруктураОтвета = Новый Структура();
    МассивЗаказов2 = Новый Массив;
    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | ES_Заказ.Ссылка КАК Ссылка,
        | ES_Заказ.Номер КАК Номер,
        | ES_Заказ.Дата КАК Дата,
        | ES_Заказ.Статус КАК Статус,
        | ES_Заказ.Клиент КАК Клиент,
        | ES_Заказ.ДатаДоставки КАК ДатаДоставки,
        | ES_Заказ.ИнтервалВремя1 КАК ИнтервалВремя1,
        | ES_Заказ.ИнтервалВремя2 КАК ИнтервалВремя2,
        | ES_Заказ.Комментарий КАК Комментарий,

```

```

|ES_Заказ.ВидОплаты КАК ВидОплаты,
|ES_Заказ.ПолныйАдрес КАК ПолныйАдрес,
|ES_Заказ.УИД КАК УИД,
|ES_Заказ.Товары.(
|Позиция КАК Позиция,
|Количество КАК Количество,
|Цена КАК Цена
|) КАК Товары
|ИЗ
|Документ.ES_Заказ КАК ES_Заказ
|ГДЕ
|ES_Заказ.Ссылка = &Ссылка";
Запрос.УстановитьПараметр("Ссылка", Ссылка);
РезультатЗапроса = Запрос.Выполнить();
Если РезультатЗапроса.Пустой() Тогда
СтруктураОтвета.Вставить("Результат", Ложь);
СтруктураОтвета.Вставить("ТекстОшибки", "Нет назначенных заявок.");
Возврат СтруктураОтвета;
КонецЕсли;
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.Следующий() Цикл
СтруктураЗаявки = ПолучитьСтруктуруЗаявки();
ЗаполнитьЗначенияСвойств(СтруктураЗаявки, Выборка);
ВидОплаты = 0;
Если Выборка.ВидОплаты = "Наличные" Тогда
ВидОплаты = 1;
ИначеЕсли Выборка.ВидОплаты = "Безналичные" Тогда
ВидОплаты = 2;
ИначеЕсли Выборка.ВидОплаты = "КартаПриват" Тогда
ВидОплаты = 3;
КонецЕсли;
СтруктураЗаявки.ВидОплаты = ВидОплаты;
МассивТоваров = Новый Массив;
Товары = Выборка.Товары.Выбрать();
Пока Товары.Следующий() Цикл
СтруктураТоваров = Новый Структура;
СтруктураТоваров.Вставить("Позиция", Товары.Позиция);
СтруктураТоваров.Вставить("Количество", Товары.Количество);
СтруктураТоваров.Вставить("Цена", Товары.Цена);
МассивТоваров.Добавить(СтруктураТоваров);
КонецЦикла;
СтруктураЗаявки.Товары = МассивТоваров;
МассивЗаказов2.Добавить(СтруктураЗаявки);
КонецЦикла;
Если Отправка = 1 Тогда
СтруктураОтвета.Вставить("Отправка", Отправка);
Иначе
Отправка = 0;
СтруктураОтвета.Вставить("Отправка", Отправка);
КонецЕсли;
СтруктураОтвета.Вставить("Результат", Истина);
СтруктураОтвета.Вставить("Заявки", МассивЗаказов2);
ВыгрузкаЗаказа(СтруктураОтвета);
Возврат СтруктураОтвета;
КонецФункции

```

Процедура передачі ідентифікатора користувача:

```

Процедура ПередатьИдентификаторПодписчикаУведомлений(
ИдентификаторПодписчикаУведомлений, ТекстОшибки) Экспорт
    Константы.ES_ИдентификаторПодписчикаУведомлений.Установить(
ИдентификаторПодписчикаУведомлений);
    СтрокаПосле = "";
    Соединение = ES_ОбщийМодульСервер.ПолучитьСоединениеСервера(СтрокаПосле);
    Запрос = Новый HTTPЗапрос;
    ПараметрыМА = Сериализовать(ИдентификаторПодписчикаУведомлений);
    Логин = Константы.ES_Логин.Получить();
    ApiKey = ES_ОбщийМодульСервер.ПолучитьЗначениеКонстанты("ES_ApiKey");
    АдресРесурса = СтрокаПосле + "/hs/es-mobile/getpushID?apikey=" + ApiKey +
"&reginfo=" + ПараметрыМА + "&login=" + Логин ;
    HTTPЗапрос = Новый HTTPЗапрос(АдресРесурса);
    HTTPЗапрос.Заголовки.Вставить("Content-Type", "text/xml;
charset=UTF8");
    Попытка
    Ответ = Соединение.Получить(HTTPЗапрос);
    Если Ответ.КодСостояния = 200 Тогда
        ОтветСтрока = Ответ.ПолучитьТелоКакСтроку();
        ОбработкаJSON = Обработки.JSON_and_UnJSON.Создать();
        ОтветСтруктура = ОбработкаJSON.UnJSON(ОтветСтрока);
        КонецЕсли;
    Искключение
    Если Ответ = Неопределено Тогда
        Сообщить("Push-уведомления не настроены");
    Иначе
    Если Ответ.КодСостояния >=300 Тогда
        Сообщить("Push-уведомления не настроены");
    КонецЕсли;
    КонецЕсли;
    КонецПопытки;
КонецПроцедуры

```

Функція створення копії елемента форми:

```

Функция СоздатьКопиюЭлемента(ЭлементИсточника, ПрефиксКопии, ГруппаЭлементов =
Неопределено)
    Тип = ТипЗнч(ЭлементИсточника);
    ЭлементПриемника = Элементы.Добавить(ЭлементИсточника.Имя + ПрефиксКопии,
Тип, ГруппаЭлементов);
    Если Тип = Тип("ГруппаФормы") Тогда
        ЭлементПриемника.Вид = ЭлементИсточника.Вид;
        Если ЭлементИсточника.Вид = ВидГруппыФормы.ОбычнаяГруппа Тогда
            ЗаполнитьЗначенияСвойств(ЭлементПриемника,
ЭлементИсточника, "ПутьКДаннымЗаголовка");
        ИначеЕсли ЭлементИсточника.Вид = ВидГруппыФормы.Страницы Тогда
            ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника, ,
"ТекущаяСтраница");
            СкопироватьСобытия(ЭлементПриемника, ЭлементИсточника);
        ИначеЕсли ЭлементИсточника.Вид = ВидГруппыФормы.Страница Тогда
            ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника, ,
"ПутьКДаннымЗаголовка");
        ИначеЕсли ЭлементИсточника.Вид = ВидГруппыФормы.ГруппаКолонок Тогда
            ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника, ,
"ПутьКДаннымШапки");

```



```

Иначе
    ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника);
КонецЕсли;
СкопироватьПодчиненныеЭлементы(ЭлементПриемника, ЭлементИсточника,
ПрефиксКопии);
ИначеЕсли Тип = Тип("ПолеФормы") Тогда
    ЭлементПриемника.Вид = ЭлементИсточника.Вид;
    ЭлементПриемника.ПутьКДанным = ЭлементИсточника.ПутьКДанным + ПрефиксКопии;
    Если ЭлементИсточника.Вид = ВидПоляФормы.ПолеВвода Тогда
        ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника,,
"ВыделенныйТекст, СвязьПоТипу, ПутьКДанным");
        Если ЭлементИсточника.СвязьПоТипу.ПутьКДанным <> "" Тогда
            ПутьКДаннымСвязиПриемника =
СтрЗаменить(ЭлементИсточника.СвязьПоТипу.ПутьКДанным, "Элементы." , "Элементы."
+ ПрефиксКопии);
            ЭлементПриемника.СвязьПоТипу = Новый
СвязьПоТипу(ПутьКДаннымСвязиПриемника,
ЭлементИсточника.СвязьПоТипу.ЭлементСвязи);
        КонецЕсли;
        ЗаполнитьКоллекциюСвойств(ЭлементПриемника.СписокВыбора,
ЭлементИсточника.СписокВыбора);
        ИначеЕсли ЭлементИсточника.Вид = ВидПоляФормы.ПолеПереключателя Тогда
            ЗаполнитьКоллекциюСвойств(ЭлементПриемника.СписокВыбора,
ЭлементИсточника.СписокВыбора);
            ЗаполнитьЗначенияСвойств(ЭлементПриемника,
ЭлементИсточника,, "ПутьКДанным");
            ИначеЕсли ЭлементИсточника.Вид = ВидПоляФормы.ПолеТекстовогоДокумента Тогда
                ЗаполнитьЗначенияСвойств(ЭлементПриемника,
ЭлементИсточника,, "ПутьКДанным, ВыделенныйТекст");
                ИначеЕсли ЭлементИсточника.Вид = ВидПоляФормы.ПолеФорматированногоДокумента
Тогда
                    ЗаполнитьЗначенияСвойств(ЭлементПриемника,
ЭлементИсточника,, "ПутьКДанным, ВыделенныйТекст");
                Иначе
                    ЗаполнитьЗначенияСвойств(ЭлементПриемника,
ЭлементИсточника,, "ПутьКДанным");
                КонецЕсли;
                СкопироватьСобытия(ЭлементПриемника, ЭлементИсточника);
            ИначеЕсли Тип = Тип("ТаблицаФормы") Тогда
                ЭлементПриемника.ПутьКДанным = ЭлементИсточника.ПутьКДанным +
ПрефиксКопии;
                ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника,,
"ПутьКДаннымКартинкиСтроки, ПутьКДанным, ДополнительныеПараметрыСоздания,
ЗапросОбновления");
                СкопироватьПодчиненныеЭлементы(ЭлементПриемника, ЭлементИсточника,
ПрефиксКопии);
                СкопироватьКнопкиКоманднойПанелиТаблицы(ЭлементПриемника.КоманднаяПанель,
ЭлементИсточника.КоманднаяПанель, ПрефиксКопии);
                СкопироватьСобытия(ЭлементПриемника, ЭлементИсточника);
            ИначеЕсли Тип = Тип("КнопкаФормы") Тогда
                ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника);
                СкопироватьКоманду(ЭлементПриемника, ЭлементИсточника, ПрефиксКопии);
            ИначеЕсли Тип = Тип("ДекорацияФормы") Тогда
                ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника);
                СкопироватьСобытия(ЭлементПриемника, ЭлементИсточника);
            Иначе
                ЗаполнитьЗначенияСвойств(ЭлементПриемника, ЭлементИсточника);
            КонецЕсли;
        
```

КонецФункции

Функція обробки структури справочника:

```

Процедура ОбработатьСтруктуруОтветаЗаполнитьСправочники(Контрагент)
    ВыборкаК = Справочники.Контрагенты.Выбрать();
    Пока ВыборкаК.Следующий() Цикл
        СпрКонтрагент = ВыборкаК.ПолучитьОбъект();
        СпрКонтрагент.Удалить();
        КонецЦикла;
        ВыборкаА = Справочники.Адреса.Выбрать();
        Пока ВыборкаА.Следующий() Цикл
            СпрАдреса = ВыборкаА.ПолучитьОбъект();
            СпрАдреса.Удалить();
            КонецЦикла;
        Для Каждого Эл Из Контрагент Цикл
            мКонт = Справочники.Контрагенты.НайтиПоРеквизиту("ID", Эл.ID);
            Если мКонт.Пустая() Тогда
                мКонт = Справочники.Контрагенты.СоздатьЭлемент();
                мКонт.Наименование = Эл.КонтрагентКлиент;
                мКонт.ID = Эл.ID;
                мКонт.МойКонтрагент = Эл.МойКлиент;
                мКонт.Записать();
            Иначе
                Если Не мКонт.Наименование = Эл.КонтрагентКлиент Тогда
                    мКонтОб = мКонт.Ссылка.ПолучитьОбъект();
                    мКонтОб.Наименование = Эл.КонтрагентКлиент;
                    мКонтОб.Записать();
                КонецЕсли;
            КонецЕсли;
            Для Каждого мАдрес Из Эл.Адрес Цикл
                мОтбор = Новый Структура;
                мОтбор.Вставить("Наименование", мАдрес);
                ВыборкаАдресов = Справочники.Адреса.Выбрать(, мКонт.Ссылка, мОтбор);
                Если Не ВыборкаАдресов.Следующий() Тогда
                    НовАдрес = Справочники.Адреса.СоздатьЭлемент();
                    НовАдрес.Владелец = мКонт.Ссылка;
                    НовАдрес.Наименование = мАдрес;
                    НовАдрес.Записать();
                КонецЕсли;
            КонецЦикла;
        КонецЦикла;
    КонецПроцедуры

```

Авторизация на сервері:

```

Функция ВыполнитьАвторизацию(УчетныеДанные) Экспорт
    СтруктураОтвета = Новый Структура;
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | ES_УчетныеЗаписиМобильногоПриложения.Контрагент КАК Контрагент,
        | ES_УчетныеЗаписиМобильногоПриложения.Пароль КАК Пароль,
        | ES_УчетныеЗаписиМобильногоПриложения.ТипПользователя КАК
ТипПользователя,
        | ES_УчетныеЗаписиМобильногоПриложения.Логин,
        | ES_УчетныеЗаписиМобильногоПриложения.Контрагент.ES_ДоступКМобильному

```

```

| ПОМЕСТИТЬ вт_пользовательmob
| ИЗ
| РегистрСведений.ES_УчетныеЗаписиМобильногоПриложения КАК
ES_УчетныеЗаписиМобильногоПриложения
| ГДЕ
|   ES_УчетныеЗаписиМобильногоПриложения.Логин = &Логин
| ///////////////////////////////////////////////////////////////////
| ВЫБРАТЬ
|   Контрагенты.Ссылка КАК Клиент,
|   Контрагенты.Код КАК КлиентЛогин,
|   Контрагенты.ES_ДоступКМобильному КАК КлиентДоступКМобильному,
|   Контрагенты.ES_ТипПользователя КАК КлиентТипПользователя
| ПОМЕСТИТЬ вт_клиентыагента
| ИЗ
|   Справочник.Контрагенты КАК Контрагенты
| ГДЕ
|   Контрагенты.ES_Агент В
|   (ВЫБРАТЬ
|     вт_пользовательmob.Контрагент
|   ИЗ
|     вт_пользовательmob КАК вт_пользовательmob)
| ;
| ///////////////////////////////////////////////////////////////////
| ВЫБРАТЬ
|   вт_пользовательmob.Контрагент,
|   вт_пользовательmob.Пароль,
|   вт_пользовательmob.ТипПользователя,
|   вт_пользовательmob.Логин,
|   вт_пользовательmob.КонтрагентES_ДоступКМобильному КАК
ДоступКМобильному
| ПОМЕСТИТЬ вт_общая
| ИЗ
|   вт_пользовательmob КАК вт_пользовательmob
|
| ОБЪЕДИНИТЬ ВСЕ
|
| ВЫБРАТЬ
|   вт_клиентыагента.Клиент,
|   вт_клиентыагента.КлиентТипПользователя,
|   вт_клиентыагента.КлиентЛогин,
|   вт_клиентыагента.КлиентДоступКМобильному
| ИЗ
|   вт_клиентыагента КАК вт_клиентыагента
| ;
| ///////////////////////////////////////////////////////////////////
|
| ВЫБРАТЬ
|   вт_общая.Контрагент КАК Контрагент,
|   вт_общая.Пароль КАК Пароль,
|   вт_общая.ТипПользователя КАК ТипПользователя,
|   вт_общая.Логин КАК Логин,
|   вт_общая.ДоступКМобильному КАК ДоступКМобильному,
|   ВложенныйЗапрос.Представление КАК Адрес
| ИЗ
|   вт_общая КАК вт_общая
| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ
|   КонтактнаяИнформация.Представление КАК Представление,
|   вт_общая.Контрагент КАК Контрагент

```

```

|ИЗ
|вт_общая КАК вт_общая
|ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КонтактнаяИнформация КАК
КонтактнаяИнформация
|ПО вт_общая.Контрагент = КонтактнаяИнформация.Объект
|ГДЕ
|КонтактнаяИнформация.Тип =
ЗНАЧЕНИЕ(Перечисление.ТипыКонтактнойИнформации.Адрес)) КАК ВложенныйЗапрос
|ПО вт_общая.Контрагент = ВложенныйЗапрос.Контрагент
|ИТОГИ
|МАКСИМУМ(Пароль),
|МАКСИМУМ(ТипПользователя),
|МАКСИМУМ(Логин),
|МАКСИМУМ(ДоступКМобильному)
|ПО
|Контрагент";
Запрос.УстановитьПараметр("Логин", СокрлП(УчетныеДанные.Получить("Логин")));
РезультатЗапроса = Запрос.Выполнить();
Если РезультатЗапроса.Пустой() Тогда
СтруктураОтвета.Вставить("Результат", Ложь);
СтруктураОтвета.Вставить("ТекстОшибки", "Выполните регистрацию.");
Возврат СтруктураОтвета;
КонецЕсли;
ВыборкаДетальныеЗаписиКонтрагент =
РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
ВыборкаДетальныеЗаписиКонтрагент.Следующий();
Если ВыборкаДетальныеЗаписиКонтрагент.Контрагент.ES_ДоступКМобильному Тогда
    Если УчетныеДанные.Получить("Пароль") =
ВыборкаДетальныеЗаписиКонтрагент.Пароль Тогда
        СтруктураОтвета.Вставить("Результат", Истина);
        СтруктураОтвета.Вставить("ТипПользователя",
ВыборкаДетальныеЗаписиКонтрагент.ТипПользователя);
        СтруктураОтвета.Вставить("Контрагент",
ВыборкаДетальныеЗаписиКонтрагент.Контрагент);
        Выборка = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
        Контрагенты = Новый Массив;
        Пока Выборка.Следующий() Цикл
            Клиенты = Новый Структура;
            Клиенты.Вставить("Клиент", Выборка.Контрагент);
            ВыборкаКлиент = Выборка.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
            МасАдреса = Новый Массив;
            Пока ВыборкаКлиент.Следующий() Цикл
                МасАдреса.Добавить(ВыборкаКлиент.Адрес);
            КонецЦикла;
            Клиенты.Вставить("Адрес", МасАдреса);
            Контрагенты.Добавить(Клиенты);
        КонецЦикла;
        СтруктураОтвета.Вставить("Контрагенты", Контрагенты);
Иначе
СтруктураОтвета.Вставить("Результат", Ложь);
    СтруктураОтвета.Вставить("ТекстОшибки", "Неправильный пароль.");
Возврат СтруктураОтвета;
КонецЕсли;
Иначе
СтруктураОтвета.Вставить("Результат", Ложь);
СтруктураОтвета.Вставить("ТекстОшибки", "Выполните регистрацию.");
Возврат СтруктураОтвета;
КонецЕсли;

```

Возврат СтруктураОтвета;
КонецФункции

Функция формування каталога на сервері:

```

Функция ПолучитьКаталогТоваров() Экспорт
    СтруктураОтвета = Новый Структура;
    МассивВитрины = Новый Массив;
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     Номенклатура.Наименование,
        |     ХранилищеДополнительнойИнформации.Хранилище,
        |     Номенклатура.Ссылка
        |ПОМЕСТИТЬ вт_1
        |ИЗ
        |     Справочник.Номенклатура КАК Номенклатура
        |     ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ХранилищеДополнительнойИнформации КАК
ХранилищеДополнительнойИнформации
        |     ПО Номенклатура.Ссылка =
ХранилищеДополнительнойИнформации.Объект
        |     И Номенклатура.ОсновноеИзображение =
ХранилищеДополнительнойИнформации.Ссылка
        |ГДЕ
        |     ХранилищеДополнительнойИнформации.ВидДанных =
ЗНАЧЕНИЕ(Перечисление.ВидыДополнительнойИнформацииОбъектов.Изображение)
        |     И Номенклатура.ES_ВыгружаетсяВМобильноеПриложение
        |;
        |
        |////////////////////////////////////
//
        |ВЫБРАТЬ
        |     ЦеныНоменклатуры.Номенклатура,
        |     КОЛИЧЕСТВО(РАЗЛИЧНЫЕ ЦеныНоменклатуры.ТипЦен) КАК ТипЦен,
        |     ЦеныНоменклатуры.Цена
        |ПОМЕСТИТЬ вт_2
        |ИЗ
        |     РегистрСведений.ЦеныНоменклатуры КАК ЦеныНоменклатуры
        |ГДЕ
        |     ЦеныНоменклатуры.ТипЦен = &Розничная
        |
        |СГРУППИРОВАТЬ ПО
        |     ЦеныНоменклатуры.Номенклатура,
        |     ЦеныНоменклатуры.Цена
        |;
        |
        |////////////////////////////////////
//
        |ВЫБРАТЬ
        |     вт_1.Наименование КАК Наименование,
        |     вт_1.Хранилище КАК Хранилище,
        |     вт_2.Цена КАК Цена
        |ИЗ
        |     вт_2 КАК вт_2
        |     ПРАВОЕ СОЕДИНЕНИЕ вт_1 КАК вт_1
        |     ПО вт_1.Ссылка = вт_2.Номенклатура";
    Розничная =
Справочники.ТипыЦенНоменклатуры.НайтиПоНаименованию("Розничная");

```

```

Запрос.УстановитьПараметр("Розничная", Розничная);
РезультатЗапроса = Запрос.Выполнить();
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    СтруктураЭлНом = Новый Структура;
    СтруктураЭлНом.Вставить("КартинкаТовара", Base64Строка(
ВыборкаДетальныеЗаписи.Хранилище.Получить().ПолучитьДвоичныеДанные()));
    СтруктураЭлНом.Вставить("НаименованиеТовара",
ВыборкаДетальныеЗаписи.Наименование);
    СтруктураЭлНом.Вставить("ЦенаТовара", ВыборкаДетальныеЗаписи.Цена);
    МассивВитрины.Добавить(СтруктураЭлНом);
КонецЦикла;
СтруктураОтвета.Вставить("МассивВитрины", МассивВитрины);
Возврат СтруктураОтвета;
КонецФункции

```

Функция получение структуры адреса по координатам:

```

Функция ПолучитьСтруктуруАдресаПоКоординатам(Долгота, Широта) Экспорт
    СтруктураАдреса = Новый Структура ("ПолныйАдрес, Город, Улица, Дом");
    АдресСервера = "geocode-maps.yandex.ru";
    АдресРесурса = "1.x/?apikey=0114e95f-bfb2-4101-b847-e510c6c061cf&geocode="
+ СтрЗаменить(Строка(Долгота), ",", ".", ".") + ","
+ СтрЗаменить(Строка(Широта), ",", ".", ".") + "&format=xml&results=1";
    Попытка
        Соединение = Новый HTTPСоединение(АдресСервера,,,,,Новый
ЗащищенноеСоединениеOpenSSL()
);
        ФайлРезультата = ПолучитьИмяВременногоФайла("xml");
        Соединение.Получить(АдресРесурса, ФайлРезультата);
        Соединение = Неопределено;
        Исключение
        КонецПопытки;
        ЧтениеXML = Новый ЧтениеXML;
        ЧтениеXML.ОткрытьФайл(ФайлРезультата);
        Пока ЧтениеXML.Прочитать() Цикл
            Если ЧтениеXML.Имя = "AddressLine" Тогда
                ЧтениеXML.Прочитать();
                Если ЧтениеXML.ТипУзла = ТипУзлаXML.Текст Тогда
                    Данные = ЧтениеXML.Значение;
                    СтруктураАдреса.Вставить("ПолныйАдрес", Данные);
                КонецЕсли;
            КонецЕсли;
            Если ЧтениеXML.Имя = "LocalityName" Тогда
                ЧтениеXML.Прочитать();
                Если ЧтениеXML.ТипУзла = ТипУзлаXML.Текст Тогда
                    Данные = ЧтениеXML.Значение;
                    СтруктураАдреса.Вставить("Город", Данные);
                КонецЕсли;
            КонецЕсли;
            Если ЧтениеXML.Имя = "ThoroughfareName" Тогда
                ЧтениеXML.Прочитать();
                Если ЧтениеXML.ТипУзла = ТипУзлаXML.Текст Тогда
                    Данные = ЧтениеXML.Значение;
                    СтруктураАдреса.Вставить("Улица", Данные);
                КонецЕсли;
            КонецЕсли;
        КонецЦикла;
    КонецФункции

```

```

Если ЧтениеXML.Имя = "PremiseNumber" Тогда
    ЧтениеXML.Прочитать();
Если ЧтениеXML.ТипУзла = ТипУзлаXML.Текст Тогда
    Данные = ЧтениеXML.Значение;
    СтруктураАдреса.Вставить("Дом", Данные);
КонецЕсли;
КонецЕсли;
КонецЦикла;
ЧтениеXML.Закреть();
ЧтениеXML = Неопределено;
УдалитьФайлы(ФайлРезультата);
Возврат СтруктураАдреса;
КонецФункции

```

Функція заповнення справочників:

```

Процедура ОбработатьСтруктуруОтветаЗаполнитьСправочник(Данные) Экспорт
    МассивВитрины = Данные.Данные.МассивВитрины;
    Для Каждого Эл Из МассивВитрины Цикл
        мНоменклатура =
Справочники.Номенклатура.НайтиПоНаименованию(Эл.НаименованиеТовара);
        Если мНоменклатура.Пустая() Тогда
            мНоменклатура = Справочники.Номенклатура.СоздатьЭлемент();
            мНоменклатура.Наименование = Эл.НаименованиеТовара;
            мНоменклатура.Записать();
        Иначе
            мНоменкл = мНоменклатура.Ссылка.ПолучитьОбъект();
            мНоменкл.Наименование = Эл.НаименованиеТовара;
            мНоменкл.Записать();
        КонецЕсли;
        мЦена = Справочники.Цена.НайтиПоНаименованию(Эл.ЦенаТовара);
        Если мЦена.Пустая() Тогда
            мЦена = Справочники.Цена.СоздатьЭлемент();
            мЦена.Наименование = Эл.ЦенаТовара;
            мЦена.Владелец = мНоменклатура.Ссылка;
            мЦена.Записать();
        Иначе
            мЦен = мЦена.Ссылка.Ссылка.ПолучитьОбъект();
            мЦен.Наименование = Эл.ЦенаТовара;
            мЦен.Владелец =
Справочники.Номенклатура.НайтиПоНаименованию(Эл.НаименованиеТовара);
            мЦен.Записать();
        КонецЕсли;
    КонецЦикла;
КонецПроцедуры

```

Функція отримання заказа на сервері

```

Функция ПолучитьЗаказ(Логин, Пароль, СтруктураОтвета) Экспорт
    МассивЗаявок = Новый Массив;
    СтруктураКонтрагент = ПолучитьКонтрагентаПоИдентификаторуИБ(Логин);
    Если СтруктураКонтрагент.Результат Тогда
        Контрагент = СтруктураКонтрагент.Контрагент;
    Иначе
        СтруктураОтвета.Вставить("Результат", Ложь);
        СтруктураОтвета.Вставить("ТекстОшибки", СтруктураКонтрагент.ТекстОшибки);
    Возврат СтруктураОтвета;

```

```

КонецЕсли;
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|   ЗаказПокупателя.Ссылка КАК Ссылка,
|   ЗаказПокупателя.Номер КАК Номер,
|   ЗаказПокупателя.Дата КАК Дата,
|   ЗаказПокупателя.АдресДоставки КАК ПолныйАдрес,
|   ЗаказПокупателя.Комментарий КАК Комментарий,
|   ЗаказПокупателя.Статус КАК Статус,
|   ЗаказПокупателя.ИнтервалВремя1 КАК ИнтервалВремя1,
|   ЗаказПокупателя.ИнтервалВремя2 КАК ИнтервалВремя2,
|   NULL КАК Поле1,
|   ЗаказПокупателя.СтруктурнаяЕдиница КАК ВидОплаты,
|   ЗаказПокупателя.Проведен,
|   ЗаказПокупателя.Контрагент.Наименование КАК Клиент,
|   ЗаказПокупателя.Товары. (
|       Номенклатура,
|       Количество,
|       Цена
|   ),
|   ЗаказПокупателя.ES_ДатаДоставки КАК ДатаДоставки
ИЗ
|   Документ.ЗаказПокупателя КАК ЗаказПокупателя
ГДЕ
|   НЕ ЗаказПокупателя.ПометкаУдаления
|   И ЗаказПокупателя.Дата >= &ДатаНеделяНазад
|   И ЗаказПокупателя.Контрагент В(&Контрагент)";
ДатаНеделяНазад = НачалоДня(ТекущаяДата())-7*24*3600;
Запрос.УстановитьПараметр("ДатаНеделяНазад", ДатаНеделяНазад);
Запрос.УстановитьПараметр("Контрагент", Контрагент);
РезультатЗапроса = Запрос.Выполнить();
Если РезультатЗапроса.Пустой() Тогда
СтруктураОтвета.Вставить("Результат", Ложь);
СтруктураОтвета.Вставить("ТекстОшибки", "Нет назначенных заявок.");
Возврат СтруктураОтвета;
КонецЕсли;
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.Следующий() Цикл
Если Выборка.Статус <> Перечисления.ES_СтатусыЗаказа.Отменен Тогда
СтруктураЗаявки = ПолучитьСтруктуруЗаявки();
ЗаполнитьЗначенияСвойств(СтруктураЗаявки, Выборка);
СтруктураЗаявки.УИД = Выборка.Ссылка.УникальныйИдентификатор();

    ВидОплаты = 0;
    Если ТипЗнч(Выборка.ВидОплаты) = Тип("СправочникСсылка.Кассы") Тогда
        ВидОплаты = 1;
    ИначеЕсли ТипЗнч(Выборка.ВидОплаты) =
Тип("СправочникСсылка.БанковскиеСчета") Тогда
        Если Выборка.ВидОплаты.Банк.Ссылка.Наименование = "Приват" Тогда
            ВидОплаты = 2;
        Иначе
            ВидОплаты = 3;
        КонецЕсли;
    КонецЕсли;
    СтруктураЗаявки.ВидОплаты = ВидОплаты;
    МассивТоваров = Новый Массив;
    Товары = Выборка.Товары.Выбрать();
    Пока Товары.Следующий() Цикл
        СтруктураТоваров = Новый Структура;

```



```

СтруктураТоваров.Вставить("Позиция", Товары.Номенклатура);
СтруктураТоваров.Вставить("Количество", Товары.Количество);
СтруктураТоваров.Вставить("Цена", Товары.Цена);
МассивТоваров.Добавить(СтруктураТоваров);
КонецЦикла;
СтруктураЗаявки.Товары = МассивТоваров;
МассивЗаявок.Добавить(СтруктураЗаявки);
КонецЕсли;
КонецЦикла; |
СтруктураОтвета.Вставить("Результат", Истина);
СтруктураОтвета.Вставить("Заявки", МассивЗаявок);
Возврат СтруктураОтвета;

```

КонецФункции

Функція получения контрагента по ідентифікатору:

Функция ПолучитьКонтрагентаПоИдентификаторуИБ(Код) Экспорт

```

СтруктураОтвета = Новый Структура;
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     ES_УчетныеЗаписиМобильногоПриложения.Контрагент КАК Контрагент,
|     Контрагенты.Ссылка КАК КонтрагентКлиент
|ИЗ
|     РегистрСведений.ES_УчетныеЗаписиМобильногоПриложения КАК
ES_УчетныеЗаписиМобильногоПриложения
|     ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Контрагенты КАК Контрагенты
|     ПО ES_УчетныеЗаписиМобильногоПриложения.Контрагент =
Контрагенты.ES_Агент
|ГДЕ
|     ES_УчетныеЗаписиМобильногоПриложения.Логин = &ИдентификаторИБ";
Запрос.УстановитьПараметр("ИдентификаторИБ", СокрЛП(Код));
РезультатЗапроса = Запрос.Выполнить();
Если РезультатЗапроса.Пустой() Тогда
СтруктураОтвета.Вставить("Результат", Ложь);
СтруктураОтвета.Вставить("ТекстОшибки", "Идентификатор информационной базы
не зарегистрирован. Проверьте параметры подключения.");
Возврат СтруктураОтвета;
КонецЕсли;
Выборка = РезультатЗапроса.Выбрать();
МасКлиентов = Новый Массив;
Пока Выборка.Следующий() цикл
Если МасКлиентов.Найти(Выборка.Контрагент)= Неопределено Тогда
МасКлиентов.Добавить(Выборка.Контрагент);
КонецЕсли;
МасКлиентов.Добавить(Выборка.КонтрагентКлиент);
КонецЦикла;
СтруктураОтвета.Вставить("Результат", Истина);
СтруктураОтвета.Вставить("Контрагент", МасКлиентов);
Возврат СтруктураОтвета;

```

КонецФункции

Функція формування структури замовлення:

Функция ПолучитьСтруктуруЗаявки() Экспорт

```

Структура = Новый Структура;
Структура.Вставить("УИД");
Структура.Вставить("Номер");

```

```

Структура.Вставить("Дата");
Структура.Вставить("Статус");
Структура.Вставить("Клиент");
Структура.Вставить("ВидОплаты");
Структура.Вставить("ПолныйАдрес");
Структура.Вставить("ДатаДоставки");
Структура.Вставить("ИнтервалВремя1");
Структура.Вставить("ИнтервалВремя2");
Структура.Вставить("Комментарий");
МассивТоваров = Новый Массив;
СтруктураТоваров = Новый Структура;
СтруктураТоваров.Вставить("Позиция");
СтруктураТоваров.Вставить("Количество");
МассивТоваров.Добавить(СтруктураТоваров);
Структура.Вставить("МассивТоваров");
Возврат Структура;

```

КонецФункции

Функция установки идентификатора:

```

Функция УстановитьIDУстройства(ИдентификаторУстройства, ТекстЗапроса) Экспорт
    СтруктураОтвета = Новый Структура();
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | ES_УчетныеЗаписиМобильногоПриложения.Контрагент,
        | ES_УчетныеЗаписиМобильногоПриложения.Логин,
        |     ES_УчетныеЗаписиМобильногоПриложения.ИдентификаторМобУстройства
        | ИЗ
        |     РегистрСведений.ES_УчетныеЗаписиМобильногоПриложения КАК
        ES_УчетныеЗаписиМобильногоПриложения
        | ГДЕ
        | ES_УчетныеЗаписиМобильногоПриложения.Логин = &Логин";
    Запрос.УстановитьПараметр("Логин", СокрЛП(ИдентификаторУстройства));
    РезультатЗапроса = Запрос.Выполнить();
    Если РезультатЗапроса.Пустой() Тогда
        СтруктураОтвета.Вставить("Результат", Ложь);
        СтруктураОтвета.Вставить("ТекстОшибки", "Идентификатор информационной базы
не зарегистрирован.");
        Возврат СтруктураОтвета;
        КонецЕсли;
        Выборка = РезультатЗапроса.Выбрать();
        Если Выборка.Количество() > 1 Тогда
            СтруктураОтвета.Вставить("Результат", Ложь);
            СтруктураОтвета.Вставить("ТекстОшибки", "Идентификатор информационной базы
не уникален.");
            Возврат СтруктураОтвета;
            КонецЕсли;
            Выборка.Следующий();
            Данные = Десериализовать(ТекстЗапроса);
            МЗ =
РегистрыСведений.ES_УчетныеЗаписиМобильногоПриложения.СоздатьМенеджерЗаписи();
            МЗ.Контрагент = Выборка.Контрагент;
            МЗ.Прочитать();
            МЗ.ИдентификаторМобУстройства = Новый ХранилищеЗначения(Данные);
            МЗ.Записать(Ложь);
            СтруктураОтвета.Вставить("Результат", Истина);
            Возврат СтруктураОтвета;

```

КонецФункции

Функция створення замовлень на сервере:

Функция СоздатьЗаказПокупателя (Логин, Пароль, СтруктураОтвета) Экспорт

Если СтруктураОтвета.Получить("Результат") Тогда

Если СтруктураОтвета.Получить("Отправка") = 1 Тогда

МассивЗаказов = СтруктураОтвета.Получить("Заявки");

Для Каждого Позиция из МассивЗаказов Цикл

Если НЕ Позиция.Получить("УИД") = "" Тогда

НовыйGUID = Новый УникальныйИдентификатор(Позиция.Получить("УИД"));

док = Документы.ЗаказПокупателя.ПолучитьСсылку(НовыйGUID);

Если ЗначениеЗаполнено(док) Тогда

ЗаказМобильное = док.ПолучитьОбъект();

ЗаказМобильное.ЭтотОбъект.Товары.Очистить();

Иначе

ЗаказМобильное = Документы.ЗаказПокупателя.СоздатьДокумент();

КонецЕсли;

Иначе

ЗаказМобильное = Документы.ЗаказПокупателя.СоздатьДокумент();

КонецЕсли;

ЗаказМобильное.Дата =

ПривестиКТипуДата(СтрЗаменить(Позиция.Получить("Дата"), "- ", ". "));

ЗаказМобильное.Статус =

ПолучитьЗначениеПеречисления("ES_СтатусыЗаказа", Позиция.Получить("Статус"));

ЗаказМобильное.Контрагент =

Справочники.Контрагенты.НайтиПоНаименованию(Позиция.Получить("Клиент"));

ЗаказМобильное.ES_ДатаДоставки =

ПривестиКТипуДата(СтрЗаменить(Позиция.Получить("ДатаДоставки"), "- ", ". "));

ЗаказМобильное.ИнтервалВремя1 =

ПолучитьЗначениеПеречисления("ES_Интервалы", СтрокаПеречисления(Позиция.Получить("ИнтервалВремя1")));

ЗаказМобильное.ИнтервалВремя2 =

ПолучитьЗначениеПеречисления("ES_Интервалы", СтрокаПеречисления(Позиция.Получить("ИнтервалВремя2")));

ЗаказМобильное.Комментарий = Позиция.Получить("Комментарий");

Если Позиция.Получить("ВидОплаты") = 1 Тогда

ЗаказМобильное.СтруктурнаяЕдиница =

Справочники.ES_ПредопределенныеСчета.НайтиПоНаименованию("Основная Касса");

ИначеЕсли Позиция.Получить("ВидОплаты") = 3 Тогда

Если Справочники.БанковскиеСчета.НайтиПоНаименованию("Приват") <> "" Тогда

ЗаказМобильное.СтруктурнаяЕдиница =

Справочники.БанковскиеСчета.НайтиПоНаименованию("Приват");

Иначе

```

НовыйБанк = Справочники.БанковскиеСчета.СоздатьЭлемент();
НовыйБанк.Наименование = "Приват";
НовыйБанк.Записать();
ЗаказМобильное.СтруктурнаяЕдиница =
Справочники.БанковскиеСчета.НайтиПоНаименованию("Приват");
КонецЕсли;
ИначеЕсли Позиция.Получить("ВидОплаты") = 2 Тогда
    ЗаказМобильное.СтруктурнаяЕдиница =
Справочники.ES_ПредопределенныеСчета.НайтиПоНаименованию("Основной Банк");
КонецЕсли;
    ЗаказМобильное.АдресДоставки = Позиция.Получить("ПолныйАдрес");
    Товары = Позиция.Получить("Товары");
    Для каждого Эл Из Товары Цикл
        НоваяСтрока = ЗаказМобильное.ЭтотОбъект.Товары.Добавить();
        НоваяСтрока.Номенклатура =
Справочники.Номенклатура.НайтиПоНаименованию(Эл.Получить("Позиция"));
        НоваяСтрока.Количество = Эл.Получить("Количество");
        НоваяСтрока.Цена = Эл.Получить("Цена");
    КонецЦикла;
    ЗаказМобильное.Записать(РежимЗаписиДокумента.Запись);
    СтрОтвет = Новый Структура;
    Если Позиция.Получить("УИД") = "" Тогда
        УИД = ЗаказМобильное.Ссылка.УникальныйИдентификатор();
        СтрОтвет.Вставить("Уид", УИД);
    Иначе
        СтрОтвет.Вставить("Уид", Позиция.Получить("УИД"));
    КонецЕсли;
    КонецЦикла
    Иначе
        МассивЗаказов = СтруктураОтвета.Получить("Заявки");
        Для Каждого Позиция из МассивЗаказов Цикл
            НовыйGUID = Новый УникальныйИдентификатор(Позиция.Получить("УИД"));
            док = Документы.ЗаказПокупателя.ПолучитьСсылку(НовыйGUID);
            ЗаказМобильное = док.ПолучитьОбъект();
            ЗаказМобильное.Статус =
ПолучитьЗначениеПеречисления("ES_СтатусыЗаказа", "Отменен");
            ЗаказМобильное.Записать(РежимЗаписиДокумента.Запись);
            СтрОтвет = Новый Структура;
            Если ЗаказМобильное.Проведен Тогда
                ДокПроведен = 1;

```

```

        СтрОтвет.Вставить("ДокПроведен", ДокПроведен);
Иначе
    ДокПроведен = 0;
    СтрОтвет.Вставить("ДокПроведен", ДокПроведен);
КонецЕсли;
КонецЦикла
КонецЕсли;
Иначе
    ТекстОтвета = "Ошибка при подключении";
    СтрОтвета = Новый Структура;
    СтрОтвета.Вставить("Результат", Ложь);
    СтрОтвета.Вставить("ТекстОшибки", ТекстОтвета);
КонецЕсли;
Возврат СтрОтвет;
КонецФункции

```

Процедура відправки повідомлення:

```

Процедура ОтправитьУведомление(Уведомление, Логин) Экспорт
    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
    |     ES_УчетныеЗаписиМобильногоПриложения.Контрагент,
    |     ES_УчетныеЗаписиМобильногоПриложения.Логин,
    |     ES_УчетныеЗаписиМобильногоПриложения.Пароль,
    |     ES_УчетныеЗаписиМобильногоПриложения.ИдентификаторМобУстройства
    |ИЗ
    |     РегистрСведений.ES_УчетныеЗаписиМобильногоПриложения КАК
ES_УчетныеЗаписиМобильногоПриложения
    |ГДЕ
    |     ES_УчетныеЗаписиМобильногоПриложения.Логин = &Логин";
    Запрос.УстановитьПараметр("Логин", Логин);
    Выборка = Запрос.Выполнить().Выбрать();
    Пока Выборка.Следующий() Цикл
        ПараметрыУстройства = Выборка.ИдентификаторМобУстройства.Получить();
        Если ПараметрыУстройства = Неопределено или ПараметрыУстройства = "" Тогда
            Продолжить;
        Иначе
            Уведомление.Получатели.Добавить(Выборка.ИдентификаторМобУстройства.Получить(
    ))
        КонецЕсли;
    КонецЦикла;
    Если Уведомление.Получатели.Количество() > 0 Тогда
        ДанныеАутентификации = "";
        Сертификат = Неопределено;
        КлючGoogle =
"AAAAfSjtaH4:APA91bHuu45roH230fNjOrnfJHldPelujNtPcLWDabGeUDNaYTL0zLT-
FZNsqNtNZtosofpjFs1Qi6XqW91C9L0gQtntIh0-geA9cHGdgS9Ga70gT4905EPpJWGADohjf-
dhHteo5412";
        ДанныеАутентификации = Новый Соответствие;
        ДанныеАутентификации.Вставить(ТипПодписчикаДоставляемыхУведомлений.FCM,
КлючGoogle);
    
```

```

    УдаленныеТокены = Новый Массив;
    ИнформацияОПроблемеОтправкиДоставляемогоУведомления = Новый Массив;
    Попытка
    ОтправкаДоставляемыхУведомлений.Отправить(Уведомление, ДанныеАутентификации, У
даленныеТокены, Ложь);
    Исключение
    Сообщить(" " + ИнформацияОбОшибке().Описание+ " " +
ИнформацияОбОшибке().ИмяМодуля + "ошибкаисключения 1");
    КонецПопытки;
    НеИспользоватьИдентификаторы(УдаленныеТокены);
    КонецЕсли;
КонецПроцедуры

```

Процедура на формі авторизації

```

&НаСервере
Процедура ВойтиНаСервере(Авторизован)
Ответ = ES_ОбщийМодульСервер.ВыполнитьАвторизацию(ЛогинАвторизация,
ПарольАвторизация);
Если Ответ.Результат Тогда
    Если Ответ.ТипПользователя = 0 Тогда
        ES_ОбщийМодульСервер.УстановитьЗначениеКонстанты("ES_ТипПользователя", ES_Общ
ийМодульСервер.ПолучитьЗначениеПеречисления("ES_ТипПользователя", "Агент"));
    Иначе
        ES_ОбщийМодульСервер.УстановитьЗначениеКонстанты("ES_ТипПользователя", ES_ОбщийМод
ульСервер.ПолучитьЗначениеПеречисления("ES_ТипПользователя", "Клиент"));
    КонецЕсли;
    Если Запомнить Тогда
        Константы.ES_Логин.Установить(ЛогинАвторизация);
        Константы.ES_Пароль.Установить(ПарольАвторизация);
    Иначе
        Константы.ES_Логин.Установить(ЛогинАвторизация);
        Константы.ES_Пароль.Установить("");
    КонецЕсли;
    ОбработатьСтруктуруОтветаЗаполнитьСправочники(Ответ.Контрагент);
    Авторизован = Истина;
    Иначе
        Сообщить(Ответ.ТекстОшибки);
        Авторизован = Ложь;
    КонецЕсли;
КонецПроцедуры

```

Процедура на формі Вітрини:

```

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
Данные = ES_ОбщийМодульСервер.ПолучитьКаталогТоваров();
Если Данные.Результат Тогда
    МассивВитрины = Данные.Данные.МассивВитрины;
    ДобавляемыеРеквизиты = Новый Массив;
    Для Счетчик = 0 По МассивВитрины.Количество() Цикл
        мРеквизит = Новый РеквизитФормы("Картинка_"+Счетчик, Новый
ОписаниеТипов("Строка"));
        ДобавляемыеРеквизиты.Добавить(мРеквизит);
    КонецЦикла;
    ИзменитьРеквизиты(ДобавляемыеРеквизиты);
    Для Каждого Позиция Из МассивВитрины Цикл

```

```
        СоздатьКопиюЭлемента(Элементы.ОписаниеВоды_,  
МассивВитрины.Найти(Позиция), Элементы.ВитринаВоды);  
        Элементы["ОписаниеВоды_" + МассивВитрины.Найти(Позиция)].Видимость = Истина;  
        Элементы["Название_" + МассивВитрины.Найти(Позиция)].Заголовок =  
Позиция.НаименованиеТовара;  
        Элементы["Цена_" + МассивВитрины.Найти(Позиция)].Заголовок =  
Позиция.ЦенаТовара;  
        мДвоичныеДанные = Base64Значение(Позиция.КартинкаТовара);  
        ИмяФайла = КаталогВременныхФайлов() + ПолучитьИмяВременногоФайла("jpg");  
        мДвоичныеДанные.Записать(ИмяФайла);  
        ЭтаФорма["Картинка_" + МассивВитрины.Найти(Позиция)] =  
ПоместитьВоВременноеХранилище(Новый Картинка(ИмяФайла));  
        удалитьФайлы(ИмяФайла);  
        КонецЦикла;  
        ES_ОбщийМодульСервер.ОбработатьСтруктуруОтветаЗаполнитьСправочник(Данные);  
        Иначе  
        Сообщить(Данные.ТекстОшибки);  
        КонецЕсли;  
КонецПроцедуры
```