

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

**на тему «Інформаційна технологія прогнозування завантаження
банкоматів готівкою»**

Виконав: студент 2 курсу,
групи 1КН-18 м
спеціальності 122 «Комп'ютерні науки»
Кусяка Д. С.

Керівник: к.т.н., доц. Колесницький О.К.

Рецензент: к. т. н., доц. Войтко В. В.

Вінниця
2019

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф.. Яровий А.А.

(підпис)
“ ” _____ 2019 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.007.18.000.ПЗ

Магістранта групи 1КН-18м Кусяки Дмитра Сергійовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія прогнозування завантаження банкоматів готівкою»

Вхідні дані: Середовище розробки Visual Studio 2017; система управління базами даних MS SQL Server 2008; кількість банкоматів – не менше 5; період прогнозування – не менше 1 доби; об'єктно – орієнтована мова програмування; операційна система сімейства Windows;

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Структура інформаційної технології прогнозування завантаження банкоматів готівкою, схема бази даних; представлення джерела даних, робочі вікна програми прогнозування завантаження банкоматів готівкою, результати роботи програми прогнозування завантаження банкоматів готівкою.

2. Текстова (пояснювальна записка): вступ, аналіз предметної області прогнозування завантаження банкоматів готівкою, розробка інформаційної технології прогнозування завантаження банкоматів готівкою на основі нейронної мережі, програмна реалізація інформаційної технології прогнозування завантаження банкоматів готівкою, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня інформаційних технологій прогнозування завантаження банкоматів готівкою. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Побудова математичних моделей функціонування нейронної мережі			Математичні моделі, розділ 2
3	Практичне застосування та оцінка ефективності розроблених моделей			розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ О. К. Колесницький
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц., доц. каф. ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ М. В. Бальзан
ініціали та прізвище

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц., доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
 _____ В. В. Войтко
ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 _____ О. К. Колесницький
ініціали та прізвище

“ ____ ” _____ 20__ р.

Завдання отримав магістрант _____ Д.С. Кусяка
(підпис) ініціали та прізвище

“ ____ ” _____ 20__ р.

АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології та програмного забезпечення для прогнозування завантаження банкоматів готівкою за допомогою інтелектуальних технологій . Були розглянуті основні методи прогнозування завантаження банкоматів готівкою, визначені їх переваги і недоліки, та на основі цього був зроблений вибір на користь методів дерев рішень та часових рядів. Було розроблено структуру та функціональну модель інформаційної технології, які слугували базисом для розробки програмного забезпечення. Програмне забезпечення написане на мові програмування C# в середовищі розробки MS Visual Studio Community 2017. З використанням для бази даних технології OLEDB та характеризується підвищеною щонайменше на 24% точністю прогнозування завантаження банкоматів готівкою.

ABSTRACT

The master's degree work is devoted to the development of information technology and software for predicting ATM cash withdrawal using intelligent technology. The basic methods of forecasting ATM cash loading were considered, their advantages and disadvantages were determined, and based on this, the decision trees and time series methods were selected. A structure and a functional model of information technology were developed to serve as a basis for software development. The software is written in C # programming language in the MS Visual Studio Community Development Environment 2017, using OLEDB technology for the database and has at least 24% higher ATM cash prediction accuracy.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ	12
1.1 Процес прогнозування завантаження банкоматів готівкою	12
1.1.1 Класифікація методів прогнозування.	13
1.1.2 Специфіка прогнозування завантаження банкоматів.....	16
1.2 Аналіз аналогів та їх характеристик.....	16
1.3 Постановка задачі.....	18
1.4 Висновок.....	19
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ.....	20
2.1 Аналіз процесу постачання банкоматної мережі готівкою.....	20
2.2 Формування вимог до інформаційної технології.....	21
2.3 Розробка технології аналітичної обробки даних.....	22
2.4 Прогнозування на основі методу дерев прийняття рішень.....	24
2.5 Прогнозування на основі методу часових рядів.....	25
2.6 Розробка структури інформаційної технології прогнозування завантаження банкоматів готівкою	28
2.7 Розробка функціональної моделі процесу прогнозування завантаження банкоматів готівкою.....	34
2.8 Висновок.....	36
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ.....	37
3.1 Обґрунтування вибору мови та середовища програмування	37
3.2 Вибір середовища програмування бази даних	39
3.3 Проектування бази даних прогнозування завантаження банкоматів	45
3.4 Розробка інтерфейсу програми.....	47
3.5 Проектування OLAP-куба	49
3.6 Реалізація функціональних частин програми.....	52
3.7 Тестування роботи програми прогнозування завантаження банкоматів готівкою.....	55
3.7.1 Методи тестування.....	57

3.7.2 Інструкція тестування програми прогнозування завантаження банкоматів готівкою.....	58
3.8 Аналіз результатів роботи програми прогнозування завантаження банкоматів готівкою.....	59
3.9 Висновки	61
4 ЕКОНОМІЧНА ЧАСТИНА.....	62
4.1 Оцінювання комерційного потенціалу розробки.....	62
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.	63
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.	66
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності..	68
4.5 Висновок	71
ВИСНОВКИ.....	73
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
Додаток А Інструкція користувача.....	77
Додаток Б Лістинг програми.....	80
Додаток В Графічна частина.....	86

ВСТУП

Актуальність. Забезпечення стабільної узгодженої роботи системи банкоматів є одним з факторів, що визначають економічну стабільність і добробут громадян. Зберігання та обробка інформації про стан банкоматів дозволяє автоматизувати і оптимізувати процеси, пов'язані з урахуванням, розподілом, переміщенням грошових коштів. Для отримання своєчасної достовірної інформації з можливістю її подальшої обробки та використання отриманих даних в процесах прийняття рішень виникає необхідність застосування програмного забезпечення прогнозування завантаження банкоматів.

В умовах накопичення великих обсягів даних і стрімкого розвитку засобів і методів їх обробки актуальність і практичну значимість набуває завдання розробки програмного забезпечення прогнозування завантаження банкоматів, рішення якої особливе значення набуває при необхідності економії фінансових ресурсів, а також для раціонального використання людських ресурсів.

При вирішенні поставлених завдань буде застосовано методи моделювання системи на основі технології SADT, метод проектування бази даних «сутність-зв'язок», заснований на побудові ER-діаграм предметної області, а також засоби програмування клієнтської частини автоматизованої системи, засновані на технології OleDb, що надає можливість звернення до бази даних за допомогою SQL-запитів, а також методи аналізу даних на основі технології OLAP-кубів.

Застосування інформаційної технології прогнозування завантаження банкоматів забезпечує:

- внесення, зберігання і редагування інформації про банкомати;
- надання та аналіз історичних даних про інкасацію;
- надання оперативної поточної і прогнозованої інформації щодо завантаження банкоматів;
- формування різноманітних звітів.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення точності прогнозування завантаження банкоматів готівкою програмними засобами за рахунок застосування інтелектуальних технологій.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз проблеми розв'язання задачі прогнозування завантаження банкоматів готівкою;
- розглянути існуючі методи вирішення задачі прогнозування завантаження банкоматів готівкою та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити математичну модель прогнозування завантаження банкоматів готівкою;
- сформулювати стадії інформаційної технології, розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес прогнозування завантаження банкоматів готівкою з використанням інтелектуальних технологій.

Предмет дослідження – інформаційна технологія та програмні засоби прогнозування завантаження банкоматів готівкою з використанням інтелектуальних технологій та точність їх роботи.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, інтелектуального аналізу даних, теорії баз даних та знань для реалізації інформаційної технології прогнозування завантаження банкоматів готівкою, методи математичної статистики для розробки процесу прогнозування завантаження банкоматів готівкою та обрахунків результатів експериментів із програмним засобом, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія прогнозування завантаження банкоматів готівкою, яка відрізняється спільним використанням технологій OLAP і Data Mining, що дозволило підвищити точність прогнозування завантаження банкоматів готівкою.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення прогнозування завантаження банкоматів готівкою.

Запропонована інформаційна технологія сприяє підвищенню точності процесу прогнозування завантаження банкоматів готівкою:

- розроблено алгоритм роботи програмного забезпечення прогнозування завантаження банкоматів готівкою на основі технологій OLAP і Data Mining;
- розроблено програмні засоби для прогнозування завантаження банкоматів готівкою на основі технологій OLAP і Data Mining.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології прогнозування завантаження банкоматів готівкою. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу прогнозування завантаження

банкоматів готівкою на основі технологій OLAP і Data Mining та методів підвищення достовірності [1,2].

Апробація результатів роботи. Результати досліджень апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи-2019», Вінниця, 2019, м. Вінниця [1] та на конференції «Молодь в науці: дослідження, проблеми, перспективи-2020», Вінниця, 2019, м. Вінниця [2].

Публікації. За результатами досліджень опубліковано тези доповіді на науково-технічній конференції [1,2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ

1.1 Процес прогнозування завантаження банкоматів готівкою

Банкомат – це механічний електронний апарат самообслуговування, головне призначення якого, на відміну від платіжних терміналів, полягає у видачі готівкових коштів. Крім цієї операції банкомат виконує функції по оплаті комунальних послуг, податкових платежів, штрафів, кредитів, поповнення мобільного телефону, переказу коштів з картки на картку та ін. При цьому, головна вимога, що пред'являється до клієнта - це наявність банківської карти. Одним з факторів забезпечення якісної роботи мережі банкоматів є ефективна робота відповідних служб щодо своєчасного та ефективного проведення інкасації банкоматів. Важливу роль у здійсненні інкасації відіграє зручність і ефективність засобів автоматизації, які обслуговують процедуру інкасації [2,3].

Фінансовий період – це інтервал часу між двома інкасаціями. Закриття фінансового періоду для банкомату оператор процесингового центру здійснює вручну, на основі повідомлення інкасатором в телефонному режимі про операції проведеної інкасації. Для визначення значення фінансового періоду виникає необхідність контролю показників функціонування банкоматів, на основі чого виникає можливість прогнозування завантаження банкоматів, яке залежить від багатьох факторів, таких як географічне розташування, календарний період роботи, характеристика банку, який обслуговує даний апарат. На основі зазначених факторів при раціональній організації процесів вилучення, збереження і обробки даних можливо розробити програмне забезпечення, основною функцією якого є прогнозування стану банкоматів в певний період, що забезпечить своєчасну реалізацію процесів інкасації та дозволить визначити необхідний фінансовий період для кожного банкомату в мережі [3].

Загальна схема завантаження банкоматів складається з наступних етапів:

1. Накопичення даних по банкоматах (стану банкоматів, дані про транзакції).
2. Первинна обробка даних (обробка та узагальнення інформації про стани банкоматів і транзакції).
3. Прогнозування попиту на готівку в банкоматах з використанням методів аналітичної обробки даних.
4. Планування завантаження банкоматів на заданий період з урахуванням побудованого прогнозу.
5. Моніторинг поточного стану банкоматів і оперативне оповіщення про критичний стан.

1.1.1 Класифікація методів прогнозування.

Прогнозування є важливим етапом процесу планування завантаження банкоматів. Прогнозування - це метод, в якому використовуються як накопичений в минулому досвід, так і поточні припущення щодо майбутнього з метою його визначення.

Процес прогнозування можна представити у вигляді послідовності етапів (рис. 1.1).

Для прогнозування потрібне використання певних методів, які являють собою набір прийомів, що дозволяють зробити висновки щодо майбутнього об'єкта на основі аналізу історичних даних, а також з урахуванням зовнішніх і внутрішніх умов.

Методи прогнозування можна розділити на дві групи [4]:

1. Формалізовані методи, які засновані на фактичній інформації про стан об'єкта прогнозування (в минулому і сьогодні) і використовують математичний апарат.
2. Інтуїтивні методи, які використовуються тоді, коли неможливе використання математичного апарату, зважаючи на складність об'єкта

прогнозування. В даному випадку застосовуються оброблені судження експертів.

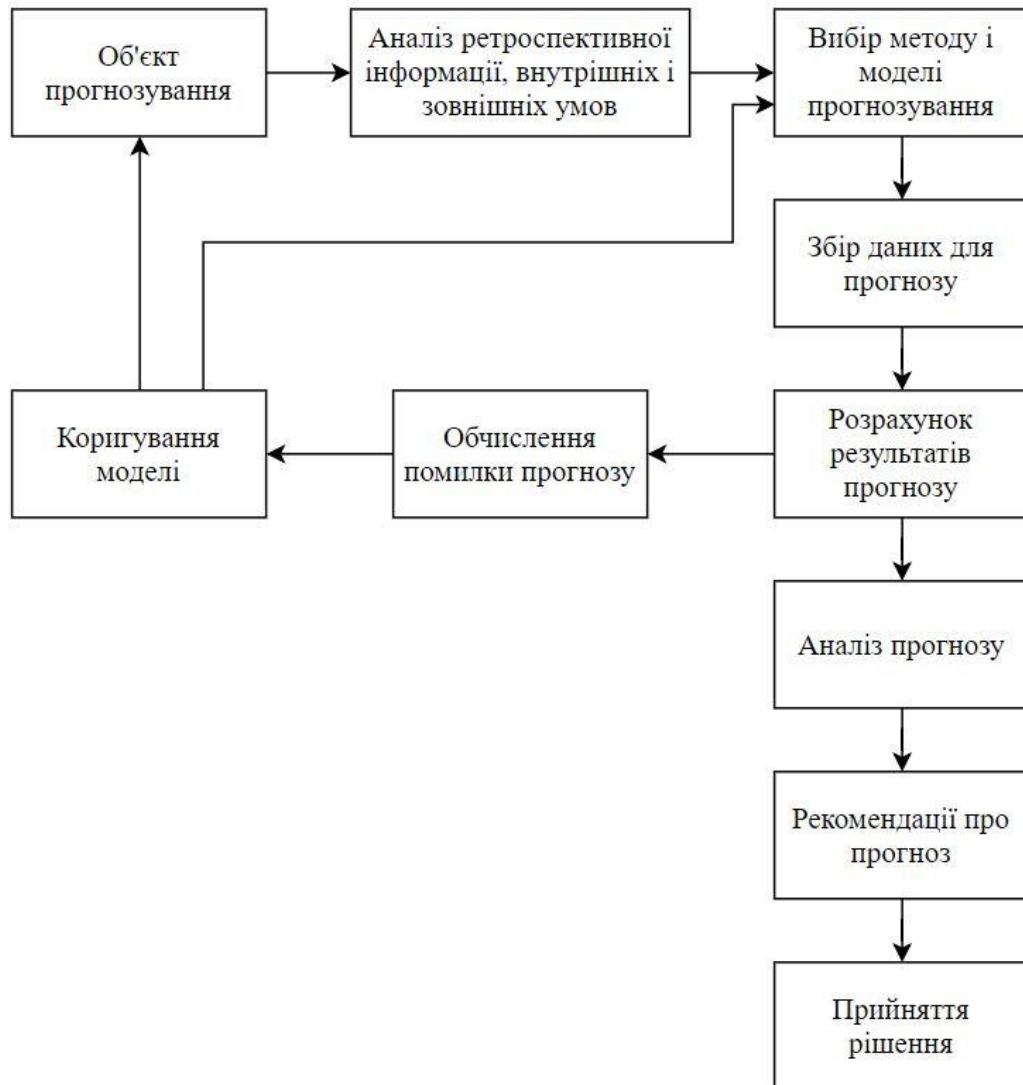


Рисунок 1.1 – Загальна схема процесу прогнозування

Розглянуті методи в свою чергу класифікуються за загальним принципом діяльності та способу обчислення прогнозних значень.

Моделі предметної області - математичні моделі прогнозування, для побудови яких використовуються залежності, властиві конкретній предметній області [5].

Класифікація методів і моделей прогнозування представлена на рис. 1.2.

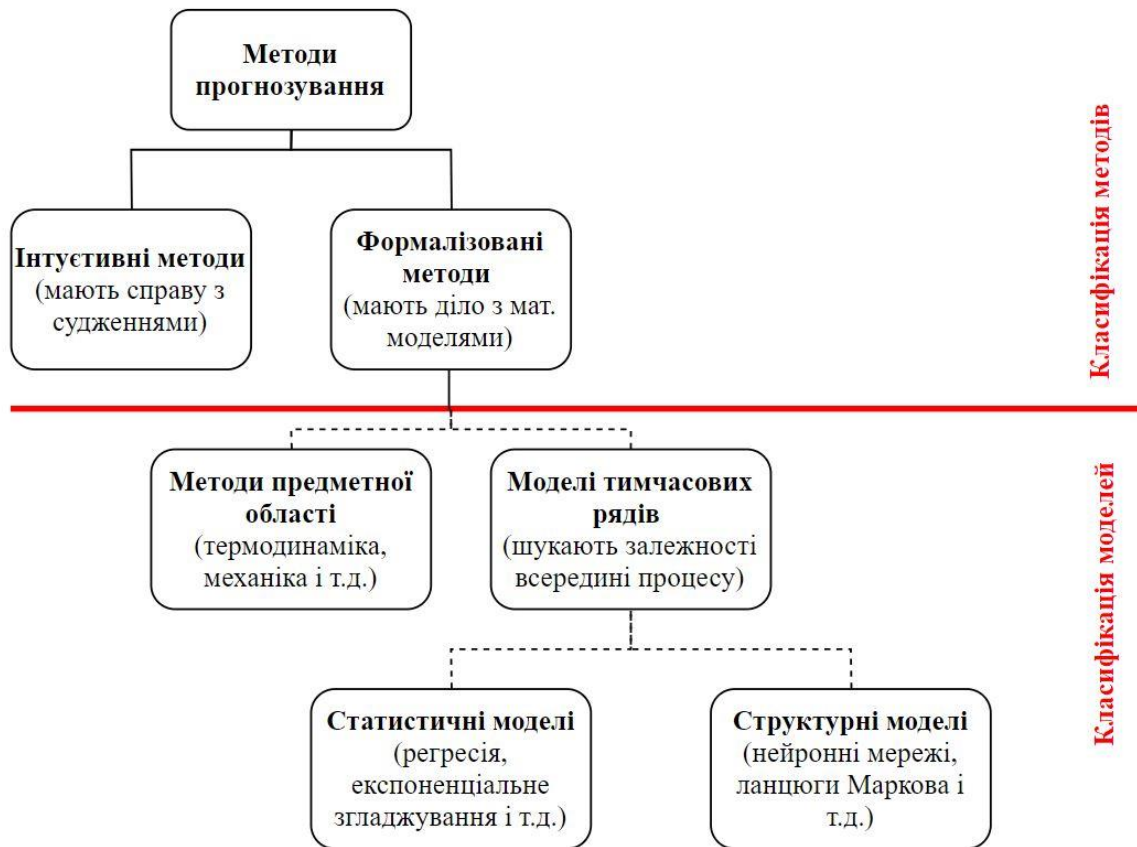


Рисунок 1.2 - Класифікація методів і моделей прогнозування

Щоденні витрати готівки в банкоматах можна розглядати як часовий ряд, тому для прогнозування завантаження необхідно використовувати моделі часових рядів. Моделі часових рядів - математичні моделі прогнозування, які використовуються для знаходження залежності майбутнього значення від історичної інформації про процес і по цій залежності обчислити прогноз. Ці моделі є універсальними для застосування в різних областях. Вони в свою чергу діляться на статистичні та структурні.

У статистичних моделях залежність майбутнього значення від минулого задається в вигляді деякого рівняння. У структурних моделях залежність майбутнього значення від минулого задається у вигляді певної структури і правил переходу по ній.

1.1.2 Специфіка прогнозування завантаження банкоматів

При прогнозуванні сум завантаження банкоматів необхідно враховувати наступні фактори:

- нерівномірність зняття готівки через банкомат в залежності від дня тижня;
- нерівномірність зняття готівки через банкомат в залежності від часу доби;
- місце розташування банкомату;
- різні купюрні набори в касетах і режим роботи співробітників касового центру;
- технічні несправності банкоматів;
- сезонність, вихідні та святкові дні.

На зазначені вище фактори впливає непередбачуваність поведінки клієнтів.

1.2 Аналіз аналогів та їх характеристик

У процесі аналітичного огляду програмного забезпечення, яке функціонує в області прогнозування завантаження банкоматів, доцільно зробити ряд уточнень [6]:

1. Прогнозування завантаження банкоматів може бути окремою частиною більш широкої задачі при вирішенні спеціалізованих питань, що стосуються сфери обслуговування клієнтів банків та питань функціонування банківської системи в цілому.

2. Як аналоги розроблюваного програмного забезпечення виступають автоматизовані інформаційні системи, які вирішують більш широкий спектр завдань в області організації діяльності банків та фінансових установ, що передбачають при цьому облік певного набору даних розглянутої предметної області.

3. Інформація про математичні і алгоритмічні засоби і методи, використовувані при реалізації розглянутих аналогів, також, як і характеристики програмного і апаратного забезпечення, в більшості випадків не знаходяться у відкритому доступі в силу державної або комерційної таємниці, а також особливостей дотримання прав інтелектуальної власності.

Одним із знайдених аналогів є ATM Cash Management Solution - рішення по прогнозуванню інкасацій та управління готівкою в банкоматних мережах, розробником якого є «Smart Technologies». До можливостей та особливостей системи Cash Management належить [7]:

- Мінімізація ручної праці при управлінні грошовими потоками в мережі банкоматів.
- Прогнозування інкасацій з високою точністю.
- Накопичення і врахування історичних даних по кожному банкомату при прогнозуванні інкасацій.
- Параметризація пікових навантажень на банкомати в дати виплат, передсвяткові та інші дні.
- Облік вартості інкасації, готівки, абстрактних коштів при плануванні завантажень.
- Управління службою інкасації, контроль інкасаторів, управління готівкою.
- Будь-який тип банкоматів, незалежно від виробника.
- Мережа банкоматів необмеженого розміру.
- Місця розташування банкоматів - будь-які.
- Мультивалютність, тобто можливість прогнозування і рекомендації відразу по декількох валютах.
- Підтримка 8-касетних банкоматів.
- Підтримка роботи cash-in і recycling пристроїв.
- Модуль оптимізації розміщення банкоматів.
- Модуль складання маршрутів інкасації.

- Гнучка, web-орієнтована, безпечна архітектура, що настроюється і адаптується для банкоматних мереж різних розмірів.

Ще одним рішенням в області прогнозування завантаження банкоматів є CashLab [8]. Даний програмний продукт вирішує такі завдання:

- проводить моніторинг рівня готівкових коштів в банкоматах;
- веде облік технічного статусу касет і банкомату;
- збирає історичні дані про завантаження і ведучих;
- інформує про критичні події.

Рішення функціонує на основі алгоритмів для прогнозування дати і розрахунку суми інкасації, враховує ліміти, а також проводить автоматичне балансування між вартістю засобів і вартістю проведення інкасацій.

Головним недоліком обох аналогів є невисока точність прогнозування завантаження банкоматів готівкою. Крім того, вони потребують значних витрат коштів на впровадження та супроводження, а також можуть не враховувати особливості функціонування певного банківського сектору

1.3 Постановка задачі

Розглянуті рішення мають закриті алгоритми, що не дозволяє вивчити методи, які використовуються в них для аналізу даних і прогнозування. Це обумовлює складність їх впровадження в уже існуючу систему в якості окремих аналітичних модулів для управління завантаженням банкоматів.

У зв'язку з цим необхідно розробити програмне забезпечення для прогнозування завантаження банкоматів.

Для вирішення поставленого завдання необхідно:

1. Провести розгляд процесу завантаження банкоматів і сформулювати вимоги до програмного забезпечення.
2. Здійснити вибір моделі прогнозування завантаження банкоматів.
3. Здійснити вибір показників оцінки якості результатів прогнозування.
4. Здійснити вибір середовища для реалізації програмного забезпечення.

5. Реалізувати базові алгоритми програмного забезпечення.

1.4 Висновок

Таким чином, в результаті аналізу предметної області та програмних засобів прогнозування завантаження банкоматів готівкою, можна зробити наступні висновки:

- функціонування програмних засобів прогнозування завантаження банкоматів дозволить вчасно проводити інкасації апаратів та автоматизувати визначення необхідного фінансового періоду для всіх банкоматів в мережі;
- існуючі аналоги контролю стану завантаження банкоматів потребують значних витрат коштів на впровадження та супроводження, а також можуть не враховувати особливості функціонування певного банківського сектору;
- для розробки програмного продукту доцільно використовувати системи управління реляційними базами даних, що дозволить забезпечити збереження достовірної інформації, а також програмний додаток для роботи користувача;
- для розробки програмного продукту сформовано завдання, яке враховує всі вимоги до певних областей функціонування програмного забезпечення прогнозування завантаження банкоматів.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ

2.1 Аналіз процесу постачання банкоматної мережі готівкою

Для формування вимог до програмного забезпечення необхідно розглянути процес постачання банкоматної мережі готівкою, який включає в себе прогнозування завантаження банкоматів.

Прогноз попиту на готівку можна побудувати, використовуючи історичні дані по станам банкоматів і транзакціях. На основі отриманого прогнозу формуються терміни інкасації та суми готівки для завантаження.

Процес підготовки і проведення інкасації банкоматів представлений на рис. 2.1

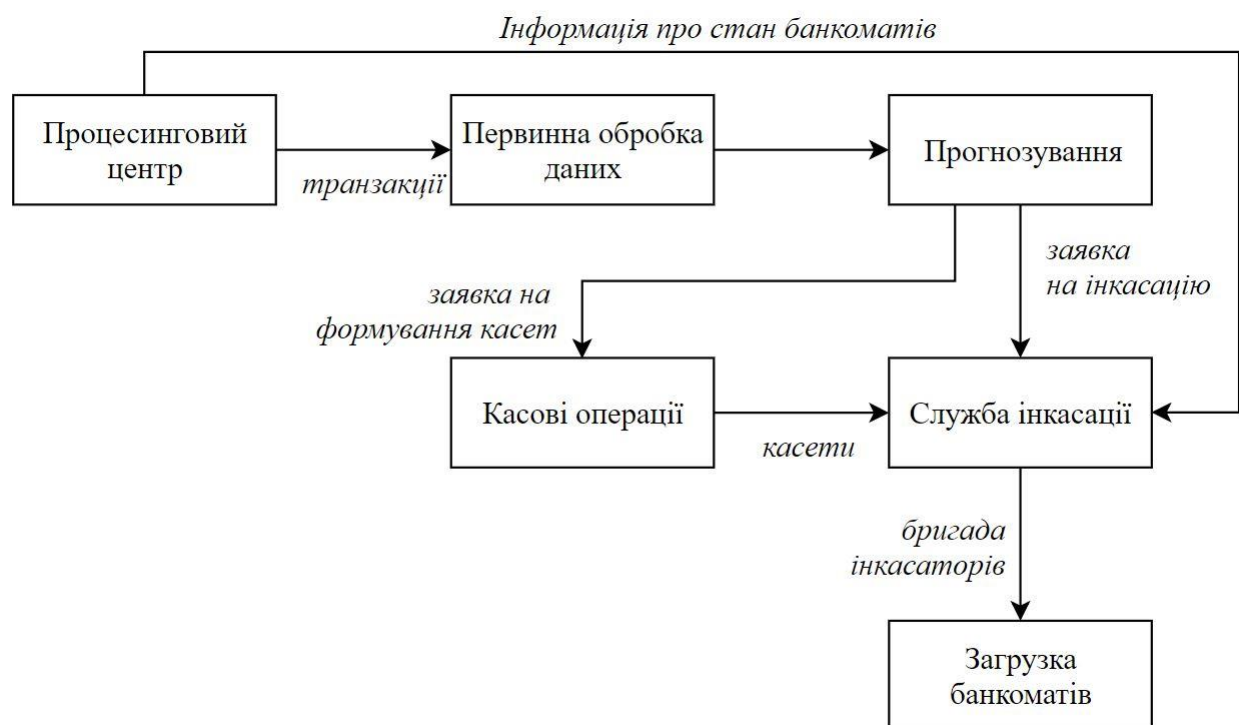


Рисунок 2.1– Схема проведення інкасації банкоматів

1. У процесинговому центрі відбувається накопичення даних по банкоматах (стану банкоматів, дані про транзакції).

2. Дані з процесингового центру надходять на первинну обробку (агрегування інформації про стани банкоматів і транзакціях).
3. Формується прогноз попиту на грошові кошти в банкоматах з використанням методів аналітичної обробки даних.
4. На основі отриманого прогнозу планується завантаження банкоматів на заданий період з урахуванням:
 - вартості обслуговування банкоматів;
 - режиму роботи грошового сховища банку і інкасаторських служб;
 - стану банкоматів і їх конфігурації;
 - розташування банкоматів.
5. Формується заявка службі інкасації на обслуговування банкоматів.
6. Служба інкасації отримує чек про стан лічильників згідно з даними процесингового центру.
7. Формується заявка касового центру на підготовку касет з готівкою.
8. Касовий центр формує касети з готівкою, одночасно з цим служба інкасації формує групу інкасаторів на певний маршрут.
9. У ході проведення інкасації банкомата вилучаються касети із залишками готівки і завантажуються нові касети, вводяться дані про кількість і номінал купюр в нових касетах, здійснюється друк чеків вивантаження і завантаження касет банкомату.

2.2 Формування вимог до інформаційної технології

На основі розглянутого процесу інкасації були сформульовані наступні вимоги до інформаційної технології прогнозування завантаження банкоматів та програмного забезпечення, що її реалізує:

1. Облік стану банкомату:
 - вибір банкоматів, які потребують завантаження.
2. Дані, що використовуються:

- дата;
 - місяць;
 - частина місяця;
 - день місяця;
 - день тижня;
 - історичні данні по сумах завантаження.
3. Прогнозоване значення:
- загальна сума готівки для завантаження банкомата.
4. Подання інформації по банкоматах:
- щоденне відображення даних;
 - загальна сума прогнозу готівки.
5. Функціонування в рамках підсистеми планування інкасації банкоматів.

2.3 Розробка технології аналітичної обробки даних

Засобом представлення і багатовимірного аналізу даних є OLAP-куб.

Проектування OLAP-куба проводиться на основі атрибутів таблиць сховища даних в наступній послідовності [9,10]:

1. Визначення та узгодження вимірювань. Вимірювання є критеріями для аналізу і визначають багатовимірний простір OLAP-куба. За рахунок фіксування значень одного або декількох вимірювань будуються зрізи. Кожен зріз являє собою запит до даних, що включає агрегації.
2. Вибір набору заходів. Заходи - це атрибути вимірювання OLAP-куба, зіставляються з атрибутами таблиць сховища даних.
3. Збереження результатів попередніх розрахунків в таблиці фактів.

У кубі існують відносини узагальнення і спеціалізації (roll-up / drill-down) по ієрархій вимірювань. Осередок високого рівня ієрархії може «спускатися» (drill-down) до осередку низького рівня і навпаки, «піднятися» (roll-up).

Одним з основних етапів побудови OLAP-куба є завдання функцій агрегування. Оскільки мета OLAP – створення багатовимірної структури даних, дані на рівнях, відмінних від фактичного, повинні бути попередньо агреговані. Для кожного вимірювання можливе завдання власної функції агрегації. Таким чином, в разі куба з n вимірами функція агрегації:

$$f(x) = (\{f_{1,1}, \dots, f_{1,k_1}\}, \dots, \{f_{n,1}, \dots, f_{n,k_n}\}) \quad (2.1)$$

де x - точка куба, а $f_{i,j}$ - j -а функція агрегування по i -ому виміру.

Для візуалізації даних, що зберігаються в OLAP-кубі, застосовуються табличні уявлення, що мають складні ієрархічні заголовки рядків і стовпців. Табличне представлення можна отримати, побудувавши двовимірний зріз OLAP-куба.

Загальну схему, яка відображатиме принцип багатовимірного аналізу даних, можна представити в наступному вигляді (рис. 2.2):

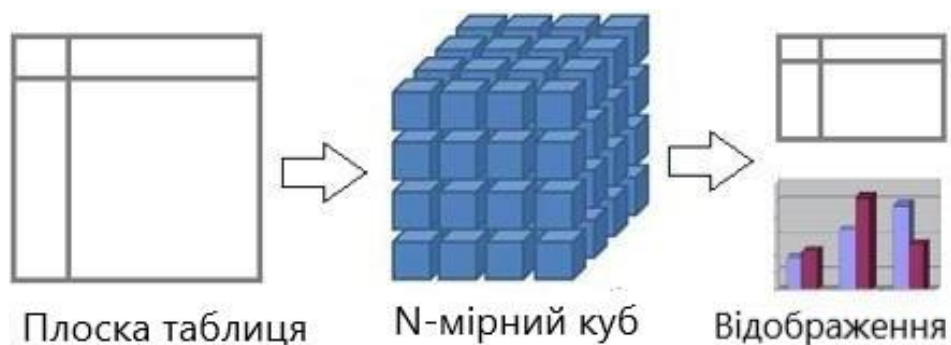


Рисунок 2.2 – Схема багатовимірного аналізу даних

Вона реалізує наступний алгоритм:

- 1) Побудова сховища даних шляхом подання вихідних даних у вигляді плоскої таблиці або результату іменованого запиту.
- 2) Витяг і перетворення інформації зі сховища даних в куб.
- 3) Відображення побудованого куба за допомогою крос-таблиць, діаграм, графіків.

У даній роботі використовується метод обробки даних, заснований на спільному використанні OLAP і технології Data Mining.

Одними з напрямків Data Mining є пошук у великому наборі даних прихованих тенденцій і рішення задач прогнозування. Процес Data Mining складається з наступних етапів [11]:

- 1) Створення структури і моделі за допомогою певного алгоритму.
- 2) Навчання моделі з використанням тренувальних даних (в тренувальних даних вказуються вихідні атрибути і атрибути, які будуть пророкуватися).
- 3) Подача на вхід моделі data mining невідомих атрибутів, які передбачаються, з подальшим визначенням їх значень.

2.4 Прогнозування на основі методу дерев прийняття рішень

Алгоритм дерева прийняття рішень використовується для прогнозування атрибутів шляхом встановлення залежності прогнозованого атрибута від незалежних вхідних змінних.

Принцип побудови дерева наступний. Дерево будується зверху вниз від кореня. В першу чергу необхідно виявити атрибут для перевірки в корені дерева. Для цього необхідно визначити, наскільки добре кожен атрибут розділяє дані на групи в залежності від прогнозованого атрибута. Далі для кожного значення обраного атрибута створюється гілка дерева, і набір даних розділяється далі відповідно до значення. Даний процес повторюється рекурсивно для кожної гілки. Також здійснюється перевірка критерію зупинки. Коли цей рекурсивний процес закінчується, формується дерево рішень.

Вихідний набір даних має наступний вигляд:

$$(x, Y) = (x_1, x_2, \dots, x_k, Y), \quad (2.)$$

Залежна змінна Y є цільовою неперервною змінною, значення якої необхідно прогнозувати. Вектор x складається з вхідних змінних x_1, x_2, x_3 і т. д., які використовуються для виконання цього завдання.

Неперервні змінні можуть бути спрогнозовані за допомогою формул лінійної регресії, заснованих на зазначених регресорах. Регресор - це неперервний вхідний атрибут, який використовується для моделювання неперервного прогнозованого атрибута лінійним чином.

Класична формула лінійної регресії:

$$Y = a + b * X + e \quad (2.3)$$

де Y - неперервний атрибут, який прогнозується;

X - регресор;

a, b - коефіцієнти регресії;

e - випадкова помилка.

Розглянутий алгоритм в кожному вузлі дерева містить формулу лінійної регресії [12].

2.5 Прогнозування на основі методу часових рядів

Для прогнозування завантаження банкоматів можна використовувати історичні дані, а на їх основі оцінювати майбутній попит на готівку в банкоматі для вибору оптимальної суми завантаження. Даному підходу відповідає алгоритм Data Mining часових рядів (Time-Series Data Mining).

Алгоритм часових рядів призначений для прогнозування неперервних значень. На відміну від алгоритму дерева прийняття рішень, модель часових рядів не вимагає додаткових відомостей для прогнозування. Побудова прогнозу здійснюється за допомогою закономірностей, що витягаються алгоритмом з вихідних даних.

Часовий ряд - це ряд послідовних значень, що характеризують зміни будь-яких параметрів досліджуваного процесу в часі. Часові ряди мають деякі важливі характеристики [12]:

- базовий рівень, який визначається як середнє значення часового ряду.
- тренд, який показує, як часові ряди змінюються від одного періоду до іншого.
- сезонні коливання. Деякі значення мають тенденцію збільшення або зниження в залежності від різних періодів часу, наприклад, від дня тижня чи місяця.

Таким чином, визначаючи тренд, накладаючи лінію тренда на базовий рівень і виявляючи сезонну складову, можна отримати модель для складання прогнозу значень:

Прогнозоване значення = Базовий рівень + Тренд + Сезонна складова.

Для аналізу часових рядів можна використовувати сукупність двох алгоритмів:

1. «Дерево авторегресії з перехресним прогнозуванням» (ARTXP), який орієнтований на використання при короткостроковому прогнозуванні і забезпечує найкращі значення прогнозу найближчого часового відрізка за останнім фактичним.

2. «Інтегровані ковзаючі середні авторегресії" (ARIMA), який використовується при довгостроковому прогнозуванні.

Алгоритм ARTXP істотно менш точний в довгострокових прогнозах в порівнянні з ARIMA.

ARIMA (або, АРИСС) модель також називається моделлю Бокса-Дженкінса і розшифровується, як модель авторегресії і інтегрованого змінного середнього.

У позначеннях Бокса і Дженкінса модель записується як ARIMA(p, d, q):

$$\Delta^d Y_t = c + \sum_{i=1}^p a_i \Delta^d Y_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t , \quad (2.4)$$

ARIMA (p, d, q) описує три основні процеси динамічного ряду:

1. AR (параметр p) - авторегресія.

Модель авторегресії - це модель часових рядів, в якій значення часового ряду в даний момент лінійно залежать від попередніх значень цього ж ряду. Авторегресійний процес порядку p визначається наступним чином:

$$Y_t = \sum_{i=1}^p a_i Y_{t-i} + \varepsilon_t , \quad (2.5)$$

де a_1, a_2, \dots, a_p - коефіцієнти авторегресії; ε_t - випадкова помилка.

2. Параметр d - число різниць, необхідних для приведення процесу до стаціонарності. Для моделі ARIMA необхідно, щоб ряд був стаціонарним (середнє є постійним, а вибіркові дисперсія і автокорреляція не змінюються в часі). Тому зазвичай необхідно брати різниці ряду до тих пір, поки він не стане стаціонарним (часто також застосовують логарифмічні перетворення для стабілізації дисперсії).

3. MA (параметр q) - ковзне середнє. На відміну від процесу авторегресії, в процесі ковзного середнього кожен елемент ряду схильний до сумарного впливу попередніх помилок. Модель ковзного середнього q-го порядку - модель часового ряду такого вигляду:

$$Y_t = \sum_{j=1}^q b_j \varepsilon_{t-j} , \quad (2.6)$$

де b_1, b_2, \dots, b_q - параметри моделі; ε_t - випадкова помилка.

ARIMA моделі охоплюють широкий спектр часових рядів, і при невеликій модифікації цих моделей дозволяють досить точно описувати тимчасові ряди з сезонністю.

2.6 Розробка структури інформаційної технології прогнозування завантаження банкоматів готівкою

Для створення програмного забезпечення прогнозування завантаження банкоматів готівкою був використаний комплексний підхід щодо використання технологій OLAP і Data Mining.

Дані, що підлягають аналізу, містяться в сховищі даних, що представляє собою певну кількість таблиць бази даних. Дані завантажуються в сховище даних з зовнішнього джерела після їх вилучення та перетворення.

На основі даних, отриманих зі сховища, побудований OLAP-куб. На рівні OLAP-куба під час процесингу відбувається агрегація показань приладів обліку за часом, послуг і адресами і будуються моделі Data Mining для прогнозування. Розроблювана інформаційна технологія включає в себе дві моделі інтелектуального аналізу даних, одна з яких використовує алгоритм дерева прийняття рішень (Microsoft Decision Trees), а інша модель - алгоритм часових рядів (Microsoft Time Series). Побудова моделей здійснюється в середовищі Business Intelligence Development Studio. Моделі здійснюють обчислення прогнозних значень обсягів завантаження банкоматів готівкою.

Структура інформаційної технології прогнозування завантаження банкоматів готівкою представлена на рис. 2.3.

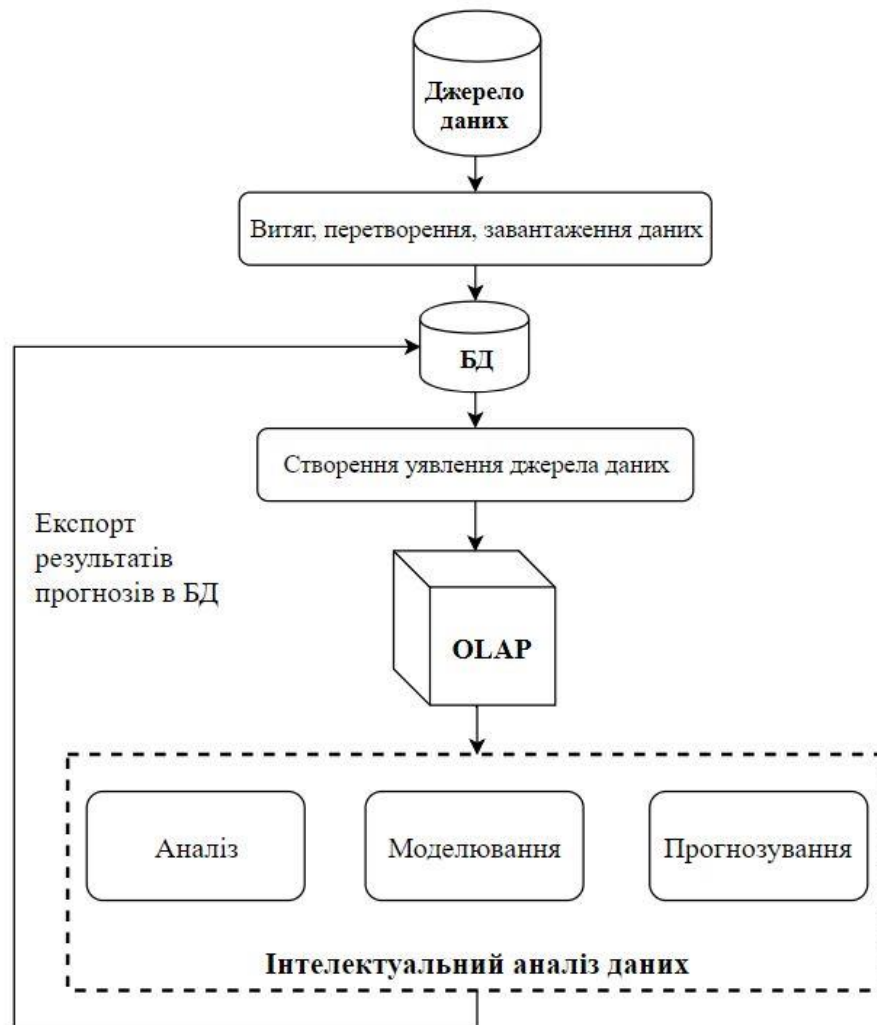


Рисунок 2.3 – Структура інформаційної технології прогнозування завантаження банкоматів готівкою

З міркувань безпеки, для того щоб з'явилася можливість підключення до примірника SQL Server, адміністратор баз даних (DBA) повинен створити для користувача реєстрацію (login). Реєстрацію можна пов'язати з ім'ям користувача і паролем в ОС Windows, і в цьому випадку вона буде називатися автентифікованою Windows. При такому варіанті реєстрації не потрібно буде вводити реєстраційне ім'я та пароль для підключення до SQL Server, оскільки користувач вже надав їх, коли реєструвався в ОС Windows.

Реєстрація може не залежати від імені користувача і пароля в Windows, в такому випадку її називають автентифікованою SQL Server (рис. 2.4).

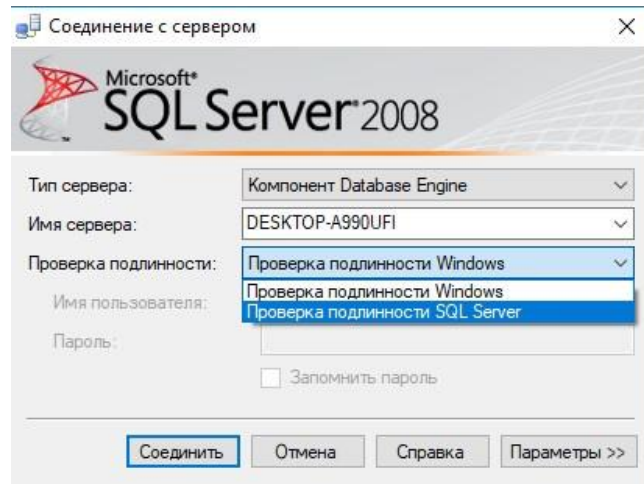


Рисунок 2.4 – Автентифікація SQL Server

При підключенні до сервера в разі реєстрації, автентифікованої SQL Server, є потреба пред'явити ім'я реєстрації і пароль (рис. 2.5).

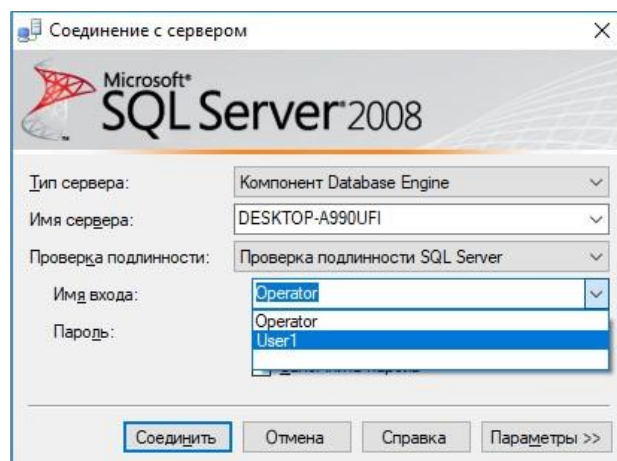


Рисунок 2.5 – Вибір імені входу

Для отримання доступу до бази даних, адміністратор баз даних повинен перевести реєстрацію в database user (користувача бази даних) для кожної бази даних, технічним обслуговуванням якої займається користувач.

База даних формується з файлу даних і файлу журналу транзакцій. При створенні бази даних, можна визначити різні властивості для кожного файлу, включаючи ім'я файлу, його місце розташування, початковий розмір, максимальний розмір і розмір його збільшення. У кожній базі даних повинен

бути хоча б один файл даних і один файл журналу транзакцій (як задано за замовчуванням в SQL Server). У файлах даних зберігаються дані об'єктів, а в файлах журналу транзакцій міститься інформація, яка необхідна SQL Server для підтримки транзакцій [13].

Незважаючи на те, що SQL Server може записувати в кілька файлів даних паралельно, в кожен момент часу він може писати тільки в один файл журналу. Отже, на відміну від файлів даних наявність декількох файлів журналу не принесе підвищення продуктивності. Додати новий файл журналу, можливо, доведеться, якщо диск, на якому зберігається файл, буде заповнений.

Файли даних об'єднані в логічні групи, названі файловими групами. Файлова група - це місце призначення при створенні об'єкта. Дані об'єктів розподілені між файлами, які належать заданій файловій групі. Файлові групи - це спосіб контролю фізичного місцезнаходження об'єктів. У бази даних повинна бути, як мінімум, одна файлова група, звана PRIMARY (основна), і, можливо, інші файлові групи. Файлова група PRIMARY містить основний файл даних (з розширенням mdf) і системний каталог бази даних. При необхідності в групу PRIMARY можна вставляти додаткові файли даних (з розширенням ndf). Призначені для користувача файлові групи містять тільки додаткові файли даних. Можна вказати, яку файлову групу помітити як групу за замовчуванням. Якщо в операторі створення об'єкта явно не задана файлова група, об'єкт створюється в файловій групі, яка використовується за замовчуванням.

База даних включає в себе схеми, а схеми містять об'єкти. Саме схеми служать контейнерами об'єктів, таких як таблиці, представлення, збережені процедури і ін. Можна управляти правами доступу на рівні схеми. Наприклад, можна надавати користувачеві права на виконання інструкції SELECT у схемі, дозволивши йому робити запит на вибірку даних всіх об'єктів схеми. Таким чином, безпека - один з важливих аспектів при визначенні способів організації об'єктів в схемах.

Крім того, схема - це простір імен, вона використовується як префікс в імені об'єкта. Якщо при посиланні на об'єкт опустити ім'я схеми, SQL Server

запустить процес дозволу імені схеми, який включає перевірку існування об'єкта в схемі користувача, яка застосовується за умовчанням, при його відсутності, подальшу перевірку існування об'єкта в схемі dbo. Рекомендується в програмному коді при посиланні на об'єкти завжди використовувати складені імена. З процесом дозволу імені об'єкта, якщо воно явно не визначено, пов'язані деякі незначні витрати. Крім того, якщо в різних схемах є об'єкти з одним і тим же ім'ям, можливий помилковий вибір.

Одна з найбільших переваг реляційної моделі полягає в тому, що цілісність даних - невід'ємна частина моделі. Цілісність даних, реалізована як компонент моделі, а саме як компонент визначень таблиці вважається декларативною цілісністю даних. Цілісність даних, яка втілюється в життя за допомогою програмного коду, такого як збережені процедури або тригери, називається процедурною цілісністю даних.

Тип даних і вибір допустимості чи недопустимості значень NULL ДЛЯ атрибутів і навіть сама по собі модель - приклади обмежень і уточнень, які забезпечують цілісність даних. В СУБД є можливість визначати обмеження при створенні таблиці, включивши їх в інструкцію CREATE TABLE, або після того, як таблиця вже створена за допомогою інструкції ALTER TABLE. Обмеження всіх типів за винятком обмеження значення за замовчуванням можна визначати як складові, тобто такі, що базуються на кількох атрибутах.

В СУБД MS SQL Server існують наступні типи обмежень [13]:

1) Обмеження за допомогою первинного ключа забезпечує унікальність рядків і, крім того, забороняє значення NULL в атрибутах з таким обмеженням. Будь-який унікальний набір значень в атрибутах з обмеженням може з'явитися в таблиці тільки один раз, іншими словами, тільки в одному рядку. Спроба поставити обмеження у вигляді первинного ключа в стовпці, що допускає значення NULL, буде відкинута СУБД. У кожній таблиці може бути тільки один первинний ключ. З встановленим обмеженням PRIMARY KEY можна бути впевненим в тому, що всі значення стовпця будуть унікальні і відомі. Спроба вставити або оновити рядок, в якому порушується це обмеження, буде

відкинута СУРБД і призведе до появи помилки. Для забезпечення унікальності логічного обмеження у вигляді первинного ключа SQL Server за кадром створить унікальний індекс. Унікальний індекс - це засіб, що застосовується SQL Server для забезпечення унікальності.

2) Обмеження типу UNIQUE забезпечують унікальність рядків, даючи можливість реалізувати концепцію альтернативних ключів, виходячи з реляційної моделі бази даних. На відміну від обмеження у вигляді первинних ключів можна визначити кілька обмежень unique в одній таблиці. Крім того, це обмеження підходить не тільки для стовпців, визначених як not null. У мові SQL стандарту ANSI є два види обмежень unique: одне з них допускає єдине значення null в стовпці з обмеженням unique, а інше допускає наявність множинних значень null. У SQL Server реалізований тільки перший вид обмеження. Як і в разі обмеження у вигляді первинного ключа, SQL Server створить за кадром унікальний індекс, що забезпечує на фізичному рівні логічне обмеження типу UNIQUE.

3) Зовнішній ключ забезпечує довідкову цілісність. Це обмеження визначається для набору атрибутів в таблиці, званої посилальною, і вказує на набір атрибутів потенційного ключа (первинного ключа або обмеження unique) у так званій довідковій таблиці. Слід зазначити, що посилальна і довідкова таблиці можуть бути однією і тією ж таблицею. Завдання обмеження у вигляді зовнішнього ключа - обмежити домен значень, допустимих в стовпцях зовнішнього ключа, тільки тими, які присутні в стовпцях довідкової таблиці.

Можливо визначати зовнішні ключі з діями, що дозволяють вдало завершити певні спроби (видалити рядки з довідкової таблиці або оновити довідкові атрибути потенційних ключів, коли в посилальній таблиці є пов'язані з ними рядки). Можна включити в визначення зовнішнього ключа варіанти ON DELETE I ON UPDATE з такими діями, як CASCADE, SET DEFAULT I SET NULL. Дія CASCADE означає, що операція (видалення або оновлення) буде послідовно застосовуватися до пов'язаних рядків. Наприклад, в разі ON DELETE CASCADE, при видаленні рядка з довідкової таблиці, СУРБД

видалить пов'язані з нею рядки з посилальної таблиці. Дії SET DEFAULT і SET NULL означають, що компенсуюча операція встановить атрибути зовнішнього ключа в пов'язаних рядках рівними значенням стовпця, використовуваного за замовчуванням, або значенням NULL відповідно. Слід враховувати, що незалежно від обраної дії посилальна таблиця буде мати тільки висячі або незв'язані рядки (orphaned rows) в разі виключення значень NULL.

4) Обмеження CHECK дозволяє визначити логічний вираз (предикат), якому повинен задовольняти рядок для того, щоб бути включеним в таблицю або зміненим. Обмеження засобами перевірки логічного виразу відкидає спробу вставити або оновити рядок, якщо обчислюване значення виразу є FALSE. Зміну буде прийнято, якщо обчислювальне значення предиката дорівнює TRUE або UNKNOWN (невідоме).

5) Обмеження типу default пов'язано з конкретним атрибутом. Воно являє собою вираз, що застосовується як значення за замовчуванням, коли при вставці рядка явно не задається значення атрибута.

2.7 Розробка функціональної моделі процесу прогнозування завантаження банкоматів готівкою

Моделювання системи є одним з найбільш важливих етапів, що визначають структуру і функціональні можливості системи, що розробляється. Модель автоматизованої системи дозволяє визначити і встановити взаємозв'язок структурних елементів, напрямок інформаційних потоків, параметри і характеристики, що впливають на подальшу поведінку системи при різному наборі координат вектора вхідних параметрів [13 – 15].

Серед безлічі методик і підходів до моделювання та проектування систем слід виділити метод SADT (IDEF0) (Structured Analysis and Design Technique), який є класичним методом процесного підходу до представлення даних і прийняття управлінських рішень. Моделювання процесу дозволяє провести комплексну структурування та оптимізацію представлення даних про процес,

уникнути невірною уявлення даних, а також можливих логічних суперечностей при поданні даних. Важливим елементом процесу моделювання системи є не тільки чітке позначення і характеристика її структурних елементів, а й визначення взаємозв'язку між ними, представленої у вигляді інформаційних потоків, що мають певний напрям, схильних до об'єднання або розгалуження для організації коректної і стабільної роботи системи в цілому [16].

При розгляді особливостей організації бізнес-процесів за методом SADT необхідно виділити основний процес, яким в даному випадку є функціонування автоматизованої системи обліку банкоматів, а також вхідні і вихідні параметри процесу, механізми впливу і керуючий вплив на процес, графічно зображувані у вигляді стрілок, спрямованих або до основного процесу, або в сторону від нього.

Вхідними параметрами моделі слід приймати дані, які грають роль «сировини» для вихідного продукту.

Модель реалізована за допомогою засобу ERWin Process Modeler, за допомогою якого створена діаграма IDEF0, зображена на рис. 2.6

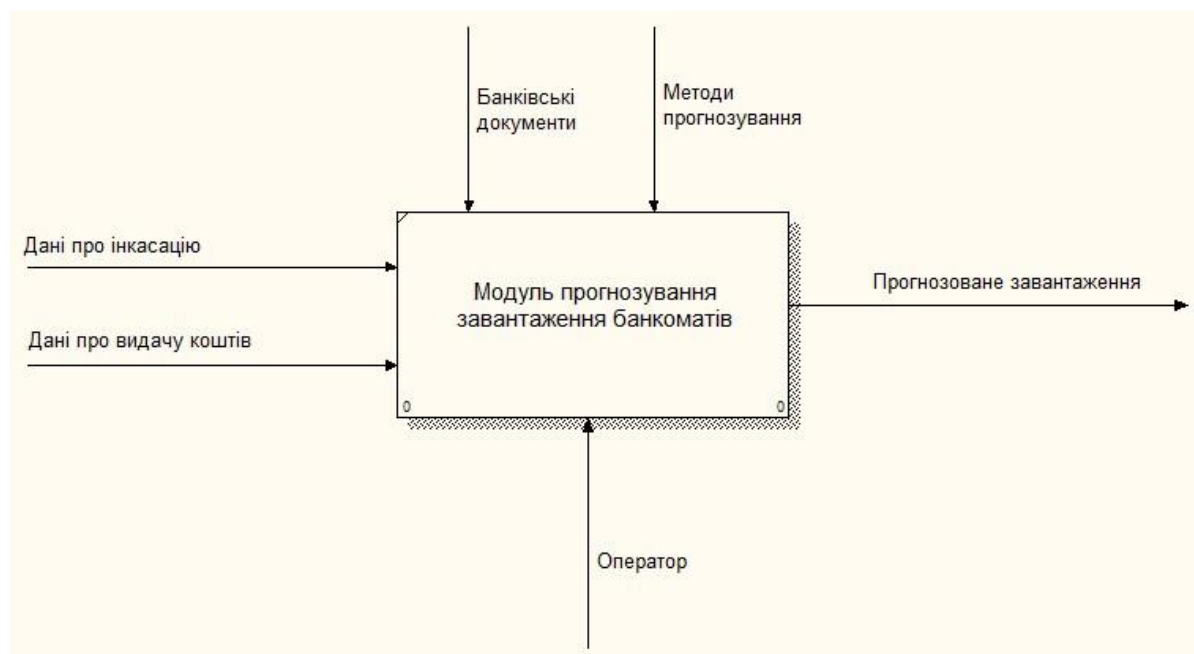


Рисунок 2.6– Функціональна модель IDEF0

2.8 Висновок

В даному розділі магістерської кваліфікаційної роботи було проаналізовано процес постачання банкоматної мережі готівкою, сформульовано вимоги до інформаційної технології прогнозування завантаження банкоматів готівкою, розроблено технологію аналітичної обробки даних, яка заснована на комплексному використанні методу дерев прийняття рішень та методу часових рядів, розроблено структуру інформаційної технології прогнозування завантаження банкоматів готівкою та розроблено функціональну модель процесу прогнозування завантаження банкоматів готівкою.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЗАВАНТАЖЕННЯ БАНКОМАТІВ ГОТІВКОЮ

3.1 Обґрунтування вибору мови та середовища програмування

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки ПО і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Microsoft Windows, Windows Mobile, Windows CE, NET Framework, Xbox, Windows Phone.NET Compact Framework і Microsoft Silverlight.

Набір легких середовищ розробки представляють собою урізану версію Visual Studio. На відміну від повної версії, кожне таке середовище налаштоване для якоїсь однієї мови програмування. Вона включає в себе невеликий набір інструментів, на відміну від повних версій: відсутній дизайнер класів і багато інших інструментів, а також підтримка плагінів і віддалених баз даних в дизайнерах даних. Компілятори в 64-бітний код також недоступні в Express редакціях.

Visual Studio 2017 містить підключену і розширену інтегровану середу розробки з інноваційними функціями для підвищення продуктивності розробника, додаткові інструменти з підтримкою розробки додатків для Windows 8.1, веб-розробки, а також поліпшення засобів налагодження та оптимізації виконуваного і керованого коду. Visual Studio 2017 також надає інноваційні засоби взаємодії розробників, розширені можливості ALM і різні нововведення для гнучкого управління портфелем проектів, забезпечення якості та DevOps.

Visual Studio 2017 змінює ставлення до процесу розробки, роблячи його цікавим і добре організованим. Новий продукт відрізняється підвищеною

швидкістю завантаження робочого середовища і відкриває розробникам доступ до конкретних проектів буквально в лічені секунди. Крім того, всі тривалі процеси виконуються в Visual Studio 2017 в фоновому режимі, що не уповільнює швидкість роботи середовища і не відволікає розробника від основних завдань.

Також робоче середовище Visual Studio 2017 має новий контекстно-залежний інтерфейс. Головна його особливість полягає в тому, що він пропонує розробникові тільки ті функції і інструменти, які йому потрібні на даному етапі роботи. Таким чином, панель інструментів не містить нічого зайвого і не ускладнює пошук потрібних функцій.

Середовище для розробки Visual Studio 2017 створено так, щоб допомагати розробнику і підказувати йому найкращі рішення. Для цього в продукт були включені спеціальні технології, які спрощують і прискорюють роботу. Наприклад, функція CodeClone шукає однакові по функціоналу коди і об'єднує їх у відповідні модулі для того, щоб розробник згодом міг скористатися вже готовим кодом при створенні нової програми.

Visual Studio 2017 дозволяє ефективно управляти повним життєвим циклом додатка від етапу його розробки до стадії експлуатації. Такий підхід передбачає командну роботу і участь в процесі великої кількості фахівців різного профілю: від архітекторів і розробників до дизайнерів і замовників проекту.

Консолідацію всіх циклів роботи над додатком і взаємодія робочої групи в Visual Studio 2017 забезпечує вдосконалене рішення TeamFoundationServer. З його допомогою всі учасники процесу розробки можуть відстежувати стан проекту, бачити його динаміку, контролювати терміни та отримувати аналітичні звіти про кожен період роботи.

Крім того, Visual Studio 2017 містить оновлені інструменти перевірки якості і працездатні додатку, що дозволяє тестувальникам моделювати поведінку додатку в момент його використання, а також вчасно виявляти недоліки в розробці. А функція PowerPointStoryBoarding дозволяє технічним

фахівцям представляти макет майбутнього рішення замовнику в зрозумілому для нього форматі.

Visual Studio 2017 [17,18] дозволяє створювати сучасні, конкурентоспроможні додатки для різних пристроїв і середовищ, найбільш ефективний продукт при інтеграції з платформою Windows, зокрема з новою операційною системою Windows 10, мобільною платформою Windows Phone і хмарної платформою Windows Azure.

Зараз це єдине середовище, яке дозволяє створювати рішення для додатка Windows Store. Visual Studio 2017 включає повний набір інструментів - від заготовлених шаблонів додатків до спеціального редактора інтерфейсів ExpressionBlend - і повністю автоматизує весь процес створення програми для Windows Store, включаючи етапи тестування і публікації в магазині додатків.

3.2 Вибір середовища програмування бази даних

Часто дані сприймаються як щось статичне, раз введене в базу даних і пізніше запитуване. Але в багатьох робочих середовищах насправді дані переходять з одного робочого середовища в інше і підлягають змінам на цьому шляху проходження [17] (рис. 3.1).

Як показано на рис. 3.1, спочатку дані вводяться в систему оперативної обробки транзакцій (OnLine Transactional Processing, OLTP). Головне завдання системи OLTP- введення, причому без складання звітів: транзакції в основному вставляють, оновлюють і видаляють дані [18]. Реляційна модель головним чином розрахована на системи OLTP, в яких нормалізована модель забезпечує як високу продуктивність введення даних, так і несуперечність даних. У нормалізованому середовищі кожна таблиця представляє один об'єкт, і надмірність мінімальна. Якщо потрібно щось змінити, треба змінити це щось тільки в одному місці, в результаті оптимізується продуктивність змін і дуже мала ймовірність помилок. Серед OLTP не годиться для складання звітів, так як нормалізована модель зазвичай має багато таблиць (по одній на кожен

об'єкт) зі складними взаємозв'язками. Навіть для простих звітів потрібно з'єднання багатьох таблиць, що призводить до складних і повільних запитів.

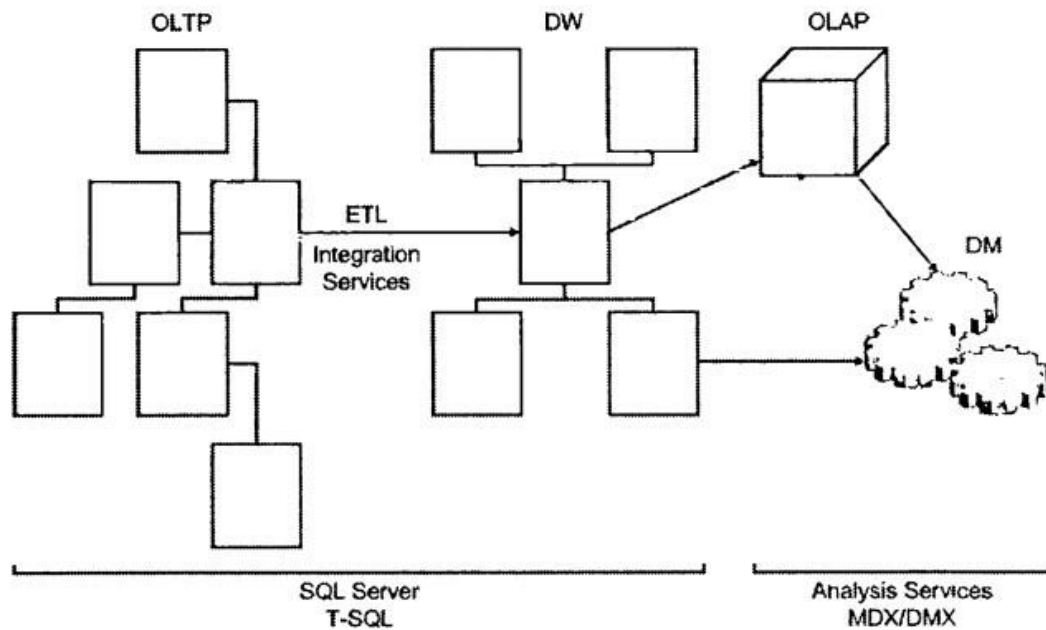


Рисунок 3.1 – Життєвий цикл даних

В SQL Server можна реалізувати базу даних OLTP, а потім управляти нею і формувати запити до неї на мові T-SQL. Сховище даних (data warehouse, DW) - середовище, розроблене для вилучення даних і формування звітів. Якщо обслуговується вся установа, це середовище називають сховищем даних; якщо ж воно призначене для обслуговування частини установи (таких, як конкретний відділ), його називають вітриною або кіоском даних (data mart). Модель даних сховища проектується і оптимізується головним чином для підтримки потреб вилучення даних. У моделі з'являється навмисна надмірність, яка припускає меншу кількість таблиць і більш прості зв'язки між ними, що, в кінцевому рахунку, призводить до формування більш простих і ефективних запитів в порівнянні з робочим середовищем OLTP.

Найпростіший варіант сховища даних називається схемою зірки, яка включає декілька таблиць вимірів (dimension table) і одну таблицю фактів (fact table). Кожна таблиця вимірювань являє об'єкт, за допомогою якого

аналізуються дані. Таблиця фактів містить фактичні дані і показники, скажімо, кількість для кожної значущої комбінації значень ключів вимірювання. Як правило, дані в сховищі заздалегідь підсумовуються або агрегуються на певному рівні деталізації на відміну від даних в робочому середовищі OLTP, які зазвичай записуються на рівні транзакцій [11-14].

У колишні роки перші версії SQL Server в основному були розраховані на робочі середовища OLTP, але з часом SQL Server почав також орієнтуватися на системи сховищ даних і потреби аналітичної обробки даних. Можливо реалізувати сховище даних за допомогою бази даних SQL Server, управляти нею і виконувати запити до неї мовою T-SQL.

Процес вилучення даних з вихідних систем (OLTP і ін.), маніпулювання і завантаження їх в сховище даних називається Extract Transform and Load (ETL, витяг, передача і завантаження). Для задоволення потреб ETL SQL Server надає засіб, названий Microsoft SQL Server Integration Services (SSIS). Системи OnLine Analytical Processing (OLAP, оперативна аналітична обробка), який підтримує динамічний оперативний аналіз згрупованих даних.

Розглянемо сховище даних, яке реалізоване як реляційна база даних в програмі SQL Server [9]. Коли користувач виконує запит на отримання агрегованих даних, додаток пред'являє базі даних запит, зазвичай скануючи і підсумовуючи великі обсяги базових даних. Але навіть при цьому більш ефективно обробляти такі запити на базі реляційних сховищ даних в порівнянні з робочим середовищем OLTP, підхід якого до вирішення цієї проблеми може виявитися недостатньо ефективним. Оперативний динамічний аналіз згрупованих даних зазвичай включає в себе часті запити на різних рівнях деталізації, для яких потрібна фігурна нарізка даних. Будь-який з таких запитів може обірватися через надмірні витрати, викликані необхідністю перегляду і підведення підсумків великих обсягів даних, і є ймовірність того, що час відгуку буде незадовільним.

Для задоволення таких потреб можна заздалегідь обчислити підсумки, згрупувавши дані на різних рівнях. Якщо заздалегідь підрахувати підсумки, запити на згруповані дані можуть бути виконані швидше.

Один варіант втілення цієї ідеї в життя - обчислення і зберігання різних рівнів угруповання в реляційному сховищі даних. Він вимагає опису складного процесу управління початковою обробкою підсумків і покроковим їх оновленням. Інший варіант - застосування спеціального програмного продукту, розробленого для систем OLAP, - Microsoft SQL Server Analysis Services (SSAS або AS). SSAS - це окрема служба - підсистема у складі служб SQL Server. Вона підтримує обчислення підсумків різних рівнів і зберігання їх в оптимізованих багатовимірних структурах, званих кубами. Вихідні дані для кубів SSAS можуть зберігатися (і зазвичай зберігаються) в реляційному сховищі даних. Крім підтримки великих обсягів підсумкових даних SSAS надає безліч потужних і складних засобів аналізу даних. Мова, що застосовується для управління кубами SSAS і виконання запитів до них, називається Multidimensional Expressions (MDX, багатовимірні вираження).

Екземпляр SQL Server - це установка механізму служби бази даних SQL Server. На одному комп'ютері можна встановити кілька екземплярів SQL Server. Кожен з них повністю незалежний від інших екземплярів з точки зору безпеки даних, якими він управляє, і у всіх інших відносинах. На логічному рівні у двох екземплярів, розміщених на одному і тому ж комп'ютері, загального стільки ж, скільки у двох екземплярів, що знаходяться на різних комп'ютерах. Звичайно ж, вони спільно використовують фізичні ресурси сервера, такі як центральний процесор, оперативна пам'ять і диски. Один з екземплярів, встановлених на комп'ютері, може бути заданий як екземпляр за замовчуванням, а інші повинні бути іменованими екземплярами. Задається екземпляр як екземпляр за замовчуванням або іменованій в процесі установки, але пізніше змінити цей вибір не можна. Для з'єднання клієнтської програми з екземпляром за замовчуванням необхідно задати ім'я комп'ютера або IP-адресу. Для з'єднання з іменованим екземпляром клієнт повинен вказати ім'я комп'ютера або IP-адресу,

за якою відбувається зворотний зв'язок, і далі ім'я екземпляра SQL Server (яке було задано при інсталяції) [13].

Для установки декількох екземплярів SQL Server на одному комп'ютері є кілька причин. Перша причина - скорочення витрат відділу технічної підтримки установи. Для того щоб тестувати роботу різних функцій у відповідь на звернення користувачів, відтворювати помилки, з якими стикаються користувачі в процесі експлуатації тощо. Відділу технічної підтримки потрібні локальні інсталяції SQL Server, що представляють робоче середовище користувачів з відповідною версією, редакцією і пакетом виправлень SQL Server. Якщо ж в установі застосовується кілька робочих середовищ користувачів, відділу технічної підтримки знадобиться кілька установок SQL Server. Замість використання декількох комп'ютерів, на кожному з яких встановлена окрема інсталяція SQL Server, що вимагає технічного супроводу, відділ технічної підтримки може мати один комп'ютер з декількома примірниками сервера.

Ще один приклад - розробники служб баз даних іноді повинні гарантувати своїм клієнтам повне безпечне розподілення даних різних клієнтів. Постачальник бази даних може мати дуже потужний інформаційний центр, розміщуючи множинні екземпляри SQL Server на одному комп'ютері замість того, щоб обслуговувати менш потужні комп'ютери, на кожному з яких встановлені різні екземпляри сервера.

Базу даних можна уявити у вигляді контейнера об'єктів, таких як таблиці, представлення, збережені процедури тощо. Кожен екземпляр SQL Server може містити кілька баз даних. Під час установки SQL Server створюється кілька системних баз даних, що зберігають системні дані і обслуговують внутрішні завдання. Після установки є можливість створювати власні призначені для користувача бази даних, які будуть зберігати дані програми. До системних баз даних, створюваних програмою установки сервера, відносяться наступні: master, Resource, model, tempdb і msdb [19].

- master. База даних master зберігає метадані, асоційовані з екземпляром сервера, його конфігурацію, відомості про всі бази даних екземпляра і установчу інформацію.

- Resource. База даних Resource була додана в версію SQL Server 2005 і містить всі системні об'єкти. При запиті інформації щодо метаданих в базі даних, вони здаються локальними по відношенню до бази даних, але на практиці вони зберігаються в базі даних Resource.

- model. База даних model використовується як шаблон для нових баз даних. Будь-яка нова база даних спочатку створюється як копія model. Тому, якщо необхідно, щоб певні об'єкти (наприклад, типи даних) з'являлися у всіх нових базах даних, або якісь властивості бази даних були налаштовані певним чином у всіх нових базах даних, то необхідно створити такі об'єкти і налаштувати відповідним чином властивості в базі даних model. Слід зважати на те, що зміни, які будуть внесені в базу даних model, ніяк не позначаться на вже існуючих базах даних, вони торкнуться тільки нових баз даних, які будуть створені надалі.

- tempdb. У базі даних tempdb SQL Server зберігає тимчасові дані, такі як робочі таблиці, область сортування, відомості для контролю версій рядків тощо. SQL Server дозволяє створювати тимчасові таблиці для власних потреб, які, як правило, зберігаються в tempdb. Слід враховувати, що ця база даних знищується і створюється як копія model при кожному запуску екземпляра SQL Server. З цієї причини, якщо потрібно створити об'єкти для тестування і не треба, щоб вони залишалися в базі даних, їх слід створювати в tempdb. Якщо навіть користувач не зможе вичистити їх, вони будуть автоматично видалені при наступному запуску сервера.

- msdb. У базі даних msdb зберігає свої дані служба, іменована SQL Server Agent. SQL Server Agent відповідає за автоматизацію, що включає такі об'єкти, як завдання, розкладу та оповіщення. SQL Server Agent - це служба, яка також відповідає за реплікацію. База даних msdb, крім того, зберігає дані, що стосуються інших засобів SQL Server, наприклад, Database Mail і Service Broker.

Можна створити стільки користувальницьких баз даних, скільки буде потрібно в одному екземплярі сервера. Призначена для користувача база даних буде зберігати об'єкти і дані для програми.

На рівні бази даних можна задати властивість collation (добірка), яка визначає мовну підтримку, сприйнятливність до регістру символів і порядок сортування символічних даних в цій базі даних. Якщо не ставити властивість collation для бази даних під час її створення, буде застосований варіант вибірки, прийнятий за замовчуванням (обраний під час встановлення) в даному екземплярі.

Для запуску програмного коду на мові T-SQL, що відноситься до бази даних, клієнтський додаток повинен підключитися до примірника SQL Server і встановити зв'язок з відповідною базою даних або використовувати її.

3.3 Проектування бази даних прогнозування завантаження банкоматів

Проектування бази даних здійснено за допомогою методу «сутність - зв'язок». При проведенні аналізу предметної області виділено такі сутності:

- Банк
- Банкомат
- Інкасація
- Видача готівки

Для всіх зазначених сутностей виділені атрибути - властивості, характерні для всіх екземплярів певної сутності. Для забезпечення унікальності записів в таблицях баз даних примірнику кожній сутності відповідає унікальний код. На рис. 3.2 зображено ER-діаграми для всіх сутностей з визначенням ступеню зв'язку.

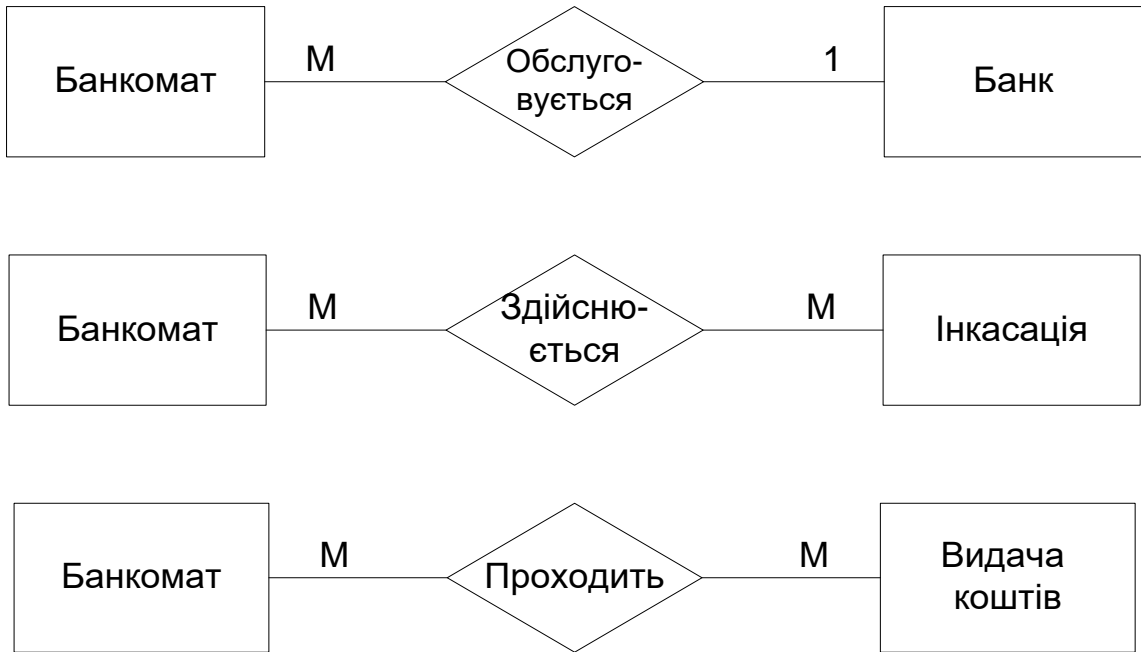


Рисунок 3.2 - ER-діаграми предметної області

В результаті проектування отримано схему даних, наведену на рис. 3.3.

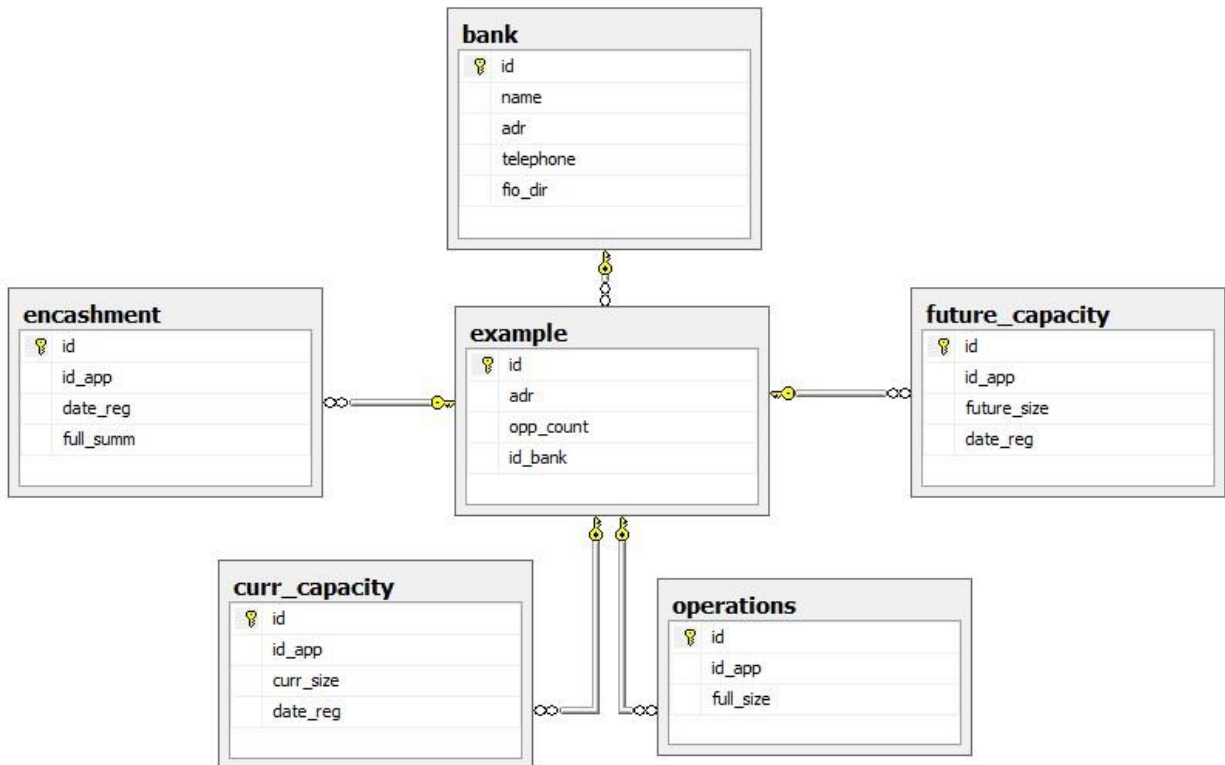


Рисунок 3.3 – Схема бази даних

3.4 Розробка інтерфейсу програми

Інтерфейс (GUI, графічний інтерфейс користувача) - це комплекс засобів для взаємодії користувача з технічною системою (в тому числі з програмним додатком, мультимедійним виданням).

У поняття призначеного для користувача інтерфейсу комп'ютерної системи входять наступні складові:

- графічне середовище - картинка на екрані;
- набір керуючих елементів призначеного для користувача інтерфейсу і їх розташування на екрані;
- технології взаємодії користувача з системою.

Керуючі елементи призначеного для користувача інтерфейсу - це графічні елементи (кнопки, списки, діалогові вікна тощо), які дозволяють здійснювати будь-які дії з комп'ютерною системою (наприклад, вибирати пункти і властивості об'єктів).

Основні вимоги до призначеного для користувача інтерфейсу:

- функціональність (відповідність завданням користувача);
- відповідність технології;
- зрозумілість і логічність;
- забезпечення високої швидкості роботи користувача;
- забезпечення захисту від людських помилок;
- швидке навчання користувача;
- суб'єктивне задоволення користувача.

При розробці інтерфейсу були розроблені макети головних форм, а вже по ним були розроблені основні форми за допомогою візуальних компонентів обраного програмного середовища.

Головне вікно програми наведено на рис. 3.4 та складається з головного меню та області з таблицею для відображення даних.

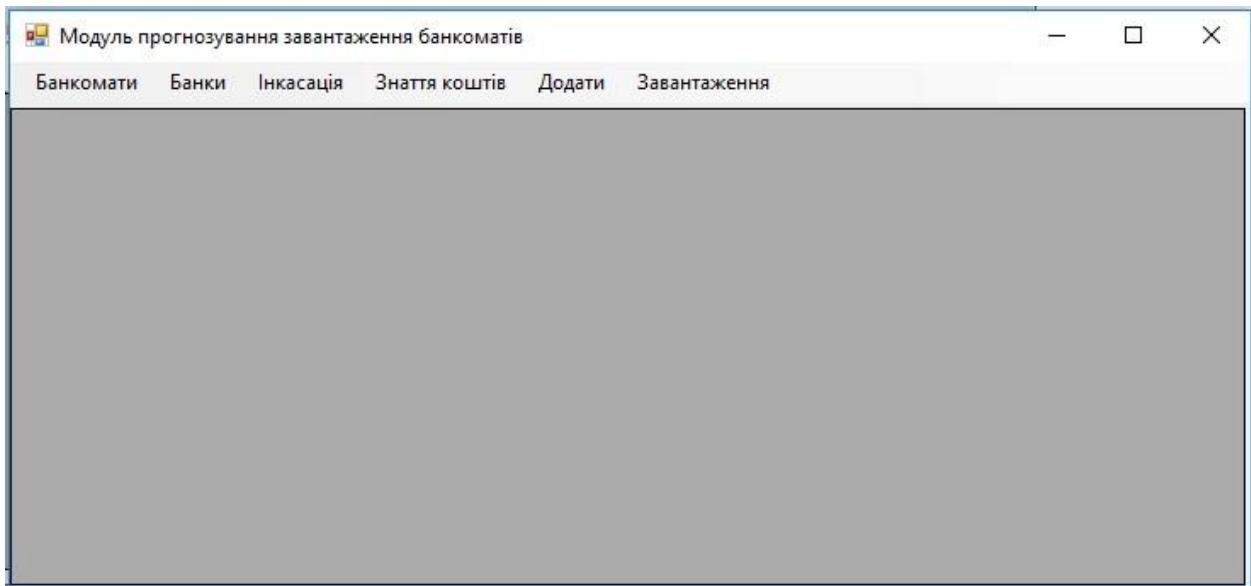


Рисунок 3.4– Головне вікно програми

Для здійснення операцій додавання даних та запиту на розрахунок прогнозуючого значення завантаження спроектовано вікна інтерфейсу, у яких передбачається наявність полів для введення даних та списків, календарів, числових полів. (рис. 3.5)

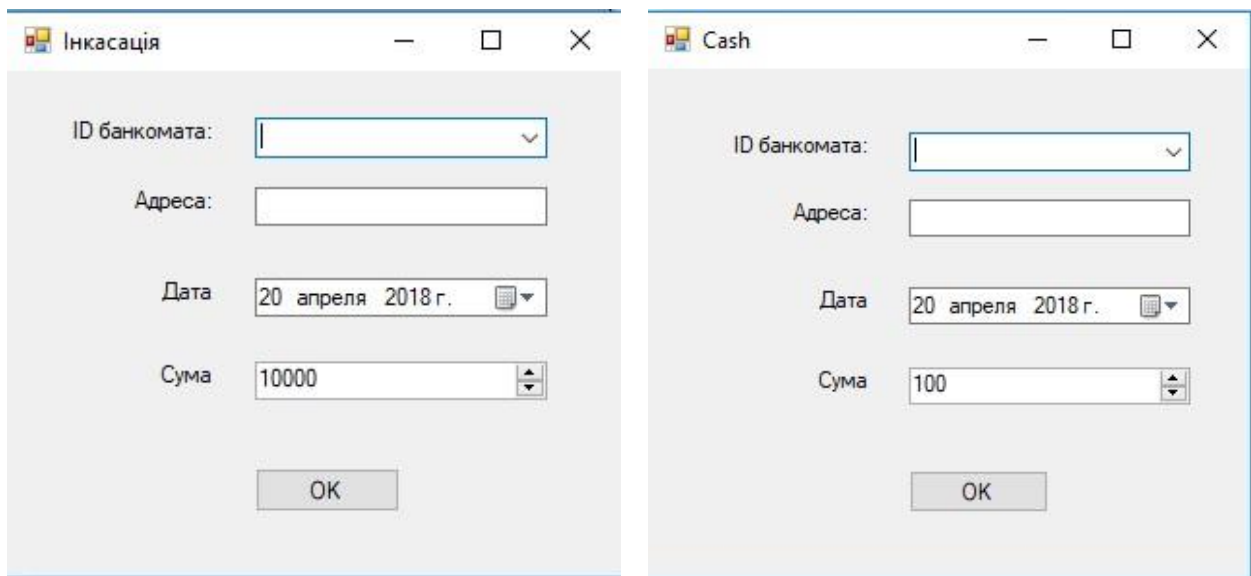


Рисунок 3.5 – Інтерфейси вікон для введення даних

The image shows a standard Windows-style dialog box with the title 'Завантаження'. It contains four input fields: a dropdown menu for 'ID банкомата', a text box for 'Адреса банкомата', a date picker for 'Дата' showing '20 апреля 2018 г.', and another text box for 'Прогнозоване завантаження'. At the bottom center is an 'OK' button.

Рисунок 3.5 (Продовження)

3.5 Проектування OLAP-куба

При проектуванні OLAP-куба в якості джерела даних (Data Source) використовувалася створена раніше реляційна база даних на SQL сервері, з'єднання з якою організовано за допомогою керування підключенням OLE DB. OLAP-куб побудований поверх створеного уявлення джерела даних (Data Source View). Схема подання джерела даних представлена на рис. 3.6:

Опис вимірювань спроектованого куба приведено в таблиці 3.1.

Таблиця 3.1 - Опис вимірів багатовимірного куба

Назва виміру	Ієрархія	Тип виміру
Rf0_addr_obj	Клієнт-район-вулиця- будинок-корпус	Звичайний (regular) вимір
Rf0_addr_reg	Немає	Звичайний (regular) вимір
Rf0_rent_line	Немає	Звичайний (regular) вимір
Tr0_rent_reg	Немає	Звичайний (regular) вимір
Time	Рік - місяць	Часовий вимір

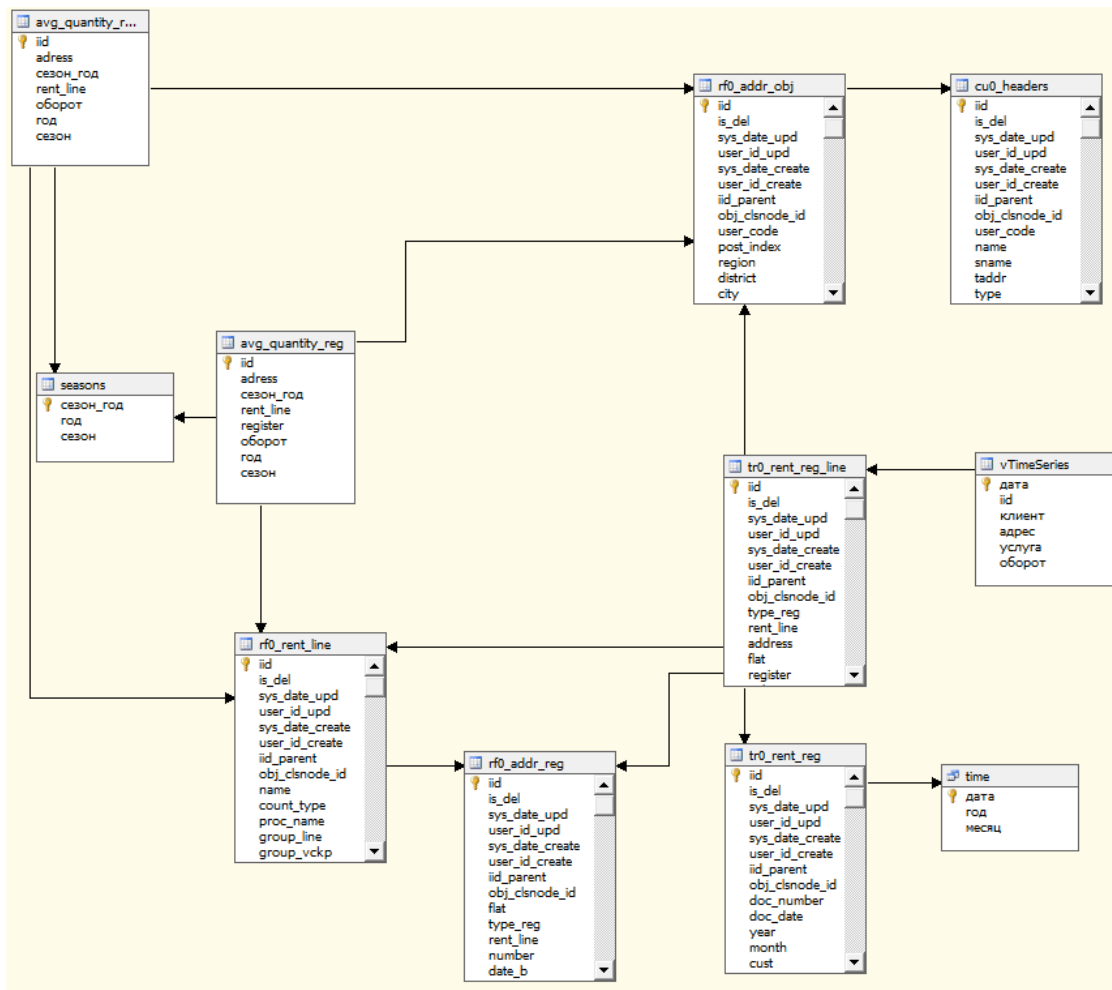


Рисунок 3.6 - Представлення джерела даних Umka.dsv

Схема побудованого OLAP-куба наведена на рис. 3.7.

Після розгортання вимірів і OLAP-куба на сервері було протестовано його функціонування. Для цього був виконаний тестовий зріз куба. Для зрізу були використані виміри «Times», «Rf0_rent_line» і ієрархія по банкоматах і прикріплених до них адрес. Крім того, для зрізу був використаний показник обороту таблиці фактів. Одним із способів отримання зрізу OLAP-куба є мова MDX, яка спеціально призначена для створення запитів до багатовимірних кубів.

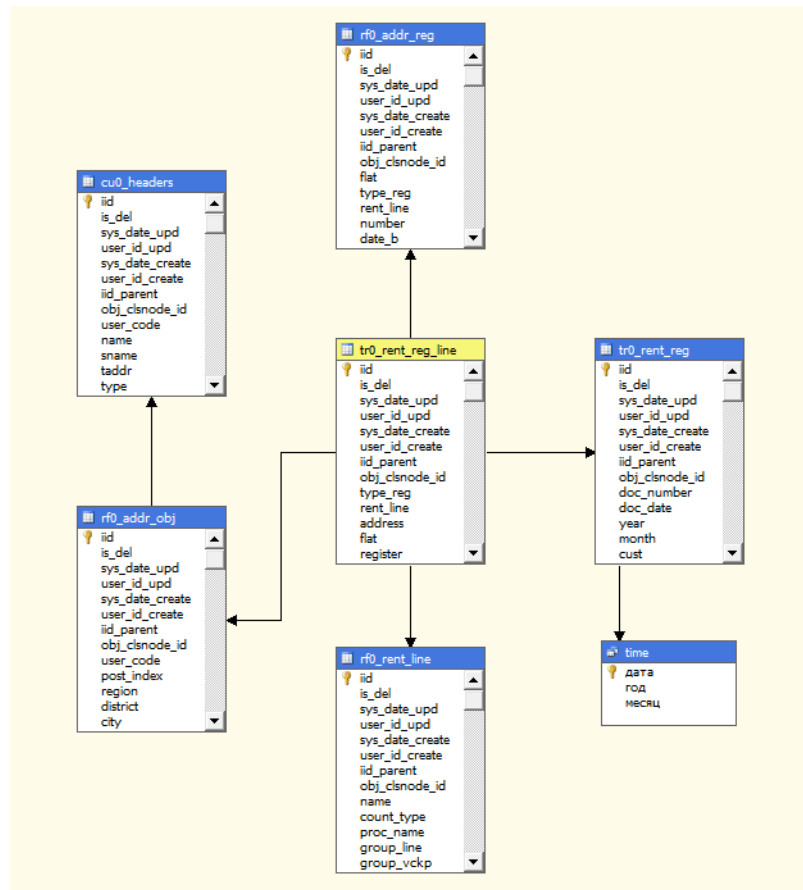


Рисунок 3.7 – Схема данных OLAP-куба

Приклад MDX запиту:

```
SELECT NON EMPTY { [Measures].[Quantity] } ON COLUMNS, NON
EMPTY { ([Tr0 Rent Reg Line].[Клиент].[Клиент].ALLMEMBERS * [Tr0 Rent Reg
Line].[Улица].[Улица].ALLMEMBERS * [Tr0 Rent Reg Line].[Дом].[Дом].ALLMEMBERS
* [Tr0 Rent Reg Line].[Корпус].[Корпус].ALLMEMBERS * [Tr0 Rent Reg
Line].[Услуга].[Услуга].ALLMEMBERS
[Time].[Год].[Год].ALLMEMBERS
[Time].[Месяц].[Месяц].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER_CAPTION,
MEMBER_UNIQUE_NAME ON ROWS FROM [Umka1 1] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR,
FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
```

Результат виконання запиту представлений на рис. 3.8.

Quantity	Названия столбцов											
Названия строк	1	2	3	4	5	6	7	8	9	10	11	12
Озерки	25058,58	22882,93	22174,36	18613,86	20333,87	18533,4	17951,66	17342,16	19467,35	56318,55	31932,42	37502,73
ул. Есенина	25058,58	22882,93	22174,36	18613,86	20333,87	18533,4	17951,66	17342,16	19467,35	56318,55	31932,42	37502,73
12	25058,58	22882,93	22174,36	18613,86	20333,87	18533,4	17951,66	17342,16	19467,35	56318,55	31932,42	37502,73
1	25058,58	22882,93	22174,36	18613,86	20333,87	18533,4	17951,66	17342,16	19467,35	56318,55	31932,42	37502,73
Водоотведение ОД ГВС	5302,94	5136,02	4654,06	5091,75	4631,45	3849,41	2845,67	1507,38	3976,86	4429,83	4234,21	4594,44
Водоотведение ОД ХВС	9432	8349	8340	6365	7690	7235	7475	7880	7650	8240	8140	8560
ГВС										13509	545	634
ГВС общедомовое	400,61	346,68	303,96	323,78	287,85	213,99	155,99	74,78	190,49	285,73	293,03	337,46
Отопление ОД	491,03	702,23	536,34	468,33	34,57	0	0	0	0	317,99	500,18	794,83
ХВС										21296	721	814
ХВС общедомовое	9432	8349	8340	6365	7690	7235	7475	7880	7650	8240	8140	8560
Электроэнергия дневная ОД	0	0	0	0	0	0	0	0	0	0	6218	8941
Электроэнергия ночная ОД	0	0	0	0	0	0	0	0	0	0	3141	4267
Общий итог	25058,58	22882,93	22174,36	18613,86	20333,87	18533,4	17951,66	17342,16	19467,35	56318,55	31932,42	37502,73

Рисунок 3.8 – Зріз куба

3.6 Реалізація функціональних частин програми

При реалізації клієнтської частини використано Entity Framework з технологією «DataBase First» (рис. 3.9)

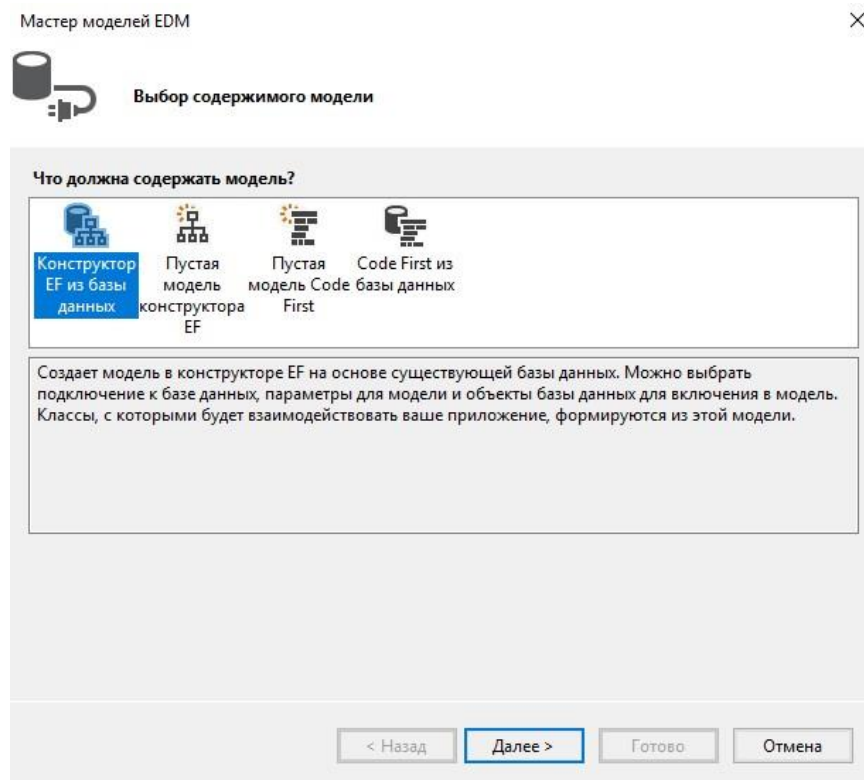


Рисунок 3.9 – Вибір технології «DataBase First»

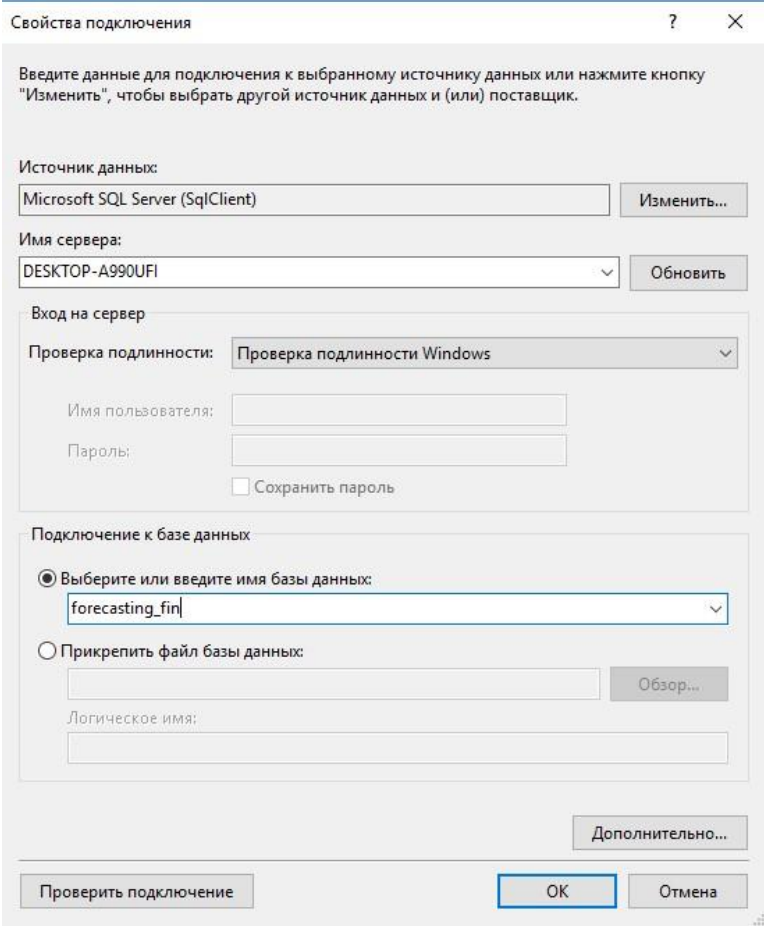


Рисунок 3.10 – Підключення до бази даних

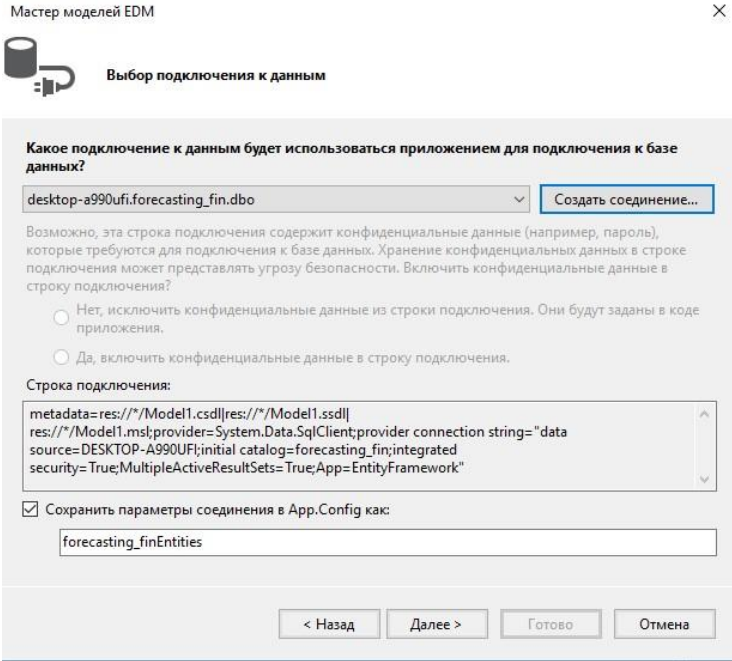


Рисунок 3.11 – Стрічка підключення

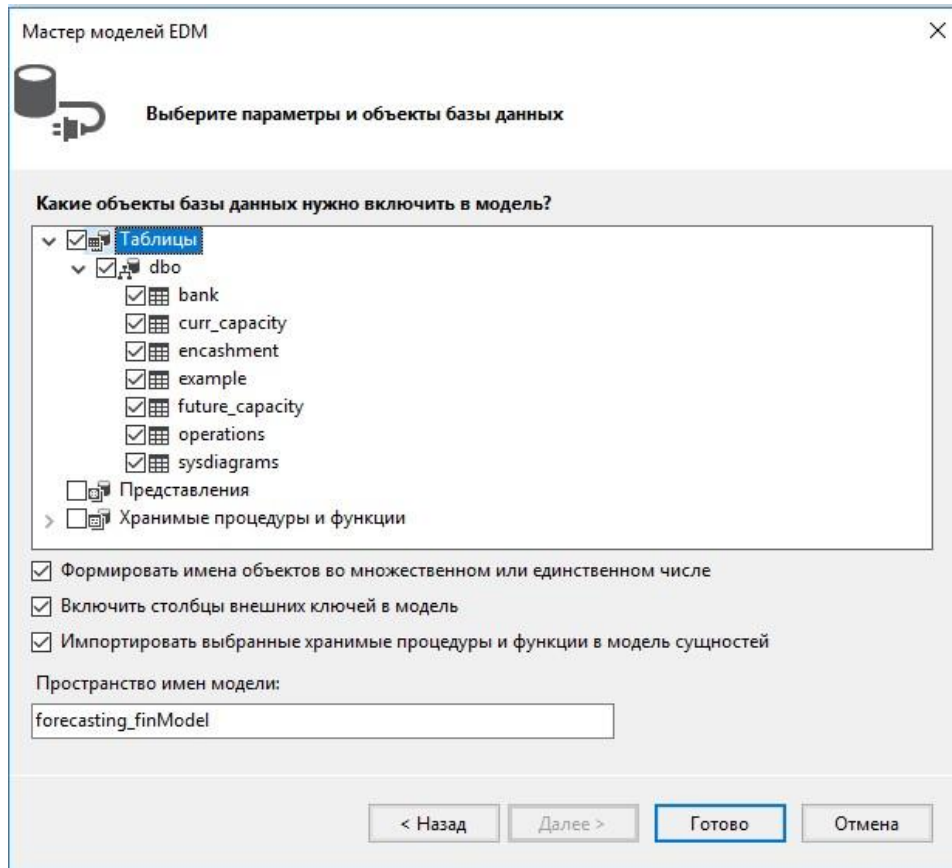


Рисунок 3.12 – Вибір таблиц з бази даних

В результаті отримано модель даних, наведену на рис. 3.13.

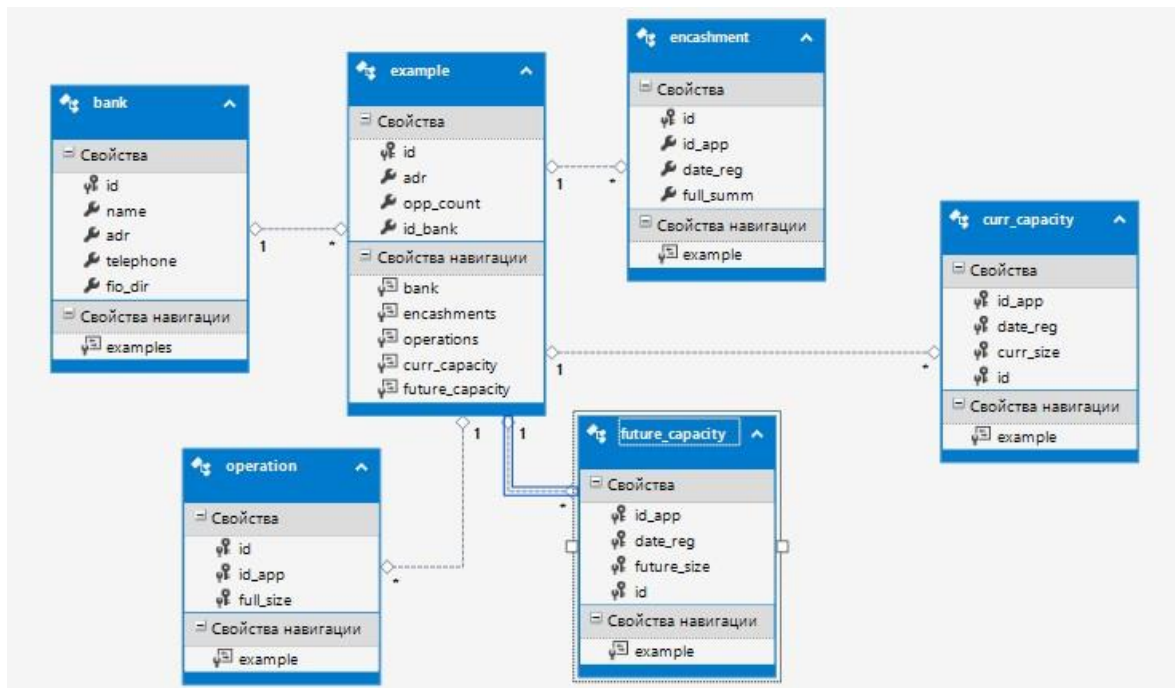


Рисунок 3.13 – Модель даних

3.7 Тестування роботи програми прогнозування завантаження банкоматів готівкою

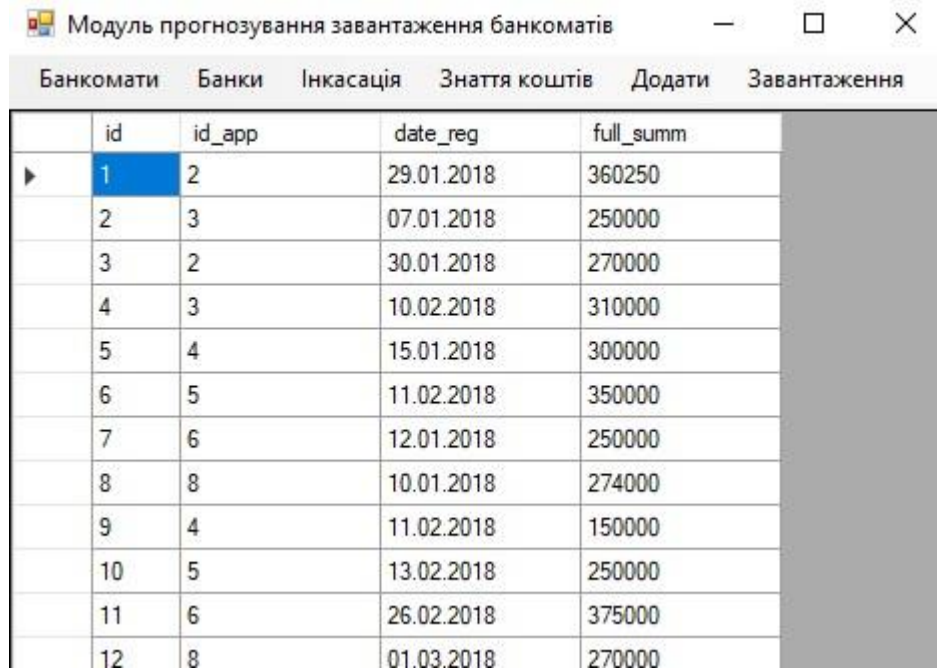
Перегляд таблиць за допомогою головного меню продемонстровано на рис. 3.14-3.18.

	id	adr	name	telephone	fio_dir
▶	1	Хмельницьке шоссе, 45	Приват	0672372717	Іванов І.В.
	2	Набережна 34	ПУМБ	0992382761	Петров В.В.
	3	Шевченка 67	Аваль	0675278276	Ткаченко С.О.
	4	Сумська 117	Укрсоцбанк	0633434632	Алісов В.К.
	5	Франка 89	Альфа-банк	0986544324	Калінін С.Д.
*					

Рисунок 3.14 – Перегляд таблиці «Банк»

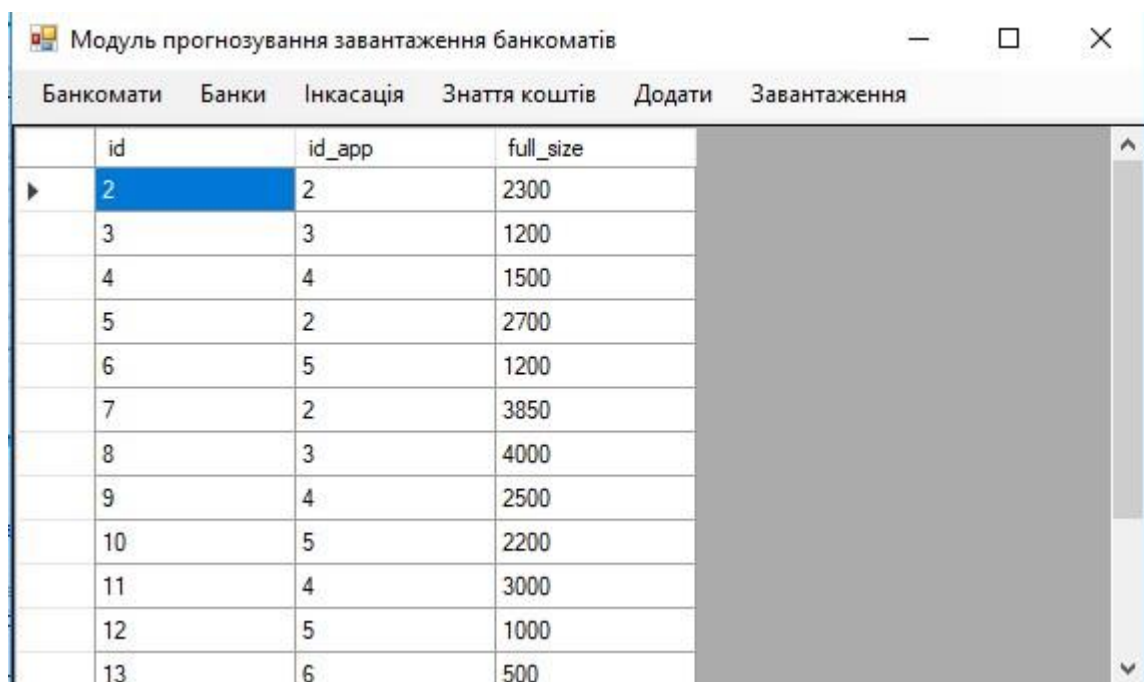
	id	adr	opp_count	id_bank
▶	2	Вишнева 112	500000	1
	3	Межова 87	500000	3
	4	Хмельницьке шоссе 95	700000	2
	5	Набережна 56	750000	2
	6	Коцюбинського 89	500000	4
	8	Хмельницьке шоссе	800000	5
*				

Рисунок 3.15 – Перегляд таблиці «Банкомат»



	id	id_app	date_reg	full_summ
▶	1	2	29.01.2018	360250
	2	3	07.01.2018	250000
	3	2	30.01.2018	270000
	4	3	10.02.2018	310000
	5	4	15.01.2018	300000
	6	5	11.02.2018	350000
	7	6	12.01.2018	250000
	8	8	10.01.2018	274000
	9	4	11.02.2018	150000
	10	5	13.02.2018	250000
	11	6	26.02.2018	375000
	12	8	01.03.2018	270000

Рисунок 3.16 – Перегляд таблиці «Інкасації»



	id	id_app	full_size
▶	2	2	2300
	3	3	1200
	4	4	1500
	5	2	2700
	6	5	1200
	7	2	3850
	8	3	4000
	9	4	2500
	10	5	2200
	11	4	3000
	12	5	1000
	13	6	500

Рисунок 3.17 – Перегляд таблиці «Зняття коштів»

Додавання даних про фінансові операції з банкоматом здійснюється за допомогою меню «Додати» (рис. 3.18)

The image shows two side-by-side dialog boxes. The left dialog box is titled 'Інкасація' and contains the following fields: 'ID банкомата' with a dropdown menu showing '2', 'Адреса' with a text box containing 'Вишнева 112', 'Дата' with a date picker showing '20 апреля 2018 г.', and 'Сума' with a spinner box showing '10000'. The right dialog box is titled 'Cash' and contains the same fields: 'ID банкомата' with a dropdown menu showing '2', 'Адреса' with a text box containing 'Вишнева 112', 'Дата' with a date picker showing '20 апреля 2018 г.', and 'Сума' with a spinner box showing '100'. Both dialog boxes have an 'OK' button at the bottom.

Рисунок 3.18 – Додавання даних

Прогнозування завантаження банкоматів здійснюється через введення даних в окремій формі після вибору відповідного пункту меню.

The image shows a dialog box titled 'Завантаження'. It contains the following fields: 'ID банкомата' with a dropdown menu showing '4', 'Адреса банкомата' with a text box containing 'Хмельницьке шоссе 95', 'Дата' with a date picker showing '20 апреля 2018 г.', and 'Прогнозоване завантаження' with a text box containing '171862'. There is an 'OK' button at the bottom.

Рисунок 3.19 – Виведення прогнозованого значення завантаження

3.7.1 Методи тестування.

Технологій тестування існує досить багато. Умовно їх можна віднести до статичних або динамічних.

Статичне тестування - це процес, який зазвичай асоціюють з аналізом ПЗ. Статичним тестуванням користуються для верифікації практично будь-якого артефакту розробки: програмного коду компонент, вимог, системних специфікацій, функціональних специфікацій, документів проектування та архітектури програмних систем і їх компонентів, і т.д. Використання статичних методів тестування - один з найбільш ефективних способів виявлення дефектів на ранніх стадіях розробки ПЗ. Дійсно, статичне тестування - це єдиний спосіб тестування без запуску програмного коду програми.

Динамічне тестування - процес тестування, здійснюваний над працюючою системою або підсистемою. Воно не може бути здійснено без запуску програмного коду програми. Якщо точніше, динамічне тестування складається з елементів:

- запуску системи або підсистеми;
- виклику необхідних функціональних елементів або модулів;
- порівняння через графічний інтерфейс користувача поведінки
- системи з очікуваним результатом поведінки.

Технології тестування використовуються при застосуванні тих чи інших методів тестування. При тестуванні програмного додатку використано динамічне тестування.

3.7.2 Інструкція тестування програми прогнозування завантаження банкоматів готівкою.

Найменування об'єкта випробувань «Програма прогнозування завантаження банкоматів готівкою».

Метою попередніх комплексних випробувань є:

- перевірка працездатності програмного продукту;
- перевірка відповідності програмного продукту вимогам завдання;
- перевірка готовності програмного продукту до проведення дослідної комплексної експлуатації (таблиця 3.1).

Таблиця 3.1 – Результати тестування програмного продукту

Опис тесту	Дія	Очікуваний результат	Статус
Перевірка запуску програми	Вибір виконуваного файлу з директорії програми	З'являється вікно головної форми продукту	Виконано успішно
Перевірка переглядання таблиць	Обрати з головного меню таблицю за назвою	Відкриття таблиці на головній формі	Виконано успішно
Перевірка додавання даних	1. Вибір пункту меню «Додати» 2. Вибір підпункту для додавання даних 3. Введення даних в текстові поля 4. Застосування кнопки додавання 5. Перевірка даних шляхом перегляду таблиці	Додання даних до бази	Виконано успішно
Перевірка прогнозування завантаження банкомату	1. Вибір відповідного пункту меню 2. Введення даних в поле введення 3. Застосування кнопки розрахунку	Отримання значення прогнозованого значення завантаження банкомату	Виконано успішно

3.8 Аналіз результатів роботи програми прогнозування завантаження банкоматів готівкою

Для доведення досягнення поставленої мети було проведено порівняння точності прогнозування розробленої програми із точністю прогнозування програми-аналога. Для цього в ході обробки результатів були обчислені оцінки

показників якості прогнозування. Для аналізу результатів було обрано такі показники:

- середня абсолютна помилка (MAE):

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - y_t^*|, \quad (3.1)$$

де y_t - реальне значення потрібної банкомату готівки в момент часу t ; y_t^* - прогнозоване значення потрібної банкомату готівки в момент часу t , отримане в результаті роботи програми; n - кількість ретроспективних спостережень.

Даний показник дозволяє кількісно визначити величину розбіжності між реальними даними і значеннями прогнозу.

- коефіцієнт невідповідності КТ, рівний нулю в разі досконалого прогнозу (даний коефіцієнт не має верхньої кінцевої межі):

$$K_T = \sqrt{\frac{\sum_{t=1}^n (y_t - y_t^*)^2}{\sum_{t=1}^n y_t^2 + \sum_{t=1}^n y_t^{*2}}} \quad (3.2)$$

- час роботи програми.

У таблиці 3.2 представлені оцінки розрахованих показників якості програм прогнозування завантаження банкоматів готівкою. Кількість ретроспективних спостережень була взята $n=100$.

Таблиця 3.2 Оцінки показників якості програм прогнозування завантаження банкоматів готівкою

Програма \ Показник якості	MAE	КТ	T (с)
Розроблена програма	11,27	0,033	13
Аналог (ATM Cash Management Solution)	15,14	0,041	24

Із табл. 3.2 видно, що розроблена програма має меншу помилку MAE ($11,27 < 15,14$; відношення $15,14/11,27=1,34$, тобто на 34% краще) та менше значення коефіцієнта КТ ($0,033 < 0,041$; відношення $0,041/0,033=1,24$, тобто на 24% краще). Це дозволяє стверджувати, що розроблена програма щонайменше на 24% точніша за аналог, а значить мета роботи досягнута – точність прогнозування завантаження банкоматів готівкою підвищена.

3.9 Висновки

В результаті виконання розділу можна зробити наступні висновки:

- При розробці клієнтського додатку доцільно використовувати Entity Eramework засобами MS Visual Studio 2017 з технологією “Model First”.
- Робота клієнтського додатку з базою даних організована за допомогою технології OLEDB на основі структурованих запитів.
- Інтерфейс програми дозволяє виконувати необхідні задачі при роботі з базою даних.
- За результатами тестування, наведеними в таблиці 3.1, можна зробити висновок, що тестування пройшло успішно, і робота програмного продукту надійна.
- Розроблена програма має меншу помилку MAE ($11,27 < 15,14$; тобто на 34% краще) та менше значення коефіцієнта КТ ($0,033 < 0,041$; тобто на 24% краще). Це дозволяє стверджувати, що розроблена програма щонайменше на 24% точніша за аналог, а значить мета роботи досягнута – точність прогнозування завантаження банкоматів готівкою підвищена.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки [20]. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Колесницький О.К. та Перевозніков С.І..

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Колесницький О.К.	2. Перевозніков С.І.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	4
5	3	4
6	3	4
7	3	3
8	3	4
9	4	4
10	4	3
11	3	4
12	3	4
Сума балів	СБ ₁ = 41	СБ ₂ = 45
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 43$	

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати [20].

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M - місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 21$ день;

t - число днів роботи розробника, $t = 40$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	7000	333,33	5	1666,65
Інженер-програміст	4500	214,28	40	8571,2
Всього:				10237,85

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = 0,1 \cdot 10237,85 = 1023,78 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (4.2)$$

$$H_{\text{зн}} = (10237,85 + 1023,78) \cdot \frac{36,3}{100} = 4087,97 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	9000	25	3	562,5
Всього:				562,5

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot Ц_i \cdot K_i, \quad (4.4)$$

де n – кількість комплектуючих;

H_i – кількість комплектуючих і-го виду;

$Ц_i$ – покупна ціна комплектуючих і-го виду, грн;

K_i – коефіцієнт транспортних витрат (приймемо $K_i = 1,1$).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	150	1	150
Пачка паперу	уп.	120	1	120
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				308

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} ; \quad (4.5)$$

де V – вартість 1кВт-години електроенергії ($V=1,7$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=180$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,8$).

$$V_e = 1,7 \cdot 0,6 \cdot 180 \cdot 0,8 = 146,88 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_v можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 \cdot (10237,85 + 1023,78) = 11261,63 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = Z_o + Z_d + H_{зп} + A + K + B_e + I_B$$

$$B = 10237,85 + 1023,78 + 4087,97 + 562,5 + 146,88 + 308 + 11261,63 = 27682,61 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha} \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{27682,61}{1} = 27682,61$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{27682,61}{0,9} = 30698,45 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}}\Delta N)_i \quad (4.9)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 25 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 25 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 300 користувачів, протягом другого року – на 200 користувачів, протягом третього року – 100 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 1500 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 25 \cdot 1500 + (200 + 25) \cdot 300 = 105000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 25 \cdot 1500 + (200 + 25) \cdot (300 + 200) = 150000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 25 \cdot 1500 + (200 + 25) \cdot (300 + 200 + 100) = 172500 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

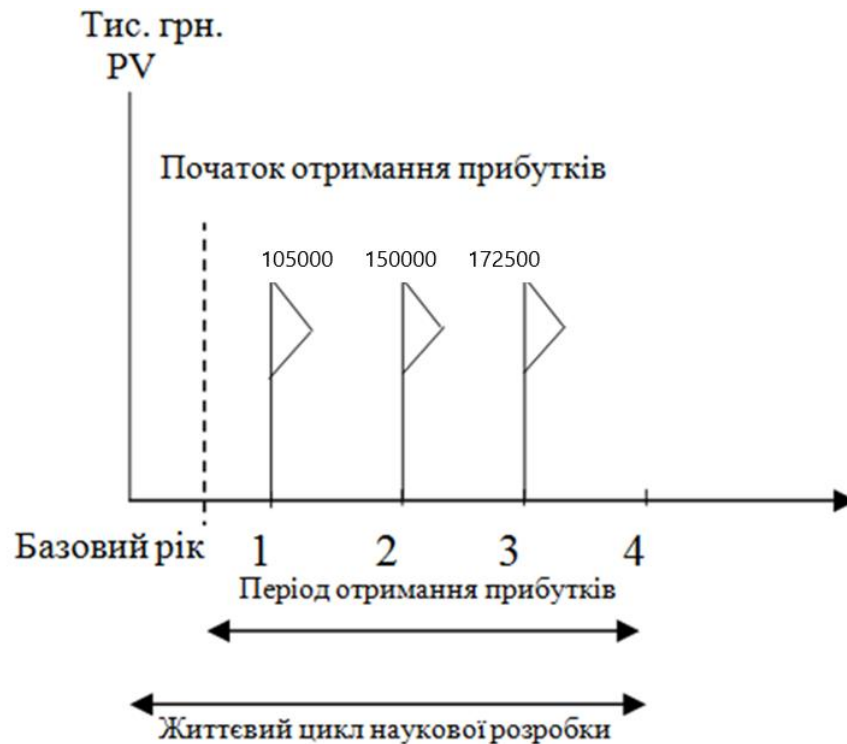


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{30698,45}{(1+0,1)^0} + \frac{105000}{(1+0,1)^2} + \frac{150000}{(1+0,1)^3} + \frac{172500}{(1+0,1)^4} = 347992,34 \text{ (грн.)}$$

Тоді розрахуємо E_{abc} :

$$E_{abc} = 347992,34 - 30698,45 = 317293,89 \text{ грн.}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1 \quad (4.12)$$

де E_{abc} – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T_j – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{317293,89}{30698,45}} - 1 = 1,24 \text{ або } 124 \%$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 124\% > \tau_{\text{мін}} = 0,3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}$$

$$T_{ок} = \frac{1}{1,24} = 0,80 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

В даному розділі було проведено економічне обґрунтування доцільності розробки програми для прогнозування завантаження банкоматів готівкою.

Незалежними експертами було здійснено оцінювання комерційного потенціалу розробки, за результатами якого було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки середньоарифметична сума балів становить 43.

Також було виконано прогнозування витрат на виконання розробки, де розраховано основну заробітну плату кожного із розробників, додаткову заробітну плату всіх розробників, нарахування на заробітну плату, амортизацію обладнання, комп'ютерів та приміщень, витрати на допоміжні матеріали, витрати на силову електроенергію тощо. Загальна сума витрат на виконання означених робіт склала 27682,61 грн.

Здійснене прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової роботи дозволило визначити дану суму – 30698,45 грн.

Виконано розрахунок ефективності вкладених інвестицій та періоду їх окупності. Визначено, що абсолютна ефективність вкладених інвестицій

становить 317293,89 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів НДДКР є доцільним.

Було розраховано відносну (щорічну) ефективність вкладених в наукову розробку інвестицій – 124 %, її величина більша за мінімальну (бар'єрну) ставку дисконтування, отже інвестор буде зацікавлений у фінансуванні даної наукової розробки.

Проведено розрахунок терміну окупності складає 0.8 року (менше 10 місяців). Це означає, що вже починаючи з 11 місяця розробка буде приносити прибуток.

В загальному, можна зробити висновок, що фінансування розробки програми для прогнозування завантаження банкоматів готівкою є економічно доцільним проектом.

ВИСНОВКИ

В результаті виконання магістерської кваліфікаційної роботи було розв'язано задачу розробки інтелектуальної інформаційної технології та програмного забезпечення прогнозування завантаження банкоматів готівкою на основі технологій Data Mining. Програмне забезпечення включає в себе базу даних, що містить інформацію про дану предметну область, і клієнтську програму, яка дозволяє виробляти різні дії з аналізу та обробки накопиченої інформації шляхом звернення до бази даних

У першому розділі магістерської роботи була поставлена проблема недостатньої точності сучасного процесу прогнозування завантаження банкоматів готівкою. Були розглянуті основні методи та програмні засоби, які виконують прогнозування завантаження банкоматів готівкою та визначені їх недоліки, сформульовано постановку задачі, яке враховує всі вимоги до певних областей функціонування програмного забезпечення прогнозування завантаження банкоматів.

У другому розділі магістерської кваліфікаційної роботи було проаналізовано процес постачання банкоматної мережі готівкою, сформульовано вимоги до інформаційної технології прогнозування завантаження банкоматів готівкою, розроблено технологію аналітичної обробки даних, яка заснована на комплексному використанні методу дерев прийняття рішень та методу часових рядів, розроблено структуру інформаційної технології прогнозування завантаження банкоматів готівкою та розроблено функціональну модель процесу прогнозування завантаження банкоматів готівкою.

У третьому розділі в ході практичної реалізації інформаційної технології прогнозування завантаження банкоматів готівкою було використано Entity Framework засобами MS Visual Studio 2017 з технологією "Model First". Робота клієнтського додатку з базою даних організована за допомогою технології OLEDB на основі структурованих запитів. Інтерфейс програми дозволяє

виконувати необхідні задачі при роботі з базою даних. За результатами тестування, зроблено висновок, що робота програмного продукту надійна та має меншу помилку МАЕ ($11,27 < 15,14$; тобто на 34% краще) та менше значення коефіцієнта КТ ($0,033 < 0,041$; тобто на 24% краще), що дозволяє стверджувати, що розроблена програма щонайменше на 24% точніша за аналог, а значить мета роботи досягнута – точність прогнозування завантаження банкоматів готівкою підвищена.

У четвертому розділі було проведено економічне обґрунтування доцільності розробки програми для прогнозування завантаження банкоматів готівкою. Було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки середньоарифметична сума балів становить 43. Загальна сума витрат на виконання означених робіт склала 27682,61 грн. Прогнозоване значення загальних витрат на виконання та впровадження результатів виконаної наукової роботи – 30698,45 грн. Визначено, що абсолютна ефективність вкладених інвестицій становить 317293,89 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів НДДКР є доцільним. Відносна (щорічна) ефективність вкладених в наукову розробку інвестицій – 124 %, терміну окупності складає 0.8 року (менше 10 місяців). Тобто фінансування розробки програми для прогнозування завантаження банкоматів готівкою є економічно доцільним проектом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кусяка Д. С., Прогнозування завантаження банкоматів готівкою // Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи-2019», Вінниця, 2019. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2019/paper/view/8183>. Дата звернення: Грудень 2019.
2. Кусяка Д. С., Колесницький О. К., Денисов І. К. Інформаційна технологія прогнозування завантаження банкоматів готівкою // Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи-2020», Вінниця, 2019. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2020/paper/view/8497>. Дата звернення: Грудень 2019.
3. Как устроен банкомат [Электронный ресурс] // Onliner : [сайт]. – Режим дступа: <https://tech.onliner.by/2011/07/29/atm>
4. Інкасація банкоматів [Електронний ресурс] // Підручники : [сайт] – Режим доступу: pidruchniki.com/12471013/bankivska_sprava/inkasatsiya_bankomativ
5. Кузин А.В. Базы данных, 5-е изд. / А.В. Кузин, С.В. Левонисова. – СПб.: Академия, 2012 – 317 с..
6. Райордан Р. Основы реляционных баз данных/Пер. с англ. – М.: Издательско-торговый дом “Русская редакция”, 2001 – 384 с.
7. Обломов, С.А. Технологии разработки программного обеспечения / С. А. Обломов. - Санкт-Петербург: Питер, 2002. – 464 с.
8. Atm Cash Management Solutions [Електронний ресурс] // Smart Technologies : [сайт] – Режим доступа: www.atmmarketplace.com/topics/atm-management/
9. СМ360: Оптимизация работы банкоматов — URL: <http://ids30.ru/решения/cm360> (дата обращения 15.05.2016)

- 10.Ицик Бен-Ган Microsoft SQL Server 2008. Основы T-SQL: Пер.с англ.-СПб.: БХВ-Петербург, 2009. – 432 с.
- 11.Классика баз данных [Электронный ресурс] // Citforum : [сайт]. – Режим доступа: <http://citforum.ru/database/classics/>
- 12.Харрингтон Д. Проектирование объектно-ориентированных баз данных / Д. Харрингтон – М.: ДМК-Пресс, 2012 – 272 с.
- 13.Джексон Дж. Проектирование реляционных баз данных для использование с микро-ЭВМ.-М.: Мир, 1991
- 14.Кузин А.В. Разработка баз данных в системе Microsoft Access / А.В. Кузин, В.М. Демин. – М.: Инфра-М, 2014 – 224 с.
- 15.Дейт К. Введение в системы баз данных. - М. : Издательский дом "Вильямс", 2017. - 1328 с.
- 16.Илюшечкин В.М. Основы использования и проектирования баз данных / В.М. Илюшечкин. – М.: Юрайт, 2016 – 214 с.
- 17.Исаев Г.Н. Проектирование информационных систем / Г.Н. Исаев. – М.: Омега-Л, 2012 – 432 с.
- 18.Подбельский В.В. Язык С#: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
- 19.Стенли Б. Липпман. С# для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
- 20.Сэвитч У. С в примерах: Москва: ЭКОМ, 1997. 736с.
- 21.Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.