

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка системи управління запасами для магазину канцелярії з використанням Google AppSheet

Виконав: студент IV курсу, групи СП-43 спеціальності
121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Радецький Я. М.

(підпис)

(прізвище та ініціали)

Керівник

Бревус В.М.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Стоянов Ю. М.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М. Р.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

2025 p.

Діаграми, знімки екрану з проробленою роботою, ілюстративні зображення, інформативні зображення для доповнення тексту.

АНОТАЦІЯ

Розробка системи управління запасами для магазину канцелярії з використанням Google AppSheet // Кваліфікаційна робота освітнього рівня «Бакалавр» // Радецький Ярослав Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-43 // Тернопіль, 2025 // Ст. – __, рис. – 20, табл. – 8, додат. – 1, бібліогр. – 26.

Ключові слова: Google AppSheet, облік товарів, управління запасами, Google Sheets, мобільний застосунок, звітність, low-code платформа.

Головною метою цієї кваліфікаційної роботи є розробка простої та ефективної системи управління товарними залишками для магазину канцелярії з використанням платформи Google AppSheet як інструменту безкодової розробки.

Перший розділ присвячено аналізу предметної області, проблем обліку на складах малого бізнесу, особливостей low-code платформ, огляду наявних рішень та обґрунтуванню вибору AppSheet для реалізації системи.

У другому розділі розглянуто архітектуру розробленого рішення, описано модель предметної області, структуру таблиць, реалізацію логіки обліку, формування звітів, адаптацію інтерфейсу та ролей користувачів, а також тестування системи в умовах модельного магазину. Також представлено результати впровадження, оцінку ефективності використання платформи Google AppSheet, а також перспективи масштабування, інтеграції з іншими сервісами та подальшого розвитку функціональності для реальних торгових об'єктів.

У третьому розділі проведено огляд проблематики Безпеки життєдіяльності та охорони праці відповідно до завдань консультанта.

ABSTRACT

Development of an Inventory Management System for a Stationery Store Using Google AppSheet // Bachelor's Qualification Work // Radetskyi Yaroslav Mykolaiovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group SP-43 // Ternopil, 2025 // Pages – __, Figures – 20, Tables – 8, Appendices – 1, References – 26.

Keywords: Google AppSheet, inventory tracking, stock management, Google Sheets, mobile application, reporting, low-code platform.

The main goal of this qualification work is the development of a simple and efficient inventory management system for a stationery store using the Google AppSheet platform as a no-code development tool.

The first section is dedicated to analyzing the subject area, challenges of inventory accounting in small business warehouses, features of low-code platforms, a review of existing solutions, and justification for selecting AppSheet for system implementation.

The second section presents the architecture of the developed solution, describes the domain model, table structure, implementation of inventory logic, report generation, interface and user role adaptation, and system testing in the context of a model store. It also includes the results of deployment, an evaluation of the effectiveness of using the Google AppSheet platform, and prospects for scaling, integration with other services, and further development of functionality for real commercial environments.

The third section provides an overview of occupational safety and health issues in accordance with the consultant's requirements.

ЗМІСТ

АНОТАЦІЯ	4
ABSTRACT	5
ЗМІСТ	6
ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1. АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1 Аналіз предметної області та теоретичні основи	11
1.2 Інструменти та технології розробки на базі Google AppSheet.....	12
1.3 Огляд існуючих рішень та конкурентів	15
1.4 Обґрунтування вибору середовища та технологій.....	18
1.5 Вартісна ефективність та оцінка low-code підходу.....	20
1.6 Безпека, контроль доступу та перспективи масштабування системи.....	21
2.ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ	24
2.1 Визначення вимог системи	24
2.1.1 Функціональні вимоги	24
2.1.2 Нефункціональні вимоги	26
2.1.3 Варіанти використання	27
2.1 Модель системи	32
2.2 Модель даних предметної області	34
2.3 Взаємодія системи	37
2.4 Реалізація системи	40
2.4.1 Використання шаблону AppSheet Inventory Management	40
2.4.2 Адаптація структури таблиць до предметної області.....	43
2.4.3 Розробка логіки обліку залишків	45
2.4.4 Налаштування форм і дій для ролей користувачів	46
2.4.5 Автоматизація процесів з використанням Bots і Actions	48
2.4.6 Реалізація звітів та аналітики	51
2.4.7 Забезпечення доступу з різних пристроїв.....	52
2.5 Результати тестування.....	53
2.6 Впровадження системи	55

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	57
3.1 Динамічні явища на поверхні землі.....	57
3.2 Організація ведення робіт в аварійних умовах	59
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ.....	68
Додаток А – Диск із кваліфікаційною роботою бакалавра	69

ПЕРЕЛІК СКОРОЧЕНЬ

API – Application Programming Interface (програмний інтерфейс прикладного програмування)

CI/CD – Continuous Integration / Continuous Delivery (безперервна інтеграція / безперервне постачання)

UI – User Interface (інтерфейс користувача)

UX – User Experience (досвід користувача)

CRUD – Create, Read, Update, Delete (створення, читання, оновлення, видалення)

ID – Identifier (ідентифікатор)

CSV – Comma-Separated Values (значення, розділені комами)

PDF – Portable Document Format (формат переносимого документа)

URL – Uniform Resource Locator (уніфікований локатор ресурсу)

AppSheet – Google AppSheet (платформа безкодової розробки додатків)

DB – Database (база даних)

OTP – One-Time Password (одноразовий пароль)

B2B – Business to Business (бізнес для бізнесу)

JSON – JavaScript Object Notation (текстовий формат обміну даними)

ВСТУП

Сьогодні інформаційні технології охоплюють практично всі сфери людської діяльності, і торгівля — не виняток. Зростання обсягів обробки даних, потреба в швидкому реагуванні на зміни попиту, конкурентний тиск з боку онлайн-платформ та зростаючі очікування клієнтів вимагають від підприємств автоматизованого та прозорого управління внутрішніми процесами. Одним із ключових напрямів цифровізації малого бізнесу є впровадження систем обліку товарів і управління запасами, що дозволяє не лише підвищити точність ведення складського обліку, а й приймати обґрунтовані управлінські рішення на основі реальних даних.

Магазини канцелярських товарів, як і багато інших локальних торгових точок, часто стикаються з труднощами в контролі залишків, своєчасному поповненні товарів та веденні історії операцій. Використання паперових журналів або базових електронних таблиць без належної структури призводить до втрат часу, неточностей в обліку, дублювання позицій, помилок під час інвентаризації та неможливості аналізу динаміки продажів. Водночас, ринок пропонує багато рішень для автоматизації, однак більшість із них або занадто складні для впровадження, або потребують значних фінансових витрат і залучення технічних фахівців.

З появою технологій low-code / no-code розробка бізнес-застосунків стала доступною навіть для користувачів без глибоких знань програмування. Одним із таких інструментів є Google AppSheet — платформа, що дозволяє будувати повнофункціональні мобільні та веб-застосунки на основі Google Sheets, Excel, SQL-баз даних та інших джерел, використовуючи зручний інтерфейс та правила автоматизації. Це дозволяє значно скоротити час розробки, забезпечити зручність у роботі та легко масштабувати систему відповідно до потреб бізнесу.

Актуальність теми полягає в необхідності забезпечення простого, швидкого та ефективного способу управління товарними залишками в невеликих торгових закладах, що не мають IT-відділу, але потребують автоматизації. Обрана тема

дозволяє поєднати сучасні технології з реальними потребами ринку, пропонуючи дієвий приклад цифрової трансформації малого бізнесу.

Метою роботи є розробка системи управління запасами для магазину канцелярських товарів із використанням Google AppSheet, що дозволяє адміністратору, продавцю та комірнику взаємодіяти з єдиною базою даних у реальному часі, відображати поточні залишки, реєструвати надходження та продажі, формувати звіти, а також отримувати автоматичні сповіщення у разі досягнення критичного рівня запасів.

Об'єктом дослідження є процеси обліку товарних залишків і операцій із товарами в умовах роздрібно́ї торгівлі.

Предметом дослідження виступають методи, засоби та платформи для побудови автоматизованих систем обліку з використанням low-code технологій.

Для досягнення поставленої мети в роботі було здійснено аналіз предметної області, зокрема проблем ведення обліку та можливостей автоматизації в сфері роздрібно́ї торгівлі. Проведено огляд наявних інструментів автоматизації та обґрунтовано доцільність вибору платформи Google AppSheet. Сформульовано функціональні та нефункціональні вимоги до системи, розроблено структуру бази даних, модель предметної області та архітектуру майбутнього рішення. Реалізовано основні функції обліку товарів у середовищі AppSheet, проведено тестування системи на прикладі модельного магазину канцелярських товарів, а також проаналізовано можливості масштабування та подальшого розвитку створеної системи.

Результатом виконаної роботи є зручна у користуванні, безпечна та доступна система управління запасами, яка може бути впроваджена не лише в магазинах канцелярії, а й адаптована для інших типів роздрібно́ї торгівлі.

1. АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЄКТУ

1.1 Аналіз предметної області та теоретичні основи

У сучасних умовах конкуренції в сфері роздрібної торгівлі важливу роль відіграє ефективне управління товарними запасами. Магазины, що спеціалізуються на продажу канцелярських товарів, часто стикаються з необхідністю підтримання актуальної інформації про наявність продукції, своєчасного поповнення асортименту та уникнення надлишкових залишків. Без належної автоматизації цих процесів зростає ризик людських помилок, що може призводити до фінансових втрат і зниження рівня обслуговування клієнтів [1].

Предметною областю даної кваліфікаційної роботи є система управління запасами, адаптована до специфіки роботи малого магазину канцелярських товарів. Така система має забезпечувати облік товарів, контроль залишків, реєстрацію надходжень і списань, а також формування звітів. Її реалізація повинна бути простою у впровадженні, доступною для користувачів без спеціальних технічних знань та не вимагати значних фінансових витрат. В умовах обмежених ресурсів доцільним є використання платформи Google AppSheet, яка дозволяє створювати функціональні додатки на основі таблиць Google Sheets без необхідності програмування. AppSheet надає інструменти для побудови інтерфейсів, реалізації логіки бізнес-процесів та доступу до системи з мобільних або десктопних пристроїв. Такий підхід дозволяє суттєво скоротити час розробки та зосередитися на потребах конкретного бізнесу.

В теоретичній частині розглядаються основні підходи до управління товарними запасами, класифікація систем обліку, базові принципи побудови інформаційних систем, а також можливості та обмеження платформи AppSheet. Аналіз існуючих рішень дозволяє оцінити сучасні підходи до вирішення аналогічних задач і обґрунтувати вибір запропонованої архітектури та інструментів.

1.2 Інструменти та технології розробки на базі Google AppSheet

Google AppSheet — це хмарна no-code платформа для створення мобільних та веб-додатків без програмування. Вона дозволяє використовувати дані з таблиць Google Sheets, Excel, SQL-баз та інших джерел для швидкого створення повнофункціональних додатків. AppSheet підтримує створення інтерфейсів для введення та перегляду даних, реалізацію бізнес-логіки, створення звітів, а також надає можливості для автоматизації процесів за допомогою «Bots» та умовної логіки [2].

Платформа дозволяє реалізовувати додатки для управління запасами, CRM-системи, трекери завдань, інвентаризаційні рішення тощо. У контексті магазину канцелярських товарів AppSheet є зручним інструментом для створення системи обліку товарів, ведення залишків, формування звітів про рух продукції, а також фіксації надходжень і продажів [3] (табл. 1.1).

Серед основних технологій, що використовуються у межах розробки системи:

- Google Sheets — база даних для зберігання записів;
- AppSheet Editor — вебінтерфейс для проєктування логіки і вигляду додатка;
- AppSheet Automation — засіб для реалізації сценаріїв автоматичної обробки даних;
- AppSheet UX Editor — візуальне налаштування форм, списків і фільтрів;
- AppSheet Security & Access — інструменти контролю доступу до даних і функцій додатка.

Таблиця 1.1 – Характеристика Google AppSheet у контексті управління запасами

Критерій	Характеристика Google AppSheet
Тип платформи	No-code (не потребує програмування)
Сумісність з даними	Google Sheets, Excel, SQL, Smartsheet тощо
Швидкість розробки	Висока, додаток можна створити за 1–2 дні
Зручність для малого бізнесу	Дуже висока — проста настройка, низький поріг входу
Функціональність для запасів	Облік товарів, залишків, руху, попереджень, фільтрація, сортування
Автоматизація процесів	Bots: нагадування, сповіщення, оновлення даних
Інтерфейс користувача	Мобільний та веб, настраюється візуально, адаптивний
Контроль доступу	Авторизація через Google, розмежування прав
Мобільний доступ	Повнофункціональний додаток для Android та iOS
Вартість	Є безкоштовний план, платні тарифи за потреби
Обмеження	Підходить для невеликих обсягів даних (до ~100 тис. рядків у Sheets)

Google Sheets використовується як основне джерело даних для AppSheet-додатків. Це хмарна електронна таблиця, яка дозволяє зберігати, організовувати та змінювати інформацію у структурованому вигляді. Її перевагами є простота використання, можливість спільного доступу до файлу, автоматичне збереження змін, а також гнучкість у побудові структури даних (рис. 1.1) [4, 5].

AppSheet Editor — це середовище розробки, доступне через браузер, яке дозволяє створювати та налаштовувати логіку роботи додатка, формувати інтерфейс, встановлювати зв'язки між таблицями та задавати поведінку системи залежно від умов. У ньому реалізовано перевірку даних, обробку введення, дії на кнопках та інші важливі механізми.

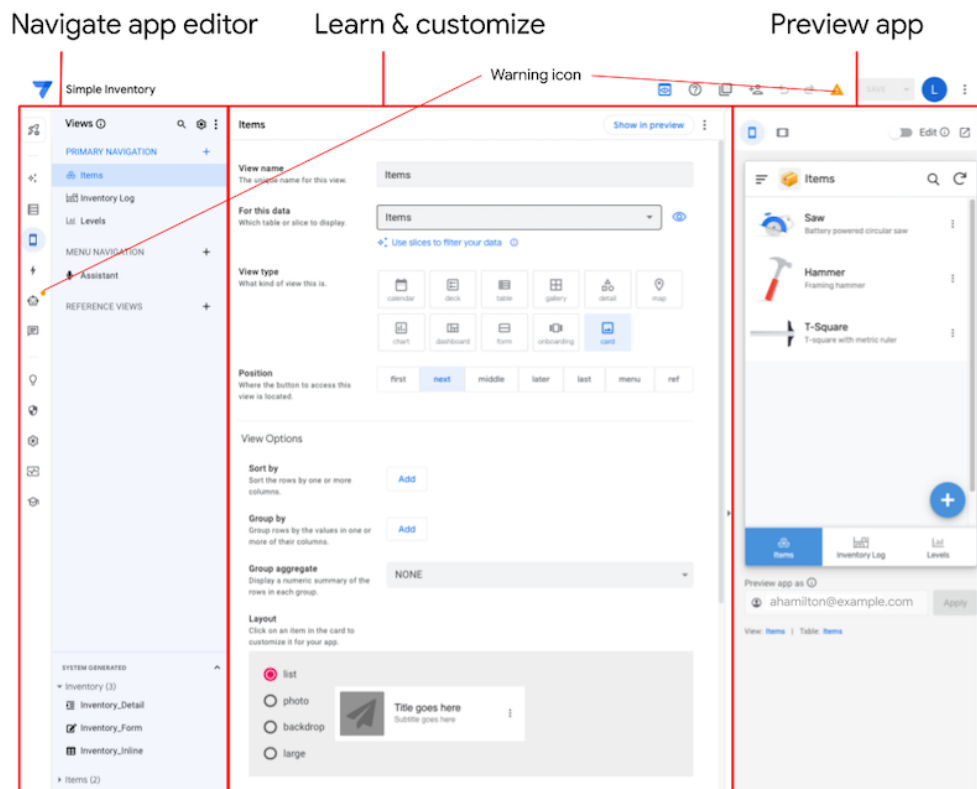


Рисунок 1.1 – Інтерфейс Google AppSheet

AppSheet Automation дає змогу створювати автоматизовані сценарії — так звані «bots». Це потужний інструмент для автоматичного виконання певних дій, таких як надсилання сповіщень, оновлення записів, створення нових рядків у таблиці при виконанні певних умов. Для системи обліку запасів це, наприклад, автоматичне повідомлення при досягненні мінімального рівня залишків [6].

AppSheet UX Editor відповідає за зовнішній вигляд додатка. Тут налаштовуються вигляд форм, списків, таблиць, фільтрів, панелей перегляду та дій користувача. UX-редактор дозволяє змінювати кольори, піктограми, елементи

інтерфейсу відповідно до потреб бізнесу, що робить додаток більш зручним для персоналу магазину [7].

AppSheet Security & Access надає функціональність для контролю доступу до додатка та його окремих частин. Тут налаштовується, хто може переглядати або змінювати інформацію, можливість обмеження доступу за логіном Gmail, ролями користувачів чи умовами. Це забезпечує базовий рівень захисту даних у системі.

1.3 Огляд існуючих рішень та конкурентів

Управління запасами є ключовим аспектом у діяльності роздрібних магазинів, зокрема таких, що реалізують канцелярські товари. Вчасне поповнення асортименту, контроль залишків, облік продажів і надходжень — усе це вимагає чіткої організації, яка в умовах ручного обліку часто виявляється неефективною. На ринку існує ряд програмних рішень, які дозволяють автоматизувати ці процеси, однак більшість з них або не враховують потреб малого бізнесу, або потребують значних фінансових та технічних ресурсів [8].

Microsoft Excel / Google Sheets часто використовуються як найпростіші засоби обліку. Вони дозволяють створювати таблиці, вести списки товарів, фіксувати надходження і витрати. Проте, ці інструменти не мають вбудованих механізмів автоматизації, не підтримують складну логіку, а також не забезпечують надійного контролю доступу чи мобільного використання. При великій кількості товарів зростає ризик помилок і дублювання даних (рис. 1.2).

Airtable — це популярна no-code платформа, яка поєднує простоту електронної таблиці з базовими функціями бази даних. Вона дозволяє будувати візуальні інтерфейси, створювати фільтри, пов'язувати таблиці, вести облік запасів у форматі канбан-дошок або календаря. Airtable має інтеграції з іншими сервісами, але обмежена в автоматизації та масштабуванні, а також вимагає платної підписки для розширених можливостей (рис. 1.3).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		Назва	USDA	Висота	Ширина	Доц	Хвороби	Аромат	Цвітіння						
2		Абрахам Дербі (A	5	120-150	100	1	2	3	2		A				
3		Аваланш (Avalan	6	80	50	2	2	1	2		Срезочные (Florists Rose)				
4		Алі Дорате (Ali D	6	70	90	2	3	1	2		прос	Ф			
5		Амаретто (Amare	5	250-300	100	1	3	2	2		К				
6		Ангела (Angela)	5	100-150	100	3	3	1	3						
7		Артемід (Artemis)	6	100-120	80	3	3	2	2		Ш				
8		Аскот (Ascot)	6	60-80	50	2	2	1	2						
9		Аспирин (Aspirin)	6	50-80	80	3	3	1	3		Ф				
10		Бельв'ю (Bellevue	6	80	50	2	3	1	2		ЧГ				
11		Блю фо Ю (Blue f	6	60-80	50	1	2	3	2		Ф				
12		Букаву (Bukavu)	6	120-150	150	3	3	1	2		пер	Гибрид Moschata/мускусные гибриды (Hybrid Musk)			
13		Бьенвеню (Bienve	6	250-300		2	2	3	2		К				
14		Вільям Морріс (W	6	120-150	90	2	3	2	2		A				
15		Вільям Шекспір (V	6	100-120	100	2	3	3	2		A				
16		Вінчестер Кафед	6	100-120	100		3	1	2		A				
17		Гертруда Джекіл	5	100-120	90	1	2	3	2		A				
18		Гирлянда де Амур	5	300-350	250		3	3	2		Гибрид Moschata/мускусные гибриды (Hybrid Musk)				

Рисунок 1.2 – Інтерфейс Microsoft Excel

	Name	Amount	Lead Type	Stage
1	FireBasket	\$65,000	Trade Show	Closed - Win
2	Zenprospr	\$50,000	Word of Mouth	Closed - Win
3	pear.ly	\$225,000	Trade Show	Analysis/Value Proposition
4	EnnWe	\$70,000	Word of Mouth	Closed - Win
5	Earworm	\$40,000	Word of Mouth	Closed - Lost
6	Handcrafted Mediocrity	\$10,000	Advertisement	Closed - Win
7	Objectively Edible	\$13,000	Word of Mouth	Closed - Win
8	Consens.us	\$100,000	Advertisement	Analysis/Value Proposition
9	EnnWe	\$80,000	Word of Mouth	Analysis/Value Proposition
10	Exploittr	\$25,000	Trade Show	Closed - Win
11	Fastidio.us	\$40,000	Advertisement	Closed - Win
12	FireBasket	\$22,000	Advertisement	Closed - Win

Рисунок 1.3 – Інтерфейс Airtable

Zoho Inventory — це потужне рішення для управління запасами та логістикою, що дозволяє контролювати запаси на складах, відстежувати замовлення та формувати звіти. Воно орієнтоване більше на середній бізнес, має потужний функціонал і розширені можливості автоматизації. Проте, складність у налаштуванні та англomовний інтерфейс можуть бути бар'єром для невеликих торгових точок (рис. 1.4).

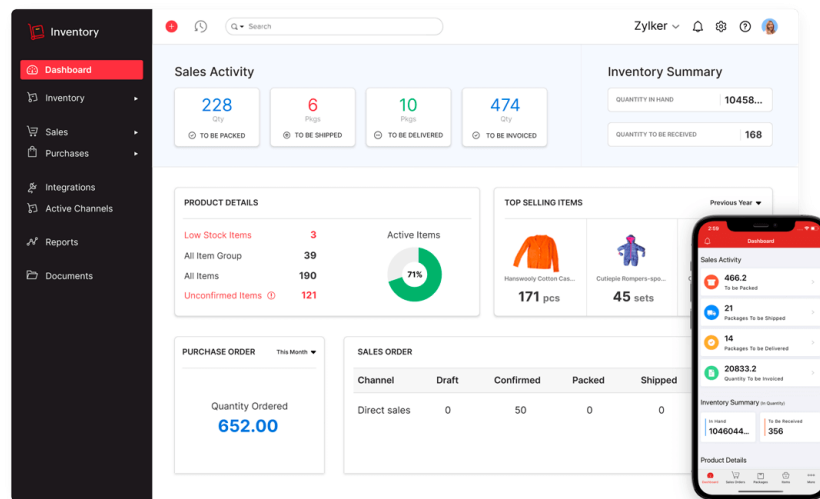


Рисунок 1.4 – Інтерфейс Zoho Inventory

Google AppSheet вирізняється на фоні інших платформ завдяки поєднанню простоти, гнучкості та повної безкодової моделі. Воно дозволяє створити додаток спеціально під потреби магазину — з урахуванням категорій товарів, фіксації залишків, надходжень і продажів, автоматичних повідомлень при низькому рівні запасів, зручною фільтрацією та сортуванням. AppSheet працює напряду з Google Sheets, підтримує мобільний доступ і має безкоштовний тарифний план для невеликих команд (табл. 1.2).

Таблиця 1.2 – Порівняння рішень для автоматизації управління запасами

Рішення	Простота використання	Можливість автоматизації	Мобільний доступ	Вартість
Excel / Sheets	Висока	Немає	Частково	Безкоштовно
Airtable	Висока	Часткова (обмежена)	Є (через додаток)	Від \$12/міс.
Zoho Inventory	Середня	Висока	Є	Від \$39/міс.
Google AppSheet	Висока	Висока (Bots, логіка)	Так (Android, iOS)	Від \$0

Таким чином, серед конкурентів Google AppSheet демонструє найкраще співвідношення гнучкості, функціональності та доступності саме для малих магазинів, які прагнуть ефективної, але простої автоматизації без зайвих витрат та складнощів у впровадженні.

1.4 Обґрунтування вибору середовища та технологій

Розробка системи управління запасами для магазину канцелярських товарів вимагає вибору такого інструменту, який би забезпечував просту реалізацію основних функцій — облік товарів, фіксація надходжень і списань, формування звітів, контроль залишків — без складної інфраструктури та витрат на програмування. У цьому контексті Google AppSheet є одним із найоптимальніших рішень. Вона дозволяє створювати мобільні та веб-застосунки без написання коду, базуючись на вже звичних для користувачів електронних таблицях Google Sheets. Це особливо зручно для малого бізнесу, де відсутні спеціалізовані ІТ-фахівці або бюджет на складне програмне забезпечення.

Однією з ключових переваг є швидкість розробки. За один-два дні можна створити повноцінну систему, яка враховує особливості конкретного магазину: категорії товарів, кількість на складі, дату останнього надходження, критичний рівень запасів. Завдяки вбудованим ботам (AppSheet Bots) можна реалізувати автоматичні сповіщення — наприклад, повідомлення адміністратору про те, що товар закінчується [9].

Візуальний інтерфейс AppSheet дозволяє налаштовувати вигляд додатка без технічної складності: таблиці, форми, списки, фільтри, панелі керування. Все це робиться через графічний редактор, що дає змогу адаптувати інтерфейс до різних ролей користувачів — продавця, адміністратора чи бухгалтера [10, 11].

Ще однією важливою перевагою є мобільність: працівники можуть працювати із системою зі смартфонів або планшетів, не будучи прив'язаними до

офісного ПК. Це зручно для роботи на складі, під час приймання товару або інвентаризації.

З боку безпеки та доступу, AppSheet дозволяє налаштувати права користувачів через Google-акаунти, обмежувати редагування для окремих таблиць або навіть окремих записів. Це важливо для уникнення несанкціонованого внесення змін у дані про товарні залишки.

У порівнянні з іншими платформами, такими як Zoho Inventory чи Airtable, AppSheet має перевагу в тому, що дозволяє збудувати рішення "під себе", без надлишкового функціоналу й без потреби платити за кожного користувача на старті. Особливо це важливо для магазинів з невеликим штатом працівників, де система має бути простою, ефективною і дешевою (табл. 1.3).

Таблиця 1.3 – Переваги Google AppSheet у розробці системи обліку запасів

Критерій	Оцінка AppSheet
Простота розробки	Повна відсутність програмування
Підтримка мобільного доступу	Android/iOS підтримуються "з коробки"
Візуальне налаштування	Гнучке, адаптивне, з шаблонами
Автоматизація процесів	Потужні Bots (сповіщення, обчислення)
Вартість впровадження	Є безкоштовний план для малого бізнесу
Права доступу	Доступ через Google-акаунти та ролі

Таким чином, вибір Google AppSheet є цілком обґрунтованим для реалізації системи управління запасами в магазині канцелярії. Це середовище дозволяє з мінімальними витратами створити повноцінне рішення, що покриває всі ключові потреби бізнесу та легко адаптується під конкретні умови.

1.5 Вартісна ефективність та оцінка low-code підходу

Однією з ключових переваг розробки системи управління запасами за допомогою Google AppSheet стала її вартісна ефективність. У порівнянні з традиційною розробкою програмного забезпечення на замовлення або з нуля, використання low-code платформи дозволяє значно зменшити витрати на оплату праці розробників, технічну підтримку та інфраструктуру (табл. 1.4).

Таблиця 1.4 – Порівняння витрат традиційної розробки та Google AppSheet

Параметр	Традиційна розробка	Розробка з Google AppSheet
Орієнтовний час розробки	120–160 годин	30–40 годин
Середня ставка розробника	15–20 \$/год	Не застосовується
Загальна вартість	1800–3200 \$	0–50 \$ (або за тарифом AppSheet)
Потреба у програмуванні	Висока	Мінімальна / відсутня
Час на впровадження	4–6 тижнів	до 2 тижнів
Потреба у хостингу та підтримці	Так	Ні (використовується Google Cloud)

За орієнтовними оцінками, створення аналогічного функціоналу засобами повноцінної веб-розробки (з бекендом, фронтом та базою даних) зайняло б не менше ніж 120–160 годин роботи фахівця рівня middle. При середній ринковій вартості 15–20 \$/год, загальна сума могла б скласти від 1800 до 3200 \$. Натомість розробка рішення на базі AppSheet зайняла приблизно 30–40 годин, при цьому більшість дій виконувалася без знання коду, а плата за використання сервісу

залишається у межах безкоштовного плану або мінімального платного тарифу (до 10 \$/користувача/місяць у разі масштабування).

Крім економії фінансів, було досягнуто й значної економії часу — впровадження системи від ідеї до повністю готового рішення зайняло менше ніж два тижні, включаючи налаштування шаблону, адаптацію структури, розробку логіки, тестування та навчання персоналу. Це робить low-code платформу Google AppSheet особливо привабливою для малого бізнесу та швидкого прототипування управлінських систем.

1.6 Безпека, контроль доступу та перспективи масштабування системи

Забезпечення безпеки даних і контроль доступу є критично важливими аспектами при розробці будь-якої інформаційної системи, особливо якщо вона пов'язана з обліком матеріальних запасів. У середовищі Google AppSheet ці функції реалізуються через вбудовані механізми і тісну інтеграцію з сервісами Google Workspace. Авторизація користувачів здійснюється через облікові записи Google, що забезпечує надійний рівень автентифікації без потреби у власному модулі реєстрації та входу. Крім того, кожному користувачеві може бути надано різний рівень доступу до функціональності додатку за допомогою ролей, умов видимості елементів інтерфейсу та фільтрації даних на основі виразів ('USEREMAIL()' та 'USERROLE()').

Для збереження конфіденційної інформації використовується Google Sheets, що зберігається у хмарі Google Drive з усіма відповідними політиками безпеки. Увесь доступ до таблиць можна обмежити правами редагування або лише для читання. Це дозволяє розділити доступ між користувачами, наприклад, надавши керівникові можливість бачити аналітичні дані без права їх змінювати. В AppSheet також передбачено ведення логів змін — фіксації дій користувачів, записів у

таблицях, оновлень або видалення, що дозволяє здійснювати базовий аудит і відслідковувати активність у системі.

Крім цього, важливою перевагою є автоматичне резервне копіювання: всі зміни в таблицях Google Sheets зберігаються в історії версій, що дозволяє відновити дані у випадку помилки або втрати. Це надає додатковий рівень надійності в умовах роботи з великою кількістю даних.

Окремо варто відзначити можливості масштабування AppSheet-рішення. Оскільки застосунок базується на хмарній архітектурі Google, він може обробляти зростаючі обсяги даних без необхідності зміни інфраструктури. У разі потреби додаткові таблиці, поля або зв'язки можуть бути легко додані в Google Sheets і підключені до AppSheet без перезапуску системи. Також є потенціал для інтеграції з іншими сервісами Google, такими як Google Calendar, Gmail, Google Maps або зовнішні API (через Webhooks, AppScript або Make.com), що дозволяє адаптувати систему під більш складні бізнес-процеси рис. (1.5).

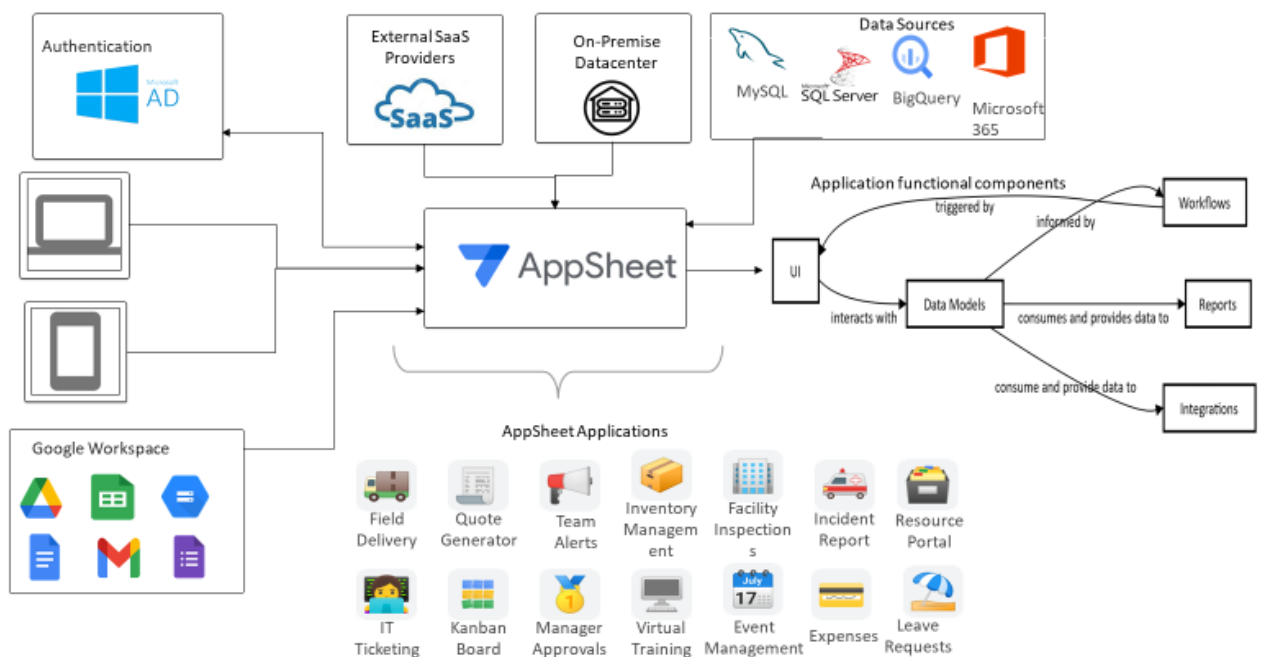


Рисунок 1.5 – Інтерфейс Zoho Inventory

Крім технічної гнучкості, AppSheet дозволяє працювати в багатокористувацькому режимі з одночасним доступом кількох працівників, без

блокування ресурсів і конфліктів у даних. Завдяки цьому розроблене рішення не тільки безпечне та стабільне, але й придатне для масштабування на більшу кількість користувачів, торгових точок або розширення функціоналу в майбутньому.

2. ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ

2.1 Визначення вимог системи

2.1.1 Функціональні вимоги

Система управління запасами має забезпечувати підтримку основних бізнес-процесів магазину канцелярських товарів: облік товарів, фіксацію надходжень і продажів, моніторинг залишків, формування звітів і сповіщень. Оскільки рішення створюється за допомогою Google AppSheet, особлива увага приділяється зручності інтерфейсу, простоті введення даних, мобільному доступу та можливості автоматизації [12].

Основні функціональні вимоги включають:

- 1) Облік товарів. Система повинна дозволяти зберігати повний перелік товарів магазину із зазначенням назви, унікального коду, категорії, ціни, кількості на залишку, постачальника та мінімального допустимого рівня запасів.
- 2) Надходження продукції. Користувач повинен мати можливість зареєструвати нову партію товарів із фіксацією дати, кількості та постачальника. При цьому кількість відповідного товару на складі автоматично збільшується.
- 3) Реєстрація продажів. У системі реалізовано функцію для внесення інформації про продаж товарів: дату, найменування, кількість та касира. При кожному продажі залишок товару відповідно зменшується.
- 4) Контроль залишків. Користувачі повинні мати доступ до перегляду залишків товарів у розрізі категорій, з можливістю фільтрації за назвою або кількістю. При досягненні критичного рівня запасів система автоматично генерує сповіщення.
- 5) Автоматичні сповіщення. Використовуючи функціонал AppSheet Automation, система надсилає повідомлення (на email або в інтерфейсі) про досягнення мінімального залишку або необхідність замовлення нового товару.

6) Журнал операцій. Усі дії щодо додавання товару, надходжень або продажів фіксуються в таблиці журналу для подальшої перевірки або формування звітів.

7) Розмежування доступу. Адміністратор має доступ до всіх функцій додатку, у той час як продавці можуть лише переглядати список товарів, оформлювати надходження й продажі, не змінюючи структуру системи.

8) Звіти. Система повинна формувати базові звіти: залишки на складі, загальна кількість реалізованого товару за період, аналіз надходжень, критичні позиції.

9) Мобільна сумісність. Усі функції системи повинні бути доступні як у веб-версії, так і через мобільні пристрої на базі Android та iOS.

10) Пошук і фільтрація. Інтерфейс повинен забезпечувати швидкий пошук товару за назвою, категорією або іншим параметром.

11) Історія змін. Для кожного товару система повинна зберігати історію змін — коли саме і ким була змінена кількість, додано нову партію чи зафіксовано продаж. Це дозволяє відстежувати всі операції з конкретним товаром.

12) QR-коди або штрих-коди. Для прискорення пошуку товарів у системі повинна бути передбачена можливість сканування штрих-кодів або QR-кодів через мобільний застосунок. У AppSheet це реалізується через сканування камери з подальшим фільтруванням записів.

13) Можливість резервування товарів. У разі потреби система може бути розширена функцією резервування — коли товар ще не проданий, але вже «заброньований» для клієнта, і тому не враховується в доступному залишку.

14) Агреговані показники. На головній сторінці або в окремому звіті бажано відображати ключові метрики: загальна кількість позицій, загальний залишок, кількість критичних товарів, сума продажів за день/тиждень.

15) Експорт звітів. Користувач повинен мати можливість експортувати звіти про продажі, залишки або надходження у форматі PDF або Excel. У AppSheet це реалізується через шаблони Google Docs та функцію "Workflow Email".

Ці функціональні вимоги є основою для подальшого проєктування моделі предметної області, інтерфейсів і логіки додатку в AppSheet. Вони покривають типові задачі магазину канцелярії й адаптовані до можливостей no-code середовища, що дозволяє реалізувати їх без програмування.

2.1.2 Нефункціональні вимоги

Окрім функціональних можливостей, система управління запасами для магазину канцелярських товарів повинна відповідати ряду нефункціональних вимог, які забезпечують її ефективність, стабільність і зручність у використанні. Ці вимоги є критично важливими для щоденної роботи магазину та довготривалої підтримки системи [13].

1) Надійність. Система повинна працювати стабільно незалежно від навантаження, коректно обробляти типові помилки введення даних і не втрачати інформацію при втраті інтернет-з'єднання. Завдяки збереженню даних у Google Sheets, забезпечується хмарне резервування змін у режимі реального часу.

2) Доступність. Користувачі повинні мати змогу працювати з додатком 24/7 з будь-якого пристрою, який має доступ до Інтернету. Платформа AppSheet забезпечує доступ як з комп'ютерів, так і зі смартфонів та планшетів, не потребуючи локальної інсталяції.

3) Безпека. Доступ до системи має бути захищений через авторизацію користувачів за допомогою облікових записів Google. Крім того, система повинна підтримувати розмежування прав доступу — наприклад, щоб продавець не мав змоги змінювати довідник товарів або переглядати звіти адміністратора.

4) Масштабованість. Система повинна легко адаптуватися до розширення — додавання нових категорій товарів, нових користувачів або переходу від Google Sheets до повноцінної бази даних (наприклад, MySQL чи PostgreSQL), не змінюючи основної логіки.

5) Продуктивність. Основні операції (відображення залишків, фільтрація, додавання записів) повинні виконуватись протягом кількох секунд навіть при великій кількості товарних позицій.

6) Юзабіліті (зручність інтерфейсу). Інтерфейс повинен бути інтуїтивно зрозумілим для працівників без спеціальної технічної підготовки. Усі основні дії мають бути доступні через прості кнопки, форми і випадаючі списки.

7) Портативність. Система не прив'язана до конкретної операційної системи чи пристрою, тому її можна використовувати на будь-якому гаджеті з браузером або додатком AppSheet.

8) Можливість резервного копіювання. Завдяки інтеграції з Google Drive, автоматичне збереження змін у Google Sheets забезпечує регулярне резервування всіх даних без додаткових налаштувань.

2.1.3 Варіанти використання

Розроблена система управління запасами орієнтована на малий роздрібний бізнес і призначена для підвищення прозорості, зручності та ефективності контролю товарного обігу в магазині канцелярських товарів. Бізнес-модель передбачає впровадження мобільного та веб-застосунку, який може бути використаний без необхідності програмування або закупівлі спеціалізованого обладнання [14].

Ключові зацікавлені сторони (учасники бізнес-процесів):

— Адміністратор магазину — створює/редагує товари, переглядає звіти, формує замовлення на постачання.

— Комірник / працівник складу — реєструє надходження товарів, виконує інвентаризацію.

— Продавець — здійснює реєстрацію продажів, переглядає залишки на полиці або на складі.

— Керівник бізнесу — аналізує обсяги продажів, динаміку товарообігу, приймає управлінські рішення.

Варіанти використання системи (Use Case Scenarios) (рис. 2.1):

1) Додавання нового товару до системи

— Актор: Адміністратор

— Опис: Через форму в AppSheet адміністратор створює нову товарну позицію, вказує назву, категорію, початкову кількість, ціну, мінімальний рівень залишку та фото.

— Результат: Новий товар з'являється в системі й доступний для подальших операцій.

2) Реєстрація надходження товару

— Актор: Комірник

— Опис: У день отримання партії товарів, працівник додає нові надходження, вказуючи товар, кількість, постачальника та дату.

— Результат: Поточний залишок товару автоматично збільшується; зберігається запис у журналі.

3) Продаж товару

— Актор: Продавець

— Опис: Після продажу товару продавець через мобільний інтерфейс фіксує назву товару, кількість і дату.

— Результат: Залишок автоматично зменшується, створюється запис у таблиці продажів.

4) Контроль залишків і сповіщення

— Актор: Система (автоматично)

— Опис: Якщо залишок певного товару падає нижче мінімального рівня, система автоматично надсилає email або push-сповіщення адміністратору.

— Результат: Вчасне реагування, формування нового замовлення.

5) Генерація звіту про продажі

— Актор: Керівник

— Опис: За запитом формується зведений звіт за день/тиждень/місяць із деталями продажів.

— Результат: Прийняття рішень про поповнення, сезонне планування, знижки тощо.

Вигоди від використання системи:

- Зменшення помилок при ручному обліку.
- Швидкий доступ до залишків товарів у реальному часі.
- Автоматичне сповіщення про нестачу товарів.
- Стандартизація операцій продажу та надходжень.
- Спрощення контролю за діяльністю працівників магазину.



Рисунок 2.1 – Основні варіанти використання

На діаграмі варіантів використання відображено основні сценарії взаємодії користувачів із системою управління запасами канцелярських товарів. Серед основних акторів системи — адміністратор, комірник, продавець та керівник. Кожен із них виконує специфічні дії відповідно до своїх повноважень: адміністратор керує даними про товари, категоріями, ролями користувачів і звітами; комірник фіксує надходження, контролює залишки та отримує автоматичні сповіщення; продавець виконує операції з продажу та може сканувати QR-коди товарів; керівник аналізує звіти про продажі та залишки. Така модель забезпечує чітке розмежування функціоналу між користувачами й підвищує ефективність управління товарними запасами [15].

Сценарій «Продаж товару» передбачає фіксацію факту реалізації товару клієнту через інтерфейс AppSheet. Продавець обирає потрібну позицію зі списку, вказує кількість, дату та підтверджує продаж. Система автоматично зменшує залишок на складі, записує дію до журналу, а також перевіряє, чи не досягнуто мінімального рівня запасів — у такому разі адміністратор отримує відповідне сповіщення (рис. 2.2).

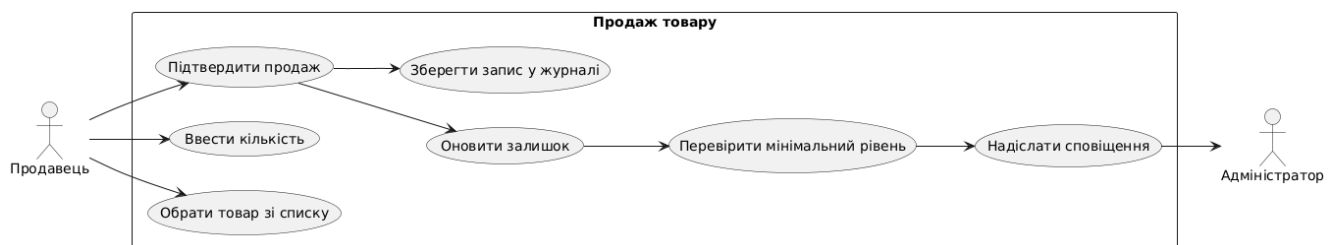


Рисунок 2.2 – Сценарій «Продаж товару»

Пояснення до діаграми:

- Продавець починає з вибору товару та введення кількості.
- Після підтвердження продажу:
 - а) залишок оновлюється;
 - б) дію записано у журналі змін;
 - в) система перевіряє критичність залишку;
 - г) якщо товару мало — адміністратор отримує сповіщення.

«Зареєструвати надходження товару» - цей сценарій реалізується комірником або адміністратором у момент отримання нової партії товарів. Користувач вводить інформацію про постачальника, обирає товар із довідника, зазначає кількість, дату та підтверджує операцію. Після цього система оновлює залишок товару, створює відповідний запис у журналі змін і (за потреби) повідомляє адміністратора про появу нового товару або зміну цін (рис. 2.3)[16].



Рисунок 2.3 – Сценарій «Зареєструвати надходження товару»

Пояснення до діаграми:

Комірник:

- обирає товар;
- вводить дату, кількість і постачальника;
- підтверджує дію.

Система:

- оновлює залишок;
- додає запис до журналу;
- перевіряє, чи існує товар у системі;
- якщо товар новий — створює запис і повідомляє адміністратора.

Сценарій «Сформувати звіт про продажі» дозволяє керівнику або адміністратору отримати зведену інформацію про реалізовані товари за певний період. Користувач вибирає діапазон дат, фільтрує за категоріями або товарами, після чого система формує таблицю з кількістю продажів, виручкою, а також

статистикою по кожному товару. Звіт може бути збережений або експортований. Нижче наведено графічне представлення даного варіанту використання (рис 2.4).

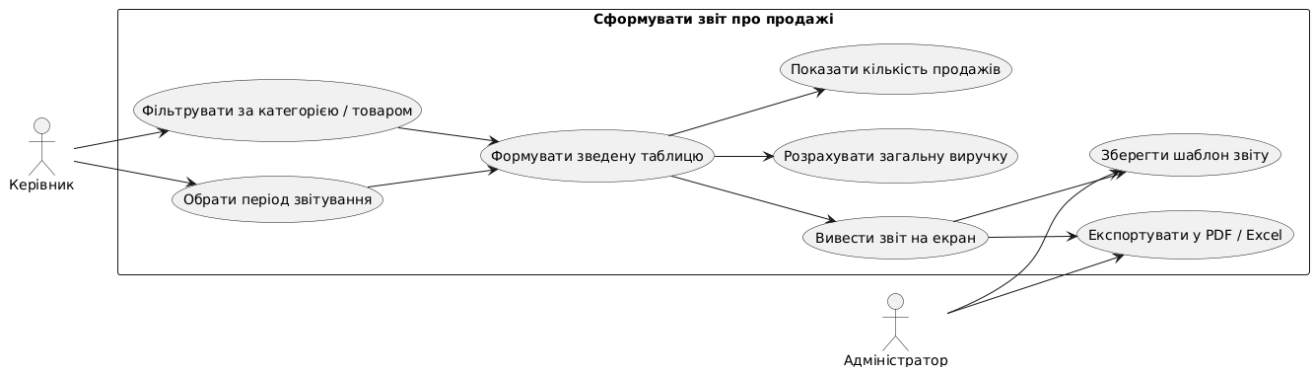


Рисунок 2.4 – Сценарій «Сформувати звіт про продажі»

Пояснення до діаграми:

Керівник:

- задає діапазон дат;
- фільтрує товари або категорії;
- переглядає звіт на екрані;
- експортує або зберігає шаблон.

Адміністратор:

- має доступ до експорту звітів і шаблонів.

2.1 Модель системи

Розроблена система управління запасами побудована з використанням хмарної low-code платформи Google AppSheet, яка виступає як візуальний інтерфейс для взаємодії з даними, збереженими в Google Sheets. Така архітектура дозволяє створити повноцінний застосунок без необхідності програмування, а також забезпечити швидкий доступ до функціональності з будь-якого пристрою, підключеного до Інтернету [17].

Оснoву системи становлять Google Таблиці, які слугують як база даних. У них зберігається уся необхідна інформація про товари, запаси, продажі, надходження, постачальників та журнали подій. Ці таблиці зв'язані між собою через ключові поля (наприклад, назву товару, ідентифікатор операції) та мають логічну структуру, придатну до фільтрації, агрегації та обробки через AppSheet.

Google AppSheet виступає як проміжна логіка та користувацький інтерфейс, який автоматично синхронізується з цими таблицями. У застосунку реалізовано форми введення, таблиці перегляду даних, фільтри, кнопки дій, сповіщення, автоматичні обчислення та правила. Усі зміни, які вносить користувач через застосунок, одразу відображаються в таблицях, і навпаки — оновлення в таблицях миттєво відображаються в інтерфейсі [18].

Доступ до системи можливий із будь-якого сучасного пристрою — смартфона, планшета або комп'ютера, що має браузер або встановлений мобільний додаток AppSheet. Така мультиплатформеність дозволяє працівникам магазину взаємодіяти із системою як безпосередньо в торговому залі (через смартфон чи планшет), так і з офісу або вдома (через вебінтерфейс).

- Загальна модель системи включає такі компоненти:
- Хмарне сховище Google Sheets, де зберігаються всі дані;
- AppSheet-застосунок, який працює як візуальний і логічний рівень доступу;
- Користувацькі пристрої, які підключаються до системи: телефони (Android/iOS), планшети, ноутбуки, ПК.

Зв'язок між компонентами реалізується через інтеграційні можливості AppSheet, який безпосередньо зчитує, записує та оновлює дані в Google Таблицях. Завдяки цьому забезпечується реальний час оновлення даних, підтримка одночасного доступу кількох користувачів і надійна синхронізація без встановлення серверної частини.



Рисунок 2.5 – Модель предметної області

2.2 Модель даних предметної області

Модель предметної області охоплює всі ключові об'єкти, які беруть участь у процесі обліку товарів, руху запасів, керування категоріями, операційною діяльністю працівників магазину, генерацією звітності та контролем повноважень користувачів. Ця модель є логічною основою для побудови структури таблиць у Google Sheets та взаємодії компонентів у AppSheet (рис 2.6).

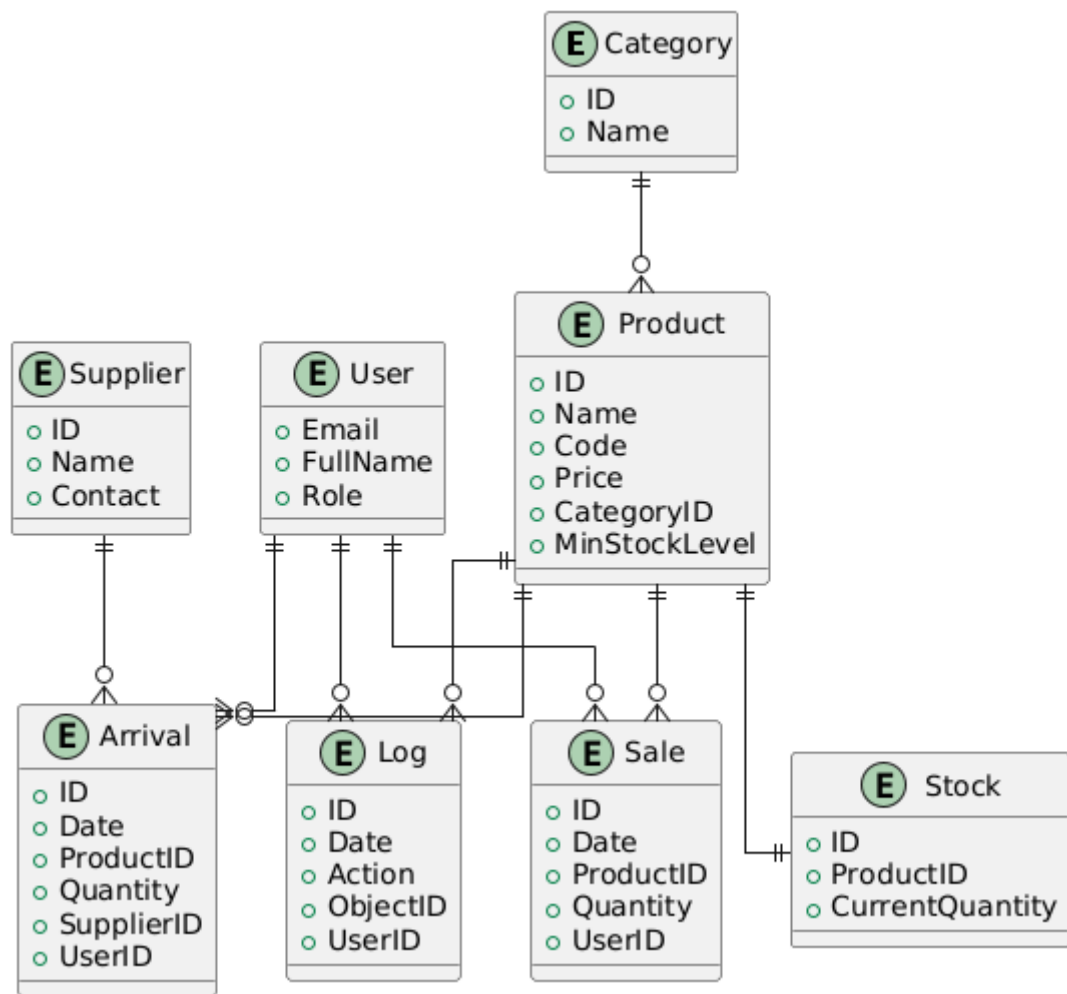


Рисунок 2.6 – ER діаграма

Основні сутності:

1) Товар (Product)

— Атрибути: ID, Назва, Код, Категорія, Ціна, Одиниця виміру, Мінімальний залишок, Фото, Опис.

— Функція: виступає головною одиницею обліку, що продається чи надходить.

2) Категорія товару (Category)

— Атрибути: ID, Назва категорії, Опис.

— Функція: групує товари за призначенням (папір, письмове приладдя, архівування тощо).

3) Надходження (Arrival)

— Атрибути: ID, Дата, Товар, Кількість, Постачальник, Ціна закупки, Користувач.

— Функція: фіксує факт поповнення запасів.

4) Продаж (Sale)

— Атрибути: ID, Дата, Товар, Кількість, Ціна продажу, Касир, Коментар.

— Функція: реєструє реалізацію продукції.

5) Залишок (Stock)

— Атрибути: ID, Товар, Поточна кількість, Критичний рівень, Розрахункова кількість.

— Функція: дає змогу контролювати доступність продукції.

6) Постачальник (Supplier)

— Атрибути: ID, Назва, Контакти, Адреса, Примітка.

— Функція: дозволяє враховувати, хто поставляє товари, для аналітики.

7) Користувач (User)

— Атрибути: Email, ПІБ, Роль, Дата додавання, Активність.

— Функція: визначає доступ до функціоналу системи.

8) Операція (Log / AuditTrail)

— Атрибути: Дата, Тип дії, ID товару, Зміна, Хто виконав.

— Функція: прозора історія змін у системі.

Типи ролей користувачів:

— Адміністратор: повний доступ до даних, налаштувань, користувачів і звітів.

— Продавець: доступ до додавання продажів і перегляду залишків.

— Комірник: додає надходження, формує звіти, контролює запаси.

Додаткові логічні зв'язки:

— Supplier (1) → (∞) Arrival

— Product (1) → (∞) Log

— User (1) → (∞) Sale / Arrival / Log

— Product (1) → (1) Stock

— Category (1) $\rightarrow$ (∞) Product

Додаткові можливості розвитку моделі:

- Впровадження таблиці резервів (резервування товару клієнтом).
- Додавання механізму списання зіпсованого чи простроченого товару.
- Облік знижок, акційних цін, коментарів до продажів.
- Формування щомісячної статистики або плану закупівель.

2.3 Взаємодія системи

Однією з важливих складових при розробці інформаційної системи є моделювання взаємодії між користувачами та внутрішніми компонентами системи. Це дозволяє чітко визначити логіку обробки запитів, послідовність дій та відповідальність кожного учасника процесу. У межах даної роботи для демонстрації поведінки системи в динаміці використано діаграми послідовності (sequence diagrams), які ілюструють обмін повідомленнями між об'єктами під час виконання конкретних сценаріїв.

На основі функціональних вимог до системи було змодельовано три основні сценарії взаємодії: реєстрація надходження товару, здійснення продажу, а також формування звіту про продажі. Кожна з діаграм відображає послідовність викликів між користувачем, мобільним застосунком AppSheet, базою даних Google Sheets та логічними правилами, реалізованими у вигляді автоматичних дій та умов.

На діаграмі послідовності для сценарію "Продаж товару" зображено, як продавець через мобільний інтерфейс обирає товар, вводить кількість, після чого система оновлює залишок у таблиці, створює відповідний запис у журналі змін та ініціює перевірку на мінімальний рівень. Якщо залишок менший за критичне значення — автоматично створюється сповіщення для адміністратора (рис. 2.7).

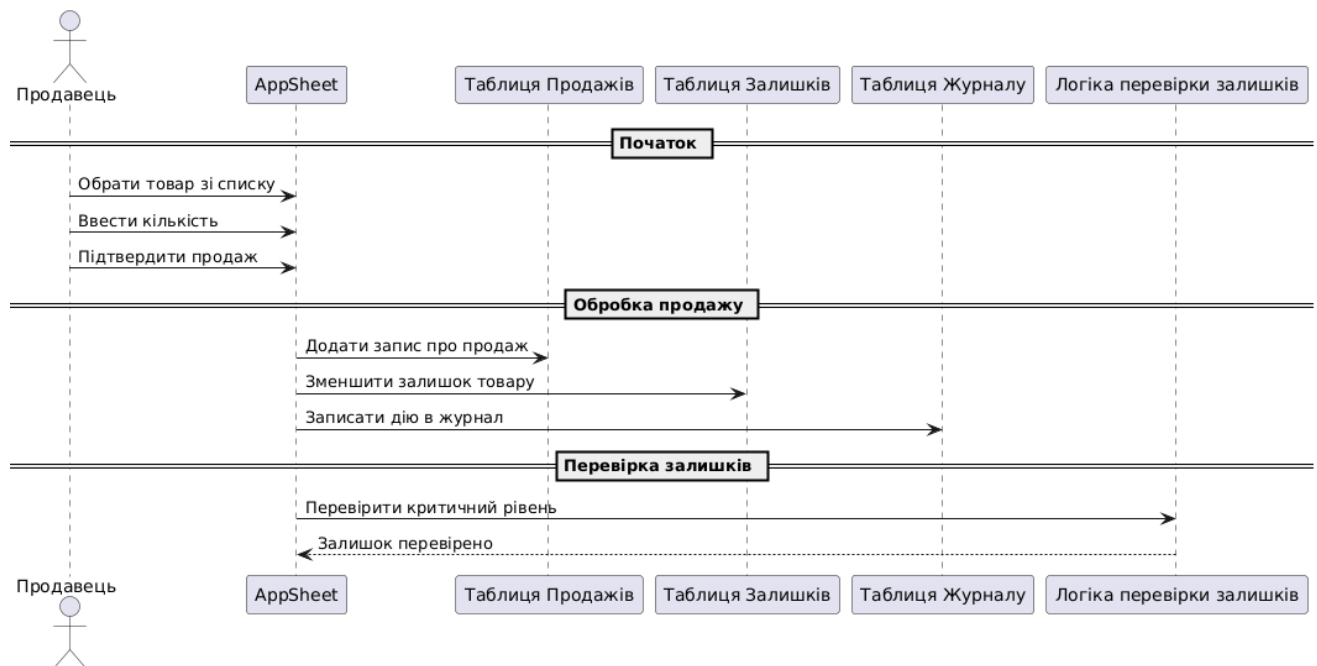


Рисунок 2.7 – Діаграма послідовності для сценарію "Продаж товару"

На діаграмі послідовно показано, як:

- Продавець вводить дані через AppSheet;
- Система оновлює таблиці продажів і залишків;
- Записує дію у журнал;
- Якщо товар закінчується — адміністратор отримує сповіщення.

Інша діаграма демонструє сценарій "Надходження товару", де комірник вводить дані про нову поставку, обираючи постачальника та товар із довідника. Система збільшує залишок відповідного товару, фіксує дату надходження, записує подію до журналу й, у разі потреби, створює новий запис для товару, якщо такий ще не існує (рис. 2.8).

На діаграмі послідовно показано, як:

Продавець виконує три дії: обирає товар, вводить кількість, підтверджує продаж.

- AppSheet оновлює таблицю продажів, зменшує залишки, фіксує дію в журналі.
- Окремо виконується перевірка залишку — без надсилання сповіщення.
- Третя діаграма показує процес

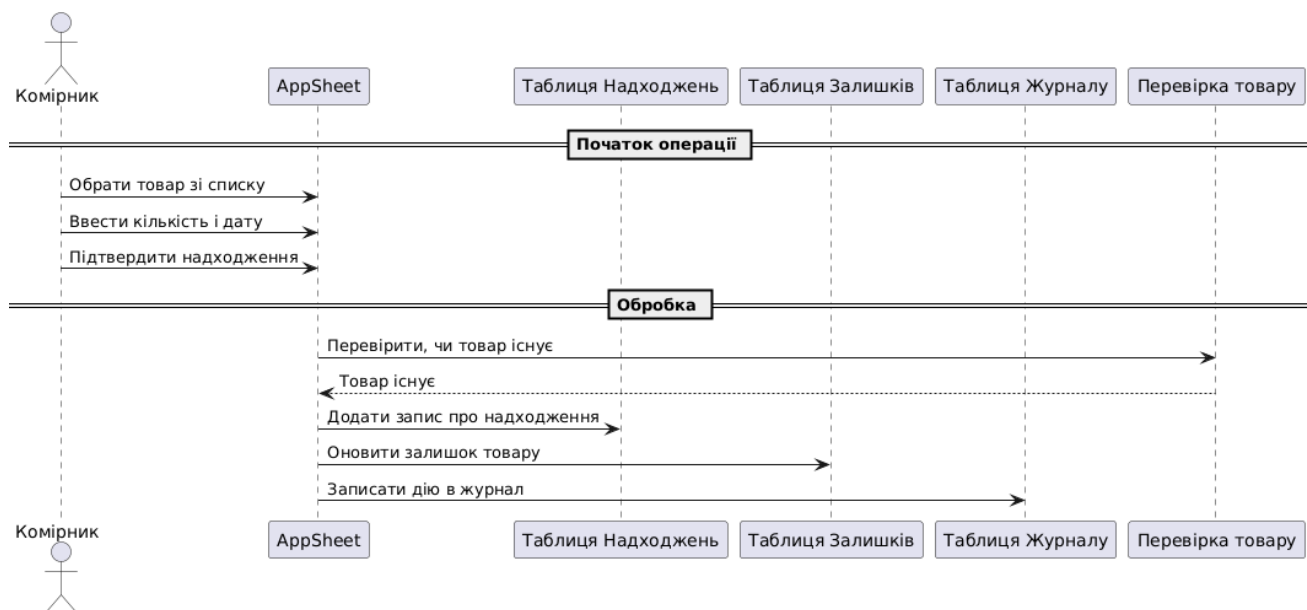


Рисунок 2.8 – Діаграма послідовності для сценарію "Продаж товару"

"Формування звіту про продажі", ініційований керівником. У цьому сценарії користувач обирає звітний період, фільтрує дані за категоріями або товарами, а система формує зведену таблицю продажів, розраховує загальну виручку, виводить її на екран і надає можливість експорту в зовнішній формат (PDF/Excel) (рис. 2.9).

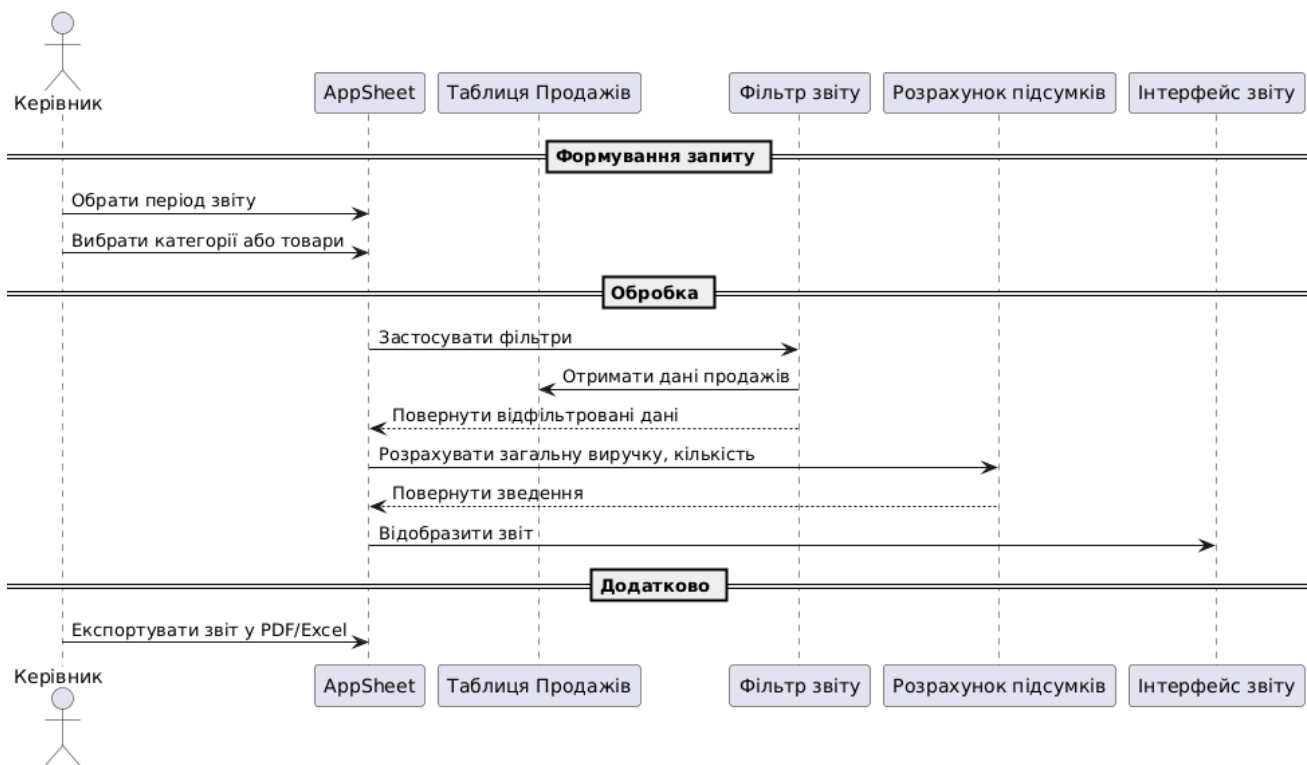


Рисунок 2.9 – Діаграма послідовності для сценарію "Продаж товару"

Сценарій охоплює:

- Дії керівника при виборі параметрів звіту.
- Застосування фільтрації до таблиці продажів.
- Розрахунок зведених показників (сума, кількість).
- Виведення звіту на екран та опційний експорт.

Усі діаграми послідовності створені відповідно до стандартів UML і забезпечують наочне подання логіки роботи системи. Вони відіграють важливу роль у документуванні рішень, підвищують прозорість архітектури та сприяють ефективному супроводу і розвитку розробленого програмного комплексу.

2.4 Реалізація системи

2.4.1 Використання шаблону AppSheet Inventory Management

Для оптимізації процесу створення системи управління запасами було проаналізовано декілька стандартних шаблонів, доступних на платформі Google AppSheet. AppSheet надає набір готових застосунків, які можуть бути використані як стартова точка для власних проєктів. Ці шаблони дозволяють швидко отримати доступ до перевірених структур даних, форм взаємодії, логіки обробки записів, а також прикладів автоматизації.

Серед аналізованих рішень були такі шаблони, як:

- Inventory Management — базовий шаблон для обліку товарів і залишків;
- Field Inventory App — застосунок для мобільного обліку запасів у полі з підтримкою сканування;
- Order Capture App — шаблон для введення замовлень клієнтів із динамічною таблицею;
- Warehouse Management — система для складу з надходженнями та розподілом товарів;

— Equipment Tracking App — приклад системи з базовими звітами, статусами й журналом змін.

Після аналізу функціональності, структури даних та інтерфейсу було обрано шаблон Inventory Management як базовий. Цей шаблон надає початкову модель взаємопов'язаних таблиць — «Items», «Stock», «Transactions» — а також базові дії для додавання, редагування та списання товарів. Його використання дозволило пришвидшити створення власної системи завдяки вже реалізованій логіці взаємодії між таблицями, наявності форм для введення даних, шаблонів повідомлень та прикладів дашбордів (рис. 2.10).



Рисунок 2.10 – Шаблон Inventory Management

У процесі розробки шаблон було суттєво адаптовано під потреби канцелярського магазину: розширено модель даних (додано таблиці «Постачальники», «Надходження», «Журнал»), реалізовано нові ролі користувачів (продавець, комірник, керівник), налаштовано додаткові правила перевірки залишків, а також автоматизовано формування звітів. Таким чином, шаблон відіграв роль архітектурної основи, на якій було реалізовано повноцінну систему, адаптовану до конкретної предметної області (табл. 2.1).

Таблиця 2.1 – Характеристики Inventory Management в порівняння з вимогами проєкту

Характеристика	Inventory Management	Система для магазину канцелярії
Тип товарів	Загальні товари	Канцелярські товари (з деталізацією категорій)
Таблиці	Items, Stock, Transactions	Товари, Запаси, Продажі, Надходження, Постачальники, Журнал
Ролі користувачів	Один користувач або універсальний	Продавець, Комірник, Керівник/Адміністратор
Облік продажів	Відсутній	Реалізовано з фіксацією дати, кількості, товару
Облік надходжень	Частково через загальну транзакцію	Виділена таблиця «Надходження» з деталізацією
Повідомлення про залишки	Відсутні	Автоматичне сповіщення при критичному рівні запасу
Звіти та дашборди	Базова статистика	Агреговані звіти за періодами, експорт у Excel/PDF
Мобільний доступ	Є	Є
Автоматизація (Bots)	Обмежена (в прикладі)	Розширена логіка: оновлення залишків, логування, сповіщення
QR-коди / Пошук товарів	Не реалізовано	Пошук за назвою, підтримка сканування (за потреби)

2.4.2 Адаптація структури таблиць до предметної області

Після вибору шаблону AppSheet Inventory Management як стартової основи, було проведено адаптацію структури таблиць для точного відображення предметної області — управління запасами в магазині канцелярських товарів. Базова структура шаблону включала лише три основні таблиці: Items, Stock та Transactions, чого було недостатньо для забезпечення повного циклу обліку товару, включаючи надходження, продажі, постачальників та журналізацію змін.

Для цього було внесено наступні зміни та доповнення:

Перейменовано та локалізовано назви таблиць відповідно до термінології предметної області:

- Items → Товари – список товарів із назвами, категоріями, цінами.
- Transactions → Продажі – реєстрація операцій продажу з датою, кількістю, користувачем.
- Stock → Запаси – актуальні залишки товарів на складі.

Додано нові таблиці:

- Надходження – таблиця для фіксації отриманих партій товару з датою, кількістю та посиланням на постачальника.
- Постачальники – довідник контрагентів з контактною інформацією та зв'язком із таблицею «Надходження».
- Журнал – автоматична таблиця, куди зберігаються всі зміни у запасах (дата, дія, відповідальний користувач).

Оновлено зв'язки між таблицями:

- Установлено референції між таблицями для забезпечення узгодженості даних (наприклад, продаж → товар, надходження → постачальник).
- Впроваджено декілька типів залежностей, що дозволяють реалізувати автоматичне оновлення залишків та обчислення загальної кількості товарів.

Оптимізовано структуру полів:

— Додано поля, необхідні для реалізації бізнес-логіки: статус товару, мінімальний залишок, відповідальний працівник, тип транзакції тощо.

— Налаштовано формули обчислення та валідаційні правила безпосередньо в Google Sheets та AppSheet.

Таким чином, адаптована модель даних дозволила забезпечити точне відображення внутрішніх процесів магазину, розмежувати функції користувачів та автоматизувати облік товарів на кожному етапі — від надходження до продажу й звітності (рис. 2.11).

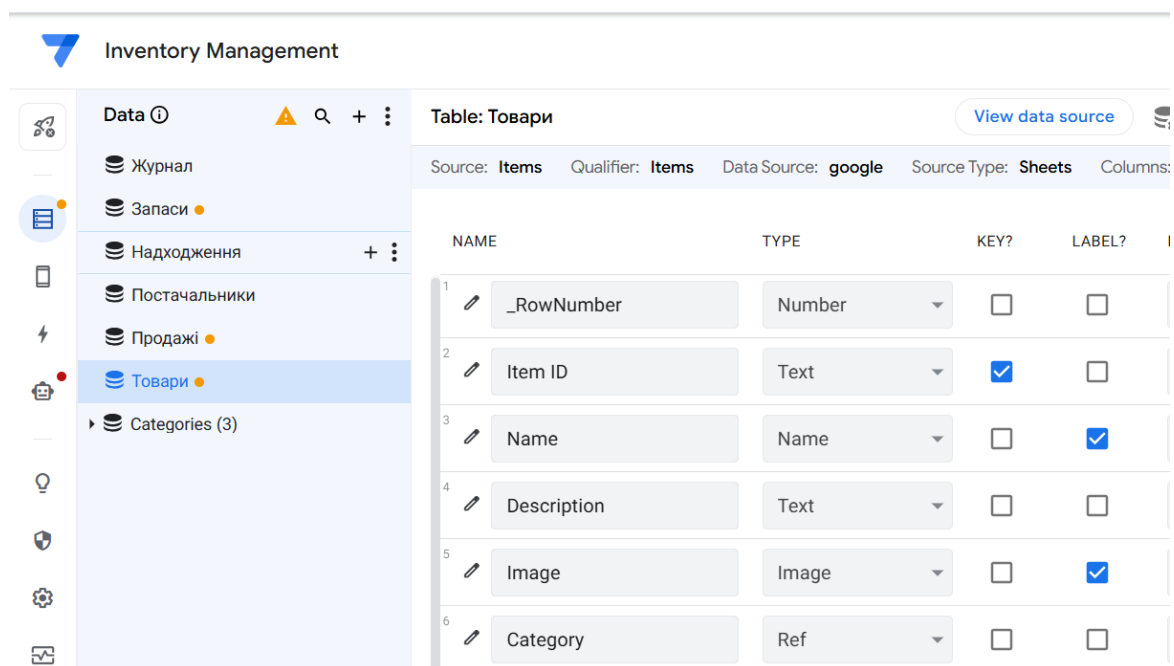


Рисунок 2.11 – Робота з таблицями даних

Ці зміни стали базовими для подальшої реалізації бізнес-логіки системи, збереження цілісності даних та зручної візуалізації інформації через інтерфейс AppSheet.

2.4.3 Розробка логіки обліку залишків

Облік товарних залишків є критично важливою функцією системи управління запасами, адже саме вона забезпечує контроль доступної кількості продукції на складі, оперативну реакцію на зміну стану запасів та формування звітності для керівництва. У розробленій системі ця логіка була реалізована на базі функціональності Google AppSheet із використанням таблиць Google Sheets та вбудованих дій, виразів (expressions) і автоматизованих сценаріїв (bots) (рис. 2.12).

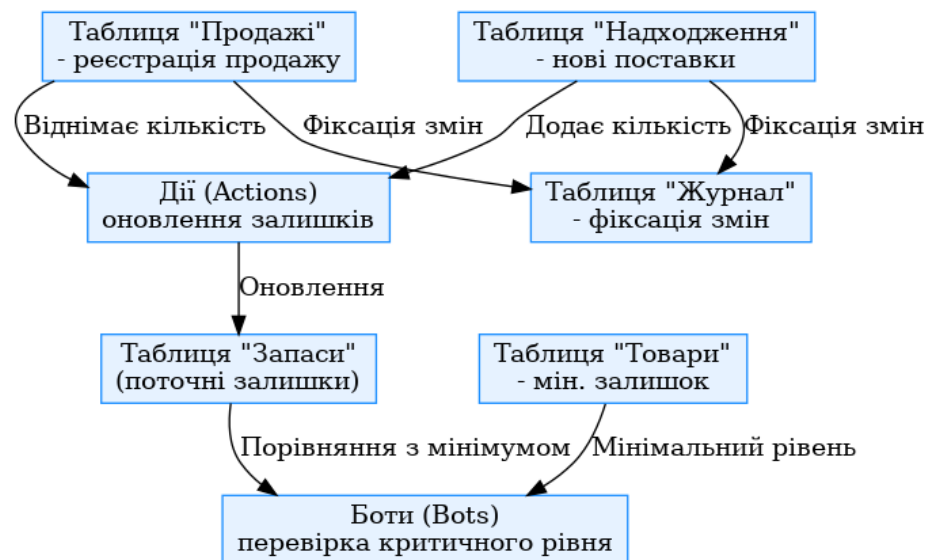


Рисунок 2.12 – Логіка роботи таблиць

Основні принципи побудови логіки обліку:

— Централізована таблиця "Запаси" містить поточну кількість кожного товару в наявності. Ця таблиця оновлюється автоматично в залежності від дій користувачів.

— Продаж товару: при реєстрації запису в таблиці "Продажі", кількість проданих одиниць автоматично віднімається від відповідного запису в таблиці "Запаси". Це реалізовано за допомогою дії типу "Data: execute an action on a set of rows", яка оновлює поле кількості.

— Надходження товару: при створенні запису в таблиці "Надходження", відповідна кількість одиниць додається до запасів того самого товару. Це виконується через окрему автоматизовану дію з використанням виразів типу $[Запаси] + [Кількість]$.

— Перевірка критичного рівня: у таблиці "Товари" для кожної позиції вказується мінімально допустимий залишок. Після кожного продажу система автоматично порівнює фактичну кількість у запасі з критичним рівнем.

— Зв'язки між таблицями забезпечують цілісність обліку. Наприклад, кожен запис у продажах або надходженнях обов'язково містить посилання на відповідний товар, що виключає помилки у підрахунках.

— Логування змін: усі зміни в кількості фіксуються в таблиці "Журнал" з інформацією про тип операції, дату, користувача та зміну в залишках.

Завдяки цій логіці система забезпечує реальний час оновлення залишків, мінімізує людський фактор та надає актуальну інформацію для прийняття рішень про поповнення складу або аналіз динаміки продажів.

2.4.4 Налаштування форм і дій для ролей користувачів

Щоб інтерфейс був зрозумілим і безпечним для кожного працівника, у AppSheet реалізовано роль-орієнтований доступ. Кожен користувач входить під корпоративним Google-акаунтом; у таблиці Users йому призначається роль (Admin, Warehouse, Seller, Manager). Подальші форми, подання (views) і дії (actions) показуються або приховуються через вираз `CONTEXT("UserRole")` (табл.2.2).

Таблиця 2.2 - Дії для ролей користувачів

Роль	Доступні подання	Ключові форми / дії	Обмеження даних
Admin	«Товари», «Запаси», «Продажі», «Надходження», «Постачальники», «Звіти», «Журнал»	Add / Edit Product Add / Edit Product Bulk Update Min-Stock Export Report (PDF)	Жодних – повний доступ
Warehouse	«Запаси», «Надходження»	New Arrival (Quick-Action) • Adjust Stock	Бачить лише свої записи через security-filter [UserEmail]=USEREMAIL()
Seller	«Продажі», «Товари (view-only)»	New Sale (Form) Scan QRCode → Quick Add	Не може редагувати довідник товарів або змінювати запаси напряму
Manager	«Звіти», «Запаси (read-only)»	Generate Sales Report Export to Excel	Лише читання операцій, без права запису

Форма «New Sale» (для продавця)

- Відображається коли `CONTEXT("UserRole")="Seller"`.
- Поля: `OrderDate (auto-NOW())`, `Product (Ref → Товари)`, `Quantity`.
- Валідація: вираз `IF([Quantity] <= ANY(SELECT(Stock[CurrentQuantity], [ProductID]=[_THISROW].[ProductID])))`, `TRUE`, `FALSE`) блокує продаж, якщо товару недостатньо.

— Після збереження виконується дію “Update Stock-Minus” → залишок зменшується.

Форма «New Arrival» (для комірки)

- Доступна лише полі Warehouse.
- Поля: ArrivalDate, Supplier (Ref), Product, Quantity.
- Дія “Update Stock-Plus” додає кількість; бот «Low-Stock Resolved» скасовує попереднє попередження, якщо рівень запасу піднявся вище критичного.

Дії для адміністратора

- Inline-Edit Price – дозволяє швидке редагування поля Price без відкриття форми.
- Bulk Update Min-Stock – масове встановлення мінімального запасу для вибраних позицій через action type “set the values of some columns in this row”.
- Export Report (PDF) – workflow, що формує PDF-файл згідно з Google Docs-шаблоном і надсилає його на e-mail адміністратора.

Безпека та фільтрація

- Security Filters обмежують рядки для ролей Seller і Warehouse.
- Поля типу Enum + “Suggested values” гарантують коректний вибір категорій та постачальників.
- User-Settings зберігає мову та тему інтерфейсу — персоналізований UI без впливу на дані.

Таким чином, налаштування форм і дій забезпечує чітке розмежування відповідальності, мінімізує ризик помилок під час введення даних і робить щоденну роботу персоналу інтуїтивно зрозумілою.

2.4.5 Автоматизація процесів з використанням Bots і Actions

Для підвищення зручності використання та зменшення кількості ручної роботи в системі управління запасами було реалізовано автоматизовану обробку

даних за допомогою вбудованих інструментів AppSheet: Actions (дії) та Bots (автоматичні сценарії).

Дії (Actions). У системі реалізовано понад 10 користувацьких дій, зокрема:

- UpdateStock_Add – додає кількість товару до поточного залишку при надходженні.
- UpdateStock_Subtract – віднімає кількість при оформленні продажу.
- OpenForm_Sale / OpenForm_Arrival – швидкий перехід до відповідної форми.
- MarkAsCritical – позначає товар як такий, що вимагає поповнення (статус: “Мало”).

ResetStatus – повертає статус товару до нормального після надходження.

Ці дії прив’язані до подій у формах або таблицях, виконуються миттєво після збереження запису.

Боти (Bots). Боти AppSheet запускають складні автоматизовані процеси на основі тригерів (додавання, оновлення, видалення даних). У реалізованій системі налаштовано кілька ключових ботів:

- Bot_StockAlert
 - а) Тригер: Зменшення залишку нижче мінімального рівня.
 - б) Дія: Надсилання email-сповіщення керівнику з деталями товару, залишку та датою.
 - в) Додатково: Зміна статусу товару на “Мало”.
- Bot_LogChange
 - а) Тригер: Будь-яка дія в таблицях «Продажі» або «Надходження».
 - б) Дія: Створення нового запису в таблиці «Журнал» з інформацією: тип операції, дата, користувач, кількість.
- Bot_ReportGeneration
 - а) Тригер: Кнопка або час (раз на тиждень).
 - б) Дія: Генерація звіту (PDF/Excel) із Google Docs-шаблону та надсилання на email керівника.
 - в) Фільтрація: Дані фільтруються за останні 7 днів автоматично.

— Bot_ResetFlags

а) Тригер: Додавання запису про нове надходження.

б) Дія: Якщо залишок перевищив мінімальний, бот скасовує "критичний" статус товару.

Усі боти обмежуються ролями користувачів і умовами безпеки. Наприклад, бот зі сповіщенням про критичні запаси не виконується для записів, створених у тестовому режимі або користувачем без прав адміністратора (рис. 2.13 та 2.14).

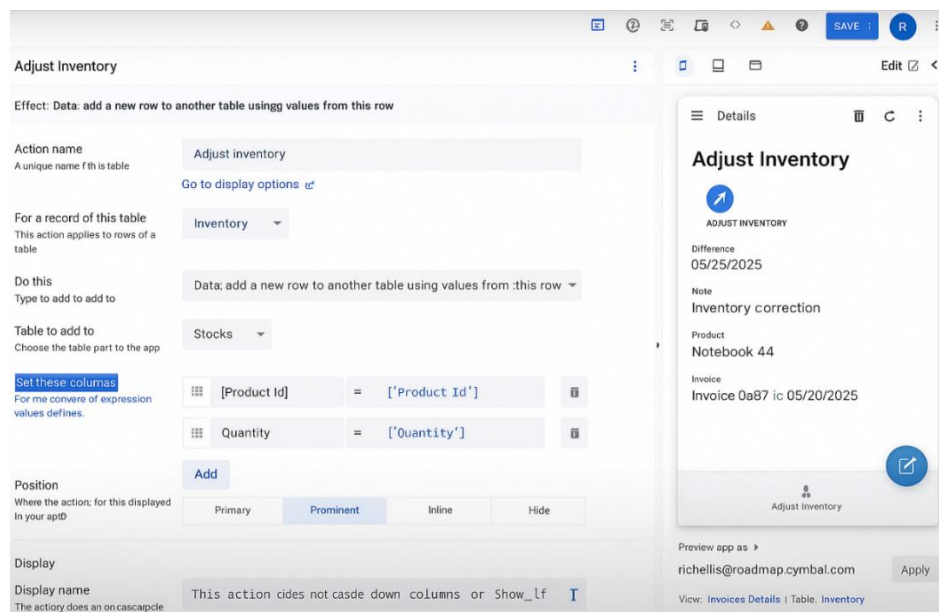


Рисунок 2.13 – Налаштування боту для перевірки наявності

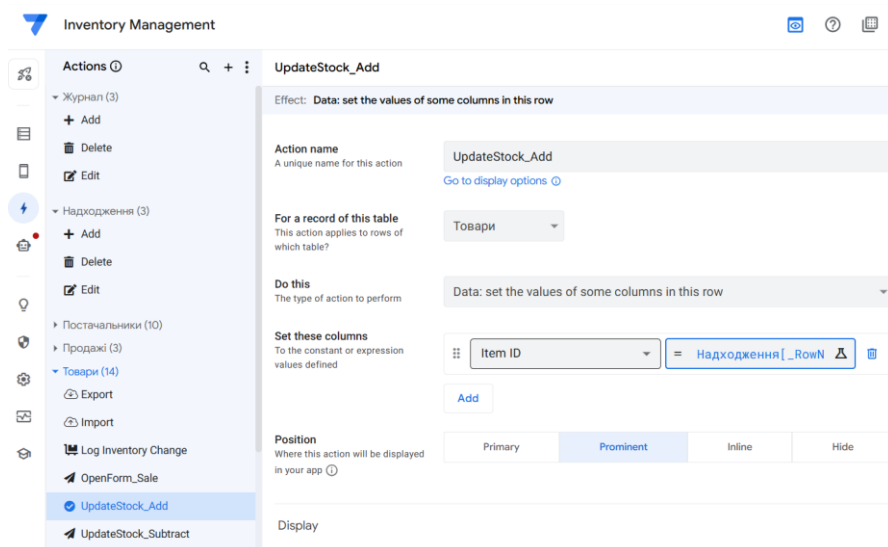


Рисунок 2.14 – Налаштування боту для додання товару

Завдяки поєднанню Actions + Bots, система стала майже повністю автономною в частині розрахунків, оновлень, фіксацій дій і повідомлень. Це дозволяє скоротити час на ручну роботу персоналу, мінімізує ризики помилок і значно підвищує прозорість процесів.

2.4.6 Реалізація звітів та аналітики

Для підтримки процесів прийняття рішень та контролю за станом запасів у системі реалізовано функціональність формування звітів і перегляду аналітичної інформації. У межах Google AppSheet було налаштовано інтерактивні дашборди, що складаються з таблиць, діаграм, зведених сум і фільтрів. Зокрема, керівник має змогу переглядати статистику продажів за період, залишки товарів у розрізі категорій, а також список критичних позицій, які потребують поповнення.

Уся аналітика будується на основі реальних записів у таблицях "Продажі", "Запаси", "Надходження", які оновлюються в режимі реального часу. Для цього використано вбудовані агрегатні функції AppSheet (SUM(), COUNT(), SELECT() та інші), що дозволяють обчислювати загальні суми, кількість операцій, залишки та середні показники. Також застосовано фільтри за датою, категорією товару або постачальником, що забезпечує гнучкість при перегляді даних. Звіти відображаються у вигляді таблиць, гістограм та кільцевих діаграм, з можливістю їх сортування та деталізації (рис. 2.15).

Крім того, було реалізовано можливість генерації звітів у форматі PDF або Excel із використанням Google Docs-шаблонів. Керівник може натисканням кнопки сформувати щотижневий або щомісячний звіт, який буде автоматично згенеровано та надіслано на його електронну пошту. Така функція дозволяє не лише бачити поточний стан справ, а й зберігати історичні дані у вигляді документів, придатних для друку, подальшого аналізу або подання у звітності.

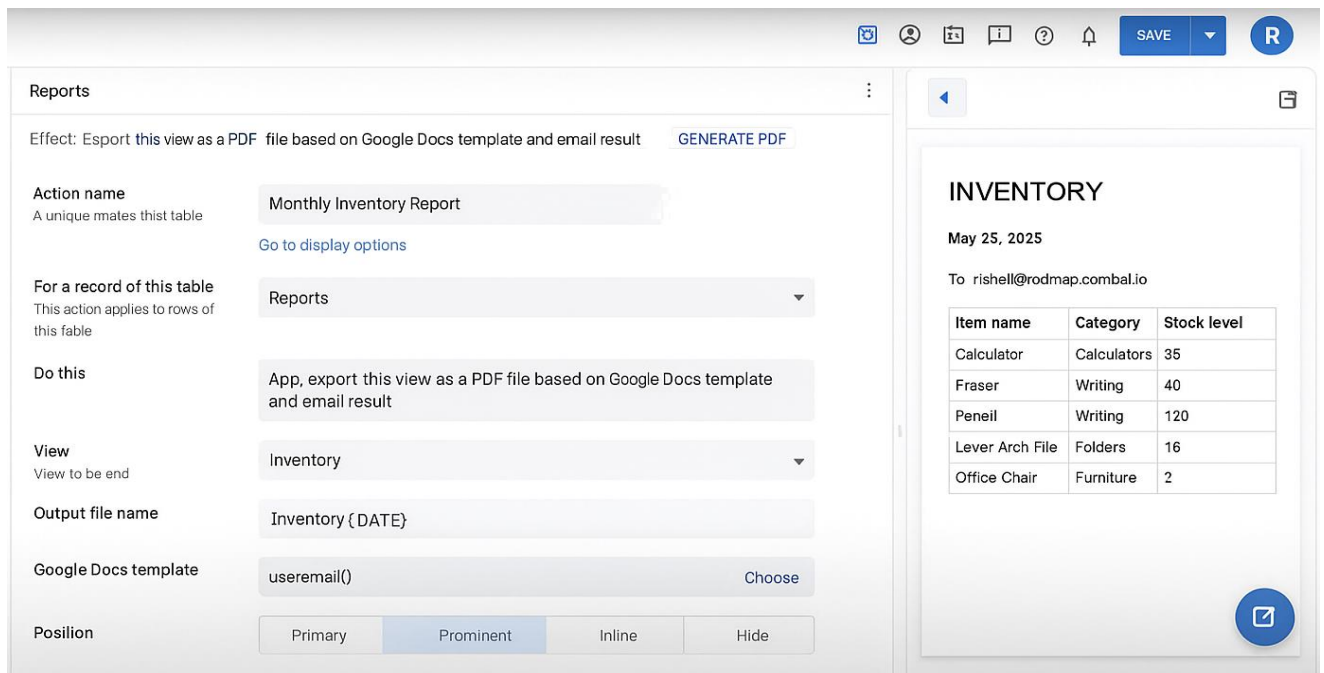


Рисунок 2.15 – Налаштування формування звіту

2.4.7 Забезпечення доступу з різних пристроїв

Однією з ключових переваг використання платформи Google AppSheet є можливість доступу до розробленого застосунку з будь-якого пристрою, що має підключення до Інтернету. У розробленій системі управління запасами це дозволило організувати ефективну роботу персоналу магазину незалежно від місця знаходження. Наприклад, продавці можуть вносити продажі з мобільного телефону прямо на торговій точці, а керівник має змогу контролювати залишки та формувати звіти з домашнього комп'ютера чи планшета.

Застосунок працює як у веббраузері, так і у вигляді мобільного додатку, що встановлюється на пристрої з Android або iOS. Інтерфейс автоматично адаптується до розмірів екрана, забезпечуючи зручність використання на телефонах із невеликим дисплеєм. Усі зміни, які вносяться з одного пристрою, миттєво синхронізуються із базою даних Google Sheets, і стають доступними на інших пристроях у реальному часі.

Таке рішення забезпечує гнучкість і мобільність у роботі з системою, дозволяє скоротити затримки при внесенні даних та уникнути дублювання інформації. Крім того, AppSheet підтримує можливість роботи в офлайн-режимі з подальшою синхронізацією при підключенні до мережі, що особливо корисно у випадках поганого інтернет-зв'язку або використання застосунку в польових умовах. Це робить систему надійною та зручною для всіх категорій користувачів.

2.5 Результати тестування

Після завершення налаштування основних функцій системи управління запасами було проведено комплексне тестування для перевірки правильності її роботи, зокрема коректності оновлення залишків, формування звітів та взаємодії між таблицями. Тестування здійснювалося як вручну, так і за допомогою вбудованих інструментів Google AppSheet, що дозволяють відстежувати виконання дій та автоматизованих сценаріїв.

Результати показали, що система успішно обробляє сценарії надходження та продажу товарів: відповідні значення у таблиці "Запаси" змінюються згідно з логікою, вказаною у діях. Також перевірено функціональність журналу змін — усі операції логуються з точністю до користувача, дати та типу дії. Окрему увагу приділено роботі з критичним рівнем запасів: система коректно визначає товари з недостатнім залишком і формує відповідні попередження (табл. 2.3).

Таблиця 2.3 – Результати тестування основних функцій системи

№	Тестований функціонал	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація надходження товару	Запаси збільшуються відповідно до кількості в записі	Запаси оновлено правильно	Пройдено
2	Реєстрація продажу товару	Запаси зменшуються на відповідну кількість	Дані зменшено коректно	Пройдено
3	Автоматичне сповіщення про низький рівень	Виводиться попередження при зниженні запасів до критичного	Повідомлення з'являється автоматично	Пройдено
4	Створення PDF-звіту	Генерується документ з актуальними даними	Звіт сформовано і надіслано	Пройдено
5	Робота з мобільного пристрою	Інтерфейс адаптується, усі функції доступні	Повна сумісність	Пройдено
6	Логування змін у таблиці "Журнал"	Всі зміни відображаються із зазначенням дати, типу дії, автора	Дані логуються без помилок	Пройдено

Інтерфейс користувача протестовано на різних пристроях (смартфон, планшет, ПК), що дозволило переконатися у коректному відображенні та зручності використання. Не виявлено жодних критичних помилок або втрати даних у процесі тестування. Таким чином, реалізоване рішення відповідає вимогам, поставленим на етапі проєктування, та може бути використане для щоденної роботи магазину без додаткових доробок.

2.6 Впровадження системи

Після успішного завершення етапу розробки та тестування було проведено впровадження системи управління запасами у модельному магазині канцелярських товарів. Впровадження включало підготовку базових довідників (списку товарів, категорій, постачальників), налаштування доступів для різних користувачів та проведення первинного навчання персоналу щодо використання мобільного застосунку на базі AppSheet.

Завдяки використанню хмарної платформи Google AppSheet, процес впровадження пройшов максимально швидко й без потреби у встановленні додаткового програмного забезпечення. Усі дані зберігаються в Google Sheets, що дозволяє легко оновлювати інформацію, при цьому не порушуючи структуру системи. Застосунок було надано співробітникам магазину шляхом надання посилання на мобільну версію програми з подальшим доступом через акаунти Google. Для кожної ролі було налаштовано окремий рівень доступу: продавці можуть тільки вносити продажі, комірники — фіксувати надходження, а керівник має повний доступ до усіх функцій.

Таблиця 2.4 – Етапи впровадження системи в магазині канцелярії

№	Етап	Опис
1	Підготовка довідкових даних	Створення записів про товари, категорії, постачальників
2	Налаштування ролей	Призначення прав доступу для продавця, комірника та керівника
3	Тестова експлуатація	Перевірка дійсності логіки та сценаріїв на обмеженій кількості даних
4	Навчання персоналу	Інструктаж працівників щодо використання AppSheet-додатку
5	Повноцінний запуск	Перехід до щоденної експлуатації системи в магазині

Після декількох днів експлуатації система показала високу стабільність, коректну роботу логіки оновлення залишків, своєчасне надсилання попереджень про критичний рівень запасів та формування звітності. Працівники швидко адаптувалися до роботи з інтерфейсом, а керівництво магазину відзначило підвищення прозорості обліку та зниження кількості помилок, що раніше виникали при веденні паперової звітності або використанні окремих таблиць без загальної системи. Отриманий результат підтвердив доцільність використання low-code технологій для автоматизації бізнес-процесів у малому роздрібному бізнесі.

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Динамічні явища на поверхні землі

Динамічні явища на поверхні Землі включають широкий спектр природних процесів, які можуть мати значний вплив на безпеку людей та інфраструктури. Ці явища охоплюють геологічні, метеорологічні та гідрологічні зміни, кожне з яких має свої специфічні ризики та можливості для запобігання катастрофам [19].

Усі надзвичайні ситуації в Україні регулюються відповідно до національного законодавства, зокрема згідно з наступними ключовими документами: Кодекс цивільного захисту України, положення про підсистему реагування на надзвичайні ситуації та проведення аварійно-рятувальних та інших невідкладних робіт у рамках єдиної державної системи цивільного захисту, положення про штаб з ліквідації наслідків надзвичайної ситуації та організацію оперативно-технічної і звітної документації, наказ Державної служби України з надзвичайних ситуацій [20-23].

До динамічних явищ на поверхні землі належать абіотичні та біотичні небезпеки. Одним із проявів природних небезпек є стихійні лиха.

Стихійні лиха - це природні явища, які мають надзвичайний характер та призводять до порушення нормальної діяльності населення, загибелі людей, руйнування і знищення матеріальних цінностей.

За причиною виникнення стихійні лиха поділяють на: тектонічні (пов'язані з процесами, які відбуваються в надрах земної кори), топологічні (пов'язані з процесами, які відбуваються на поверхні землі), метеорологічні (пов'язані з процесами, які відбуваються в атмосфері).

Види абіотичних небезпек: літосферні, гідросферні, атмосферні, космічні. Серед літосферних небезпек розрізняють землетруси, вулкани, зсуви, селі.

— Землетрус — підземні поштовхи та коливання земної поверхні, зумовлені раптовими зміщеннями і розривами в корі або у верхній частині мантії, які передаються на великі відстані у вигляді пружних коливань.

— Зсув - сповзання мас гірських порід вниз по схилу, яке виникає через порушення рівноваги. Зсуви бувають повільні (см/доба), середньої швидкості (м/год), швидкі (км/год).

— Селі - це паводки з великою концентрацією ґрунту, мінеральних частин, каміння, уламків гірських порід (від 10-15 до 75% об'єму потоку). За складом матеріалу, що переносить потік, розрізняють: грязьові, грязекам'яні, водокам'яні.

Гідросферні небезпеки - повені, снігові лавини, шторми, цунамі.

— Повінь — значне затоплення місцевості внаслідок підйому рівня води в річці, озері, водосховищі. В Україні повені спостерігаються в басейнах Дніпра, Дністра, Прип'яті. В останні роки найчастіше спостерігаються в Закарпатті (Західний Буг та Тиса).

— Снігова лавина - величезна маса снігу, яка зсувається або падає зі стрімких гірських схилів, захоплюючи різні об'єкти, що трапляються на шляху. Лавина супроводжується утворенням передлавинної поверхневої хвилі, що має найбільшу руйнівну силу. Розрізняють сухі (зимові) та мокрі (весняні) снігові лавини.

— Шторм - тривалий, дуже сильний вітер, що спричиняє значні руйнування на суші та велике хвилювання на морі.

— Цунамі - великі хвилі, що виникають на поверхні океану під час підводних землетрусів. Атмосферні небезпеки - бурі, урагани, тайфуни, цунамі, смерчі, морози, засуха тощо.

— Ураган - вітер руйнівної сили зі швидкістю 35 м/с.

— Буря - шторм, тривалий, дуже сильний вітер (понад 20 м/с), спричинений зазвичай циклоном.

— Смерч - сильний локальний атмосферний вихор (діаметр до 1000м), в якому повітря обертається зі швидкістю до 100 м/с.

3.2 Організація ведення робіт в аварійних умовах

Організація ведення робіт в аварійних умовах є надзвичайно важливим аспектом забезпечення безпеки та ефективності в небезпечних ситуаціях. Одні із основних кроків такої ситуації [24, 25]:

- оцінка ризиків: Перед початком будь-яких аварійних робіт необхідно провести оцінку ризиків. Визначте потенційні небезпеки, що виникають у зв'язку з аварійною ситуацією, і розробіть план заходів для зниження цих ризиків.

- створення команди: Сформууйте команду, яка буде відповідальна за проведення аварійних робіт. Команда повинна складатися з кваліфікованих фахівців з необхідними навичками та досвідом в роботі з аварійними ситуаціями.

- встановлення комунікації: Забезпечте ефективну систему комунікації всередині команди та з іншими відповідними структурами (пожежна служба, поліція, медична допомога тощо). Засоби зв'язку повинні бути надійними та доступними, щоб забезпечити швидкий обмін інформацією.

- планування та координація: Розробіть детальний план робіт, включаючи послідовність дій, розподіл завдань, використання необхідного обладнання та ресурсів. Координуйте дії команди таким чином, щоб досягти ефективного вирішення проблеми.

- навчання та підготовка працівників: Працівники, які беруть участь у роботах в аварійних умовах, повинні мати необхідні навички та знання. Забезпечте їм належне навчання з процедур безпеки, використання спеціального обладнання та заходів, які слід вживати під час аварій.

Складовою частиною процесу ліквідації наслідків НС є рятування людей. Цей процес представляє собою взаємопов'язаний комплекс робіт, які за характером виконання діляться на три специфічні групи: рятувальні, спеціальні і допоміжні.

Рятувальні роботи, що безпосередньо пов'язані із рятуванням людей включають у себе:

- пошук постраждалих у місцях їхнього можливого блокування;

- деблокування постраждалих (забезпечення доступу до них);
- надання постраждалим домедичної допомоги;
- евакуація постраждалих із місць блокування.
- Спеціальні роботи включають:
- гасіння пожеж;
- ліквідацію аварії на комунально-енергетичних і технологічних мережах;
- улаштування проїздів (проходів) у завалах;
- зміцнення (обвалення) нестійких конструкцій.

У результаті виконання спеціальних робіт створюються умови найбільш сприятливі для виконання рятувальних робіт і запобігання додаткового ураження людей. Допоміжні роботи пов'язані з інженерною та організаційною підготовкою ділянки рятувальних робіт та робочих місць. До них відносяться:

- розчищення майданчиків;
- установлення на них техніки;
- огороження, попереджувальних знаків;
- освітлення робочих місць.

Час, необхідний для виконання технологічних операцій, є основним критерієм, що характеризує доцільність їх застосування у технологічному процесі порятунку людей, у певних організаційнотехнологічних умовах.

У практиці рятування постраждалих при обваленні будівель використовуються наступні рятувальні технології [26]:

- пошук постраждалих за допомогою спеціально навчених собак (кінологічний спосіб);
- пошук постраждалих за допомогою спеціальних приладів;
- деблокування постраждалих із завалу, що складається із дрібних уламків, способом розбирання завалу зверху;
- деблокування постраждалих із завалу, що складається із великих уламків, способом розбирання завалу зверху;

- деблокування постраждалих із завалу способом суцільного горизонтального розбирання;
- деблокування постраждалих способом улаштування лазу у завалі;
- деблокування постраждалих із завалених приміщень;
- деблокування постраждалих з верхніх поверхів будівлі з використанням вертольоту;
- деблокування постраждалих з верхніх поверхів будівлі із застосуванням автодрабин;
- порятунок постраждалих з верхніх поверхів будівлі за допомогою автовишок та автопідйомників;
- порятунок постраждалих з верхніх поверхів будівлі по збереженим або тимчасово відновленим сходовим маршам;
- порятунок постраждалих з верхніх поверхів будівлі з використанням канатної дороги;
- деблокування постраждалих з верхніх поверхів будівлі з використанням альпіністських засобів.

У загальному вигляді процес рятування постраждалих може бути представлений як комплексний технологічний процес, що включає наступні етапи [27]:

- загальна спеціальна розвідка осередку ураження та об'єкта робіт;
- підготовчі роботи;
- аварійно-технічні роботи;
- пошуково-рятувальні роботи;
- роботи з деблокування та витягання постраждалих;
- надання домедичної та лікарської допомоги, медична евакуація поранених;
- евакуація, упізнання та поховання загиблих.

На кожному з наведених технологічних етапів здійснюються відповідні види робіт, а вони, у свою чергу виконуються певними способами. Найбільш

складним технологічним етапом при обваленні будівель і споруд є інженерні роботи з деблокування та витягання постраждалих.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто основної мети — створення інформаційної системи для обліку товарів у магазині канцелярії з використанням хмарної платформи Google AppSheet. Розроблена система забезпечує повний цикл управління запасами: від надходження товару на склад до фіксації продажів, автоматичного оновлення залишків та формування звітності. Важливим аспектом стало те, що рішення створено із застосуванням low-code підходу, що дозволило зосередитися на логіці бізнес-процесів замість технічної реалізації з нуля. Система побудована на базі Google Sheets, що забезпечує зручність у редагуванні структури таблиць та максимальну прозорість даних для користувачів.

Для досягнення поставлених завдань було проаналізовано існуючі шаблони AppSheet, зокрема "Inventory Management", який було обрано як основу. У межах розробки шаблон було суттєво розширено та адаптовано під специфіку предметної області: додано нові сутності (таблиці "Постачальники", "Продажі", "Надходження", "Журнал"), реалізовано зв'язки між ними, впроваджено перевірку залишків і контроль критичних рівнів. Також у системі налаштовано окремі інтерфейси та дії для трьох основних ролей: продавця, комірника та керівника. Автоматизація ключових процесів досягнута за допомогою вбудованих механізмів Bots та Actions, які забезпечують своєчасне оновлення даних, логування операцій і надсилання сповіщень без участі користувача.

Практична реалізація проєкту показала високу ефективність хмарного рішення у поєднанні з мобільним доступом. Користувачі можуть працювати з додатком як з комп'ютера, так і з мобільних пристроїв, що значно підвищує гнучкість і оперативність. Система легко масштабована та придатна до подальшого розширення: можлива інтеграція з фінансовими або CRM-сервісами, додавання аналітичних панелей на основі Data Studio або BigQuery, а також підключення форм онлайн-замовлень. Таким чином, створена система є повноцінним цифровим

інструментом, що не лише автоматизує основні операції, а й відкриває нові можливості для розвитку торгової діяльності малого бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрущенко М.І., Лисенко О.В. Хмарні технології в управлінні бізнесом. – Київ: Освіта України, 2021. – 216 с.
2. Гаврилюк А.П., Піскунова Н.В. Інформаційні системи в торгівлі. – Харків: ФОП Сидоренко, 2020. – 232 с.
3. Романенко Ю.О., Литвиненко І.С. Розробка мобільних додатків із використанням Google AppSheet. – Львів: Нові Знання, 2023. – 188 с.
4. Ситник М.А. Хмарні платформи Google Workspace у сучасній освіті та бізнесі. – Тернопіль: ТНТУ, 2022. – 154 с.
5. Ковальчук І.І. Автоматизація обліку в малому бізнесі. – Дніпро: Бізнес Академія, 2021. – 178 с.
6. Google AppSheet Help Center. [Електронний ресурс] – Режим доступу: <https://support.google.com/appsheet>
7. AppSheet Expression Reference. [Електронний ресурс] – Режим доступу: <https://help.appsheet.com/en/articles/1974764-expression-types>
8. Google Sheets API Documentation. [Електронний ресурс] – Режим доступу: <https://developers.google.com/sheets/api>
9. AppSheet Sample Templates Gallery. [Електронний ресурс] – Режим доступу: <https://www.appsheet.com/templates>
10. Google Workspace Learning Center. [Електронний ресурс] – Режим доступу: <https://workspace.google.com/learning-center/>
11. Дмитренко В.С., Саприкін О.О. Управління запасами в малих торгових підприємствах. – Київ: КНЕУ, 2022. – 204 с.
12. OpenAI. Створення інтерфейсів на основі LLM. [Електронний ресурс] – Режим доступу: <https://openai.com/research/>
13. Гуменюк С.О., Демченко І.В. Програмна інженерія та low-code платформи. – Вінниця: ВДТУ, 2023. – 198 с.

14. AppSheet Automation (Bots & Events). [Електронний ресурс] – Режим доступу: <https://help.appsheet.com/en/articles/3838780-introduction-to-appsheet-automation>
15. AppSheet Sample: Inventory Management. [Електронний ресурс] – Режим доступу: <https://www.appsheet.com/templates/Inventory-Management>
16. Боярчук Д.Ф. Інформаційні технології в логістиці. – Одеса: ОНПУ, 2021. – 212 с.
17. Google AppSheet YouTube Channel. [Електронний ресурс] – Режим доступу: <https://www.youtube.com/@appsheet>
18. Стандарти оформлення електронної звітності в бізнесі. – Львів: Техніка, 2022. – 136 с.
19. Запорожець О.І., Халмурадов В.І., Применко В.І. Безпека життєдіяльності: Підручник. – К.: Центр учбової літератури, 2013. – С. 84-130.
20. Кодекс цивільного захисту України [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/5403-17#Text>
21. Про затвердження Положення про підсистему реагування на надзвичайні ситуації, проведення аварійно-рятувальних та інших невідкладних робіт єдиної державної системи цивільного захисту : наказ Міністерства внутрішніх справ України від 04.05.2016 №356.
22. Про затвердження Положення про штаб з ліквідації наслідків надзвичайної ситуації та видів оперативно-технічної і звітної документації штабу з ліквідації наслідків надзвичайної ситуації : наказ Міністерства внутрішніх справ України від 26.12.2014 №1406.
23. Про організацію роботи штабу з ліквідації наслідків надзвичайної ситуації та забезпечення його готовності : наказ Державної служби України з надзвичайних ситуацій від 16.03.2015 №149.
24. Рекомендації для роботодавців щодо організації виконання робіт підвищеної небезпеки під час воєнних (бойових) дій [Електронний ресурс] – Режим доступу: <https://dsp.gov.ua/faq/rekomendatsii-dlia-robotodavtsiv-shchodo-orhanizatsii-vykonannia-robit-pidvyshchenoi-nebezpeky-pid-chas-voiennykh-boiovykh-dii/>

25. Аветисян В.Г., Тригуб В.В., Грицина І.М., Остапов К.М. Організації аварійнорятувальних робіт : курс лекцій – Х: НУЦЗУ, 2017. – 141 с.

26. Організація аварійно-рятувальних робіт : навчальний посібник /Р.Т. Ратушний, В.Б. Лоїк, О.Д. Синельніков, В.М. Ковальчук – Львів: Видавництво ЛДУ БЖД, 2020. – 394 с.

ДОДАТКИ

Додаток А – Диск із кваліфікаційною роботою бакалавра