

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка програмного забезпечення для автоматизації роботи

менеджера магазину побутової техніки "Сатурн"

Виконав: студент групи 3157ст

_____ Алієв А.Ш.

(підпис, ПІБ)

Керівник роботи:

доцент кафедри. ПЗАС, к.т.н., доцент

(посада, науковий ступень вчене звання)

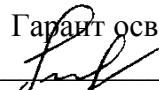
_____ Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми

_____ доц. Макарова Л.М.
(підпис)
«___» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Алієву Анару Шихалі огли _____
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" _____

Керівник роботи Макарова Л.М. _____

Затверджені наказом ректора №257-уч від «17» березня 2023 р.

2. Термін подання роботи: 29.05.2023 р. _____

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів _____

Титульний слайд, Мета та завдання кваліфікаційної роботи, Аналіз вимог до ПЗ, Діаграма варіантів використання, Сценарії варіантів використання, Концептуальна модель даних, Діаграма класів, Діаграми взаємодії, Логічна модель бази даних, Технічний проект ПЗ, Результати розробки, Висновки, Заключний слайд. _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 20.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	24.03.2023	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	31.03.2023	ПП*
3.	Аналіз вимог до ПЗ	07.04.2023	ПП*
4.	Розробка архітектури ПЗ	21.04.2023	
5.	Розробка проекту ПЗ	05.05.2023	
6.	Реалізація та тестування ПЗ	12.05.2023	
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023	
8.	Підготовка розділу з охорони праці	19.05.2023	ПП*
9.	Підготовка розділу ВИСНОВКИ	24.05.2023	
10.	Оформлення списку використаних джерел та додатків	26.05.2023	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	29.05.2023	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	02.06.2023	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	05.06.2023	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	12.06.2023	

* - за результатами переддипломної практики (ПП), яка була з 01.02.2023 до 19.03.2023 р.

Студент

(підпис)

Алієв А.Ш.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов – розробці програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", що дозволить спростити роботу менеджера магазину побутової техніки та полегшити ведення обліку магазину.

У кваліфікаційній роботі представлені аналіз практичної задачі інженерії програмного забезпечення, аналіз існуючих аналогів та постановка задачі на розробку програмного забезпечення. Виконано аналіз вимог до програмного забезпечення. Розроблено ескізний, технічний та робочий проекти програмного забезпечення, представлено результати розробки та розділ з охорони праці.

Кваліфікаційна робота викладена на 76 сторінках машинописного тексту та містить: 13 рисунків, 23 таблиці, 5 додатків, список використаних джерел з 16 найменувань.

ABSTRACT

The qualification work is dedicated to solving the practical task of software engineering, characterized by complexity and uncertainty of conditions – Development of the software for automated manager workplace of a appliance store "Saturn", that will simplify the work of the manager of a appliance store and make it easier to keep a store accounting.

In qualification work presents analysis of the practical task of software engineering on the topic of qualification work, analysis of existing analogues and formulation of the problem for software development. Analysis of software requirements has been completed. The sketch, technical and working projects of the software, results of development and the section on labor protection are developed.

The qualification work is presented on 76 pages of typewritten text and contains: 13 figures, 23 tables, 5 appendices, list of references from 16 names.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення	9
1.2 Аналіз існуючих програмних продуктів	12
1.3 Постановка задачі на розробку програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"	14
2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"	15
2.1 Аналіз вимог до програмного забезпечення	15
2.1.1 Побудова моделі варіантів використання	15
2.1.2 Специфікація варіантів використання	16
2.2 Проект програмного забезпечення	22
2.2.1 Ескізний проект програмного забезпечення	22
2.2.2 Технічний проект програмного забезпечення	28
2.2.3 Робочий проект програмного забезпечення	34
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"	41
4 ОХОРОНА ПРАЦІ	45
4.1 Аналіз небезпечних і шкідливих факторів у відділі розробки програмного забезпечення	45
4.2 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів у відділі розробки програмного забезпечення	46

ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
Додаток А Технічне завдання	52
Додаток Б Текст програми	59
Додаток В Опис програми	70
Додаток Г Інструкція користувача	72
Додаток Д Програма та методика випробування програмного забезпечення	75

ВСТУП

Зараз вже важко знайти магазини, в яких працівники мали б тільки рахівниці та лотки для грошей. Сучасні магазини обладнані комп'ютерами, контрольно-касовими апаратами, електронними вагами, POS-терміналами, сканерами штрих-коду та іншими сучасними пристроями.

Облік діяльності магазину зазвичай ведеться за допомогою програмного забезпечення, яке дозволяє уникнути трудомісткої ручної роботи, зменшити кількість помилок та оперативно отримувати необхідні дані про продажі, стан складу і таке інше. Автоматизація торгівлі є одним з важливих чинників, що впливають на успіх торгового бізнесу. Як показує практика, в результаті автоматизації обліку знижується вірогідність помилок персоналу, підвищується продуктивність праці, зменшуються витрати підприємства і, як наслідок, збільшується прибуток.

Сфера магазинів побутової техніки характеризується високим рівнем конкуренції. І навіть на сьогодні цей ринок є одним з динамічних ринків, не зважаючи на пандемію та військовий стан в Україні. Згідно з даними дослідницької компанії GfK-Україна, в Україні дещо знизилися обсяги продажу побутової техніки та змінилися пріоритети покупців [1]. Зараз робота магазинів побутової техніки та електроніки залежить в значній мірі і від локації магазину. Оператори цього ринку також відзначають, що в цей час магазини стріт-рїтейл працюють краще, ніж великі магазини у структурі торговельних центрів [2]. Ритейлери також прогнозують дефіцит недорогої техніки та електроніки [3].

Виходячи з цього, можна сказати, що розробка програмного забезпечення для автоматизації діяльності невеликих торговельних фірм та магазинів є актуальною.

Метою кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробка програмного забезпечення автоматизації

роботи менеджера магазину побутової техніки "Сатурн", для чого потрібно, в першу чергу, виконати аналіз практичної задачі інженерії програмного забезпечення, а саме аналіз роботи магазину побутової техніки "Сатурн", зокрема, аналіз сучасних тенденцій автоматизації його діяльності, а також доцільність розробки власного програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки.

Програмне забезпечення, що розробляється, призначено для спрощення роботи менеджера магазину побутової техніки "Сатурн" та полегшення ведення обліку магазину.

Для досягнення поставленої мети необхідно:

- проаналізувати роботу магазину побутової техніки "Сатурн", зокрема, сучасні тенденції автоматизації його діяльності;
- зробити висновок про доцільність розробки власного програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн";
- здійснити постановку задачі та розробити технічне завдання на розробку програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн";
- провести аналіз вимог до програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн";
- розробити проєкт програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", який включає ескізний технічний та робочий проєкти цього програмного забезпечення;
- виконати реалізацію та тестування програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн".

Об'єктом кваліфікаційної роботи є процес розробки програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн".

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"

1.1 Аналіз практичної задачі інженерії програмного забезпечення

Починати розробку програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" потрібно з аналізу практичної задачі інженерії програмного забезпечення, включаючи огляд тих задач, які потрібно буде вирішити в процесі роботи та формулювання вимог до майбутнього програмного забезпечення.

Розробка програмного забезпечення для автоматизації роботи підприємств різних сфер діяльності є однією з важливих практичних задач інженерії програмного забезпечення. Підприємства торгівельної сфери не є виключенням. І як вже було відмічено у вступі, автоматизації діяльності невеликих магазинів побутової техніки є актуальною задачею.

Якщо розглядати сегмент продажу побутової техніки, то можна відмітити, що вона продається на декількох торгівельних майданчиках. Це, в першу чергу, так звані «відкриті ринки», тобто невеличкі торгівельні павільйони, для яких характерні наявність вузького асортименту та переважно нижній ціновий діапазон. Далі йдуть мультибрендові магазини, які представляють вже ширший спектр продукції та орієнтовані середній ціновий діапазон. Третій формат – це гіпермаркети електроніки, як правило, мережеві, наприклад, «Фокстрот», «Comfy», для яких характерний найширший асортимент товару та орієнтація на всі цінові діапазони.

В якості об'єкту автоматизації розглядається робота мультибрендового магазину побутової техніки низького цінового діапазону, який відноситься до стріт-рітейл.

Стріт-рітейл – одне з найпопулярніших форматів торгових приміщень роздрібної торгівлі, які, зазвичай, прагнуть взяти в оренду як власники малого бізнесу, так і великих підприємств через ряд переваг. Відмітною особливістю сучасних стріт-рітейлів та їх головною перевагою є розташування на першому поверсі будівлі та наявність окремого входу. Стріт-рітейл також користується великою популярністю насамперед завдяки "кроковій доступності" та передбачає постійний потік покупців через зручне розташування на жвавій вулиці.

Магазин, який розглядається в якості об'єкту автоматизації, розташований на одній із жвавих вулиць з великим потоком потенціальних покупців неподалік від центрального ринку обласного центру. Магазин складається з декількох приміщень (торгівельний зал, склад, підсобні приміщення). З магазином співпрацюють постачальники, які здійснюють постачання товару на склад магазину.

В магазині є свій штат співробітників: директор, бухгалтер, менеджери з продажу товарів, комірник, вантажник. Також можуть працювати експедитор та водій при необхідності доставки великогабаритної техніки, що додатково обговорюється при продажі товару.

Обов'язки в магазині розподілені наступним чином. Директор здійснює загальне керівництво діяльністю магазину. Бухгалтер займається веденням бухгалтерської звітності, складанням бухгалтерського балансу та податкової звітності, нарахуванням заробітної плати і таке інше.

Менеджер з продажу товарів працює у торговельному залі магазину. До його обов'язків належить: формування заявок на поставку товарів, консультація покупців по товарам та їх характеристикам, демонстрація товарів, оформлення продажу товарів, в тому числі відбиття чеку та його видача, операції з грошовими коштами, формування звітів.

Комірник слідкує за наявністю товару, відповідає за прийом товару на склад магазину та видачу товару зі складу магазину. Відстежує роботу вантажника, експедитора та водія.

Покупець, приходячи в магазин, взаємодіє з менеджером з продажу товарів, який консультує клієнта за характеристиками та вартістю товарів, демонструє товари, оформлює продаж товарів та гарантійних документів.

При купівлі товарів покупець отримує чек із зазначенням товарів та їх вартістю. Якщо якийсь товар відсутній у торгівельному залі магазину, покупець звертається з чеком до складу, на підставі чого йому відвантажують необхідні товари. Покупець може самостійно забрати товари зі складу або замовити їх доставку.

Крім того, менеджер з продажу товарів відповідає за формування наступних звітів: зведена інформація з усіх товарів, проданих за певний період (кожен день, щотижня, помісячно, поквартально, раз на півроку, раз на рік), зведена інформація по типам товару, зведена інформація про кількість замовлених товарів и таке інше.

Розглянувши процес формування звітів можна виявити ряд проблем, пов'язаних з ручним заповненням:

- внесення нових даних може нести за собою помилки;
- при заповненні нового звіту необхідно заповнювати всі його дані заново, втрачається час на внесення однотипної інформації;
- може бути присутня паперова тяганина;
- можлива втрата звіту, інформації про продаж товару і таке інше.

З вище зазначеного можна зробити висновок, що проблема, пов'язана з неавтоматизованим обліком, не є новою. Отже, є нагальна необхідність автоматизувати роботу менеджера магазину побутової техніки "Сатурн" за допомогою впровадження відповідного програмного забезпечення, яке дозволить автоматизувати заявки на постачання товарів, оформлення продажу товарів, формування звітів, що значно зменшить можливість помилок при підготовці звітів та загальне навантаження на менеджера магазину побутової техніки "Сатурн".

1.2 Аналіз існуючих програмних рішень

Гарним рішення з точки зору функціоналу для автоматизації роботи як менеджерів з продажу, так і інших співробітників торгівельних фірм різного розміру та форм власності були програми сімейства "1С", але вони були заборонені для використання ще з 2017 року. На сьогоднішній день існує кілька українських програмних рішень для автоматизації роботи менеджера з продажу товарів. Розглянемо більш детально деякі з них.

Одним з таких рішень є «MASTER:Бухгалтерія» від компанії IT-Enterprise – програмне забезпечення, яке дозволяє вести та повністю автоматизувати різні види обліку, зокрема:

- основних засобів;
- ТМЦ;
- витрат за елементами;
- розрахунків з іншими компаніями;
- кредиторської та дебіторської заборгованостей;
- податкового обліку, тощо [4].

Це програмне рішення підходить в тому числі і для приватних компаній різного розміру та дозволяє вести облік виробництва, контрактно-договірної обліку, автотранспорту, лікарських засобів, харчових продуктів і таке інше.

Також як перевагу цього програмного забезпечення можна відмітити і можливість використовувати інтегрований сервіс електронного документообігу Signy, оскільки обидві програми належать одному розробнику.

Але як недолік для автоматизації роботи менеджера магазину побутової техніки "Сатурн" можна відмітити більшу орієнтованість цього програмного забезпечення на великий бізнес та деяку "надлишковість" у тих функціях, які реалізовані в цьому програмному забезпеченні.

Ще одним українським програмним рішенням для ведення бухгалтерської звітності є Dilovod – базове рішення для бізнесу, яке не потребує докладних знань бухгалтерського обліку від одноіменної фірми-розробника. Це український сервіс, в якому можна вести облік та подавати звіти. Його можуть використовувати фізичні особи – підприємці, юридичні особи, виробництва, кафе і ресторани. Він орієнтований на роботу магазинів одягу, магазинів продуктів. Є спеціалізація також і для інтернет-магазинів [5].

В цій програмі можна вести такі види обліку:

- замовлень;
- продаж;
- закупівель;
- витрат виробництва тощо.

Це програмне рішення має в своєму складі програмний реєстратор розрахункових операцій та CRM-систему.

Але основним недоліком цього програмного продукту з точки зору автоматизації роботи менеджера магазину побутової техніки "Сатурн" є те, що Dilovod – повністю онлайн-бухгалтерія, тобто взагалі не має в своєму складі оффлайн-версії програмного забезпечення.

Отже, необхідність розробки власного програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", яке буде задовольняти потреби саме цього замовника, є актуальною. Розробка даного програмного забезпечення допоможе підвищити ефективність роботи менеджера магазину побутової техніки "Сатурн", а саме: оперативне отримання актуальної інформації про наявність товару, зменшення часу обслуговування клієнтів, автоматизацію формування звітів.

1.3 Постановка задачі на розробку програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Виходячи з попередньо проведеного аналізу практичної задачі інженерії програмного забезпечення, для підвищення ефективності роботи менеджера магазину побутової техніки "Сатурн", було прийнято рішення розробити власне програмне забезпечення, призначенням якого є спрощення роботи менеджера магазину побутової техніки "Сатурн" та полегшення ведення обліку магазину. Користувачем даного програмного забезпечення є менеджер магазину побутової техніки "Сатурн".

Дане програмне забезпечення повинно вирішувати наступні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;
- зберігання та використання інформації з продажу товару;
- складання звітності.

Було виділено вимоги до складу та параметрів технічних засобів. Система повинна задовольняти наступним вимогам в мінімальній комплектації:

- CPU: Intel(R) i3;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

Також є вимоги до інформаційної та програмної сумісності. Для повноцінного функціонування системи необхідно використовувати операційну систему Windows версії не нижче 7. Необхідна наявність встановленого Standard Edition Development Kit 1.8 (JDK 1.8).

Детальна постановка задачі представлена у технічному завданні, яке наведено у додатку А.

2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"

2.1 Аналіз вимог до програмного забезпечення

2.1.1 Побудова моделі варіантів використання

Діаграма варіантів використання (Use Case Diagram) – це UML діаграма за допомогою якої в графічному вигляді можна представити вимоги до програмного забезпечення, що розробляється. Діаграма варіантів використання – це вихідна концептуальна модель проекрованої системи.

Діаграма варіантів використання складається з ряду елементів. Основними елементами є: варіанти використання або прецеденти, актор або дійова особа та відносини між ними.

Варіант використання відображає сервіси, які програмне забезпечення надає актору.

Відношення слугують семантичним зв'язок між окремими елементами моделі.

Один актор може взаємодіяти з різними варіантами використання, а не тільки з одним. Тоді актор звертається до кількох сервісів програмного забезпечення. Так само і один варіант використання може взаємодіяти з різними акторами, надаючи для всіх них свій сервіс.

Існують такі відносини: асоціації, узагальнення, залежність (складається з включення та розширення) [6].

Діаграма варіантів використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" наведена на рис. 2.1.

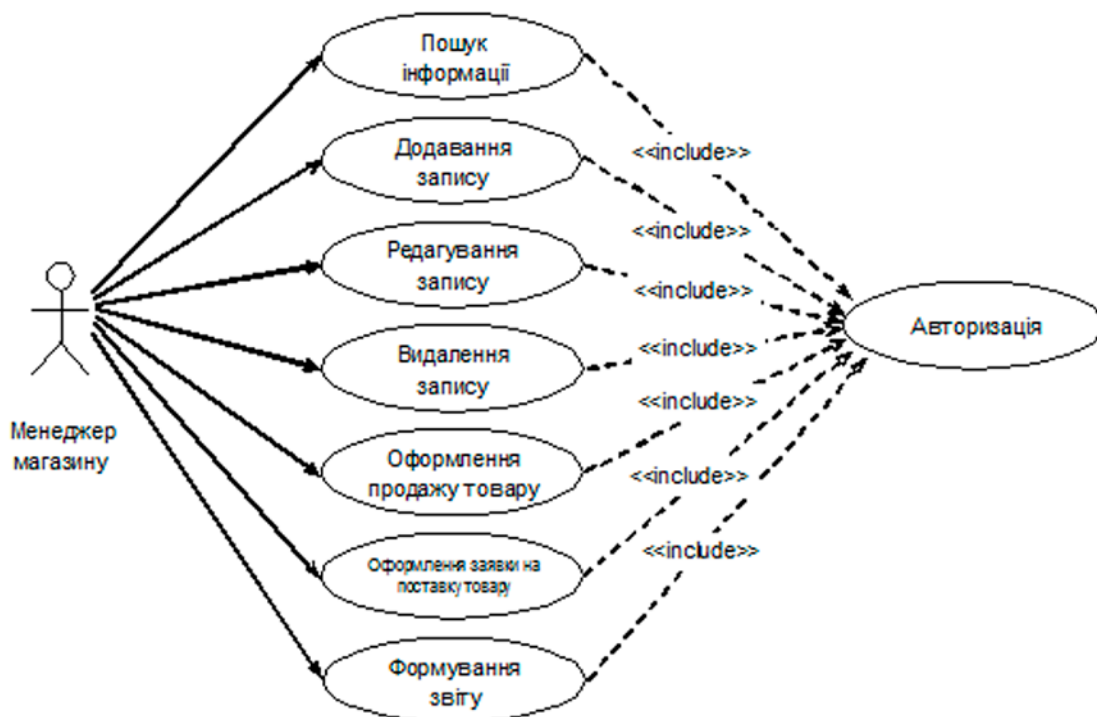


Рисунок 2.1 – Діаграма варіантів використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

2.1.2 Специфікація варіантів використання

Розглянемо більш детально кожний варіант використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн". Потік подій кожного варіанту використання складається з:

- короткого опису;
- передумов;
- основного потоку подій;
- альтернативного потоку подій;
- постумов.

Детальний опис цих складових частин для кожного варіанту використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" наведений у таблицях 2.1 – 2.8.

Таблиця 2.1 – Опис варіанту використання "Авторизація"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу пройти перевірку для доступу до системи.
Передумови	Система запущена та виводить форму для авторизації.
Основний потік подій	1. Система виводить форму для вводу логіну та паролю. 2. Користувач вводить логін та пароль і підтверджує ввід. 3. Система перевіряє вірність логіну та паролю. 4. Якщо дані достовірні, то завантажується головне вікно програми.
Альтернативний потік подій	1. Система виводить форму для вводу логіну та паролю. 2. Користувач вводить логін та пароль і підтверджує ввід. 3. Система перевіряє вірність логіну та паролю. 4. Якщо дані недостовірні, повідомлення про те, що логін або пароль не вірні. 5. Надання можливості повторного вводу логіну та паролю.
Постумови	Користувач авторизований в системі.

Таблиця 2.2 – Опис варіанту використання "Пошук інформації (перегляд записів)"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу здійснювати пошук необхідної інформації.
Передумови	Користувач авторизований у системі.

Продовження таблиці 2.2

1	2
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю по якій буде відбуватись пошук необхідної інформації. 4. Користувач вводить необхідні дані для пошуку та натискає кнопку «Знайти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю по якій буде відбуватись пошук необхідної інформації. 4. Користувач вводить необхідні дані для пошуку та натискає кнопку «Відміна».
Постумови	Вивід знайденої інформації.

Таблиця 2.3 – Опис варіанту використання "Додавання запису"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу додати новий запис до таблиці
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Для додавання нового запису у таблицю, користувач натискає кнопку «Додати». З'являється відповідна панель, де вносяться дані. 5. Збереження внесених даних за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Для додавання нового запису у таблицю, користувач натискає кнопку «Додати». З'являється відповідна панель, де вносяться дані. 5. Для відміни запису даних, користувач натискає кнопку «Відміна».
Постумови	Додано новий запис до таблиці.

Таблиця 2.4 – Опис варіанту використання "Редагування запису"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу відредагувати вже існуючий запис у таблиці.
Передумови	Користувач авторизований у системі.
1	2
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає та відкриває таблицю. 4. Далі робота відбувається на панелі «Дані таблиці». 5. Для редагування вже існуючого запису, користувач обирає запис та натискає кнопку «Редагувати». З'являється відповідна панель, де редагуються дані. 6. Збереження внесених змін за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає та відкриває таблицю. 4. Далі робота відбувається на панелі «Дані таблиці». 5. Для редагування вже існуючого запису, користувач обирає запис та натискає кнопку «Редагувати». З'являється відповідна панель, де редагуються дані. 6. Для відміни запису даних, користувач натискає кнопку «Відміна».
Постумови	Збереження та вивід відредагованого запису.

Таблиця 2.5 – Опис варіанту використання "Редагування запису"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу видалити вже існуючий запис у таблиці.
Передумови	Користувач авторизований у системі.

Продовження таблиці 2.5

1	2
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Натискає кнопку «Видалити» 5. Підтверджує внесені зміни за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Натискає кнопку «Видалити» 5. Для відміни видалення запису, користувач натискає кнопку «Відміна».
Постумови	Запис видалено.

Таблиця 2.6 – Опис варіанту використання "Оформлення продажу товару"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу оформити продаж товару.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю «Продаж» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач обирає клієнта, товар та інші дані для формування продажу і натискає кнопку «Сформувати». 5. З'являється сформована продаж товару, користувач натискає кнопку «Друк».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю «Продаж» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач обирає клієнта, товар та інші дані для формування продажу і натискає кнопку «Відміна».
Постумови	Оформлено новий продаж товару.

Таблиця 2.7 – Опис варіанту використання "Оформлення заявки на поставку товару"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу сформувати заявку на поставку товару.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає таблицю «Товар» та натискає кнопку «Заявка». 4. З'являється відповідне вікно, де користувач обирає товар, постачальника та інші дані для формування заявки на поставку товару і натискає кнопку «Сформувати». 5. З'являється сформована заявка на поставку товару, користувач натискає кнопку «Друк».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає таблицю «Товар» та натискає кнопку «Заявка». 4. З'являється відповідне вікно, де користувач обирає постачальника та інші дані для формування заявки на поставку товару і натискає кнопку «Відміна».
Постумови	Оформлено нову заявку на поставку товару.

Таблиця 2.8 – Опис варіанту використання "Формування звіту"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу сформувати необхідний звіт.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає дію «Звіт». 4. З'являється відповідне вікно, де користувач обирає тип звіту та всі необхідні дані для цього типу звіту і натискає кнопку «Сформувати». 5. З'являється сформований звіт, користувач натискає кнопку «Друк».

продовження таблиці 2.8

1	2
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає дію «Звіт». 4. З'являється відповідне вікно, де користувач обирає тип звіту та всі необхідні дані для цього типу звіту і натискає кнопку «Відміна».
Постумови	Сформовано для друку новий звіт

Таким чином, використовуючи детальні специфікації варіантів використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", можемо приступати до подальшого проєктування вказаного програмного забезпечення.

2.2 Проєкт програмного забезпечення

2.2.1 Ескізний проєкт програмного забезпечення

Сценарії основних варіантів використання

Кожен об'єкт програмного забезпечення може володіти своєю поведінкою, може знаходитися в деяких станах, переходити з одного в інший стан та в процесі реалізації свого сценарію поведінки здійснювати певні дії. Поведінка об'єкта відображається його станами і діаграми діяльності як раз і дозволяють це відобразити графічно [7].

Сценарії основних варіантів використання наведено на рисунках 2.2 – 2.4.

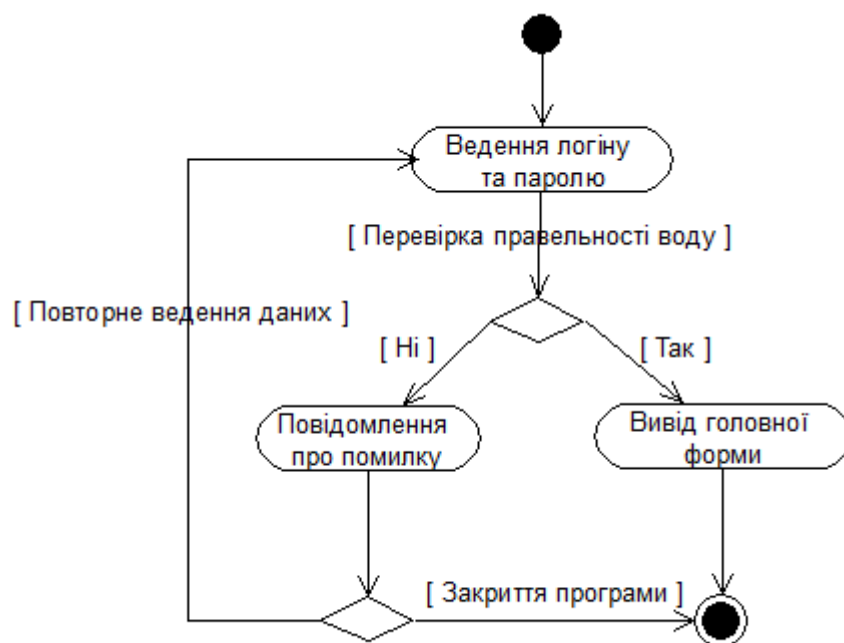


Рисунок 2.2 – Діаграма діяльності для прецеденту «Авторизація»

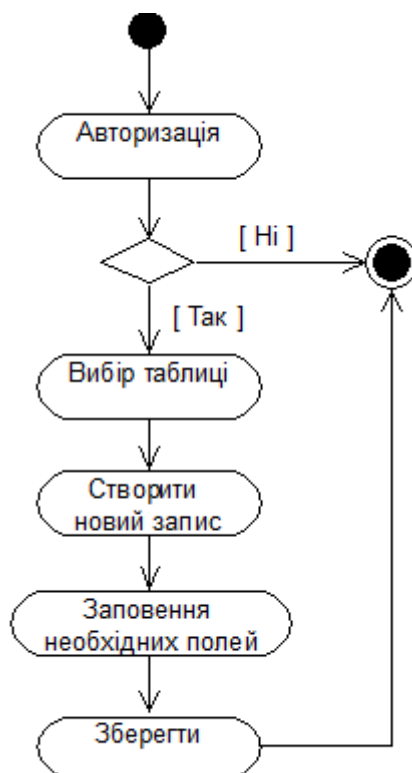


Рисунок 2.3 – Діаграма діяльності для прецеденту «Додавання запису»

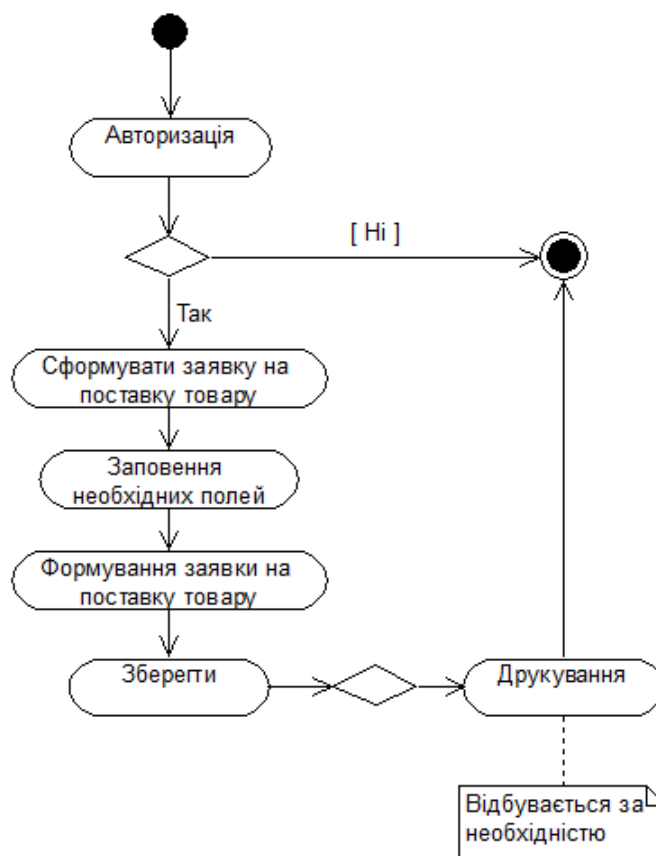


Рисунок 2.4 – Діаграма діяльності для прецеденту «Формування заявки на поставку»

Розробка концептуальної моделі

На основі аналізу предметної галузі та постановки задачі було розроблено концептуальну модель програмного забезпечення – модель, яка відображає структуру програмного забезпечення, властивості його елементів та зв'язки між ними, притаманні програмі та суттєві для розуміння основної мети.

Модель «сутність – зв'язок» є концептуальною моделлю, яка не враховує особливості конкретної системи управління базами даних. Найбільше поширення отримала ER – модель (Entity – relationship model), запропонована Пітером Ченом у 1976 р.

Процес створення моделі «сутність – зв'язок» є ітераційним. Розробивши перший наближений варіант моделі, можна уточнювати її, спираючись на знання відповідних експертів. Документацією, де фіксуються результати бесід при цьому, є сама модель «сутність – зв'язок» [8]. Концептуальна модель

програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" наведена на рис. 2.5.

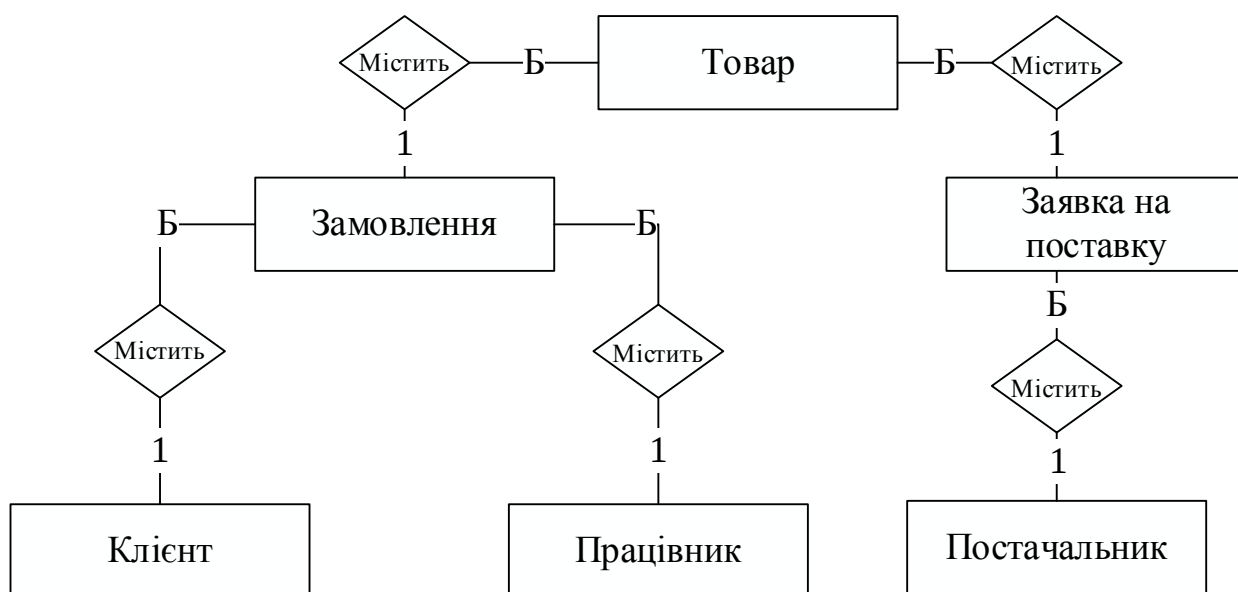


Рисунок 2.5 – Концептуальна модель програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Розглянемо більш детально концептуальну модель програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" та виділимо атрибути кожної сутності.

Сутність «Працівник» має наступні атрибути:

- Код працівника;
- ПІБ працівника;
- Посада;
- Адреса;
- Телефон;
- Дата прийняття;
- Дата звільнення.

Сутність «Постачальник» має наступні атрибути:

- Код постачальника;
- Фірма;

- Представник;
- Телефон;
- Адреса.

Сутність «Клієнт» має наступні атрибути:

- Код клієнта;
- ПІБ клієнта;
- Адреса;
- Телефон.

Сутність «Заявка на поставку» має наступні атрибути:

- Код заявки;
- Код постачальника;
- Тип товару;
- Найменування товару;
- Кількість;
- Дата формування заявки.

Сутність «Товар» має наступні атрибути:

- Код товару;
- Код постачальника;
- Тип товару;
- Найменування товару;
- Кількість;
- Ціна закупки;
- Ціна продажу.

Сутність «Замовлення» має наступні атрибути:

- Код замовлення;
- Код працівника;
- Код клієнта;
- Найменування товару;
- Кількість;
- Сума продажу;

– Дата продажу.

Розробка прототипу інтерфейсу користувача

Виходячи з опису діаграми варіантів використання програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", було розроблено інтерфейс користувача даного програмного забезпечення. Інтерфейс користувача – це компоненти та елементи вихідних форм програмного забезпечення, які впливають на ефективну взаємодію між користувачем та програмним забезпеченням [9]. Розроблений ескіз графічного інтерфейсу програмного забезпечення наведено на рис. 2.6 – 2.8.

Рисунок 2.6 – Форма авторизації програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Рисунок 2.7 – Головна форма програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Рисунок 2.8 – Форма «Заявка на поставку» програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Маючи розроблений ескізний проєкт програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", можемо переходити до наступної стадії проєктування – розробки технічного проєкту програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн".

2.2.2 Технічний проєкт програмного забезпечення

Діаграма класів в об'єктно-орієнтованому моделюванні призначена для представлення статичної структури програмного забезпечення в термінології класів об'єктно-орієнтованого програмування. Вона може відображати різні аспекти взаємозв'язків між різними сутностями предметної галузі, зокрема, об'єктами і підсистемами. Також вона призначена для описує їх внутрішньої структури та відносин між ними [10]. Діаграма класів представляє собою подальший розвиток концептуальної моделі проєктованого програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" та наведена на рис. 2.9.

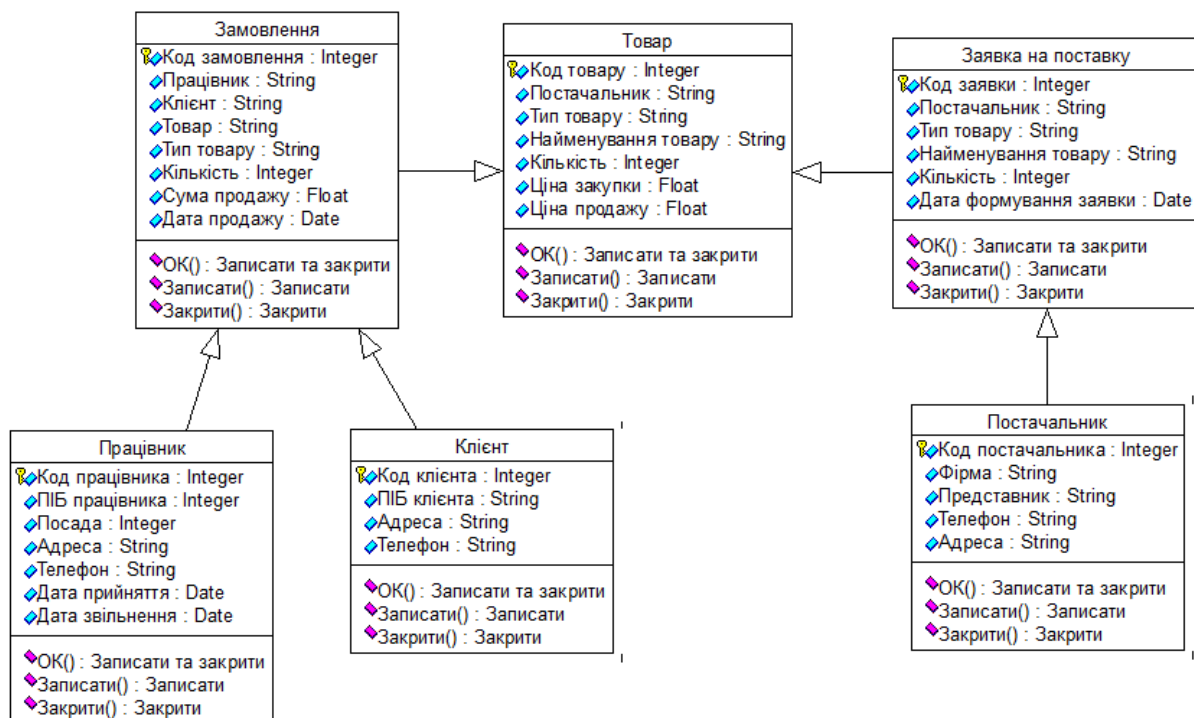


Рисунок 2.9 – Діаграма класів програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Розробимо більш детальну специфікацію класів програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", яка наведена в табл. 2.9 – 2.14.

Таблиця 2.9 – Специфікація класу "Працівник"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по працівникам
Атрибут	Код працівника, ПІБ працівника, посада, адреса, телефон, дата прийняття, дата звільнення
Операції зі специфікаціями нетривіальних операцій	OK(): записати та закрити Записати(): записати Закрити(): закрити

Таблиця 2.10 – Специфікація класу "Постачальник"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по постачальникам товару
Атрибут	Код постачальника, фірма, представник, телефон, адреса
Операції зі специфікаціями нетривіальних операцій	ОК(): записати та закрити Записати(): записати Закрити(): закрити

Таблиця 2.11 – Специфікація класу "Клієнт"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по клієнтам
Атрибут	Код клієнта, ПІБ клієнта, адреса, телефон
Операції зі специфікаціями нетривіальних операцій	ОК(): записати та закрити Записати(): записати Закрити(): закрити

Таблиця 2.12 – Специфікація класу "Заявка на поставку"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по заявкам на поставку товару
Атрибут	Код заявки, код постачальника, тип товару, найменування товару, кількість, дата формування заявки
Операції зі специфікаціями нетривіальних операцій	ОК(): записати та закрити Записати(): записати Закрити(): закрити

Таблиця 2.13 – Специфікація класу "Товар"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по товарам.
Атрибут	Код товару, код постачальника, тип товару, найменування товару, кількість, ціна закупки, ціна продажу
Операції зі специфікаціями нетривіальних операцій	ОК(): записати та закрити Записати(): записати Закрити(): закрити

Таблиця 2.13 – Специфікація класу "Замовлення"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по замовленням
Атрибут	Код замовлення, код працівника, код клієнта, найменування товару, кількість, сума продажу, дата продажу
Операції зі специфікаціями нетривіальних операцій	ОК(): записати та закрити Записати(): записати Закрити(): закрити

Після розробки детальної специфікації класів програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" можна побудувати динамічну модель програмного забезпечення.

Динамічна модель програмного забезпечення

Динамічну модель програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" можна представити у вигляді основних діаграм послідовностей. Діаграми послідовностей можна віднести до діаграм взаємодії, які представляють поведінкові аспекти програмного забезпечення та розглядають взаємодію об'єктів у часі, тобто відображають часові особливості роботи об'єктів. Діаграми послідовностей можна використовувати для уточнення та доповнення діаграм прецедентів та більш детального опису логіки роботи сценаріїв використання [11]. Динамічна модель програмного забезпечення у вигляді основних діаграм послідовностей наведена на рис. 2.10 – 2.12.

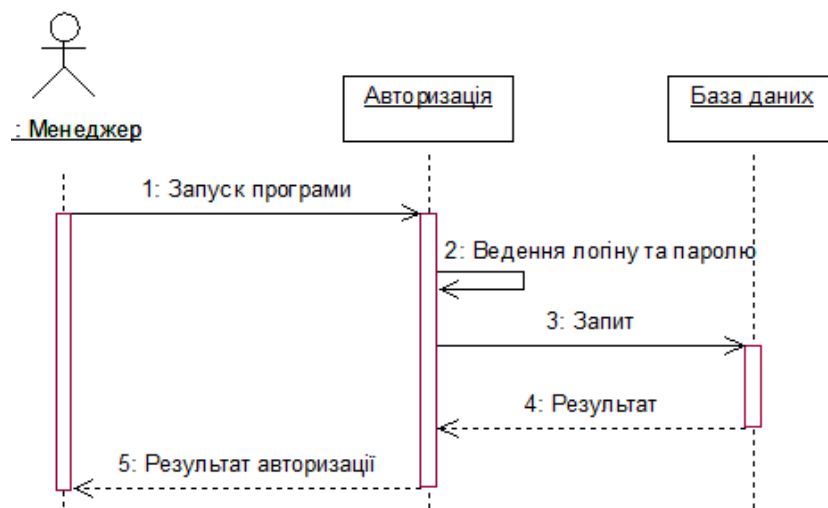


Рисунок 2.10 – Діаграма послідовності «Авторизація»

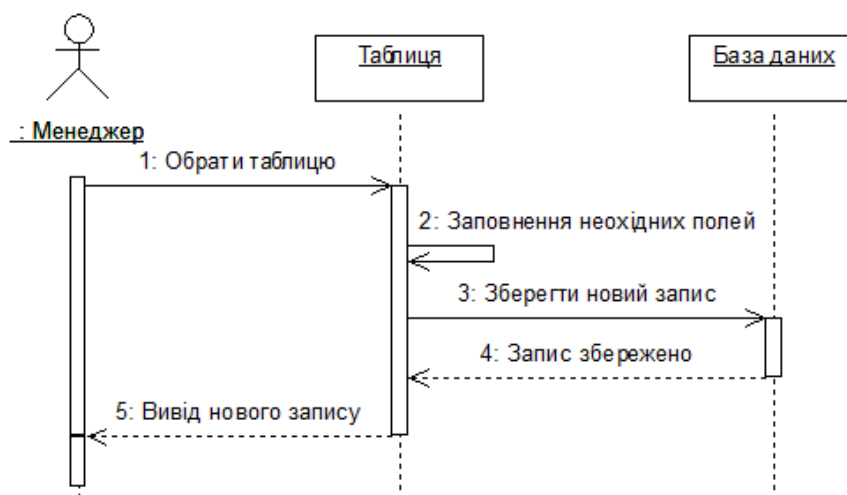


Рисунок 2.11 – Діаграма послідовності «Додавання запису»



Рисунок 2.12 – Діаграма послідовності «Формування заявки на поставку»

Розробка логічної моделі бази даних

Перед тим як безпосередньо розробляти таблиці та інші об'єкти бази даних, необхідно задати відповідну структуру бази даних. Гарна структура бази даних є основою для створення адекватної вимогам, ефективної бази даних. Сам процес проектування бази даних являє собою доволі складний процес.

Для розробленої концептуальної моделі програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" необхідно представити предметну область у вигляді логічної моделі даних, яка може бути використана для розробки програмного забезпечення незалежно від вибору мови програмування та середовища розробки програмного забезпечення.

На основі концептуальної моделі програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", наведеної на рис. 2.5, було створено логічну модель бази даних. Логічна модель даних є початковим прототипом майбутньої бази даних. Логічна модель будується в термінах інформаційних одиниць, але без прив'язки до конкретної системи управління базою даних. Логічна модель бази даних програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" наведена на рис. 2.17.

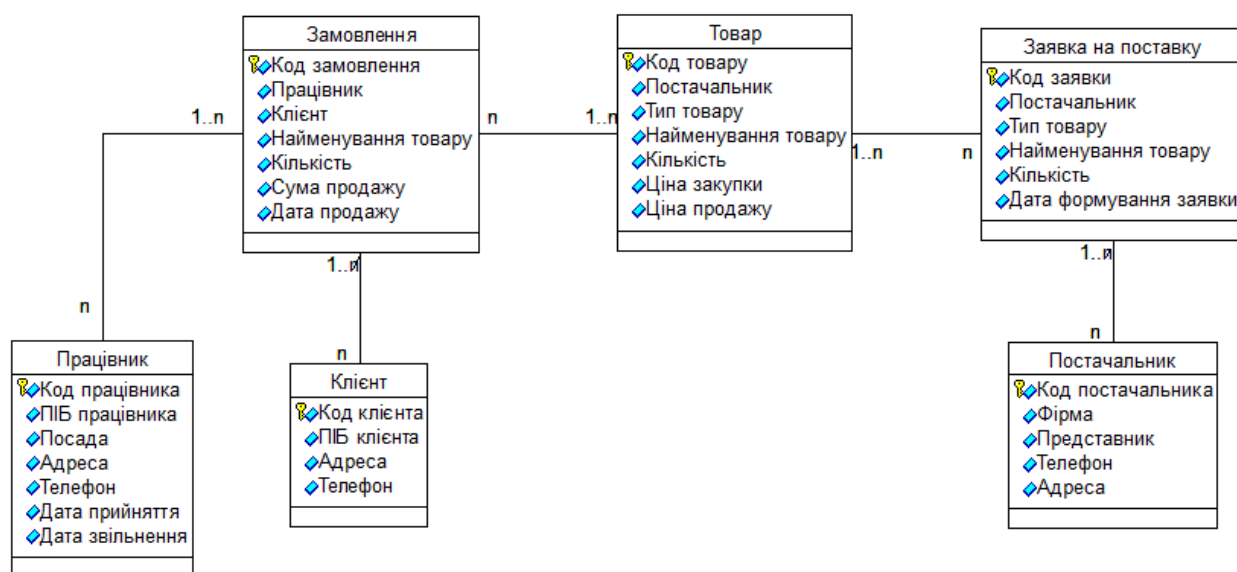


Рисунок 2.17 – Логічна модель бази даних програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Маючи розроблений технічний проєкт програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", можемо переходити до наступної стадії проєктування – розробки робочого проєкту програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн".

2.2.3 Робочий проєкт програмного забезпечення

Вибір мови програмування та СКБД

В більшості випадків вибір мови програмування відіграє чи не вирішальну роль при створенні програмного забезпечення. Деякі мови програмування універсальні та дозволяють вирішувати алгоритмічні задачі будь-якої складності, деякі – спеціалізовані, тобто використовуються для вирішення доволі вузького кола задач. Процес написання програмного коду та сам програмний код повинні бути доволі простим. Обрана мова програмування має полегшувати читання та написання програми.

Після дослідження предметної галузі, огляду способів та засобів вирішення поставлених проблем, аналізу переваг і недоліків існуючих програмних рішень та виходячи із вимог до програмного забезпечення, що наведені у технічному завданні, були вибрані наступні інструменти розробки програмного забезпечення та системи управління базою даних, які є оптимальними для вирішення поставлених задач.

В якості мови програмування була обрана Java – об'єктно-орієнтована мова програмування, що була випущена у 1995 році Sun Microsystems в якості основного компоненту їх платформи Java [12]. Наразі підтримкою та удосконаленням мови займається Oracle, яка придбала у 2009 році Sun Microsystems. Синтаксис мови набагато в чому походить від C/C++. Java-програми компілюються у байт-код, який вже при виконанні інтерпретується

віртуальною Java-машиною для кожної конкретної платформи. Компілятор Java та віртуальна машина Java задовольняють специфікації Java Community Process, та надаються під ліцензією GNU General Public License [13].

В якості системи управління базою даних була обрана MySQL – популярна реляційна система управління базою даних з відкритим кодом, яка також розповсюджується під GNU General Public License або під власною комерційною ліцензією, на вибір.

MySQL є гарним рішенням як для малих додатків, так і для додатків середнього розміру. Звичайно MySQL може використовуватися як сервер, до якого підключаються клієнти, але його також можна включати в автономні програми завдяки бібліотеці внутрішнього сервера, яка входить в дистрибутив [14].

В якості середовища розробки було обрано середовище NetBeans IDE – інтегроване середовище розробки з відкритим вихідним кодом для розробників програмного забезпечення. NetBeans IDE надає розробнику всі необхідні засоби для створення професійних застосунків: корпоративних, мобільних, десктоп- та веб-застосунків на базі таких мов програмування, як Java, C, C++, PHP, JavaScript та деяких інших. Робоча область NetBeans IDE повністю налаштовується під потреби розробника: є можливість налаштування дій, які виконуються за допомогою панелі інструментів, призначення "гарячих" клавіш і т.і.

Присутні додаткові засоби для контекстно-залежного пошуку по всьому середовищу NetBeans IDE, довідкових матеріалах, а також всіх відкритих проектах та файлах. Є можливість почати роботу з проектом з шаблону або створити власний проект у вільному форматі. Для перегляду змін використовується редактор з колірними позначеннями [15].

Фізична модель бази даних

На основі логічної моделі бази даних програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" було

побудовано фізичну модель бази даних програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", яка наведена в таблицях 2.15 – 2.20.

Таблиця 2.15 – Опис структури таблиці «Працівник»

Ім'я поля	Тип	Ключ
Код працівника	int	РК
ПІБ працівника	varchar(50)	
Посада	varchar(50)	
Адреса	varchar(50)	
Телефон	varchar(20)	
Дата прийняття	timestamp	
Дата звільнення	timestamp	

Таблиця 2.16 – Опис структури таблиці «Постачальник»

Ім'я поля	Тип	Ключ
Код постачальника	int	РК
Фірма	varchar(50)	
Представник	varchar(50)	
Телефон	varchar(20)	
Адреса	varchar(50)	

Таблиця 2.17 – Опис структури таблиці «Клієнт»

Ім'я поля	Тип	Ключ
Код клієнта	int	РК
ПІБ клієнта	varchar(50)	
Адреса	varchar(50)	
Телефон	varchar(20)	

Таблиця 2.18 – Опис структури таблиці «Заявка на поставку»

Ім'я поля	Тип	Ключ
Код заявки	int	РК
Код постачальника	int	FK
Тип товару	varchar(50)	
Найменування товару	varchar(50)	
Кількість	int	
Дата формування заявки	timestamp	

Таблиця 2.19 – Опис структури таблиці «Товар»

Ім'я поля	Тип	Ключ
Код товару	int	PK
Код постачальника	int	FK
Тип товару	varchar(50)	
Найменування товару	varchar(50)	
Кількість	int	
Ціна закупки	float	
Ціна продажу	float	

Таблиця 2.20 – Опис структури таблиці «Замовлення»

Ім'я поля	Тип	Ключ
Код замовлення	int	PK
Код працівника	int	FK
Код клієнта	int	FK
Найменування товару	varchar(50)	
Кількість	int	
Сума продажу	float	
Дата продажу	Timestamp	

Реалізація та тестування основних класів еквівалентності програмного забезпечення

Далі проаналізуємо правильні та неправильні класи еквівалентності, проведемо тестування на правильність роботи програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" в різних ситуаціях. Реалізація класів проводиться на основі діаграми класів програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн". Для прикладу наведені тести до класу «Працівник».

Представимо код класу, який тестуватиметься:

```
public static DefaultTableModel
executeSelectQuery(String... q) {
    tableData.clear();
    columnNames.clear();
    CURRENT_TABLE = q[0];
    String query = "Select * from " + CURRENT_TABLE +
```

```

" " + (q.length == 1 ? "" : q[1]);
    System.out.println(query);
    try {
        resultSet =
getStatement().executeQuery(query);
        initColumnNames(resultSet.getMetaData());
        while (resultSet.next()) {
            row.clear();
            for (int i = 0; i < columnNames.size();
i++) {
                row.add(resultSet.getObject(i + 1));
            }
            tableData.add((Vector) row.clone());
        }
        tableModel.setDataVector(tableData,
columnNames);
        return tableModel;
    } catch (SQLException e) {
        e.printStackTrace();
        return new DefaultTableModel();
    }
}

```

Текст програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" наведено в Додатку Б.

В таблиці 2.21 представлені класи еквівалентності та граничні значення вхідних даних:

- ПІБ працівника;
- посада;
- адреса;
- телефон;
- дата прийняття;
- дата звільнення.

Таблиця 2.21 – Класи еквівалентності вхідних даних для класу «Працівник»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
ПІБ працівника	Рядок	Графічні зображення (в тому числі і смайлики)
Посада	Рядок	Графічні зображення (в тому числі і смайлики)
Адреса	Рядок	Графічні зображення (в тому числі і смайлики)
Телефон	Рядок	Графічні зображення (в тому числі і смайлики)
Дата прийняття	Дата	Графічні зображення (в тому числі і смайлики)
Дата звільнення	Дата	Графічні зображення (в тому числі і смайлики)

Почнемо з тестування для правильних класів еквівалентності класу «Працівник», яке наведено в табл.2.22. За результатами правильних класів еквівалентності програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" працює правильно.

Таблиця 2.22 – Тести з правильних класів еквівалентності для класу «Працівник»

№ тесту	Вхідні дані	Вихідні дані
1	В поле «ПІБ працівника» введено ім'я користувача – Сидоров Петро.	Виведення на екран імені працівника. Результат вірний.
2	В поле «Посада» введено посаду працівника – менеджер.	Виведення на екран посади працівника. Результат вірний.
3	В поле «Адреса» введено адресу працівника – пр. Центральний, 47, кв. 7.	Виведення на екран адреси працівника. Результат вірний.
4	В поле «Телефон» введено телефон працівника – 099-35-07-222.	Виведення на екран телефону працівника. Результат вірний.
5	В поле «Дата прийняття» введено дату прийняття працівника – 20-02-2023	Виведення на екран дати прийняття працівника. Результат вірний.
6	В поле «Дата звільнення» введено дату звільнення працівника – 25-03-2023	Виведення на екран дати звільнення працівника. Результат вірний.

Протестуємо неправильні класи еквівалентності класу «Працівник», що наведено в табл. 2.23.

Таблиця 2.23 – Тести з неправильних класів еквівалентності класу «Працівник»

№ тесту	Вхідні дані	Вихідні дані
1	В поле «ПІБ працівника» введемо будь-який графічний символ ASCII	Програма виводить повідомлення про помилку. Результат підтверджено
2	В поле «Посада» ведемо графічний елемент – смайлик	Після ведення смайлика виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено
3	В поле «Адреса» ведемо будь-який графічний символ ASCII.	Після ведення ASCII виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено
4	В поле «Телефон» ведемо графічний елемент – смайлик	Після ведення смайлика виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено
5	В поле «Дата прийняття» ведемо будь-який графічний символ ASCII.	Програма виводить повідомлення про помилку. Результат підтверджено
6	В поле «Дата звільнення» ведемо графічний елемент – смайлик	Після ведення смайлика виводиться повідомлення про помилку. Програма далі працює правильно. Результат підтверджено

За результатами тестування неправильних класів еквівалентності класу «Працівник», програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" працює правильно, не допускаючи до введення будь-яких інших елементів, окрім стандартних символів вводу та випадального списку.

При тестуванні програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" розглядалося як чорний ящик. Тестування зводилися до послідовного вводу тестових наборів даних і аналізу отриманих після цього результатів.

З РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ МЕНЕДЖЕРА МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ "САТУРН"

В даній кваліфікаційній роботі була розв’язана практична задача інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов – було розроблено програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн". Програмне забезпечення призначено для спрощення роботи менеджера магазину побутової техніки "Сатурн" та полегшення ведення обліку магазину. Розробка програмного забезпечення виконана згідно стандартів. Зовнішній вигляд головного вікна програмного забезпечення наведено на рисунку 3.1.

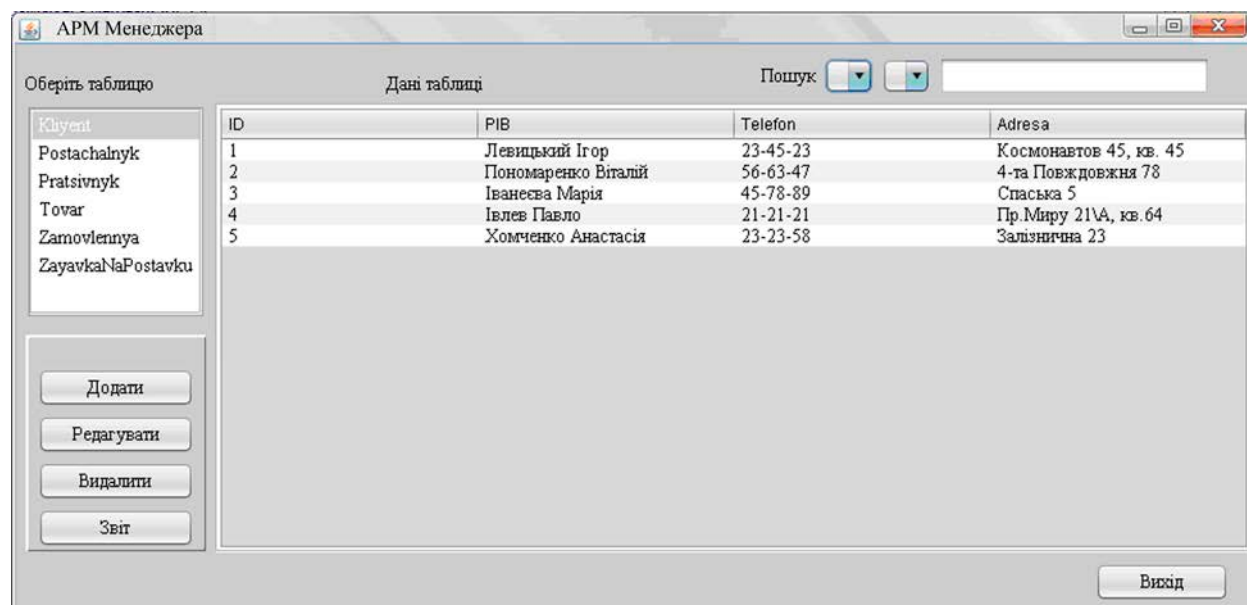


Рисунок 3.1 – Зовнішній вигляд головного вікна програмного забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн"

Проведено аналіз роботи магазину побутової техніки "Сатурн", зокрема, аналіз сучасних тенденцій автоматизації його діяльності. Проведено огляд літературних джерел, нормативних документів, Internet-джерел для аналізу

роботи та методів можливої автоматизації роботи менеджера магазину побутової техніки "Сатурн". Проведено аналіз існуючих програмних рішень.

Проведено аналіз вимог до програмного забезпечення та розроблено проєкт програмного забезпечення, виконані реалізація та тестування розробленого програмного забезпечення.

В рамках ескізного проєкту засобами мови моделювання UML побудовані концептуальна модель програмного забезпечення та діаграми діяльності. Також розроблений ескіз інтерфейсу.

В рамках технічного проєкту побудована логічна модель бази даних, діаграма класів та динамічна модель програмного забезпечення у вигляді діаграм послідовностей.

В рамках робочого проєкту виконано вибір і обґрунтування інструментів розробки та системи управління базою даних, побудована фізична модель бази даних, виконано кодування та тестування програмного забезпечення.

В якості мови програмування була обрана Java. В якості системи управління базою даних була обрана MySQL. В якості середовища розробки було обрано середовище NetBeans IDE. Для використання розробленого програмного забезпечення необхідна наявність встановленого Standard Edition Development Kit 1.8 (JDK 1.8).

Було виконано тестування та випробування програмного забезпечення, яке показало його повну працездатність та відповідність вимогам, наведеним у технічному завданні.

Також була розроблена наступна програмна документація: технічне завдання, опис програми, інструкція користувача, програма і методика випробувань. Наведено роздрук фрагментів програми.

Розроблене програмного забезпечення реалізує наступні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;

- зберігання та використання інформації з продажу товару;
- складання звітності.

програмного забезпечення було розроблено для використання в магазині побутової техніки "Сатурн".

Було проведено випробування програмного забезпечення. Програму і методику випробування наведено у Додатку Д.

Протокол випробування

Під час випробувань було проведено повне функціональне випробування, а також навантажувальне випробування та випробування на відмову всього програмно-апаратного комплексу.

Всі виявлені недоліки програмного забезпечення були зафіксовані в протоколах і усунуті до моменту впровадження програмного забезпечення у дію.

Методи випробувань

Випробування було проведено за стратегією «чорного ящика». Через велику кількість функцій, які потрібно було випробувати, представлено результати випробувань декількох функцій програми.

1. Проведено перевірку програми на відповідність технічному завданню:

- перевірка роботи функції «Додавання запису»

На головному вікні обрали таблицю та натиснули кнопку «Додати». З'явилось відповідне вікно, в якому внесли необхідну інформацію. Після чого натиснули кнопку «Зберегти».

- перевірка роботи функції «Пошук інформації»

На головному вікні обрали таблицю, потім водимо дані для пошуку, та натиснули кнопку «Пошук». З'явилися результати пошуку.

- перевірка роботи функції «Формування замовлення»

На головному вікні обрали таблицю «Замовлення», та натиснули кнопку «Звіт». З'явилось відповідне вікно у якому обрали клієнта та натиснули кнопку «Сформувати». У результаті отримали сформоване замовлення, яке можливо роздрукувати.

2. Проведено перевірку програми на її програмно-апаратну сумісність: виконано випробування на апаратно-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; виконано перевірку на наявність і усунення всіх помилок, зазначених у п.1.

3. Проведено візуальний перегляд програми: виконано остаточне налагоджено на предмет виявлення та усунення дрібних помилок.

4 ОХОРОНА ПРАЦІ

Охорона праці – це система соціально-економічних, правових, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, які спрямовані на збереження життя, здоров'я та працездатності людини під час трудової діяльності [16].

Ця тема є актуальною, бо у теперішній час Україна знаходиться в небезпечному стані щодо питань охорони праці, життя та здоров'я суспільства. Багато законів залишаються лише у документі, а насправді не діють. Чимало працівників не мають офіційного оформлення, а тому не можуть законно захищати свої права.

Охорона праці охоплює безпосередньо три головні складові: правові норми трудового законодавства, гігієну та техніку безпеки, виробничу санітарію, а також протипожежний захист і електробезпеку.

Метою охорони праці є забезпечення безпечних, незагрозливих та придатних для праці умов.

4.1 Аналіз небезпечних і шкідливих факторів у відділі розробки програмного забезпечення

Під час роботи на ПК робітники відділу мають справу з дією таких небезпечних факторів: підвищена температура, погане освітлення робочого місця, електричний струм, інтелектуальне або емоційне перевантаження, перенапруження очей, монотонна праця. Проаналізуємо чинники більш докладно.

Небезпечними є монітори з електронно-променевими трубками (ЕЛТ). Робота дисплея спричиняє електростатичні та електромагнітні випромінювання. Згідно зі стандартами не рекомендується використовувати

дисплеї з частотою менше, ніж 75 Гц.

Найбільш прийнятними мікрокліматичними параметрами вважається температура 23° Цельсія і 18° Цельсія у теплу та холодну пору року відповідно за відносної вологості 55%.

Також можуть виникнути й інші загрози у процесі трудової діяльності.

Загроза ураження електричним струмом. Прийнятним струмом можна вважати такі, що не становлять загрози організму людини, а саме змінний струм промислової частоти 0,6-1,5 мА та постійний струм 5-7 мА.

Погане освітлення спричиняє пониження працездатності та продуктивності, надає негативний психологічний вплив на працюючих.

Прийнятна освітленість для роботи за монітором варіюється у межах 400-750 лк, а яскравість в полі зору працівників – у межах 1:5-1:10.

Небезпека виникнення пожежі. Категорія пожежної небезпеки "В" включає споруди, де розміщені персональні комп'ютери. У таких приміщеннях як правило чимало джерел спалаху: кабелі, техніка, устаткування, меблі, папір тощо.

Підвищення технічного рівня виробництва нерозривно зв'язано із широким використанням різних видів електроніки, телебачення, оптичних квантових генераторів, електронно-обчислювальних машин. Усі ці пристрої є джерелами електромагнітних випромінювань, що зараз прийнято називати єдиним терміном неіонізуючі випромінювання. Коло осіб, що піддаються впливові таких випромінювань, увесь час розширюється.

4.2 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів у відділі розробки програмного забезпечення

Монітори, повинні відповідати стандартам з вертикальною частотою розгортки не менше 75 Гц. Мікрокліматичні показники повинні бути оптимальними (23° С та 18° С у теплу та холодну пору року відповідно),

рекомендовано, щоб в офісі був кондиціонер. Вологе прибирання у приміщенні з ПК повинно проводитись щодня. Потрібно встановити пристрої заземлення для безпеки персоналу.

Щодо освітлення у темні години дня слід користуватись люмінесцентними лампами, бо вони є довговічними (10000+год.) та мають високу світлову віддачу (75+лм/Вт).

Система пожежної безпеки задовольняє норми, але для покращення умов можна ще встановити підлогу з негорючої сировини, димові датчики тощо.

Шум може негативно впливати на працездатність людини на роботі. Враховуються такі параметри звуку, тиск, інтенсивність, частота коливань. Відповідно до норм ДСН 3.3.6.037-99, рівень шуму не повинен перевищувати 50 дБ.

Нижче наведено перелік елементів, які можуть створювати шум у відділі:

- система охолодження персонального комп'ютера;
- працюючий принтер;
- транспорт із зовні;
- кондиціонер
- система охолодження серверу.

Згідно з НАПБ Б.03.002-2007 відділ належить до категорії В – приміщення, у якому наявні горючі тверді, волокнисті матеріали.

Відповідно до ППБУ, у кімнаті, що аналізується, необхідно мати:

1. 1 вуглекислотний вогнегасник (3+кг/20м²). Із розрахунку на площу даного приміщення: 4 переносні вогнегасники марки ВВ-5 (по 3,5 кг);
2. План для евакуації з офісу;
3. Система протипожежних датчиків, пожежна сигналізація.

В офісній кімнаті 1 вихід та менше 25 працівників. Між робочими місцями у кімнаті проходи шириною більше 1 м. Сходові клітини комбінують природне та штучне освітлення. Сходи та офіс мають систему евакуаційного освітлення. Робітники ознайомлені з порядком і планом евакуування з офісу. Отже, шляхи евакуації з приміщення повністю відповідають нормам.

Для робітників, які постійно працюють з комп'ютером, рекомендується проводити заходи психологічного розвантаження, кожну годину робити перерви (15 хв.).

Аналіз показав, що мікроклімат, штучне освітлення, заземлення та система пожежної безпеки у відділі відповідає нормам та стандартам.

Швидкий темп змін в умовах праці великого контингенту трудящих приводить до посилення впливу на працюючу людину нових раніше невідомих або мало вивчених небезпечних і шкідливих виробничих факторів фізичного, біологічного й особливо психофізіологічного характеру. У зв'язку з цим необхідно удосконалювати роботи з охорони праці, упроваджувати нові ефективні засоби забезпечення безпеки праці.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розв'язано практичну задачу інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов – розроблено програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн", а саме:

- проведено аналіз практичної задачі інженерії програмного забезпечення, зокрема, аналіз сучасних тенденцій автоматизації діяльності магазину побутової техніки "Сатурн", проведено аналіз існуючих програмних рішень, зроблено висновок про доцільність розробки власного ПЗ та виконана постановка задачі;

- проведено аналіз вимог до програмного забезпечення;
- виконано ескізне, технічне та робоче проектування програмного забезпечення;
- були представлені результати розробки.

Також були розглянуті загальні питання охорони праці в Україні.

Розроблене програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" призначене для спрощення роботи менеджера магазину побутової техніки та полегшення ведення обліку магазину і дозволяє виконувати наступні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;
- зберігання та використання інформації з продажу товару;
- складання звітності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Через війну в Україні на 28% впав ринок побутової техніки. URL: <https://mediasat.info/uk/2022/08/31/cherez-vijnu-v-ukraini-na-28-vpav-rinok-robutovoi-tehniki/> (дата звернення: 17.02.2023).
2. Гаджетизація торгівлі: що відбувається на ринку торгівлі побутовою технікою та електронікою. URL: <https://rau.ua/novyni/gadzhelizacija-torgivli/> (дата звернення: 17.02.2023).
3. Ритейлери прогнозують дефіцит недорогої техніки та електроніки. URL: <https://forbes.ua/inside/riteyleri-tehniki-ta-elektroniki-prognozuyut-defitsit-nedorogoi-tehniki-ta-elektroniki-12052022-5932> (дата звернення: 17.02.2023).
4. MASTER:Бухгалтері. URL: <https://masterbuh.com/product/buhgalteriya> (дата звернення: 23.02.2023).
5. Онлайн-бухгалтерія Dilovod. URL: <https://dilovod.ua/> (дата звернення: 23.02.2023).
6. Якобсон А. Унифицированный процесс разработки программного обеспечения. СПб: Питер, 2002. 496 с.: ил.
7. Діаграма діяльності (activity diagram). URL: http://ni.biz.ua/6/6_3/6_33177_diagramma-deyatelnosti-activity-diagram.html (дата звернення: 05.04.2023).
8. Проектирование и разработка корпоративных информационных систем. Концептуальные модели и схемы баз данных. URL: http://citforum.ck.ua/cfin/prcorpsys/infisistpr_09.shtml (дата звернення: 05.04.2023).
9. Інтерфейс користувача. URL: <http://sven.ua/ua/service/glossary/ukr/11/> (дата звернення: 12.04.2023).
10. Діаграма класів. URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?id=28088> (дата звернення: 16.04.2023).

11. Діаграма взаємодії. URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?r=16538> (дата звернення: 16.04.2023).
12. Докладніше про технологію Java. URL: <https://www.java.com/about/> (дата звернення: 11.05.2023).
13. The GNU General Public License. URL: <http://www.gnu.org/licenses/licenses.en.html> (дата звернення: 11.05.2023).
14. MySQL. The world's most popular open source database. URL: <https://www.mysql.com/> (дата звернення: 11.05.2023).
15. Офіційний сайт NetBeans. URL: <https://netbeans.apache.org/> (дата звернення: 11.05.2023).
16. Про охорону праці: Закон України від 21.11.2002 р. № 229-IV. Дата оновлення: 04.02.2021. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text>. (дата звернення: 17.05.2023).

Додаток А Технічне завдання

Вступ

Найменування програми – Програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн". Область застосування –магазин побутової техніки "Сатурн".

1 Підстава для розробки

Підставою для розробки даного програмного забезпечення є наказ на затвердження тем кваліфікаційних робіт бакалаврів зі спеціальності 121 Інженерія програмного забезпечення в Національному університеті кораблебудування ім. адмірала Макарова.

2 Призначення розробки

2.1 Експлуатаційне призначення

Експлуатаційним призначенням програмного забезпечення є спрощення роботи менеджера магазину побутової техніки "Сатурн" та полегшення ведення обліку магазину.

2.2 Функціональне призначення

Функціональним призначенням програмного забезпечення є автоматизація процесів:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;
- зберігання та використання інформації з продажу товару;
- складання звітності.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програма повинна виконувати наступні функції:

Авторизація користувача

Авторизація користувача в програмному забезпеченні для надання функціональних можливостей. Обов'язковими полями є: логін та пароль.

Представлення даних

Данні представляються у вигляді наступних таблиць у БД: працівник, постачальник, товар, заявка на поставку, клієнт, замовлення. Атрибути таблиць та їх опис наведено у таблицях А.1 – А.6.

Таблиця А.1 – Опис атрибутів таблиці БД «Працівник»

Атрибут	Тип даних
Код працівника	Число
ПІБ	Текст
Посада	Текст
Адреса	Текст
Телефон	Текст
Дата прийняття на роботу	Дата/Час
Дата звільнення	Дата/Час

Таблиця А.2 – Опис атрибутів таблиці БД «Постачальник»

Атрибут	Тип даних
Код постачальника	Число
Найменування	Текст
Представник	Текст
Адреса	Текст
Телефон	Текст

Таблиця А.3 – Опис атрибутів таблиці БД «Товар»

Атрибут	Тип даних
Код товару	Число
Постачальник	Число
Тип товару	Текст
Найменування	Текст
Ціна закупки	Число
Кількість	Число
Ціна продажу	Число

Таблиця А.4 – Опис атрибутів таблиці БД «Заявка на поставку»

Атрибут	Тип даних
Код заявки	Число
Код постачальника	Число
Тип товару	Текст
Найменування товару	Текст
Кількість	Число
Дата формування заявки	Дата/Час

Таблиця А.5 – Опис атрибутів таблиці БД «Клієнт»

Атрибут	Тип даних
Код клієнта	Число
ПІБ	Текст
Адреса	Текст
Телефон	Текст

Таблиця А.6 – Опис атрибутів таблиці БД «Замовлення»

Атрибут	Тип даних
Код замовлення	Число
Код клієнта	Число
Тип товару	Текст
Товар	Число
Код працівника	Число
Сума продажу	Число
Кількість	Число
Дата продажу	Дата/Час

Перегляд даних

Для перегляду даних потрібно обрати таблицю.

Пошук необхідної інформації

Функція пошуку необхідної інформації відбувається за певними параметрами: необхідно обрати таблицю та вести атрибути для пошуку.

Додавання нового запису

Для додавання нового запису потрібно обрати таблицю.

Редагування запису

Для редагування запису потрібно обрати таблицю та відповідний запис.

Видалення запису

Для видалення запису потрібно обрати таблицю та відповідний запис.

Формування заявки на поставку

Формування заявки на поставку товару від постачальника.

Оформлення продажу товару

Оформлення продажу товару клієнту з надання чеку.

Формування звітів

Формування звітів повинно відбуватися автоматично після обрання необхідних відповідних параметрів.

3.1.2 Вимоги до організаційних вхідних та вихідних даних

Дані зберігаються в файлі БД з довільним доступом типу .mud.

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних (клавіатура та миша). Дані або вводяться вручну, або обираються із представлених списків.

Вихідні дані: виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

3.2 Вимоги до надійності

Програмний продукт повинен нормально функціонувати при безперебійній роботі комп'ютера. При виникненні збоїв в роботі, відновлення нормальної роботи повинне проводитися після перезавантаження програми.

Для забезпечення надійності інформації повинна використовуватися СКБД, що забезпечує цілісність транзакцій і несе відповідальність за цілісність

інформації.

Система повинна продовжувати коректне функціонувати при втраті частини інформації. У випадку неможливого продовження коректної роботи має бути повідомлення про це.

У разі введення користувачем некоректної інформації система повинна повідомити про помилку і надати можливість виправити її.

3.3 Вимоги до умов експлуатації

Необхідний рівень підготовки користувачів: мінімальні навички в користуванні комп'ютером. Комп'ютер призначений для роботи в закритому опалювальному приміщенні.

3.4 Вимоги до складу та параметрів технічних засобів

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- CPU: Intel(R) i3;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

3.5 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування системи необхідно використовувати ОС Windows версії не нижче 7.

Необхідна наявність встановленого Standard Edition Development Kit 1.8 (JDK 1.8).

3.6 Вимоги до маркування та пакування

Продукт буде упакований в електронний архів і мати найменування `manager.rar` та містити:

- виконуваний файл `manager.jar`;
- файл з описом `ReadMe.txt`;

– файл, який містить структуру БД manager.sql.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- текст програми;
- опис програми;
- інструкцію користувача;
- програму і методику випробувань програмного забезпечення.

5 Техніко-економічні показники

Не розраховуються в зв'язку з тим, що продукт розробляється в рамках кваліфікаційної роботи.

6 Стадії та етапи розробки

Стадії та етапи розробки представлені в таблиці А.7.

Таблиця А.7 – Стадії та етапи розробки програмного забезпечення

№ п/п	Назва етапів роботи	Терміни виконання
1.	Аналіз практичної задачі інженерії ПЗ	31.03.2023
2.	Аналіз вимог до ПЗ	07.04.2023
3.	Розробка архітектури ПЗ	21.04.2023
4.	Розробка проекту ПЗ	05.05.2023
5.	Реалізація та тестування ПЗ	12.05.2023

7 Порядок контролю та приймання

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Приймання проводяться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі

знаходження помилок підчас прийому програмного виробу складається акт про знайдені помилки, який підписуються представниками замовника і розробника і затверджуються керівником організації – замовника та організації – розробника. Розробник повинен протягом не більше ніж двох тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

Додаток Б Текст програми

```
package com.boss;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
public class DBHelper {
    public static Connection connection = null;
    final private static Vector<Vector> tableData = new Vector<>();
    final private static Vector columnNames = new Vector();
    final private static Vector row = new Vector();
    private static final DefaultTableModel tableModel = new
DefaultTableModel();
    public static ResultSet resultSet = null;
    final public static ArrayList id = new ArrayList();
    public static Statement statement = null;
    public static String CURRENT_TABLE;
    static public Connection openConnection(String db) {
        if (connection == null) {
```

```

        try {
            Class.forName("org.sqlite.JDBC");
            System.out.println(db);
            connection = (Connection)
DriverManager.getConnection("jdbc:sqlite:db/" + db);
            if (connection != null) {
                System.out.println("Connected");
            } else {
                System.out.print("Connection failed");
            }
        } catch (ClassNotFoundException | SQLException ex) {
            System.err.println(ex.getCause());
        }
    }
    return connection;
}

public static void closeConnection() {
    try {
        if (statement != null && !statement.isClosed()) {
            statement.close();
        }
        getConnection().close();
        System.out.println("Connection closed.");
    } catch (SQLException ex) {
        System.err.println(ex.getCause());
    }
}

public static DefaultTableModel executeSelectQuery(String... q) {
    tableData.clear();
    columnNames.clear();
    CURRENT_TABLE = q[0];
    String query = "Select * from " + CURRENT_TABLE + " " + (q.length
== 1 ? "" : q[1]);
    System.out.println(query);
    try {
        resultSet = getStatement().executeQuery(query);
        initColumnNames(resultSet.getMetaData());
        while (resultSet.next()) {

```

```

        row.clear();
        for (int i = 0; i < columnNames.size(); i++) {
            row.add(resultSet.getObject(i + 1));
        }
        tableData.add((Vector) row.clone());
    }
    tableModel.setDataVector(tableData, columnNames);
    return tableModel;
} catch (SQLException e) {
    e.printStackTrace();
    return new DefaultTableModel();
}
}

public static void executeInsertQuery(Vector colNames, Vector
rowValues) {
    StringBuilder query = new StringBuilder("INSERT INTO ");
    StringBuilder queryValues = new StringBuilder(" VALUES ('");
    query.append(CURRENT_TABLE).append(" (");
    for (int i = 0; i < rowValues.size(); i++) {
        if (i != rowValues.size() - 1) {
            query.append(colNames.get(i)).append(",");
            queryValues.append(rowValues.get(i)).append("'", "");
        } else {
            query.append(colNames.get(i)).append(")");
            queryValues.append(rowValues.get(i)).append("'");
        }
    }
    query.append(queryValues);
    System.out.println(query);
    try {
        getStatement().execute(query.toString());
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void executeDeleteQuery(int indexToDelete) {
    try {
        getStatement().execute("Delete from " + CURRENT_TABLE + "

```

```

Where ID = " + indexToDelete);
        System.out.println("Index " + indexToDelete + " from " +
CURRENT_TABLE + " was deleted;");
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void executeUpdateQuery(Vector colNames, Vector
rowValues, int ID) {
    StringBuilder query = new StringBuilder("UPDATE ");
    query.append(CURRENT_TABLE).append(" SET ");
    for (int i = 0; i < rowValues.size(); i++) {
query.append(colNames.get(i)).append("=").append(rowValues.get(i)).appen
d("'" + " ");
        if (i != rowValues.size() - 1) {
            query.append(", ");
        } else {
            query.append(" WHERE ID = ").append(ID);
        }
    }
    System.out.println(query);
    try {
        getStatement().execute(query.toString());
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

private static void initColumnNames(ResultSetMetaData metaData)
throws SQLException {
    for (int i = 0; i < metaData.getColumnCount(); i++) {
        columnNames.add(metaData.getColumnName(i + 1));
    }
}

private static Connection getConnection() throws SQLException {
    if (connection == null || connection.isClosed()) {
        System.err.print("Connection does not exists.");
    }
    return connection;
}

```

```

private static Statement getStatement() throws SQLException {
    if ((statement == null || statement.isClosed()) && (connection !=
null)) {
        statement = getConnection().createStatement();
    }
    return statement;
}
public static Vector getColumnNames() {
    return columnNames;
}
/**
 * Returns column names without ID column
 *
 * @return
 */
public static Vector getExtraColumnNames() {
    Vector extraColumnNames = (Vector) columnNames.clone();
    extraColumnNames.remove(0);
    return extraColumnNames;
}
public static Vector<String> getTables() {
    Vector<String> tableNames = new Vector<>();
    try {
        DatabaseMetaData metaData = connection.getMetaData();
        resultSet = metaData.getTables("", "", "%", null);
        while (resultSet.next()) {
            if (!resultSet.getString(3).startsWith("sqlite_")) {
                tableNames.add(resultSet.getString(3));
            }
        }
    } catch (SQLException ex) {
        null, ex);
    }
    return tableNames;}

```

```

package com.boss.view;
import javax.swing.JOptionPane;
public class Autorization extends javax.swing.JFrame {
    public Autorization() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {
        LoginjLabel1 = new javax.swing.JLabel();
        PasswjLabel1 = new javax.swing.JLabel();
        PasswordjTextField1 = new javax.swing.JTextField();
        EnterjButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        ExitjButton1 = new javax.swing.JButton();
        LoginjTextField1 = new javax.swing.JTextField();
        LogojLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("APM Менеджера");
        setBackground(new java.awt.Color(255, 255, 255));
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        LoginjLabel1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
        LoginjLabel1.setText("Login");
        PasswjLabel1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
        PasswjLabel1.setText("Пароль");
        PasswordjTextField1.setFont(new java.awt.Font("Times New Roman",
0, 14)); // NOI18N
        PasswordjTextField1.setText("manager");
        PasswordjTextField1.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                PasswordjTextField1ActionPerformed(evt);
            }
        });
        EnterjButton1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
        EnterjButton1.setText("Вхід");

```

```

        EnterjButton1.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

        EnterjButton1.addActionListener(new
java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                EnterjButton1ActionPerformed(evt);

            }

        });

        jLabel1.setFont(new java.awt.Font("Monospaced", 1, 24)); //
NOI18N

        jLabel1.setText("BIOS - Buy In Our Shop");

        ExitjButton1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N

        ExitjButton1.setText("Вихід");

        ExitjButton1.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

        ExitjButton1.addActionListener(new
java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                ExitjButton1ActionPerformed(evt);

            }

        });

        LoginjTextField1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N

        LoginjTextField1.setText("manager");

        LoginjTextField1.addActionListener(new
java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                LoginjTextField1ActionPerformed(evt);

            }

        });

        LogojLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/ic1.PNG"))); // NOI18N

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .add(jLabel1)

                        .add(LoginjTextField1)

                        .add(ExitjButton1)

                        .add(LogojLabel2)

                    )

                    .addContainerGap())

                .add(layout.createSequentialGroup()

                    .add(EnterjButton1)

                    .addContainerGap())

        );

        javax.swing.GroupLayout gLayout = new javax.swing.GroupLayout(this);
        gLayout.setHorizontalGroup(
            gLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(layout)
        );
        gLayout.setVerticalGroup(
            gLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(layout)
        );

        this.setLayout(gLayout);
    }

    /**
     * The method is called when the user clicks the "BIOS" button.
     * It opens a new window titled "BIOS" and displays the BIOS
     * information.
     */
    private void EnterjButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        BIOS bios = new BIOS();
        bios.setVisible(true);
    }

    /**
     * The method is called when the user clicks the "Exit" button.
     * It closes the current window.
     */
    private void ExitjButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    /**
     * The method is called when the user clicks the "Login" button.
     * It checks the username and password and displays the
     * appropriate message.
     */
    private void LoginjTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
        String username = LoginjTextField1.getText();
        String password = LoginjPasswordField1.getText();

        if (username.equals("manager") && password.equals("123456")) {
            JOptionPane.showMessageDialog(this, "Login successful");
        } else {
            JOptionPane.showMessageDialog(this, "Invalid username or password");
        }
    }

    /**
     * The method is called when the user clicks the "Logo" button.
     * It displays the BIOS logo.
     */
    private void LogojLabel2MouseClicked(java.awt.event.MouseEvent evt) {
        BIOS bios = new BIOS();
        bios.setVisible(true);
    }
}

```

ADING)

```

        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1))
        .addGroup(layout.createSequentialGroup()
            .addGap(76, 76, 76)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ADING)
            .addComponent(ExitjButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(EnterjButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ADING)
            .addComponent(LoginjLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(PasswjLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(37, 37, 37)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(LogojLabel2)
                .addGap(0, 0, Short.MAX_VALUE))
            .addComponent(LoginjTextField1)
            .addComponent>PasswordjTextField1))))
        .addContainerGap())
    );
    layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()

```

```

        .addComponent(jLabel1)
        .addGap(34, 34, 34)
        .addComponent(LogojLabel2)
        .addGap(31, 31, 31)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(LoginjLabel1)
        .addComponent(LoginjTextField1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(PasswjLabel1)
        .addComponent>PasswordjTextField1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(EnterjButton1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 32,
            javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(ExitjButton1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 32,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE))
    );
    pack();
}

private void
LoginjTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void
PasswordjTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void EnterjButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

```

```

String login = LoginjTextField1.getText();
String password = PasswordjTextField1.getText();
if (login == null || login.isEmpty()) {
    JOptionPane.showMessageDialog(rootPane, "Введіть логін");
} else if (password == null || password.isEmpty()) {
    JOptionPane.showMessageDialog(rootPane, "Введіть пароль");
} else if ("manager".equals(login.trim()) &&
"manager".equals(password.trim())) {
    dispose();
    new View();
} else {
    JOptionPane.showMessageDialog(rootPane, "Логін або пароль не
вірний");
}

private void ExitjButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
    dispose();
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex)
    {
        java.util.logging.Logger.getLogger(Autorization.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        @Override

```

```
        public void run() {
            new Autorization().setVisible(true);
        }
    });
}
private javax.swing.JButton EnterjButton1;
private javax.swing.JButton ExitjButton1;
private javax.swing.JLabel LoginjLabel1;
private javax.swing.JTextField LoginjTextField1;
private javax.swing.JLabel LogojLabel2;
private javax.swing.JLabel PasswjLabel1;
private javax.swing.JTextField PasswordjTextField1;
private javax.swing.JLabel jLabel1;
}
```

Додаток В Опис програми

1 Загальні відомості

Розроблене ПЗ для автоматизації роботи менеджера магазину побутової техніки "Сатурн" представляє собою програму для співробітника магазину побутової техніки, який займається вводом та формуванням інформації про заявки на поставку товару та продаж товару.

Програма призначена для спрощення роботи менеджера магазину побутової техніки та полегшення ведення обліку магазину побутової техніки.

Для розробки програмного забезпечення були обрані наступні засоби:

- в якості мови програмування була обрана Java;
- в якості СКБД була обрана MySQL;
- в якості середовища розробки було обрано середовище NetBeans IDE.

Для використання розробленого ПЗ необхідна наявність встановленого Standard Edition Development Kit 1.8 (JDK 1.8).

2 Функціональне призначення

Програма виконує наступні основні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;
- зберігання та використання інформації з продажу товару;
- складання звітності.

3 Опис логічної структури

Програма розроблена на модульній основі, тобто кожний компонент (модуль) програми відповідає за конкретну дію, наприклад: формування заявки на поставку товару (відповідає за формування заявки на поставку товару). Дані, необхідні для роботи програми, зберігаються у нормалізованих таблицях бази даних.

4 Технічні та програмні засоби

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- CPU: Intel(R) i3;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

Для повноцінного функціонування системи необхідно використовувати ОС Windows версії не нижче 7.

Необхідна наявність встановленого Standard Edition Development Kit 1.8 (JDK 1.8).

5 Виклик та завантаження

Програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" встановлюється безпосередньо на комп'ютер користувача. Програмне забезпечення відкривається або за допомогою ярлика, який знаходиться на робочому столі, або за допомогою вибору відповідного пункту меню «Пуск» – «Програми». Програма має один режим використання.

Після запуску програми з'являється вікно авторизації користувача, де пропонується ввести логін та пароль для входу в систему. Якщо дані користувача введені вірно, відкривається головна форма програми.

6 Вхідні та вихідні дані

Дані зберігаються в файлі БД з довільним доступом типу .myd.

Вхідні дані:

Введення даних здійснюється за допомогою пристроїв введення даних (клавіатура та миша). Дані або вводяться вручну, або обираються із представлених списків.

Вихідні дані:

Виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

Додаток Г Інструкція користувача

Програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн" встановлюється безпосередньо на комп'ютер користувача. Програмне забезпечення відкривається або за допомогою ярлика, який знаходиться на робочому столі, або за допомогою вибору відповідного пункту меню «Пуск» – «Програми». Програма має один режим використання.

Після запуску програми з'являється вікно авторизації користувача, де пропонується ввести логін та пароль для входу в систему. Вікно авторизації наведено на рис. Г.1.

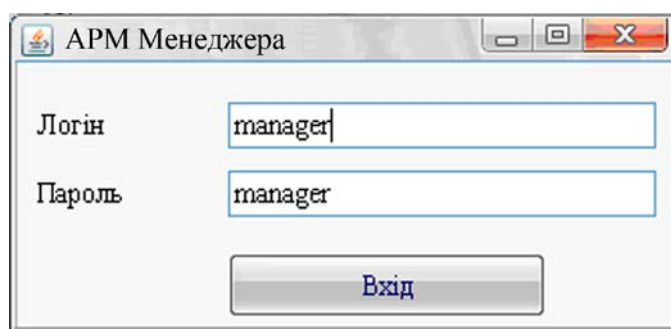


Рисунок Г.1 – Вікно авторизації користувача

Якщо було введено не вірні дані користувача, з'являється діалогове вікно з попередженням, далі програма надає змогу для повторного введення даних. Якщо дані користувача введені вірно, відкривається головна форма програми, наведена на рис. Г.2.

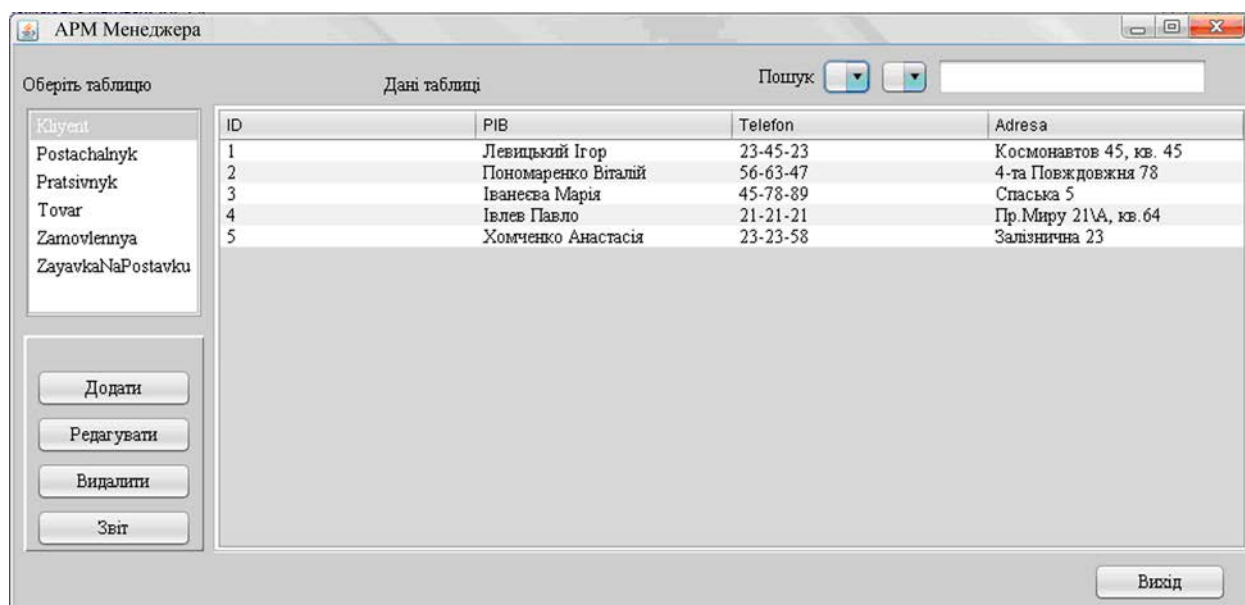


Рисунок Г.2 – Головна форма програми

Для роботи із записами таблиці необхідно спочатку обрати таблицю зі списку доступних таблиць, що розташований в лівій частині головної форми програми. Після цього дані з обраної таблиці з'являться на панелі «Дані таблиці», що наведено на рис. Г.3.

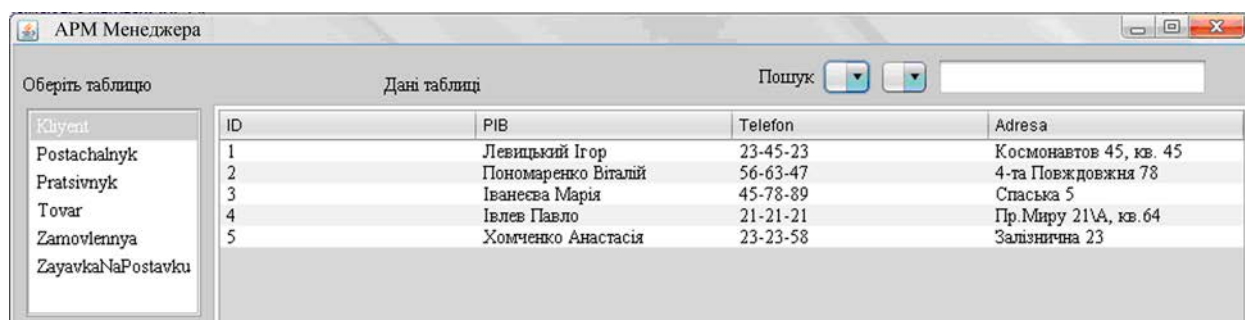


Рисунок Г.3 – Обрана таблиця та панель «Дані таблиці»

Для додавання нового запису у таблицю необхідно:

1. Обрати таблицю.
2. На головній формі натиснути кнопку «Додати» (див. рис. Г.2).
3. На формі для додавання нового запису у таблицю заповнити всі необхідні поля на натиснути кнопку «Зберегти». У разі помилки натиснути

кнопку «Відмінити». Форму для додавання нового запису у таблицю наведено на рис. Г.4.

Рисунок Г.4 – Форма для додавання нового запису

Для редагування запису в таблиці необхідно:

1. Обрати таблицю.
2. Обрати запис із таблиці.
3. На головній формі, натиснути кнопку «Редагувати» (див. рис. Г.2).
4. На формі для редагування запису з таблиці відредагувати всі необхідні поля на натиснути кнопку «Зберегти». У разі помилки натиснути кнопку «Відмінити».

Для видалення запису з таблиці необхідно:

1. Обрати таблицю.
2. Обрати запис із таблиці.
3. На головній формі натиснути кнопку «Видалити» (див. рис. Г.2).

Для друкування замовлення необхідно:

1. Обрати таблицю «Замовлення».
2. На головній формі натиснути кнопку «Звіт» (див. рис. Г.2).
3. У вікні «Друкувати звіт» обрати клієнта з випадаючого списку та натиснути кнопку «Друкувати». Вікно «Друкувати звіт» з обраним клієнтом наведено на рис. Г.5.

Рисунок Г.5 – Вікно «Друкувати звіт» з обраним клієнтом

Додаток Д Програма та методика випробування програмного забезпечення

1. Об'єкт випробувань

1.1 Назва об'єкту

Повна назва об'єкта випробувань: програмне забезпечення для автоматизації роботи менеджера магазину побутової техніки "Сатурн".

Коротка назва: програма.

1.2 Область застосування

Програма призначена для спрощення роботи менеджера магазину побутової техніки "Сатурн" та полегшення ведення обліку магазину побутової техніки. Програма забезпечує:

- реєстрацію та авторизацію користувачів;
- зберігання та використання інформації про постачальників;
- зберігання та використання інформації про товар;
- зберігання та використання інформації про покупців;
- зберігання та використання інформації з продажу товару;
- складання звітності.

2 Мета випробувань

Мета проведення випробувань – оцінка експлуатаційних характеристик програми, перевірка і підтвердження працездатності програми в умовах, максимально наближених до умов реальної експлуатації.

3 Вимоги до програми

Програма має реалізовувати функції, описані в технічному завданні, яке наведено в додатку А.

4 Вимоги до програмної документації

Для проведення тестування повинна бути надана наступна документація:

- технічне завдання на розробку програмного забезпечення;
- текст програми;
- опис програми;

- інструкція користувача;
- програма і методика випробувань програмного забезпечення.

5 Склад, порядок та методи випробувань

5.1 Перевірка програми на відповідність технічному завданню:

- перевірка роботи функції «Додавання запису»;
- перевірка роботи функції «Редагування запису»;
- перевірка роботи функції «Видалення запису»;
- перевірка роботи функції «Пошук інформації»;
- перевірка роботи функції «Формування замовлення».

5.2 Перевірка програми на її програмно-апаратну сумісність: тестування на апаратних-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; перевірка на наявність і усунення всіх помилок, зазначених у п. 5.1.

5.3 Візуальний перегляд програми: остаточне налагодження на предмет виявлення та усунення дрібних помилок.