

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів»

Виконала: студентка 2-го курсу, групи ІПІ-21м
спеціальності 121 – Інженерія програмного
забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Сікорська М.О.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Романюк О.В.
(прізвище та ініціали)

«14» 12 2022 р.


Опонент: к.т.н., доц. каф. ЗІ Войтович О.П.
(прізвище та ініціали)

«14» 12 2022 р.

Допущено до захисту
Завідувач кафедри ПЗ
к.т.н. проф. Романюк О.Н.
(прізвище та ініціали)

«14» 12 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти II-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення



ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
« 16 » вересня 2022 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Сікорській Марії Олександрівні

1. Тема роботи – Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів.

Керівник роботи: Романюк Оксана Володимирівна, к.т.н., доц. кафедри ПЗ, затверджені наказом вищого навчального закладу від 15 вересня 2022 року №205-А.

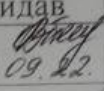
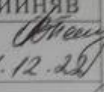
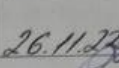
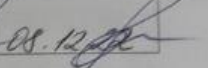
2. Строк подання студентом роботи
9 грудня 2022 р.

3. Вихідні дані до роботи: методи прогнозування продажу – арифметичне згладжування, експоненційне згладжування, правило півтора, метод Хольта-Вінтерса, метод прогнозування сезонного попиту з урахуванням коефіцієнту сезонності; особисті дані користувача, дані продажу минулого періоду; середовища розробки – Visual Studio 2022 та Microsoft SQL Server; мова розробки – C#; операційна система – Windows 10; вихідні дані – сформований прогноз продажу товарів на основі запропонованого методу прогнозування.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз стану питання та постановка задач дослідження; розробка структури системи та методів підвищення ефективності прогнозування продажів товарів; розробка програмних засобів; тестування програми; економічна частина; висновки; список використаних джерел; додатки.

5. Перелік графічного матеріалу: титульний слайд; мета, об'єкт та предмет дослідження; задачі дослідження; наукова новизна; аналіз аналогів; адаптивний метод прогнозування продажів товарів; метод автоматизованого формування звіту з прогнозування продажів, блок-схема алгоритму формування звіту; блок-схема алгоритму авторизації; загальна блок-схема алгоритму роботи програми; тестування функціональності; економічна частина; публікація та апробація матеріалів магістерської кваліфікаційної роботи; висновки; завершальний слайд.

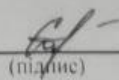
6. Консультанти розділів роботи

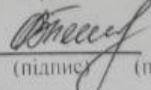
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Романюк О.В., к.т.н., доцент кафедри ПЗ	 16.09.22.	 14.12.22
5	Глущенко Л.Д., к.е.н., доцент кафедри ЕПВМ	 26.11.22.	 08.12.22

7. Дата видачі завдання 16 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз методів та методик прогнозування	17.09.2022 – 30.09.2022	Вик.
2	Розробка структури системи, адаптивного методу підвищення ефективності прогнозування продажів товарів, методу формування звіту з прогнозування	01.10.2022 – 19.10.2022	Вик.
3	Розробка програмного забезпечення для прогнозування продажів товарів	19.10.2022 – 10.11.2022	Вик.
4	Тестування програмного додатку	11.11.2022 – 25.11.2022	Вик.
5	Економічна частина	26.11.2022 – 8.12.2022	Вик.

Студентка  Сікорська М.О.
(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи  Романюк О.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.91

Сікорська М.О. Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів. Магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2022. – 123 с.

На укр. мові. Бібліогр.: 23 назв; рис.: 45; табл. 11.

У магістерській кваліфікаційній роботі розроблено методи, алгоритми та програмні засоби для підвищення ефективності прогнозування продажів товарів. Уперше запропоновано адаптивний метод прогнозування продажу товарів, особливість якого полягає в автоматизованому аналізі вхідних даних про продажі за минулі періоди та підборі найбільш оптимального методу прогнозування, що дозволило підвищити ефективність прогнозу.

Уперше запропоновано метод автоматизованого формування звіту прогнозування, який дозволяє фіксувати макроекономічні показники, що дозволило підвищити ефективність оцінювання результатів прогнозування майбутніх обсягів збуту товарів.

Розроблено програмний засіб для прогнозування продажів з використанням технології Windows Forms інтегрованого середовища розробки Microsoft Visual Studio та мови програмування C#. Системою управління базами даних обрано Microsoft SQL Server, що має вбудовану підтримку .NET Framework.

Ключові слова: прогнозування, продажі, метод прогнозування.

ABSTRACT

Sikorska M. Development of methods and software tools to increase the effectiveness of product sales forecasting. Vinnitsa: VNTU, 2022. – 123 p.

In Ukrainian language. Bibliographer: 23 titles; fig.: 45; tabl. 11.

In the master's thesis, methods, algorithms and software tools were developed to increase the effectiveness of product sales forecasting. For the first time, an adaptive method of forecasting the sale of goods was proposed, the feature of which is the automated analysis of input data on sales for past periods and the selection of the most optimal forecasting method, which made it possible to increase the effectiveness of the forecast.

For the first time, a method of automated formation of a forecasting report was proposed, which allows recording macroeconomic indicators, which made it possible to increase the effectiveness of evaluating the results of forecasting future sales volumes of goods.

A software tool for forecasting sales was developed using the Windows Forms technology of the Microsoft Visual Studio integrated development environment and the C# programming language. The database management system was chosen to be Microsoft SQL Server, which has built-in support for the .NET Framework.

Keywords: forecasting, sales, forecasting method.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	12
1.1 Аналіз стану питання	12
1.2 Порівняльний аналіз аналогів	13
1.3 Аналіз методів прогнозування продажу	17
1.4 Аналіз методик складання прогнозу продажів	19
1.5 Постановка задач розробки	21
1.6 Висновки	21
2 РОЗРОБКА СТРУКТУРИ СИСТЕМИ ТА МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОГНОЗУВАННЯ ПРОДАЖІВ ТОВАРІВ	22
2.1 Аналіз даних.....	22
2.2 Розробка адаптивного методу прогнозування продажів товарів	23
2.2 Розробка методу формування звіту з прогнозування продажів	26
2.4 Розробка структури інтерфейсу та вибір кольорової гами для програмного засобу прогнозування продажів.....	28
2.5 Розробка алгоритмів роботи програми	32
2.6 Висновки	38
3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ	39
3.1 Варіантний аналіз засобів створення програмного продукту	39
3.2 Обґрунтування вибору середовища розробки	40
3.3 Розробка бази даних.....	42
3.4 Програмна реалізація додатку для прогнозування продажів	45
3.5 Висновки	54
4 ТЕСТУВАННЯ ПРОГРАМИ	55
4.1 Аналіз методів тестування програмного забезпечення.....	55
4.2 Тестування розробленого програмного продукту	56
4.3 Вимоги до персонального комп'ютера та інструкція користувача.....	65
4.4 Висновки	66

5 ЕКОНОМІЧНА ЧАСТИНА.....	67
5.1. Проведення комерційного аудиту науково-технічної розробки	67
5.2 Розрахунок витрат на здійснення науково-технічної розробки	72
5.3 Розрахунок економічної ефективності та обґрунтування економічної доцільності комерціалізації науково-технічної розробки.....	76
5.4 Висновки	81
ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
ДОДАТКИ.....	85
ДОДАТОК А. Технічне завдання	Ошибка! Закладка не определена.
ДОДАТОК Б. Протокол перевірки роботи ...	Ошибка! Закладка не определена.
ДОДАТОК В. Лістинг програми	89
ДОДАТОК Г. Ілюстративна частина.....	114

ВСТУП

Обґрунтування вибору теми дослідження. Прогнозування є важливою частиною процесу управління. Без нього неможливо сформулювати тактичний або стратегічний план корпоративного розвитку. Для комерційних та виробничих компаній важливо передбачити продажі товарів на наступний період. Від цього буде залежати як кількість створених запасів, так і накопичені для цього кошти.

Під методологією прогнозування розуміють галузь знань про методи, способи й системи прогнозування, а саме:

- метод прогнозування — спосіб дослідження об'єкта, спрямований на розробку прогнозу;
- методика прогнозування — сукупність одного або декількох методів прогнозування;
- система прогнозування — впорядкована сукупність методик прогнозування та засобів їх реалізації.

Відомо, що теорія прогнозування включає: аналіз об'єкта прогнозування; методи прогнозування (математичні — формалізовані, експертні — інтуїтивні); системи прогнозування.

Прогнозування продажів – це науково-обґрунтоване передбачення обсягів збуту продукції, з певною ймовірністю, результат отримується за рахунок відповідної інформації за попередні роки з урахуванням змін факторів в майбутньому, що можуть так чи інакше вплинути на його результат. Головною метою даної процедури є визначення ефективної кількості продукції, яку фірма може продати в перспективі, щоб вдало розподілити свої ресурси для отримання найбільшого прибутку і досягнення поставлених цілей [1].

Програмне забезпечення для прогнозування продажів – інструмент для складання прогнозу попиту на продукцію за допомогою набору математичних моделей прогнозування і швидкого внесення необхідних коригувань з урахуванням наявних оперативних даних.

На сьогоднішній день більшість великих та малих компаній в Україні та закордоном використовують програмне забезпечення для управлінських процесів та зокрема для управління бюджетом та планування майбутніх продажів. Такі компанії мають окремі відділи, які займаються даними процесами, та в чій обов'язки входить опанування та використання таких програм. Загалом для того щоб ефективно використовувати програмне забезпечення такого плану потрібно проводити окреме навчання, на це витрачається як багато час, так і грошей.

Щодо до малого бізнесу, який все більше і більше поширюється серед людей зі зростання популярності соціальних мереж та за рахунок карантину. Ті хто перетворює своє хобі на бізнес, за часту не мають чіткого уявлення про процеси менеджменту та економічної складової бізнесу. У таких випадках актуальність використання простого програмного засобу, для управління бюджетом та прогнозування продажів, є особливо високою. Саме прогнозування продажів є дуже важливим, для передбачення майбутнього обсягу збуту товару. Однак, постає питання, як обрати метод прогнозування і що взагалі потрібно розрахувати для того щоб дізнатися об'єм продаж на наступний період. Для новачків, які займаються продажем продукції власного виробництва, використання складних та професійних програм по управлінню бюджетом не має сенсу.

З огляду на все вище сказане, актуальними є питання підвищення ефективності прогнозування продажів товарів в програмних додатках для бізнесу. Це передбачає розробку нового адаптивного методу прогнозування продажів, який на основі автоматизованого аналізу наявних вхідних даних про продажі минулих періодів будував би прогноз за найбільш оптимальним методом прогнозування, а також розробку програмних засобів для реалізації запропонованого методу.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась відповідно до плану науково-дослідних робіт кафедри програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення ефективності прогнозування продажів товарів за рахунок автоматизованого підбору оптимального методу прогнозування, що відповідає наявним вхідним даним про продажі товару за минулі періоди.

Для цього необхідно виконати такі **завдання**:

- провести аналіз існуючих методів прогнозування продажів;
- запропонувати новий адаптивний метод прогнозування продажу товарів;
- розробити метод автоматизованого формування звіту прогнозування;
- розробити базу даних;
- розробити графічний інтерфейс програмного засобу;
- розробити програмні компоненти на основі запропонованих методів;
- провести тестування розроблених програмних засобів.

Об'єкт дослідження – процес прогнозування продажів товарів в програмних додатках для бізнесу.

Предмет дослідження – методи та засоби підвищення ефективності прогнозування продажів товарів.

Методи дослідження. У процесі дослідження використовувались: теорія чисел, лінійна алгебра та математична статистика для розробки адаптивного методу прогнозування продажів, теорія алгоритмів для розробки алгоритмів і програмних модулів, теорія баз даних для розробки структури бази даних, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Наукова новизна одержаних результатів.

1. Уперше запропоновано адаптивний метод прогнозування продажу товарів, особливість якого полягає в автоматизованому аналізі вхідних даних про продажі за минулі періоди та підборі найбільш оптимального методу прогнозування, що дозволило підвищити ефективність прогнозу.

2. Уперше запропоновано метод автоматизованого формування звіту прогнозування, який дозволяє фіксувати макроекономічні показники, що

дозволило підвищити ефективність оцінювання результатів прогнозування майбутніх обсягів збуту товарів.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та програмні засоби для підвищення ефективності прогнозування продажів товарів.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У науковій роботі, опублікованій у співавторстві, автору належать такі результати: розробка програмного забезпечення для прогнозування продажів товарів [2], розробка адаптивного підходу до підбору методу прогнозування продажів [3].

Апробація матеріалів магістерської кваліфікаційної роботи. Результати магістерської кваліфікаційної роботи доповідалися та обговорювалися на:

- IV міжнародній науковій конференції «Науковий простір: актуальні питання, досягнення та інновації» (Івано-Франківськ, 2022);
- Міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» (Вінниця, 2022).

Публікації. Основні результати досліджень опубліковано в 2 наукових працях, у матеріалах конференцій.

1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану питання

Успішна діяльність компаній у сфері продажу вимагає здійснення такої управлінської функції як прогнозування. В результаті опитування фінансових директорів України виявилось що 75% компаній, які приділяють увагу плануванню бюджету з 1 січня починають не лише новий рік а й новий бюджет. Тому наприкінці року активно відбувається бюджетування в ході якого проводять прогнозування майбутніх продажів. Для того що вчасно підготувати звіти по бюджету, деякі фінансисти компаній готують загальний опис основних параметрів майбутнього бюджету ще влітку. Один з таких параметрів є опису майбутніх обсягів збуту товарів. Такі параметри неможливо визначити без попереднього прогнозування продажів [4].

Прогнозування продажів — це використання інформації про поточні та минулі продажі для прогнозування продажів компанії на наступний місяць, квартал, півроку або рік. Ви можете використовувати методами прогнозування для виявлення та вирішення внутрішніх або зовнішніх проблем з продажами, щоб у вас було достатньо часу для досягнення ваших цілей.

Прогнозування продажів потребує актуальних знань, щоб передбачити майбутні зміни, тому на прогнози можуть впливати:

- нещодавнє зростання або спад розвитку індустрії;
- економіка загалом;
- продажі подібних товарів або послуг конкурентів;
- виведення на ринок останнього продукту або послуги компанії;
- коливання типових операційних витрат та цін продажу компанії;
- нові правові норми, що обмежують звичайну діяльність компанії;
- маркетингова стратегія компанії [5].

Хоча прогнози продажів екстраполюються з поточних даних, вони здебільшого стосуються майбутніх умов.

Прогнози продажу відіграють важливу роль у маркетинговій діяльності компанії. Вони є необхідними для організації нормальної роботи не тільки відділу маркетингу, але й інших функціональних підрозділів фірми.

Тому прогнозування продажів наразі є досить популярним, а нових методів прогнозування та відповідного програмного продукту, який би зміг спростити процес прогнозування для користувача і підвищити ефективність прогнозування, є актуальною.

1.2 Порівняльний аналіз аналогів

Існує досить велика кількість програм, які допомагають в управлінських процесах і містять в собі засоби для прогнозування продажу. Але більшість з них нагромаджені різними функціями, що ускладнює роботу і вимагає багато часу для опанування програми. Також майже всі вони сфокусовані саме на плануванні і управлінні компанією, а не виключно на розрахунку майбутніх продажів. Через велику кількість функцій, ці програми мають досить високі ціни на підписки, а більшість з них дає змогу вкласти угоду лише на тривалий період часу. Найбільш відомі існуючі програмні засоби для прогнозування продажів, які так чи інакше є близькими до створюваного програмного продукту по реалізації та призначенню є:

- Infor CloudSuite SCP;
- NetSuite Demand Planning;
- Anaplan.

Infor CloudSuite – це інтегрована платформа планування, яка дозволяє компаніям створювати, управляти, співпрацювати та координувати плани по всьому своєму ланцюжку постачання (рисунок 1.1).

Переваги Infor CloudSuite включають:

1. Можливість налаштування відображення даних інструментами візуалізації.
2. Відслідковування виробничого процесу від початку до кінця.

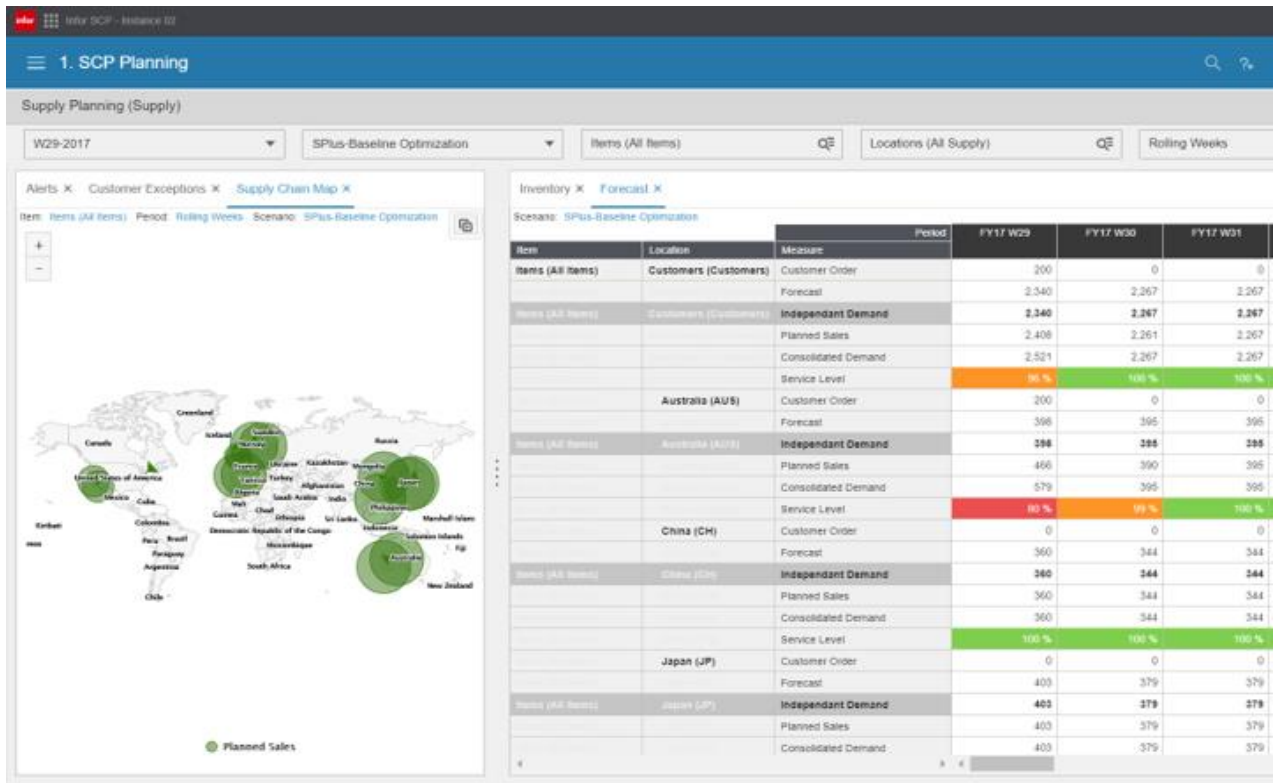


Рисунок 1.1 – Інтерфейс програми Infor CloudSuite

До недоліків можна віднести такі:

1. Перегляд форм займає багато часу.
2. Помилки введення даних важко відстежити.
3. Навчання складне і вимагає великої кількості введення даних.

NetSuite – це програмне забезпечення на базі Oracle. Це потужне програмне забезпечення з обширним функціоналом, який є хорошими і поганими одночасно. Від стартапів до прибутку до малих та середніх організацій, швидкозростаючий бізнес використовує NetSuite для автоматизації основних процесів та отримання реальних даних про операційні та фінансові показники (рисунок 1.2).

Переваги NetSuite:

1. Є функція додання нових бізнесів та філіалів по мірі зростання бізнесу.
2. Підходить для компаній будь якого розміру.

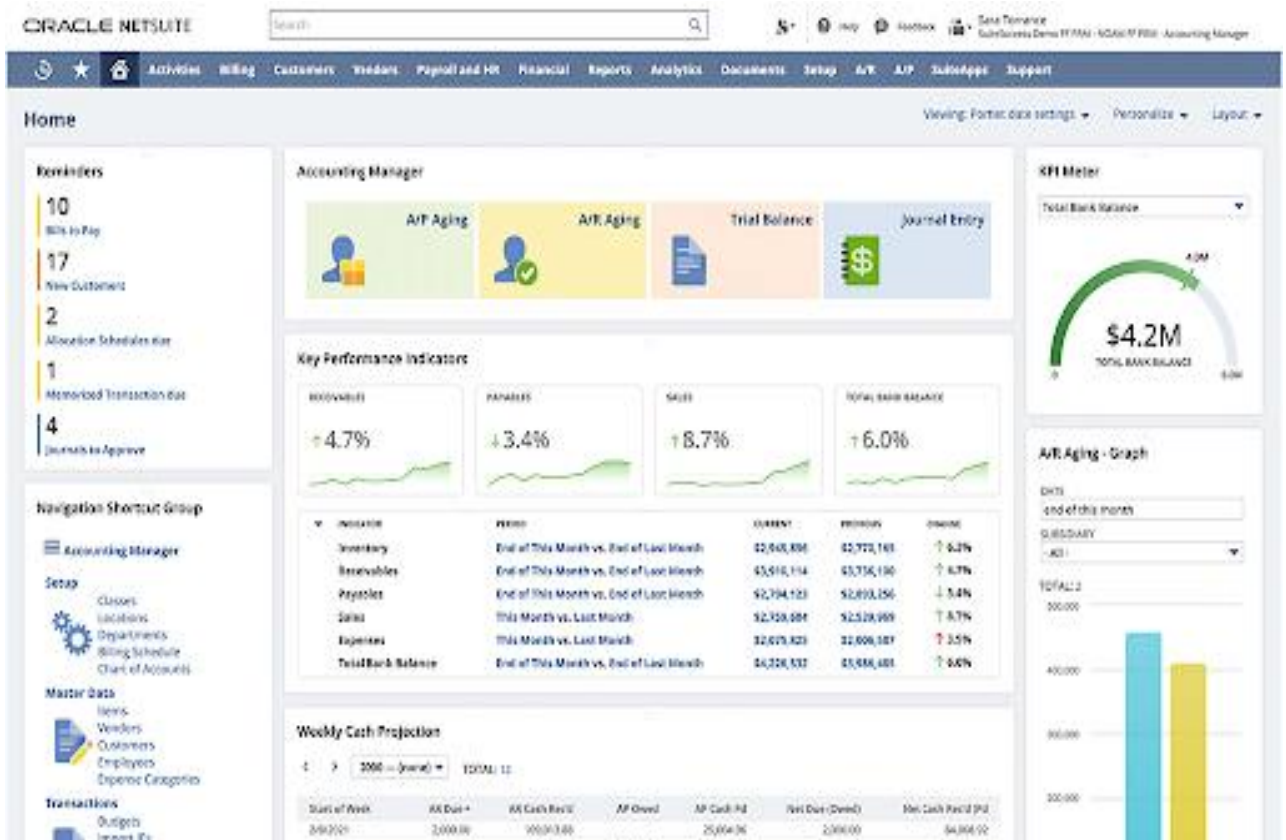


Рисунок 1.2 – Інтерфейс програми NetSuite

Недоліки NetSuite:

1. Багато несподіваних повідомлень про помилки та повідомлень про помилки, які спотворюють фактичну проблему.
2. Складний і нагромаджений інтерфейс.
3. Дорого вартісний, і не можна вносити корективи в ліцензії без тривалих контрактів.

Anaplan – це американська компанія з планування програмного забезпечення зі штаб-квартирою в Сан-Франциско, штат Каліфорнія. Anaplan продає підписку на хмарне програмне забезпечення для бізнес-планування та надає дані для прийняття рішень (рисунок 1.3).

Переваги Anaplan:

1. Підходить для компаній, які шукають ПЗ для побудови стратегії планування, прогнозування та складання бюджету, планування робочої сили та відстеження даних конкурентів.

Недоліки Anaplan:

1. Початкове налаштування вимагає багато навчання та часу. Їх тренінг триває кілька днів, і все ще вимагає додаткової подальшої роботи та спілкування з командою з впровадження.
2. Складний і нагромаджений інтерфейс.
3. Мало графіків та методів прогнозування [6].

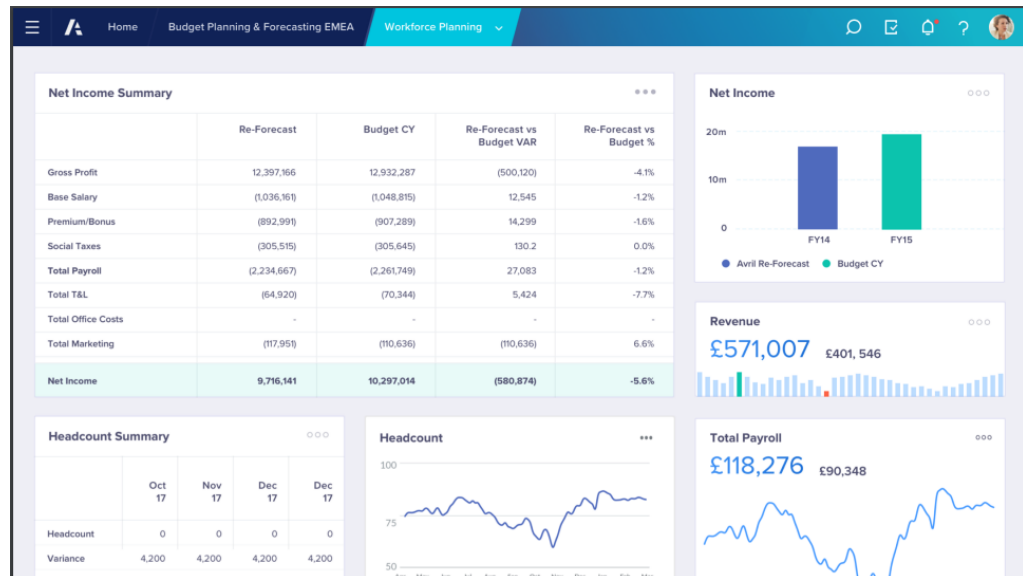


Рисунок 1.3 – Інтерфейс програми Anaplan

Результати порівняння аналогів зведено в табл. 1.1.

Таблиця 1.1 – Порівняльні характеристики програмних продуктів

Критерій	Infor CloudSuite	NetSuite	Anaplan	Власний додаток
Зручний інтрефейс	0	0	0	1
Безкоштовний пробний період	0	0	1	1
Легке навчання для використання	0	0	0	1
Можливість прогнозування продажу	1	1	1	1
Сума	1	2	2	4

В результаті проведеного аналізу було поставлено задачу – розробити такий програмний продукт, за допомогою якого можна було б швидко переглянути тенденції продажу товарів, спрогнозувати майбутні продажі найпоширенішими методами прогнозування, які найкраще відповідають наявним даним про продажі минулих періодів, а саме головне, щоб інтерфейс був легкий та зрозумілий кожному.

1.3 Аналіз методів прогнозування продажу

Методи прогнозування поділяються на два основні види це інтуїтивні і формалізовані методи (рис.1.4).

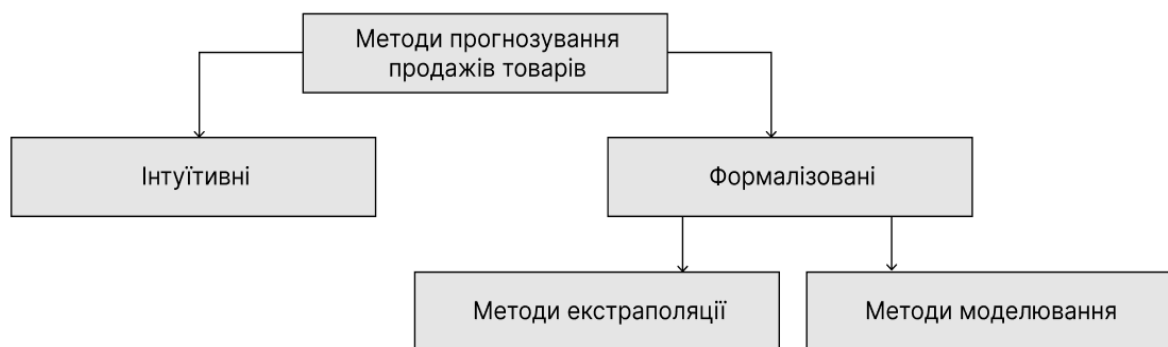


Рисунок 1.4 – Класифікація методів прогнозування

Інтуїтивні методи – вид методів прогнозування, оснований на використанні експертних оцінок, за допомогою яких припускається поведінка об'єкта прогнозування в майбутньому. Зазвичай використовуються, коли через складність об'єкта прогнозування неможливо врахувати велику кількість факторів, що впливають на об'єкт.

До інтуїтивних методів належать:

- Індивідуальні експертні оцінки:
 - інтерв'ю;
 - аналітичні записки;
 - побудова сценаріїв;
- Колективні експертні оцінки:

- метод мозкової атаки;
- метод комісій;
- метод Делфі;
- метод дерева цілей;
- метричний метод.

Під час прогнозування інтуїтивними методами проходять чотири основні етапи. Перший етап це організація процедури експертного опитування. Для ефективного вирішення проблеми важлива раціональна організація проведення експертами її аналізу з кількісною оцінкою їх суджень і математико-статистичним опрацюванням отриманих результатів. Підсумкове думку експертів приймається як вирішення проблеми.

На другому етапі обирають експертів і формують так звані експертні групи. Можуть бути такі групи: керівник групи прогнозування, організаційна група, експерти, група обробки даних. Під час проведення опитування в середньому необхідно задіяти 120-150 учасників. Основними видами опитувань є: анкетування; інтерв'ювання; мозковий штурм; дискусія.

Четвертий етап – обробка результатів опитування. В ході попереднього етапу експерти дають оцінки різного характеру, які залежать від їхнього досвіду та інтуїції. Тому при обробці результатів застосовують закон розподілу ймовірностей значень або, наприклад, метод Делфі [7].

Формалізовані методи – засновані на математичних методах, що дозволять збільшити точність, достовірність і прискорити процес розробки прогнозу (рис.1.5).

При формуванні прогнозів за допомогою методів екстраполяції виходять зі статистично склавшихся тенденцій зміни попиту. Вихідними даними для їх застосування є часові ряди, які складаються з упорядкованих в часі рівнів (значень) обсягів продажів у минулому.

Є такі основні кількісні методи прогнозування:

- наївний;
- арифметичне згладжування;

- експоненційне згладжування;
- метод Брауна;
- екстраполяція тренду;
- метод Вінтерса та інші.

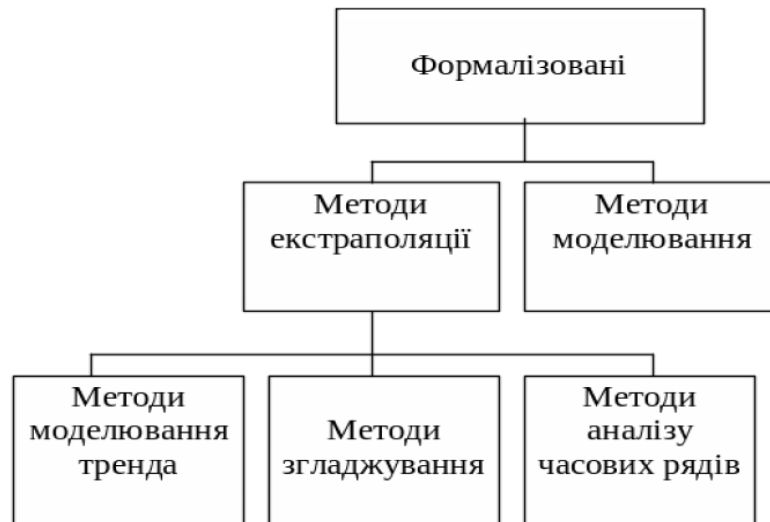


Рисунок 1.5 – Класифікація формалізованих методів прогнозування

Отже, метод прогнозування обирається з огляду на особливості бізнесу, кількість доступних коштів, що виділені на прогнозування, час, який виділено на проведення прогнозу. Важливо чітко розрізняти методи прогнозування, адже не правильно підібраний метод може призвести до безглузвих результатів, які не будуть мати ніякої цінності, а кошти та час будуть вже витрачені.

1.4 Аналіз методик складання прогнозу продажів

Для підвищення ефективності прогнозування продажів, потрібно вміти комбінувати різні види методів прогнозування залежно від умов та особливостей підприємства чи компанії. Саме комплексний підхід та зіставлення отриманих результатів допоможе скласти чітку картину майбутніх продажів. Тому вибір методики прогнозування є важливою складовою процесу передбачення майбутнього збуту продукції.

Поняття методика прогнозування являє собою сукупність методів та технік для складання прогнозу попиту на продукцію.

При складанні стратегії прогнозування експерти поєднують інтуїтивні та формалізовані методи, комбінуючи їх таким чином щоб вони ефективно показували результат для певного формату компанії та типу продукції.

Одна з методик проведення прогнозу попиту пропонує такі кроки:

1. Визначити, для якої продукції буде складено прогноз.
2. Визначити термін, на який буде складено прогноз.
3. Обрати одиниці виміру.
4. Визначити основні методи для складання прогнозу.
5. Визначити джерела збору інформації для прогнозу.
6. Розрахувати ймовірні варіанти прогнозів по кожному методу.
7. Розрахувати оптимістичні і песимістичні результати.
8. Зіставити отриманні дані.

Інша методика пропонує наступне:

1. Опитати групи керівників різних служб і відділів.
2. Узагальнити оцінки окремих експертів компанії.
3. Провести прогнозування на базі минулого попиту.
4. Проаналізувати тенденції та цикли.
5. Проаналізувати «частку ринку».
6. Провести аналіз кінцевого використання.
7. Провести аналіз асортименту товарів [8].

Усі ці методики прогнозування включаються в себе інтуїтивні методи, які базуються на опитуванні людей та вимагають експертних оцінок. Такі методи займають багато часу та є дорого вартісними. Вони не підходять для тих, кому потрібен швидкий, але в той самий час достовірний прогноз, з мінімальним бюджетом та за короткий проміжок часу. Для власників малого та приватного бізнесу немає потреби у експертах та групі людей для оцінки ринку і прогнозу попиту.

Отже, проаналізувавши існуючі підходи до проведення прогнозів попиту, було вирішено створити власну методику, яка б дозволила провести всі розрахунки автоматично та швидко, але водночас було б достатньо інформації для оцінки можливостей компанії на ринку.

1.5 Постановка задач розробки

Після аналізу поточного стану питання та порівняння існуючих рішень за визначеними критеріями визначено завдання, які необхідно виконати, для розробки системи:

1. Провести аналіз існуючих методів прогнозування продажів.
2. Запропонувати новий адаптивний метод прогнозування продажу товарів.
3. Розробити метод автоматизованого формування звіту з прогнозування.
4. Розробити базу даних.
5. Розробити графічний інтерфейс програмного засобу.
6. Розробити програмні компоненти на основі запропонованих методів.
7. Провести тестування розроблених програмних засобів.

Технічне завдання наведено у Додатку А.

1.6 Висновки

У першому розділі було розглянуто стан питання прогнозування продажів. Проаналізовано існуючі засоби, а саме Infor CloudSuite SCP, NetSuite Demand Planning, Anaplan, проведено їх порівняння з розроблюваним додатком. У результаті порівняння було доведено доцільність розробки власного програмного продукту, який покриває усі недоліки існуючих рішень.

Проведено аналіз існуючих підходів до вирішення поставленої задачі. У результаті аналізу було обрано методи реалізації необхідного функціоналу.

Встановлено основні завдання, які необхідно виконати для розробки програмного продукту.

2 РОЗРОБКА СТРУКТУРИ СИСТЕМИ ТА МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОГНОЗУВАННЯ ПРОДАЖІВ ТОВАРІВ

2.1 Аналіз даних

Прогноз продажів визначає, якими будуть фактичні продажі компанії за певного ступеня маркетингових зусиль компанії. Зі свого боку, потенціал збуту оцінює, які продажі можливі на різних рівнях маркетингових зусиль, за умов наявності певних умов навколишнього середовища.

Важливо проводити своєчасний прогноз, для ефективного планування бюджету та закупок товарів. Прогноз можна проводити раз на рік, або кілька разів на рік зменшивши період майбутнього прогнозу.

Розроблено програмний продукт, який дає змогу спрогнозувати продажі непоширенішими методами прогнозування. Запропоновано методику для побудови прогнозу, яка включає ряд пунктів: визначення макроекономічних показників ринку, розрахунок майбутніх витрат, підбір методу прогнозування з огляду на стан та тип бізнесу, прогнозування продажу, формування звіту.

Користувач при роботі з системою повинен зареєструватись або увійти в систему, якщо в нього вже є аккаунт. Після авторизації розпочинається робота з додатком. Можна спрогнозувати продажі на майбутні періоди, переглянути минулі прогнози для оцінки їх ефективності.

При прогнозуванні продажу важливим етапом є вибір методу, який буде розраховувати майбутній об'єм проданого товару. Для полегшення цього етапу було розроблено метод, який автоматично обирає техніку прогнозування, що буде найкраще підходити для бізнесу користувача. Потрібно відповісти на кілька питань, обравши варіант відповідей з перелічених. В результаті програма самостійно опираючись на дані відповідей підбере метод прогнозування.

В особистому кабінеті користувач може змінити своє ім'я, логін, пароль або аватар. Також тут знаходитиметься кнопка виходу з систему.

Усі дані про користувача, звіти прогнозів та інформація про методи прогнозування зберігаються в базі даних.

2.2 Розробка адаптивного методу прогнозування продажів товарів

Вибір правильного методу прогнозування є що не найважливішим кроком у прогнозування. Якщо обрати не той метод, прогноз буде критично відрізнятись від реального, що призведе до хибних висновків про майбутні показники продажів, в результаті чого буде створено неефективну стратегію компанії на ринку. Також потрібно враховувати очікування від прогнозу, чи потрібен тренд, чи враховувати сезонність, на який період потрібно розрахувати майбутні продажі [9].

Спочатку розглянемо чим математичні методи прогнозування відрізняються один від одного і в яких випадках їх потрібно використовувати.

1. Арифметичне згладжування – це один за найпростіших методів, полягає в тому, що прогноз дорівнює середньому арифметичному значенню продажів за всі минулі періоди.

2. Ковзні середні популярні з двох причин: по-перше, тому що ними легко користуватися; по-друге, тому що вони добре працюють у визначенні тенденції. Є кілька варіацій на тему. Прості ковзні середні обчислюються додаванням значень, як правило, за встановлену кількість періодів, а потім діленням суми на кількість періодів. Зважені ковзні середні обчислюються шляхом надання більшої ваги останнім даним. Тому вони більш чутливі до останніх коливань цін.

Ковзні середні фільтрують шум і полегшують визначення тенденцій. Якщо лінія ковзної середньої висхідна, то ринок перебуває у висхідному тренді, і навпаки. Поєднання двох ковзних середніх на одному графіку зазвичай робить картину ще більш чіткою.

Метод Хольта-Вінтерса (ковзкої середньої) (2.1) – це спосіб моделювання трьох аспектів часового ряду: типове значення (середнє), нахил (тенденція) у часі та циклічний повторюваний шаблон (сезонність).

$$y_{t+1} = m_{t+1} + \frac{1}{n} * (y_t - y_{t-1}) \quad (2.1)$$

3. Експоненційне згладжування — це техніка ковзкої середньої, яка особливо ефективна, коли потрібне часте повторне прогнозування та коли прогнози мають бути виконані швидко. Це метод короткострокового прогнозування, який часто використовується у середовищі виробництва та запасів, де потрібно прогнозувати лише значення наступного періоду. Оскільки для виконання експоненціального згладжування потрібні лише три числа, цю техніку легко оновити. Потрібними даними є історичне спостереження, останні дані спостереження та коефіцієнт згладжування або константа.

Коефіцієнт згладжування α – це значення від 0 до 1. Невелике значення, скажімо, від 0,05 до 0,10, призводить до високого ступеня згладжування та має такий самий ефект, як велика кількість спостережень у обчисленні ковзного середнього. Високе значення коефіцієнта призводить до меншого згладжування і, отже, до високої чутливості до варіацій даних. Більше значення коефіцієнту α відноситься до новіших періодів.

4. Правило півтора (“Правило 1.5”) частіше за все використовують у роздрібній торгівлі торгові агенти або якщо підприємство починає працювати з новим товаром, за яким немає жодної статистики продажів. Для того щоб визначити об'єм замовлення на наступний період (день, тиждень, місяць), необхідно дотримуватися кроку 1,5. Правило можна записати за допомогою формули (2.2).

$$Z_{ам} = (Z_{п.п} + P_{ост} - Z_{к.п}) * 1,5 - Z_{к.п} \quad (2.2)$$

де $Z_{ам}$ – новий обсяг замовлення, шт.; $Z_{п.п}$ – залишки на початок звітного періоду, шт.; $P_{ост}$ – постачання у звітному періоді, шт.; $Z_{к.п}$ – залишки на кінець звітного періоду, шт.

5. Метод прогнозування сезонного попиту – оснований на коефіцієнті тенденції. Цей коефіцієнт показує, наскільки змінилися продажі порівняно з попередніми роками. Використовується для прогнозування продажів товарів сезонного попиту, для цього порівнюють такі самі періоди минулих років та

коригують отримані дані за допомогою експертних оцінок. Порівняння періодів необхідне для врахування зміни в тенденціях — фіксації зростання або зменшення попиту порівняно з аналогічними періодами минулих років. Зробити це допомагає коефіцієнт тенденції, знаходиться він за формулою (2.3) [10].

$$K = \frac{x_1 - x_2}{x_2} * 100\% \quad (2.3)$$

Отже, ознайомившись з усіма техніками прогнозування і обравши найпопулярніші, можна обрати критерії за якими програмний засіб буде підбирати метод прогнозування.

Для того щоб автоматично підібрати метод прогнозування, користувачу необхідно на екрані обрати ті варіанти відповідей на питання, які притаманні його бізнесу. Питання на які користувачу потрібно дати відповідь.

1. Чи залежить продаж товару від сезону?
2. Ви давно працюєте з цим товаром?
3. Скільки є попередніх періодів?
4. Необхідний тренд?
5. На який період буде проведено прогнозу

В залежності від обраних варіантів відповідей, буде підібрано метод прогнозування. Наприклад:

Якщо в нас немає багато даних за минулий період, не потрібна сезонність та тренд, прогноз потрібен на наступний період, в такому випадку підходить арифметичне згладжування або експоненційне згладжування.

Або, наприклад, коли користувачу потрібен прогноз на середній або на великий строк, тренд необов'язковий, і якщо даних за минулий період не велика кількість в даному випадку підійде метод Хольта–Вінтерса. В таблиці 2.1 наведено з відповіді на питання, які стосуються кожного методу прогнозування.

Таблиця 2.1 – Характеристика методів прогнозування

Метод Питання	Арифм. згладж.	Метод Хольта Вінтерса	Експоненційне згладжування	Правило 1.5	Прогнозува ння для сезонності
Чи залежить продаж товару від сезону?	-	+	-	-	+
Ви давно працюєте з цим товаром?	-	+	+	-	+
Скільки є попередніх періодів?	немає різниці	дуже багато	середня	середня	певного сезону
Необхідний тренд?	-	+	+	-	-
На який період буде проведено прогнозу	наступний	середній, довго- строковий	наступний, середній	наступний	наступний, середній, довго- строковий

Отже, якщо користувач – новачок, і не знається на методах прогнозування, він може покластись на систему. Йому потрібно відповісти на кілька питань, обравши варіанти відповідей, тоді система автоматично обере метод прогнозування. В майбутньому можна збільшувати кількість методів прогнозування.

2.2 Розробка методу формування звіту з прогнозування продажів

Для планування майбутніх продажів не вистачить лише одного прогнозу. Потрібно провести деякі додаткові обрахунки, наприклад, визначити стан ринку, щоб врахувати його вплив на майбутні продажі. Також важливими є, наприклад, макроекономічні індикатори. Людині, яка вперше зіштовхується з плануванням продажів, важко зорієнтуватися та визначитись з тим, що саме потрібно робити. Тому було вирішено розробити метод формування звіту з прогнозування продажів, який також міститиме інформацію про макроекономічні показники, що дозволить підвищити ефективність оцінювання результатів прогнозування майбутніх обсягів збуту товарів і буде корисним будь-якому підприємцю.

Метод формування звіту з прогнозування складається з 8 кроків. Кожен крок наближає нас до звіту, який буде потім збережений в особистому кабінеті і за допомогою якого можна отримати уявлення не тільки про спроможність продукту приносити прибуток, а й про стан ринку загалом.

Крок 1.

Перший крок – вступ. Вводяться дані про бізнес (назва, контакти, важливі посилання, тип продукції, скільки товар вже на ринку). Так як компанія може продавати різну продукцію, а прогнозування може робитися для кожного виду товару окремо, важливо все це зазначати у звітах.

Крок 2.

Далі потрібно проаналізувати ринок, на екрані з'явиться інформація про деякі макроекономічні індикатори, такі як ВВП та рівень інфляції. Інформація буде оновлюватись автоматично. Актуальна інформація буде збережена у звіті, в майбутньому її можна буде порівняти.

Крок 3.

Оцінка рентабельності продукту. Деяким цей пункт може бути непотрібним. Але, якщо це новий бізнес або продажі стрімко падають, потрібно розрахувати чи підходить обрана стратегія, можливо потрібно внести якісь зміни.

Крок 4.

Встановлення цілей. Наприклад, «збільшити продажі на 20%». В майбутньому користувач зможе оцінити чи вдалося втілити задуми у життя і можливо встановити іншу важчу ціль. Цей крок збільшує мотивацію та дає наснаги працювати плідно.

Крок 5.

Кожен хто має свій бізнес знає, які в нього конкуренти. На цьому кроці виносяться основні конкуренти, можна додати опис, що в них є такого кращого, що можна було б додати в себе. В майбутньому можна буде оцінити зміни, а можливо старі конкуренти перестануть бути конкурентами взагалі.

Крок 6.

Вибір методу прогнозування. Користувач може обрати метод самостійно, або за допомогою системи. У програмі на даний момент можна провести прогнозування продажів п'ятьма основними методами: арифметичне згладжування, метод Хольта-Вінтерса, Експоненціальне згладжування, правило 1.5, екстраполяція тренду. Для автоматичного підбору методу, користувач має відповісти на 6 питань, після чого переходить на наступний крок.

Крок 7.

На цьому кроці відбувається саме прогнозування. З'являється екран, зверху написано метод прогнозування, нижче наведено поля для вводу даних для обчислень. Коли всі дані введено, потрібно натиснути на кнопку «розрахувати» і очікувати результат.

Крок 8.

Це фініш. Формується звіт. Зберігається в базу даних. Його можна надрукувати.

Отже, сформований звіт включатиме основні моменти по плануванню продажів. Користувач отримає чітке розуміння про його бізнес, спроможність продукції на ринку, власне і про стан ринку, а також прогноз на обраний період. Таке планування витрачає мінімум часу та коштів, не потрібно проводити анкетування, збирати та опитувати групи експертів, але в той же час є ефективним, так як метод прогнозування правильно підібраний для компанії, користувач оцінює картину загалом і не спирається лише на цифри, створює власне судження про майбутнє.

2.4 Розробка структури інтерфейсу та вибір кольорової гами для програмного засобу прогнозування продажів

Для ефективного використання додатку було розроблено його інтерфейс за допомогою Windows Forms.

Після запуску додатку з'являється екран входу в систему (рис.2.1). Потрібно ввести логін і пароль. У випадку коли немає аккаунту обрати пункт «Реєстрація».

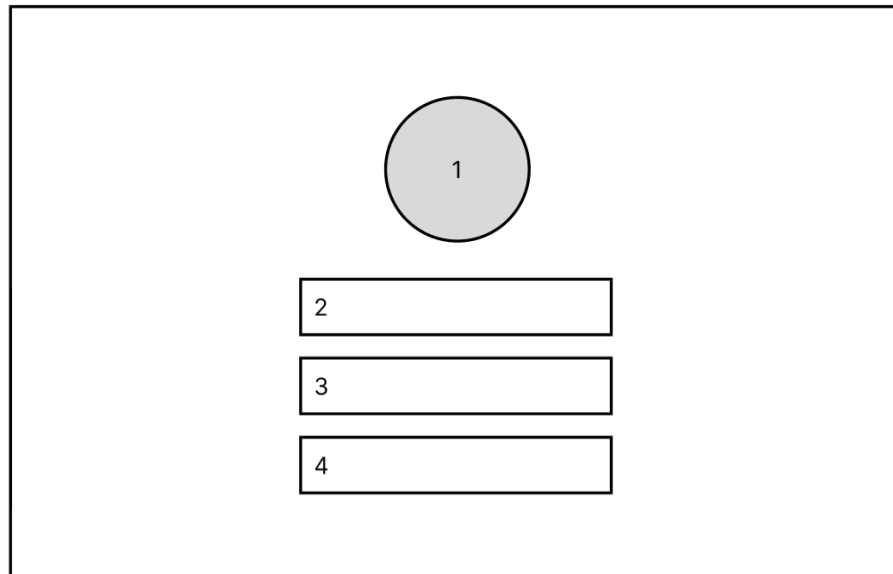


Рисунок 2.1 – Екран входу в систему

Форма входу в систему містить такі складові:

1. Логотип додатку.
2. Логін.
3. Пароль.
4. Реєстрація.

Після входу в систему користувач може перейти до роботи з додатком. На екрані з'являється головний екран (рис.2.2). Зверху знаходиться меню. Також є основна робоча область, яка змінюється з кожним кроком прогнозування.

Форма головного екрану містить такі складові:

1. Меню.
2. Робоча область.
3. Прогнозування продажів.
4. Звіти.
5. Особистий кабінет та іконка аватару користувача.
6. Назва методу прогнозування або номер звіту.

7. Область для виводу та вводу інформації.
8. Кнопка відміни або «Назад».
9. Кнопка «Далі».

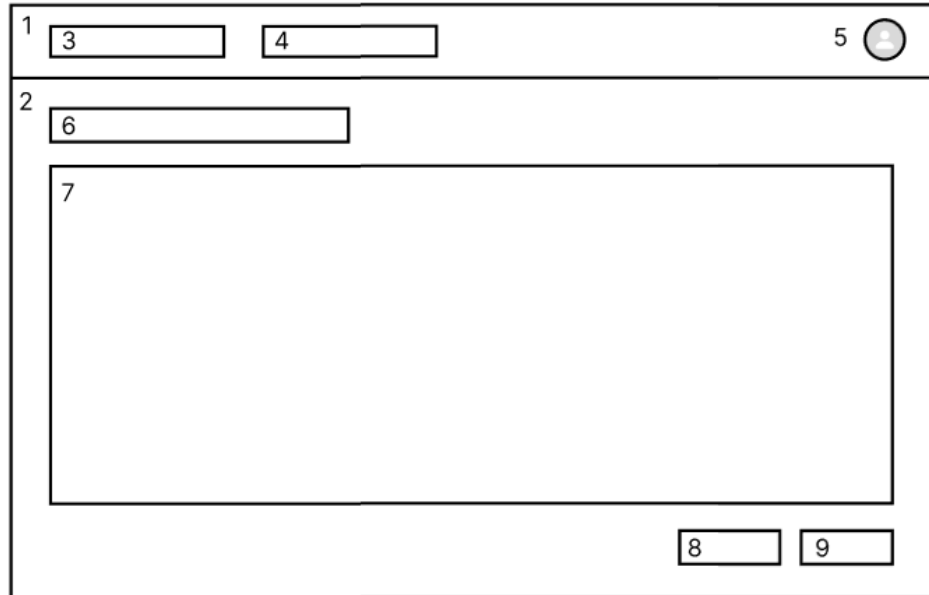


Рисунок 2.2 – Головний екран

Натиснувши на іконку свого аватару, користувач потрапляє в особистий кабінет (рис. 2.3). Тут він може зміни свої особисті дані (ім'я, пароль, логін, аватар) та вийти з облікового запису.

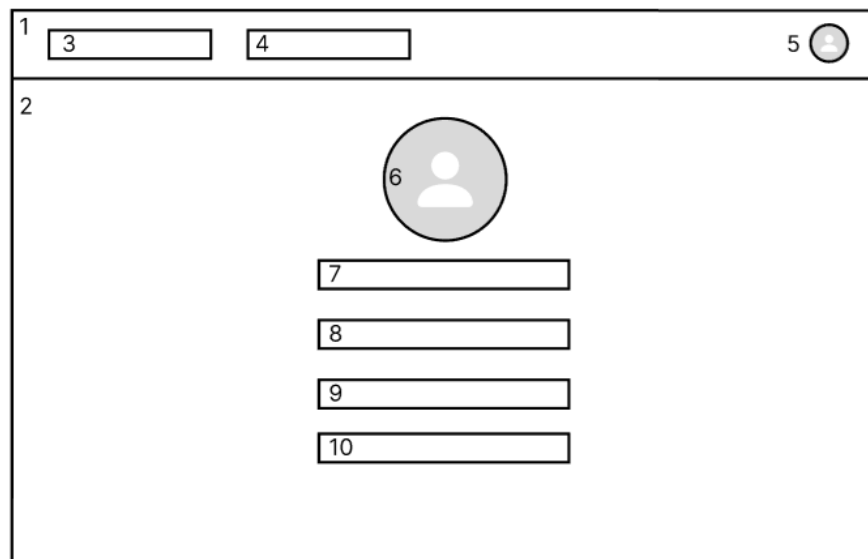


Рисунок 2.3 – Особистий кабінет

Форма вікна особистого кабінету містить такі складові:

1. Меню.
2. Робоча область.
3. Прогнозування продажів.
4. Звіти.
5. Особистий кабінет та іконка аватару користувача.
6. Аватар користувача
7. Ім'я користувача.
8. Логін.
9. Пароль.
10. Вихід з системи.

Колір відіграє важливу роль у кожному аспекті нашого життя і часто може описати це краще, ніж слова. Так само колір відіграє важливу роль у дизайні програм чи веб-сайтів. Іноді незважаючи на чудові функції та вміст сайту чи програми, йому не вдається залучити й утримати користувачів. Кольори програми мають відображати призначення програми, підкреслювати важливі частини та направляти користувача. Тому при розробці додатків дизайнери повинні підібрати правильну комбінацію кольорів і врахувати можливість їх поєднання з іншими елементами веб-дизайну. Перед вибором кольору слід враховувати різні фактори, а для цього потрібно розуміти колір і його значення.

З точки зору людської психології, поєднання кольорів може суттєво вплинути на сприйняття людиною інформації, яку надає картинка. Тому при виборі поєднання кольорів, наприклад тексту з фоном, рекомендується, щоб текст легко читався, при цьому читач не втомлювався [11].

Основними кольорами було вибрано: зелений, чорний, білий, червоний.

Кольори інтерфейсу зображено на рисунку 2.3.



Рисунок 2.4 – Кольори інтерфейсу

Зелений – колір, який асоціюється з природою, сприймається невимушено та налаштовує на спокій. У дизайні програм зелений колір викликає позитивні емоції, почуття гармонії і розслаблення.

Білий – асоціюється з легкістю та простотою, підходить для програма будь-якого призначення, тому він достатньо популярний. Білий використовується зазвичай як фон для додатків, завдяки ньому збільшується контраст та можна використовувати будь-який акцентний колір, адже поєднується з усіма кольорами. Білий не буде відволікати увагу користувача.

Червоний – яскравий акцентний колір, який досить активно привертає на себе увагу, зазвичай використовується для елементів, які відображають помилку або показують важливі повідомлення. Асоціюється з коханням та яскравими емоціями.

Сірий – зазвичай використовується для об'єктів які розділяють частини екрану, також є досить хорошим фоном так як не привертає багато уваги. Асоціюється з серйозністю.

Чорний – колір елегантності, строгості. асоціюється з технікою та чимось вишуканим. Дуже контрастний. Зазвичай використовується для текстів та не великих об'єктів.[12].

Отже, в якості кольору фону було обрано білий, так як він мінімалістичний та не привертає особливої уваги. Меню буде стриманого, контрастного чорного кольору. Деякі елементи та графіки будуть червоного та зеленого кольорів.

2.5 Розробка алгоритмів роботи програми

Діаграми варіантів використання це модель, яка показує можливі способи використання програми актором. Варіант використання – це функція системи, яка надає користувачу певний вимірювальний та відчутний результат. Варіант використання відповідає окремому сервісу, що надається модельованою системою, коли користувач робить запит, тобто обирає спосіб використання цієї системи. Зазвичай діаграми такого типу використовуються для специфікації

зовнішніх вимог до проєктної системи або функціональної поведінки вже існуючої системи [13].

Діаграма варіантів використання наведена на рис.2.5.

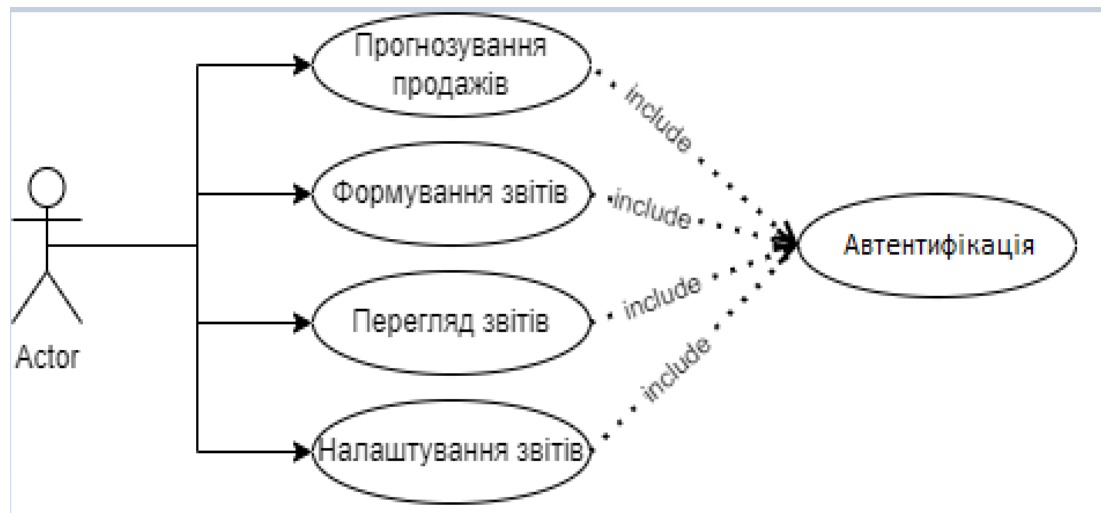


Рисунок 2.5 – Діаграма варіантів використання

На діаграмі показані основні взаємодії користувача з системою, а саме: прогнозування продажів, налаштування профілю, формування і перегляд звітів. Також на діаграмі видно, що для виконання будь-яких дій у програмі потрібно буде зареєстрованим та авторизованим, тільки тоді будуть доступні функції для роботи.

Розроблено алгоритм авторизації користувача. Спочатку перевіряється чи є акаунт. Якщо акаунт є – здійснюється вхід. Вводиться логін, система перевіряє правильність введення, також вводиться пароль, якщо логін введено неправильно пропонується ввести ще раз, якщо ж пароль введено не правильно – то є два варіанти або спробувати ще раз, або відновити пароль, в такому випадку новий пароль буде надіслано на пошту. Якщо всі дані введені правильно здійснюється вхід. Інший варіант – якщо немає профілю. Відбувається реєстрація користувача. Вводяться особисті дані, користувач додається до бази даних, оброблюється запит та створюється профіль, після чого здійснюється вхід в систему. Блок-схему наведено на рисунку 2.6.

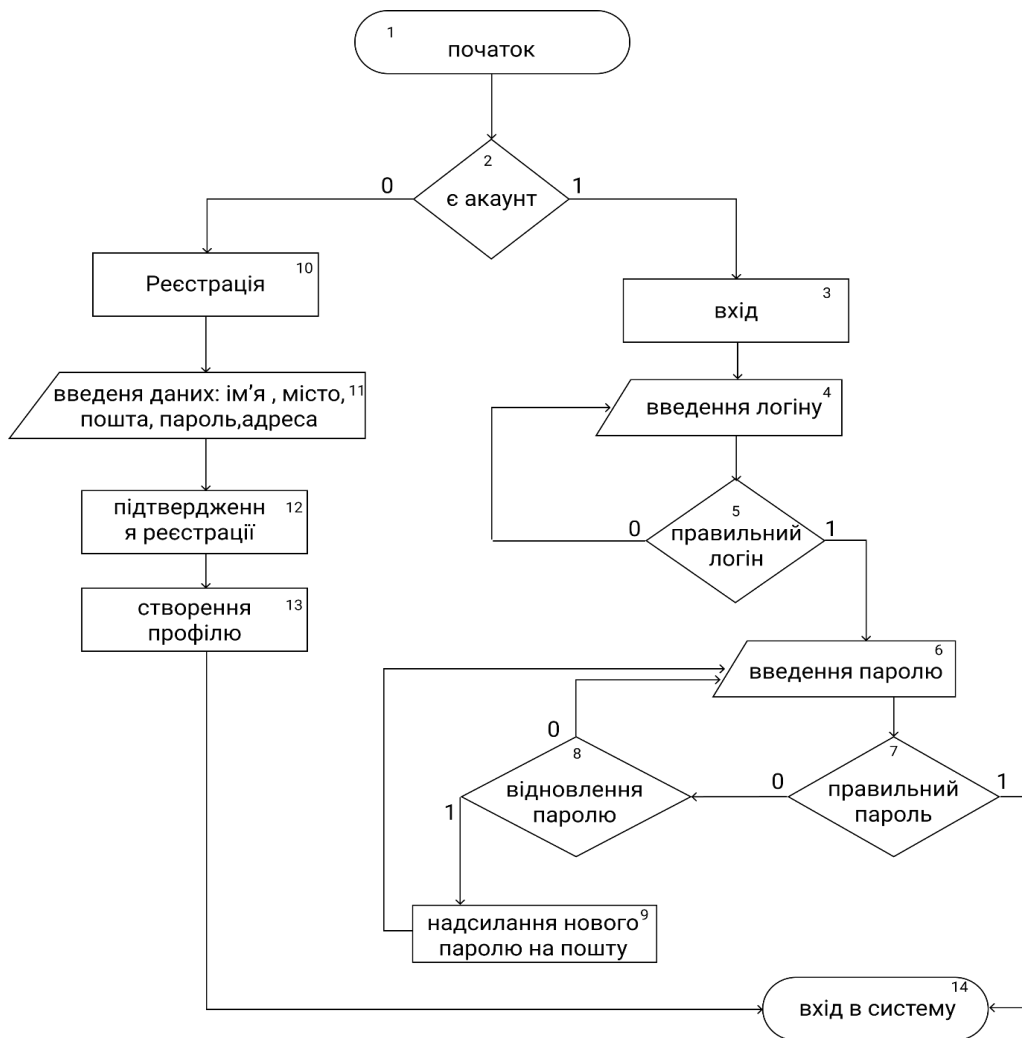


Рисунок 2.6 – Блок-схема алгоритму авторизації

Опис алгоритму авторизації:

Крок 1. Натискання кнопки профілю у шапці сайту.

Крок 2. Вибір між реєстрацією та входом.

Крок 3. Вхід в систему.

Крок 4. Введення логін.

Крок 5. Перевірка введених даних.

Крок 6. Введення паролю.

Крок 7. Перевірка паролю.

Крок 8. При неправильному введенні паролю пропонується відновлення паролю.

Крок 9. Надсилання нового тимчасового паролю на пошту.

Крок 10. Вибір реєстрації.

Крок 11. Введення особистих даних для реєстрації.

Крок 12. Підтвердження системи.

Крок 13. Створення профілю.

Крок 14. Вхід в систему.

Розроблено алгоритм формування звіту прогнозування. Блок-схему зображено на рисунку 2.7.

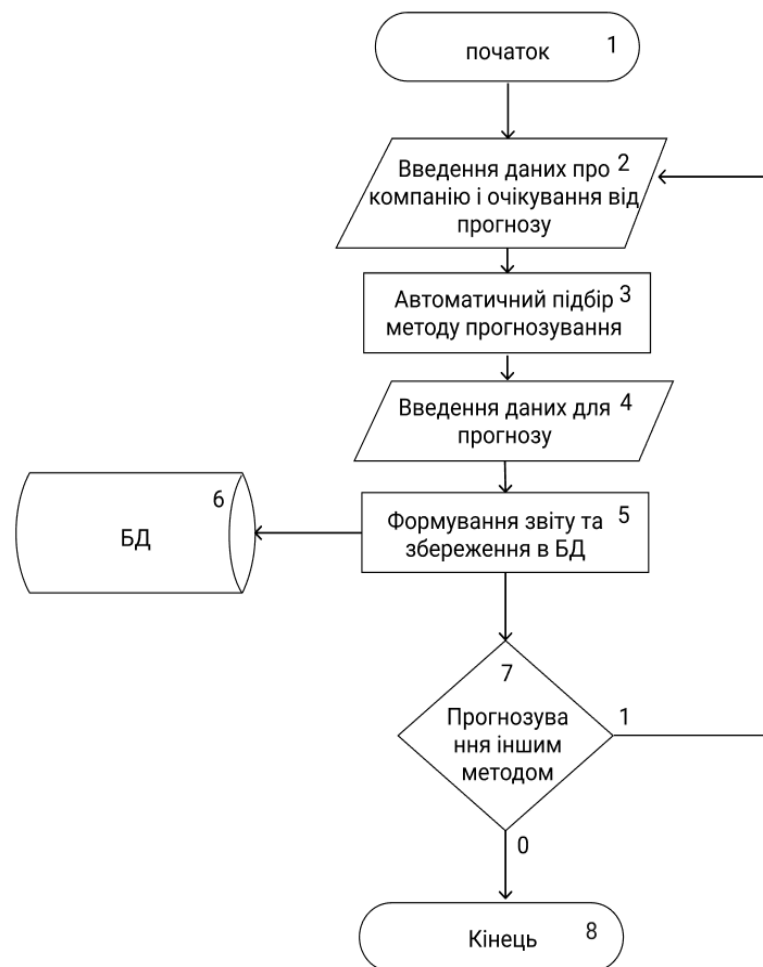


Рисунок 2.7 – Блок-схема алгоритму формування звіту прогнозування

Обирається метод прогнозування, вводяться необхідні дані. Формується звіт, який зберігається в базі даних. Потім можна спрогнозувати продажі іншим методом або вийти з системи.

Опис алгоритму додання товару у кошик:

- Крок 1. Початок.
- Крок 2. Введення даних про бізнес та очікувань від прогнозу.
- Крок 3. Програма автоматично підбирає метод прогнозування..
- Крок 4. Введення даних попередніх періодів для прогнозу..
- Крок 5. Формування звіту.
- Крок 6. Збереження звіту в БД.
- Крок 7. Прогнозування іншим методом за бажанням.
- Крок 8. Кінець.

На загальній блок-схемі зображений весь алгоритм використання програми. Спочатку користувач може ввійти в свій аккаунт в системі або ж ні. Якщо в нього немає аккаунту він може зареєструватись. Всі його особисті дані будуть збережені в Базі даних. Після авторизації користувач обирає з пункту меню необхідну йому функцію. Він може спрогнозувати майбутні продажі, обравши відповідний пункт меню. Після чого він обиратиме потрібний йому метод прогнозування, введе необхідні дані, відбудеться формування прогнозу, а звіт буде збережений у базі даних, потім його можна буде переглянути в особистому кабінеті.

Також користувач може обрати пункт перегляду статистики продажів. Після вибору даного пункту меню, йому треба буде ввести необхідні дані. Перед ним з'явиться графік, який теж можна зберегти і переглянути потім. Коли всі необхідні дії виконано користувач виходить з системи, на цьому робота програми завершується.

Загальний алгоритм роботи програми наведено на рисунку 2.8.

Опис загального алгоритму роботи програми:

- Крок 1. Початок.
- Крок 2. Перевірка наявності аккаунту.
- Крок 3. Вхід в систему.
- Крок 4. Отримання даних користувача.
- Крок 5. Збереження даних.
- Крок 6. Створення аккаунту.

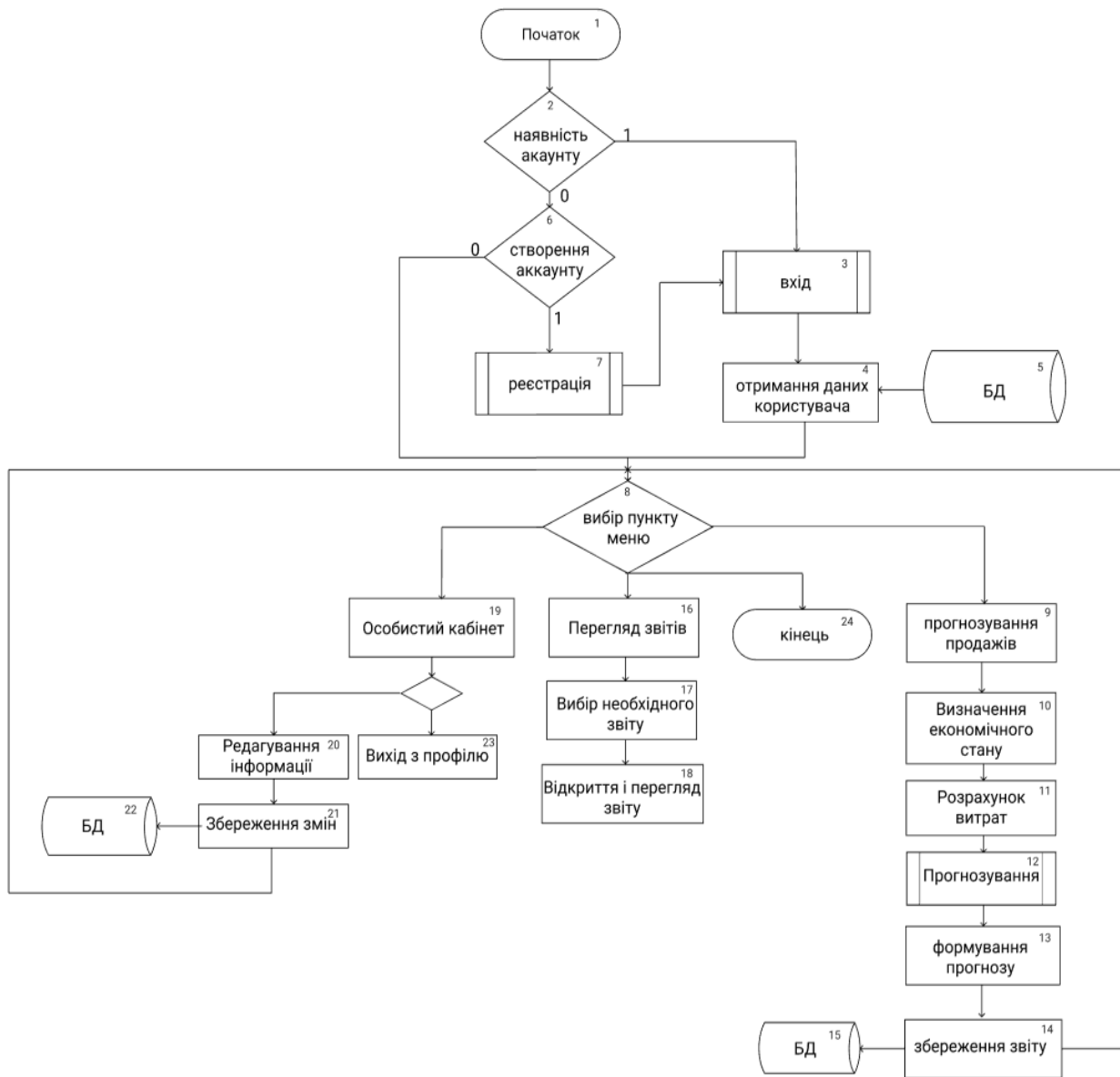


Рисунок 2.8 – Загальна блок-схема алгоритму роботи програми

- Крок 7. Реєстрація.
- Крок 8. Вибір пункту меню.
- Крок 9. Вибір пункту «Прогнозування продажів».
- Крок 10. Визначення економічного стану.
- Крок 11. Розрахунок витрат.
- Крок 12. Прогнозування, алгоритм наведено на рисунку 2.8.
- Крок 13. Формування прогнозу.
- Крок 14. Збереження звіту.
- Крок 15. Додання звіту до бази даних.

Крок 16. Вибір пункту меню «Перегляд звітів».

Крок 17. Вибір необхідного звіту.

Крок 18. Перегляд звіту.

Крок 19. Перехід до особистого кабінету.

Крок 20. Редагування особистої інформації.

Крок 21. Збереження змін.

Крок 22. Додання змін в базу даних.

Крок 23. Вихід з профілю.

Крок 24. Вихід з системи.

Для збереження даних про користувача та звітів з результатами прогнозування було розроблено базу даних. Модель бази даних наведено на рисунку 2.9.

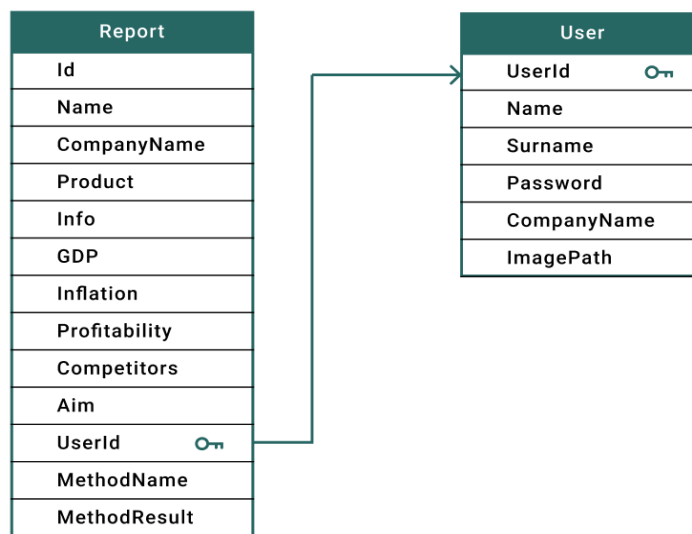


Рисунок 2.9 – Модель бази даних

2.6 Висновки

Отже, в даному розділі було визначено структуру додатку, також вибрано кольорову гаму. Розроблено адаптивний метод прогнозування, метод автоматизованого формування звіту з прогнозування. Розроблено діаграму варіантів використання, такі алгоритми як: формування звіту прогнозування, загальний алгоритм роботи, авторизація. Наведено модель бази даних додатку.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ

3.1 Варіантний аналіз засобів створення програмного продукту

Серед мов високого рівня найпопулярнішими є C++, C#, Java. Кожна з них підтримується. Проведемо аналіз кожної з них.

C++ – це покращена мова C, яка зазвичай використовується для об'єктно-орієнтованого програмування. Код написаний на C++ компілюється напряму в машинний код, що робить дану мову одною з найшвидших в світі. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також ігор [14].

C# – об'єктно орієнтована мова, розроблена компанією Microsoft спеціально для використання можливостей .NET Framework. C# відноситься до мов програмування з C-подібним синтаксисом. Мова має строгу статичну типізацію, підтримує перевантаження операторів, анонімні функції, узагальнені типи і методи, вказівники на члени функції класів [15].

Java – об'єктно-орієнтована мова програмування, одна з найпоширеніших мов програмування в світі. мова програмування, яка виробляє програмне забезпечення для декількох платформ [16].

Порівняння мов програмування наведено в табл.3.1.

Таблиця 3.1 – Порівняння мов програмування

Критерій	C++	C#	Java
Ефективність	+/-	+	+
Швидкодія	-	+	+/-
Незалежність від платформи	-	+	+
Ручне управління пам'яттю	+	+	-

Після проведеного аналізу було обрано мову програмування C#, так як дана мова найкраще підходить для поставленої мети.

3.2 Обґрунтування вибору середовища розробки

Code::Blocks 17.12 – Безкоштовне середовище розробки від команди Code::Blocks дозволить вам писати не лише на C/C++, а й надає підтримку (з деякими обмеженнями) для таких мов програмування, як Fortran і D. Ви можете розширити стандартний набір інструментів, встановивши розширення (див. рис. 1.4). Продукт розроблено для Windows, Mac OS X і Linux, збірку вихідного коду можна використовувати для встановлення середовища на будь-якій Unix-подібній системі [17]. Інтерфейс програмного додатку «Code::Blocks» наведено на рис.3.1.

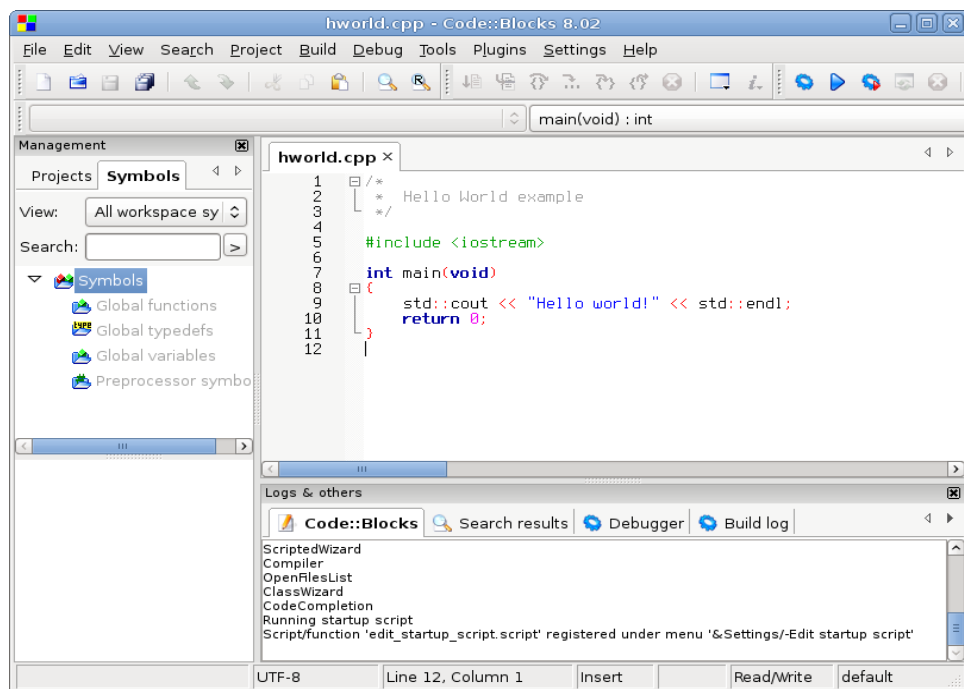


Рисунок 3.1 – Інтерфейс програмного додатку «Code::Blocks»

Eclipse – безкоштовне модульне інтегроване середовище розробки програмного забезпечення. Він розробляється та підтримується Eclipse Foundation і включає такі проекти, як платформа Eclipse, набори інструментів

програміста, системи контролю версій, конструктори GUI тощо [18]. Інтерфейс програмного додатку «Eclipse» наведено на рис.3.2.

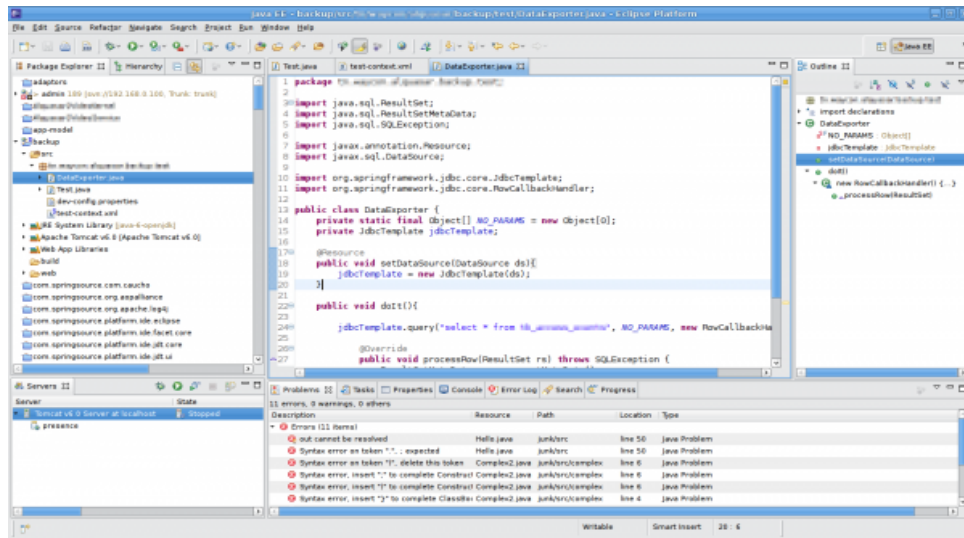


Рисунок 3.2 – Середовище розробки Eclipse

Microsoft Visual Studio – інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів компанії Microsoft (рис.3.3.). Дає можливість розробляти як консольні додатки, ігри та програми з графічним інтерфейсом, в тому числі з підтримкою технології, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ [19].

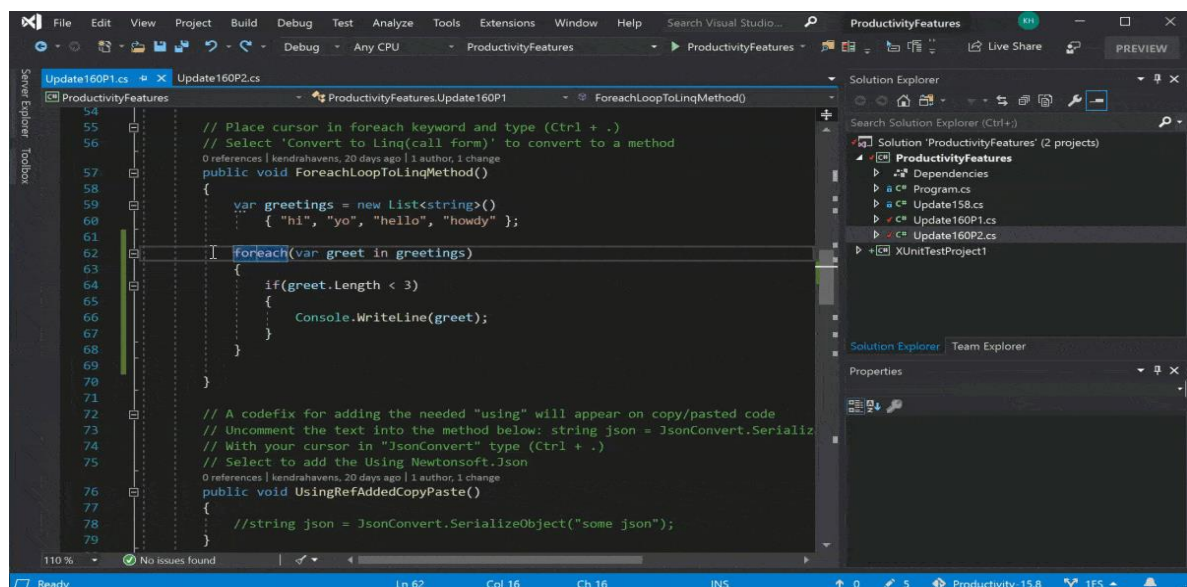


Рисунок 3.3 – Середовище розробки Microsoft Visual Studio

Порівняльний аналіз середовищ розробки наведено у таблиці 3.2.

Таблиця 3.2 – Порівняння технологій розробки веб-сайтів

	Code::Blocks	Eclipse	Microsoft Visual Studio
Простота використання	1	1	1
Вбудована підтримка Git	0	0	1
Ненагромадженість функціями	1	1	1
Швидкість	1	0	1
Розробка GUI (графічні форми)	0	1	1
Сума	3	3	5

Отже, після порівняльного аналізу середовищ розробки було обрано Microsoft Visual Studio, через його зручність, швидкість.

3.3 Розробка бази даних

Для збереження звітів та даних користувача потрібна база даних. У проекті використовується MS SQL server та Entity framework.

Entity framework – це платформа ORM із відкритим вихідним кодом, що підтримується Microsoft для програм .NET. Це дозволяє розробникам працювати з даними за допомогою об'єктів класу домену, не турбуючись про базові таблиці та стовпці бази даних, які зберігають ці дані. За допомогою Entity Framework розробники можуть працювати на вищому рівні абстракції під час роботи з даними та використовувати менше коду для створення та підтримки програм, орієнтованих на дані, ніж традиційні програми.

Офіційне визначення: «Entity Framework — це об'єктно-реляційний маппер (O/RM), який дозволяє розробникам .NET взаємодіяти з базами даних за допомогою об'єктів .NET. Це усуває потребу у більшості коду доступу до даних, який розробникам зазвичай потрібно писати» [20].

MS SQL server – система керування базами даних, розроблена корпорацією Microsoft. Як сервер даних він виконує основну функцію зберігання та обслуговування даних у відповідь на запити від інших програм, які можуть працювати на тому самому сервері або в мережі. Мова, яку він використовує для запитів, — Transact-SQL, створений спільно Microsoft і Sybase. Transact-SQL — це реалізація стандарту ANSI/ISO із розширеною структурованою мовою запитів SQL. Він підходить як для малих і середніх баз даних, так і для великих корпоративних баз даних. Протягом багатьох років вона успішно конкурувала з іншими системами управління базами даних [21].

Entity framework автоматично створив файл міграцій, запустивши яку була створена база даних з таблицями “Reports” (рис.3.4) та “Users” (рис.3.5).

```

public override void Up()
{
    CreateTable(
        "dbo.Reports",
        c => new
        {
            Id = c.Int(nullable: false, identity: true),
            Name = c.String(),
            CompanyName = c.String(),
            Product = c.String(),
            Info = c.String(),
            GDP = c.Double(nullable: false),
            Inflation = c.Double(nullable: false),
            Profitability = c.Int(nullable: false),
            Competitors = c.String(),
            Aim = c.String(),
            UserId = c.Int(nullable: false),
        })
        .PrimaryKey(t => t.Id)
        .ForeignKey("dbo.Users", t => t.UserId, cascadeDelete: true)
        .Index(t => t.UserId);
}

```

Рисунок 3.4 – Скрипт створення таблиці “Reports”

```

--
.PrimaryKey(t => t.Id)
.ForeignKey("dbo.Users", t => t.UserId, cascadeDelete: true)
.Index(t => t.UserId);

CreateTable(
    "dbo.Users",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        Name = c.String(),
        Surname = c.String(),
        Login = c.String(),
        Password = c.String(),
    })
    .PrimaryKey(t => t.Id);

```

Рисунок 3.5 – Скрипт створення таблиці “Users”

Головною одиницею роботи з Entity Framework є “context” за допомогою якого відбувається зв’язок програми з MS SQLServer.

На рисунку 3.6 зображено контекст сутності “Reports”. Поля відповідають колонкам в базі даних. Entity Framework дозволяє встановлювати певні конфігурації для полів. Тут є конфігурації для поля “Name”, це поле відноситься до назви звітів.

Конфігурація [StringLength(40)] – встановлює максимальну кількість символів у рядку, тобто назва звіту не може перевищувати 40 знаків.

Конфігурація [Index(IsUnique = true)] – означає, що поле “Name” має бути унікальним.

Конфігурація [Required] – зазначає, що поле “Name” повинно бути обов’язково заповненим.

```

namespace SalesForecastApp.Entities
{
    Ссылка 16
    public class Report
    {
        Ссылка 7
        public int Id { get; set; }

        [StringLength(40)]
        [Index(IsUnique = true)]
        [Required]
        Ссылка 8
        public string Name { get; set; }
        Ссылка 3
        public string CompanyName { get; set; }
        Ссылка 3
        public string Product { get; set; }
        Ссылка 3
        public string Info { get; set; }
        Ссылка 3
        public double GDP { get; set; }

        Ссылка 3
        public double Inflation { get; set; }
        Ссылка 3
        public int Profitability { get; set; }
        Ссылка 3
        public string Competitors { get; set; }
        Ссылка 3
        public string Aim { get; set; }
        Ссылка 0
        public User User { get; set; }
        Ссылка 2
        public int UserId { get; set; }
        Ссылка 7
        public string MethodName { get; set; }
        Ссылка 7
        public string MethodResult { get; set; }
    }
}

```

Рисунок 3.6 – Конфігурація контексту “Reports”

На рисунку 3.7 зображено контекст сутності “Users”. Поля відповідають колонкам в базі даних.

В кінці є поле: `public virtual ICollection<Report> Reports { get; set; }`, воно потрібно для зв’язку звітів і юзерів.

```

namespace SalesForecastApp.Enteties
{
    Ссылка 4
    public class User
    {
        Ссылка 3
        public int Id { get; set; }
        Ссылка 3
        public string Name { get; set; }
        Ссылка 0
        public string Surname { get; set; }
        Ссылка 5
        public string Login { get; set; }
        Ссылка 5
        public string Password { get; set; }
        Ссылка 1
        public string CompanyName { get; set; }
        Ссылка 2
        public string ImagePath { get; set; }
        Ссылка 0
        public virtual ICollection<Report> Reports { get; set; }
    }
}

```

Рисунок 3.7 – Конфігурація контексту “Users”

Отже, у підрозділі 3.3 було описано створення бази даних програми.

3.4 Програмна реалізація додатку для прогнозування продажів

Під час програмної реалізація проекту було розроблено велику кількість алгоритмів, серед яких:

- алгоритм авторизації та реєстрації;
- алгоритм збереження та заповнення звіту;
- алгоритм підбору методу прогнозування;
- алгоритм розрахунку методом «Арифметичне згладжування»;
- алгоритм розрахунку методом «Експоненційне згладжування»;
- алгоритм розрахунку методом Хольта-Вінтерса;
- алгоритм розрахунку методом «Правило 1.5»;

- алгоритм розрахунку методом «Сезонного прогнозування за допомогою коефіцієнта сезонності».

Спочатку було реалізовано реєстрацію. Для цього створена форма “Authorization”. На форму є два елемента типу `textBox` для введення логіну та пароля, а також кнопка підтвердження та кнопка для реєстрації.

Перевірка наявності акаунту у базі даних відбувається при натисканні кнопки «Підтвердити» (рис. 3.8).

```

Ссылка 1
private async void AuthorizationConfirmBtn_Click(object sender, EventArgs e)
{
    using (var context = new SalesAppContext())
    {
        var user = await Task.Run(async () => await context.User
            .FirstOrDefaultAsync(s => s.Login.Equals(Authorization_Login.Text) &&
                s.Password.Equals(Authorization_Password.Text)));

        if (user != null)
        {
            mainScreen mainScreen = new mainScreen();

            currentUser.SetupData(user.Id, user.Name, "company", user.Password, user.Login, user.ImagePath);
            mainScreen.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Невірні дані для входу");
        }
    }
}

```

Рисунок 3.8 – Програмний код авторизації у програмі

Відбувається перевірка наявності логіна користувача в базі даних, після чого звіряється чи правильно був введений пароль відповідного користувача. Якщо усі умови виконано, тоді ми переходимо на головний екран. В іншому випадку з’являється повідомлення – «Невірні дані для входу».

У випадку, коли користувач ще не створив акаунт, він може зареєструватися. Код реєстрації наведено на рисунку 3.9.

У базі даних створюється новий користувач. Полям у таблиці логін, пароль та ім’я присвоюються значення відповідних текст боксів. Після чого відбувається перевірка, чи не існує вже користувач з таким логіном, якщо існує – виводиться повідомлення у `MessageBox`, про те що користувач вже існує. У іншому випадку дані зберігаються та створюється новий акаунт.

```

private void RegistrConfirmBtn_Click(object sender, EventArgs e)
{
    User newUser = new User
    {
        Login = RegistrLoginTextBox.Text,
        Password = RegistrPasswordTextBox.Text,
        Name = RegistrNameTextBox.Text
    };

    using (var context = new SalesAppContext())
    {
        var existedUser = context.User.FirstOrDefault(s => s.Login.Equals(newUser.Login));

        if (existedUser != null)
        {
            MessageBox.Show("Користувач вже існує");
            return;
        }
        context.User.Add(newUser);
        context.SaveChanges();
    }

    this.DialogResult = DialogResult.OK;
    this.Close();
}

```

Рисунок 3.9 – Програмний код реєстрації у програмі

Звіти містять в собі інформацію по кожному кроку методолгії прогнозування. Основні кроки розміщені на формі “ForecastStart”. Коли користувач потрапляє на форму він заповнює усі зазначені на ній поля. А саме: назва компанії, назва продукту, інформація про компанію, для розрахунку рентабельності вказує суму прибутку та вартість продукції, зазначає мету та конкурентів. Також користувач може ознайомитися з деякими макроекономічними показниками ринку – інфляція та ввп, ці дані актуальні і постійно оновлюються.

Текст елементів textbox присвоюється відповідним колонкам таблиці “Reports”. Після чого запускається конструкція мови C# (рис.3.10). Основною задачею цієї конструкції є автоматичне звільнення ресурсів, які використовувались об’єктом SalesAppContext, що дозволяє не перейматись про звільнені ресурси в подальшому.

context – абстракція, яка дозволяє працювати з сервером бази даних.

context.Reports.Add – додає створений звіт в цю абстракцію і позначає його, як об’єкт, який потрібно створити в базі даних.

await context.SaveChangesAsync() – вносить зміни до бази даних.

```

using (var context = new SalesAppContext())
{
    context.Reports.Add(report);
    await context.SaveChangesAsync();

    currentUser.CurrentReportId = report.Id;
}

```

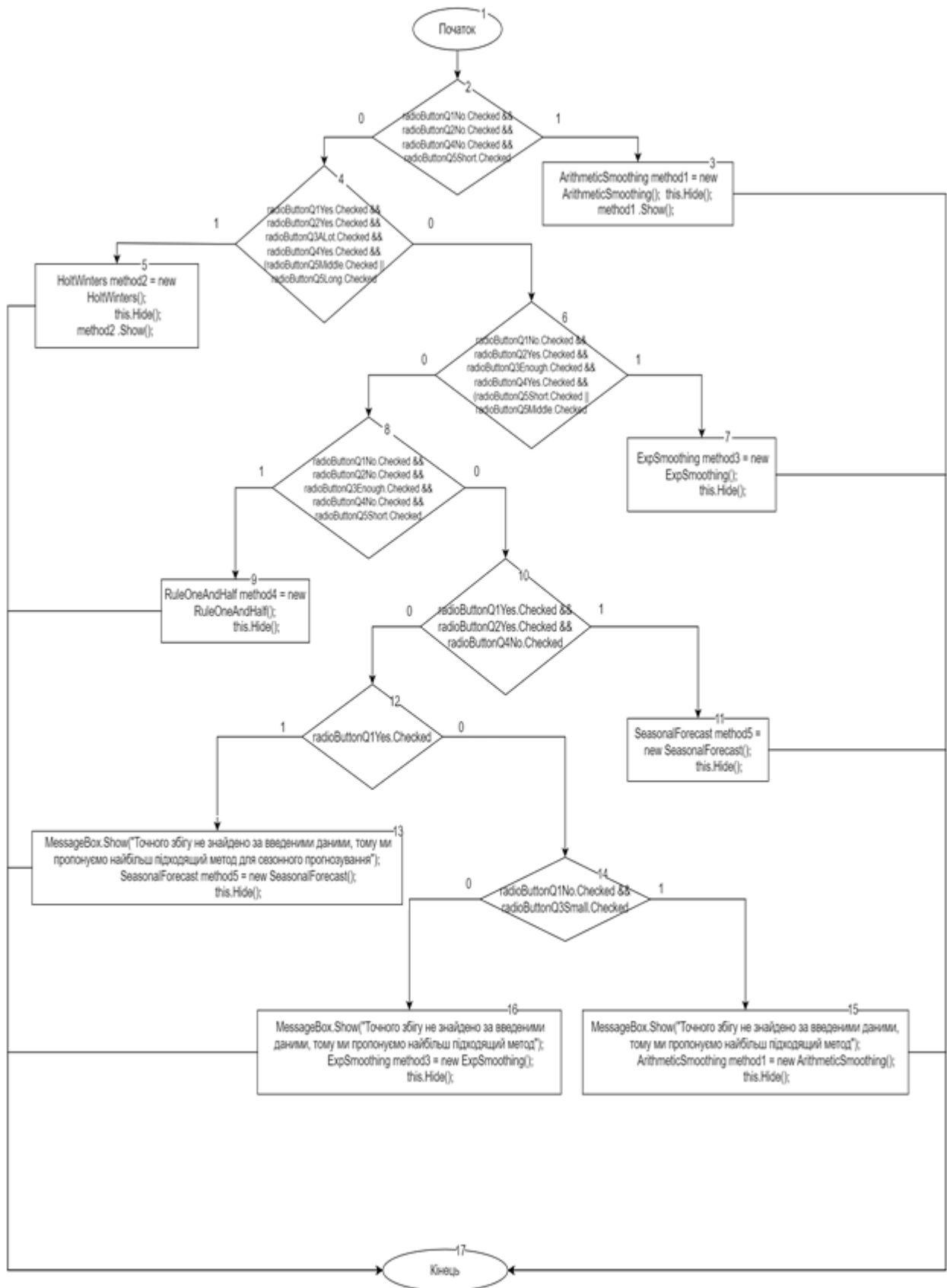
Рисунок 3.10 – Програмний код збереження даних звіту в БД

Далі відбувається наступний крок методології – Вибір методу прогнозування. Був створений алгоритм підбору методу. Користувачу на формі представлені 5 запитань та варіанти відповідей. Він повинен відповісти на кожне запитання активувавши елементи `radioButton`. Після чого натиснути кнопку «Далі». Алгоритм підбору методу наведено на рисунку 3.11.

Опираючись на характеристики кожного методу прогнозування, обирається метод. Алгоритм перебирає відповіді порівнює значення відповідей з тими, що притаманні певному методу та виводить форму методу з яким найбільше було знайдено співпадінь. Якщо вказано комбінацію, яка незадовільняє жоден метод, тоді опираючись на відповіді на основні питання (сезонність і кількість періодів) метод підбирає варіант, який частково покриє потреби користувача і зробить найбільш точний прогноз. В такому випадку з'явиться повідомлення – «Точного збігу не знайдено, пропонуємо найбільш підходящий метод».

Після того як програма вибрала метод прогнозування, з'являється вікно для проведення розрахунків. Всього на даний момент в програмі реалізовано 5 методів прогнозування продажів.

Також у користувача є можливість самостійно обрати метод прогнозування продажів товарів. Для цього він обирає назву методу з випадючого списку, ігноруючи запитання. Алгоритм виконуватись не буде, користувач одразу потрапить на вікно відповідного методу прогнозування.



Рисунк 3.11 – Алгоритм підбору методу прогнозування

Метод арифметичне згладжування. Код методу наведено на рисунку 3.12.

```

Ссылка: 1
private void calculate_Click(object sender, EventArgs e)
{
    var a = textBox1.Text.Split('\n').Select(x =>
    {
        double result;
        var success = double.TryParse(x, out result);
        return new { a = success, b = result };
    })
    .Where(s => s.a)
    .Select(s => s.b).ToArray();

    ArismaticResult.Text = (a.Sum() / a.Length).ToString();
}

```

Рисунок 3.12 – Фрагмент коду методу «Арифметичне згладжування»

Користувач вводить дані на формі у `textBox`, кожне нове число з нового рядка. Потім за допомогою методу `Split`, ми розділяємо дані і формуємо масив елементів з типом даних `double`. Після чого записуємо результат формули у елемент `Label` з назвою “`ArismaticResult`”.

Метод «Правило 1.5». Код методу наведено на рисунку 3.13.

```

Ссылка: 1
private void Result_Click(object sender, EventArgs e)
{
    var zpp = Convert.ToDouble(Zpp.Text);
    var pst = Convert.ToDouble(Pst.Text);
    var zkp = Convert.ToDouble(Zkp.Text);

    ArismaticResult.Text = ((zpp+pst-zkp) * 1.5 -zkp).ToString();
}

```

Рисунок 3.13 – Фрагмент коду методу «Правило 1.5»

Користувач у вікні даного методу заповнює 3 різні `textBox`, кожен з яких відповідає параметрам: залишок на початку звітнього періоду, постачання у звітньому періоді, залишок на кінець звітнього періоду. Ці дані ми конвертуємо у `double` і записуємо у змінні. Після чого підставляємо дані у формулу, результат якої записуємо у `label` для виводу на екран.

Метод «Експоненційне згладжування». Код методу наведено на рисунку 3.14.

```

string exponential_smoothing(double[] series, double alpha)
{
    result1.Add(series[0]);

    for (int i = 1; i < series.Length; i++)
    {
        result1.Add( (alpha * series[i] + (1 - alpha) * series[i - 1]));
    }
    void fillChart(List<double> c)
    {
        double[] newarr = c.ToArray();
        for (int i = 0; i < newarr.Length; i++)
        {
            chart1.Series["ExponentialSmoothing"].Points.AddXY((i + 1), newarr[i]);
        }
    }

    fillChart(result1);
    string result = string.Join("\n", result1.Select(x => x.ToString()));

    return result;
}

```

Рисунок 3.14 – Фрагмент коду методу «Експоненційне згладжування»

Створюємо функцію для розрахунку прогнозу за даним методом. Функція приймає масив елементів типу `double` (це дані попередніх періодів) та ще один елемент типу `double` (коефіцієнт альфа для згладжування). Спочатку функція перебирає масив і кожне число підставляє у формулу для згладжування. Далі записує результат у новий масив. Цей масив передається методу “`fillChart`” для відображення графіку. Також ми конвертуємо цей масив у масив типу `string`, цей масив і повертає функція як результат.

Користувач вводить дані попередніх періодів в `textbox`, кожне нове число з нового рядка. Ми конвертуємо ці значення у масив елементів типу `double`. При натисканні кнопки «Розрахувати» ми передаємо отриманий масив у розроблену раніше функцію і отримуємо результат, який записуємо у `label`.

Метод «Прогнозування сезонного попиту». Код методу наведено на рисунку 3.15.

З форми ми отримуємо дані. Конвертуємо їх у тип даних `double`. Передаємо ці значення змінним, які потім використовуються у формулі. Отримане значення записуємо у `label`.

```

private void result_Click(object sender, EventArgs e)
{
    var a = textBox1.Text.Split('\n').Select(x =>
    {
        double result;
        var success = double.TryParse(x, out result);
        return new { a = success, b = result };
    })
    .Where(s => s.a)
    .Select(s => s.b).ToArray().Sum();

    var d = textBox3.Text.Split('\n').Select(x =>
    {
        double result;
        var success = double.TryParse(x, out result);
        return new { d = success, c = result };
    })
    .Where(s => s.d)
    .Select(s => s.c).ToArray().Sum();
    var g = Convert.ToDouble(textBox2.Text);
    result1.Text = (g+(g*(d-a)/a)).ToString()+"%";
}

```

Рисунок 3.15 – Фрагмент коду методу «Прогнозування сезонного попиту»

Метод «Хольта-Вінтерса». Код методу наведено на рисунку 3.16.

```

<small>Ссылка 1</small>
private static double[] calculateHoltWinters(int[] y, double a0, double b0, double alpha, double beta, double gamma, double[] initialSeasonalIndices, int period, int m)
{
    double[] St = new double[y.Length];
    double[] Bt = new double[y.Length];
    double[] It = new double[y.Length];
    double[] Ft = new double[y.Length + m];

    St[1] = a0;
    Bt[1] = b0;
    for (int i = 0; i < period; i++)
    {
        It[i] = initialSeasonalIndices[i];
    }
    Ft[m] = (St[0] + (m * Bt[0])) * It[0];
    Ft[m + 1] = (St[1] + (m * Bt[1])) * It[1];
    for (int i = 2; i < y.Length; i++)
    {
        if ((i - period) >= 0)
        {
            St[i] = alpha * y[i] / It[i - period] + (1.0 - alpha) * (St[i - 1] + Bt[i - 1]);
        }
        else
        {
            St[i] = alpha * y[i] + (1.0 - alpha) * (St[i - 1] + Bt[i - 1]);
        }

        Bt[i] = gamma * (St[i] - St[i - 1]) + (1 - gamma) * Bt[i - 1];

        if ((i - period) >= 0)
        {
            It[i] = beta * y[i] / St[i] + (1.0 - beta) * It[i - period];
        }

        if (((i + m) >= period))
        {
            Ft[i + m] = (St[i] + (m * Bt[i])) * It[i - period + m];
        }
    }
    return Ft;
}

```

Рисунок 3.16 – Фрагмент коду методу «Хольта-Вінтерса»

Створено клас HoltWintersTripleExponentialImpl, знабором функцій для розрахунку за методом Хольта-Вінтерса.

В класі створено функцію “calculateHoltWinters”, яка розраховує продажі на майбутній період за даним методом. Функція повертає масив елементів типу double. В дану функцію потрібно передати масив елементів типу int (продажі за попередні періоди), далі 3 коефіцієнти згладжування (рівень, тренд, сезонність), масив елементів типу double для сезонних індикаторів. кількість періодів та число майбутніх точок прогнозу.

Спочатку ми створюємо масиви для сезонних індикаторів, які потім будемо заповнювати значеннями отриманих за допомогою формул. Ці значення розраховуються в окремій функції – «calculateSeasonalIndices».

Користувач заповнює дані минулих продажів у textbox, а також може сам зазначити коефіцієнти згладжування альфа, бета і гама. Краще всього їх обрати можна методом перебору. Тому слід спробувати кілька варіантів значень цих коефіцієнтів.

Отримані дані ми конвертуємо у масив елементів типу даних int, а коефіцієнти конвертуємо у double. Підставляємо дані у функцію (рис.3.17). Отриманий результат конвертуємо у масив типу string та записуємо в label.

```
private void Calculate_Click(object sender, EventArgs e)
{
    // var Arr = new int[];
    var a = textBox1.Text.Split('\n').Select(x =>
    {
        int result;
        var success = int.TryParse(x, out result);
        return new { a = success, b = result };
    })
    .Where(s => s.a)
    .Select(s => s.b).ToArray();

    int p= (a.Length/2);
    int mm = p;

    double alpha = double.Parse(alphatext.Text);

    double beta = double.Parse(betatext.Text);

    double gamma = double.Parse(gammatext.Text);

    var c = HoltWintersTripleExponentialImpl.forecast(a, alpha, beta, gamma, p, mm).Where(x => x > 0).ToArray();
    var ch = c.Select(x => x.ToString());
    string k = String.Join("\r", ch);

    Result.Text = k;

    void fillChart()
    {
        for (int i = 0; i < c.Length; i++) {
            chart1.Series["HoltWinters"].Points.AddXY((i+1), c[i]);
        }
    }
    fillChart();
}
```

Рисунок 3.17 – Ініціалізація методу «Прогнозування сезонного попиту»

Запускаємо метод “fillChart” і передаємо туди результат. Даний метод побудує графік.

Коли користувач розрахував прогноз методом, який був йому підібраний, він натискає кнопку «Далі». При натисканні цієї кнопки отриманий прогноз записується у базу даних в таблицю “Reports”.

Після чого запускається форма “Result”. У richTextBox ми передаємо дані щойно створеного звіту з бази даних (рис.3.18).

```
private void Result_Load(object sender, EventArgs e)
{
    textBox1.Text = $"{_report.Name}";

    richTextBox1.Text = $"{_report.Name}\n\n" +
        $"Дата: {DateTime.Now.ToShortDateString()}\n\n" +
        $"Назва компанії:\n" + $"{_report.CompanyName}\n\n" +
        $"Тип продукції: \n {_report.Product}\n\n" +
        $"Основна інформація: \n {_report.Info}\n\n" +
        $"Інфляція: {_report.Inflation}\n\n" +
        $"ВВП: {_report.GDP}\n\n" +
        $"Рентабельність: {_report.Profitability}\n\n" +
        $"Мета: \n {_report.Aim}\n\n" +
        $"Конкуренти: \n {_report.Competitors}\n\n"+
        $"Прогноз методом {_report.MethodName}: \n {_report.MethodResult}\n";
}
```

Рисунок 3.18– Фрагмент коду відображення звіту.

Текст звіту можна зберегти на локальний диск або надрукувати, натиснувши відповідні кнопки «Зберегти» та «Друк».

3.5 Висновки

В даному розділі було проведено аналіз мов програмування C++, C#, та Java, середовищ розробки CodeBlocks, Eclipse та Visual Studio. В результаті аналізу було прийнято рішення використовувати середовище розробки Visual Studio, мову C#. Також було описано розроблену базу даних та розроблені основні програмні модулі програмного засобу для прогнозування продажів.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Аналіз методів тестування програмного забезпечення

Для того що перевірити працездатність програмного забезпечення потрібно провести тестування. Існує три основних методи проведення тестувань: чорна скринька, біла скринька та сіра скринька.

Техніка тестування «чорна скринька» не передбачає перевірки архітектури системи та вихідного коду, натомість проводиться поверхнєве тестування функціоналу. Тобто тестувальник взаємодіє виключно з інтерфейсом користувача системи, надаючи вхідні дані та перевіряючи вихідні.

Перевагами цього методу є те що можна швидко протестувати програмне забезпечення, навіть у випадку коли програма дуже велика та має складну архітектуру. Не потрібен доступ до коду та провести тестування може навіть той хто не розуміється на програмуванні.

Недоліками є те що не всі помилки можуть бути виявлені, обмежена кількість тест-кейсів. У випадку коли сталася помилка важко відстежити, що саме її спричинило.

Тестування «білої скриньки» – детальне дослідження внутрішньої логіки та структури коду. Для того, щоб виконати тестування програми за цим принципом, тестувальник повинен знати внутрішню роботу коду. Тестер повинен зазирнути всередину вихідного коду та з'ясувати, яка частина/фрагмент коду поводить ся неналежним чином.

Перевагами є те, що під час тестування можна оптимізувати код, окрім наслідків розробник одразу отримує причину і може швидко виправити помилку. Багато варіантів перевірки.

Недоліки зростає необхідність у кваліфікованому тестувальнику, що збільшує витрати. Через велику кількість зон для тестування, деякі можуть лишитися не поміченими, вимагає спеціальних програмних засобів для аналізу коду.

Техніка «сіра скринька» – для тестування програми з обмеженим знанням внутрішньої роботи програми. На відміну від тестування чорної скриньки, де тестер перевіряє лише інтерфейс користувача програми; у тестуванні сірого ящика тестувальник має доступ до проектних документів і бази даних. Володіючи цими знаннями, тестувальник може підготувати кращі тестові дані та тестові сценарії під час складання плану тестування.

Перевагою є те, що цей метод поєднує в собі особливості тестування чорної та білої скриньки. Тестувальник приділяє увагу інтерфейсу, базі даних та функціональній специфікації. Тестування в результаті буде проведено з точки зору користувача а не розробника.

Недоліком є те що код не перевіряється [22].

Отже в результаті проведеного аналізу, було обрано метод чорної скриньки, для того щоб ефективно протестувати інтерфейс користувач та програму з точки зору користувача.

4.2 Тестування розробленого програмного продукту

Для тестування програмного продукту була обрана методика «чорної скриньки». Основою методики є використання тейст-кейсів. Тест-кейс це артефакт, який описує порядок кроків, умов та параметрів, потрібних для перевірки функціональності програми.

Під тест кейсом мається на увазі наступна структура: дія, очікуваний результат, отриманий результат.

Буде створено декілька тест-кейсів для перевірки правильності роботи основних функцій програмного додатку. Ось, деякі з них:

Тест-кейс №1 – Реєстрація у додатку.

1. Запустити програму.
2. Натиснути кнопку «Реєстрація».
3. Ввести дані.
4. Отримати результат.

Очікуваний результат – на екрані відображається повідомлення «Ваш профіль створено».

Для роботи з програмою користувачу потрібно створити аккаунт. При запуску програми відображається вікно входу, для реєстрації потрібно натиснути однойменну кнопку. Користувач заповнює поля на екрані, а саме: ім'я, логін, пароль (рис. 4.1).

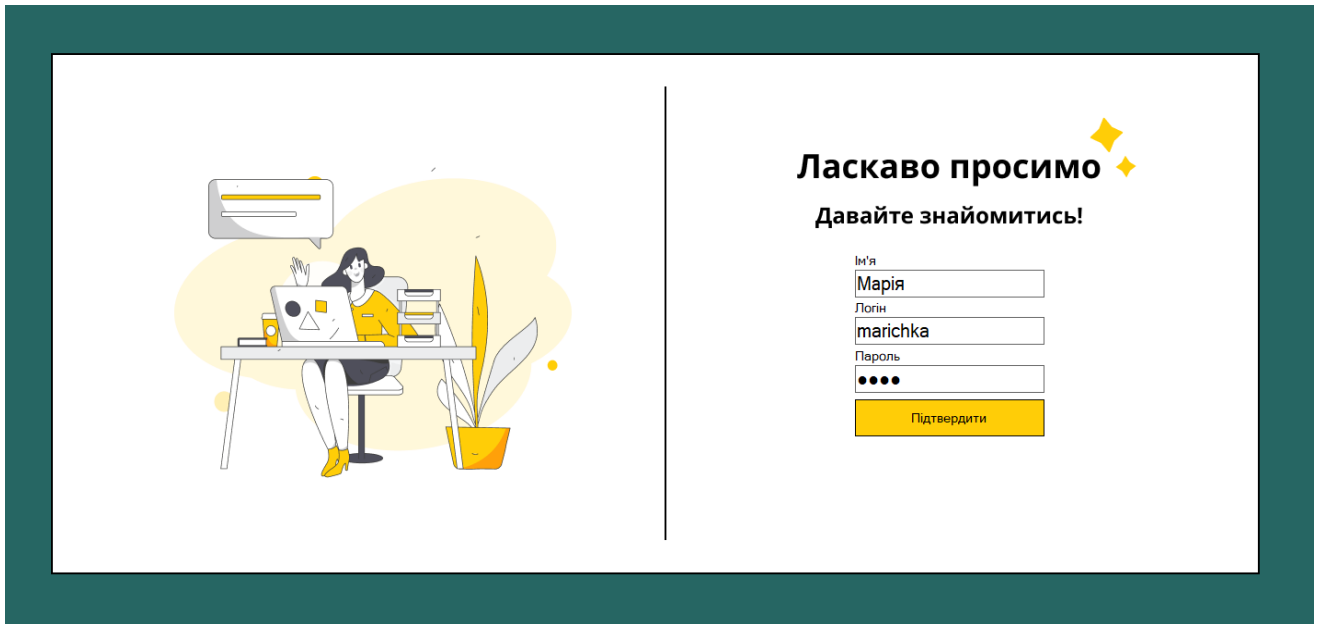


Рисунок 4.1 – Вікно реєстрації

Логін користувача повинен бути унікальним, якщо вже існує аккаунт з таким логіном – з'явиться помилка. Якщо всі поля заповнені, потрібно натиснути кнопку «Підтвердити».

Результат виконання тест-кейсу №1 зображено на рисунку 4.2.

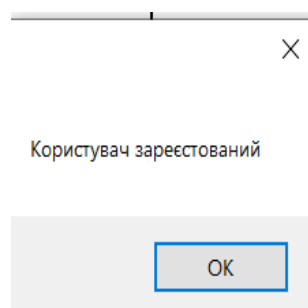


Рисунок 4.2 – Результат виконання тест-кейсу №1

Отриманий результат відповідає очікуваному, отже тест-кейс №1 успішно пройдено.

Тест-кейс №2 – Прогнозування продажів.

1. Натиснути кнопку «Прогнозування»
2. Ввести дані на кожен крок.
3. Натиснути кнопку «Далі».
4. Відповісти на запитання відмітивши підходящі радіо батони.
5. Натиснути кнопку «Далі».
6. Ввести дані в поля.
7. Натиснути кнопку «Розрахувати».
8. Натиснути кнопку «Далі».
9. Отримати звіт.

Очікуваний результат – на екрані відображається вікно зі звітом, який включає у себе всі відповіді користувача на кожному кроці.

Після вдалої авторизації, потрапивши на головний екран користувач натискає кнопку «Прогнозування» (рис. 4.3).

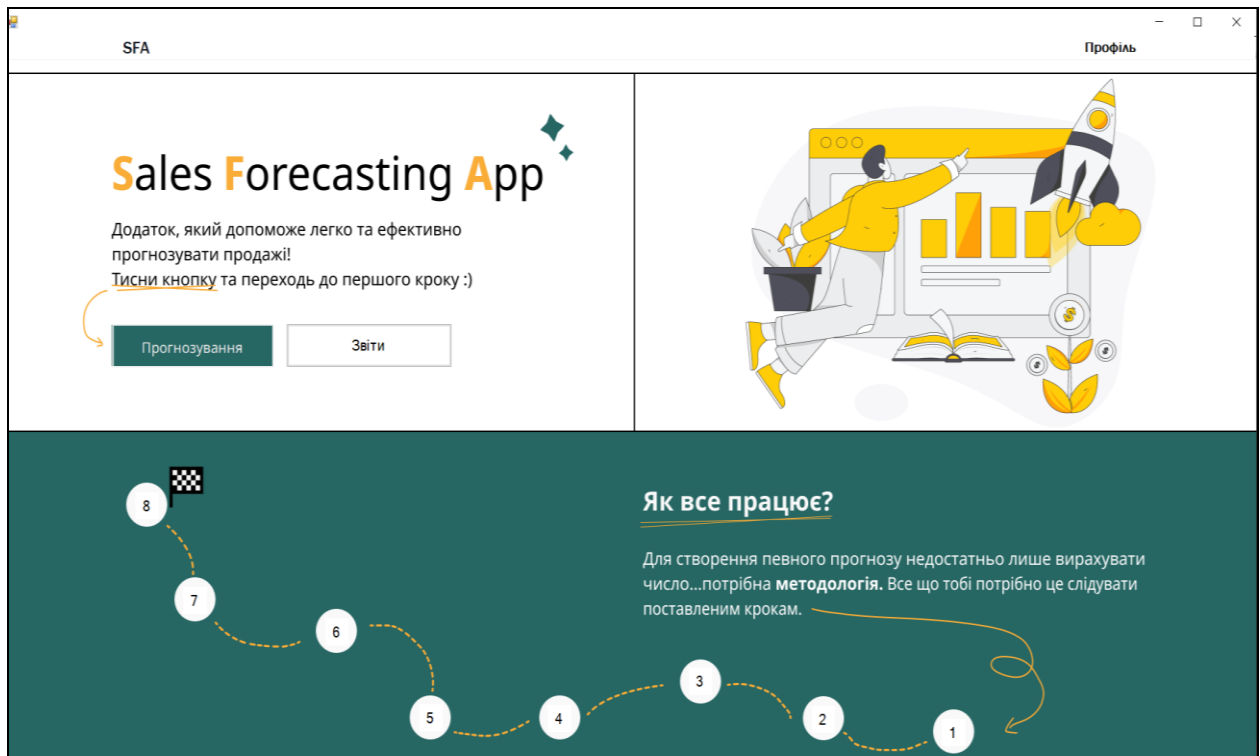


Рисунок 4.3 – Головне вікно

На екрані з'являється вікно з кроками, які користувач повинен заповнити (рис. 4.4). Після чого натиснути кнопку «Далі».

Прогнозування

Крок 1

Назва компанії: MyCompany

Тип продукції: someproduct

Основна інформація: Інформація про продукт і компанію

Крок 2

Ukraine Inflation Rate

October 2022 Data - 1995-2021 Historical - November Forecast

Month	Inflation Rate (%)
Jan 2022	10.3
Feb 2022	10
Mar 2022	10
Apr 2022	10.7
May 2022	13.7
Jun 2022	16.4
Jul 2022	18
Aug 2022	21.5
Sep 2022	22.2
Oct 2022	23.8
Nov 2022	24.6
Dec 2022	26.6

Крок 3

Ukraine GDP

GDP nominal (2017): \$112,154,185,121

Рентабельність: сума прибутку: 576, вартість активів: 678

Крок 4

Мета: збільшити прибуток на 20%

Крок 5

Конкуренти: конкурент1, конкурент2

Назад Далі

Рисунок 4.4 – Вікно з кроками прогнозування

Наступне вікно це вибір методу прогнозування (рис. 4.5).

Вибір методу прогнозування

Дайте відповіді на поставлені запитання і ми підберемо для вас метод ;)

Крок 6: Вибір методу прогнозування

Ви можете обрати метод самостійно

1 Чи залежить продаж товару від сезону? Так Ні

2 Ви давно працюєте з цим товаром? Так Ні

3 Скільки є минулих періодів збуту товару? Багато Не багато Мало

4 Необхідний тренд? Так Ні

5 На який період буде проведено прогнозу Короткий Середній Довгий

Назад Далі

Рисунок 4.5 – Вікно «Вибір методу прогнозування»

Користувач може обрати метод самостійно, якщо він точно знає який метод підходить йому. Для цього потрібно вибрати назву з випадючого списку. Або користувач може відповісти на 5 запитань. Після чого система обере метод самостійно. Коли всі дії виконано – натиснути кнопку «Далі».

З огляду на вибрані відповіді під час тестування, система підбрала метод «Експоненційне згладжування». На екрані з'являється вікно методу, є опис самого методу, інструкції та поля для розрахунку. Ввівши дані за попередні періоди, користувач отримає результат набір згладжених точок майбутніх продажів та побудований по ним графік (рис. 4.6).

Експоненційне згладжування

Опис методу
 Експоненційне згладжування забезпечує наочне уявлення про тренд і дає змогу робити короткострокові прогнози.
 Коефіцієнт згладжування α – це значення від 0 до 1.

Як це працює?

- Вам потрібно ввести у грн. продажі за попередні періоди
- Дані потрібно вводити з нового рядка.
- Потім натиснути кнопку розрахувати.
- Готово!

Поле для введення

123
234
454
344

Розрахувати

Результат:
405,875

Експоненційне згладжування

Період	Продажі
1	123
2	234
3	454
4	344
5	405,875
6	405,875

Далі

Рисунок 4.6– Метод прогнозування «Експоненційне згладжування»

Натиснувши кнопку «Далі», ми потрапляємо на вікно звіту. Результат виконання тест-кейсу №2 зображено на рисунку 4.7.

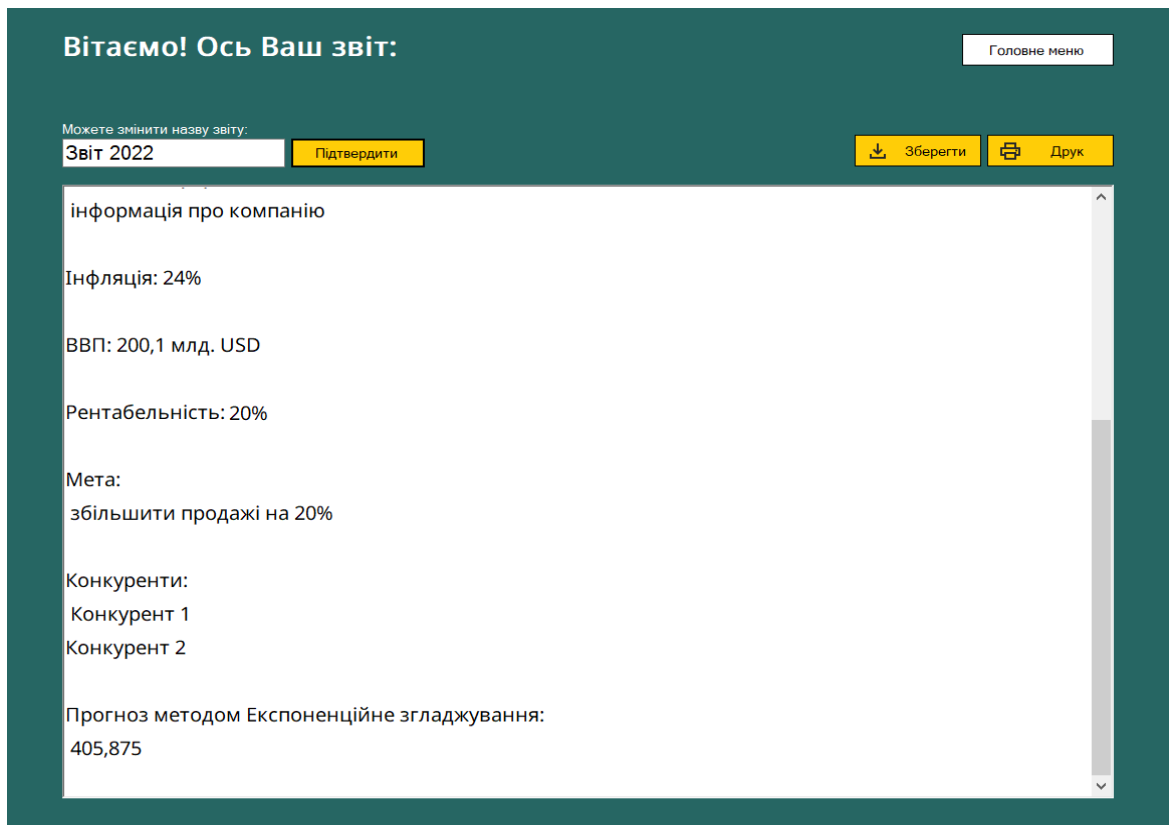


Рисунок 4.7 – Результат виконання тест-кейсу №2

На цьому вікні користувач може зберегти звіт на локальний диск, надрукувати або змінити стандартне ім'я.

Отриманий результат відповідає очікуваному, отже тест-кейс №2 успішно пройдено.

Тест-кейс №3 – Видалення звіту.

1. На головному вікні натиснути кнопку «Звіти»
2. З списку вибрати звіт, натиснути 2 рази.
3. Натиснути кнопку «Видалити».
4. Отримати сповіщення.

Очікуваний результат – на екрані відображається повідомлення «Звіт видалено».

Для перегляду звітів збережених раніше, потрібно на головному екрані натиснути кнопку «Звіти». Перед нами з'являється список звітів, вікно для виводу звіту, кнопки керування (рис.4.8).

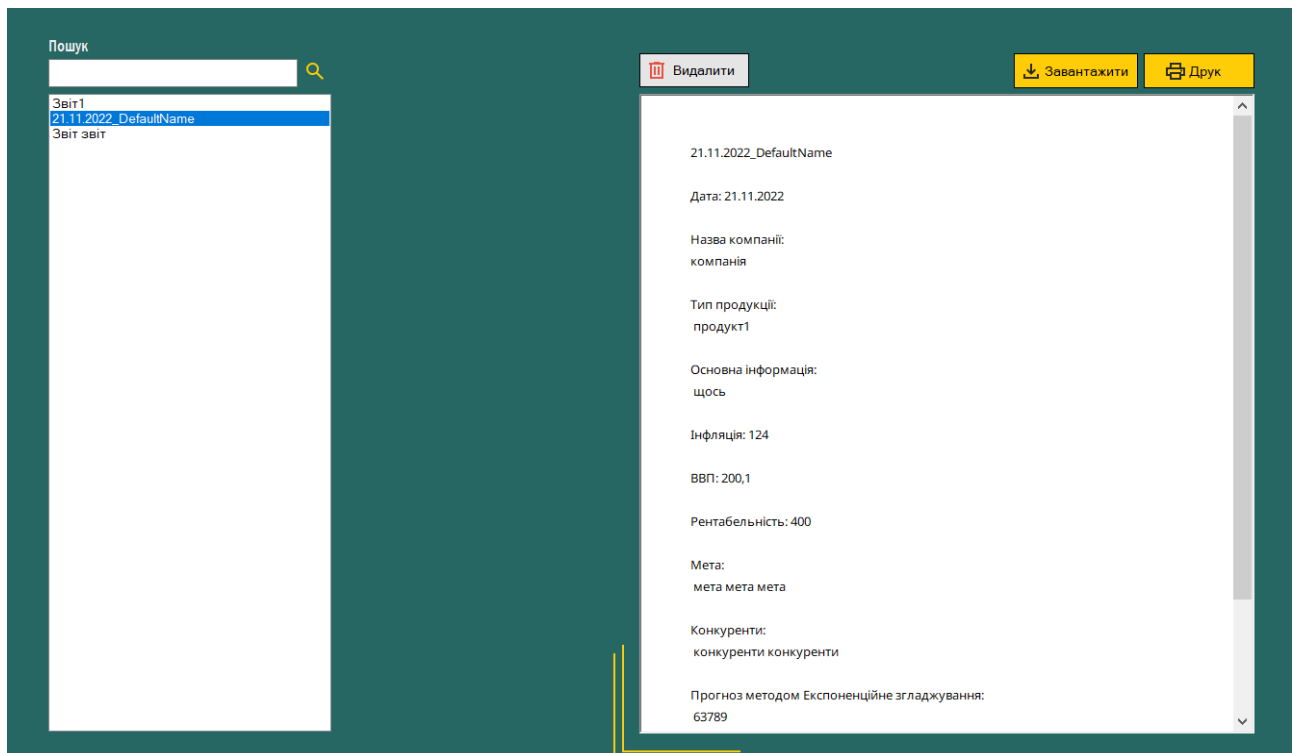


Рисунок 4.8 – Вікно «Звіти»

Обравши непотрібний звіт, натискаємо на нього два рази лівою кнопкою миші. Звіт стає активним, текст звіту відображається на екрані. Натискаємо кнопку «Видалити».

Результат виконання тест-кейсу №3 зображено на рисунку 4.9.

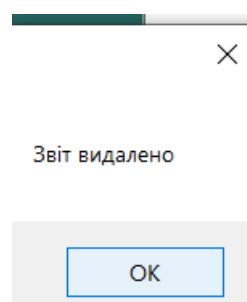


Рисунок 4.9– Результат виконання тест-кейсу №3

Отриманий результат відповідає очікуваному, отже тест-кейс №3 успішно пройдено.

Тест-кейс №4 – Зміна особистих даних користувача у профілі.

5. На головному вікні натиснути кнопку «Профіль»

6. Ввести нові дані в поля, які необхідно змінити.
7. Натиснути кнопку «Зберегти зміни».
8. Отримати сповіщення.

Очікуваний результат – на екрані відображається повідомлення «Зміни збережено».

Відкриваємо профіль користувача (рис.4.10).

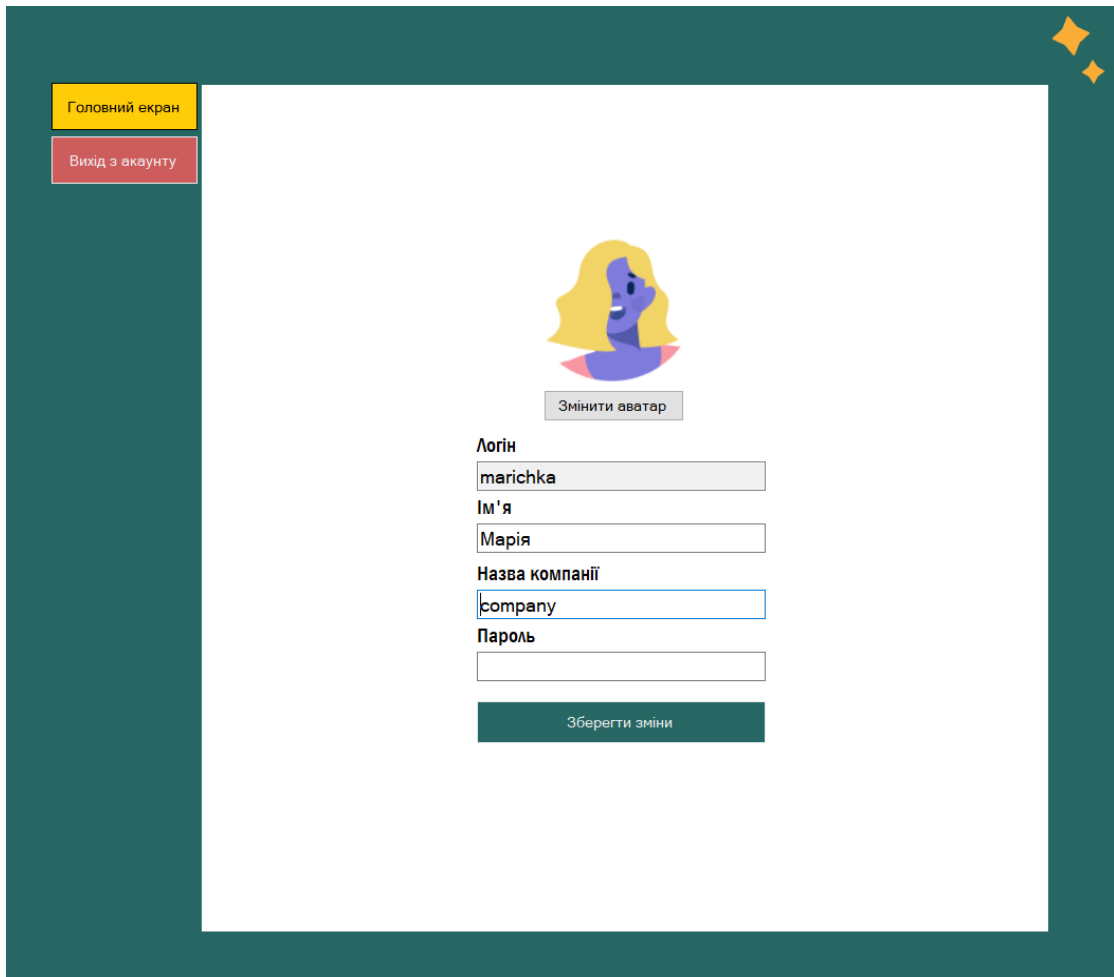


Рисунок 4.10 – Профіль користувача

Поле логін не можна змінити, так як кожен юзер має унікальний логін. Текст бокс логіну заблокований та на відміну від інших він сірого кольору, що вказує на його не активність. Вводимо нове значення в поле «Назва компанії» та змінюємо аватар. Для зміни автару натискаємо кнопку «Змінити аватар». Відкривається діалогове вікно вибору картинки з локального диску. Нові дані зображено на рисунку 4.11.

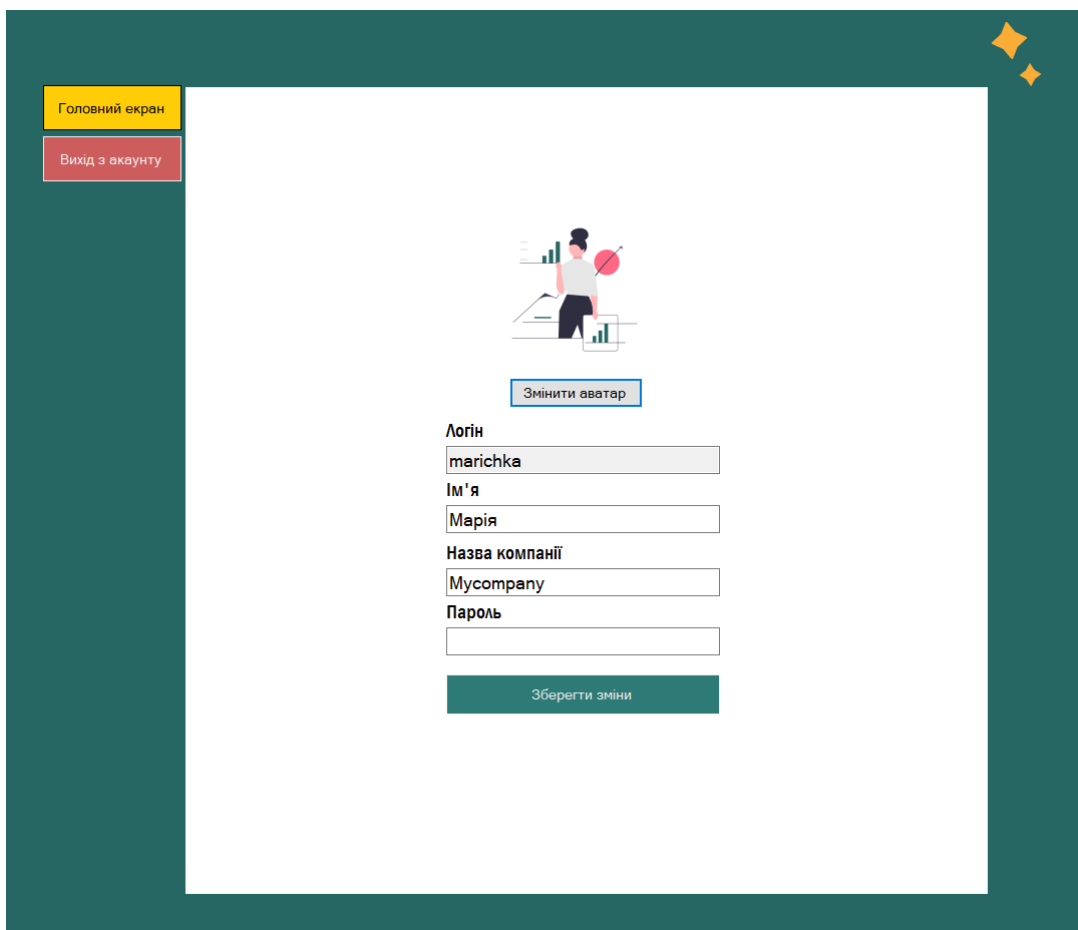


Рисунок 4.11 – Оновлені дані користувача

Натискаємо кнопку «Зберегти зміни». Результат виконання тест-кейсу №3 зображено на рисунку 4.12.

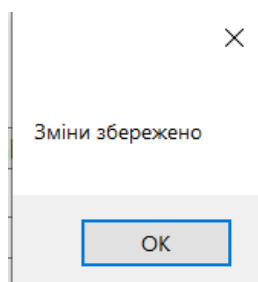


Рисунок 4.12 – Результат виконання тест-кейсу №4

Отриманий результат відповідає очікуваному, отже тест-кейс №3 успішно пройдено.

Отже, в результаті проведеного тестування було доведено повну працездатність додатку.

4.3 Вимоги до персонального комп'ютера та інструкція користувача

Для роботи з додатком потрібно мати мінімальну конфігурацію персонального комп'ютера. Додаток підтримує операційну систему Windows.

Мінімальні характеристики комп'ютера наведено в таблиці 4.1.

Таблиця 4.1 – Мінімальна конфігурація

Тип процесора	32-розрядний (x86) або <u>64-розрядний (x64)</u> процесор з тактовою частотою 2 ГГц
Об'єм оперативної пам'яті	512 Мб
Відеопам'ять	64 Мб
Вінчестер	350 Мб
Операційна система	Windows 7,8,10

Для встановлення програми потрібно завантажити архів з файлами для інсталяції та розархівувати його. Відкрити папку та натиснути на файл інсталяції з назвою "SetupSFA". З'явиться вікно для встановлення програми на комп'ютер (рис. 4.13), потрібно дотримуватися кроків. В результаті отримаємо програму готову до роботи.

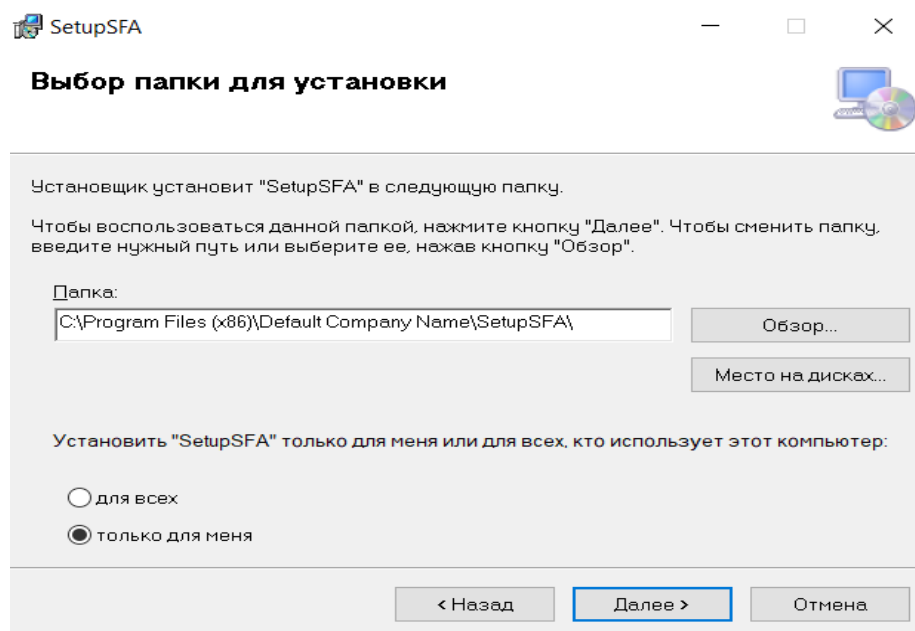


Рисунок 4.13 – Встановлення програми

4.4 Висновки

У четвертому розділі було проведено тестування програми, у результаті якого було доведено його повну працездатність та відповідність поставленому технічному завданню. Визначено мінімальну та рекомендовану конфігурації персональних комп'ютерів для коректної роботи програмного засобу. Розроблено інструкцію користувача.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1. Проведення комерційного аудиту науково-технічної розробки

Результатом магістерської кваліфікаційної роботи «Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів» є програмний продукт, який дозволяє оптимізувати та полегшити процес прогнозування продажів, за рахунок розробленої спрощеної методології прогнозування та автоматичного підбору методу прогнозування для якісного та більш точного розрахунку прогнозу опираючись на дані про певний бізнес. Результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Перед реалізацією програми потрібно оцінити її науково технічний рівень та рівень комерційного потенціалу розробки програмного продукту.

Для проведення технологічного аудиту залучено трьох незалежних експертів: к.т.н. доц. кафедри ІІІ Майданюк Володимир Павлович, к.т.н. доц. кафедри ІІІ Ракитянська Ганна Борисівна, к.т.н. доц. кафедри ІІІ Романюк Оксана Володимирівна.

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 5.1 [23].

Таблиця 5.1 – Критерії оцінювання комерційного потенціалу розробки
бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження таблиці 5.1

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промислового комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 5.2.

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Майданюк	2 – Ракитянська	3 – Романюк
	Бали, виставлені експертами:		
1	3	3	4
Ринкові переваги (недоліки):			
2	1	2	2
3	4	4	4
4	3	4	4
5	4	3	3
Ринкові перспективи			
6	2	4	3
7	3	3	3

Практична здійсненність			
8	4	3	4
9	4	4	4
10	4	4	4
11	4	4	3
12	3	2	3
Сума балів	СБ ₁ =39	СБ ₂ =40	СБ ₃ =41
Середньоарифметична сума балів $\overline{СБ}$	40		

За даними таблиці 5.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 5.3.

Таблиця 5.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 40 бали, що відповідає рівню «вище середнього».

Проаналізуємо суть технічної проблеми та коротко розглянемо аналоги.

Anaplan – це американська компанія з планування програмного забезпечення зі штаб-квартирою в Сан-Франциско, штат Каліфорнія. Anaplan продає підписку на хмарне програмне забезпечення для бізнес-планування та надає дані для прийняття рішень. Перевагою Anaplan є те що цей програмний продукт підходить для компаній, які шукають стратегію планування,

прогнозування та складання бюджету, планування робочої сили та відстеження даних конкурентів.

Недоліки:

1. Початкове налаштування вимагає багато навчання та часу. Їх тренінг триває кілька днів, і все ще вимагає додаткової подальшої роботи та спілкування з командою з впровадження.
2. Складний і нагромаджений інтерфейс.
3. Мало графіків та методів прогнозування [6].

Розроблений програмний продукт не містить недоліків аналога, дозволяє прискорити процес реєстрації використовуючи логін та пароль.

У таблиці 5.4 наведені основні технічні показники аналога і нового програмного продукту

Таблиця 5.4 – Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка
Зручний інтрефейс	0	1
Безкоштовний пробний період	1	1
Легке навчання для використання	0	1
Можливість прогнозування продажу	1	1
Адаптивний підбір методу прогнозування	0	1

Розроблений додаток а також його методи роботи будуть використовуватись у сфері бізнесу та торгівлі. Додаток забезпечує можливість прогнозування продажів товарів.

Використовувати додаток можуть люди, які займаються продажем, плануванням бюджету компанії і бажають спрогнозувати продажі продукції.

Розробка програмного продукту знаходиться на останньому етапі розробки.

5.2 Розрахунок витрат на здійснення науково-технічної розробки

Проведемо розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи.

Обчислимо основну заробітну Z_o , за формулою:

$$Z_o = \frac{M}{T_p} \cdot t [\text{грн}], \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника, грн;

T_p – число робочих днів в місяці - 22;

t – число днів роботи розробника - 66.

Обчислимо заробітну плату розробника з місячним окладом 13000 грн і кількістю робочих днів у місяці – 22. Число днів роботи розробника 66.

$$Z_o = \frac{13000}{22} \cdot 66 = 39000,00 [\text{грн}].$$

Результати розрахунків зведемо до таблиці 5.5.

Таблиця 5.5 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Розробник	12000,00	545,45	66	36000,00
Керівник роботи	15000,00	681,81	3	2045,40
Дизайнер	5800	263,63	5	1318,15
Всього				39 363,55

Обчислимо додаткову заробітну плату розробників Z_d . Розрахуємо її як 11% від основної заробітної плати:

$$Z_d = Z_o \cdot 0,11 [\text{грн}]. \quad (5.2)$$

$$Z_d = 39363,55 \cdot 0,11 = 4329,99 (\text{грн}).$$

Згідно діючого законодавства нарахування на заробітну плату (Єдиний соціальний внесок) становлять 22% від суми основної та додаткової заробітної плати.

$$H_z = (Z_{op} + Z_d) \cdot 0,22 \text{ [грн]}. \quad (5.3)$$

$$H_z = (39\,363,55 + 4329,99) \cdot 0,22 = 9612,46 \text{ (грн)}.$$

Амортизація обладнання та приміщення, яке використовувалось для проведення розробки, розраховується за формулою:

$$A = \frac{Ц \cdot H_a \cdot T}{100 \cdot 12} \text{ [грн]}, \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

H_a – річна норма амортизаційних відрахувань;

T – термін використання під час розробки, місяців.

Норма амортизації розраховується за формулою:

$$H_a = \frac{B_n - B_l}{B_n \cdot T_{кв}} \cdot 100 \text{ [грн]}, \quad (5.5)$$

де B_n і B_l – відповідно первісна та ліквідаційна вартість основних фондів;

$T_{кв}$ – строк корисного використання.

$$H_a = \frac{20000 - 2000}{20000 \cdot 5} \cdot 100 = 18\% \text{ (грн)}.$$

Розрахуємо амортизаційні витрати на ноутбук, балансова вартість якого становить 40000 грн, а термін використання – 3 місяці:

$$A = \frac{40000 \cdot 18 \cdot 3}{100 \cdot 12} = 1800 \text{ (грн)}.$$

Інформацію про матеріали, що використовуються при виготовленні даного продукту винесемо до таблиці 5.7.

Таблиця 5.7 – Матеріали, що використовуються при виготовленні даного продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір (пачка)	265,00	1	265,00
Ручка	8,00	1	8,00
Флешка	129,00	1	129,00
Всього			402

Під час розробки програмного продукту використовувались безкоштовні програмні засоби.

Також потрібно врахувати витрати на доступ до мережі Інтернет, що використовувався під час виконання роботи.

Витрати за доступ до Інтернет можна розрахувати за формулою:

$$B_{di} = C_{di} \cdot T \text{ [грн]}, \quad (5.6)$$

де C_{di} – це ціна доступу за місяць;

T – кількість місяців використання доступу до мережі.

Отже, витрати на доступ до мережі Інтернет становлять:

$$B_{di} = 290 \cdot 3 = 870,00 \text{ (грн)}.$$

Інші витрати B_{in} охоплюють: витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних

засобів, витрати на опалення, електроенергію, водопостачання. Інші витрати I_B можна прийняти як 100% від суми основної заробітної плати розробника:

$$V_{ін} = 1 \cdot 36000 = 36000 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – V .

$$V = Z_o + Z_d + H_{зп} + A + V_{мат} + V_e + V_{ін} \text{ [грн]},$$

$$V = 39\,363,55 + 4330 + 9612,46 + 1800 + 402 + 870 + 36000 = 59978 \text{ (грн)}.$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{V_{заг}}{\beta} \text{ [грн]}, \quad (5.7)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0,1$;
- на стадії технічного проектування, то $\beta \approx 0,2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0,3$;
- на стадії розробки технологій, то $\beta \approx 0,4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0,5$;
- на стадії розробки промислового зразка, $\beta \approx 0,7$;
- на стадії впровадження, то $\beta \approx 0,9$.

Отже, підставимо дані в формулу й отримаємо результат:

$$ZB = \frac{59978}{0,9} = 66\,642,23 \text{ (грн)}.$$

5.3 Розрахунок економічної ефективності та обґрунтування економічної доцільності комерціалізації науково-технічної розробки

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Виконання даної наукової роботи та впровадження її результатів складає приблизно до 1 року. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальніше прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Збільшення чистого прибутку розраховуємо за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i \text{ [грн]}, \quad (5.8)$$

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.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток збільшиться на 70 грн., а кількість реалізованих послуг збільшиться: протягом першого року – на 800 од., протягом другого року – ще на 1000 од., протягом третього року – ще на 1200 од.

Орієнтовно: реалізація до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 100 грн.

Збільшення чистого прибутку $\Delta\Pi_1$ протягом першого року складе:

$$\Delta\Pi_1 = 100 \cdot 1 + (100 + 70) \cdot 800 = 136100 \text{ (грн.)}$$

Обчислимо збільшення чистого прибутку $\Delta\Pi_2$ протягом другого року:

$$\Delta\Pi_2 = 100 \cdot 1 + (100 + 70) \cdot (800 + 1000) = 306100 \text{ (грн.)}$$

Збільшення чистого прибутку а $\Delta\Pi_3$ протягом третього року становитиме:

$$\Delta\Pi_3 = 100 \cdot 1 + (100 + 70) \cdot (800 + 1000 + 1200) = 510100 \text{ (грн.)}$$

Отже, розрахунки показують, що комерційний ефект від впровадження розробки виражається у значному щорічному збільшенні чистого прибутку підприємства.

Основними показниками, які визначають доцільність фінансування наукової розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^m \frac{\Delta\Pi_i}{(1 + \tau)^i} [\text{грн}], \quad (5.9)$$

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

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

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$\text{ПП} = \frac{136100}{(1 + 0,1)^1} + \frac{306100}{(1 + 0,1)^2} + \frac{510100}{(1 + 0,1)^3} = 759\,948,16 \text{ (грн)}.$$

Розраховуємо величину початкових інвестицій PV , які розробник (замовник) має вкласти для здійснення науково-технічної розробки.

Для цього можна використати формулу:

$$PV = k_{\text{розр}} \cdot ZB, \quad (5.10)$$

де розр k – коефіцієнт, що враховує витрати розробника (замовника) на впровадження науково-технічної розробки. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай розр $k = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 3 \cdot 66\,642,23 = 199\,926,69 \text{ (грн)}.$$

Розраховуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$.

Для цього використовується формула:

$$E_{\text{абс}} = (\text{ПП} - PV) [\text{грн}], \quad (5.11)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій PV = 3B, грн.

Розрахуємо абсолютну ефективність інвестицій, вкладених у реалізацію проекту:

$$E_{\text{абс}} = 759\,948,16 - 199\,926,69 = 560\,021,47 \text{ (грн)}.$$

Якщо $E_{\text{абс}} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту (роботи).

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів є доцільним.

Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$. Для цього використовуємо формулу:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1 [\text{грн}], \quad (5.12)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

Відносна (щорічна) ефективність вкладених інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_{\text{в}} = \sqrt[3]{1 + \frac{560\,021,47}{199\,926,69}} - 1 = \sqrt[3]{3,8} - 1 = 0,60 \text{ або } 60\%$$

Розрахована величина $E_{\text{в}}$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування τ мін, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ мін визначається за формулою:

$$\tau = d + f, \quad (5.13)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; $d = 0,05$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = 0,05$.

Якщо величина $E_B > \tau$ мін, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде. Спочатку спрогнозуємо величину τ мін.

За даних умов:

$$\tau \text{ мін} = 0,05 + 0,05 = 0,1.$$

Оскільки $E_B = 60\% > \tau \text{ мін} = 0,1 = 10\%$, це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

Розраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій.

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

$$T_{ок} = \frac{1}{E_g} [\text{грн}]. \quad (5.14)$$

Для нашої розробки термін окупності вкладених у реалізацію проекту інвестицій $T_{ок}$ складе:

$$T_{ок} = \frac{1}{0,60} = 1,6 \text{ (роки)}.$$

Оскільки термін окупності $T_{ок} = 1,6$ року і є меншим 3-ох років, то фінансування даної науково-технічної розробки буде доцільним.

5.4 Висновки

В даному розділі було виконано оцінювання комерційного потенціалу розробки програмного продукту. Проведено комерційний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.

Згідно із розрахунками всіх статей витрат на виконання науково-технічної роботи загальні витрати на розробку складають 66 642,23 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 560 021,47 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 60%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 10%. Це означає потенційну зацікавленість інвестора у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1 рік 7 місяців, що також свідчить про доцільність комерціалізації розробки.

ВИСНОВКИ

У магістерській кваліфікаційній роботі були розроблені методи для підвищення ефективності прогнозування продажів товарів, працездатність яких демонстрована за допомогою розробленого програмного додатку для прогнозування продажів мовою С# у середовищі розробки Visual Studio 2022.

Проаналізовано стан питання прогнозування продажів. Було проаналізовано переваги та недоліки аналогів програмного засобу для прогнозування, в ході якого підтвердилась необхідність створення власного продукту. Поставлено та виконано усі задачі для розробки.

Було розроблено адаптивний метод прогнозування продажів, який підбирає найбільш оптимальний метод прогнозування, що підвищує ефективність прогнозу. Розроблено метод автоматизованого формування звіту, за допомогою якого підвищилась ефективність оцінювання результатів прогнозування. Розроблено структуру інтерфейсу програмного засобу. Розроблено алгоритми роботи програми.

Було проведено аналіз мов програмування, та обрано засоби мови С#. Обґрунтовано вибір середовища програмування Visual Studio. Розроблено базу даних. Розроблені основні модулі програми. В процесі розробки було описано програмну реалізацію основних модулів додатку.

Було проаналізовано методи тестування програмних додатків. Проведено тестування обраним методом, в ході якого була підтверджена працездатність програмного продукту. Встановлено вимоги до персонального комп'ютера користувача. Розроблено інструкцію користувача.

Проведено оцінювання комерційного потенціалу розробки, яке показало, що розробка має достатній рівень комерційного потенціалу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Прогнозування продажу та його методи в системі управління підприємством. [Електронний ресурс].URL: <http://surl.li/ebpkmk>
2. Сікорська М.О., Романюк О.В. Розробка програмного забезпечення для прогнозування продажів товарів // Цифрові технології в освіті та промисловості – 2022 : матеріали Міжнародної науково-практичної Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» Вінниця 28-29 листопада 2022 р. [Електронний ресурс]. URL: <http://academia.vinnica.ua/>
3. Сікорська М.О., Романюк О.В. Розробка адаптивного підходу до підбору методу прогнозування продажів // Комп'ютерна та програмна інженерія – 2022 : матеріали IV міжнародна наукова конференція науковий простір: актуальні питання, досягнення та інновації Івано-Франківськ 2022 р. [Електронний ресурс].URL:<https://archive.mcnd.org.ua/index.php/conference-proceeding/issue/view/02.12.2022/14>
4. Ефективне прогнозування продажів. [Електронний ресурс]. URL: <https://coworkingplatforma.com/ua/blog/efektivne-prognozuvanna-prodaziv/>
5. Прогнозування продажу [Електронний ресурс]. URL: <https://core.ac.uk/download/pdf/161260659.pdf>
6. Найкращі програми для прогнозування продажів. URL: <https://gmdhsoftware.com/uk/unleashed-sales-forecasting-software/>
7. Інтуїтивні методи прогнозування. [Електронний ресурс]. URL: https://stud.com.ua/91185/investuvannya/intuyitivni_metodi_prognozuvannya
8. Техніки скаладання прогнозу збуту [Електронний ресурс]. URL: <http://surl.li/ebqxt>
9. Аналіз методів прогнозування продажів в інтернет-магазині / О. С. Маліцький, О. В. Романюк // Матеріали XLVI науково-технічної конференції підрозділів ВНТУ, Вінниця, 22-24 [Електронний ресурс]. URL: <http://surl.li/ebpml>
10. Методи прогнозування продажів товарі. [Електронний ресурс].URL: <http://surl.li/vlrn>

11. Веб-дизайн та кольорова гамма. [Електронний ресурс]. URL:
<http://www.websvit.com/blog/2015/01/31/veb-dyzajn-ta-kolorova-gamma/>
12. Значення кольорів. [Електронний ресурс]. URL:
<https://lemarbet.com/ua/otkrytie-internet-magazina/znachenie-tsveta-v-dizajne-internet-magazina/>
13. Діаграми варіантів використання. [Електронний ресурс]. URL:
<http://surl.li/cefsh>
14. C++. [Електронний ресурс]. URL:
<https://uk.wikipedia.org/wiki/C%2B%2B>
15. C#. [Електронний ресурс]. URL:
<https://metanit.com/sharp/tutorial/1.1.php>
16. Java. [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki/Java>
17. CodeBlocks. [Електронний ресурс]. URL: <https://www.codeblocks.org/>
18. Eclipse. [Електронний ресурс]. URL:
<https://uk.wikipedia.org/wiki/Eclipse>
19. MS Visual Studio. [Електронний ресурс]. URL:
https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
20. Entity Framework. [Електронний ресурс]. URL:
<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>
21. SQL Server. [Електронний ресурс]. URL:
<https://metanit.com/sql/sqlserver/1.1.php>
22. Software testing methods. [Електронний ресурс]. URL:
www.tutorialspoint.com/software_testing/software_testing_methods.htm
23. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. 42 с.

ДОДАТКИ

ДОДАТОК А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

д.т.н., проф. О. Н. Романюк


"16" вересня 2022 р.

Технічне завдання

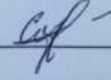
на магістерську кваліфікаційну роботу «Розробка методів та програмних засобів для підвищення ефективності прогнозування продажів товарів» за спеціальністю

121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

 к.т.н., доц. каф. ПЗ Романюк О.В.
"15" вересня 2022 р.

Виконав:

 студентка гр.1ПІ-21м М. О. Сікорська
"15" вересня 2022 р.

Вінниця – 2022 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів».

Галузь застосування – бізнес та торгівля.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № 205-А ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення ефективності прогнозування продажів товарів за рахунок автоматизованого підбору оптимального методу прогнозування, що відповідає наявним вхідним даним про продажі товару за минулі періоди.

Призначення роботи – розробка методів і засобів для підвищення ефективності прогнозування продажів.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Прогнозування продажу та його методи в системі управління підприємством/ Дячун О.Д., 2016. – 129 с.

2. Аналіз методів прогнозування продажів в інтернет-магазині / О. С. Маліцький, О. В. Романюк // Матеріали XLVI науково-технічної конференції підрозділів ВНТУ, Вінниця, 22-24 березня 2017 р.

3. Sales Forecasting Methods / Nihamoni S.// EconomicsDiscussion, 2018 р.

5. Технічні вимоги

Методи прогнозування продажу – арифметичне згладжування, експоненційне згладжування, правило півтора, метод Хольта-Вінтерса, метод прогнозування сезонного попиту з урахуванням коефіцієнту сезонності; особисті дані користувача, дані продажу минулого періоду; середовища розробки – Visual Studio 2022 та Microsoft SQL Server; мова розробки – C#; операційна система – Windows 10.

6. Конструктивні вимоги

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз методів та методик прогнозування	17.09.2022 – 30.09.2022
2	Розробка структури системи, адаптивного методу підвищення ефективності прогнозування продажів товарів, методу формування звіту з прогнозування	01.10.2022 – 19.10.2022
3	Розробка програмного забезпечення для прогнозування продажів товарів	19.10.2022 – 10.11.2022
4	Тестування програмного додатку	11.11.2022 – 25.11.2022
5	Економічна частина	26.11.2022 – 8.12.2022

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком

ДОДАТОК Б
ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ)
РОБОТИ

Назва роботи: **Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів.**

Тип роботи: кваліфікаційна робота

Підрозділ : кафедра програмного забезпечення, ФІТКІ, ІПП-21м

Науковий керівник: к.т.н. доц. каф. ПЗ Романюк О. В.

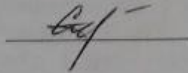
Unicheck	
Оригінальність	94,1%
Схожість	5,9%

Аналіз звіту подібності

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомена з повним звітом подібності, який був згенерований Системою щодо роботи «Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів».

Автор



Сікорська Марія Олександрівна


Опис прийнятого рішення: **допустити до захисту**

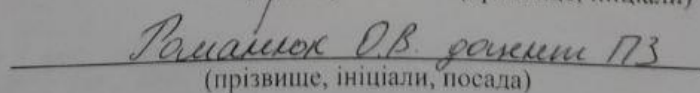
Особа, відповідальна за перевірку


(підпис)

Черноволик Г. О.
(прізвище, ініціали)

Експерт
(за потреби)


(підпис)


(прізвище, ініціали, посада)

ДОДАТОК В

ЛІСТИНГ КОДУ

User.cs.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SalesForecastApp.Enteties
{
    public class User
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Surname { get; set; }
        public string Login { get; set; }
        public string Password { get; set; }
        public string CompanyName { get; set; }
        public string ImagePath { get; set; }
        public virtual ICollection<Report> Reports { get; set; }
    }
}
```

Report.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SalesForecastApp.Enteties
{
    public class Report
    {
        public int Id { get; set; }

        [StringLength(40)]
        [Index(IsUnique = true)]
        [Required]
        public string Name { get; set; }
        public string CompanyName { get; set; }
        public string Product { get; set; }
        public string Info { get; set; }
        public double GDP { get; set; }

        public double Inflation { get; set; }
        public int Profitability { get; set; }
        public string Competitors { get; set; }
        public string Aim { get; set; }
        public User User { get; set; }
        public int UserId { get; set; }
        public string MethodName { get; set; }
        public string MethodResult { get; set; }
    }
}
```

Authorization.cs

```

using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class Authorization : Form
    {
        public Authorization()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }

        private async void AuthorizationConfirmBtn_Click(object sender, EventArgs e)
        {
            using (var context = new SalesAppContext())
            {
                var user = await Task.Run(async () => await context.User
                    .FirstOrDefaultAsync(s => s.Login.Equals(Authorization_Login.Text)
                    &&
                    s.Password.Equals(Authorization_Password.Text)));
                if (user != null)
                {
                    MainScreen mainScreen = new MainScreen();

                    CurrentUser.SetupData(user.Id, user.Name, "company",
                    user.Password, user.Login, user.ImagePath);
                    mainScreen.Show();
                    this.Hide();
                }
                else
                {
                    MessageBox.Show("Невірні дані для входу");
                }
            }
        }

        private void AuthorizationRegistrationBtn_Click(object sender, EventArgs e)
        {
            Registration registration = new Registration();

```

```

        if (registration.ShowDialog(this) == DialogResult.OK)
        {
            MessageBox.Show("Користувач зареєстрований");
        }
    }
}

```

Registration.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class Registration : Form
    {
        public Registration()
        {
            InitializeComponent();
        }

        private void Registration_Load(object sender, EventArgs e)
        {
        }

        private void RegistrConfirmBtn_Click(object sender, EventArgs e)
        {
            User newUser = new User
            {
                Login = RegistrLoginTextBox.Text,
                Password = RegistrPasswordTextBox.Text,
                Name = RegistrNameTextBox.Text
            };

            using (var context = new SalesAppContext())
            {
                var existedUser = context.User.FirstOrDefault(s =>
s.Login.Equals(newUser.Login));

                if (existedUser != null)
                {
                    MessageBox.Show("Користувач вже існує");
                    return;
                }
                context.User.Add(newUser);
                context.SaveChanges();
            }

            this.DialogResult = DialogResult.OK;
            this.Close();
        }
    }
}

```

MainScreen.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class MainScreen : Form
    {
        public MainScreen()
        {
            InitializeComponent();
        }

        private void MainScreen_Load(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Reports reports = new Reports();
            this.Hide();

            if (reports.ShowDialog() == DialogResult.OK)
            {
                this.Show();
            }
        }

        private void toolStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }

        private void toolStripButton2_Click(object sender, EventArgs e)
        {
            Profile profile = new Profile();
            this.Hide();

            if (profile.ShowDialog() == DialogResult.OK)
            {
                this.Show();
            }
        }

        private void buttonForecastMain_Click(object sender, EventArgs e)
        {
            ForecastStart frstart = new ForecastStart();
            this.Hide();

            if (frstart.ShowDialog() == DialogResult.OK)
            {
                this.Show();
            }
        }
    }
}

```

```

    }
}

private void button1_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(button1, "Крок 1: Введення основної інформації");
}
private void button2_MouseHover(object sender, EventArgs e)
{
    toolTip2.SetToolTip(button2, "Крок 2: Аналіз макроекономічних
індикаторів");
}

private void button3_MouseHover(object sender, EventArgs e)
{
    toolTip3.SetToolTip(button3, "Крок 3: Введення даних для визначення
рентабельності");
}

private void button4_MouseHover(object sender, EventArgs e)
{
    toolTip4.SetToolTip(button4, "Крок 4: Встановлення мети");
}

private void button5_MouseHover(object sender, EventArgs e)
{
    toolTip5.SetToolTip(button5, "Крок 5: Визначення конкурентів");
}

private void button6_MouseHover(object sender, EventArgs e)
{
    toolTip6.SetToolTip(button6, "Крок 6: Введення даних для визначення методу
прогнозування");
}

private void button7_MouseHover(object sender, EventArgs e)
{
    toolTip7.SetToolTip(button7, "Крок 7: Прогнозування");
}

private void button8_MouseHover(object sender, EventArgs e)
{
    toolTip8.SetToolTip(button8, "Крок 8: Отримання звіту!");
}

private void toolStripLabel1_Click(object sender, EventArgs e)
{
}

private void toolTip1_Popup(object sender, PopupEventArgs e)
{
}

private void button4_Click(object sender, EventArgs e)
{
}
}}

```

ArithmeticSmoothing.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class ArithmeticSmoothing : Form
    {
        public ArithmeticSmoothing()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void calculate_Click(object sender, EventArgs e)
        {
            var a = textBox1.Text.Split('\n').Select(x =>
            {
                double result;
                var success = double.TryParse(x, out result);
                return new { a = success, b = result };
            })
                .Where(s => s.a)
                .Select(s => s.b).ToArray();

            ArismaticResult.Text = (a.Sum() / a.Length).ToString();
        }

        private void ArismaticResult_Click(object sender, EventArgs e)
        {
        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            char ch = e.KeyChar;
            if (!Char.IsDigit(ch) && ch != '\r')
            {
                e.Handled=true;
            }
        }

        private async void Nextbtn_Click(object sender, EventArgs e)
        {
            Report report;

            using (var context = new SalesAppContext())
            {

```

```

        report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(x => x.Id == CurrentUser.CurrentReportId));
        report.MethodResult = ArismeticResult.Text;
        report.MethodName = "Арифметичне згладжування";

        await context.SaveChangesAsync();
    }

    Result result = new Result(report);

    result.ShowDialog();
    this.Close();

}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{

}

}
}

```

ExpSmoothing.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class ExpSmoothing : Form
    {
        public ExpSmoothing()
        {
            InitializeComponent();

        }

        List<double> result1 = new List<double>();

        private void ExpResult_Click(object sender, EventArgs e)
        {

        }

        private void ExpCalculate_Click(object sender, EventArgs e)
        {

```



```

string exponential_smoothing(double[] series, double alpha)
{
    result1.Add(series[0]);

    for (int i = 1; i < series.Length; i++)
    {
        result1.Add( (alpha * series[i] + (1 - alpha) * series[i - 1]));
    }
    void fillChart(List<double> c)
    {
        double[] newarr = c.ToArray();
        for (int i = 0; i < newarr.Length; i++)
        {
            chart1.Series["ExponentialSmoothing"].Points.AddXY((i + 1),
newarr[i]);
        }
    }

    fillChart(result1);
    string result = string.Join("\n", result1.Select(x => x.ToString()));

    return result;
}
double[] ser = textBox1.Text.Split('\n').Select(x =>
{
    double res;
    var success = double.TryParse(x, out res);
    return new { a = success, b = res};
})
.Where(s => s.a)
.Select(s => s.b).ToArray();
var k = exponential_smoothing(ser, 0.9);
ExpResult.Text = k;

}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch) && ch != '\r' && ch != '\b')
    {
        e.Handled = true;
    }
}
private async void Nextbtn_Click(object sender, EventArgs e)
{
    Report report;

    using (var context = new SalesAppContext())
    {
        report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(x => x.Id == CurrentUser.CurrentReportId));
        report.MethodResult = ExpResult.Text;
        report.MethodName = "Експоненційне згладжування";

        await context.SaveChangesAsync();
    }
}

```

```

        Result result = new Result(report);
        this.Close();
        result.ShowDialog();

    }

    private void chart1_Click(object sender, EventArgs e)
    {

    }

    private void textBox1_KeyPress_1(object sender, KeyPressEventArgs e)
    {

        char ch = e.KeyChar;
        if (!Char.IsDigit(ch) && ch != '\r')
        {
            e.Handled = true;
        }
    }

    private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
    {

        char ch = e.KeyChar;
        if (!Char.IsDigit(ch) && ch != '\r')
        {
            e.Handled = true;
        }
    }
}
}

```

ForecastStart.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class ForecastStart : Form
    {
        public ForecastStart()
        {
            InitializeComponent();
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

```

```

}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
}

private void groupBox1_Enter_1(object sender, EventArgs e)
{
}

private void label5_Click(object sender, EventArgs e)
{
}

private async void buttonNextForecastStart_Click(object sender, EventArgs e)
{
    int a = 0;
    if (!string.IsNullOrWhiteSpace(textBox1.Text) &&
        !string.IsNullOrWhiteSpace(textBox2.Text))
    {
        a = (Convert.ToInt32(textBox1.Text)) /
            (Convert.ToInt32(textBox2.Text))*100;
    }
    Report report = new Report
    {
        UserId = CurrentUser.Id,
        CompanyName = CompanyNameTextBox.Text,
        Product = ProductTypeTextBox.Text,
        Info = InforichTextBox.Text,
        Aim = AimRichTextBox.Text,
        Competitors = CompetitorsRichTextBox.Text,
        Profitability = a,
        Inflation = 124,
        GDP = 200.1,
        Name = $"{DateTime.Now.ToShortTimeString()}_DefaultName"
    };
    using (var context = new SalesAppContext())
    {
        context.Reports.Add(report);
        await context.SaveChangesAsync();

        CurrentUser.CurrentReportId = report.Id;
    }
    MethodPicker methodpick = new MethodPicker();
    this.Hide();

    if (methodpick.ShowDialog() == DialogResult.OK)
    {
        this.Show();
    }
}

private void webBrowser1_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
{
}

```

```

private void ForecastStart_Load(object sender, EventArgs e)
{
    webBrowser1.Navigate("https://tradingeconomics.com/ukraine/inflation-
cpi");

    webBrowser2.Navigate("https://www.worldometers.info/gdp/ukraine-gdp/");
    //"fhttps://tradingeconomics.com/ukraine/gdp"
}

private void webBrowser1_Navigating(object sender,
WebBrowserNavigatingEventArgs e)
{
    this.Text = e.Url.ToString() + "Завантаження...";
}

public bool ScriptErrorsSuppressed { get; set; }
private void SuppressScriptErrorsOnly(WebBrowser browser)
{
    // Ensure that ScriptErrorsSuppressed is set to false.
    browser.ScriptErrorsSuppressed = false;

    // Handle DocumentCompleted to gain access to the Document object.
    browser.DocumentCompleted +=
        new WebBrowserDocumentCompletedEventHandler(
            browser_DocumentCompleted);
}

private void browser_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
{
    ((WebBrowser)sender).Document.Window.Error +=
        new HTMLElementErrorEventHandler(Window_Error);
}

private void Window_Error(object sender,
HTMLElementErrorEventArgs e)
{
    // Ignore the error and suppress the error dialog box.
    e.Handled = true;
}

private void groupBox2_Enter(object sender, EventArgs e)
{
}

private void buttonReturnForecastStart_Click(object sender, EventArgs e)
{
    MainScreen main = new MainScreen();
    this.Hide();

    if (main.ShowDialog() == DialogResult.OK)
    {
        this.Show();
    }
}
}
}
}
}
}

```

HoltWinters.cs

```
using SalesForecastApp.Enteties;
```

```

using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class HoltWinters : Form
    {
        public string result;

        public HoltWinters()
        {
            InitializeComponent();
        }

        /**
         * This method is the entry point. It calculates the initial values and returns
         the forecast
         * for the m periods.
         *
         * @param y - Time series data.
         * @param alpha - Exponential smoothing coefficients for level, trend, seasonal
         components.
         * @param beta - Exponential smoothing coefficients for level, trend, seasonal
         components.
         * @param gamma - Exponential smoothing coefficients for level, trend, seasonal
         components.
         * @param perdioid - A complete season's data consists of L periods. And we need
         to estimate
         * the trend factor from one period to the next. To accomplish this, it is
         advisable to use
         * two complete seasons; that is, 2L periods.
         * @param m - Extrapolated future data points.
         * @param debug - Print debug values. Useful for testing.
         *
         *           4 quarterly
         *           7 weekly.
         *           12 monthly
         */

        public static class HoltWintersTripleExponentialImpl
        {
            public static double[] forecast(int[] y, double alpha, double beta, double
            gamma, int period, int m)
            {
                if (y == null)
                {
                    return null;
                }
                var seasons = (int)(y.Length / period);
                var a0 = HoltWintersTripleExponentialImpl.calculateInitialLevel(y,
            period);

```

```

    var b0 = HoltWintersTripleExponentialImpl.calculateInitialTrend(y,
period);
    double[] initialSeasonalIndices = calculateSeasonalIndices(y, period,
seasons);

    double[] forecast =
HoltWintersTripleExponentialImpl.calculateHoltWinters(y, a0, b0, alpha, beta, gamma,
initialSeasonalIndices, period, m);

    return forecast;
}
// *
// * This method realizes the Holt-Winters equations.
// *
// * @param y
// * @param a0
// * @param b0
// * @param alpha
// * @param beta
// * @param gamma
// * @param initialSeasonalIndices
// * @param period
// * @param m
// * @param debug
// * @return - Forecast for m periods.
private static double[] calculateHoltWinters(int[] y, double a0, double b0,
double alpha, double beta, double gamma, double[] initialSeasonalIndices, int period,
int m)
{
    double[] St = new double[y.Length];
    double[] Bt = new double[y.Length];
    double[] It = new double[y.Length];
    double[] Ft = new double[y.Length + m];
    // Initialize base values
    St[1] = a0;
    Bt[1] = b0;
    for (int i = 0; i < period; i++)
    {
        It[i] = initialSeasonalIndices[i];
    }
    Ft[m] = (St[0] + (m * Bt[0])) * It[0];
    // This is actually 0 since Bt[0] = 0
    Ft[m + 1] = (St[1] + (m * Bt[1])) * It[1];
    // Forecast starts from period + 2
    // Start calculations
    for (int i = 2; i < y.Length; i++)
    {
        // Calculate overall smoothing
        if ((i - period) >= 0)
        {
            St[i] = alpha * y[i] / It[i - period] + (1.0 - alpha) * (St[i - 1]
+ Bt[i - 1]);
        }
        else
        {
            St[i] = alpha * y[i] + (1.0 - alpha) * (St[i - 1] + Bt[i - 1]);
        }
        // Calculate trend smoothing
        Bt[i] = gamma * (St[i] - St[i - 1]) + (1 - gamma) * Bt[i - 1];
        // Calculate seasonal smoothing
        if ((i - period) >= 0)
        {
            It[i] = beta * y[i] / St[i] + (1.0 - beta) * It[i - period];
        }
        // Calculate forecast

```

```

        if ((i + m) >= period)
        {
            Ft[i + m] = (St[i] + (m * Bt[i])) * It[i - period + m];
        }
    }
    return Ft;
}

// * 1st period's average can be taken. But y[0] works better.
// *
// * @return - Initial Level value i.e. St[1]
private static double calculateInitialLevel(int[] y, int period)
{
    // *
    // *      double sum = 0;
    // *      for (int i = 0; i < period; i++) {
    // *          sum += y[i];
    // *      }
    // *      return sum / period;
    // *
    return y[0];
}

// *
// *
// *
// * @return - Initial trend - Bt[1]
private static double calculateInitialTrend(int[] y, int period)
{
    var sum = 0;
    for (int i = 0; i < period; i++)
    {
        sum += (y[period + i] - y[i]);
    }
    return sum / (period * period);
}

private static double[] calculateSeasonalIndices(int[] y, int period, int
seasons)
{
    double[] seasonalAverage = new double[seasons];
    double[] seasonalIndices = new double[period];
    double[] averagedObservations = new double[y.Length];
    for (int i = 0; i < seasons; i++)
    {
        for (int j = 0; j < period; j++)
        {
            seasonalAverage[i] += y[(i * period) + j];
        }
        seasonalAverage[i] /= period;
    }
    for (int i = 0; i < seasons; i++)
    {
        for (int j = 0; j < period; j++)
        {
            averagedObservations[(i * period) + j] = y[(i * period) + j] /
seasonalAverage[i];
        }
    }
    for (int i = 0; i < period; i++)
    {
        for (int j = 0; j < seasons; j++)
        {
            seasonalIndices[i] += averagedObservations[(j * period) + i];
        }
    }
}

```

```

        seasonalIndices[i] /= seasons;
    }
    return seasonalIndices;
}
// *
// * Utility method to print array values.
// *
// * @param description
// * @param data
}

private void Calculate_Click(object sender, EventArgs e)
{
    // var Arr = new int[];
    var a = textBox1.Text.Split('\n').Select(x =>
    {
        int result;
        var success = int.TryParse(x, out result);
        return new { a = success, b = result };
    })
    .Where(s => s.a)
    .Select(s => s.b).ToArray();

    /* int[] a = {30,21,29,31,40,48,53,47,37,39,31,29,17,9,20,24,27,35,41,38,
    27,31,27,26,21,13,21,18,33,35,40,36,22,24,21,20,17,14,17,19,
    26,29,40,31,20,24,18,26,17,9,17,21,28,32,46,33,23,28,22,27,
    18,8,17,21,31,34,44,38,31,30,26,32};*/

    int p= (a.Length/2);
    int mm = p;

    double alpha = double.Parse(alphatext.Text);

    double beta = double.Parse(betatext.Text);

    double gamma = double.Parse(gammatext.Text);

    var c = HoltWintersTripleExponentialImpl.forecast(a, alpha, beta, gamma,
p,mm).Where(x => x > 0).ToArray();
    var ch = c.Select(x => x.ToString());
    string k = String.Join("\r", ch);

    Result.Text = k;

    void fillChart()
    {
        for (int i = 0; i < c.Length; i++) {

            chart1.Series["HoltWinters"].Points.AddXY((i+1), c[i]);

        }
    }
    fillChart();
}

private async void NextBtn_Click(object sender, EventArgs e)
{
    Report report;

    using (var context = new SalesAppContext())
    {

```



```

        report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(x => x.Id == CurrentUser.CurrentReportId));
        report.MethodResult = Result.Text;
        report.MethodName = "Метод Хольта Вінтерса";

        await context.SaveChangesAsync();
    }

    Result result = new Result(report);

    result.ShowDialog();
    this.Close();

}

private void textBox1_KeyPress_1(object sender, KeyPressEventArgs e)
{

    char ch = e.KeyChar;
    if (!Char.IsDigit(ch) && ch != '\r')
    {
        e.Handled = true;
    }
}
}
}

```

RuleOneandHalf.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class RuleOneAndHalf : Form
    {
        public RuleOneAndHalf()
        {
            InitializeComponent();
        }

        private void Title_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }
    }
}

```

```

private void ArismaticResult_Click(object sender, EventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}

private void Result_Click(object sender, EventArgs e)
{
    var zpp = Convert.ToDouble(Zpp.Text);
    var pst = Convert.ToDouble(Pst.Text);
    var zkp = Convert.ToDouble(Zkp.Text);

    ArismaticResult.Text = ((zpp+pst-zkp) * 1.5 -zkp).ToString();
}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch))
    {
        e.Handled = true;
    }
}

private void Pst_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch))
    {
        e.Handled = true;
    }
}

private void Zkp_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch))
    {
        e.Handled = true;
    }
}

private async Task Nextbtn_Click(object sender, EventArgs e)
{
    Report report;

    using (var context = new SalesAppContext())
    {
        report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(x => x.Id == CurrentUser.CurrentReportId));
        report.MethodResult = ArismaticResult.Text;
        report.MethodName = "Правило 1.5";

        await context.SaveChangesAsync();
    }

    Result result = new Result(report);

    result.ShowDialog();
}

```

```

        this.Close();
    }

    private void label6_Click(object sender, EventArgs e)
    {

    }

    private void label7_Click(object sender, EventArgs e)
    {

    }

    private void label5_Click(object sender, EventArgs e)
    {

    }
}
}
}

```

SeasonalForecast.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SalesForecastApp
{
    public partial class SeasonalForecast : Form
    {
        public SeasonalForecast()
        {
            InitializeComponent();
        }

        private void result_Click(object sender, EventArgs e)
        {

            var a = textBox1.Text.Split('\n').Select(x =>
            {
                double result;
                var success = double.TryParse(x, out result);
                return new { a = success, b = result };
            })
                .Where(s => s.a)
                .Select(s => s.b).ToArray().Sum();

            var d = textBox3.Text.Split('\n').Select(x =>
            {
                double result;

```

```

        var success = double.TryParse(x, out result);
        return new { d = success, c = result };
    })
    .Where(s => s.d)
    .Select(s => s.c).ToArray().Sum();
    var g = Convert.ToDouble(textBox2.Text);
    result1.Text = (g+(g*(d-a)/a)).ToString()+"%";
}

private void textBox1_KeyPress_1(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch) && ch != '\r')
    {
        e.Handled = true;
    }
}

private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (!Char.IsDigit(ch) && ch != '\r')
    {
        e.Handled = true;
    }
}

private void result1_Click(object sender, EventArgs e)
{
}

private async void button1_Click(object sender, EventArgs e)
{
    Report report;

    using (var context = new SalesAppContext())
    {
        report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(x => x.Id == CurrentUser.CurrentReportId));
        report.MethodResult = result1.Text;
        report.MethodName = "Прогнозування сезного попиту";

        await context.SaveChangesAsync();
    }

    Result result = new Result(report);

    result.ShowDialog();
    this.Close();
}
}
}

```

Result.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;

```

```

using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Printing;

namespace SalesForecastApp
{
    public partial class Result : Form
    {
        private Report _report;

        public Result()
        {
            InitializeComponent();
        }

        public Result(Report report)
        {
            _report = report;
            InitializeComponent();
        }

        private void Title_Click(object sender, EventArgs e)
        {
        }

        private void Result_Load(object sender, EventArgs e)
        {
            textBox1.Text = $"{_report.Name}";

            richTextBox1.Text = $"{_report.Name}\n\n" +
                $"{DateTime.Now.ToShortDateString()}\n\n" +
                $"Назва компанії:\n" + $"{_report.CompanyName}\n\n" +
                $"Тип продукції: \n {_report.Product}\n\n" +
                $"Основна інформація: \n {_report.Info}\n\n" +
                $"Інфляція: {_report.Inflation}\n\n" +
                $"ВВП: {_report.GDP}\n\n" +
                $"Рентабельність: {_report.Profitability}\n\n" +
                $"Мета: \n {_report.Aim}\n\n" +
                $"Конкуренти: \n {_report.Competitors}\n\n"+
                $"Прогноз методом {_report.MethodName}: \n {_report.MethodResult}\n";
        }

        private void buttonSave_Click(object sender, EventArgs e)
        {
            buttonSave.Click += buttonSave_Click;
            saveFileDialog1.Filter = "Text files(*.txt)|*.txt|All files(*.*)|*.*";

            if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
                return;
            // отримуємо вибраний файл
            string filepath = saveFileDialog1.FileName;
        }
    }
}

```

```

        // сохраняем текст в файл
        System.IO.File.WriteAllText(filepath, richTextBox1.Text);
        MessageBox.Show("Файл сохранен");
    }

    private async void button1_Click(object sender, EventArgs e)
    {
        using (var context = new SalesAppContext())
        {
            var report = await Task.Run(async () => await
context.Reports.FirstOrDefaultAsync(s => s.Id == CurrentUser.CurrentReportId));
            report.Name = textBox1.Text;

            await context.SaveChangesAsync();
            MessageBox.Show("Зміни збережено");
        }
    }

    private void buttonReportsToMenu_Click(object sender, EventArgs e)
    {
        MainScreen mainScreen = new MainScreen();

        mainScreen.Show();
        this.Close();
    }
    void PrintPageHandler(object sender, PrintPageEventArgs e)
    {
        e.Graphics.DrawString(richTextBox1.Text, new Font("Arial", 14),
Brushes.Black, 0, 0);
    }
    PrintDocument printDocument = new PrintDocument();
    private void buttonPrint_Click(object sender, EventArgs e)
    {
        printDocument.PrintPage += PrintPageHandler;
        PrintDialog printDialog = new PrintDialog();
        printDialog.Document = printDocument;
        if (printDialog.ShowDialog() == DialogResult.OK)
            printDialog.Document.Print();
    }
}
}

```

Reports.cs

```

using SalesForecastApp.Enteties;
using SalesForecastApp.Enteties.Context;
using SalesForecastApp.Globals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Printing;

```

```

namespace SalesForecastApp
{
    public partial class Reports : Form
    {
        private List<Report> _reports;
        private List<Report> _reportsSearch;

        public Reports()
        {
            InitializeComponent();
        }

        private async void Reports_Load(object sender, EventArgs e)
        {
            richTextBox1.SelectionIndent = 50;

            // TODO: данная строка кода позволяет загрузить данные в таблицу
            // "salesAppDataSet.Reports". При необходимости она может быть перемещена или удалена.
            using (var context = new SalesAppContext())
            {
                var reportsAndIndexes = (await Task.Run(async () => await
context.Reports
                .Where(r => r.UserId == CurrentUser.Id)
                .ToListAsync()))
                .Select((report, index) => new { Report = report, Index = index });

                _reports = reportsAndIndexes.Select(x => x.Report).ToList();

                foreach (var reportIndex in reportsAndIndexes)
                {
                    listBox1.Items.Add(reportIndex.Report.Name);
                }
                _reportsSearch = new List<Report>();
            }

            private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
            {
            }

            private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
            {
            }

            private void listBox1_DoubleClick(object sender, EventArgs e)
            {
                try
                {
                    Report _selectedReport;
                    if (string.IsNullOrEmpty(textBox1.Text))
                    {
                        _selectedReport = _reports[listBox1.SelectedIndex];
                    }
                    else
                    {
                        _selectedReport = _reportsSearch[listBox1.SelectedIndex];
                    }

                    richTextBox1.Text = $"\\n\\n{_selectedReport.Name}\\n\\n" +
                        $"Дата: {DateTime.Now.ToShortDateString()}\\n\\n" +
                        $"Назва компанії: \\n" + $"{_selectedReport.CompanyName}\\n\\n" +

```

```

        $"Тип продукції: \n {_selectedReport.Product}\n\n" +
        $"Основна інформація: \n {_selectedReport.Info}\n\n" +
        $"Інфляція: {_selectedReport.Inflation}\n\n" +
        $"ВВП: {_selectedReport.GDP}\n\n" +
        $"Рентабельність: {_selectedReport.Profitability}\n\n" +
        $"Мета: \n {_selectedReport.Aim}\n\n" +
        $"Конкуренти: \n {_selectedReport.Competitors}\n\n"+
        $"Прогноз методом {_selectedReport.MethodName}: \n
{_selectedReport.MethodResult}\n";
    }
    catch
    { }

}

private void buttonReportsToMenu_Click(object sender, EventArgs e)
{
    MainScreen mainScreen = new MainScreen();

    mainScreen.Show();
    this.Close();
}

private async void textBox1_TextChanged(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    _reportsSearch.Clear();

    var searchReportResult = _reports.Where(x =>
x.Name.StartsWith(textBox1.Text, StringComparison.OrdinalIgnoreCase))
        .Select((report, index) => new { Report = report, Index = index });

    _reportsSearch.AddRange(searchReportResult.Select(x => x.Report));

    foreach (var reportIndex in searchReportResult)
    {
        listBox1.Items.Add(reportIndex.Report.Name);
    }
}

private async void button2_Click(object sender, EventArgs e)
{
    Report _selectedReport;
    if (string.IsNullOrWhiteSpace(textBox1.Text))
    {
        _selectedReport = _reports[listBox1.SelectedIndex];
    }
    else
    {
        _selectedReport = _reportsSearch[listBox1.SelectedIndex];
    }

    using (var context = new SalesAppContext())
    {
        context.Entry(_selectedReport).State = EntityState.Deleted;
        await context.SaveChangesAsync();
    }

    _reports.Remove(_selectedReport);
    listBox1.Items.RemoveAt(listBox1.SelectedIndex);
    MessageBox.Show("Звіт видалено");
}
}

```



```

private void buttonSave_Click(object sender, EventArgs e)
{
    buttonSave.Click += buttonSave_Click;
    saveFileDialog1.Filter = "Text files(*.txt)|*.txt|All files(*.*)|*.*";

    if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
        return;
    // получаем выбранный файл
    string filepath = saveFileDialog1.FileName;
    // сохраняем текст в файл
    System.IO.File.WriteAllText(filepath, richTextBox1.Text);
    MessageBox.Show("Файл сохранен");
}

void PrintPageHandler(object sender, PrintPageEventArgs e)
{
    e.Graphics.DrawString(richTextBox1.Text, new Font("Arial", 14),
Brushes.Black, 0, 0);
}
PrintDocument printDocument = new PrintDocument();

private void button3_Click(object sender, EventArgs e)
{
    printDocument.PrintPage += PrintPageHandler;
    PrintDialog printDialog = new PrintDialog();
    printDialog.Document = printDocument;
    if (printDialog.ShowDialog() == DialogResult.OK)
        printDialog.Document.Print();
}
}
}

```

ДОДАТОК Г

ІЛЮСТРАТИВНА ЧАСТИНА

**РОЗРОБКА МЕТОДІВ І ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПІДВИЩЕННЯ
ЕФЕКТИВНОСТІ ПРОГНОЗУВАННЯ ПРОДАЖІВ ТОВАРІВ**

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Магістерська кваліфікаційна робота

На тему: «Розробка методів і програмних засобів для підвищення ефективності прогнозування продажів товарів»

Автор: ст. Групи Іпі-21м Сікорська М.О.

Науковий керівник: к.т.н., доц. каф. ПЗ Романюк О.В.

Рисунок Г.1 – Титульний слайд

Мета, об'єкт та предмет дослідження

- ▶ **Мета роботи** – підвищення ефективності прогнозування продажів товарів за рахунок автоматизованого підбору оптимального методу прогнозування, що відповідає наявним вхідним даним про продажі товару за минулі періоди.
- ▶ **Об'єкт дослідження** – процес прогнозування продажів товарів в програмних додатках для бізнесу.
- ▶ **Предмет дослідження** – методи та засоби підвищення ефективності прогнозування продажів товарів.

Рисунок Г.2 – Мета, об'єкт та предмет дослідження

Задачі дослідження:

- ▶ провести аналіз існуючих методів прогнозування продажів;
- ▶ запропонувати новий адаптивний метод прогнозування продажу товарів;
- ▶ розробити метод автоматизованого формування звіту прогнозування;
- ▶ розробити базу даних;
- ▶ розробити графічний інтерфейс програмного засобу;
- ▶ розробити програмні компоненти на основі запропонованих методів;
- ▶ провести тестування розроблених програмних засобів.

Рисунок Г.3 – Задачі дослідження

Наукова новизна:

- ▶ Уперше запропоновано адаптивний метод прогнозування продажу товарів, особливість якого полягає в автоматизованому аналізі вхідних даних про продажі за минулі періоди та підборі найбільш оптимального методу прогнозування, що дозволило підвищити ефективність прогнозу.
- ▶ Уперше запропоновано метод автоматизованого формування звіту прогнозування, який дозволяє фіксувати макроекономічні показники, що дозволило підвищити ефективність оцінювання результатів прогнозування майбутніх обсягів збуту товарів.

Рисунок Г.4 – Наукова новизна

Порівняльний аналіз аналогів:

Критерій	Infor CloudSuite	NetSuite	Aparlan	Власний додаток
Зручний інтерфейс	0	0	0	1
Безкоштовний пробний період	0	0	1	1
Легке навчання для використання	0	0	0	1
Можливість прогнозування продажу	1	1	1	1
Сума	1	2	2	4

Рисунок Г.5 – Аналіз аналогів

Адаптивний метод прогнозування продажів товарів.

Питання \ Метод	Арифм. згладж.	Метод Хольта Вінгера	Експоненційне згладжування	Правило 1.5	Прогнозування для сезонності
Чи залежить продаж товару від сезону?	-	+	-	-	+
Ви давно працюєте з цим товаром?	-	+	+	-	+
Скільки попередніх періодів?	Немає різниці	Дуже багато	середня	середня	Певного сезону
Необхідний тренд?	-	+	+	-	-
На який період буде проведено прогнозу	наступний	Середній, довгостроковий	Наступний, середній	Наступний	Наступний, середній, довгостроковий

Вибір методу прогнозування
 Дайте відповіді на поставлені запитання і ми підберемо для вас метод.

Крок 6: Вибір методу прогнозування

1. Чи залежить продаж товару від сезону? Так Ні

2. Ви давно працюєте з цим товаром? Так Ні

3. Скільки в минулому періоді було товару? Мало Не багато Багато

4. Необхідний тренд? Так Ні

5. На який період буде проведено прогнозу Наступний Середній Довгий

На основі обраних методів, рекомендуємо:

Рисунок Г.6 – Адаптивний метод прогнозування продажів товарів

Метод автоматизованого формування звіту з прогнозування продажів.

▶ Крок 1.

Вводяться дані про бізнес (назва, контакти, важливі посилання, тип продукції, скільки товар вже на ринку). Так як компанія може продавати різну продукцію, а прогнозування може робитися для кожного виду товару окремо, важливо все це зазначити у звітах.

▶ Крок 2.

На екрані з'явиться інформація про деякі макроекономічні індикатори, такі як ВВП та рівень інфляції. Інформація буде оновлюватись автоматично. Актуальна інформація буде збережена у звіті, в майбутньому її можна буде порівняти.

▶ Крок 3.

Оцінка рентабельності продукту. Якщо це новий бізнес або продажі стрімко падають, потрібно розрахувати чи підходить обрана стратегія, можливо потрібно внести якісь зміни.

▶ Крок 4.

Встановлення цілей. Цей крок збільшує мотивацію та дає насаги працювати плідно.

▶ Крок 5.

На цьому кроці виносяться основні конкуренти. В майбутньому можна буде оцінити зміни.

▶ Крок 6.

Вибір методу прогнозування. Користувач може обрати метод самостійно, або за допомогою системи. На цьому кроці відбувається саме прогнозування.

▶ Крок 8.

Формується звіт. Зберігається в базу даних. Його можна надрукувати.

Рисунок Г.7 – Метод автоматизованого формування звіту з прогнозування продажів



Рисунок Г.8 – Блок-схема алгоритму формування звіту



Блок-схема алгоритму авторизації

Рисунок Г.9 – Блок-схема алгоритму авторизації



Загальна блок-схема алгоритму роботи програми

Рисунок Г.10 – Загальна блок-схема алгоритму роботи програми

Тестування функціональності

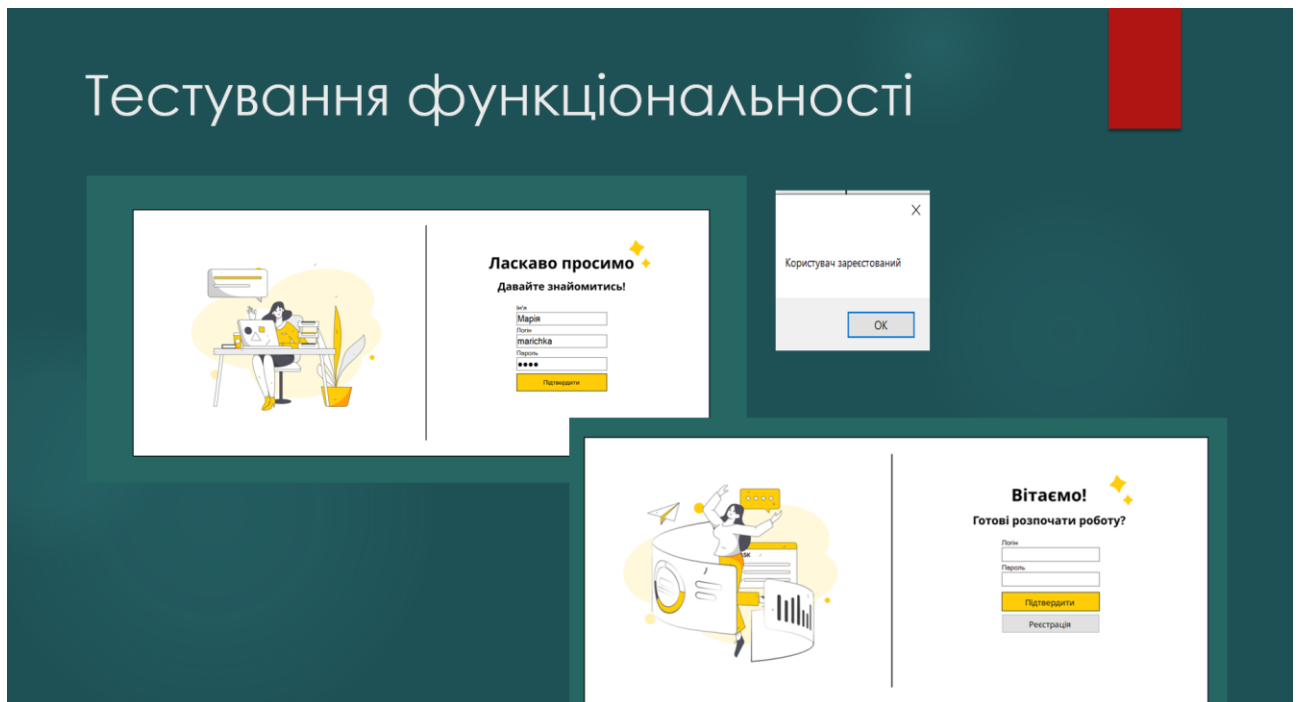


Рисунок Г.11 – Тестування функціональності

Тестування функціональності

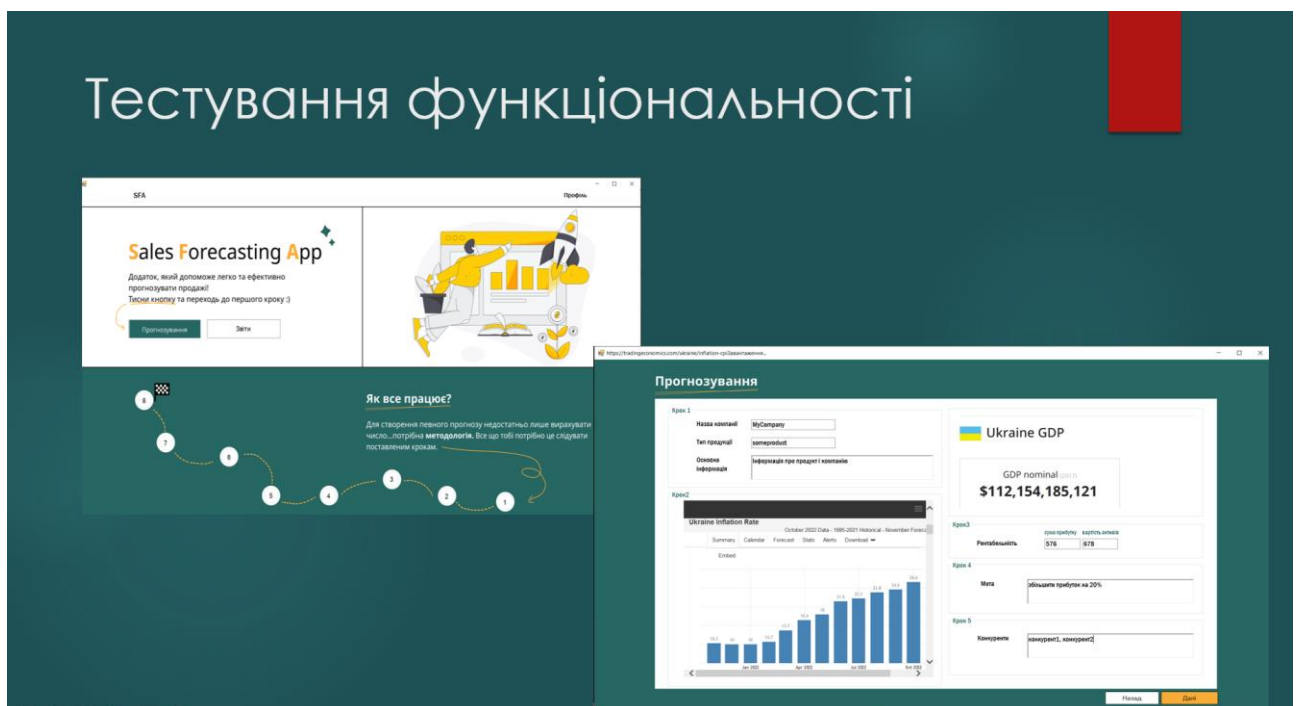


Рисунок Г.12 – Тестування функціональності

Тестування функціональності

Вибір методу прогнозування

Крок 8. Вибір методу прогнозування

1. Чи зменшить продаж товару від сезону? Так Ні

2. Чи дають продажі і ціни тенденції? Так Ні

3. Сильні в минулих періодах збуди товару? Так Ні Немає

4. Невідомий тренд? Так Ні

5. На який період буде проведено прогнозу? Короткий

Експоненційне згладжування

Опис методу: Експоненційне згладжування забезпечує точніше узгодження про тренд. І для знову робити короткостроковий прогноз. Коefіцієнт згладжування α - це значення від 0 до 1.

Чи це працює?

- Вам потрібно ввести і тем. продажі за попередні періоди
- Дані потрібно вводити з нового рядка.
- Потім натисніть кнопку розрахувати.
- Готово!

Дані для тестування

123
234
345
246

Результат: 405,875

Експоненційне згладжування

Графік показує тенденцію зростання даних за періоди 1-6.

Вітаємо! Ось Ваш звіт:

Дата: 2023

Інформація про компанію

Інфляція: 24%

ВВП: 200,1 млрд. USD

Рентабельність: 20%

Мета: збільшити продажі на 20%

Конкуренти: Конкурент 1, Конкурент 2

Прогноз методом Експоненційне згладжування: 405,875

Рисунок Г.13 – Тестування функціональності

Економічна частина

- ▶ Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР:

$$ЗВ = \frac{59978}{0,9} = 66\,642,23 \text{ грн}$$

- ▶ Термін окупності:

$$T_{ок} = \frac{1}{0,6} = 1,6 \text{ (роки)}.$$

Рисунок Г.14 – Економічна частина

Публікація та апробація матеріалів магістерської кваліфікаційної роботи

Основні результати досліджень опубліковано в 2 наукових працях, у матеріалах конференцій.

Результати магістерської кваліфікаційної роботи доповідалися та обговорювалися на:

- ▶ IV міжнародна наукова конференція науковий простір: актуальні питання, досягнення та інновації (Івано-Франківськ 2022);
- ▶ Міжнародна науково-практична Інтернет-конференція «Електронні інформаційні ресурси: створення, використання, доступ» (Вінниця 2022).



Рисунок Г.15 – Публікація та апробація матеріалів магістерської кваліфікаційної роботи

Висновки

- ▶ Проведено аналіз існуючих методів прогнозування продажів;
- ▶ Розроблено метод автоматизованого формування звіту;
- ▶ розроблено адаптивний метод прогнозування продажів;
- ▶ розроблено модулі для прогнозування продажів товарів;
- ▶ розроблено базу даних;
- ▶ розроблено графічний інтерфейс програмного засобу;
- ▶ проведено тестування програмного продукту.

Рисунок Г.16 – Висновки



Дякую за увагу!

Рисунок Г.17 – Завершальний слайд