

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ
ПРОКАТУ САМОКАТІВ
SOFTWARE OF THE AUTOMATED SCOOTER RENTAL SYSTEM

Виконав: студент 4 курсу, групи КНТз-119
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЛІСІЧКІН Д.Ю.

(прізвище та ініціали)

Керівник ПАРХОМЕНКО А.В.

(прізвище та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЛІСІЧКІНА Данили Юрійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення автоматизованої системи прокату самокатів. Software of the Automated Scooter Rental System
керівник проєкту (роботи) к.т.н., доцент ПАРХОМЕНКО Анжеліка Володимирівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 4 ” квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 15 травня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Дослідження особливостей використання платформи BAS для розробки автоматизованої системи обліку. 3. Проєктування автоматизованої системи обліку прокату самокатів. 4. Конструювання автоматизованої системи обліку прокату самокатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “4” квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Дослідження особливостей платформи BAS	3 тиждень	Розділ 2
4	Аналіз вимог. Розробка структурної та функціональної схем програми програми	4 тиждень	Розділ 3
5	Розробка алгоритму функціонування автоматизованої системи	5 тиждень	Розділ 4
5	Конструювання автоматизованої системи	6 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	ПЗ, Додатки
7	Захист роботи	8 тиждень	

Студент

_____ Данило ЛІСІЧКІН
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Анжеліка ПАРХОМЕНКО
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 93 с., 3 табл., 48 рис., 2 дод., 17 джерел.

БІЗНЕС-ПРОЦЕС, СИСТЕМА ОБЛІКУ, ПРОКАТ САМОКАТІВ, ПЛАТФОРМА АВТОМАТИЗАЦІЇ, ЗАПИТ, ДОВІДНИК, ЗВІТ, РЕГІСТР, СКРИПТОВА МОВА, МОВА ЗАПИТІВ.

Об'єкт дослідження – бізнес-процес обліку оренди електросамокатів.

Предмет дослідження – методи та програмні засоби автоматизації бізнес-процесу обліку оренди електросамокатів.

Мета роботи – дослідження та розробка автоматизованої системи обліку електросамокатів для підвищення ефективності процесів видачі-прийняття обладнання та інвентаря та проведення фінансових розрахунків.

Матеріали, методи та технічні засоби: структурне програмування, технологічна платформа BAS, вбудована скриптова мова програмування та мова запитів системи BAS.

Результати. Створено програмний застосунок, що надає можливість контролювати процеси закупівлі самокатів для пункту прокату, видачі-прийняття самокатів та захисного спорядження, розрахунку вартості оренди з урахуванням терміну використання.

Висновки. Розроблено програмне забезпечення системи для пункту прокату електросамокатів студентського містечка, яка автоматизує бізнес-процес обліку електросамокатів.

Галузь використання – комерційна діяльність з прокату різних видів обладнання та інвентаря.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 93 pages, 3 tables, 48 figures, 2 appendixes, 17 sources.

BUSINESS PROCESS, ACCOUNTING SYSTEM, SCOOTER RENTAL, AUTOMATION PLATFORM, INQUIRY, DIRECTORY, REPORT, REGISTER, SCRIPT LANGUAGE, REQUEST LANGUAGE.

Object of study is the business process of accounting for the rental of electric scooters.

Subjects of study are methods and software tools for automating the business process of accounting for the rental of electric scooters.

The goal of the work is research and development of an automated accounting system for electric scooters to improve the efficiency of inventory issuance and acceptance processes and financial settlements.

Materials, methods and technical tools: structural programming, BAS technology platform, the embedded scripting language and the query language of the BAS system.

The results. A software application has been created that allows to control the processes of purchasing scooters for a rental point, issuing and accepting scooters and protective equipment, calculating the cost of rent taking into account the period of usage.

Conclusions. The system software for the campus electric scooter rental point, which automates the business process of accounting for electric scooters, has been developed.

The fields of application are commercial activities for the rental of various types of equipment and inventory.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Особливості автоматизації бізнес-процесів.....	10
1.2 Огляд функціональних можливостей BPMS.....	12
1.3 Аналіз та оцінка програмного забезпечення BPMS.....	16
1.4 Постановка завдань дипломної роботи.....	21
2 Дослідження особливостей використання платформи BAS для розробки автоматизованої системи обліку.....	23
2.1 Структура системи BAS.....	23
2.2 Проєктування інформаційної бази.....	25
2.3 Опис об'єктів конфігурації.....	30
2.4 Висновки до розділу 2.....	34
3 Проєктування автоматизованої системи обліку прокату самокатів.....	35
3.1 Аналіз вимог до автоматизованої системи.....	35
3.2 Розробка структури АС.....	37
3.3 Вибір інструментарію розробки.....	39
3.3.1 Вбудована мова програмування.....	39
3.3.2 Мова запитів.....	40
3.3.3 Система компоновки даних.....	43
3.4 Висновки до розділу 3.....	45
4 Конструювання автоматизованої системи обліку прокату самокатів...	46
4.1 Схема алгоритму функціонування АС.....	46
4.2 Розробка АС обліку прокату самокатів.....	49
4.2.1 Реалізація базового функціоналу АС.....	49
4.2.2 Реалізація додаткового функціоналу АС.....	54
4.3 Висновки до розділу 4.....	65
Висновки.....	67

Перелік джерел посилань.....	68
Додаток А Текст програми.....	70
Додаток Б Слайди презентації.....	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
BPMN	– Business Process Model and Notation;
BPMS	– Business Process Management Suite;
CRM	– Customer Relationship Management;
ERP	– Enterprise Resource Planning;
HRIS	– Human Resource Information System;
HRM	– Human Resource Management;
RPA	– Robotic Process Automation;
АС	– автоматизована система;
БД	– база даних;
ІБ	– інформаційна база;
ІТ	– інформаційні технології;
ПЗ	– програмне забезпечення;
СУБД	– система управління базою даних.

ВСТУП

Згідно зі звітом Forrester під назвою «Жалкий стан цифрової трансформації у 2018 році» [1], до 22% компаній досі не мали досконалих та автоматизованих робочих процесів. І протягом останніх п'яти років ситуація суттєво не змінилася. Але саме цей аспект є ключовим для задоволеності клієнтів, зростання прибутку, тобто успішності бізнесу.

Автоматизація бізнес-процесів — це використання технології для автоматизації ручних повторюваних завдань, що надає співробітникам можливість зосередитися на більш важливих критичних завданнях. Автоматизація бізнес-процесів дозволяє мінімізувати витрати, підвищити ефективність і оптимізувати процеси від простих до складних. Автоматизація бізнес-процесів відрізняється від управління бізнес-процесами, яке включає управління складними загальноорганізаційними процесами за допомогою різних методологій [2].

Основною метою автоматизації бізнес-процесів є підвищення соціально-економічної ефективності підприємства або компанії шляхом цілеспрямованого впровадження програмних систем, що підвищують швидкість і точність реалізації бізнес-процесів, мінімізують при цьому помилкові дії і рішення, що виникають під впливом так званого "людського фактору". Існуючий рівень розвитку програмного забезпечення дозволяє автоматизувати переважну частину бізнес-процесів для ефективної організації різних видів діяльності. Але необхідність урахування специфіки конкретного бізнес-процесу потребує адаптації існуючих програмних засобів до виконання конкретних задач. Тому, тема роботи є актуальною.

Мета роботи — дослідження та розробка автоматизованої системи обліку електросамокатів для підвищення ефективності процесів видачі-прийняття обладнання та інвентаря та проведення фінансових розрахунків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості автоматизації бізнес-процесів

Автоматизація бізнес-процесів є сходинкою на шляху до впровадження культури безперервної трансформації бізнесу.

Автоматизація бізнес-процесів вимагає певної ясності щодо процесу безпосередньо на етапі проектування. Крім того, відображення процесів може надати ясність усім співробітникам і також служити навчальним ресурсом. Статистика, яку можливо отримати в результаті аналізу автоматизованого процесу, може чітко показати розрив між існуючим бізнес-процесом та тим, яким він має бути в ідеалі. Одним із суттєвих результатів автоматизації бізнес-процесів є їх оптимізація. Чітка підзвітність, настроювані сповіщення, цінні відомості та швидший час виконання робіт полегшують усунення марнотратної діяльності та зосереджуються на покращенні завдань, які додають цінності [2].

Завдяки автоматизації бізнес-процесів кожна деталь конкретного бізнес-процесу записується. Ця інформація може бути представлена для демонстрації відповідності під час аудитів. При автоматизації бізнес-процесів з'являється стабільний стандарт результатів. Стандартизація допомагає позиціонувати компанію як надійну, що, у свою чергу, може сприяти збільшенню клієнтської бази. Задоволеність клієнтів є ключовим фактором, що характерне для будь-якої галузі. Зосередження на досконалості процесів і операцій допоможе з легкістю перевершити очікування клієнтів. Якщо діяльність компанії відповідає певним стандартам, клієнти оберуть саме її.

Автоматизація бізнес-процесів не обмежується кількома функціями. Деякі фактори, які можуть вказувати на необхідність автоматизації, включають [2]:

- великий обсяг завдань;
- для виконання завдань потрібні кілька людей;
- чутливий до часу характер завдань;

- значний вплив на інші процеси та системи;
- необхідність відповідності певним стандартам та аудиту.

Якщо діяльність відповідає всім критеріям, перерахованим вище, потрібно автоматизувати бізнес-процес.

Типові автоматизовані процеси в організаціях [2]:

- електронна пошта та push-сповіщення;
- служба підтримки;
- створення прикладів клієнтів;
- агрегація та міграція даних;
- резервне копіювання та відновлення;
- заявки на відпустку співробітників;
- закупівлі;
- процеси кол-центру;
- замовлення на продаж;
- облік робочого часу та відвідуваності;
- нарахування заробітної плати;
- виставлення рахунків.

Основні переваги автоматизації бізнес-процесів:

- збільшення продуктивності завдяки розширеному доступу.

Хмарні засоби автоматизації бізнес-процесів зберігають дані в центральній базі даних, що допоможе отримати доступ до даних з будь-якого місця чи пристрою, коли це потрібно;

- більша прозорість бізнес-процесів. Можливо відстежувати та контролювати процеси під час їх виконання, що дозволяє покращити підзвітність і видимість;

- можливість відстежувати процеси на ходу допоможе стежити за помилками та виправляти їх по мірі їх виникнення. Звіти про продуктивність надають статистику, щоб вживати запобіжних заходів проти повторюваних помилок;

- швидший час виконання робіт і скорочення витрат завдяки меншій кількості ручних втручань;
- покращений розподіл робочої сили, оскільки повсякденні повторювані завдання будуть автоматизовані, а співробітники зосередяться на завданнях, які вимагають людських зусиль і суджень [2].

Приклад схеми бізнес-процесу наведено на рисунку 1.1.

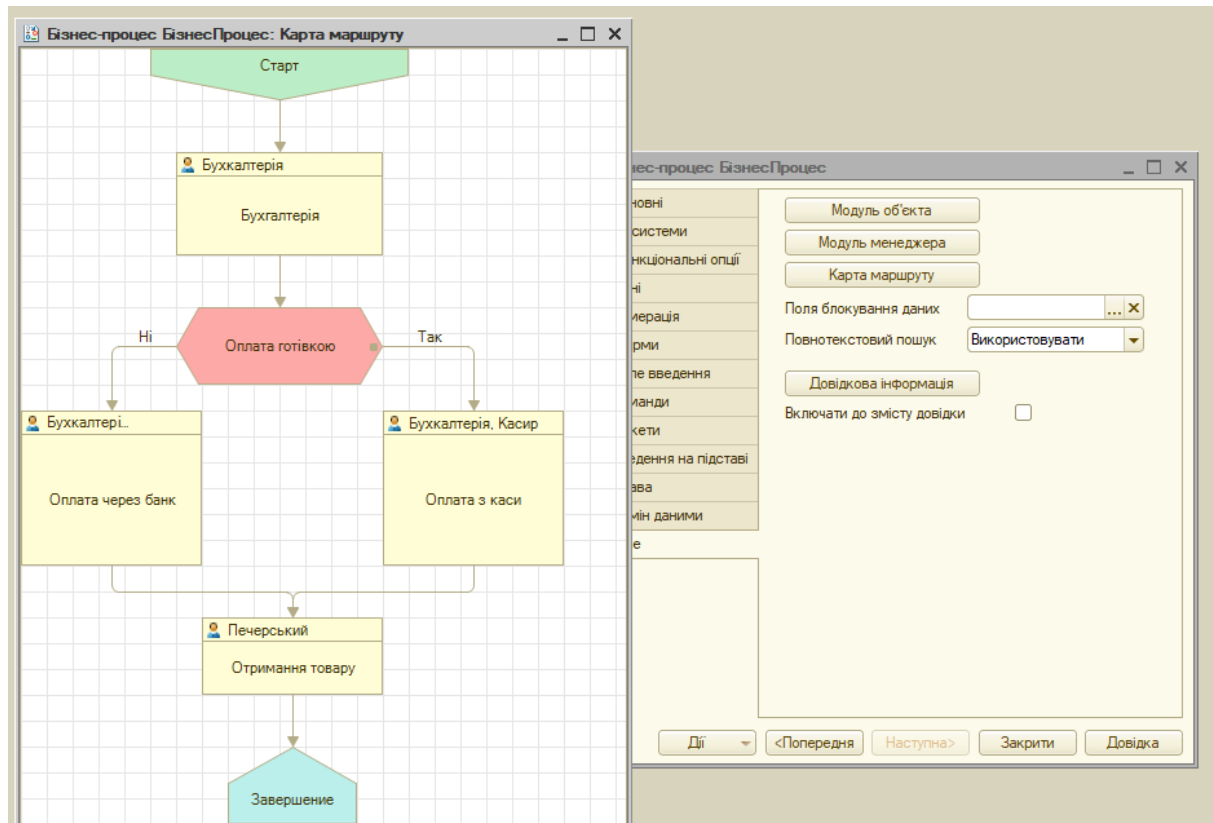


Рисунок 1.1 – Приклад схеми бізнес-процесу

1.2 Огляд функціональних можливостей BPMS

Система управління бізнес-процесами (Business Process Management Suite, BPMS) допомагає підприємствам проєктувати, впроваджувати, контролювати та оптимізувати свої процеси. Це зменшує витрати, спрощує спілкування між командами, покращує обслуговування клієнтів та значно підвищує ефективність компанії [3].

BPMS — це платформа, яка автоматизує чітко визначені бізнес-процеси. Це створює інфраструктуру, яка підвищує гнучкість компанії, роблячи роботу швидшою та легшою.

BPMS — це простий інтуїтивно зрозумілий інструмент, який допомагає компаніям виконувати, керувати та покращувати бізнес-процеси шляхом їх визначення, моделювання, автоматизації та аналізу. Це дозволяє підприємствам збирати дані та використовувати їх для оптимізації ключових бізнес-процесів [3].

Основні етапи роботи BPMS наведено на рисунку 1.2.

Етап 1. Початок. Аналіз діяльності компанії або підприємства.

Етап 2. Документування бізнес-процесів та визначення відповідних ефективних дій для виконання завдань.

Етап 3. Відображення процесів у вигляді діаграм або блок-схем та встановлення набору правил для гнучкого моделювання процесу.

Етап 4. Автоматизація процесів, шляхом збирання дій та створення моделі бізнес-процесів для точного моделювання поведінки процесу та результатів.

Етап 5. Виконання процесів. Завичай виконується ядром системи, яке зчитує виконувану схему та проводить кожен бізнес-процес від початку до завершення. Зазвичай існує декілька варіантів запуску бізнес-процесу: вручну користувачем або автоматично за тригером (певною подією). Під час виконання процесу деякі транзакції система проводить автоматично, а деякі виконує користувач за допомогою системи.

Етап 6. Моніторинг процесу. Відбувається на кожному кроці виконання процесу шляхом вимірювання метрик (наприклад, тривалість або середній час виконання), їх накопичення та оформлення.

Етап 7. Перевірка необхідності удосконалення та оптимізації бізнес-процесів, щоб постійно покращувати здатність досягати бажаних бізнес-результатів. Відбувається шляхом оперативного вношення необхідних змін в схему процесу, правила моделювання та налаштування [3].



Рисунок 1.2 – Узагальнений алгоритм роботи BPMS

Основні характеристики BPMS наступні.

Автоматизація робочого процесу. Системи керування бізнес-процесами мають вбудоване програмне забезпечення (ПЗ) для автоматизації, яке допомагає підприємствам визначати, відображати та автоматизувати загальні процеси та робочі процеси. Вони можуть розробляти критичні шляхи та функції та надавати дозволи для автоматизації з мінімальним наглядом. Системи також допомагають підприємствам постійно контролювати, виявляти та усувати вузькі місця процесу.

Управління даними. Інструменти BPM мають надійні та комплексні функції керування даними, оскільки організації створюють величезні обсяги даних. Вони збирають потрібну інформацію, ефективно керують нею та аналізують її. Користувачі можуть визначати та підтримувати елементи даних і зв'язки між об'єктами даних [3].

Візуальне моделювання робочого процесу. BPMS зі зручним інтерфейсом дозволяє користувачам легко створювати та редагувати візуальну карту робочих процесів. Вони можуть оглянути важливу інформацію та побачити, як просувається робота в режимі реального часу. Система також дозволяє користувачам легко моделювати веб-форми та процеси.

Аналітика та звіти. Інструменти BPM забезпечують аналіз ефективності процесу. Звіти пов'язані з ключовими показниками ефективності та містять дані про окремі дії, процеси, завдання та учасників процесу. Звіти та аналітика допомагають організаціям покращувати існуючі процеси, визначати очікування клієнтів, прогнозувати поведінку ринку та підвищувати загальну норму прибутку.

Інтеграція. Небагато бізнес-процесів існує в одній системі. Наприклад, залучення нового клієнта може включати його адаптацію, надання послуги та отримання оплати. Технологія BPMS повинна координувати виконання процесів у різних системах, щоб підвищити ефективність, обслуговування та взаємодію з клієнтами, одночасно зменшуючи витрати. Система повинна інтегруватися з програмним забезпеченням Customer Relationship Management

(CRM), платформами Human Resource Information System та Human Resource Management (HRIS/HRM) і платформами Enterprise Resource Planning (ERP). Це дозволяє легко запускати та завершувати процеси в цих системах [3].

1.3 Аналіз та оцінка програмного забезпечення BPMS

ПЗ для керування бізнес-процесами допомагає компаніям ефективніше керувати своїми повсякденними операціями та процесами. Воно надає інструменти для проєктування, моделювання, виконання, автоматизації та постійного вдосконалення бізнес-процесів. BPMS має на меті оптимізувати роботу, зменшити неефективність і підвищити загальну продуктивність і задоволеність клієнтів [4].

ПЗ відстежує існуючі операційні процеси, допомагає спростити їх шляхом автоматизації повторюваних завдань і узгоджує їх із цілями організації. Відображаючи повсякденні процеси, BPMS допомагає виявити робочі вузькі місця та усунути їх, роблячи процеси більш пов'язаними та ефективними.

Зробити правильний вибір ПЗ для керування бізнес-процесами на ринку досить складно. Неправильний вибір не тільки означає марну інвестицію в ПЗ, але й може призвести до втрати часу та зусиль.

До найкращих постачальників програмного забезпечення BPMS відносяться наступні [4].

Kissflow – це ПЗ, яке створює цифрове робоче місце. Це допомагає оптимізувати й автоматизувати робочі процеси, проєкти, справи та співпрацю. Деякі з його найкращих функцій включають зручну інформаційну панель, налаштовані шаблони звітів і розширений дизайнер форм. Він пропонує гнучкі тарифні плани від 199 доларів США на місяць для різних компаній, які охоплюють широкий спектр функцій і базу користувачів.

Processmaker – це рішення, яке дозволяє проєктувати, автоматизувати та розгортати бізнес-процеси. Воно ідеально підходить для компаній, яким

потрібне програмне забезпечення BPM корпоративного масштабу, готове до виробництва. Він пропонує широкий вибір хостингу: локальний, хмарний або відкритий. Має три тарифні плани від 1500 доларів на місяць.

Zoho Creator – дозволяє користувачам створювати власні форми, проєктувати робочі процеси та створювати інформаційні сторінки. Найкраще підходить для компаній, які шукають рішення DIY (Do it yourself). Ціни можуть коливатися від 10 до 15 доларів США за користувача на місяць.

Nintex робить цифрову трансформацію реальністю для малого та середнього бізнесу та підприємств. Надає опцію хмари разом із традиційним локальним хостингом. Його розроблено для роботи разом із такими інструментами, як Office365, Sharepoint і Project Server. Платформа Nintex має можливості для розширеного робочого процесу, сучасних форм, відображення та керування процесами, мобільних застосунків і RPA (Robotic Process Automation). У пакеті Nintex є різноманітні продукти за ціною від 850 доларів США на місяць [4].

Bizagi пропонує рішення автоматизації для підприємств у формі 3 продуктів – Bizagi Modeler, Bizagi Studio та Bizagi Automation. Modeler використовує унікальну версію Bizagi BPMN, що означає, що потрібні відповідні ноу-хау та навчання. Bizagi пропонує варіанти хмарного та локального хостингу. Bizagi не публікує свої ціни на своєму вебсайті та вимагає, щоб ви зв'язувалися з ними для отримання пропозиції.

Bpm'online (Україна) – це BPM-система, яку можна придбати як у вигляді окремих продуктів так і як пакетну опцію. Пропонується як локальний, так і хмарний хостинг. Студія Bpm'online має безкоштовну версію та корпоративну версію, яка коштує 25 доларів США на користувача на місяць [4].

Bpm'online – система, що поєднує CRM-функціональність з інструментами моделювання та автоматизації бізнес-процесів. Користувачі отримують можливість створювати власні рішення та бізнес-процеси на платформі bpm'online. Вони також можуть використовувати готові програми

для оптимізації продажів, маркетингу, телефонії клієнтського сервісу та інших процесів [5].

Bizagi (Велика Британія) – це BPM-система, спрямована на моделювання, виконання, автоматизацію та аналіз бізнес-процесів. Система управління бізнес-процесами Bizagi допомагає організаціям трансформуватись у цифрові компанії. BPMS надає функціональні можливості для створення нових робочих моделей, поєднуючи людей, бізнес-застосунки, пристрої та дані на підприємстві. Подібно до інших рішень класу BPMS, система включає безкоштовний інструмент для опису та покращення процесів. Bizagi допомагає автоматизувати будь-який налаштований бізнес-процес і оцифрувати операції, що входять до нього. BizAgі є одним із лідерів ринку систем BPMS. На думку експертів, BizAgі BPMS відрізняє від інших систем орієнтованість на задоволення потреб бізнесу, на протипагу завданням чистої автоматизації [6].

SAP NetWeaver BPM (Німеччина) – це новий крок у розвитку систем автоматизації бізнес-процесів. Як відомо SAP ERP є набором кращих практик від великих світових компаній, об'єднаних в єдиному рішенні. Ці практики дозволяють автоматизувати бізнес-процеси компаній у жорстко заданих рамках. Величезна різниця між SAP Netweaver BPM та тим, що пропонують чисті BPM-гравці, полягає в тому, що універсальний BPM-інструментарій може працювати в гетерогенному ІТ середовищі, у той час як BPM інструментарій SAP працює тільки навколо їх власних процесів.

Результати детального порівняння декілької систем наведено в таблиці 1.1 [5] - [7].

Дослідження показали, що основні функції ПЗ BPM наступні.

Інструмент візуального діаграмування процесу. Найкраще програмне забезпечення для BPM має мати візуальний інструмент діаграми процесів, який дозволяє користувачам легко проектувати робочі процеси без ноу-хау програмування. Це знімає тиск з команди ІТ і передає контроль у руки кінцевого користувача.

Таблиця 1.1 – Порівняння популярних BPM-систем

Критерій	SAP	Bpm'online	Bizagi
Налаштування процесів бізнес-користувачів без програмування	+	+	+
Установка на свій сервер	+	+	+
Підтримка нотації BPMN для моделювання бізнес-процесів	+	+	+
Наявність візуального редактора для моделювання бізнес-процесів	+	+	+
Єдине середовище розробки та виконання бізнес-процесів	+	-	+
Можливість вивантажувати регламенти щодо реалізованих бізнес-процесів	+	-	+
Наявність механізмів налагодження процесів	+	+	+
Визначення поточного кроку виконання процесу у реальному часі	+	+	+
Можливість покращення процесів	+	+	+
Контроль ефективності працівників	+	+	+
Інтеграція з BAS	+	+	+
Інтеграція з IP-телефонією, email-сервісами	+	+	+
Наявність мобільної версії	+	+	+

Дизайнер форм перетягування. За допомогою конструкторів форм ви можете вибирати між простими та складними конструкторами. Вибираючи, враховуйте два аспекти. По-перше, що буде комфортно для кінцевих користувачів? По-друге, чи можете ви передбачити користь від цього, оскільки процеси стають складнішими?

Мобільність. Системи BPM, які не підтримуються мобільними пристроями, налаштовують вас на провал. Коли схвалення на ходу не сприяють, це ставить процеси під загрозу затримок.

Звітність та аналітика. Одним із факторів, який робить впровадження BPM набагато кращим за звичайну оцифровку, є потужні можливості аналітики та звітності. Вони мають вирішальне значення для отримання розуміння того, як працюють процеси та які вдосконалення потрібні.

Легка інтеграція. Можливість інтеграції з іншим програмним забезпеченням має вирішальне значення для найкращого програмного забезпечення BPM. Якщо вибране програмне забезпечення BPM не може інтегруватися, це означатиме багато часу та зусиль, витрачених на передачу даних туди й назад між програмами.

Адаптивний кейс-менеджмент. Не вся робота, що виконується в організаціях, чітко визначена та структурована, як процеси. Існує достатня кількість невизначеності, якою не можна керувати за допомогою традиційної методології BPM. Коли ПЗ BPM містить функції адаптивного керування справами, воно забезпечує більш комплексне рішення [4].

Управління даними та документами. Сьогодні підприємства працюють блискавично. Цілком логічно, що обсяг створених даних і документів є величезним. Хороша система BPM може ефективно керувати даними та документами, створюючи та зберігаючи дуже цінний ресурс знань.

Контроль доступу на основі ролей. Безпека даних є серйозною проблемою для організацій у всьому світі. Виберіть програмне забезпечення BPM, яке полегшує детальний контроль доступу до даних. Обмеження доступу до конфіденційних даних на основі ролей є ефективним способом гарантувати безпеку.

Співпраця. Незалежно від характеру роботи, завжди потрібна ефективна комунікація між виконавцями завдань. Для оптимальної продуктивності найкраще, щоб усі розмови про роботу відбувалися на одній уніфікованій платформі. Перемикання між додатками для спілкування хаотично і займає багато часу.

Інтеграція управління проектами. Компанії також виконують одноразові дії, які не можна класифікувати як процеси. Іноді навіть проекти

потребують упорядкування та оптимізації. Вибір програмного забезпечення BPM, яке враховує цю можливість і містить можливість ефективного керування проектами, виявиться дуже цінним.

Для оцінки та вибору ПЗ BPM можна використовувати контрольний список корисних запитань [4].

Які потреби вашої команди в BPM? Яких бізнес-цілей ви сподіваєтеся досягти?

Який бюджет затвердила ваша фінансова команда?

Який хостинг вам потрібен — локальний чи хмарний?

Якій моделі ліцензування ви віддаєте перевагу – з відкритим вихідним кодом, безкоштовним обмеженим версіям чи повністю платним?

Скільки коштує програмне забезпечення? Які приховані витрати?

Які больові моменти відчувають поточні клієнти, якщо такі є?

Яку рентабельність інвестицій отримують поточні клієнти?

Яким рівнем технологічних ноу-хау володіють кінцеві користувачі?

Чи може програмне забезпечення BPM масштабуватися?

Чи може він відповідати галузевим нормам?

Скільки навчання знадобиться команді перед використанням програмного забезпечення?

Які відгуки про підтримку продукту?

1.4 Постановка завдань дипломної роботи

Як показали проведені дослідження, існує багато напрямків і програмних продуктів для автоматизації бізнес-процесів, а рівень автоматизації залежить як від бюджету підприємства, так і від його ІТ-стратегії: організація діяльності підприємства; управління підприємством; маркетинг; фінанси підприємства; виробництво продукції; управління якістю; зовнішні комунікації [8].

Процесний підхід в управлінні організацією потребує формування та підтримки інформаційних баз даних управління бізнес-процесами [9].

Формування бізнес-моделі підприємства вимагає чіткого розуміння ключових компетенцій та бізнес-процесів компанії [10] - [11].

Сучасні BPM-технології дозволяють виконати реінжинірінг бізнес-процесів компанії або підприємства, що є важливим кроком до цифрової трансформації [12].

Вибір ПЗ BPM є складною задачею, що потребує урахування комплексу вимог, зокрема до вартості, відповідності стандартам, масштабування, підтримки хмарних сервісів, тощо.

Тому, тема роботи є актуальною.

Мета роботи – дослідження та розробка автоматизованої системи (АС) обліку електросамокатів для підвищення ефективності процесів видачі-прийняття обладнання та інвентаря та проведення фінансових розрахунків.

Завдання роботи:

- дослідити структурні та функціональні особливості платформи BAS;
- виконати аналіз вимог до автоматизованої системи обліку прокату самокатів;
- розробити структурну та функціональну схеми автоматизованої системи;
- обґрунтувати вибір інструментарію розробки;
- розробити алгоритм функціонування автоматизованої системи;
- виконати конструювання програмного забезпечення та перевірку його працездатності на платформі BAS.

2 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ ПЛАТФОРМИ BAS ДЛЯ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ

BAS – це система управління та обліку, що дає можливість організувати єдиний інформаційний простір підприємства. Підсистема управління продажами та відносинами з клієнтами забезпечує зокрема автоматизацію бізнес-процесів взаємодії з клієнтами та моніторинг їхнього проходження [13] - [14].

2.1 Структура системи BAS

Система BAS об'єднує технологічну платформу та конфігурації та може бути запущена в одному з двох режимів: «Підприємство» або «Конфігуратор» (рис. 2.1) [15].



Рисунок 2.1 – Компоненти та режими BAS [15]

Технологічна платформа – це набір різних механізмів, які використовуються для автоматизації економічної діяльності і не залежать від

конкретного законодавства та методології обліку. Конфігурації є прикладними рішеннями. Кожна конфігурація орієнтована на автоматизацію визначеної сфери економічної діяльності.

Режим «Підприємство» – це режим користувача. В цьому режимі користувачі додають, змінюють, видаляють дані, формують звіти та виконують інші прикладні задачі. Режим «Конфігуратор» – це режим розробника. У цьому режимі розробляються прикладні рішення та виконується адміністрування інформаційних баз.

Об'єкти конфігурації – це складові елементи, з яких складається будь-яке прикладне рішення. Вони є проблемно-орієнтованими об'єктами, що підтримуються на рівні технологічної платформи. Дерево об'єктів конфігурації показує все прикладне рішення у вигляді деревоподібної структури, кожна вітка якої визначає певну складову конфігурації [15].

Вікно конфігурації в режимі «Конфігуратор» дозволяє:

- створювати нові об'єкти конфігурації;
- редагувати існуючі об'єкти конфігурації;
- видаляти об'єкти конфігурації з контролем наявності посилань на об'єкт, що видаляється;
- розташовувати об'єкти конфігурації у необхідному порядку в межах «своїх» групи;
- знаходити в дереві об'єкт, дані якого на цей момент редагуються (у вікні редагування об'єкта, у редакторі форми, макета, модуля);
- сортувати об'єкти конфігурації, підпорядковані одному об'єкту конфігурації, за ім'ям, синонімом або коментарем;
- шукати посилання на об'єкт конфігурації в інших об'єктах конфігурації;
- шукати посилання на інші об'єкти конфігурації в поточному об'єкті конфігурації;
- відбирати об'єкти конфігурації, що належать будь-яким підсистемам, визначеним у конфігурації, або не належать жодній із підсистем;

- запускати конструктори, пов'язані з об'єктом конфігурації [15].

Кожен об'єкт конфігурації повинен мати ім'я, синонім та тип.

Ім'я – це ідентифікатор об'єкта у системі. Ім'я має бути унікальним та записано за певними правилами: ім'я має складатися з одного слова (не мати пробілів), починатися з літери і не містити спеціальних символів, крім «_».

Синонім – зручна для користувача назва. У написанні синоніма немає жодних обмежень. Поле синоніма заповнюється автоматично після заповнення імені та може бути відредаговано.

Редагувати об'єкти конфігурації можна за допомогою вікна редагування та палітри властивостей.

Вікно редагування об'єкта конфігурації призначене для редагування властивостей основних об'єктів конфігурації, керування складом підлеглих об'єктів та налаштування взаємодії об'єктів один з одним.

Палітра властивостей використовується для редагування об'єкта конфігурації. Вона є вікном, що містить набір властивостей об'єкта (як доступних, так і не доступних для редагування), набір посилань на пов'язані з об'єктом форми та ін. [15] - [16].

2.2 Проєктування інформаційної бази

Дані, що визначають логіку функціонування системи BAS належать до інформаційної бази (ІБ). Зберігання ІБ здійснюється в базі даних у вигляді набору таблиць. Можливо використовувати одну з чотирьох систем управління базами даних (СУБД):

- вбудовану СУБД (файловий варіант інформаційної бази). У цьому випадку всі дані бази даних зберігаються у файлі з ім'ям 1CV8.1CD. Цей файл має двійковий формат і є базою даних для вбудованої СУБД;

- Microsoft SQL Server (клієнт-серверний варіант інформаційної бази). Усі дані інформаційної бази зберігаються у базі даних Microsoft SQL Server;

- PostgreSQL (клієнт-серверний варіант інформаційної бази). Усі дані інформаційної бази зберігаються у базі даних PostgreSQL;
- IBM DB2 (клієнт-серверний варіант інформаційної бази). Усі дані бази даних зберігаються в базі даних IBM DB2 [16].

На рівні об'єктів бази даних (таблиць, полів, індексів тощо) як файловий, так і клієнт-серверний варіант ІБ мають подібний формат.

Список і структура таблиць бази даних визначається конкретною конфігурацією, зокрема, визначеними у ній об'єктами метаданих. Ім'я кожної таблиці складається з буквеного префікса та наступного за ним номера. Префікс визначає призначення таблиці, а номер дозволяє розрізняти таблиці однакового призначення, що стосуються різних об'єктів метаданих. Якщо як СУБД використовується IBM DB2, то описану структуру мають імена таблиць, які псевдоніми [17].

Для створення ІБ використовується режим «Підприємство» (рис. 2.2).

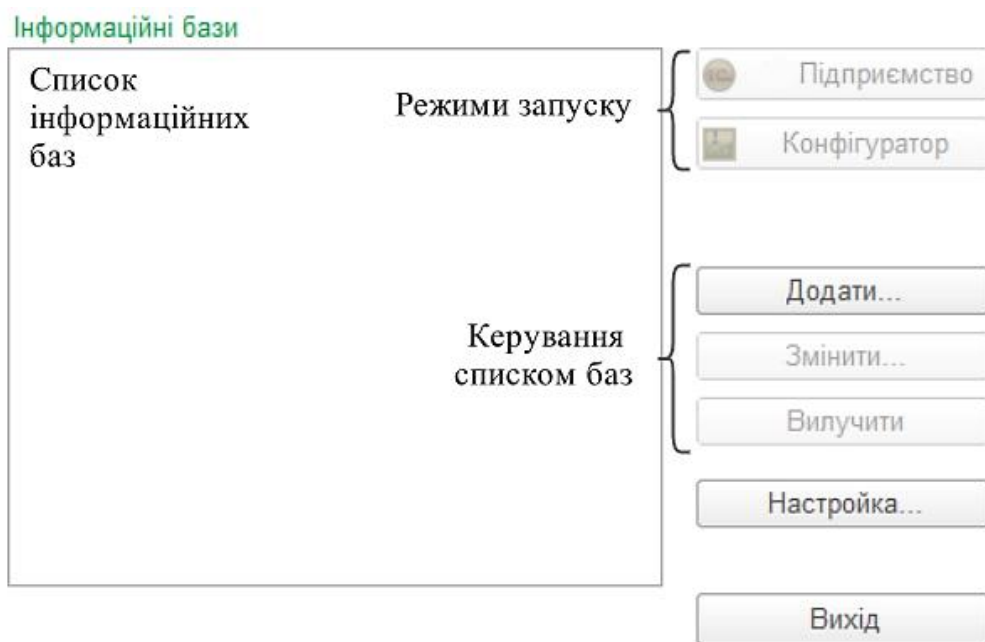


Рисунок 2.2 – Структура вікна ІБ [17]

Можливо створювати нову ІБ на основі шаблону або абсолютно порожню. ІБ привласнюється ім'я, а також вказується каталог для її встановлення. Параметри ІБ за замовчуванням показано на рисунку 2.3 [17].

Будь-яка ІБ може бути запущена у двох режимах: «Підприємство» та «Конфігуратор». Для запуску ІБ в режимі «Підприємство» вона обирається зі списку баз. Далі система автоматично формує для користувача форму, а також додає деякі додаткові функції, наприклад, калькулятор та календар. При запуску ІБ в режимі «Конфігуратор» відкриється режим розробника, в якому і проводитиметься конфігурація (налаштування) ІБ.

Рисунок 2.3 – Параметри ІБ за замовчуванням [17]

Для розробки ІБ використовуються спеціальні об'єкти, що представлені у вікні конфігурації, що зображено на рисунку 2.4.

У вікні конфігурації представлено її дерево, що складається із різних об'єктів конфігурації, кожен з яких виконує певну функцію.

Існує кілька способів додавання/створення нового об'єкта конфігурації. Вилучити об'єкт конфігурації можна також кількома способами і якщо об'єкт не пов'язаний з іншими об'єктами посиланнями, то його буде вилучено [17].

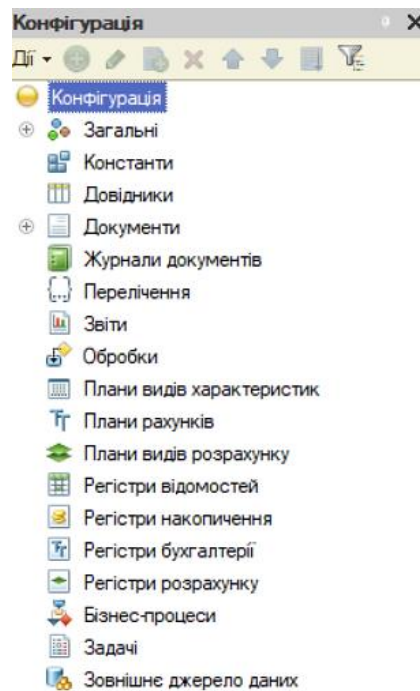


Рисунок 2.4 – Вікно конфігурації ІБ [17]

Редагування об'єктів конфігурації може здійснюватися у вікні редагування об'єкта конфігурації, яке зображено на рисунку 2.5. Для налаштування того чи іншого об'єкта необхідно переключатися між вкладками. Для будь-якого об'єкта конфігурації можна відкрити палітру властивостей, яка зображена на рисунку 2.6. Палітра властивостей може відображатися списком чи закладками.

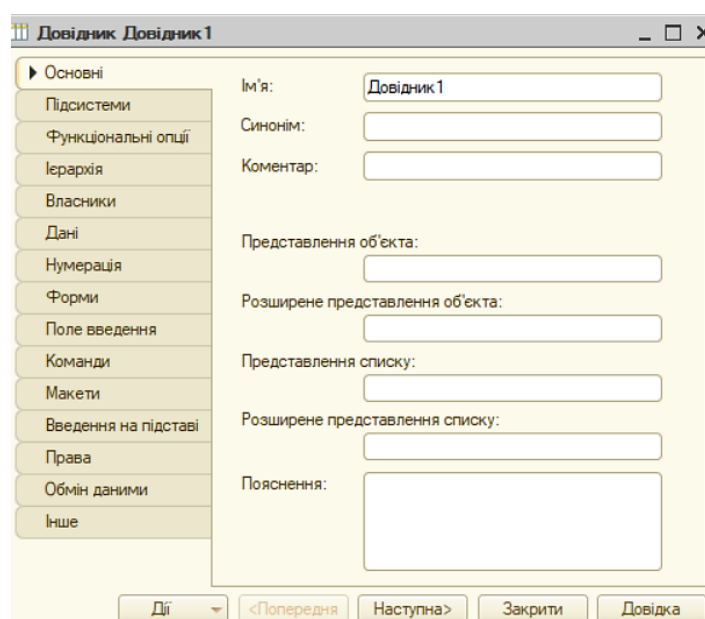


Рисунок 2.5 – Вікно редагування об'єкта конфігурації ІБ [17]

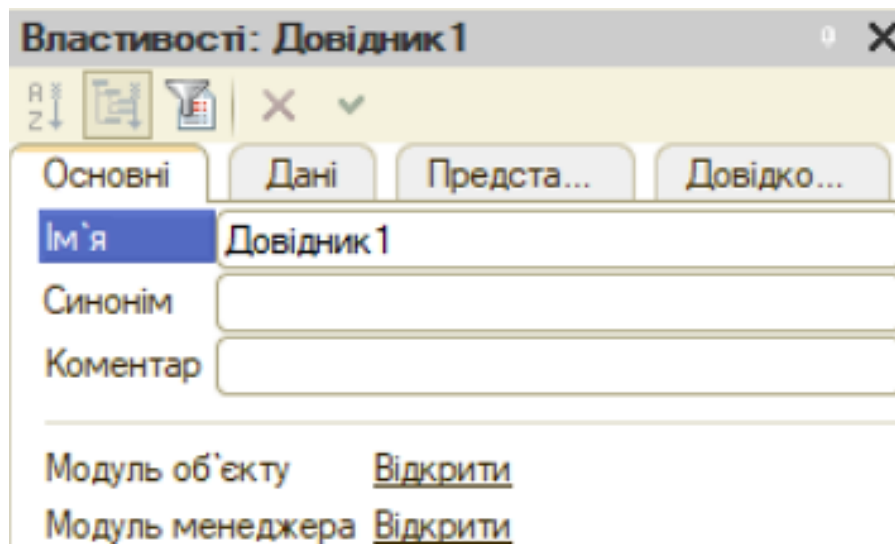


Рисунок 2.6 – Палітра властивостей об'єкта конфігурації [17]

Налаштування у вікні редагування об'єкта конфігурації та на палітрі властивостей взаємопов'язані. Наприклад, якщо змінити ім'я на палітрі властивостей, воно обов'язково зміниться у вікні редагування об'єкта конфігурації.

Деякі об'єкти (наприклад, константи) не мають вікна редагування конфігурації. Усі налаштування здійснюються виключно на палітрі властивостей.

Кожен об'єкт конфігурації повинен мати ім'я, синонім та тип. Якщо хочемо дати константі ім'я «Дата створення організації», то правильний запис буде виглядати так: «ДатаСтворенняОрганізації» або «Дата_створення_організації».

Крім імені та синоніма, необхідно звернути увагу на тип даних, який ця константа буде зберігати. За замовчуванням тип завжди заповнений одним із допустимих значень. Наприклад, константа за замовчуванням завжди має рядковий тип із довжиною 10 символів. Тип даних може бути змінено.

У разі потреби ІБ може бути вивантажена у файл. Завантаження інформаційної бази відбувається аналогічно. Потрібно вибрати у меню «Адміністрування» пункт «Завантажити інформаційну базу», а далі обрати файл з розширенням *.dt [17].

2.3 Опис об'єктів конфігурації

Типи об'єктів конфігурації наведено в таблиці 2.1.

Таблиця 2.1 – Опис об'єктів конфігурації [17]

Тип об'єкта	Призначення	Опис
Довідники	Дозволяють зберігати в ІБ дані, що мають однакову структуру та характер списку.	Елемент довідника характеризується кодом та найменуванням, а також може містити додаткову інформацію.
Звіти	Призначені для обробки накопиченої інформації та отримання зведених даних у зручному для перегляду та аналізу вигляді.	Вихідні дані, які використовує звіт: набори даних, зв'язки між наборами даних, обчислювані поля, ресурси тощо.
Документи	Зберігають інформацію про будь-які бізнес-процеси.	Містить номер, дату, час, додаткову інформацію, яка детально описує цей документ.
Регістри накопичення	Є основою механізму обліку руху коштів (фінансів, товарів, матеріалів тощо), що дозволяє автоматизувати такі напрями, як складський облік, взаєморозрахунки, планування.	Містить запис, кожен з яких містить значення вимірів та відповідні їм значення ресурсів.

Всі створені об'єкти конфігурації є таблицями бази даних. У режимі «BAS: Підприємство» виконується заповнення цих таблиць даними. Щоб отримати ці дані для відображення у звіті, потрібно сформулювати запит до бази даних, для чого скористуватися конструктором запиту, який надає три функціональні частини: всі об'єкти конфігурації, які є в базі даних; вибрані об'єкти, звідки хочемо отримувати дані для конкретного звіту; значення (поля), які хочемо побачити у звіті.

Довідники - це прикладні об'єкти конфігурації. Вони дозволяють зберігати в ІБ дані, що мають однакову структуру та характер списку. Це може бути, наприклад, список співробітників, перелік товарів, список постачальників чи покупців. Кожен елемент довідника характеризується кодом та найменуванням. Крім коду та найменування, кожен елемент довідника зазвичай містить деяку додаткову інформацію, яка докладно описує цей елемент. Наприклад, для товару це може бути інформація про артикул, упаковку тощо. Крім цього, кожен елемент довідника може містити деякий набір інформації, яка однакова за своєю структурою, але різна за кількістю, для різних елементів довідника [16].

Наприклад, для кожного співробітника у довіднику «Фізичні особи» це може бути контактна інформація або інформація про склад сім'ї, освіту. Для зберігання таких даних використовують табличні частини довідника. Довідники можуть підтримувати ієрархічне розташування елементів. Наприклад, у довіднику «Номенклатура» можуть бути створені групи: «Побутова техніка», «Взуття», «Продукти» тощо.

Звіти – це прикладні об'єкти конфігурації. Вони призначені для обробки накопиченої інформації та отримання зведених даних у зручному для перегляду та аналізу вигляді. Як правило, для формування вихідних даних звіт використовує систему компонування даних. Об'єкт конфігурації «Звіт» реалізує властивість «Основна схема компонування даних». Конструктор схеми компонування даних дозволяє описати вихідні дані, які

використовуватиме звіт: набори даних, зв'язки між наборами даних, обчислювані поля, ресурси тощо [16].

У системі «Підприємство» довідники можуть бути певним чином структуровані. Близькі за змістом (у тому чи іншому розумінні) елементи можна поєднувати в одну групу, щоб не плутати їх із елементами іншого змісту. Як приклад можна навести довідник «Номенклатура», в якому можуть бути створені групи «Товари» та «Послуги». Для цього використовується ієрархія. Якщо для довідника є необхідність розділити елементи за групами, треба використовувати вкладку «Ієрархія» та «Ієрархічний довідник» (рис. 2.7). Вид ієрархії «Ієрархія груп та елементів» дозволить створювати окремо групи елементів та самі елементи. Тепер, крім елементів довідника, можна створювати групи. В режимі «Підприємство» можливо створити кілька груп (рис. 2.8). Тепер можливо створити елемент та додати його до певної групи, вказавши її у полі «Батьківський елемент» [16]

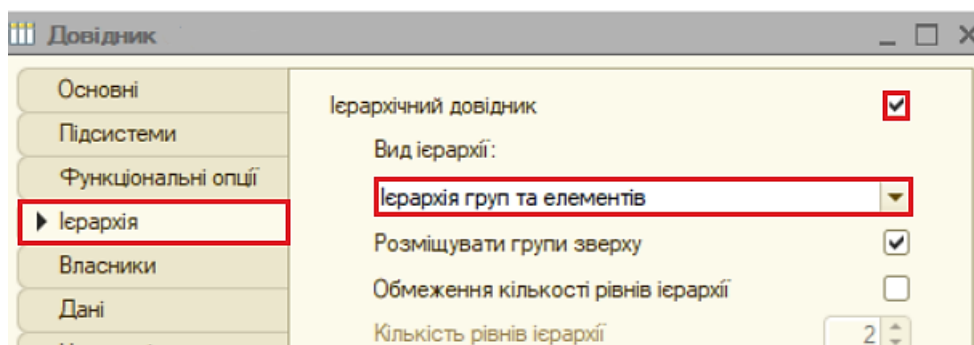


Рисунок 2.7 – Вкладка «Ієрархія» [17]

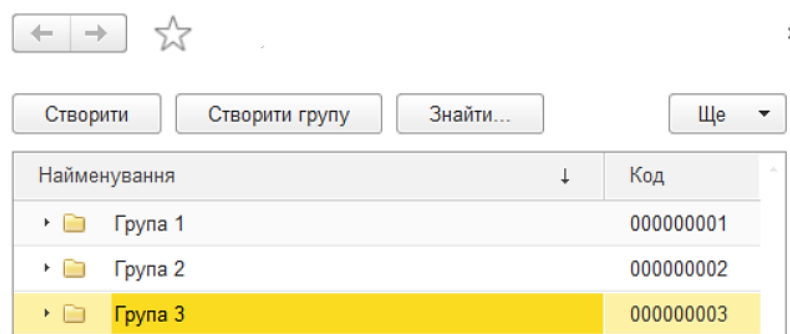


Рисунок 2.8 – Список створених груп [17]

Стандартні реквізити платформа створює автоматично, виходячи з властивостей конкретного об'єкта конфігурації, але їх можна змінити. Можна додати синонім для зміни назв цих полів для користувача [16].

Документи - це об'єкти конфігурації, що зберігають інформацію про бізнес-процеси. Наприклад, за допомогою документа можна зареєструвати (тобто зберегти дані документа для подальшого опрацювання) продаж товарів або нарахування зарплати. Кожен документ характеризується номером, датою та часом. Крім номера, дати та часу, кожен документ, як правило, містить деяку додаткову інформацію, яка детально описує цей документ. Наприклад, для документа «Надходження товарів та послуг» це може бути інформація про постачальника товарів (контрагента), склад, на який приходять товари тощо. Крім цього, кожен документ може містити деякий набір інформації, яка однакова за своєю структурою, але різна за кількістю, для різних документів. Наприклад, для документа «Надходження товарів та послуг» це може бути інформація про товари, що надійшли на підприємство (найменування, кількість, і т.д.), серійні номери та зворотну тару. Для зберігання таких даних є табличні частини документа [17].

Будь-який документ може бути в одному з двох станів: підготовлений до виконання або виконаний:

- щоб підготувати документ для використання у майбутньому, потрібно його записати;
- щоб позначити документ як виконаний – провести.

Важливою властивістю документа є можливість проведення. Якщо документ проводиться, він може змінити стан тих чи інших даних. Якщо ж документ не є документом, що проводиться, це означає, що подія, яку він відображає, не впливає на стан обліку, який ведеться в цьому прикладному рішенні [15].

Регістри накопичення – це прикладні об'єкти конфігурації. Вони є основою механізму обліку руху коштів (фінансів, товарів, матеріалів тощо), що дозволяє автоматизувати такі напрями, як складський облік,

взаєморозрахунки, планування. Регістр накопичення утворює багатовимірну систему вимірів і дозволяє «накопичувати» числові дані в розрізі декількох вимірів. Інформація в реєстрі накопичення зберігається у вигляді записів, кожен з яких містить значення вимірів та відповідні їм значення ресурсів. Виміри реєстру описують розрізи, у яких зберігається інформація, а в ресурсах реєстру накопичуються необхідні числові дані. Оскільки реєстр накопичення слугує для накопичення числових значень, кожен запис виконує зміну ресурсів, що зберігаються – рух.

Рухи можуть або додавати деякий приріст до збережених ресурсів, або забирати його. Якщо має виконуватися збільшення ресурсів, що зберігаються, – такий рух називається рухом прибутку («+»), якщо зменшення ресурсів, що зберігаються, – рухом видатку («-»). Разом з кожним записом, що знаходиться в реєстрі накопичення, можна зберігати додаткову інформацію. Для цього є реквізити реєстру накопичення. Зміна стану реєстру накопичення відбувається, зазвичай, під час проведення документа. Тому кожен запис реєстру пов'язаний із певним документом-реєстратором. Існує два види реєстрів накопичення: реєстри накопичення залишків та реєстри накопичення оборотів. Регістр накопичення залишків дозволяє зберігати як підсумкові значення ресурсів – залишки, так і зміни цих ресурсів – обороти [17].

2.4 Висновки до розділу 2

В даному розділі виконано дослідження структурних та функціональних особливостей системи BAS та наведено опис етапів розробки ІБ для системи автоматизації бізнес-процесів та об'єктів конфігурації. Показано, що її вибір в якості платформи розробки є обґрунтованим через поширене використання в Україні.

3 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ПРОКАТУ САМОКАТІВ

3.1 Аналіз вимог до автоматизованої системи

Автоматизована система (АС) призначена для обліку бізнес-процесу прокату електросамокатів, що включає етапи закупівлі електросамокатів, здачі самокатів в оренду студентам студмістечка, а також повернення самокату студентом.

Функціональні вимоги:

- фіксація кількості придбаних для пункту прокату електросамокатів та їх марок;
- розрахунок вартості послуги в залежності від терміну прокату, марки самокату, типу захисного спорядження;
- надання можливості вибору комплектів захисту (Базовий, Стандарт). Базовий: шолом (безкоштовно). Стандарт: шолом, нарукавники, наколінники (+30 грн. до підсумкової суми прокату). При оформленні прокату автоматично додавати до самокату базовий комплект захисту, але забезпечити можливість змінити його на Стандарт та перерахувати вартість прокату;
- фіксація даних щодо студента (ПІБ, телефон та група), який взяв в оренду або повернув самокат;
- фіксація доступних на базі залишків електросамокатів та інформації про їх марки;
- фіксація граничної дати повернення самоката студентом на базу;
- формування документа під підпис з перерахуванням отриманої техніки, вартості послуг та прийнятих під заставу документів (рис.3.1);
- формування звіту, який повинен містити результуючу інформацію про самокати, кількість годин прокату та підсумкову суму виручки за вказаний період часу.

Видав: ТОВ Прокат №1 (Головний корпус, тел. 0XX-XXX-XX-XX)				
Отримав: Котигорошко Іван Іванович (Група КНТ-XXX, моб. XXX-XXX-XX-XX)				
Документ залога: Студентський, № XXXXXXX від 01.09.2023				
Дата повернення: XX.XX.2023 19:00:00				
№	Об'єкти прокату	Кількість	Ціна	Сума
1	Комлект захисту (шолом, нарукавники, наколінники)	1,00	30,00	30,00
2	Самокат Matomit 09ZX4	1,00	30,00	150,00
Усього:				180,00
Студент, Котигорошко Іван Іванович (Група КНТ-XXX моб. XXX-XXX-XX-XX) має повернути перераховані вище об'єкти прокату в тому самому стані, в якому їх було видано, у випадку пошкодження буде застосовано штрафні санкції.				
XX.XX.2023 (Дата)				
(Підпись)				

Рисунок 3.1 – Шаблон документу

Нефункціональні вимоги:

- видача і повернення повинні фіксуватися окремо, причому кожен студент може орендувати (і, відповідно, здати) більше одного самоката;
- якщо гранична дата більше або дорівнює поточній даті, підсвічувати рядок червоним кольором;
- вартість послуг може змінюватися один раз на день;
- зберігати інформацію про заставу (тип документа, серія, номер);
- формувати документ-згоду про обробку персональних даних;
- після повернення самокату рахувати реальну кількість годин: якщо час не сплив (кратність 1 година), то здійснювати повернення коштів, якщо час перевищує заданий, розрахувати автоматично суму доплати.

Вимоги до апаратної платформи:

- процесор Intel Core i5 або AMD Ryzen 5 та наступні моделі;
- оперативна пам'ять 4 Гбайт та вище (рекомендується 8 Гбайт);
- жорсткий диск або твердотільний накопичувач 40Гб та вище;
- відеокартка, що підтримує монітор с мінімальною роздільною здатністю 1920x1080 точок.

3.2 Розробка структури АС

При розробці АС будуть використовуватись такі об'єкти конфігурації, як довідники, документи, реєстри накопичення та звіти.

Модульна структура АС містить модулі двох типів – модулі об'єктів, що відповідають за обробку інформації, що міститься в довідниках та модулі форм документа, що відповідають за формування звітів (рис. 3.2).

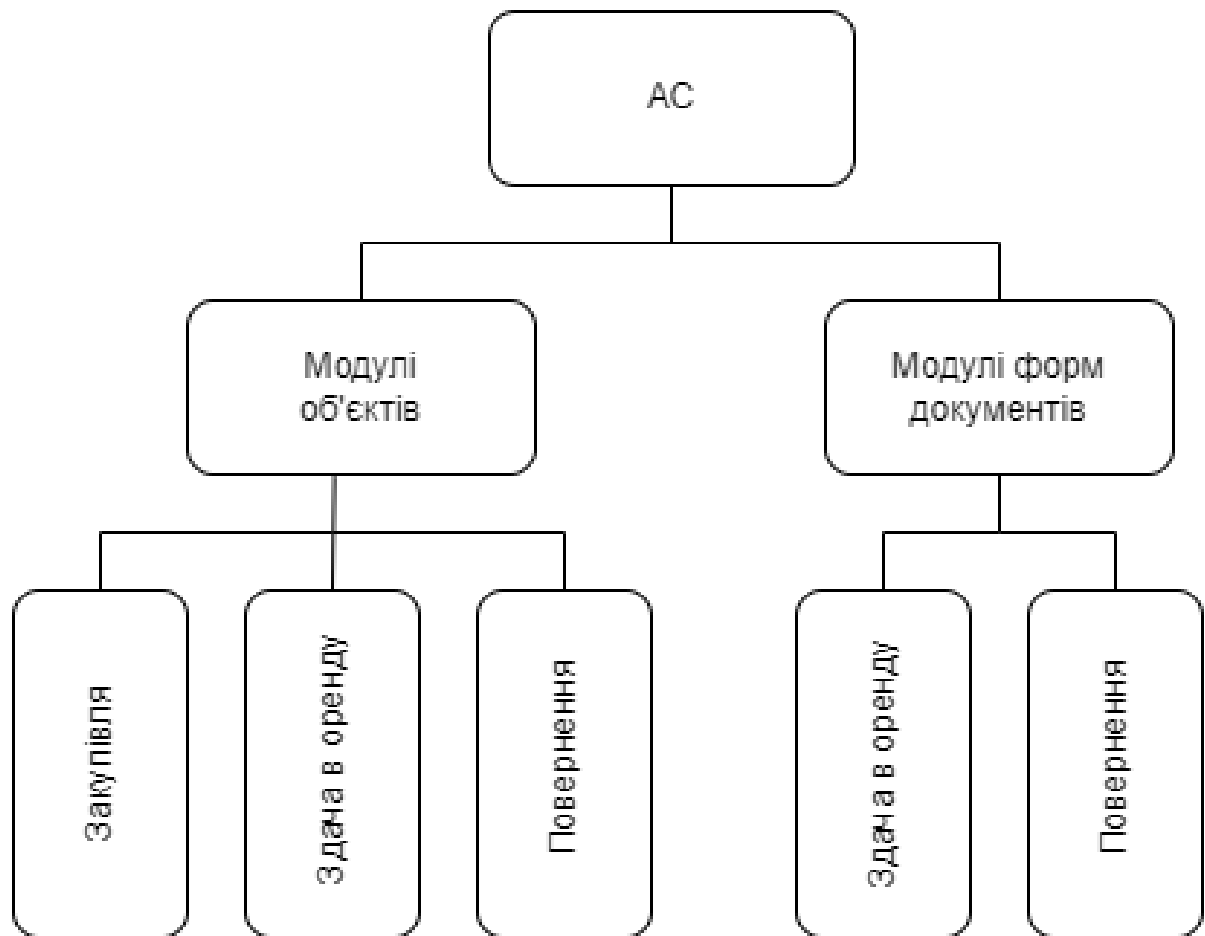


Рисунок 3.2 – Структурна схема АС

Функціональну схему АС наведено на рисунку 3.3. Вона показує взаємозв'язок документів і реєстрів накопичення.

При проведенні документа «Закупівля» в реєстр накопичення «ЗалишкиСамокатів» буде передано ту кількість самокатів, яка буде вказана в документі. При проведенні документа «ЗдачаВОренду» проводитиметься віднімання одного самокату з реєстру накопичення «ЗалишкиСамокатів», але

при цьому студент, який узяв самокат, буде внесений до реєстру «ОрендаСамокатів» як орендар. Тобто документ «ЗдачаВОренду» забезпечує рухи в обидва реєстри накопичення.

Документ «Повернення» робитиме все навпаки: додаватиме одиницю до «ЗалишкиСамокатів» та прибиратиме студента зі списку боржників з реквізиту «ОрендаСамокатів».

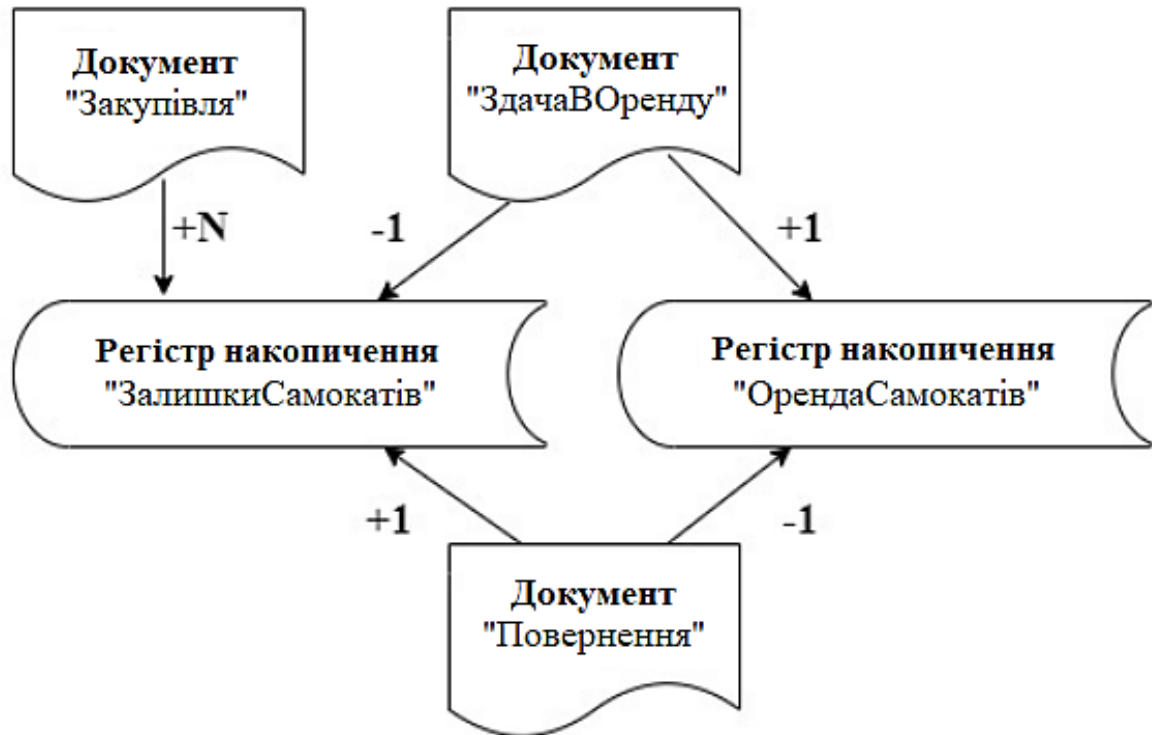


Рисунок 3.3 – Функціональна схема АС

Таким чином, кількість електросамокатів у реєстрі накопичення «ЗалишкиСамокатів» має відповідати поточній кількості самокатів на складі. А реєстр накопичення «ОрендаСамокатів» має містити список боржників.

Щоб реєстри накопичення спрацювали, потрібно зробити наступне:

- визначити, які дані потраплятимуть до кожного реєстру (визначити документи-реєстратори);
- описати, яким чином дані з кожного документа-реєстратора повинні потрапляти в реєстри.

Якщо все було зроблено правильно, імена та типи реквізитів збігаються, а реквізити реєстру заповнюються автоматично.

3.3 Вибір інструментарію розробки

3.3.1 Вбудована мова програмування

Вбудована мова BAS має багато спільних рис з такими мовами, як Pascal, Java Script, Basic, але не є прямим аналогом будь-якої з цих мов. Прикладні рішення в "BAS: Підприємство" не програмуються (кодуються) повністю, більша їх частина описується параметрично, тобто у вигляді структур метаданих, за допомогою дизайнера форм, звітів і т.д.

Відповідно, вбудована мова BAS є скриптовою мовою, призначеною насамперед для програмування бізнес-логіки в контексті об'єктної моделі. На ній програмуються обробники різних подій, що змінюють стан об'єктів системи, наприклад обробники команд користувача, обробники проведення документів та інші.

Особливості побудови мови безпосередньо відповідають моделі проєктування структур даних, реалізованій у BAS. Те, що вся розробка конфігурації ведеться на основі використання стандартних об'єктів системи, дозволяє розробнику прикладних рішень застосовувати відповідні об'єкти вбудованої мови, що мають великий набір функцій і високу гнучкість.

Технологічні особливості вбудованої мови:

- попередня компіляція, тобто перед виконанням модулі перетворюються на внутрішній код;
- кешування скомпільованих модулів у пам'яті;
- м'яка типізація - тип змінних може змінюватися в процесі роботи;
- відсутність програмного опису об'єктів конфігурації, тобто прикладне рішення може маніпулювати або вбудованими в платформу об'єктами, або об'єктами, описаними розробником у процесі візуального конструювання системи у вигляді метаданих об'єктів [17].

3.3.2 Мова запитів

Запити у системі BAS призначені для вибірки інформації з бази даних. Запит дозволяє вибрати певну інформацію з ІБ та виконати певну обробку: згрупувати, відсортувати, обчислити. Мова запитів BAS заснована на SQL, але при цьому містить значну кількість розширень, орієнтованих на відображення специфіки фінансово-економічних завдань та на максимальне скорочення зусиль з розробки бізнес-застосунків.

Прямий доступ до фізичних таблиць, в яких зберігається інформація в БД отримати не можна. Це пов'язане з тим, що у прикладному рішенні можуть використовуватись різні СУБД, що мають свою специфіку, а текст запиту повинен бути універсальним і однаково працювати на будь-якій СУБД. Тому при виконанні запиту платформа автоматично трансліює текст запиту на набір інструкцій, які "розуміє" конкретна СУБД. Крім того, фізичні таблиці та поля в них мають техногенні імена, з яких незрозуміло, що саме зберігається у цьому полі. Тому за допомогою запитів звертаємося до даних не безпосередньо, а через спеціальний «прошарок» у вигляді таблиць мови запитів.

Схему доступу до даних в системі наведено на рисунку 3.4.

Усі таблиці, до яких можна звернутись за допомогою мови запитів, є уявними, що певною мірою відповідають реальним фізичним таблицям БД. За рівнем схожості на фізичні таблиці їх прийнято розділяти на реальні та віртуальні таблиці.

Реальні таблиці містять дані будь-якої однієї фізичної таблиці, що зберігається у БД, і дуже схожі на свою фізичну таблицю. Реальні таблиці поділяються на об'єктні та необ'єктні. В об'єктних таблицях представлена інформація типів даних посилань (довідники, документи, і т. і.). В необ'єктних – всіх інших типів даних (константи, реєстри тощо. буд.).

Відмінною рисою об'єктних таблиць і те, що вони містять у собі стандартне поле Посилання, яке дозволяє однозначно ідентифікувати кожен

запис (дані про об'єкт БД). Ці таблиці можуть бути ієрархічними, підлеглими, та поля таких таблиць можуть містити вкладені таблиці (табличні частини).

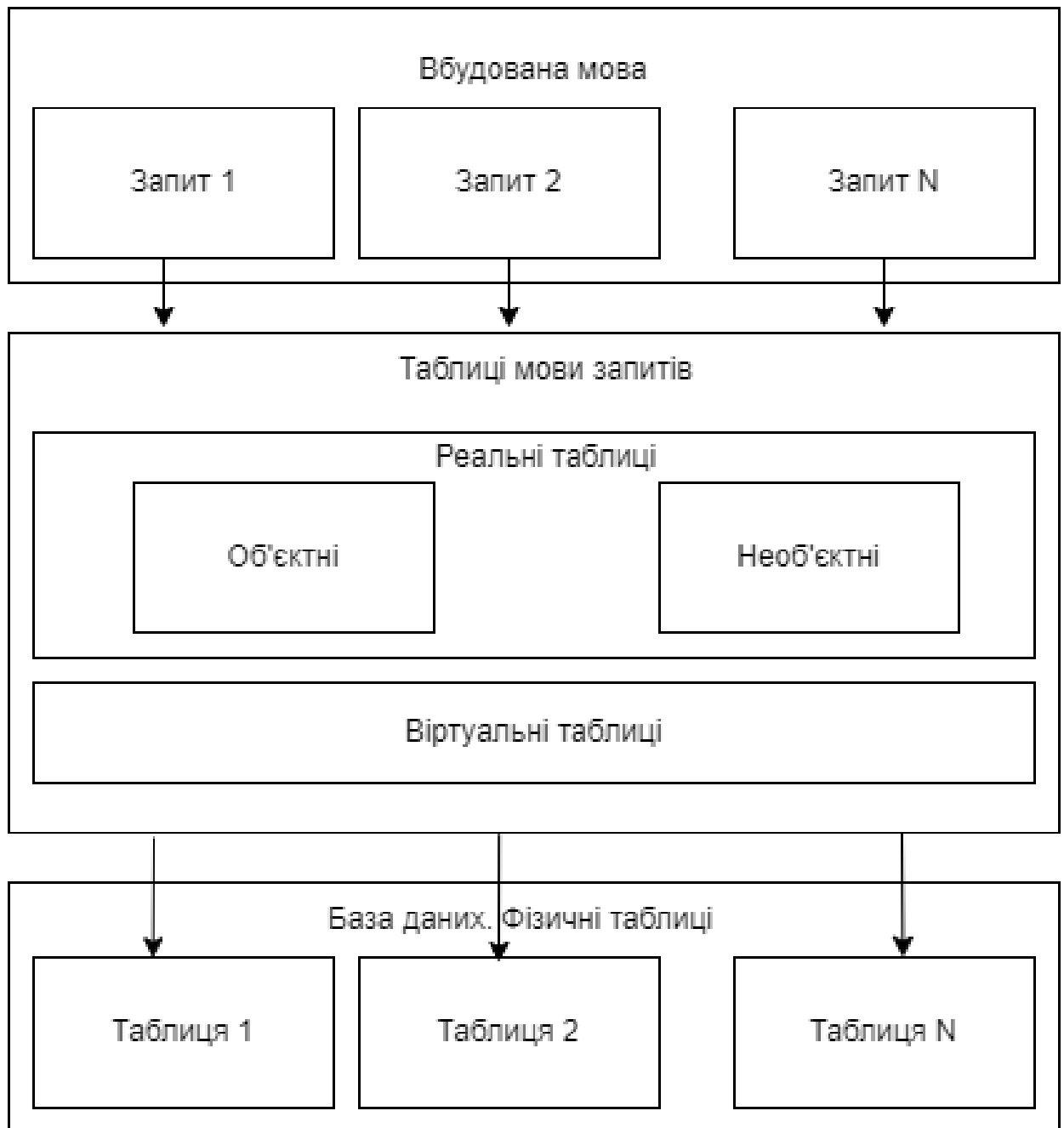


Рисунок 3.4 – Схема доступу до даних

Віртуальні таблиці формуються на момент виконання запиту з урахуванням реальних таблиць БД. Віртуальні таблиці мало схожі на якусь фізичну таблицю та містять зовсім інший склад записів, ніж справжні таблиці.

Основні можливості:

- розіменування полів посилань, тобто. звернення до полів через точку ("."). Якщо поля будь-якої таблиці мають тип посилань і зберігають посилання на об'єкти іншої таблиці, розробник може в запиті посилатися на них через точку, при цьому, що найважливіше, кількість рівнів вкладеності таких посилань система не обмежує;

- звернення до вкладених таблиць (табличних частин документів та елементів довідників). Система підтримує звернення до вкладених табличних частин як до окремих таблиць і як до цілих полів однієї таблиці. Наприклад, при зверненні до видаткової накладної (документу, що містить табличну частину зі складом товарів, що відвантажуються) можливо вважати табличну частину як окрему таблицю (при цьому отримати записи з різних об'єктів), а також можливо вважати заголовний запис видаткової накладної, в якій значенням поля будуть всі записи вкладеної таблиці, підпорядковані цьому об'єкту;

- автоматичне впорядкування. Для вибору найбільш правильного ("природного") порядку виведення інформації на екран або звіт розробнику в більшості випадків достатньо задати режим автоматичного впорядкування.

- багатомірне та багаторівневе формування підсумків. Підсумки формуються з урахуванням угруповання та ієрархії, обхід рівнів може виконуватися в довільному порядку з підбиттям підсумків, забезпечується коректна побудова підсумків за часовими вимірами;

- підтримка віртуальних таблиць. Віртуальні таблиці, що надаються системою, дозволяють отримати практично готові дані для більшості бізнес-процесів без побудови складних запитів. Наприклад, така віртуальна таблиця може надати дані щодо залишків товарів у розрізі періодів на якийсь момент часу. При цьому віртуальні таблиці максимально використовують інформацію, що зберігається, наприклад, раніше розраховані підсумки і т.д.

- у мові запитів підтримуються стандартні для SQL операції, такі як об'єднання (Union), з'єднання (Join) і т.і.;

- мова запитів дозволяє використовувати у запитах тимчасові таблиці, що дозволяє підвищити продуктивність запитів, а у деяких випадках знизити кількість блокувань і зробити текст запиту легшим для сприйняття;

- для зручнішої роботи з тимчасовими таблицями в мові запитів підтримується робота з пакетними запитами, таким чином, створення тимчасової таблиці та її використання розміщуються в одному запиті;

Таким чином, вбудована мова запитів надає широкі можливості для вибірки та маніпулювання інформацією, яка зберігається в базі даних [17].

3.3.3 Система компоновки даних

Система компоновання даних (СКД) призначена для отримання та маніпуляції інформацією, що має складну структуру, з можливістю її подальшого представлення у вигляді довільного набору таблиць і діаграм. СКД має складну структуру і може містити довільний набір таблиць та діаграм. Застосування механізму СКД дозволяє дати цілісний опис звіту, використовуючи, насамперед, лише візуальні засоби, та реалізувати такі можливості:

- використання автоматично генерованих форм перегляду та налаштування звіту;

- розбиття виконання звіту на етапи;

- виконання окремих етапів побудови звіту на різних комп'ютерах;

- незалежне використання окремих частин СКД;

- програмне управління процесом виконання звіту.

СКД інтегрована до об'єкта конфігурації Звіт. Структуру СКД наведено на рисунку 3.5. СКД надає зручний конструктор схеми компоновання даних, який дозволяє вказати джерела даних, зв'язки між ними, поля та ресурси, що обчислюються, задати налаштування відображення інформації у звіті.

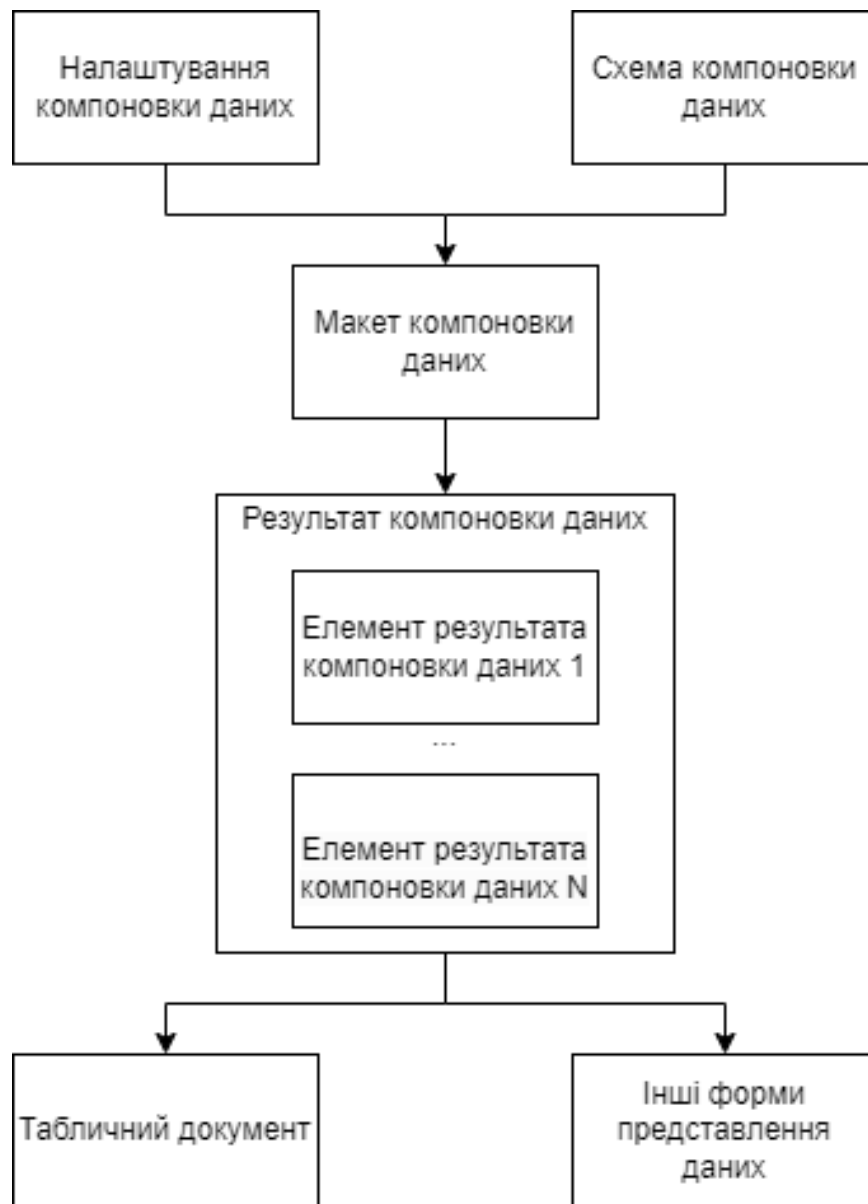


Рисунок 3.5 – Структура СКД

Схема компоновання даних визначає суть даних, що надаються звітом (звідки отримувати дані і як можна управляти компонованням даних) та є базою, на основі якої можуть бути сформовані потрібні звіти. Схема компоновання даних може містити:

- текст запиту з інструкціями системи компоновання даних;
- опис декількох наборів даних;
- докладний опис доступних полів;
- опис зв'язків між декількома наборами даних;
- опис параметрів отримання даних;

- опис макетів полів та угруповань. Налаштування компонування даних описують все, що може налаштувати розробник або користувач у певній схемі компонування даних. Налаштування компонування даних можуть містити:

- відбір;
- упорядкування;
- умовне оформлення;
- структуру звіту (складові частини майбутнього звіту);
- параметри отримання даних;
- параметри виведення даних.

Макет компонування даних є вже готовим описом того, як повинен бути сформований звіт. У ньому з'єднується схема компонування та налаштування компонування. Фактично макет компонування даних є результатом застосування конкретних налаштувань до схеми компонування і є готовим завданням процесору компонування на формування звіту потрібної структури з урахуванням конкретних налаштувань.

Результат компонування даних є набором елементів результату компонування даних. Як самостійна логічна сутність результат компонування даних не існує, існують лише його елементи. Елементи результату компонування даних можна вивести в табличний документ для представлення їх користувачеві, або в інші види документів [17].

3.4 Висновки до розділу 3

В даному розділі відповідно до мети та завдань роботи було виконано основні етапи проектування АС обліку прокату самокатів. Проаналізовано функціональні та нефункціональні вимоги до АС, обгрунтовано вибір вбудованої мови, мови запитів системи BAS, а також системи компоновки даних.

4 КОНСТРУЮВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ПРОКАТУ САМОКАТІВ

4.1 Схеми алгоритму функціонування АС

Схему алгоритму функціонування АС наведено на рисунку 4.1.

Етап 1. Початок. Аналіз бізнес-процесу.

Етап 2. Створення нової інформаційної бази.

Етап 3. Розробка довідників, в яких будуть зберігатися дані щодо об'єктів та суб'єктів пункту прокату електросамокатів:

- студенти (ПІБ, група, телефон, дані документу застави);
- електросамокати (марка, кількість);
- захисне спорядження (шолом, наколінники, тощо).

Етап 4. Створення документів, що дозволять реалізувати закупівлю електросамокатів (номер документу, дата, кількість самокатів); надавати самокати та захисне спорядження в оренду (дата, час, кількість самокатів, вид спорядження, відомості про орендаря та документ застави); приймати після експлуатації видані об'єкти.

Етап 5. Створення реєстрів накопичення, які дозволять контролювати залишки самокатів на базі

Етап 6. Перевірка працездатності. Полягає у визначенні документів-реєстраторів та руху даних з них, а також перевірці роботи реєстрів в системі.

Етап 7. Результати задовільні? Якщо так, то Етап 8, якщо ні, то повернення на Етап 3 для внесення змін в довідники, документи, реєстри.

Етап 8. Формування звітів по орендарях, залишках самокатів на бізі, фінансових розрахунках, неповерненню самокатів у заданий термін, тощо.

Етап 8. Кінець.



Рисунок 4.1 – Схема алгоритму функціонування АС

Виконання запиту на вбудованій мові складається з наступних етапів:

- створення об'єкта Запит із потрібним текстом запиту мовою запитів;
- встановлення налаштувань запиту за допомогою методу ВстановитиПараметр;
- виконання запиту.

Далі можливі два варіанти. Варіант 1:

- отримання вибірки з результату запиту;
- обхід вибірки та обробка даних, тобто виконання дій, для яких був потрібен запит, наприклад, виведення області при формуванні табличного документа.

Варіант 2 (рис. 4.2):

- вивантаження результату в таблицю значень чи дерево значень.
- обробка даних таблиці/дерева значень (наприклад, перебір рядків).
-

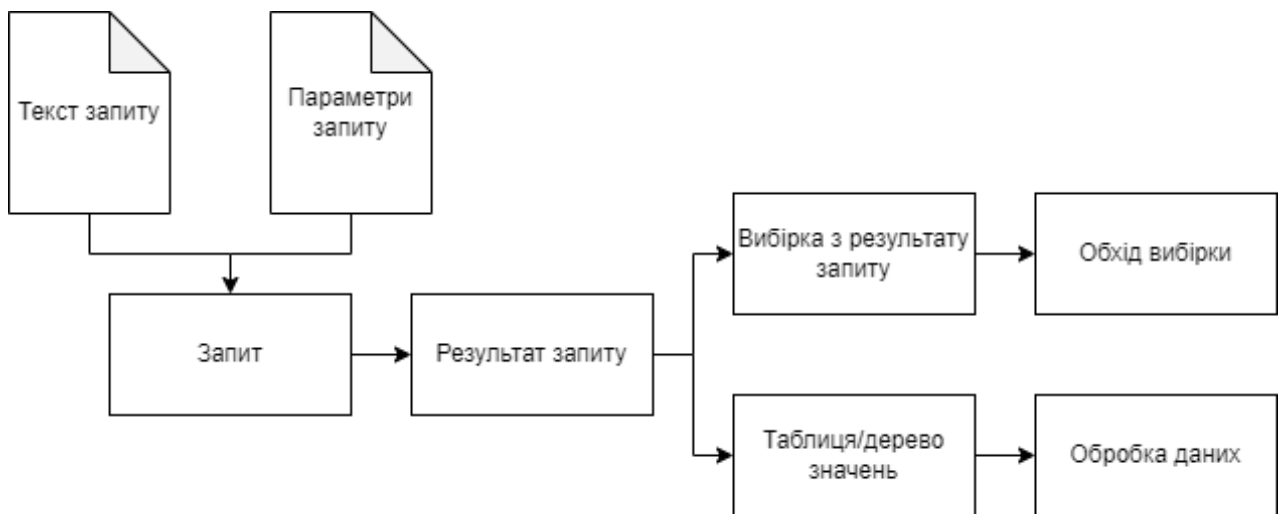


Рисунок 4.2 – Схема виконання запиту

Для формування та виконання запиту, а також для отримання та обробки його результатів у вбудованій мові використано такі програмні об'єкти: Запит, РезультатЗапиту, ВибіркаРезультатуЗапиту.

4.2 Розробка АС обліку прокату самокатів

4.2.1 Реалізація базового функціоналу АС

На основі створеної нової ІБ в режимі «Конфігуратор» створено Довідник «Студенти» (рис. 4.3), в якому буде зберігатися список студентів, які можуть орендувати електросамокат в студентському містечку.

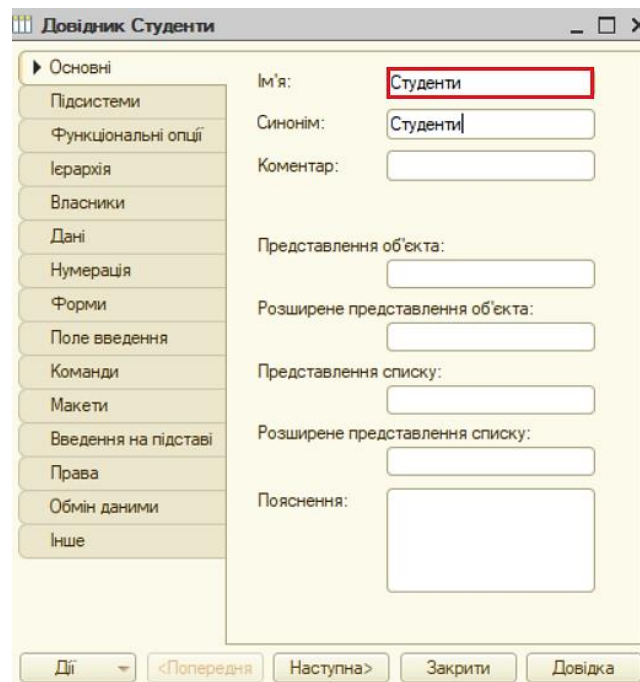


Рисунок 4.3 – Відкрита нова ІБ в режимі «Конфігуратор» та створення довідника «Студенти»

Далі потрібно реалізувати закупівлю електросамокатів для пункту прокату. Для цього використано об'єкт конфігурації «Документ» та створено документ «Закупівля» для збереження інформації щодо самокатів (рис. 4.4). Всі самокати в прокаті можуть бути однакові, тому зберігати дані про самокати поки не потрібно. Важливо лише враховувати, скільки таких самокатів було закуплено для прокату. Для цього додано новий реквізит документа «Кількість». Тип даних – «Число» (рис. 4.4).

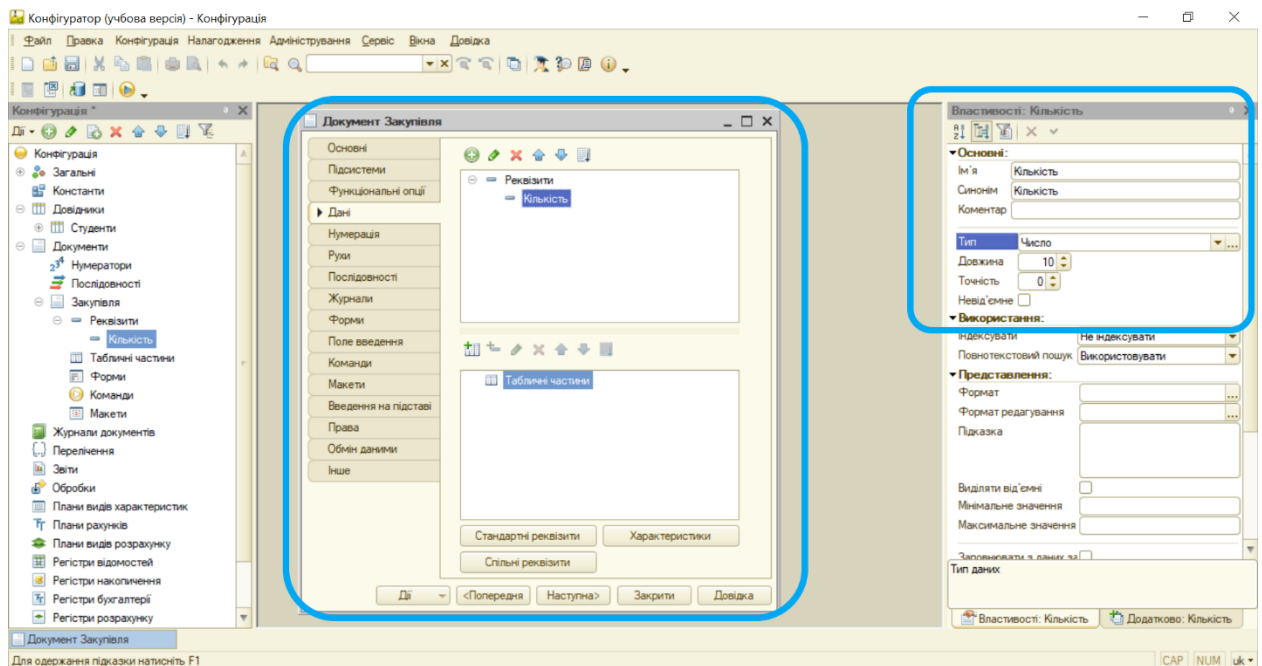


Рисунок 4.4 – Створення документу «Закупівля» з реквізитом «Кількість»

В режимі «BAS: Підприємство» зафіксуємо закупівлю кількох електросамокатів (рис. 4.5).

Закупівля (створення) *

Провести та закрити **Записати** **Ще**

Номер:

Дата:

Кількість:

Рисунок 4.5 – Фіксування закупівлі електросамокатів

Система згенерувала для документа інші стандартні реквізити: «Номер» та «Дата». Обидва поля заповнюються автоматично, дата може бути змінена. Таким чином, документ для реєстрації закупівлі електросамокатів було успішно реалізовано.

Далі аналогічним чином створено документ «ЗдачаВОренду» (рис. 4.6).

Рисунок 4.6 – Створення документу «ЗдачаВОренду»

На рисунку 4.7 показано створену форму реєстрації студентів, які взяли в оренду самокат. Тепер інформаційна система здатна реєструвати дату, час та ПІБ студента, який орендував самокат.

Дата	Номер	Студент
01.10.2022 12:00:00	000000001	Іваненко
01.10.2022 12:05:00	000000002	Сидоренко
01.10.2022 13:00:00	000000003	Петренко

Рисунок 4.7 – Реєстрація студентів, що орендували самокат

Реєстрація повернення самокатів виглядатиме аналогічно реєстрації задачі в оренду. Новий документ «Повернення» створюється копіюванням документа «ЗдачаВОренду».

Для того, щоб створені документи пов'язати між собою використано реєстр накопичення, що міститиме підсумкові значення для розрахунку залишків.

Новий реєстр накопичення «ЗалишкиСамокатів» виду «Залишки» показано на рисунку 4.8. Вид реєстру «Залишки» дозволяє налаштувати цей реєстр таким чином, що якісь об'єкти будуть вносити дані, а якісь, навпаки, віднімати з нього. У такий спосіб і відбувається зберігання залишків.

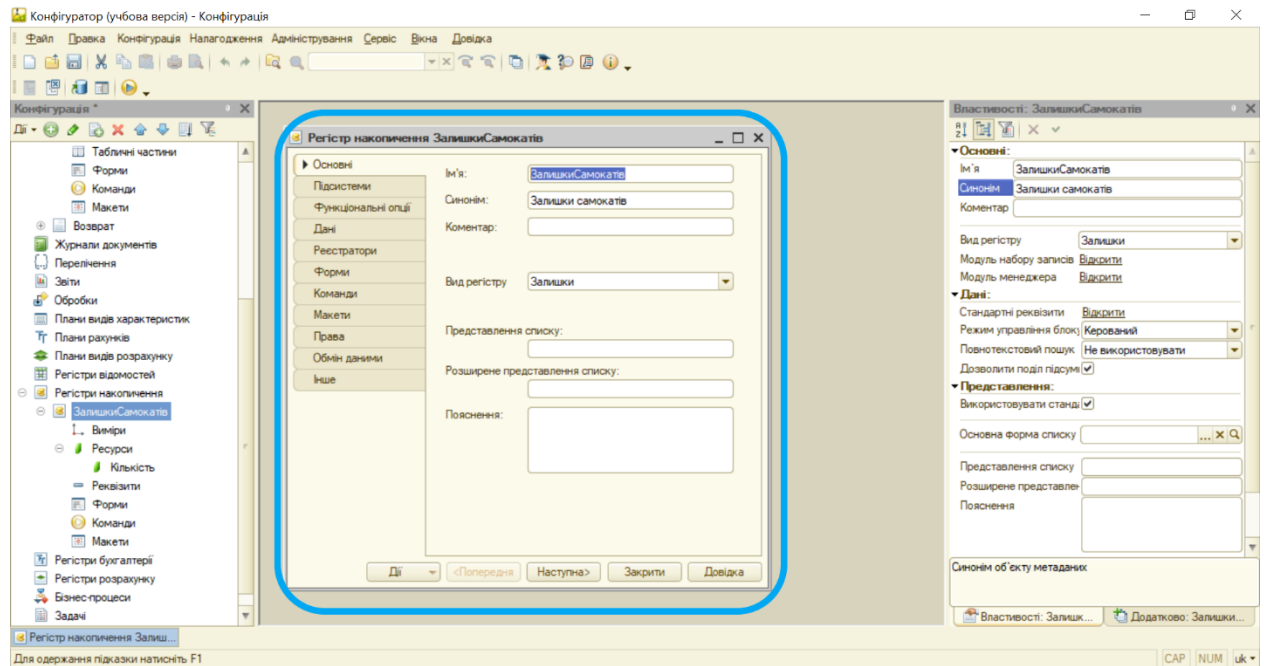


Рисунок 4.8 – Створення реєстру накопичення «ЗалишкиСамокатів»

Необхідно створити ще один реєстр накопичення «ОрендаСамокатів» виду «Залишки», який враховуватиме, хто взяв чи повернув самокат.

Щоб реєстри накопичення спрацювали, потрібно зробити наступне:

- визначити, які дані потраплятимуть до кожного реєстру (визначити документи-реєстратори);

- описати, яким чином дані з кожного документа-реєстратора повинні потрапляти в реєстри.

З документа «Закупівля» дані про кількість закуплених самокатів необхідно передавати в реєстр накопичення «ЗалишкиСамокатів».

Програмний код при успішному проведенні документа здійснить рух у реєстр накопичення, тобто скопіює дані з документа в реєстр накопичення.

Документ «ЗдачаВОренду» повинен робити рухи в обидва регістри накопичення: «ЗалишкиСамокатів» та «ОрендаСамокатів».

Документ «Повернення» налаштовується аналогічно документу «ЗдачаВОренду», тобто він повинен робити рухи в обидва регістри накопичення: «ЗалишкиСамокатів» та «ОрендаСамокатів».

Тепер кожен із цих документів копіює дані в регістри накопичення, додаючи або віднімаючи кількість електросамокатів.

Результат перевірки регістру в системі показано на рисунку 4.9.

Період	Регістратор	Номер р...	Кількість
+ 01.10.2022 11:00:00	Закупівля 000000001 від 01.10.2022 11:00:00	1	12
- 01.10.2022 12:00:00	Здача в оренду 000000001 від 01.10.2022 12:00:00	1	1
- 01.10.2022 12:05:00	Здача в оренду 000000002 від 01.10.2022 12:05:00	1	1
+ 01.10.2022 12:30:00	Повернення 000000001 від 01.10.2022 12:30:00	1	1
- 01.10.2022 13:00:00	Здача в оренду 000000003 від 01.10.2022 13:00:00	1	1

Рисунок 4.9 - Вміст регістру накопичення «ЗалишкиСамокатів»

Також потрібно побудувати звіт, який відображатиме список студентів, які ще не повернули самокати. Для цього потрібно створити новий звіт «ЗвітЗаОрендарями» та налаштувати його заповнення за допомогою конструктора схеми компоновання даних.

Усі створені об'єкти конфігурації є таблицями бази даних. У режимі «BAS:Підприємство» виконується заповнення цих таблиць даними. Щоб отримати дані для відображення у звіті, потрібно сформулювати запит до бази даних за допомогою конструктора запиту. Дані вибираються не з регістру накопичень безпосередньо, а з віртуальної таблиці, яку створює цей регістр автоматично. Ця віртуальна таблиця дозволяє отримати вже підсумовані значення за всіма документами.

Отриманий звіт виводить список орендарів, які не повернули самокат до пункту прокату вчасно (рис. 4.10).

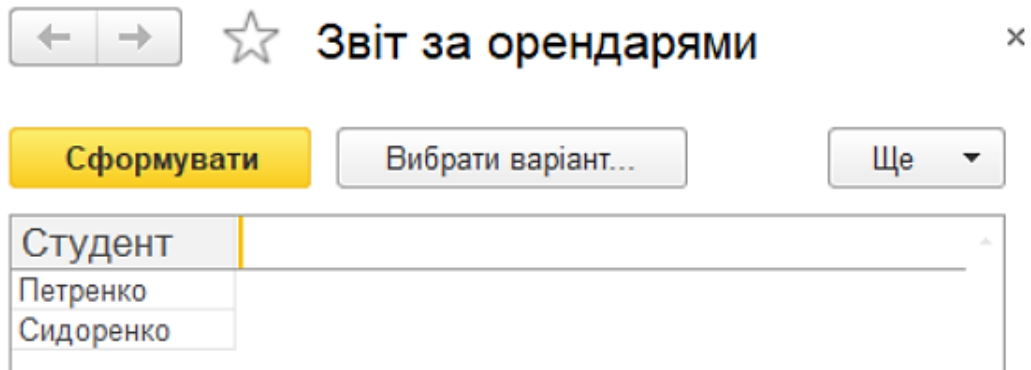


Рисунок 4.10 – Звіт за орендарями

4.2.2 Реалізація додаткового функціоналу АС

Для того, щоб зберігати інформацію про марку самокатів, було додано довідник «Самокат» (рис. 4.11). Додано перевірку, що при відсутності заповнення поля «Марка» буде видаватися помилка. Також модифіковано довідник «Студенти», шляхом додавання реквізитів «Телефон» і «Номер групи» (рис. 4.12). Приклад заповненого довідника «Самокат» наведено на рисунку 4.13.

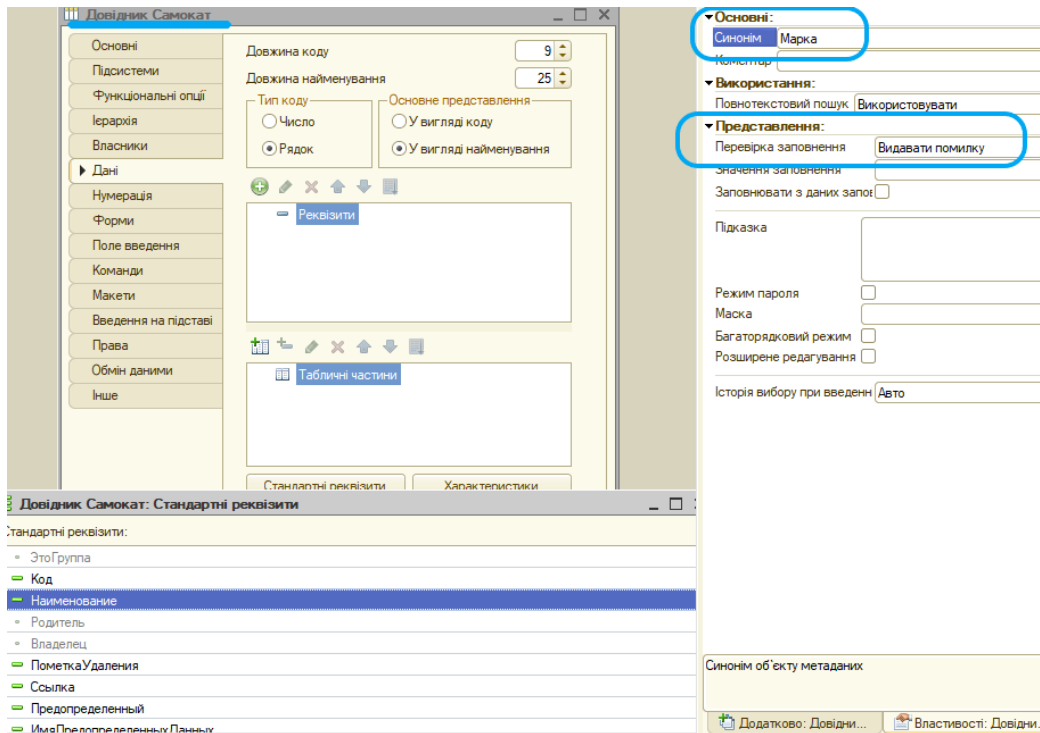


Рисунок 4.11 – Довідник «Самокат»

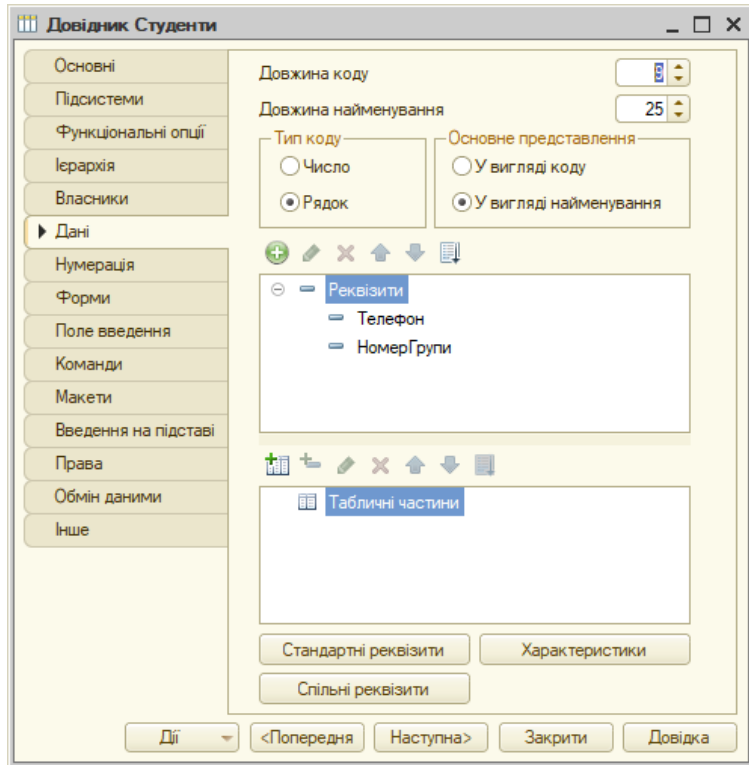


Рисунок 4.12 – Оновлений довідник «Студенти»

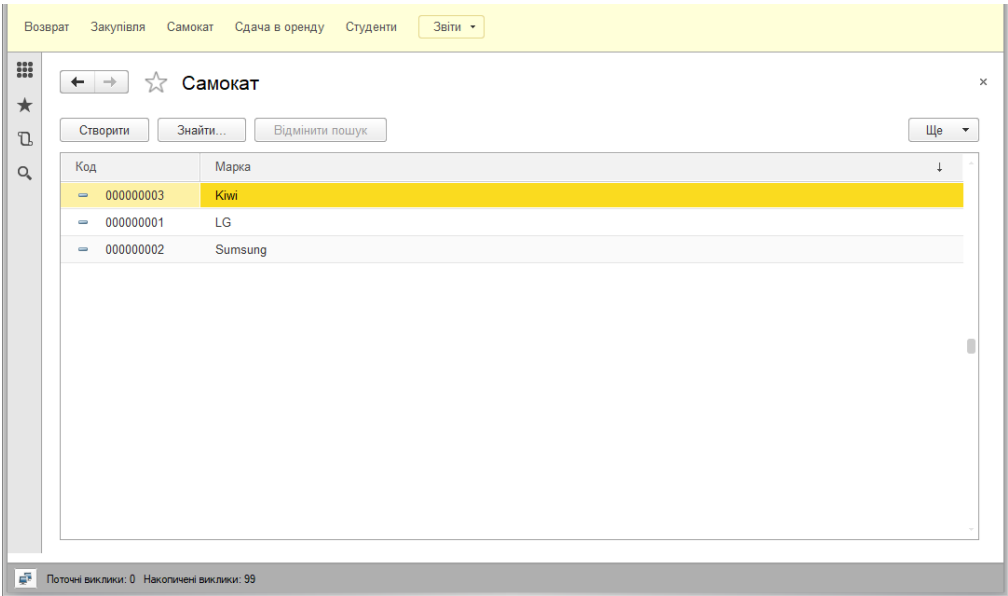


Рисунок 4.13 – Заповнений довідник «Самокат»

Зміна документа «ЗдачаВОренду» виконана таким чином, щоб також зазначалася кількість самокатів, що береться в оренду (рис. 4.14).

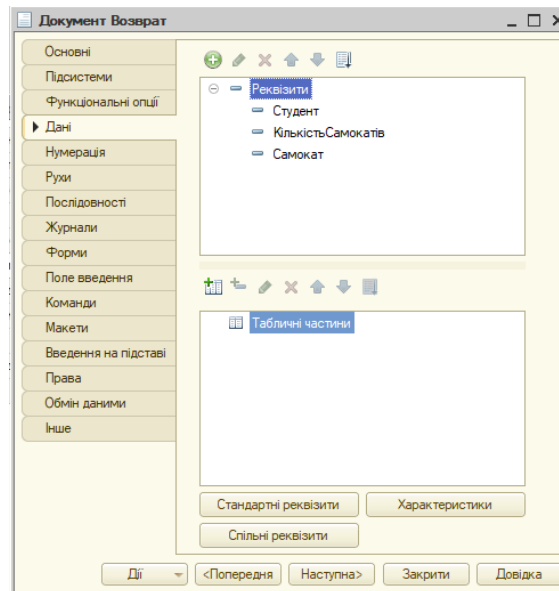


Рисунок 4.14 – Доданий реквізит «КількістьСамокатів»

Оскільки додано кількість самокатів, що беруться в оренду, то тепер при оформленні реєстр накопичення має оброблювати значення цієї кількості, а не «1», як це було раніше (рис. 4.15).

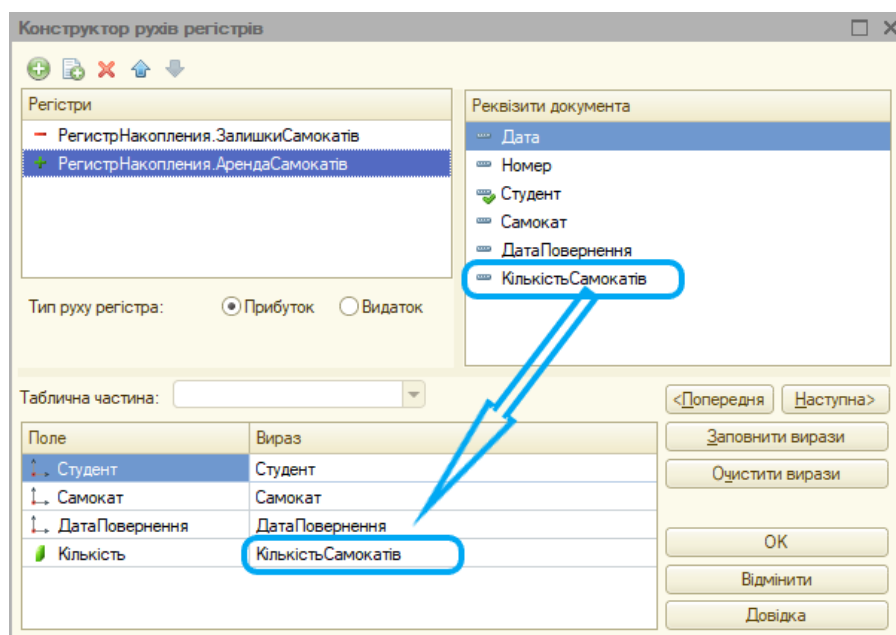


Рисунок 4.15 – Обробка «КількостіСамокатів» у реєстрі накопичення

Також додаються допоміжні реквізити для того, щоб на їх основі будувати звіти. Створену закупівлю в режимі «BAS:Підприємство» наведено на рисунку 4.16.

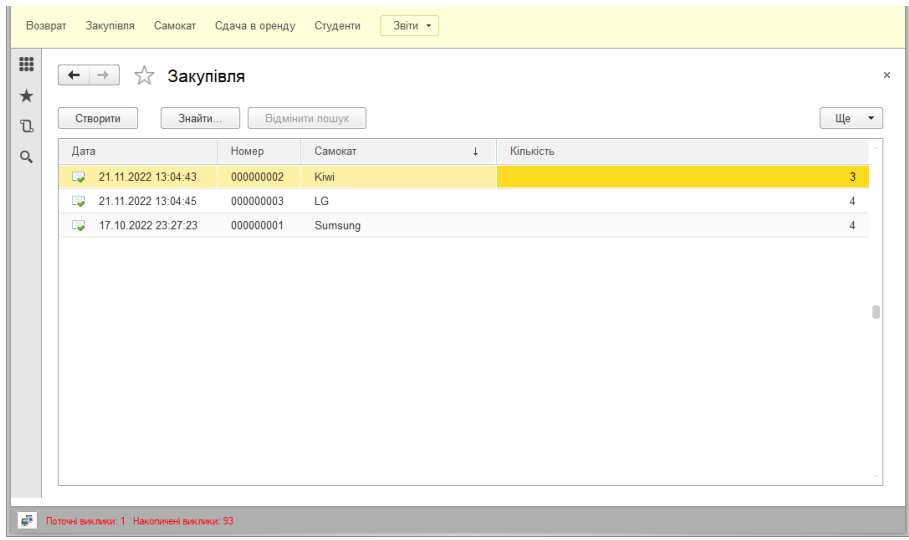


Рисунок 4.16 – Заповнення документу «Закупівля»

Результати додавання звіту «СамокатиНаБазі» та результат його формування наведено на рисунках 4.17 – 4.18.

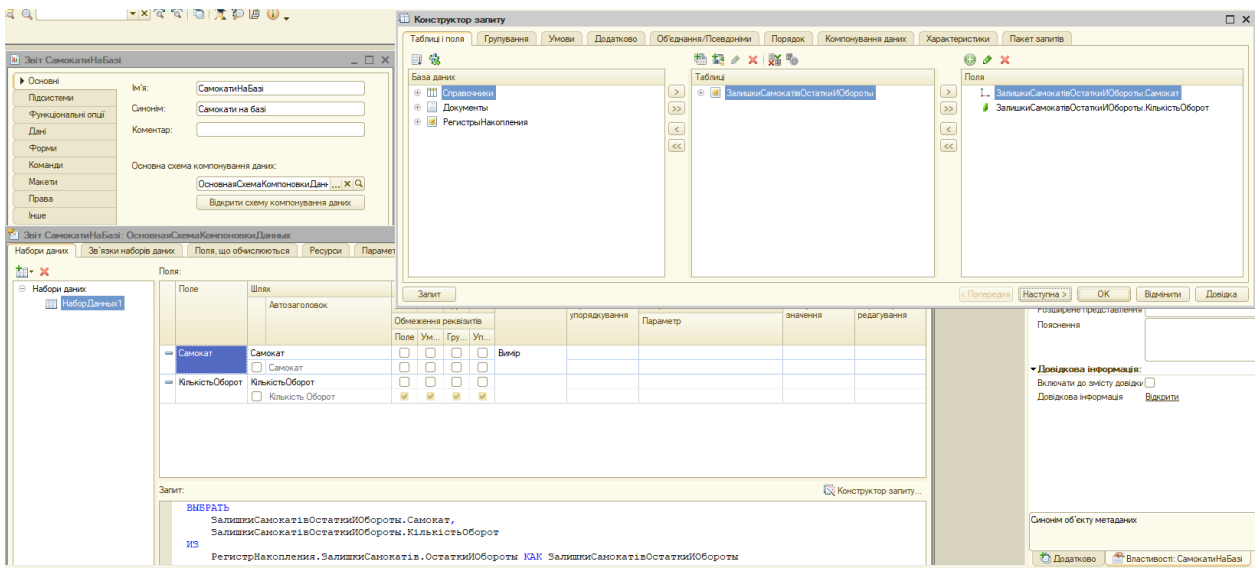


Рисунок 4.17 – Створення звіту «СамокатиНаБазі»

Для модифікації звіту «ЗвітПоОрендаторам», необхідно додати інформацію про орендаторів, самокати що вони орендували та дату (рис. 4.19).

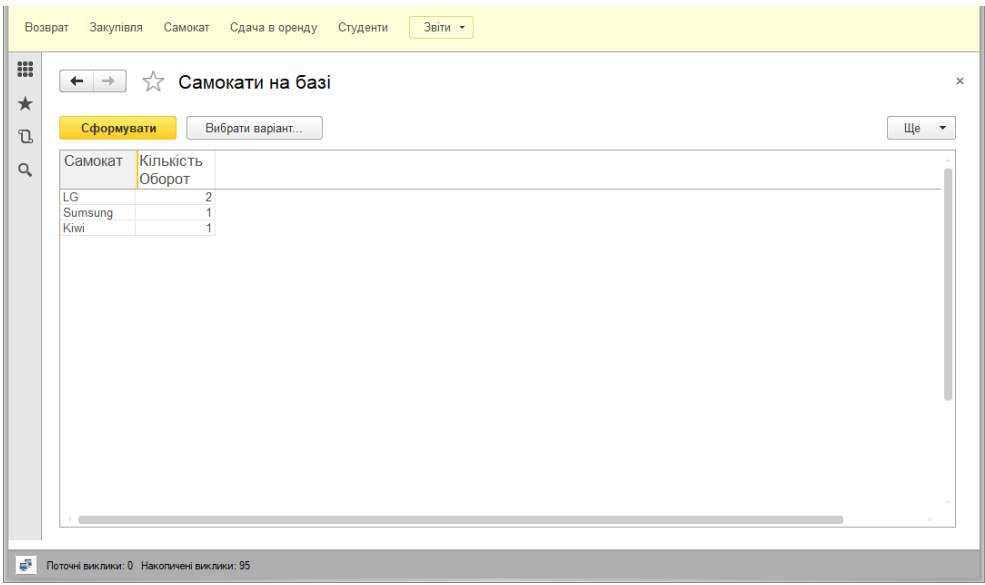


Рисунок 4.18 – Звіт «СамокатиНаБазі»

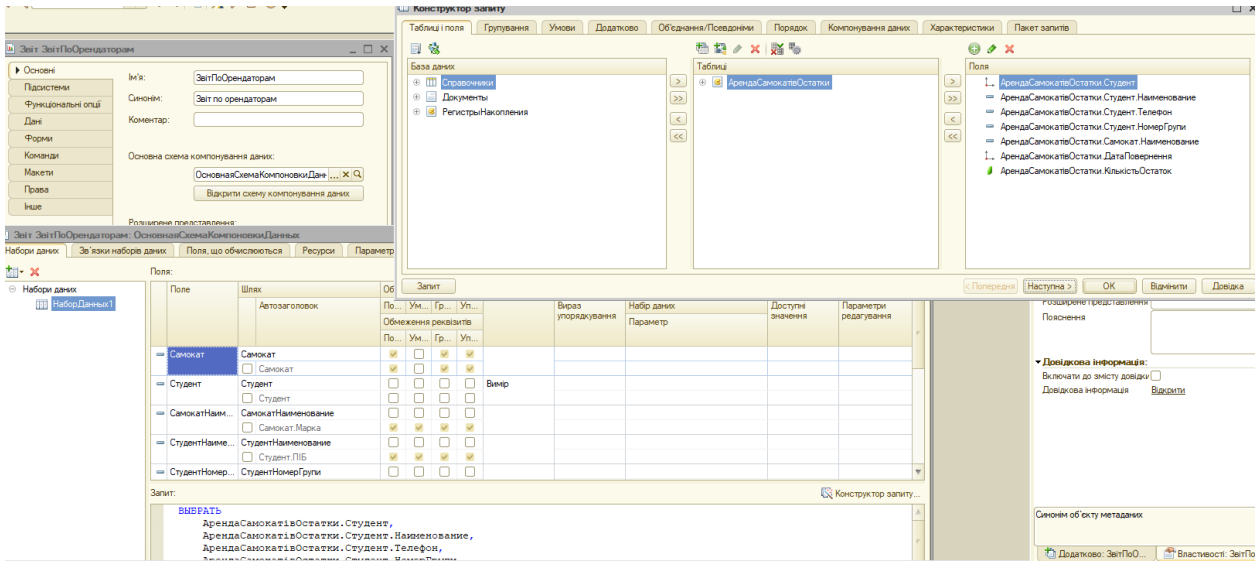


Рисунок 4.19 – Модифікація звіту по орендаторам

Для реалізації функції надавання комплекту захисту двох типів («Базовий» та «Стандартний»), додано довідник «КомплектЗахисту» (рис. 4.20). Результат його заповнення - на рисунку 4.21. В першу чергу, приведені вище комплекти відрізняються за ціною, де базовий – 0 грн, стандартний – 30 грн, також самокати мають різні марки, а відповідно і ціни їх прокату бути різними.

Рисунок 4.20 – Довідник «КомплектЗахисту»

Рисунок 4.21 – Заповнення довідника «КомплектЗахисту»

Для встановлення цін було розроблено документ «Установка цін», зокрема саме через цей документ ціни будуть змінюватися у подальшому (рис. 4.22). Для того щоб встановлювати ціни із періодичністю, розроблено реєстри відомостей «УстановкаЦіниСамокатів» та «УстановкаЦіниКомплектуЗахисту», періодичність 1 день (рис. 4.23).

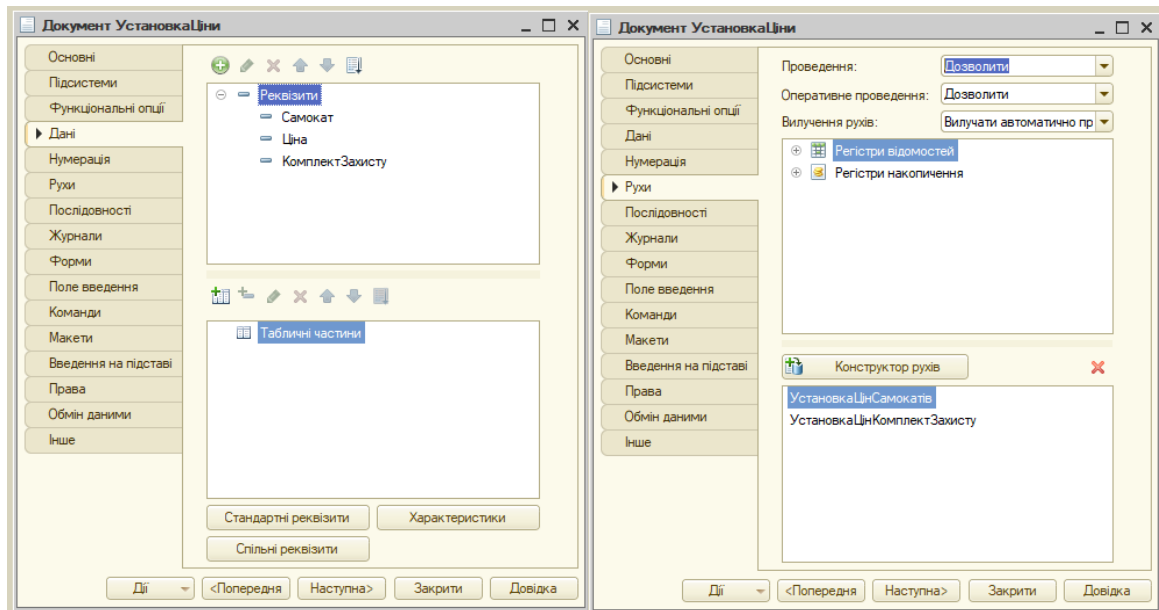


Рисунок 4.22 – Документ «УстановкаЦіни»

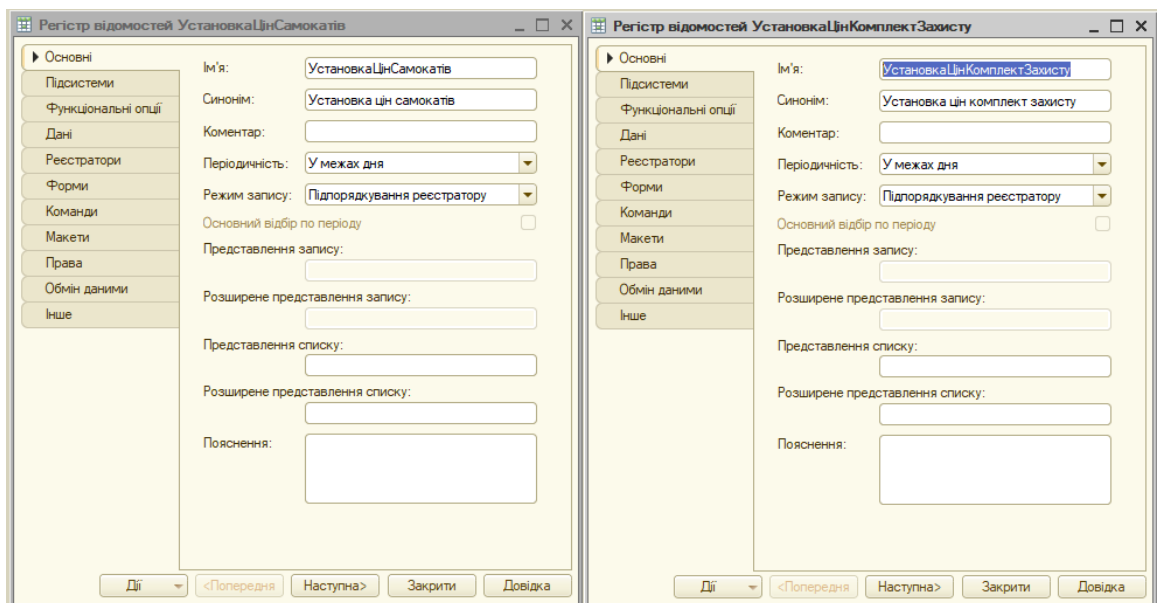


Рисунок 4.23 – Регістри відомостей «УстановкаЦінСамокатів» та «УстановкаЦінКомплектуЗахисту»

Результат заповнення документа «УстановкаЦін» на рисунку 4.24.

Для того, щоб мати можливість відстежувати надходження та витрати, створено регістр накопичення «Кошти» (рис. 4.25).

При здачі в оренду самоката необхідно залишати деякий документ під заставу, документи характеризуються типом, серією та номером, тому для створено та заповнено довідник «ТипДокументу» (рис. 4.26).

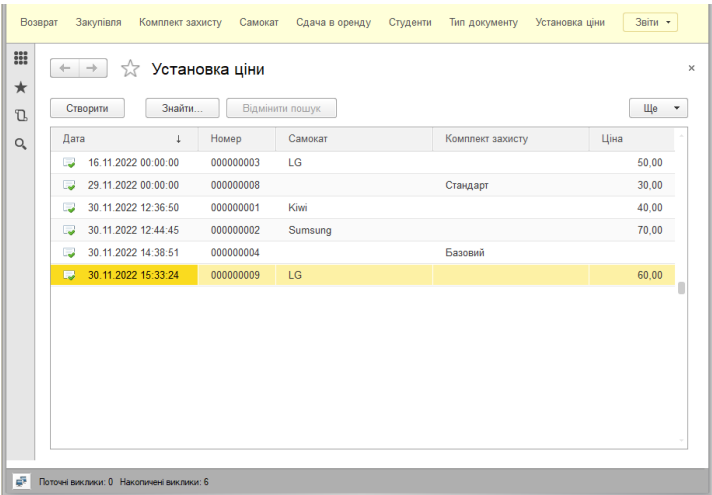


Рисунок 4.24 – Заповнення документа «УстановкаЦіни»

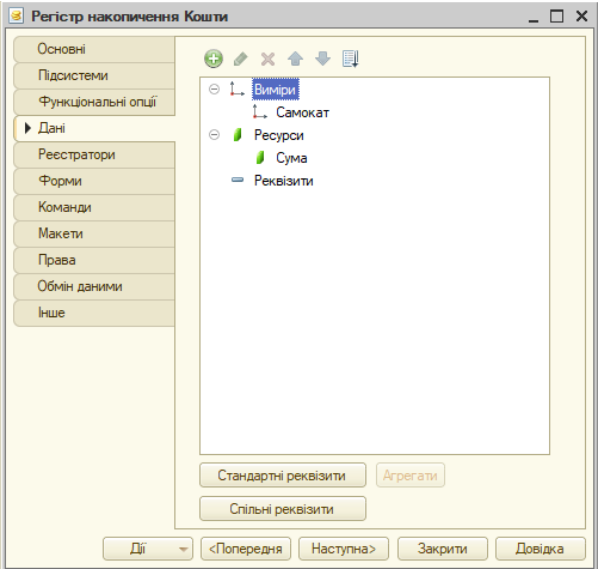


Рисунок 4.25 – Створення реєстру накопичення «Кошти»

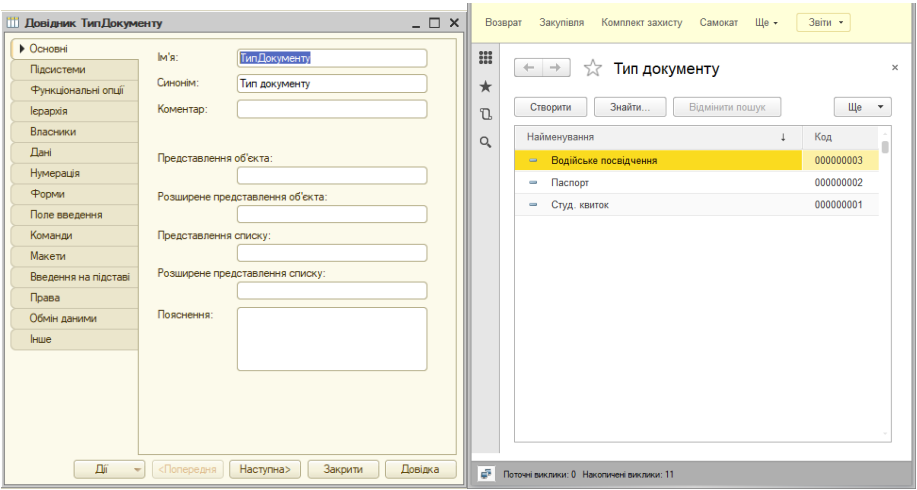


Рисунок 4.26 – Довідник «ТипДокументу»

Документ «ЗдачаВОренду» змінено таким чином, щоб можна було вказувати дані про документ застави (рис. 4.27) та створено його форму (рис. 4.28). За замовченням при відкритті документу «ЗдачаВОренду» необхідно щоб тип комплекту захисту автоматично був «Базовим», для цього в модуль форми додано обробник події «При відкритті», в якій встановлено значення комплекту захисту.

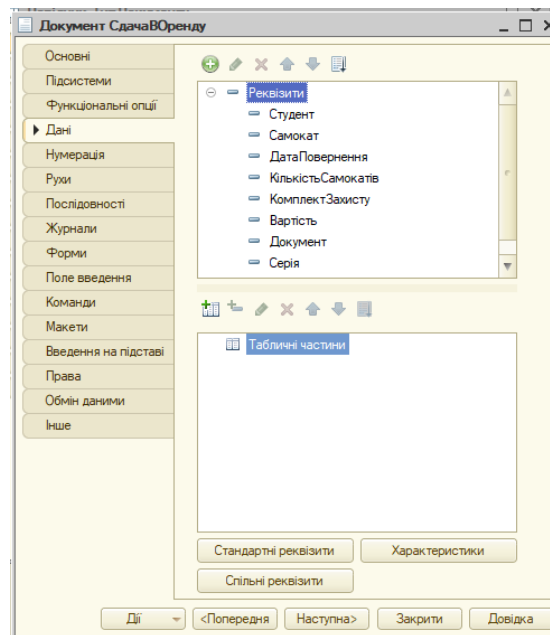


Рисунок 4.27 – Документ «ЗдачаВОренду»

Рисунок 4.28 – Форма документу «ЗдачаВОренду»

Після введення всіх даних оренди необхідно провести розрахунок вартості отриманої конфігурації, для цього додано кнопку «Розрахунок вартості» та обробник до неї (рис. 4.29).

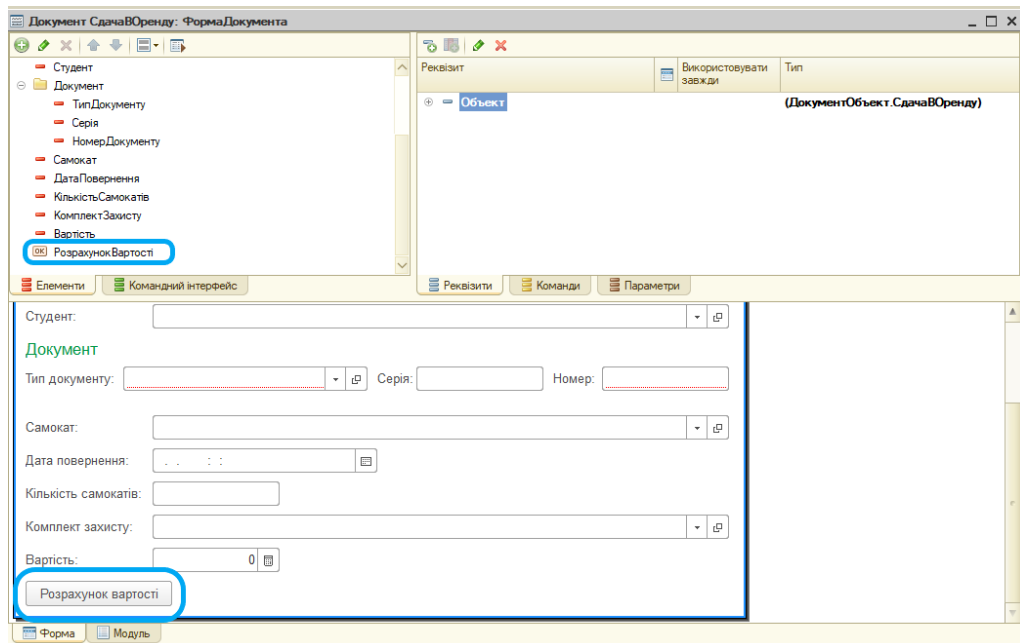


Рисунок 4.29 – Кнопка «Розрахунок вартості»

При оформленні оренди треба забезпечити можливість друку розписки, тому у режимі «Конфігурації» до документу «ЗдачаВОренду» додано макет розписка та кнопку для друку на командній панелі форми «Здача в оренду».

При реалізації процесу повернення інвентарю, важливим моментом є те, що необхідно контролювати час оренди, та просити доплатити або ж навпаки повернути частину грошей, тому документ «Повернення» було розширено та додано форму (рис 4.30). Макет документу повернення наведено на рисунку 4.31.

Для забезпечення коректного поводження регістрів накопичення, код їх рухів було написано власноруч відповідно до можливого сценарію.

Для розробки звіту з інформацією про те, скільки годин який самокат брали й скільки грошей той чи інший самокат приніс, було додано новий регістр накопичення для підрахунку кількості годин «КількістьГодин» (рис. 4.32).

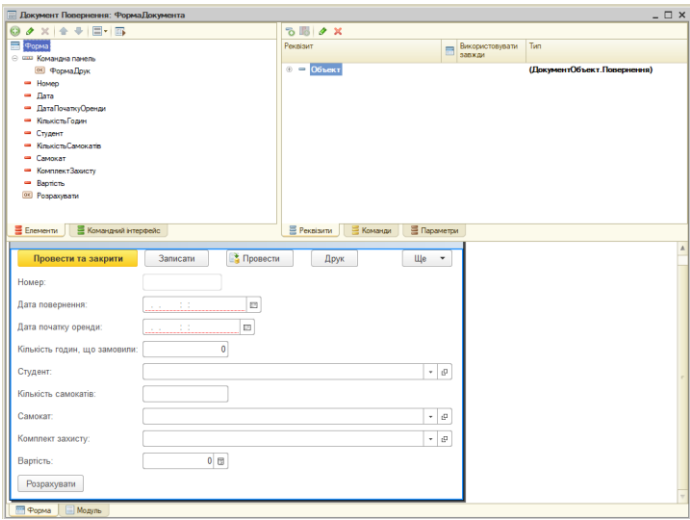


Рисунок 4.30 – Форма документу «Повернення»

Документ Повернення: ЛистПовернення								
	1	2	3	4	5	6	7	8
1								
2		Документ повернення № <Номер>						
3								
4		Видав: ТОВ Прокат №1 (Головний корпус, тел. 067-777-77-77)						
5								
6		Повернув: <[Студент] (Група [Група], моб. [телефон])>						
7		<Дата початку оренди [ДатаПочаткуАренди]>						
8		Дата повернення: <[Дата] Час: [Час]>						
9								
10		№	Об'єкт проката		Кол-во	Ціна	Вартість	
11		1	<КомплектЗахисту>		1,00	<ЦінаКомпл	<ВартістьКомплектуЗахисту>	
12		2	<Самокат>		<КількістьС	<ЦінаСамок	<ВартістьСамокату>	
13		Всього: <Вартість>						
14		<Повідомлення: [ТекстПов]; Сума: [Сума] грн>						
15								
16								
17		<Студент, [Студент] (Група [Група], моб. [телефон]) повернув вище перелічені об'єкт						
18								
19								
20								
21		<Дата>						
22		(Дата)						
23		(Підпись)						

Рисунок 4.31 – Макет документу «Повернення»

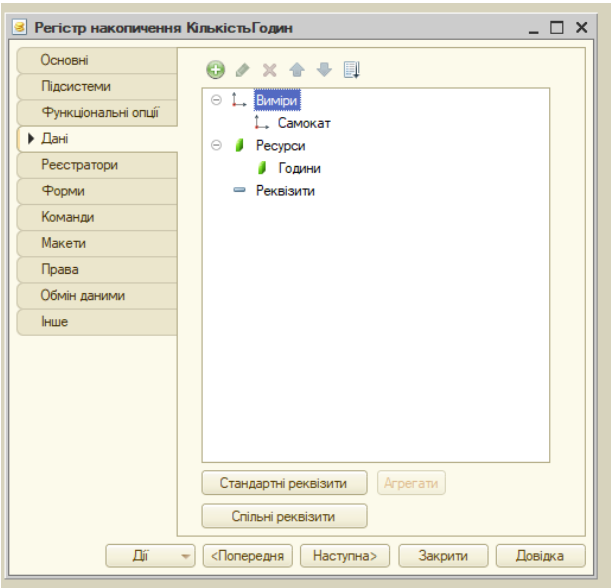


Рисунок 4.32 – Регістр «КількістьГодин»

Для формування звіту «Виручка» було створено запит, що дозволяє вибрати дані з двох реєстрів «КількістьГодин» та «Кошти» та об'єднати їх за «Самокатом». Результат роботи звіту – на рисунку 4.33.

Самокат	Години Прибуток	Сума Оборот
Kiwi	2,00	250,00
LG	1,00	300,00
Sumsung	3,00	420,00

Рисунок 4.33 – Результат роботи звіту «Виручка»

4.3 Висновки до розділу 4

В цьому розділі відповідно до мети та завдань дипломної роботи розроблено автоматизовану систему обліку прокату самокатів.

Для реалізації системи розроблено довідники, звіти, документи, що наведені в таблиці 4.1.

Програмно реалізовано запити, що дозволяють сформувати певні звіти, а також створено обробники подій та реєстри накопичення, за допомогою яких організовано рух даних для реалізації бізнес-процесу.

Розроблена АС дозволяє виконувати облік самокатів, які є на базі, знаходяться в оренді та були придбані.

При роботі системи контролюється правильність заповнення полів, а також позаштатні ситуації типу неповернення самокату вчасно арендарем.

Система виконує також розрахунок витрат та прибутку пункту прокату.

Таблиця 4.1 – Розроблені елементи конфігурації

Тип об'єкта	Призначення	Опис
Довідники	«Студенти»	Зберігає список студентів (ПІБ, група, телефон).
	«Самокат»	Містить відомості про марку самоката
	«КомплектЗахисту»	Містить інформацію про Базовий та Стандартний комплекти захисту доступні під час оренди.
	«ТипДокументу»	Містить інформацію про тип документа під заставу (тип, серія, номер)
Звіти	«ЗвітЗаОрендарями»	Відображає список студентів, які ще не повернули самокати.
	«СамокатиНаБазі»	Відображає інформацію про кількість та марки доступних для оренди самокатів на базі.
	«Самокат»	Відображає інформацію про марки самокатів
	«Виручка»	Відображає інформацію про прибуток
	«Закупівля»	Відображає інформацію про кількість та марки закуплених самокатів
	«Комплектзахисту»	Відображає інформацію про типи комплектів захисту
Документи	«Закупівля»	Дозволяє реалізувати закупівлю електросамокатів для пункту прокату (номер, дата, кількість самокатів, марки).
	«ЗдачаВОренду»	Дозволяє фіксувати дату, час, ПІБ студента, який орендував самокат, кількість взятих самокатів, а також дані про документ застави.
	«Повернення»	Дозволяє фіксувати дату та час, а також ПІБ студента, який повернув самокат.
	«Установка цін»	Дозволяє змінювати вартість оренди
Регістри накопичення	«Залишки»	Пов'язує документи «ЗдачаВОренду» та «Повернення» та містить підсумкові значення для розрахунку залишків самокатів на базі
	«ОрендаСамокатів»	Враховує хто взяв чи повернув самокат додаванням або відніманням.
	«Кошти»	Дозволяє відстежувати надходження коштів та їх витрати
	«КількістьГодин»	Виконує підрахунок фактичної кількості годин оренди

ВИСНОВКИ

Як показали проведені дослідження, автоматизація є актуальним і важливим напрямком у сфері теорії та практики управління бізнес-процесами. Її основною метою є підвищення соціально-економічної ефективності компанії шляхом цілеспрямованого впровадження програмних систем, що підвищують швидкість і точність реалізації бізнес-процесів, мінімізують при цьому помилкові дії і рішення, що виникають під впливом так званого "людського фактору".

Існуючий рівень розвитку програмного забезпечення дозволяє автоматизувати переважну частину бізнес-процесів для ефективної організації різних видів діяльності.

Результатом виконання дипломної роботи є розробка системи обліку, яка забезпечує автоматизацію бізнес-процесу прокату електросамокатів, що включає етапи закупівлі електросамокатів, здачі самокатів в оренду студентам студмістечка, а також повернення самокатів студентами.

Розробку виконано на платформі BAS з використанням вбудованої скриптової мови програмування та мови запитів системи.

Основою розробленої АС є інформаційна база, що містить інформацію про самокати, їх марки та кількість; про студентів, їх ПІБ, документи застави, телефон, групу; результати фінансових розрахунків, що пов'язані з витратами на закупівлю самокатів та прибутками від їх оренди.

Програмний застосунок надає можливість контролювати процеси видачі-прийняття самокатів та захисного спорядження, а також розрахувати вартість оренди з урахуванням терміну використання. У разі перевищення вказаного терміну оренди, система формує повідомлення. Розроблене програмне забезпечення дозволяє підвищити ефективність процесів видачі-прийняття обладнання та інвентаря та проведення фінансових розрахунків пункту прокату.

В подальшій роботі можливо розширити функціональність АС, зокрема шляхом додавання процесу страхування орендаря.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Schadler, T. The sorry state of digital transformation in 2018 [Electronic resource] / T. Schadler. – Access mode: <https://www.forrester.com/blogs/the-sorry-state-of-digital-transformation-in-2018/>.
2. Business process automation - the complete guide [Electronic resource]. – Access mode: <https://kissflow.com/workflow/bpm/business-process-automation/bpa-overview/>.
3. Business process management system (BPMS) – everything you need to know [Electronic resource]. – Access mode: <https://kissflow.com/workflow/bpm/what-is-bpms/>.
4. Why business process management (BPM) software? How to choose the best platform? [Electronic resource]. – Access mode: <https://kissflow.com/workflow/bpm/best-bpm-software/>.
5. BPM'ONLINE [Electronic resource]. – Access mode: <https://academy.terrasoft.ua/content/obzor-terrasoft-creatio>.
6. Fadila, L. Modelling business processes for outsourcing into the fog and cloud computing / L. Fadila, Z. Karim, B. Djamel // Data-driven process discovery and analysis: 8th International symposium Seville, Spain, 13–14 December 2018: proceedings. – 2018. - P. 18-31.
7. Part I: an introduction to SAP netweaver BPM [Electronic resource]. – Access mode: <https://blogs.sap.com/2009/08/04/part-i-an-introduction-to-sap-netweaver-bpm/>.
8. Автоматизація бізнес-процесів підприємства [Електронний ресурс]. – Режим доступу: https://stud.com.ua/37085/ekonomika/avtomatizatsiya_biznes_protseviv_pidpriemstva.
9. Нетепчук, В.В. Управління бізнес-процесами: навчальний посібник / В.В. Нетепчук. - Рівне: НУВГП, 2014. - 158 с.

10. Швиданенко, Г.О. Формування бізнес-моделі підприємства: навчальний посібник / Г. О. Швиданенко, Н. В. Ревуцька. - К.: КНЕУ, 2013. – 423 с.
11. Бунке, О.С. Автоматизація бізнес процесів: навчальний посібник / О.С. Бунке. – К.: КПП ім. Ігоря Сікорського, 2021. – 39 с.
12. Данченко, О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. – К.: Університет економіки та права «КРОК», 2017. – 238 с.
13. BAS ERP [Electronic resource]. – Access mode: <https://www.bas-soft.eu/soft/bas-corp/bas-erp/>.
14. Лекція від Центру сертифікованого навчання компанії ПРОКОМ [Електронний ресурс]. – Режим доступу: <https://youtu.be/TXlb3tqfBJA>.
15. BAS програми - встановлення та налаштування програм автоматизації бізнесу [Електронний ресурс]. – Режим доступу: <https://www.1c.kiev.ua/ua/>.
16. Інформаційна система ІТС [Електронний ресурс]. – Режим доступу: <https://its.1c.ua/db/v83docu#browse:13:-1:50001:50004>.
17. Демонстраційні бази (онлайн) [Електронний ресурс]. – Режим доступу: <https://www.bas-soft.eu/soft/bas-demo/>.

ДОДАТОК А
Текст програми

A.1 Модуль об'єкта документа «Закупівля»

```
Процедура ОбробкаПроведення(Відмова, Режим)
    // реєстр ЗалишкиСамокатів Прихід
    Рухи.ЗалишкиСамокатів.Записувати = Істина;
    Рух = Рухи.ЗалишкиСамокатів.Додати();
    Рух.ВидРуху = ВидРухуНакопичення.Прихід;
    Рух.Період = Дата;
    Рух.Самокат = Самокат;
    Рух.Кількість = Кількість;
КінецьПроцедури
```

A.2 Модуль об'єкта документа «ЗдачаВОренду»

```
Процедура ОбробкаПроведення(Відмова, Режим)
    // реєстр ЗалишкиСамокатів Витрата
    Рухи.ЗалишкиСамокатів.Записувати = Істина;
    Рух = Рухи.ЗалишкиСамокатів.Додати();
    Рух.Вид Рухи = ВидРухуНакопичення.Витрата;
    Рух.Період = Дата;
    Рух.Самокат = Самокат;
    Рух.Кількість = КількістьСамокатів;

    // реєстр ОрендаСамокатів Прихід
    Рухи.ОрендаСамокатів.Записувати = Істина;
    Рух = Рухи.ОрендаСамокатів.Додати();
    Рух.Вид Руху = Вид РухиНакопичення.Прихід;
    Рух.Період = Дата;
    Рух.Студент = Студент;
    Рух.Самокат = Самокат;
```

```

Рух.ДатаПовернення = ДатаПовернення;
Рух.Кількість = КількістьСамокатів;

// реєстр Кошти Прихід
Рухи.Кошти.Записувати = Істина;
Рух = Рух.Кошти.Додати();
Рух.Вид Рухи = Вид РухиНакопичення.Прихід;
Рух.Період = Дата;
Рух.Самокат = Самокат;
Рух.Сума = Вартість;

// реєстр КількістьГодин Прихід
Рухи.КількістьГодин.Записувати = Істина;
Рух = Рухи.КількістьГодин.Додати();
Рух.ВидРухи = ВидРухиНакопичення.Прихід;
Рух.Період = Дата;
Рух.Самокат = Самокат;
КінецьПроцедури

```

А. 3 Модуль форми документа «ЗдачаВОренду»

```

&НаКлієнті
Процедура РозрахунокВартості(Команда)
    РозрахунокВартостіНаСервері();
КінецьПроцедури

&НаСервері
Процедура РозрахунокВартостіНаСервері()
    // Вставити вміст обробника.
    Змін зп;

```

```

кз = 0;
Рег = РегистриВідомостей.УстановкаЦінСамокатів;
Вибірка = Рег.Вибрати();
Доки Вибірка.Наступний() Цикл

        Якщо          (Вибірка.Самокат.Найменування          =
Об'єкт.Самокат.Найменування) Тоді
                зп = Вибірка.Ціна;
        КінецьЯкщо;
КінецьЦиклу;

        Якщо          (Об'єкт.КомплектЗахисту.Найменування          <>
Довідники.КомплектЗахисту.Базовий.Найменування) Тоді
                Рег          =
РегистриВідомостей.УстановкаЦінКомплектЗахисту;
                Вибірка = Рег.Вибрати();
                Доки Вибірка.Наступний() Цикл
                        Якщо
(Вибірка.КомплектЗахисту.Найменування          =
Об'єкт.КомплектЗахисту.Найменування) Тоді
                                кз = Вибірка.Ціна;
                        КінецьЯкщо;
                КінецьЦиклу;
        КінецьЯкщо;

тд = (Об'єкт.ДатаПовернення - Об'єкт.Дата)/3600;
Об'єкт.Вартість = Об'єкт.КількістьСамокатів * (зп *
тд) + кз;

        //Повідомити(тд);

```

```

КінецьПроцедури
&НаКлієнті
Процедура ПриВідкритті (Відмова)
    ПриВідкриттіНаСервері ();
КінецьПроцедури

```

```

&НаСервері
Процедура ПриВідкриттіНаСервері ()
    Об'єкт.КількістьСамокатів = 1;
    Об'єкт.КомплектЗахисту
Довідники.КомплектЗахисту.Базовий;
КінецьПроцедури

```

```

&НаКлієнті
Процедура Друк (Команда)
    ТабДок = Новий ТабличнийДокумент;

    ДрукНаСервері (ТабДок);

    ТабДок.Показати ("Розписка");
КінецьПроцедури

```

```

&НаСервері
Процедура ДрукНаСервері (ТабДок)
    змін зп;
    //отримання макету
    Макет
Документи.ЗдачаВОренду.ОтриматиМакет ("Розписка");

```

```

//Отримання областей
ОбластьШапка = Макет.ОтриматиОбласть ("Шапка");
ОбластьШапкаЗамовник                                     =
Макет.ОтриматиОбласть ("ШапкаЗамовник");
ОбластьШапкаТабл                                           =
Макет.ОтриматиОбласть ("ШапкаТаблиці");
ОбластьРядкиТабл                                           =
Макет.ОтриматиОбласть ("РядкиТаблиці");
ОбластьПідвал = Макет.ОтриматиОбласть ("Підвал");
ОбластьЗагальнийТекст                                     =
Макет.ОтриматиОбласть ("ЗагальнийТекст");

//заповнення параметрів і вивод у табл док
ОбластьШапка.Параметри.Номер = Об'єкт.Номер;
ТабДок.Вивести (ОбластьШапка);

ОбластьШапкаЗамовник.Параметри.Студент                   =
Об'єкт.Студент.Найменування;
ОбластьШапкаЗамовник.Параметри.Група                     =
Об'єкт.Студент.НомерГрупи;
ОбластьШапкаЗамовник.Параметри.телефон                  =
Об'єкт.Студент.Телефон;
ОбластьШапкаЗамовник.Параметри.Документ                  =
Об'єкт.Документ;
ОбластьШапкаЗамовник.Параметри.Серія                     =
Об'єкт.Серія;
ОбластьШапкаЗамовник.Параметри.НомерДокументу            =
Об'єкт.НомерДокументу;
ОбластьШапкаЗамовник.Параметри.ДатаПовернення           =
Об'єкт.ДатаПовернення;

```

```

ТабДок.Вивести (ОбластьШапкаЗамовник) ;

ТабДок.Вивести (ОбластьШапкаТабл) ;

ОбластьРядкиТабл.Параметри.КомплектЗахисту =
Об'єкт.КомплектЗахисту.Найменування;

кз = 0;

Рег =
РегістриВідомостей.УстановкаЦінКомплектЗахисту;

Вибірка = Рег.Вибрати();

Доки Вибірка.Наступний() Цикл
    Якщо (Вибірка.КомплектЗахисту.Найменування =
Об'єкт.КомплектЗахисту.Найменування) Тоді
        кз = Вибірка.Ціна;
    КінецьЯкщо;
КінецьЦиклу;

ОбластьРядкиТабл.Параметри.ЦінаКомплектуЗахисту = "
"+кз;

ОбластьРядкиТабл.Параметри.ВартістьКомплектуЗахисту
= " "+кз;

Повідомити ("Тест"+кз) ;

ОбластьРядкиТабл.Параметри.Самокат =
Об'єкт.Самокат.Найменування;

ОбластьРядкиТабл.Параметри.КількістьСамокатів =
Об'єкт.КількістьСамокатів;

Рег = РегістриВідомостей.УстановкаЦінСамокатів;
Вибірка = Рег.Вибрати();
Доки Вибірка.Наступний() Цикл

```

```

        Якщо (Вибірка.Самокат.Найменування =
Об'єкт.Самокат. Найменування) Тоді
            зп = Вибірка.Ціна;
        КінецьЯкщо;
    КінецьЦиклу;
    ОбластьРядкиТабл.Параметри.ЦінаСамокату = зп;
    ОбластьРядкиТабл.Параметри.ВартістьСамокату =
Об'єкт.Вартість-кз;
    ТабДок.Вивести (ОбластьРядкиТабл) ;

    ОбластьПідвал.Параметри.Вартість = Об'єкт.Вартість;
    ТабДок.Вивести (ОбластьПідвал) ;

    ОбластьЗагальнийТекст.Параметри.Студент =
Об'єкт.Студент. Найменування;
    ОбластьЗагальнийТекст.Параметри.Група =
Об'єкт.Студент.НомерГрупи;
    ОбластьЗагальнийТекст.Параметри.телефон =
Об'єкт.Студент.Телефон;
    ОбластьЗагальнийТекст.Параметри.Дата = Об'єкт.Дата;
    ТабДок.Вивести (ОбластьЗагальнийТекст) ;
КінецьПроцедури

```

А.4 Модуль об'єкта документа «Повернення»

```

Процедура ОбробкаПроведення(Відмова, Режим)
    // реєстр ЗалишкиСамокатів Прихід
    Рухи.ЗалишкиСамокатів.Записувати = Істина;
    Рух = Рухи.ЗалишкиСамокатів.Додати();
    Рух.Вид Руху = ВидРухуНакопичення.Прихід;

```

```

Рух.Період = Дата;
Рух.Самокат = Самокат;
Рух.Кількість = КількістьСамокатів;

// реєстр ОрендаСамокатів Витрата
Рухи.ОрендаСамокатів.Записувати = Істина;
Рух = Рухи.ОрендаСамокатів.Додати();
Рух.ВидРуху = Вид РухуНакопичення.Витрата;
Рух.Період = Дата;
Рух.Студент = Студент;
Рух.Самокат = Самокат;
Рух.ДатаПовернення = Дата;
Рух.Кількість = КількістьСамокатів;

// реєстр Кошти Прихід
Рухи.Кошти.Записувати = Істина;
Рух = Рухи.Кошти.Додати();

Якщо Вартість > 0 Тоді
    Рух.ВидРуху = Вид РухНакопичення.Прихід;
Інакше
    Рух.ВидРуху = ВидРухуНакопичення.Витрата;
КінецьЯкщо;

Рух.Період = Дата;
Рух.Самокат = Самокат;
Рух.Сума = Макс(Вартість, -Вартість);

Рухи.КількістьГодин.Записувати = Істина;

```

```

Рух = Рухи.КількістьГодин.Додати();
Рух.Вид Руху= Вид РухуНакопичення.Прихід;
Рух.Период = Дата;
Рух.Самокат = Самокат;
Рух.Години = КількістьГодин;
КінецьПроцедури

```

А.5 Модуль форми документа «Повернення»

```

&НаКлієнті
Процедура Розрахувати(Команда)
    РозрахуватиНаСервері();
КінецьПроцедури

```

```

&НаСервері
Процедура РозрахуватиНаСервері()

```

```

    Змін зп;
    кз = 0;
    Рег = РегистриВідомостей.УстановкаЦінСамокатів;
    Вибірка = Рег.Вибрати();
    Доки Виборка.Наступний() Цикл

        Якщо (Вибірка.Самокат.Найменування =
Об'єкт.Самокат. Найменування) Тоді
            зп = Вибірка.Ціна;
        КінецьЯкщо;
    КінецьЦикла;

```

```

        Якщо (Об'єкт.КомплектЗахисту. Найменування <>
Довідники.КомплектЗахисту.Базовий. Найменування) Тоді
            Рег =
РегістриВідомостей.УстановкаЦінКомплектЗахисту;
            Вибірка = Рег.Вибрати();
            Доки Вибірка.Наступний() Цикл
                Якщо (Вибірка.КомплектЗахисту. Найменування
= Об'єкт.КомплектЗахисту. Найменування) Тоді
                    кз = Вибірка.Ціна;
                КінецьЯкщо;
            КінецьЦикла;
        КінецьЯкщо;

        ПрогнозованаВартість = Об'єкт.КількістьСамокатів *
зп * Об'єкт.КількістьГодин + кз;

        тд = (Об'єкт.Дата - Об'єкт.ДатаПочаткуОренди) / 3600;
        Фактична Вартість = Об'єкт.КількістьСамокатів * (зп
* тд) + кз;

        Об'єкт.Вартість = ФактичнаВартість -
ПрогнозованаВартість;

        Якщо ФактичнаВартість > ПрогнозованаВартість Тоді
            Повідомити ("Потрібно доплатити: "+
Макс(Об'єкт.Вартість, -Об'єкт.Вартість)+" грн!");
        Інакше
            Повідомити("Потрібно повернути:
"+Макс(Об'єкт.Вартість, -Об'єкт.Вартість)+" грн!");
        КінецьЯкщо;

```

КінецьПроцедури

&НаСервері

Процедура ПриЗаписуНаСервері (Відмова, ПоточнийОб'єкт,
ПараметриЗапису)

КінецьПроцедури

&НаКлієнті

Процедура Друк (Команда)

ТабДок = Новий ТабличнийДокумент;

ДрукНаСервері (ТабДок);

ТабДок.Показати ("Лист повернення");

КінецьПроцедури

&НаСервері

Процедура ДрукНаСервері (ТабДок)

змін зп;

//отримання макету

Макет =

Документи.Повернення.ОтриматиМакет ("ЛистПовернення");

//Отримання областей

ОбластьШапка = Макет.ОтриматиОбласть ("Шапка");

ОбластьШапкаЗамовник =

Макет.ОтриматиОбласть ("ШапкаЗамовник");

ОбластьШапкаТабл =

Макет.ОтриматиОбласть ("ШапкаТаблиці");

```

        ОбластьРядкиТабл                                     =
Макет.ОтриматиОбласть ("РядкиТаблиці");
        ОбластьПідвал = Макет.ОтриматиОбласть ("Підвал");
        ОбластьЗагальнийТекст                                 =
Макет.ОтриматиОбласть ("ЗагальнийТекст");

        //заповнення параметрів і вивід у табл док
        ОбластьШапка.Параметри.Номер = Об'єкт.Номер;
        ТабДок.Вивести (ОбластьШапка);

        ОбластьШапкаЗамовник.Параметри.Студент              =
Об'єкт.Студент.Найменування;
        ОбластьШапкаЗамовник.Параметри.Група                =
Об'єкт.Студент.НомерГрупи;
        ОбластьШапкаЗамовник.Параметри.телефон              =
Об'єкт.Студент.Телефон;
        ОбластьШапкаЗамовник.Параметри.ДатаПочаткуОренди    =
Об'єкт.ДатаПочаткуОренди;
        ОбластьШапкаЗамовник.Параметри.Дата = Об'єкт.Дата;
        ОбластьШапкаЗамовник.Параметри.Час                  =
Окр ((Об'єкт.Дата - Об'єкт.ДатаПочаткуОренди) / 3600, 2);

        ТабДок.Вивести (ОбластьШапкаЗамовник);

        ТабДок.Вивести (ОбластьШапкаТабл);

        ОбластьРядкиТабл.Параметри.КомплектЗахисту          =
Об'єкт.КомплектЗахисту.Найменування;
        кз = 0;

```

```

Рег =
РегістриВідомостей.УстановкаЦінКомплектЗахисту;
Вибірка = Рег.Вибрати();
Доки Вибірка.Наступний() Цикл
    Якщо (Вибірка.КомплектЗахисту.Найменування =
Об'єкт.КомплектЗахисту.Найменування) Тоді
        кз = Вибірка.Ціна;
        КінецьЯкщо;
    КінецьЦикла;
    ОбластьРядкиТабл.Параметри.ЦінаКомплектуЗахисту = "
"+кз;
    ОбластьРядкиТабл.Параметри.ВартістьКомплектуЗахисту
= " "+кз;
    Сообщить ("Тест"+кз);
    ОбластьРядкиТабл.Параметри.Самокат =
Об'єкт.Самокат.Найменування;
    ОбластьРядкиТабл.Параметри.КількістьСамокатів =
Об'єкт.КількістьСамокатів;

Рег = РегістриВідомостей.УстановкаЦінСамокатів;
Вибірка = Рег.Вибрати();
Доки Вибірка.Наступний() Цикл
    Якщо (Вибірка.Самокат.Найменування =
Об'єкт.Самокат.Найменування) Тоді
        зп = Вибірка.Ціна;
        КінецьЯкщо;
    КінецьЦикла;
    ОбластьРядкиТабл.Параметри.ЦінаСамокату = зп;

```

```

        ОбластьРядкиТабл.Параметри.ВартістьСамокату =
Об'єкт.Вартість-кз;
        ТабДок.Вивести(ОбластьРядкиТабл);

        ОбластьПідвал.Параметри.Вартість = Об'єкт.Вартість;
        Якщо Об'єкт.Вартість < 0 Тоді
            ОбластьПідвал.Параметри.ТекстПов = "Повернення
раніше! Конпенсації -> ";
            ОбластьПідвал.Параметри.Сума =
Макс(Об'єкт.Вартість, -Об'єкт.Вартість);

            ОбластьЗагальнийТекст.Параметри.ТекстПов =
"Повернення раніше! Конпенсації -> ";
            ОбластьЗагальнийТекст.Параметри.Сума =
Макс(Об'єкт.Вартість, -Об'єкт.Вартість);
        Інакше
            ОбластьПідвал.Параметри.ТекстПов = "Повернення
з затримкою! Борг -> ";
            ОбластьПідвал.Параметри.Сума =
Макс(Об'єкт.Вартість, -Об'єкт.Вартість);
            ОбластьЗагальнийТекст.Параметри.ТекстПов =
"Повернення з затримкою! Борг -> ";
            ОбластьЗагальнийТекст.Параметри.Сума =
Макс(Об'єкт.Вартість, -Об'єкт.Вартість);
        КінецьЯкщо;
        ТабДок.Вивести(ОбластьПідвал);

        ОбластьЗагальнийТекст.Параметри.Студент =
Об'єкт.Студент.Найменування;
        ОбластьЗагальнийТекст.Параметри.Група =
Об'єкт.Студент.НомерГрупи;

```

```
        ОбластьЗагальнийТекст.Параметри.телефон      =  
Об'єкт.Студент.Телефон;  
        ОбластьЗагальнийТекст.Параметри.Дата = Об'єкт.Дата;  
        ТабДок.Вивести (ОбластьЗагальнийТекст) ;  
КінецьПроцедури
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»

Кафедра програмних засобів

Дипломна работа

на тему:

Програмне забезпечення автоматизованої системи прокату самокатів

Виконав
ст. гр. КНТз – 119

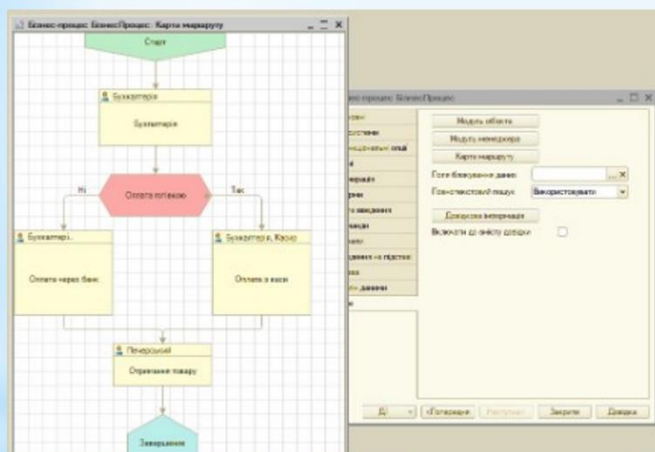
Данило ЛІСІЧКІН

Керівник
К.Т.Н., доц.

Анжеліка ПАРХОМЕНКО

Рисунок Б.1 – Слайд 1

Актуальність роботи



Приклад схеми бізнес -процесу



Узагальнений алгоритм роботи BPMS

Рисунок Б.2 – Слайд 2

Актуальність роботи

Критерій	SAP	Bpm'online	Bizagi
Налаштування процесів бізнес-користувачів без програмування	+	+	+
Установка свій сервер	+	+	+
Підтримка нотатції BPMN для моделювання бізнес-процесів	+	+	+
Наявність візуального редактора для моделювання бізнес-процесів	+	+	+
Єдине середовище розробки та виконання бізнес-процесів	+	-	+
Можливість вивантажувати регламенти щодо реалізованих бізнес-процесів	+	-	+
Наявність механізмів налагодження процесів	+	+	+
Визначення поточного кроку виконання процесу у реальному часі	+	+	+
Можливість покращення процесів	+	+	+
Контроль ефективності працівників	+	+	+
Інтеграція з BAS	+	+	+
Інтеграція з IP-телефонією, email-сервісами	+	+	+
Наявність мобільної версії	+	+	+

4

Рисунок Б.3 – Слайд 3

Мета та завдання роботи

Мета роботи: дослідження та розробка автоматизованої системи обліку електросамокатів для підвищення ефективності процесів видачі - прийняття обладнання та інвентаря та проведення фінансових розрахунків.

Завдання роботи:

- дослідити структурні та функціональні особливості платформи BAS;
- виконати аналіз вимог до автоматизованої системи обліку прокату самокатів;
- розробити структурну та функціональну схеми автоматизованої системи;
- обґрунтувати вибір інструментарію розробки;
- розробити алгоритм функціонування автоматизованої системи;
- виконати конструювання програмного забезпечення та перевірку його працездатності на платформі BAS.

4

Рисунок Б.4 – Слайд 4

Дослідження особливостей використання системи BAS

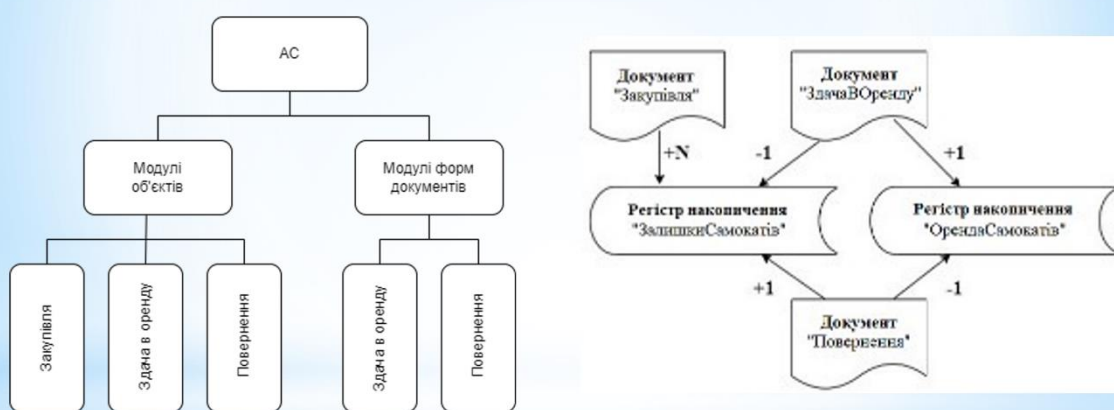


Компоненти та режими BAS

5

Рисунок Б.5 – Слайд 5

Проектування АС



Структурна та функціональна схеми АС

6

Рисунок Б.6 – Слайд 6

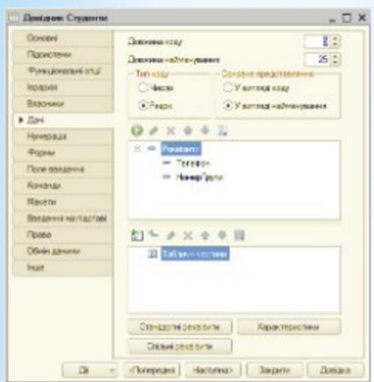


Рисунок Б.7 – Слайд 7

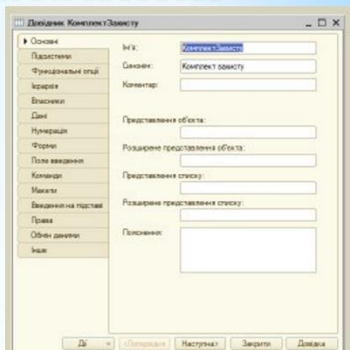


Рисунок Б.8 – Слайд 8

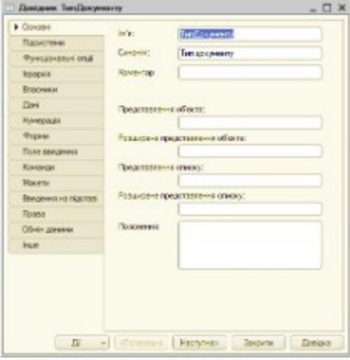
Результат роботи: Довідники



Довідник «Студенти»



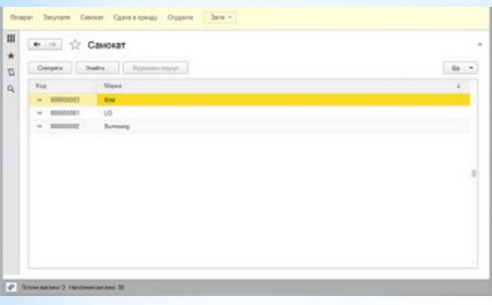
Довідник «КомплектЗахисту»



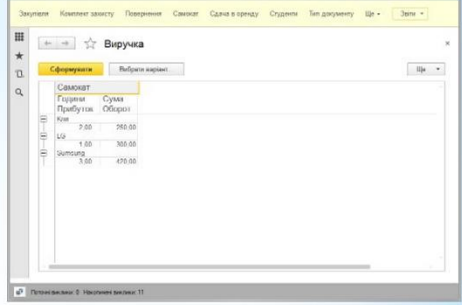
Довідник « ТипДокументу »

Рисунок Б.9 – Слайд 9

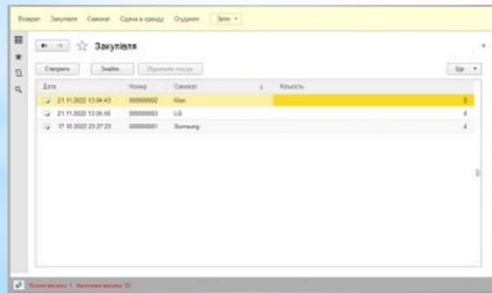
Результат роботи: Звіти



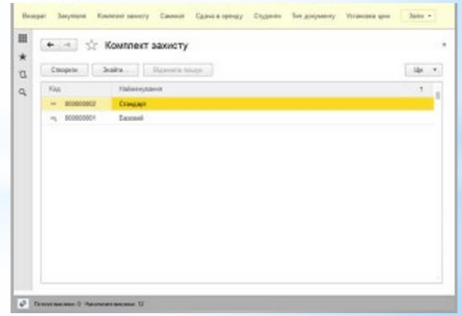
Звіт «Самокат»



Звіт «Виручка»



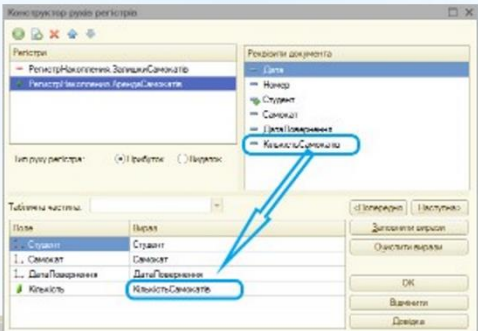
Звіт «Закупівля»



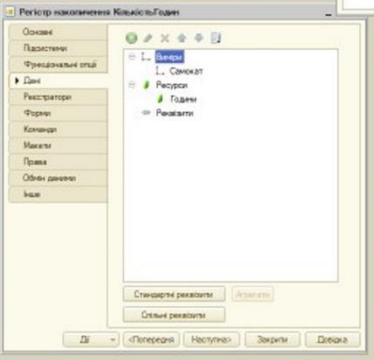
Звіт «КомплектЗахисту»

Рисунок Б.10 – Слайд 10

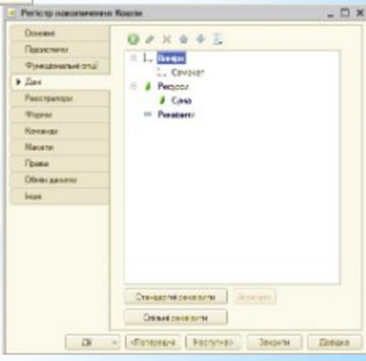
Результат роботи: Регістри



Обробка «КількостіСамокатів» у реєстрі накопичення



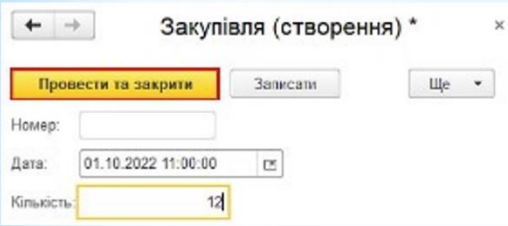
Регістр «КількістьГодин»



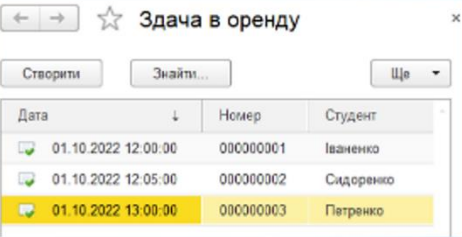
Регістр «Кошти»

Рисунок Б.11 – Слайд 11

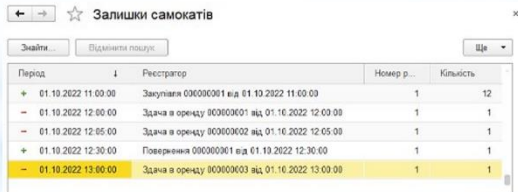
Результат роботи: Обробка



Фіксування закупівлі електросамокатів



Реєстрація студентів, що орендували самокат



Перевірка реєстру накопичення «ЗалишкиСамокатів»

Рисунок Б.12 – Слайд 12

Висновки

Результатом виконання дипломної роботи є розробка системи обліку, яка забезпечує автоматизацію бізнес-процесу прокату електросамокатів, що включає етапи закупівлі електросамокатів, здачі самокатів в оренду студентам студмістечка, а також повернення самокатів студентами.

Розробку виконано на платформі BAS з використанням вбудованої скриптової мови програмування та мови запитів системи.

Основою розробленої АС є інформаційна база, що містить інформацію про самокати, їх марки та кількість; про студентів, їх ПІБ, документи застави, телефон, групу, результати фінансових розрахунків, що пов'язані з витратами на закупівлю самокатів та прибутками від їх оренди.

Програмний застосунок надає можливість контролювати процеси видачі-прийняття самокатів та захисного спорядження, а також розрахувати вартість оренди з урахуванням терміну використання. У разі перевищення вказаного терміну оренди, система формує повідомлення.

Розроблене програмне забезпечення дозволяє підвищити ефективність процесів видачі-прийняття обладнання та інвентаря та проведення фінансових розрахунків пункту прокату.

В подальшій роботі можливо розширити функціональність АС, зокрема шляхом додавання процесу страхування орендаря.

Рисунок Б.13 – Слайд 13