

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«_____» _____ 2018 р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна система автоматизації контролю та обліку товарів
підприємства

Освітньо-кваліфікаційний рівень “бакалавр”
Напрямок 6.050102 – “Комп’ютерна інженерія”

Керівник проекту:

(підпис)

Сафонова С.О.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Рябоконт Д.А.

(ініціали, прізвище)

Група:

КІ-14бд

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.050102 Комп'ютерна інженерія
(шифр і назва)
Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
_____ І.С. Скарга-Бандурова
« _____ » _____ 2018 р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Рябоконею Дмитру Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система автоматизації контролю та обліку
товарів підприємства

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "14 " 05 _____ 2018 р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області і постановка задачі. Проектування бази даних. Проектування архітектури програми. Вибір середовища розробки. Розробка програмного забезпечення. Інструкція по роботі з системою. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. кафедри КНІ Критська Я.О.		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області і постановка задачі	14.05.18-19.05.18	
2	Проектування бази даних	20.05.18-25.05.18	
3	Проектування архітектури програми. Вибір середовища розробки.	26.05.18-02.06.18	
4	Розробка програмного забезпечення.	03.06.18-06.06.18	
5	Розробка інструкції по роботі з системою	07.06.18-09.06.18	
6	Розробка розділу охорона праці	10.06.18-12.06.18	
7	Оформлення електронних плакатів	13.06.18-15.06.18	
8	Оформлення пояснювальної записки	14.05.18-15.05.18	

Здобувач вищої освіти _____

(підпис)

Рябоконт Д.А. _____

(прізвище та ініціали)

Керівник _____

(підпис)

Сафонова С.О. _____

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 86 с., 48 рис., 3 табл., 21 бібліографічних джерел посилань, 3 додатки.

Об'єкт розробки: процеси обліку товарів підприємства, що поширює побутову техніку.

Мета роботи: побудова системи обліку товарів підприємства для ефективного та якісного отримання та обробки інформації, а також забезпечення контролю та обліку реалізації продукції.

В проекті виконано:

1. Проведено аналіз діяльності підприємства, сформульована постановка задачі.
2. Розроблена концептуальна, логічна і фізична моделі даних.
3. Вибір засобів розробки програмного забезпечення.
4. Спроектована та розроблена інформаційна система.
5. Здійснений аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкта, що впливають на персонал.

Отримано наступні результати: розроблена система дозволяє оформляти операції продажу та закупівлі товарів і формувати звіти. Перевагами ІС є можливість обробки запитів від багатьох користувачів, незалежність від ОС та розширюваність завдяки архітектурі ПЗ.

Практичне значення, галузь застосування роботи: розроблена інформаційна система дозволяє швидко і зручно проводити операції з продажу та закупівлі товарів, формувати звіти, керувати даними з БД.

Ключові слова: JAVA, БАЗА ДАНИХ, СУБД, SQL, МОДЕЛЮВАННЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, АРХІТЕКТУРА, ORM-ІНСТРУМЕНТ, ДІАГРАМА, КЛАС.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А., м. Сєверодонецьк, 93400.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 АНАЛІЗ ЗАДАЧ ТОВАРООБІГУ	10
1.1 Аналіз предметної області.....	10
1.2 Короткий огляд існуючих систем управління підприємством.....	12
1.3 Аналіз роботи підприємства	15
1.3.1 Схема роботи підприємства	15
1.3.2 Бізнес правила	16
1.4 Постановка задачі.....	17
1.4.1 Специфікація вимог	17
1.4.2 Необхідні засоби розробки.....	19
2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ	21
2.1 Концептуальне проектування	22
2.1.1 Сутності та атрибути	22
2.1.2 Зв'язки між сутностями	23
2.2 Логічне проектування	24
2.3 Фізичне проектування.....	27
3 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМИ	28
3.1 Архітектура програмного забезпечення	28
3.2 Проектування класів	31
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
4.1 Засоби зберігання даних	37
4.2 Система управління базами даних	39
4.3 Object-relational mapping solutions	43
4.3.1 Збереженість даних	43
4.3.2 ORM – Object-relational mapping	45
4.4 Вибір ORM інструментів.....	46

4.4.1 Java Persistence API	46
4.4.2 Hibernate	47
4.4.3 MyBatis	48
4.4.4 Порівняння ORM інструментів.....	50
4.5 Інструкція по роботі з системою	51
4.5.1 Облік працівників підприємства	53
4.5.2 Облік товарів	55
4.5.3 Облік операцій.....	57
4.5.4 Звітність роботи підприємства	59
4.5.5 Зміна особистих даних.....	59
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	61
5.1 Вступ.....	61
5.2 Аналіз умов праці в офісі на робочому місці оператора ЕОМ.....	61
5.3 Промислова безпека у офісному приміщенні	66
5.4 Виробнича санітарія в офісі	69
5.5 Пожежна безпека в офісі	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	73
ДОДАТОК А Комп'ютерна презентація	75
ДОДАТОК Б Лістинг програми Main.java.....	754
ДОДАТОК В Лістинг програми Employeeer.java.....	75

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ЕОМ – електронно-обчислювальна машина

ІС – інформаційна система

СУБД – система управління базами даних

ООП – об'єктно орієнтоване програмування

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

API – application programming interface

EJB – enterprise java beans

CRUD – create, read, update, delete

DAO – data access object

DTO – data transfer object

BOM – business object model

JSF – java server faces

JDBC – java data base connectivity

ORM – object-relational mapping

POJO – plain old java object

RDBMS – relational database management system

ВСТУП

На даний час перед підприємствами та організаціями дуже гостро ставиться проблема автоматизації управління. Це пов'язано з кількома причинами. По-перше, зменшується рентабельність бізнесу організацій і підприємств, загострюється ринкова конкуренція. По-друге, змінюється співвідношення кількості торгових і виробничих компаній. Підприємства, які починають випуск будь-якої продукції, обов'язково стикаються з проблемами управління вже на початковому етапі своєї діяльності. По-третє, зростання компаній (кількості співробітників, поява регіональних філій, віддалених офісів, тощо), розширення асортименту товарів, кількості клієнтів і постачальників, також сприяють підвищенню складності бізнес-процесів і стимулюють керівників до пошуку інструментів, необхідних для управління. Четверта причина пов'язана з тим, що із завданнями впровадження автоматизованих систем управління підприємствами обов'язково стикаються підприємства, які бажають отримати закордонні інвестиції. Часто наявність автоматизованих систем управління підприємствами – це одна з вимог інвестора.

Для більшості сучасних організацій найбільш актуальними залишаються проблеми автоматизації процесів обліку, контролю та аналізу інформації, без вирішення яких неможливий подальший рух до ефективного управління.

Великі переваги автоматизації показує накопичений досвід автоматизації управління різними об'єктами і процесами. Деякі недоліки організації, обліку, збереження і обробки великих обсягів даних були виявлені вже на перших етапах використання обчислювальної техніки. Усунення цих недоліків можливо при використанні нового підходу до розробки та впровадженні інформаційних систем. Цей підхід пов'язаний з

організацією баз даних, які дозволяють накопичувати великі інформаційні масиви в систематизованій формі. У деяких випадках БД можуть служити закінченою системою, наприклад, автоматизованою інформаційно-довідковою системою або інформаційно-аналітичною системою.

У залежності від конкретної області застосування інформаційно-управляючі системи можуть дуже сильно відрізнятись за своїми функціями, архітектурою та реалізацією. Однак можна виділити, принаймні, дві властивості, які є загальними для всіх інформаційних систем: по-перше, будь-яка система призначена для збору, збереження та обробки інформації; по-друге, інформаційні системи орієнтовані на кінцевого користувача і тому зобов'язані мати простий, зручний, легко освоюваний інтерфейс.

Сьогодні інформаційні технології все глибше проникають в усі галузі промисловості та господарства. Це передбачає проведення комплексу організаційних, технічних, методичних та юридичних заходів щодо забезпечення всіх сфер виробництва та суспільства країни засобами обчислювальної техніки, що, в свою чергу, дозволить кардинально підвищити продуктивність праці, заощадити енергетичні, матеріальні, часові, трудові та природні ресурси і якісно перебудувати невиробничу сферу діяльності.

Поширеність сучасних комп'ютерних систем створює сприятливі умови для автоматизації ведення завдань промислового, комерційного та відомчого діловодства. Переважна більшість таких завдань пов'язана з урахуванням відомчої та службової інформації з метою її подальшої обробки. При цьому систематизація та обробка інформації в основному виконується вручну.

Інформаційні системи мають велике значення при вирішенні основних завдань управління підприємством:

- дослідження ринку;
- зниження витрат на виробництво;

- визначення стратегії і тактики руху підприємства.

Сприяючи ефективності вирішення перерахованих вище завдань, інформаційні системи здійснюють позитивний вплив на конкурентоспроможність підприємства.

Крім того, інформаційні системи використовуються для збільшення ефективності ключових факторів успіху, таких як:

- культура виробництва;
- стратегія;
- персонал;
- базовий процес;
- організація;
- технологія.

Таким чином, використання інформаційних систем пронизує процес виробництва, маркетингову діяльність підприємства, а також сприяє збільшенню швидкості і точності прийняття рішень.

Результативність господарювання багато в чому залежить від уміння працівників управління ретельно готувати й обґрунтовувати прийняті рішення. Для цього необхідно попередньо аналізувати стан керованого об'єкта (процесу), вивчати вплив діючих на нього факторів, взаємозв'язок з іншими об'єктами (процесами), прогнозувати можливі альтернативи його розвитку. Глибокий і всебічний аналіз економічних аспектів завжди повинен передувати розробці планових аспектів на підприємстві на плановий період. Основним джерелом інформації для такого аналізу на рівні підприємства є звітні та статистичні дані про стан і розвиток підприємства, його підрозділів.

Мета даної дипломної роботи – проектування та розробка інформаційної системи обліку товарів підприємства, що поширює побутову техніку.

1 АНАЛІЗ ЗАДАЧ ТОВАРООБІГУ

1.1 Аналіз предметної області

Одним з основних показників ефективності роботи підприємства є його продуктивність – товарообіг. Товарообіг – це обсяг продажів товарів організацією торгівлі в грошовому вираженні за певний період часу. Аналіз товарообігу дозволяє оцінювати відповідність наявних товарів попиту населення для вжиття заходів щодо оптимізації структури товарообігу, збільшення обсягу продажів, прискоренню оборотності товарів, ритмічності та рівномірності продажів. Товарообіг підрозділяється на оптовий, роздрібний [1].

Оптовий товарообіг – обсяг продажу товарів виробниками або торговими посередниками покупцям для подальшого використання в комерційному обороті. Обов'язкова ознака операції, яку відносять до оптової торгівлі – наявність рахунку-фактури.

Роздрібний товарообіг – це найважливіший показник торгівельного підприємства. Його можна розглядати як результат діяльності підприємства торгівлі, і від нього залежить обсяг валового доходу і прибутку, що характеризує ефективність торгової діяльності. Про значимість даного підприємства на ринку споживчих товарів можна судити за обсягом товарообігу, який виражається розміром грошової виручки за продані товари. Товарообіг має якісну та кількісну характеристики. Якісна пов'язана зі структурою товарообігу, тобто асортиментним складом, кількісна характеристика товарообігу – з обсягом реалізації в грошовому обчисленні.

Управління товарообігом являє собою обробку інформації, при цьому важливі такі параметри як якість, кількість і швидкість обробки інформації. Організації отримують початкову інформацію в різних видах: документи, що доставляють інформацію у вигляді слів і цифр; вербальна інформація; дані

від ЕОМ, часто в електронній формі. Кінцева інформація виробляється в таких же видах. Виробничий цикл організації може включати обробку інформації, зміну інформації, об'єднання даної інформації з іншою, накопичення інформації. У зв'язку з тим, що обсяги інформації постійно зростають, її обробка ускладнюється і виникає ситуація, коли людина не може самостійно здійснювати виробничий процес. У такому випадку явним вирішенням даної проблеми стає автоматизація обробки інформації [1].

Багато підприємств проходять три основних етапи автоматизації ділових процесів. На першому етапі відбувається впровадження персональних комп'ютерів (ПК). У процесі його освоєння з'ясовується, що за допомогою ПК можна вирішувати широке коло завдань, і, насамперед, автоматизувати бухгалтерський облік.

На другому етапі найчастіше підприємства набувають кілька комп'ютерів, призначаючи їх уже для інших потреб. Це можуть бути складська база даних або ж система обліку кадрів. Причому кожен ПК використовується і для інших цілей, наприклад для підготовки різних документів.

Потім з'ясовується необхідність перенесення інформації з одного комп'ютера на інший. Потреба в цьому виникає при встановленні більш сучасного програмного забезпечення, коли потрібно поєднати обробку бухгалтерських документів, кадрових даних, складської звітності. В обох випадках керівництво неминуче приходять до висновку про необхідність установки локальної обчислювальної мережі, що об'єднує ресурси всіх комп'ютерів фірми.

Таким чином, діяльність підприємства буде ефективною тільки при наявності загальної інформаційної системи, що об'єднує управління фінансами, персоналом, постачанням, збутом і (або) виробництвом [1].

1.2 Короткий огляд існуючих систем управління підприємством

«1С: Підприємство» – програмний продукт компанії «1С», призначений для автоматизації діяльності на підприємстві. Спочатку «1С: Підприємство» було призначено для автоматизації бухгалтерського та управлінського обліків (включаючи нарахування зарплати і управління кадрами), але сьогодні цей продукт знаходить своє застосування в областях, далеких від власне бухгалтерських завдань.

Технологічна платформа «1С: Підприємства» являє собою програмну оболонку над базою даних. Крім того, з версії 8.1 зберігання даних можливо в СУБД PostgreSQL і IBM DB2, а з версії 8.2 додалася і Oracle. Має свою внутрішню мову програмування, що забезпечує, крім доступу до даних, можливість взаємодії з іншими програмами за допомогою COM-з'єднання.

Клієнтська частина платформи функціонує в середовищі Microsoft Windows і, починаючи з версії 8.3, в середовищі Linux. Починаючи з версії 8.1, серверна частина платформи в клієнт-серверному варіанті роботи «1С: Підприємство» може функціонувати на ОС Linux.

Існують спеціальні версії середовища виконання «1С» для ноутбуків і PDA (Personal Digital Assistant), програмне забезпечення для створення веб-додатків, взаємодіючих з базою даних «1С: Підприємство».

«Concorde XAL» відноситься до класу систем управління підприємством і являє собою повністю інтегроване рішення. Число одночасно працюючих користувачів в системі – 400. «Concorde XAL» можуть одночасно використовувати клієнти, що працюють на різних платформах – від DOS і Windows до різних версій ОС UNIX. При цьому всі вони звертаються до єдиної БД. Система підтримує СУБД MS SQL Server, Oracle, Sybase, DB2. Для невеликих підприємств є варіант системи з власною вбудованою СУБД. Продукт заснований на клієнт-серверних технологіях, в нього вбудована апаратно-незалежна мова 4-го покоління XAL (eXtended

Application Language), що забезпечує гнучке розширення системи, відповідно до мінливих вимог бізнесу.

До складу «Concorde XAL» входять усі обов'язкові для системи модулі – від «Управління виробництвом» до «Управління персоналом». Особливість системи - детальне розбиття її можливостей на окремі блоки. Це важливо для невеликих підприємств, які на кожному етапі свого розвитку можуть придбати тільки ту функціональну частину, яка відповідає потребам сьогодення.

«Фінанси і виробництво». Основною метою є забезпечення менеджерів всією необхідною інформацією для швидкого прийняття рішення та оперативного реагування на ситуацію, що склалася. Система підтримує такі операції, як кредитування, закупівлю, транспортування, переробку, бартерний обмін, продаж та ін. Система повністю інтегрована з модулями бюджетування та бухгалтерського обліку. В якості БД використовуються такі корпоративні СУБД, як Oracle і Progress, що дозволяють працювати в багаторівневій архітектурі клієнт-сервер.

«BMS - Бухгалтерія». Система автоматизації бухгалтерського обліку є комплексним рішенням для комп'ютеризації багатовалютного бухгалтерського обліку на підприємстві. Крім основного, вона включає модуль основних засобів, заробітної плати, управління кадрами та ін. Система дозволяє підтримувати план рахунків, що враховує специфіку діяльності компанії, володіє гнучкими інструментами для роботи з бухгалтерією по кожному рахунку (субрахунку), дає можливість отримувати інформацію в будь-яких заданих умовах. Система дозволяє генерувати необхідні для податкової інспекції звіти у форматі Excel.

Система комплексної автоматизації роботи підприємства SAP R/ 3. SAP R / 3 являє собою інтегровану модульну систему, що охоплює практично всі сторони діяльності підприємства – від управління матеріальними і фінансовими потоками до управління персоналом та підготовкою даних для прийняття рішень менеджером вищого ешелону. Призначена для середніх,

великих підприємств, транснаціональних компаній і дозволяє побудувати єдину інформаційну інфраструктуру, що охоплює практично всі аспекти діяльності об'єкта управління.

SAP R/3 будується на основі трьохрівневої технології клієнт-сервер:

1. Сервер БД. Забезпечує актуалізацію даних, їх захист і повноту. Як сервер використовується СУБД.
2. Сервер представлення даних. На цьому рівні вирішуються завдання проміжного представлення даних.
3. Рівень взаємодії з користувачем. На цьому рівні вирішуються завдання користувача інтерфейсу. Він розроблений з урахуванням оптимізації кількості часу, необхідного для введення-виведення інформації і являє собою графічну оболонку.

Має модульну структуру. Основні компоненти системи можна розбити на наступні підсистеми:

1. Базовий модуль.
2. Планування виробництва.
3. Фінанси та бухгалтерія.
4. Продажі та дистрибуція.
5. Планування виробництва та контрольні-інформаційні системи.
6. Офіс і комунікації.
7. Система якості.
8. Управління матеріальними потоками.
9. Персонал.
10. Управління основними фондами.
11. Підтримка виробництва.
12. Управління обладнанням.
13. Система проектів.
14. Індустріальні рішення.

1.3 Аналіз роботи підприємства

1.3.1 Схема роботи підприємства

Робота підприємства включає наступні етапи:

1. Поставка товарів від постачальника. Включає оформлення замовлення на продукцію, отримання рахунку, безпосередньо поставку продукції та оплату поставки.

2. Реалізація товарів. Включає отримання та оформлення замовлення від покупця, розрахунок вартості замовлення, відправка товару, прийняття оплати.

3. Можливість повернення товарів від покупця. У випадку пошкодження товару при транспортуванні або при виробничому браку замовник має право повернути товар.

4. Можливість повернення товарів постачальнику. У випадку пошкодження товару при транспортуванні або при виробничому браку підприємство повинно мати можливість повернути товар постачальнику.

5. Оформлення фінансових документів. Включає облік замовлень, ведення бухгалтерських операцій та оформлення звітів.

6. Інвентаризація товарів. Включає запис товарів по завершенню поставки та списання товарів по завершенню операції з продажу.

На рисунку 1.1 зображена функціональна схема роботи підприємства.

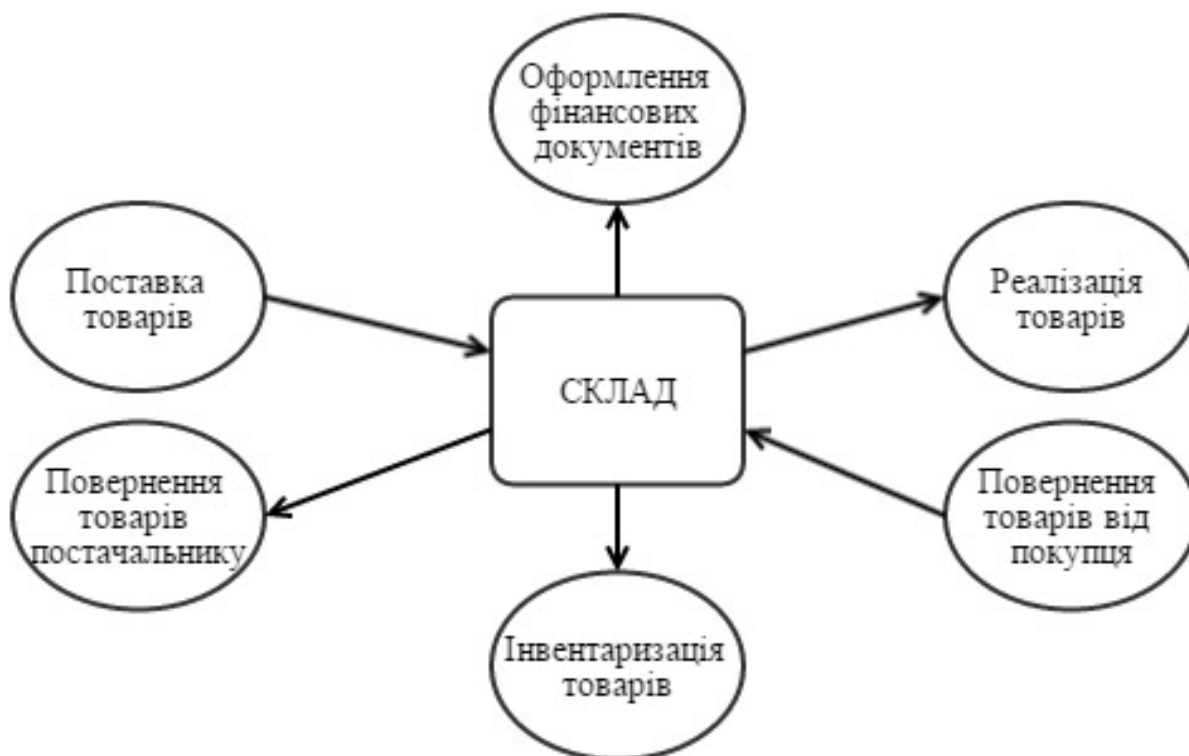


Рисунок 1.1 – Функціональна схема роботи підприємства

1.3.2 Бізнес правила

Бізнес правила мають наступний вигляд:

- усі товари поділяються за типами та підтипами;
- усі ціни вказані у гривнях;
- ціна продажу складає 120% від ціни закупівлі;
- дата проведення операції продажу або закупівлі відповідає даті створення операції у системі;
- у одній операції продажу або закупівлі може бути необмежена кількість видів товару;
- кількість одного товару в операції необмежена;
- підприємство може закуповувати один товар у декількох поставників;

- якщо на складі є товар, закуплений у різних поставників з різною вартістю, то спочатку продається дорожчий товар;
- якщо в одній операції з продажу товару міститься товар з різною ціною, тоді одиниці з меншою ціною продаються у вигляді акції;
- на підприємстві існують такі посади: системний адміністратор, менеджер з продажу та управління персоналом, бухгалтер, оператор з продажу, комірник, вантажник;
- працівник на посаді адміністратора має доступ до всіх функцій системи, а також доступ до бази даних;
- працівник на посаді менеджера має доступ до всіх функцій системи;
- працівник - бухгалтер має доступ до функцій перегляду інформації персоналу організації, товарів, автоматизованих звітів, а також має можливість змінювати власні дані;
- працівник на посаді оператора має доступ до функцій перегляду товарів, перегляду та оформленню операцій з продажу та закупівлі, а також має можливість змінювати власні дані;
- працівник на посаді комірник має доступ до функцій перегляду та зміни товарів, а також має можливість змінювати власні дані;
- працівник на посаді вантажник має доступ до функцій перегляду товарів та зміни власних даних.

1.4 Постановка задачі

1.4.1 Специфікація вимог

Основні предметно-значущі сутності: постачальники, покупці, товар, персонал, операція.

Основні предметно-значущі атрибути сутностей:

- постачальник/покупець: ім'я, адреса, телефон, поштова скринька, розрахунковий рахунок;
- товар: найменування, ціна продажу, ціна закупки, кількість, тип, опис, бренд;
- тип товару: назва, підтипи, батьківський тип;
- працівник: ім'я, посада;
- операція: номер, тип операції, постачальник/покупець, адреса доставки, загальна сума, список товарів.

Програмне забезпечення повинне автоматизувати наступні дії:

1. Облік співробітників підприємства:

- перегляд інформації про персонал з можливістю сортування за посадою;
- додавання нових співробітників;
- редагування поточних працівників.

2. Облік товарів:

- перегляд усього переліку товарів з можливістю сортування по типу;
- додавання нових товарів;
- редагування існуючих товарів;
- додавання нових типів і підтипів товару.

3. Облік операцій:

- перегляд всіх поставок і продажів з можливістю сортування за датою;
- перегляд докладної інформації про постачання або продаж;
- оформлення угоди з постачання або продажу товару.

4. Звітність роботи підприємства:

- підрахунок прибутку підприємства за рік і за місяць;
- визначення п'яти найпопулярніших товарів за рік і за місяць;
- визначення кращого співробітника (оператора) року і місяця.

5. Зміна особистих даних:

- імені та прізвища;

– логіна і пароля.

1.4.2 Необхідні засоби розробки

Для проектування та реалізації програмного забезпечення необхідно використовувати наступні програмні засоби: Gliffy Diagrams, Oracle Database (Oracle RDBMS), Oracle SQL Developer, мова програмування JAVA, ORM, JSF (PrimeFaces), Maven, IntelliJ IDEA, Apache Tomcat.

Gliffy Diagrams – online-додаток для проектування. Моделі даних допомагають візуалізувати структуру даних, забезпечуючи ефективний процес організації, управління і адміністрування.

Oracle Database (Oracle RDBMS) – об'єктно-реляційна система управління базами даних компанії Oracle.

Oracle SQL Developer – інтегроване середовище розробки на мовах SQL і PL/SQL, орієнтоване на застосування в середовищі Oracle Database.

JAVA – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Синтаксис мови багато в чому схожий на C та C++. У офіційній реалізації Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

ORM – інструмент, який зберігає об'єкти програми в реляційній базі даних. ORM надає функціональні можливості виконувати CRUD операції.

PrimeFaces – фреймворк для веб-додатків, завдяки якому є можливість створювати простий, зрозумілий та функціональний інтерфейс користувача.

Maven – це засіб автоматизованої роботи з програмними проектами. Використовується для управління та складання програм, створений

Джейсоном ван Зилом у 2002 році. У діяльності подібний до Apache Ant, але має простіший вигляд build-налаштувань, який базується на форматі XML.

IntelliJ IDEA – комерційне інтегроване середовище розробки від компанії JetBrains. Підтримує інструменти для проведення тестування JUnit, системи контролю версій Subversion, Mercurial і Git, засоби складання Maven і Ant, мови програмування Java, Scala, Clojure, Groovy. До складу входить XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проектів з Eclipse.

Apache Tomcat – контейнер сервлетів, розроблений Apache Software Foundation. Повністю написаний мовою програмування Java та реалізує специфікацію сервлетів і Java Server Pages від Sun Microsystems, що є стандартами для розробки веб-додатків на Java. Tomcat дозволяє запускати веб-додатки, містить ряд програм для самоконфігурування. Tomcat використовується в якості самостійного веб-сервера

2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

Проектування баз даних – процес створення схеми бази даних і визначення необхідних обмежень цілісності [2].

Основні завдання:

- забезпечення зберігання в БД всієї необхідної інформації;
- забезпечення можливості отримання даних по всім необхідним запитам;
- скорочення надмірності і дублювання даних;
- забезпечення цілісності бази даних.

Основні етапи проектування баз даних:

- концептуальне проектування;
- логічне проектування;
- фізичне проектування.

Концептуальне проектування – побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. Така модель є як образом реальності, так і образом проєктованої бази даних для цієї реальності [2].

Логічне проектування – створення схеми бази даних на основі конкретної моделі даних, наприклад, реляційної моделі даних. Для реляційної моделі даних модель є набором схем відносин, зазвичай із зазначенням первинних ключів, а також «зв'язків» між відносинами, що представляють собою зовнішні ключі. На етапі логічного проектування враховується специфіка конкретної моделі даних, але може не враховуватися специфіка конкретної СУБД [2].

Фізичне проектування – створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може включати в себе обмеження на іменування об'єктів бази даних та обмеження на підтримувані типи даних.

Крім того, специфіка конкретної СУБД при фізичному проектуванні включає вибір рішень, пов'язаних з фізичним середовищем зберігання даних [2].

2.1 Концептуальне проектування

Концептуальна модель – це відображення предметної області, для якої розробляється база даних. Це діаграма з прийнятими позначеннями елементів: об'єкти, які позначають речі, позначаються у вигляді прямокутника, атрибути, які характеризують об'єкт позначаються у вигляді овалу, зв'язки між об'єктами позначаються у вигляді ромбу.

2.1.1 Сутності та атрибути

При побудові концептуальної моделі бази даних визначено наступні шість сутностей: «Поставщик», «Покупатель», «Товар», «Продажа», «Поставка», «Работник». Кожна з таких сутностей сильна, тобто не залежить від інших сутностей.

Сутність «Поставщик» має такі атрибути: Імя, Адрес, Телефон, Почта, Счет. Первинним ключем цієї сутності є атрибут «Імя».

Сутність «Покупатель» має такі атрибути: Імя, Адрес, Телефон, Почта, Счет. Первинним ключем цієї сутності є атрибут «Імя».

Сутність «Товар» має такі атрибути: Название, Цена продажи, Цена покупки, Тип, Количество. Первинним ключем цієї сутності є атрибут «Название».

Сутність «Продажа» має такі атрибути: Номер, Дата, Сумма, Адрес доставки. Первинним ключем цієї сутності є атрибут «Номер».

Сутність «Поставка» має такі атрибути: Номер, Дата, Сумма. Первинним ключем цієї сутності є атрибут «Номер».

Сутність «Работник» має такі атрибути: Имя, Должность. Первинним ключем цієї сутності є атрибут «Имя».

2.1.2 Зв'язки між сутностями

Між сутністю «Поставщик» та сутністю «Поставка» є зв'язок 1:N, так як один постачальник може постачати товар необмежену кількість разів.

Між сутністю «Поставка» та сутністю «Товар» є зв'язок M:N, так як в одній операції може бути необмежена кількість товарів, а один вид товару може постачатися необмежену кількість разів.

Між сутністю «Продажа» та сутністю «Товар» є зв'язок N:M, так як в одній операції може бути необмежена кількість товарів, а один вид товару може продаватися необмежену кількість разів.

Між сутністю «Покупатель» та сутністю «Продажа» є зв'язок 1:N, так як один покупець може придбати товар необмежену кількість разів, а одна операція з продажу оформлюється на одного покупця.

Сутність «Работник» не пов'язана з іншими сутностями, так як вона створена для зберігання даних про персонал підприємства.

На рисунку 2.1 зображена концептуальна модель бази даних.

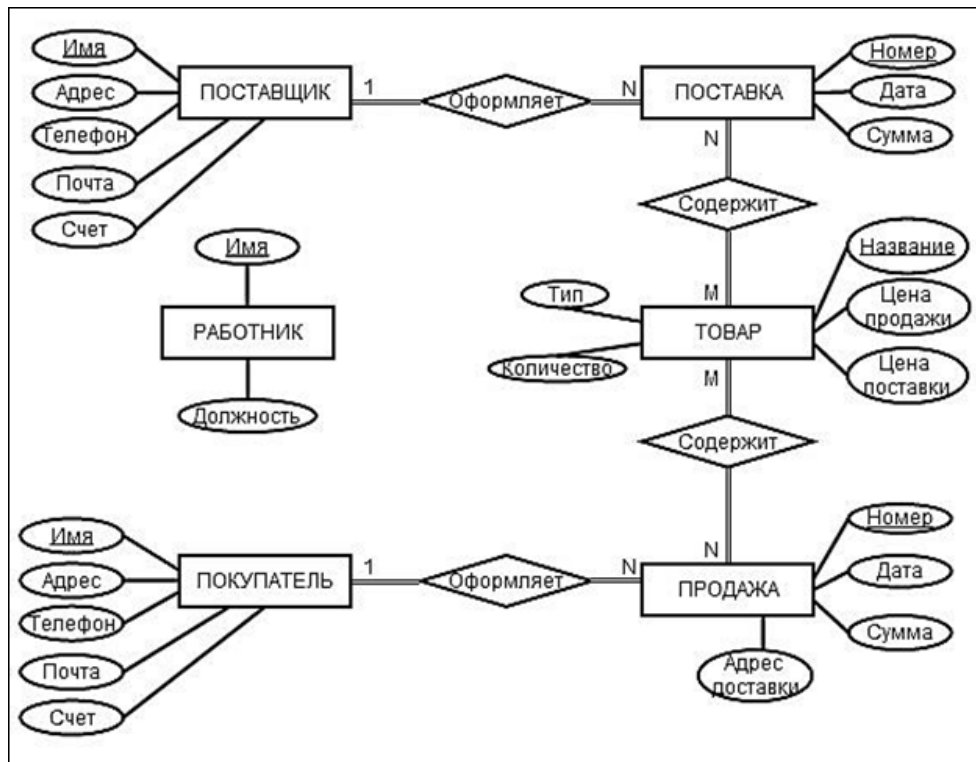


Рисунок 2.1 – Концептуальна модель БД

2.2 Логічне проектування

Побудована концептуальна модель має деякі недоліки, а саме:

- зв'язки N:M – між об'єктами «Продажа» і «Товар», та «Поставка» і «Товар» необхідно вводити проміжну таблицю;
- у об'єкті «Работник» є атрибут «Должность», який буде дублюватися для працівників на однакових посадах, для усунення цього недоліку необхідно створити ще одну таблицю, яка буде містити всі можливі посади;
- атрибут «Название» об'єкта «Товар» містить назву бренду (виробника), необхідно створити ще одну таблицю, яка буде містити всі бренди;
- дана модель БД не може забезпечити виконання бізнес правила, яке говорить, що підприємство може закуповувати один товар у декількох поставників. Треба забезпечити зберігання даних про товар від різних

поставників, для цього необхідно створити таблицю зв'язку поставника та товару, в якій зберігати ціну, а інші, загальні характеристики товару залишити в таблиці «Товар»;

- записи у кожній таблиці повинні бути унікальними та мати власний ідентифікаційний номер, для цього необхідно внести додатковий атрибут до всіх таблиць – ID_НазваТаблиці;

- кожний тип товару може містити підтипи. Для збереження ієрархії типів необхідно зробити посилання з таблиці «Тип» на саму себе.

На рисунку 2.2 зображена логічна модель БД з виправленням недоліків.

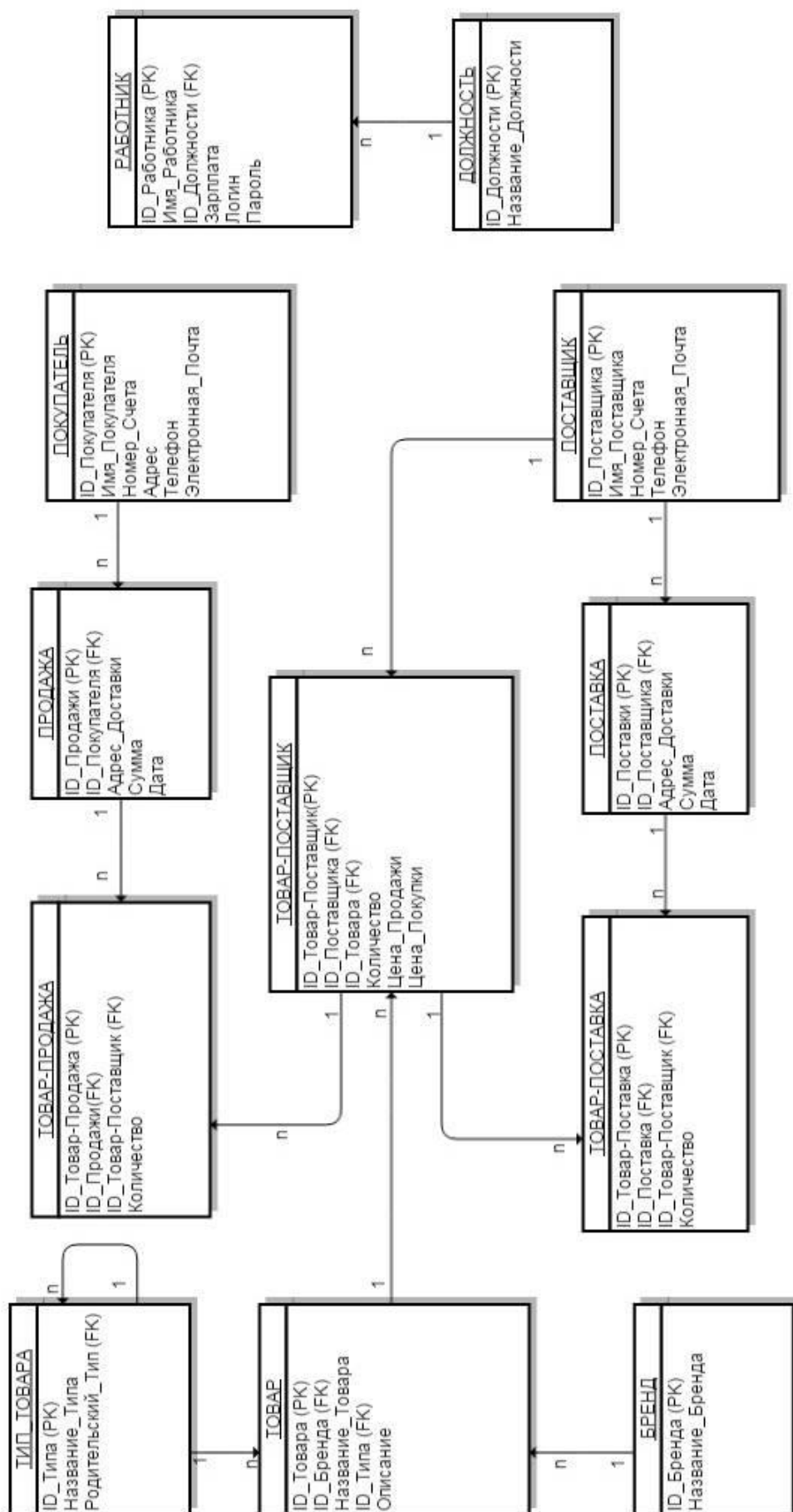


Рисунок 2.2 – Модель логічного проектування БД

2.3 Фізичне проектування

На рисунку 2.4 зображена схема БД з переведеними назвами та атрибутами таблиць, а також до кожної таблиці додані атрибути CREATE_DATE, CREATED_BY та (або) UPDATE_DATE, UPDATED_BY для збереження інформації про створення та зміну запису.

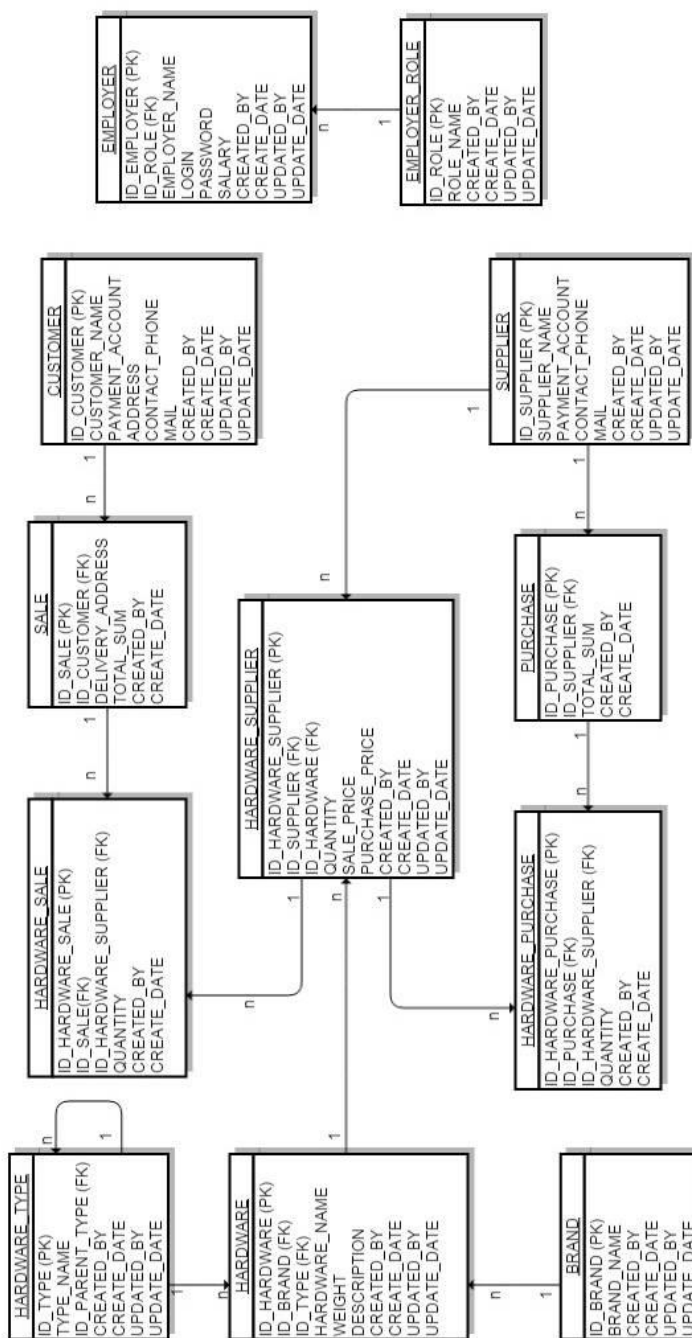


Рисунок 2.3 – Модель фізичного проектування БД

3 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМИ

3.1 Архітектура програмного забезпечення

Архітектура програмного забезпечення – це структура програми або обчислювальної системи, яка містить програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними [3].

Проектування архітектури програмного забезпечення – це процес розроблення, що виконується після етапу аналізу та формулювання вимог. Завдання такого проектування – перетворення вимог до системи у вимоги до програмного забезпечення та побудова на їхній основі архітектури системи. Побудова архітектури системи здійснюється шляхом визначення цілей системи, її вхідних і вихідних даних, декомпозиції системи на підсистеми, компоненти або модулі та розроблення її загальної структури [4].

Програмне забезпечення, що розроблюється, складається з шести взаємозв'язаних модулів (рис. 3.1).

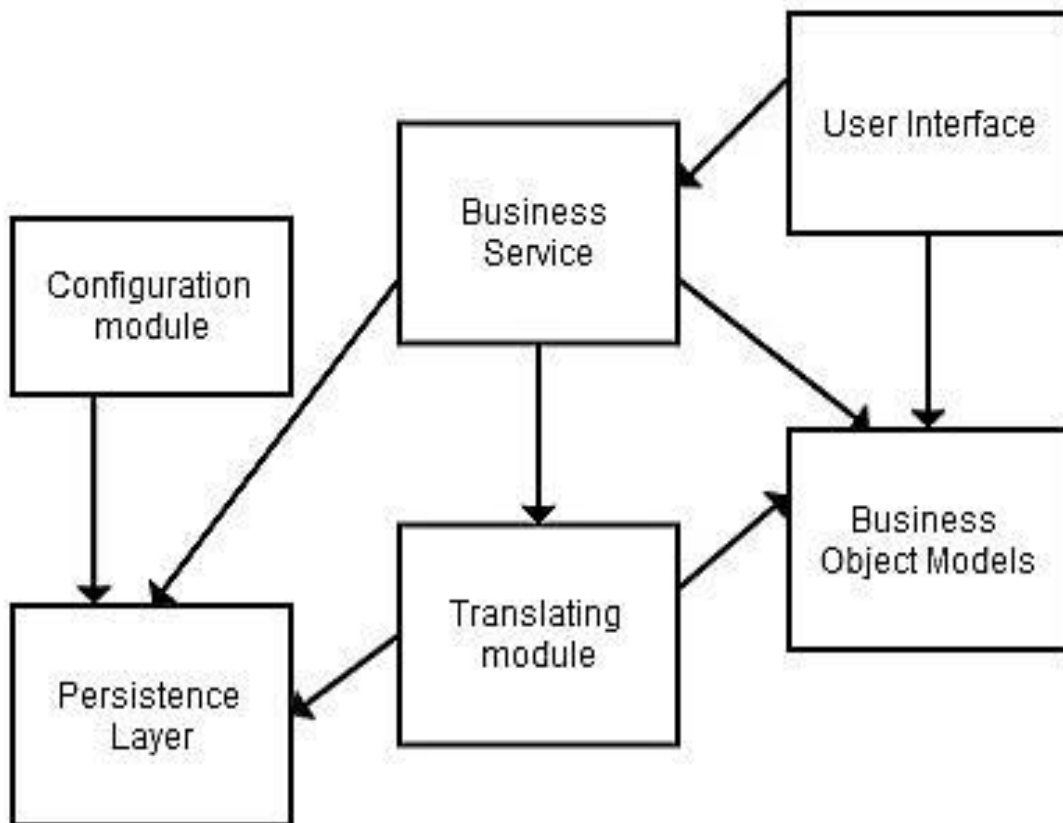


Рисунок 3.1 – Схема архітектури програмного забезпечення

Business object models. Це об'єкти у об'єктно-орієнтованих комп'ютерних програмах, які відображають сутності з моделі предметної області, що відносяться до програмного забезпечення та реалізують бізнес логіку програми. Такі об'єкти інкапсулюють усю необхідну для програми інформацію про модель предметної області. Об'єкти бізнес логіки представлені у вигляді класів об'єктно-орієнтованого програмування, які описують реальні сутності предметної області та їх поведінку. До модулю мають доступ наступні модулі: Translating module, Business service, User interface module.

Persistence layer. Модуль підключення та взаємодії з базою даних. Включає об'єкти DTO та DAO. Data Transfer Objects – об'єкти, які відображають сутності бази даних. Це один з шаблонів проектування, що використовується для передачі даних між підсистемами програмного забезпечення [4]. Data Transfer Object на відміну від Business Object Models

або Data Access Object не повинен містити будь-якої поведінки. Data Access Object – об'єкт оперування даними у базі даних, який надає абстрактний інтерфейс до будь-якого типу бази даних або механізму зберігання [4]. Забезпечує виконання CRUD (Create, Read, Update, Delete) операцій. Деякі можливості надаються незалежно від того, який механізм зберігання використовується, та без необхідності спеціальним чином відповідати цьому механізму зберігання. До модулю мають доступ наступні модулі: Translating module, Business service, Configuration module, останній забезпечує підключення до БД.

Translating module. Перетворювач об'єктів, що описують сутності бази даних (Data Transfer Objects) до об'єктів бізнес логіки (Business Object Models), та навпаки. Такі об'єкти не містять бізнес логіку, а лише переписують усі дані з об'єктів БД [4] до відповідних об'єктів модулю Business Object Models. Модуль існує для відділення логіки обробки даних від бази даних та взаємодіє з Persistence layer, Business object models та Business service.

Business service. Шар програмного забезпечення, який містить усі найважливіші функції взаємодії об'єктів бізнес логіки та реалізацію бізнес правил. Цей модуль оперує всіма об'єктами та функціями з попередніх модулів (Business object models, Persistence layer, Translating module).

User interface module. Модуль інтерфейсу користувача. Складається з розмітки web-сторінок, стилів web-сторінок та класів їх обробки. Цей модуль може взаємодіяти тільки з модулями Business service та Business object models.

Configuration module. Модуль, що містить усі конфігурації програми, а саме налаштування підключення до бази даних, конфігурацію контексту Spring, створення та заповнення віртуальної тестової бази даних та запуску тестів програми.

3.2 Проектування класів

В об'єктно-орієнтованому програмуванні клас – це спеціальна конструкція, яка використовується для групування пов'язаних змінних та функцій. Згідно з термінологією об'єктно-орієнтованого програмування глобальні змінні класу називаються полями даних, властивостями або атрибутами, а функції називають методами класу. Створений та ініціалізований екземпляр класу називають об'єктом класу. На основі одного класу можна створити множину об'єктів, що відрізнятимуться один від одного своїм станом, тобто значеннями полів [3].

На рисунку 3.2 зображена діаграма основних класів, які описують об'єкти, що приймають участь у бізнес процесі підприємства. Дані класи є частиною модуля Business object models.

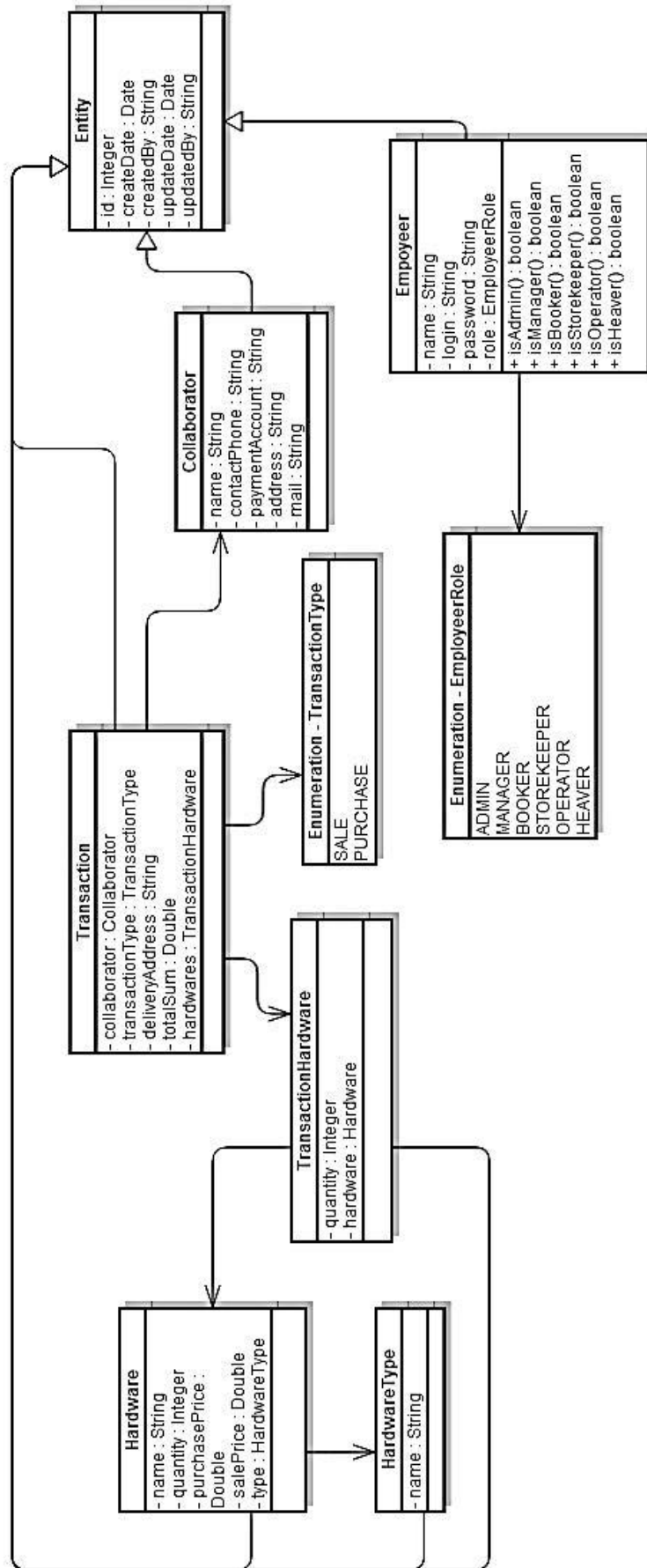


Рисунок 3.2 – Діаграма класів «Business object models»

На рисунку 3.3 зображена діаграма основних класів, що описують взаємодію об'єктів, їх обробку та маніпулювання, необхідну для забезпечення працездатності підприємства. Дані класи є частиною модуля Business service.

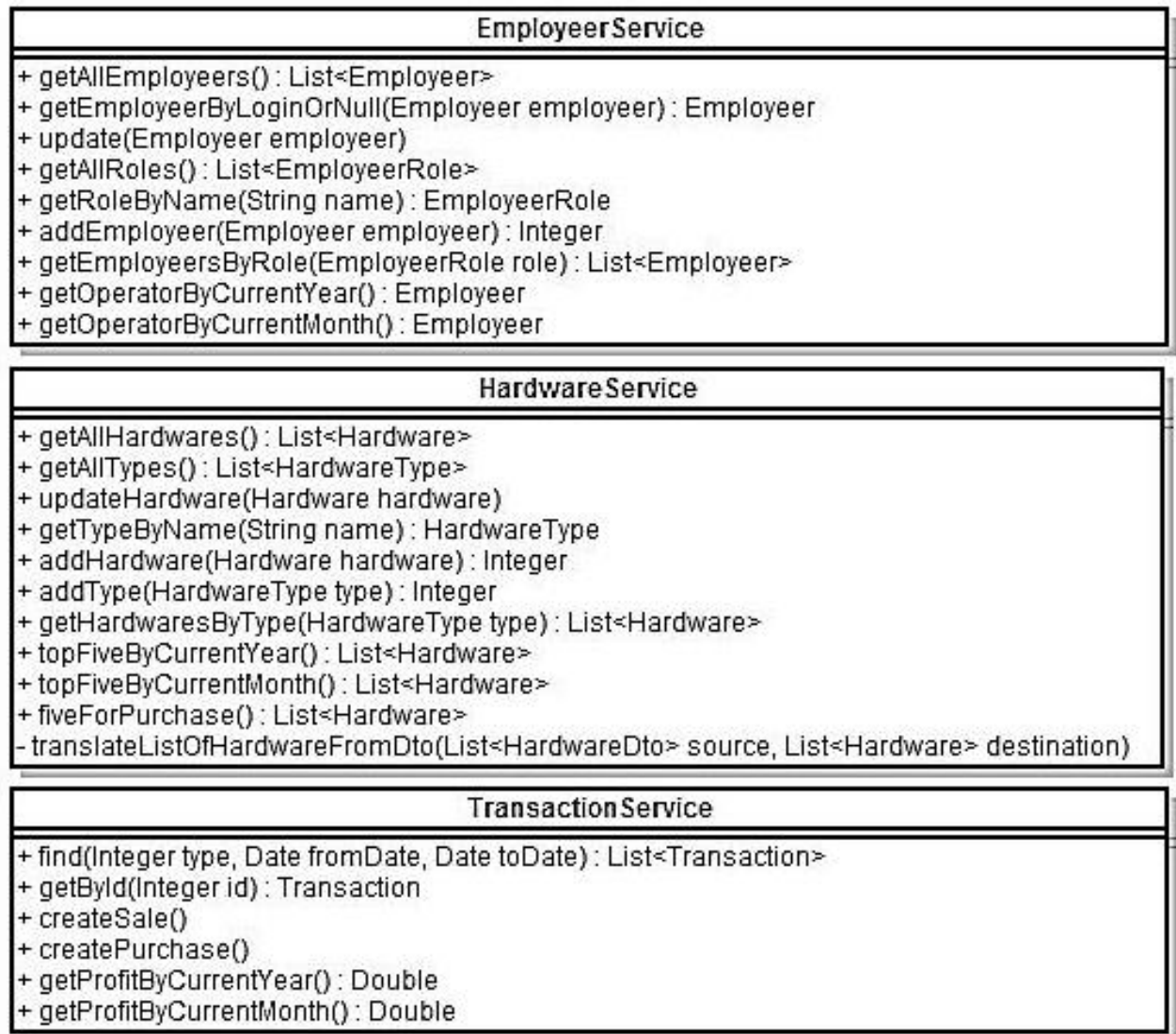


Рисунок 3.3 – Діаграма класів «Business service»

На рисунку 3.4 зображена діаграма класів, які відображують об'єкти бази даних (Data transfer objects). Ці класи є частиною модулю Persistence module.

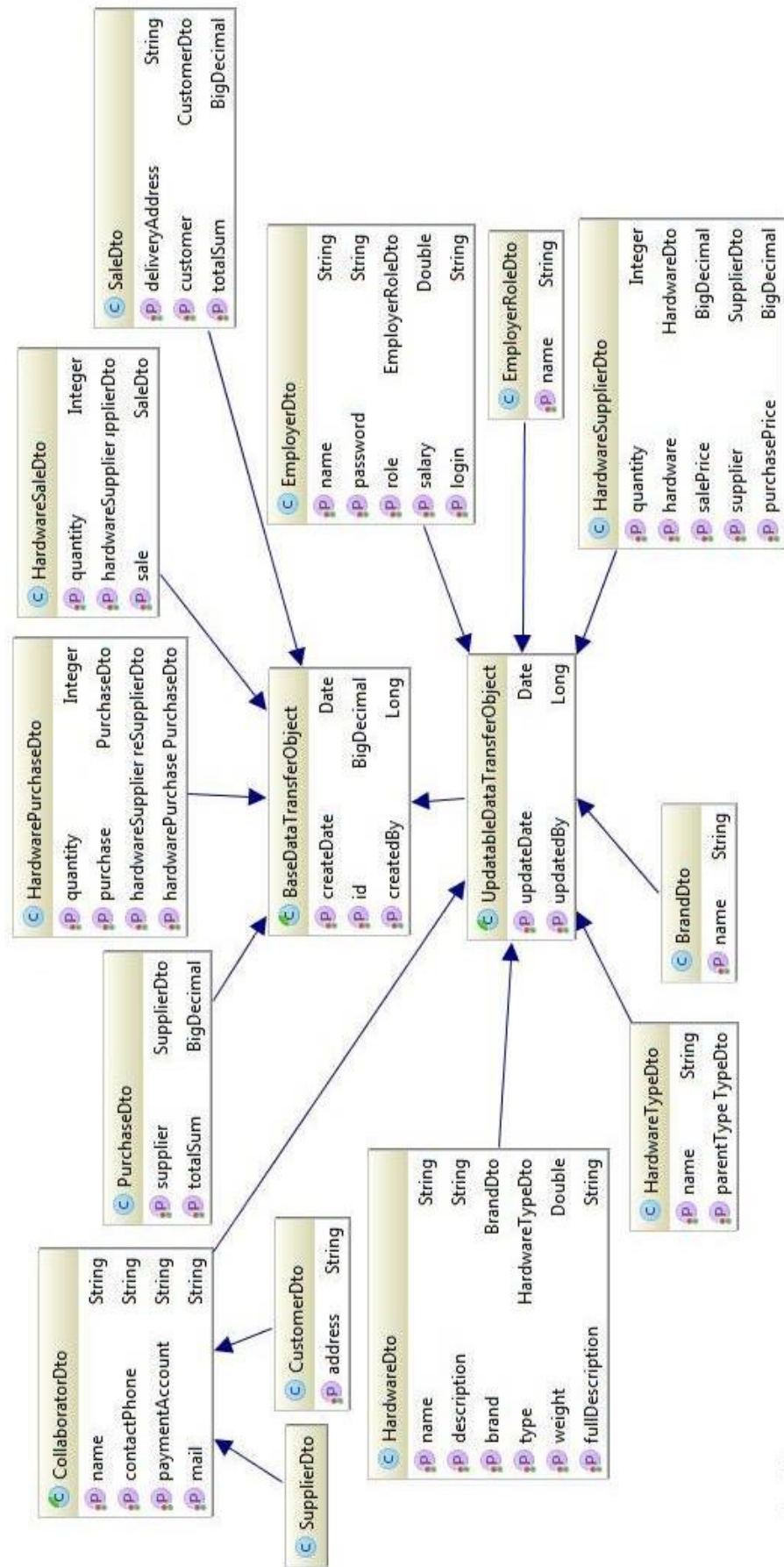
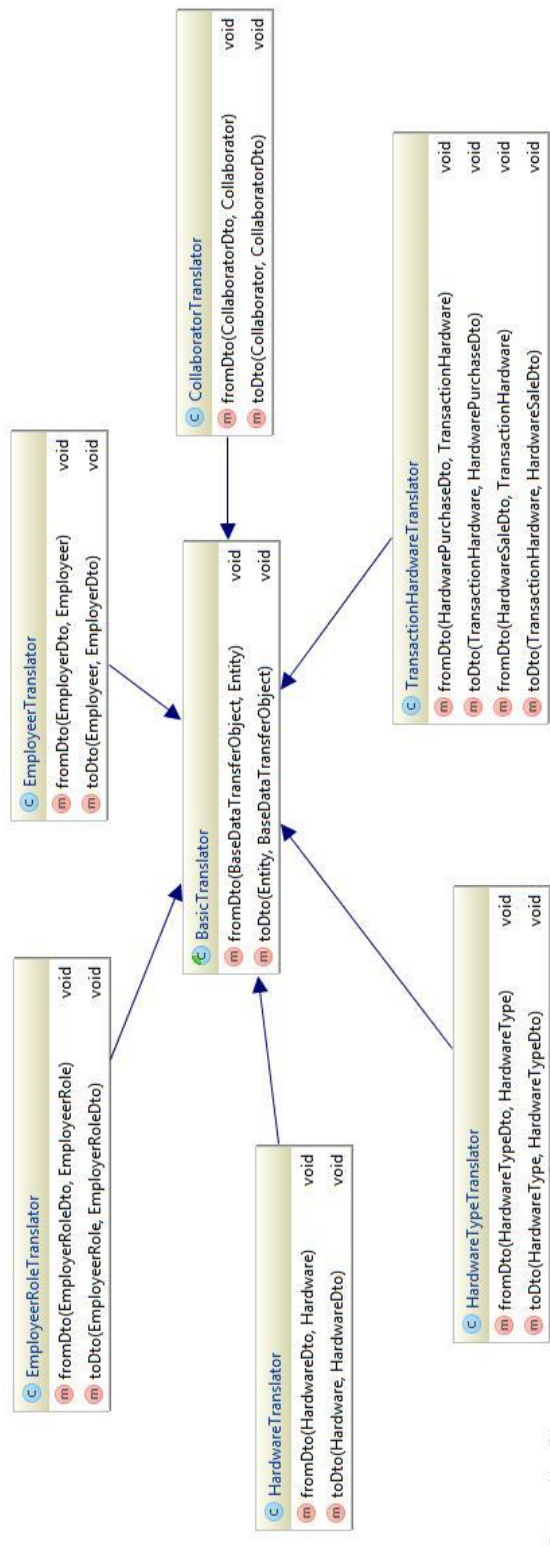


Рисунок 3.4 – Діаграма класів «Persistence layer»

На рисунку 3.5 зображена діаграма класів, які відображують об'єкти перетворення DTO до об'єктів ВОР. Ці класи є частиною модулю Translating module.



Powered by yFiles

Рисунок 3.5 – Діаграма класів «Translating module»

На рисунку 3.6 зображена діаграма класів, які відповідають за взаємодію інтерфейсу користувача с серверною частиною програми. Ці класи є частиною модулю User interface module.

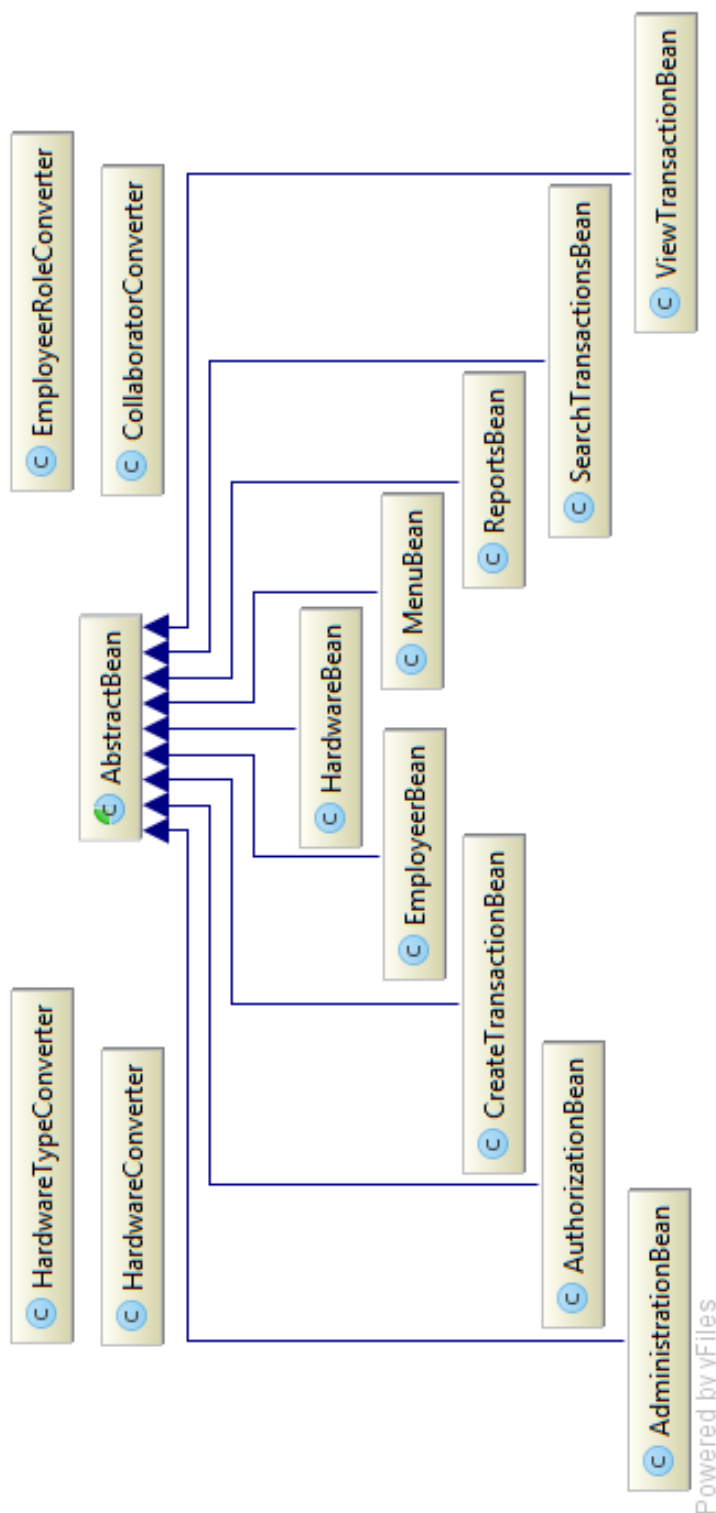


Рисунок 3.6 – Діаграма класів «User interface module»

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Засоби зберігання даних

База даних – організована відповідно до певних правил і підтримувана в пам'яті комп'ютера сукупність даних, що характеризує актуальний стан деякої предметної області та використовується для задоволення інформаційних потреб користувачів [5].

Завдання тривалого зберігання і обробки інформації з'явилося практично відразу з появою перших комп'ютерів. Для вирішення цього завдання в кінці 60-х років були розроблені спеціалізовані програми, що отримали назву систем управління базами даних. СУБД проробили тривалий шлях еволюції від системи управління файлами, через ієрархічні і мережеві бази даних. Наприкінці 80-х років домінуючою стала система керування реляційними базами даних (СУРБД). З цього часу такі СУБД стали стандартом де-факто, і для того, щоб уніфікувати роботу з ними, була розроблена структурована мова запитів (SQL), яка являє собою мову управління саме реляційними базами даних [5].

Існують такі різновиди баз даних:

- ієрархічні;
- реляційні;
- об'єктно-орієнтовані;
- гібридні.

Ієрархічна база даних заснована на деревовидній структурі зберігання інформації. У цьому сенсі ієрархічні бази даних дуже нагадують файлову систему комп'ютера.

В реляційних базах даних дані зібрані в таблиці, які в свою чергу складаються із стовпців і рядків. Запити до таких баз даних повертають таблицю, яка повторно може брати участь у наступному запиті. Дані в одних

таблицях, як правило, пов'язані з даними інших таблиць, звідки і пішла назва "реляційні".

В об'єктно-орієнтованих базах даних дані зберігаються у вигляді об'єктів. З об'єктно-орієнтованими базами даних зручно працювати, застосовуючи об'єктно-орієнтоване програмування. Однак на сьогоднішній день такі бази даних ще не досягли популярності реляційних, оскільки поки значно поступаються їм у продуктивності.

Гібридні БД поєднують в собі можливості реляційних і об'єктно-орієнтованих баз даних.

У web-програмах, як правило, використовуються реляційні бази даних.

Реляційний підхід до організації баз даних був закладений в кінці 1960-х рр. Едгаром Коддом. Реляційна база даних - база даних, заснована на реляційній моделі даних (логічна модель даних, прикладна теорія побудови баз даних, яка є додатком до завдань обробки даних таких розділів математики як теорії множин і логіки першого порядку) [6].

Перевагами реляційного підходу є такі властивості [2]:

- реляційний підхід ґрунтується на невеликому числі інтуїтивно зрозумілих абстракцій, на основі яких можливе просте моделювання найбільш поширених предметних областей;
- ці абстракції можуть бути точно і формально визначені;
- теоретичним базисом реляційного підходу до організації баз даних служить простий і потужний математичний апарат теорії множин і математичної логіки;
- реляційний підхід забезпечує можливість ненавігаційного маніпулювання даними без необхідності знання конкретної фізичної організації баз даних у зовнішній пам'яті.

Значення даних, що зберігаються в реляційній базі даних, є типізованим, тобто є відомим тип кожного збереженого значення. Поняття типу даних в реляційній моделі даних повністю відповідає поняттю типу даних в мовах програмування [6].

4.2 Система управління базами даних

Система управління базами даних – це програмне забезпечення, за допомогою якого користувачі можуть визначати, створювати і підтримувати базу даних, а також здійснювати до неї контрольований доступ [2].

Основні функції СУБД:

- управління даними у зовнішній пам'яті (на дисках);
- управління даними в оперативній пам'яті з використанням дискового кеша;
- журналізація змін, резервне копіювання і відновлення бази даних після збоїв;
- підтримка мов БД (мова визначення даних, мова маніпулювання даними).

Зазвичай сучасна СУБД містить наступні компоненти:

- ядро, яке відповідає за управління даними у зовнішній і оперативній пам'яті і журналізацію;
- процесор мови бази даних, що забезпечує оптимізацію запитів на вилучення та зміну даних і створення, як правило, машинно-незалежного виконуваного внутрішнього коду;
- підсистему підтримки виконання, яка інтерпретує програми маніпуляції даними, що створюють користувальницький інтерфейс із СУБД.

Для виконання даної роботи в якості СУБД обрана Oracle Database (або Oracle RDBMS) – об'єктно-реляційна система управління базами даних компанії Oracle. Oracle займає лідируючі позиції на ринку СУБД і, що особливо важливо, лідирує на платформах Unix і Windows.

Причина широкої поширеності Oracle полягає насамперед у високих експлуатаційних характеристиках СУБД, наявності підтримуючої інфраструктури – навчальних центрів, широкої мережі партнерів Oracle, підтримкою більшості платформ [5].

З технічної точки зору важливо те, що Oracle функціонує практично на всіх існуючих комп'ютерних платформах, в тому числі і на великих ЕОМ (OS/390), різних різновидах Unix, в тому числі Solaris, Linux.

Іншою важливою характеристикою є підтримка Oracle всіх можливих варіантів архітектур, у тому числі симетричних багатопроцесорних систем, кластерів, систем з масовим паралелізмом. Очевидна значимість цих характеристик для систем масштабу корпорації, де експлуатується безліч комп'ютерів різних моделей та виробників. В таких умовах фактором успіху є максимально можлива типізація пропонованих рішень, що ставить своєю метою істотне зниження вартості володіння програмним забезпеченням. Уніфікація систем управління базами даних - один з найбільш значущих кроків на шляху досягнення цієї мети.

Ядром СУБД Oracle є сервер бази даних, який поставляється в одному з чотирьох варіантів залежно від масштабу інформаційної системи, в рамках якої передбачається його застосування. Для систем масштабу великої організації пропонується продукт Oracle Database Enterprise Edition (корпоративна редакція), для якого є цілий набір опцій, що архітектурно і функціонально розширюють можливості сервера. Продукт Oracle Database Standard Edition (стандартна редакція) орієнтований на організації середнього масштабу або підрозділу в складі великої організації. Для персонального використання призначений продукт Oracle Database Personal Edition (персональна редакція) [6].

Найважливішою перевагою Oracle перед конкурентами є ідентичність коду різних версій сервера баз даних Oracle для всіх платформ, що гарантує ідентичність і передбачуваність роботи Oracle на всіх типах комп'ютерів, які б не входили до її складу. Всі варіанти сервера Oracle мають у своїй основі один і той же вихідний програмний код і функціонально ідентичні, за винятком деяких опцій, які, наприклад, можуть бути додані до Oracle Database Enterprise Edition і не можуть – до Oracle Database Standard Edition. Таким чином, для всіх платформ існує єдина СУБД в різних версіях, яка

поводиться однаково і надає однакову функціональність незалежно від платформи, на якій вона встановлена.

Отже, СУБД Oracle приховує деталі реалізації механізмів управління даними на кожній з платформ, що дає підставу говорити про практично повну уніфікацію базового програмного забезпечення. Додатково до цього, архітектура Oracle дозволяє переносити прикладні системи, реалізовані на одній платформі, на інші платформи без змін як в структурах баз даних, так і кодів додатків. При цьому основним критерієм, що визначає можливість перенесення тих чи інших програмних компонентів між платформами, є повне виключення з них машинно-залежного коду.

Підсумовуючи все сказане вище, можна стверджувати, що СУБД Oracle володіє унікальними якостями переносимості, а також надає відкриту платформу для розробки гнучких додатків клієнт/сервер і Internet/Intranet-додатків. Наявність декількох редакцій серверу баз даних - корпоративної, стандартної, персональної та повна переносимість додатків між ними дозволяє задовольнити потреби корпоративної інформаційної системи і кардинально вирішити завдання уніфікації базового програмного забезпечення.

Пакет Oracle, наділений найрозвиненішим набором функцій для роботи з мовою Java і доступу до даних через Інтернет, системою оптимізації одночасного доступу.

Серед основних властивостей СУБД Oracle слід відзначити такі, як [6]:

- найвища надійність;
- можливість розбиття великих баз даних на розділи (large-database partition), що дає можливість ефективно управляти гігабайтними базами;
- наявність універсальних засобів захисту інформації;
- ефективні методи максимального підвищення швидкості обробки запитів;
- індексація по бітовому відображенню;

- вільні таблиці (в інших СУБД всі таблиці заповнюються відразу при створенні);
- розпаралелювання операцій в запиті;
- наявність широкого спектра засобів розробки та адміністрування;
- підтримка відомих платформ: Windows, AIX, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, Sun Solaris;
- орієнтація на Інтернет технології;
- підтримка XML в збережених процедурах, що дозволяє розробникам (традиційних) баз даних безпосередньо використовувати переваги мови XML, застосовуючи звичний механізм збережених процедур;
- доступ по протоколу HTTP, що підтримує відправку SQL-запитів до БД із застосуванням URL-адрес;
- інтегрований засіб виявлення закономірностей, що застосовується для відбору важливої, але не обов'язково очевидної бізнес-інформації з великих наборів даних. Це забезпечує збір, зберігання, управління і аналіз потоку даних про дії користувачів при відвідуванні ними Web-вузла;
- пов'язані бази даних OLAP - це, в першу чергу, кубики OLAP, вживані для реалізації нових можливостей аналізу даних;
- засіб передачі журналів, що дозволяє синхронізувати окремі системи за допомогою автоматизованого поновлення журналу транзакцій, що підвищує надійність системи та стійкість до аварій;
- паралельні операції при створенні індексів і виконанні пошуку на комп'ютерах із симетричною багатопроцесорної обробкою, що прискорюють роботу додатків;
- поліпшені засоби відмовостійкості, що спрощують створення і управління відмовостійкими кластерами, перемикання між вузлами кластера при збої і відновлення вузла кластера, не надає впливу на інші вузли;

- підтримка великого обсягу пам'яті і симетричної багатопроцесорної обробки, що дозволяє керувати програмами з високими навантаженнями, пов'язаними з великим обсягом транзакцій;
- інструментальні засоби, що включають графічні засоби розробки схем, генерації запитів і коду.

Орієнтація на інтернет технології – основний девіз сучасних продуктів Oracle.

Також необхідно відзначити, що розроблена фірмою Oracle система оптимізації одночасного доступу (multiversioning concurrency) є однією з найважливіших характеристик архітектури Oracle. Ця функція дозволяє виключити ситуацію, коли одному користувачеві доводиться чекати поки інший завершить зміни вмісту баз даних. Ця функція дозволяє СУБД Oracle виконувати за секунду більше транзакцій у розрахунку на одного користувача, ніж будь-яка інша база даних. За рівнем продуктивності при роботі в WEB середовище Oracle займає провідні місця і при цьому значно перевершує всі інші СУБД по надійності і безпеки [5].

4.3 Object-relational mapping solutions

4.3.1 Збереженість даних

Збереженість (зберігання, persistence) – атрибут даних, який гарантує, що дані будуть доступні навіть за межами життя додатку. Для об'єктно-орієнтованої мови Java збереженість гарантує, що стан об'єкта доступний навіть після зупинки додатка, який створив цей об'єкт.

Є різні способи для досягнення зберігання. Традиційний підхід до цієї проблеми полягає у використанні файлових систем, які зберігають необхідну інформацію в файлах. Таким способом важко керувати великими обсягами даних, тому що дані розподілені по різним файлам. Підтримка цілісності

даних також є проблемною в файлових системах, тому що та ж інформація може бути продубльована в різних файлах. Пошук даних у файловій системі займає багато часу, особливо якщо ці файли відсортовані. Крім того, файлові системи забезпечують обмежену підтримку одночасного доступу, так як вони не забезпечують цілісність даних. З усіх цих причин, файлові системи не розглядаються як гарне рішення для зберігання даних.

Найбільш поширеним підходом сьогодні є використання баз даних, які служать в якості сховищ для великих обсягів даних. Ці бази даних, а також системи управління базами даних, не тільки забезпечують зберігання об'єкта, а й управління інформацією, яка зберігалася.

Java додатки традиційно використовували JDBC (Java Database Connectivity) API для збереження даних у реляційних базах даних. JDBC API використовує мову запитів SQL для виконання CRUD операцій: create, read, update, delete. JDBC-код вбудований в класи Java. Цей код також в значній мірі залежить від SQL, який не стандартизований в базах даних, що ускладнює міграцію з однієї бази даних в іншу [7].

Реляційна технологія баз даних виділяє дані і їх відносини, в той час як об'єктно-орієнтована парадигма, яка використовується в Java, концентрується не на самих даних, а на операціях, виконуваних над цими даними. Тому, коли ці дві технології повинні працювати разом, виникає конфлікт інтересів. Концепціями об'єктно-орієнтованого програмування є поняття спадкування, поліморфізму і інкапсуляції, які не призначені для реляційних баз даних. Ще одна проблема виникає в результаті цієї невідповідності, коли для користувача типи даних, визначені у додатку Java відображаються в реляційних базах даних, а останні не забезпечують необхідну підтримку типів.

4.3.2 ORM – Object-relational mapping

ORM є інструментом, який досить просто зберігає об'єкти програми в таблиці в реляційній базі даних. ORM поводить себе як віртуальна база даних, приховуючи основну архітектуру бази даних від користувача. ORM надає функціональні можливості виконувати CRUD операції, і підтримує об'єктно-орієнтовані запити. ORM також підтримує відображення метаданих і допомагає в управлінні транзакціями програми [7].

Приклад допоможе проілюструвати роботу ORM. Розглянемо простий об'єкт Car, який повинен бути збережений в базу даних. Об'єкт Car в ООП є поданням таблиці CAR в моделі даних. Атрибути об'єкта Car отримані з стовпців таблиці CAR. Існує пряме відображення між класом Car і таблицею CAR, як показано на рисунку 4.1.

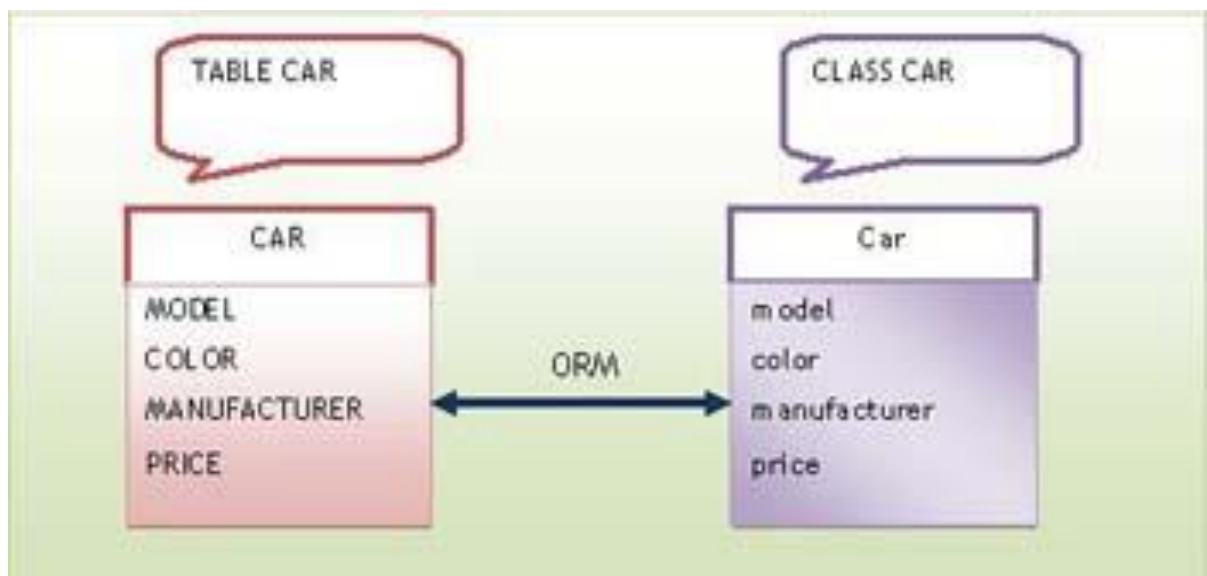


Рисунок 4.1 – Відображення об'єкта в таблиці

Є багато інструментів ORM з відкритим вихідним кодом, найпопулярнішими є Hibernate і MyBatis (iBatis) [7]. Ці інструменти забезпечують рівень абстракції між Java додатком і базою даних і трансформують об'єкти докладання в дані, які повинні бути збережені в базі даних. Об'єкти можуть бути визначені з використанням або XML файлів, або

анотацій. Метою використання фреймворків є відділення коду бази даних від коду програми, тим самим збільшуючи гнучкість додатків і спрощуючи процес розробки, надаючи оболонку для роботи з базою даних.

4.4 Вибір ORM інструментів

4.4.1 Java Persistence API

Java Persistence API є стандартом ORM для Java EE 5 платформи. JPA забезпечує платформу, на якій можуть бути використані конкретні реалізації Persistence фреймворків. Одним з головних особливостей JPA є те, що до нього може бути підключений будь-який Persistence фреймворк [7].

JPA - це заснований на POJO (Plain Old Java Object) стандарт Persistence моделі для ORM. JPA запозичує кращі ідеї від інших Persistence технологій.

JPA використовує анотації та/або XML файлу для налаштування відображення об'єктів Java в додатки і таблиці в реляційній базі даних. JPA є комплексним рішенням ORM і підтримує спадкування і поліморфізм. Він також визначає мову запитів SQL як JPQL (Java Persistence Query Language).

З JPA, можливо підключити будь-яку ORM-технологію, що реалізує специфікацію JPA просто підключаючи всі необхідні JAR-файли до додатка.

JPA використовує безліч інтерфейсів і типів анотацій, визначених у пакеті `javax.persistence`. JPA використовує класи сутностей, які зіставляються з таблицями в базі даних. Ці класи сутностей визначаються за допомогою JPA анотації.

Особливості класу сутності наступні:

- клас об'єкта анотований за допомогою `javax.persistence.Entity` анотації;
- він повинен мати `public` або `protected` конструктор без аргументів, а також може містити інші конструктори;
- клас не може бути оголошений як `final`;

- класи сутностей можуть успадковуватися від інших сутностей і класів, і навпаки;
- класи не можуть мати загальнодоступні змінні екземпляра. Члени класу повинні використовуватися тільки через методи `get` і `set`;
- немає необхідності реалізовувати будь-які спеціальні інтерфейси. Однак, якщо сутності будуть передані в якості аргументів по мережі, то вони повинні реалізовувати `Serializable` інтерфейс.

JPA слід використовувати, коли необхідно стандартне `persistence` рішення. JPA підтримує обидві можливості ООП: поліморфізм і успадкування. Недоліком JPA є необхідність технології, яка буде його реалізовувати. Ця стороння технологія також може забезпечити деякими іншими функціями, які не є частиною JPA. Однією з таких функцій є підтримка кешування, яке добре підтримується одним з найпопулярніших фреймворків `Hibernate`. Також недоліком є те, що JPA може працювати тільки з реляційними базами даних.

4.4.2 Hibernate

`Hibernate` був створений командою розробників на чолі з Гевіном Кінгом. Розвиток `Hibernate` почався в 2001 році і спочатку технологія розроблялася для `Java`, а в 2005 році була створена версія для `.Net` під назвою `nHibernate` [7].

`Hibernate` – це ORM рішення з відкритим вихідним кодом, головною особливістю якого є підтримка об'єктно-орієнтованого моделювання. Технологія використовує файли `XML` для зіставлення об'єктів бази даних з об'єктами додатків, також є підтримка анотацій, які відповідають специфікації `EJB` (`Enterprise Java Beans`).

Технологія включає дуже потужну мову запитів – `HQL` (`Hibernate Query Language`). `HQL` подібна мові `SQL` і також має деякі додаткові

можливості. HQL – повністю об'єктно-орієнтована мова, що дозволяє використовувати всю міць ООП. Запити HQL не чутливі до реєстру окрім імен і властивостей Java класів. Результатом запиту є об'єкт, який доступний і відразу ж може бути використаний. У складних запитах на кілька таблиць з використанням HQL немає необхідності об'єднувати ці таблиці, операція JOIN буде виконана автоматично.

Hibernate є ідеальним рішенням для ситуацій, коли є повний контроль як над додатком, так і над проектуванням бази даних. Існує можливість змінювати додаток таким чином, щоб він відповідав базі даних або навпаки. З Hibernate не потрібно турбуватися про ручні зміни бази даних. Якщо знадобиться змінити базу даних, це можна зробити при допомозі програми, в якій необхідно буде лише змінити файл конфігурацій Hibernate. Розробники можуть легко користуватися Hibernate і без особливих знань SQL [8].

4.4.3 MyBatis

Проект MyBatis (Ibatis) був заснований Клінтоном Бегіном і випущений в 2001 році. Фреймворк був спочатку розроблений для Java, проте на даний момент він розширений і для підтримки інших платформ, таких як .NET і Ruby [7].

ORM використовує генерацію JDBC та/або SQL коду. У деяких випадках виникає необхідність більш сильного контролю над кодом, що генерується, зокрема над SQL кодом. При створенні програми, яка містить ряд запитів для роботи з БД, буде більш ефективно написати власні запити SQL, а не покладатися на ті, що ORM-генеруються. Крім того, стандартні ORM інструменти не можуть бути використані при невідповідності об'єктної моделі та моделі даних.

MyBatis є фреймворком, який допомагає вирішити ці проблеми, так як дає можливість використовувати всі переваги мови SQL і уникнути

складності JDBC. На відміну від більшості інших Persistence-фреймворків, MyBatis забезпечує безпосереднє використання SQL і гарантує, що всі переваги SQL доступні.

Простота - велика перевага MyBatis, так як вона забезпечує побудову простого шару додатків для доступу до даних. У цьому фреймворку модель даних і об'єктна модель не повинні бути точно співставлені одна з одною, так як MyBatis використовує «Data Mapper», який відображає об'єкти збережених процедур, SQL запитів або результуючих об'єктів (ResultSets) за допомогою дескриптора XML. Таким чином, MyBatis дозволяє моделі даних і об'єктної моделі бути незалежними одна від одної.

Наведемо приклад. Для об'єкта даних CAR (рис 4.1) не обов'язково створювати клас Car з атрибутами, що відповідні всім полям таблиці (якщо в цьому немає необхідності). Наприклад, атрибут MANUFACTURER не цікавий і в класі Car він не створений, для цього достатньо не вказувати його у відповідний resultMap (рис. 4.2) в XML файлі, де поле «property» означає поле в класі, а «column» – атрибут таблиці:

```
<resultMap id="result" class="Car">  
  <result property="model" column="MODEL"/>  
  <result property="color" column="COLOR"/>  
  <result property="price" column="PRICE"/>  
</resultMap>
```

Рисунок 4.2 – resultMap таблиці CAR

MyBatis також не підходить для нереляційних баз даних, оскільки такі бази даних не підтримують транзакції і інші ключові функції, використовувани в даному фреймворку [7].

4.4.4 Порівняння ORM інструментів

Виділимо кілька критеріїв (параметрів) для оцінки:

- простота. За даним критерієм MyBatis є кращим рішенням. З розглянутих технологій MyBatis найпростіший у налаштуванні і використанні. Також, на відміну від Hibernate і JPA, він вимагає тільки знань SQL;
- комплексність ORM рішення. Традиційні рішення ORM, такі як Hibernate і JPA, можуть використовуватися тільки при досягненні повного об'єктно-реляційного відображення. Hibernate і JPA зіставляють Java об'єкти безпосередньо з таблицями бази даних, в той час як MyBatis заповнює Java об'єкти за результатами запитів SQL. У деяких випадках, об'єкти в моделі предметної області розроблені відповідно до бізнес-логіки і, можливо, не повністю відображаються на моделі даних. У такому випадку MyBatis є ідеальним рішенням;
- залежність від SQL. Якщо необхідно використовувати фреймворк без взаємодії з SQL, то Hibernate підходить найкраще, однак якщо необхідно мати повний контроль при взаємодії з БД – MyBatis є кращим рішенням;
- підтримка інших мов запитів. MyBatis підтримує тільки SQL, в той час як Hibernate і JPA використовують свої власні мови запитів (HQL і JPQL, відповідно), які схожі на SQL;
- продуктивність. Hibernate покращує продуктивність, надаючи кешування об'єктів, які допомагають з швидким витягом даних з бази даних. MyBatis використовує запити SQL, які можуть бути доопрацьовані для

підвищення продуктивності. Продуктивність JPA залежить від того, яка реалізація обрана;

- мобільність для різних БД. При використанні Hibernate це питання легко вирішується зміною діалекту бази даних у файлі конфігурації. Hibernate використовує цю властивість в якості керівництва для генерації коду SQL, специфічного для кожної бази даних. Мобільність додатків MyBatis залежить від написаного SQL. Якщо запити пишуться з використанням портативного SQL, то MyBatis буде також працювати в інших реляційних базах даних. Мобільність JPA залежить від використовуваної реалізації;

- мобільність для non-Java платформ. MyBatis підтримує .NET і Ruby. Hibernate підтримує .Net у вигляді NHibernate. JPA, є спеціалізованою Java API та не підтримує інші платформи крім Java.

MyBatis, Hibernate і JPA - три різних механізми зберігання даних в реляційній базі даних. Кожен з них має свої переваги та недоліки. MyBatis не дає прямих відповідностей бізнес об'єктів і реляційних моделей зберігання даних, що є перевагою, оскільки бізнес логіка додатків найчастіше відхиляється від моделі зберігання даних. Також, MyBatis надає повний контроль над запитамі. Таким чином, для виконання даної роботи вибором ORM рішення є MyBatis.

4.5 Інструкція по роботі з системою

При запуску програми відображається сторінка авторизації, яка представлена на рисунку 4.3.

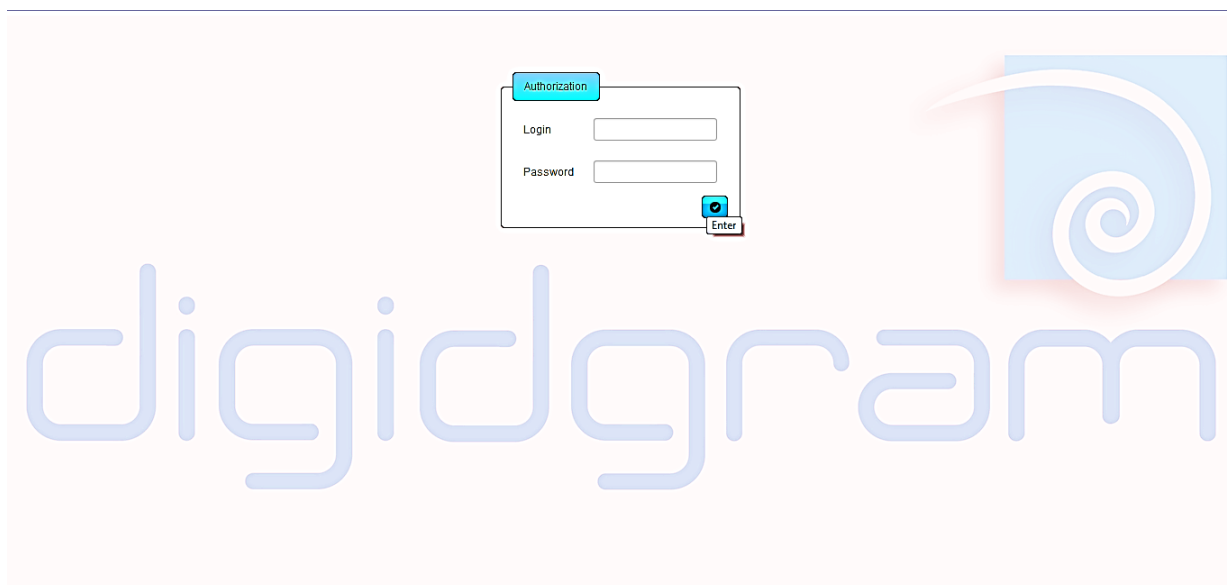


Рисунок 4.3 – Сторінка авторизації

При невірному введенні логіна або пароля виводиться повідомлення, представлене на рисунку 4.4.

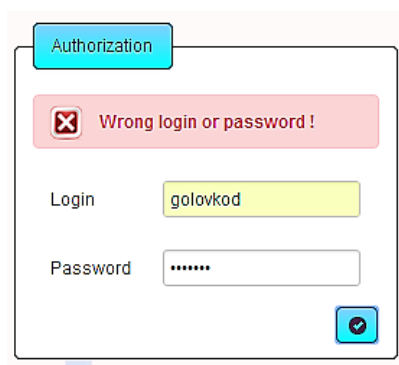


Рисунок 4.4 – Помилка авторизації

При успішній авторизації відкривається стартова сторінка з привітанням. На верхній панелі знаходиться основне робоче меню (Додаток Б). Сторінка представлена на рисунку 4.5.

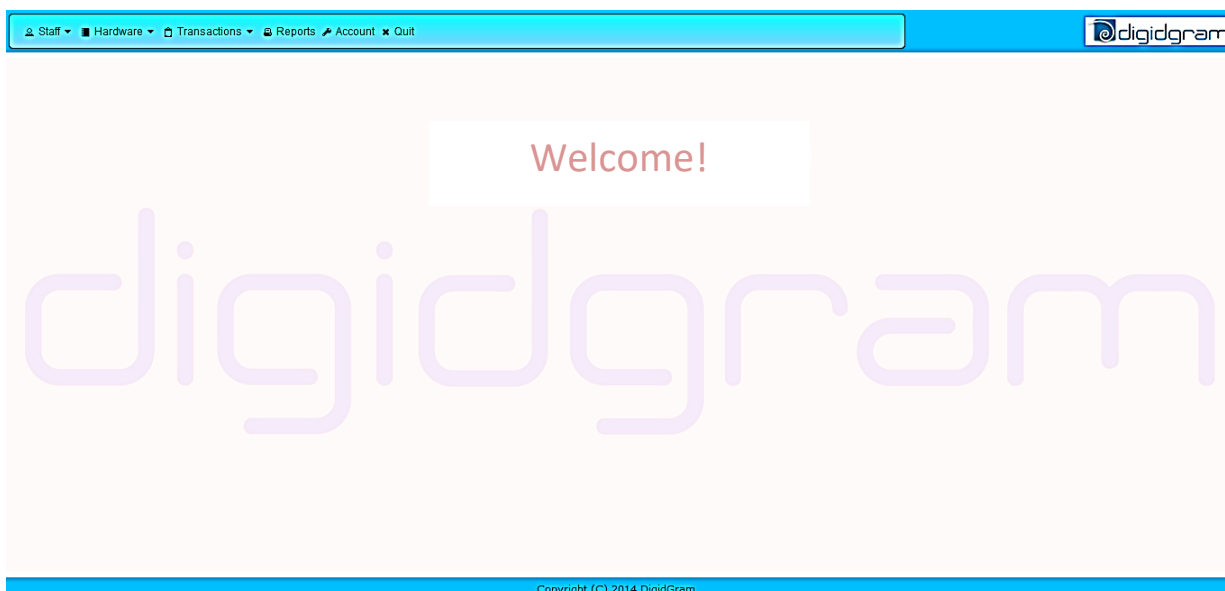


Рисунок 4.5 – Привітання

4.5.1 Облік працівників підприємства

Перегляд інформації про персонал. Вибираємо пункт «Staff» -> «Show» основного меню, потрапляємо на сторінку з таблицею, яка містить список всіх працівників підприємства. Дана сторінка представлена на рисунку 4.6.

	Name	Role	Login	Hire date
1	Dmitriy Golovko	Admin	golovkod	2014-05-27 17:13:11.0
2	Alex Shot	Manager	shota	2014-05-27 17:13:11.0
3	Marina Shevchenko	Booker	shevchenkom	2014-05-27 17:13:11.0
4	Yriy Merkel	Storekeeper	merkely	2014-05-27 17:13:11.0
5	Victoria Riht	Operator	rihtv	2014-05-27 17:13:11.0
6	Ivan Pupkin	Operator	pupkini	2014-05-27 17:13:11.0
7	Petya Petrov	Heaver	petropv	2014-05-27 17:13:11.0
8	Migel Butch	Heaver	butchm	2014-05-27 17:13:11.0
9	Vasily Vlasov	Heaver	vlasov	2014-05-27 17:13:11.0
10	Fedor Demidov	Heaver	demidovf	2014-05-27 17:13:11.0

Рисунок 4.6 – Перегляд інформації про персонал

Редагування персоналу. Вибираємо пункт «Staff» -> «Add/Edit» основного меню, потрапляємо на сторінку редагування. Дана сторінка представлена на рисунку 4.7.

ID	Name	Role	Login	Hire date	Edit
5	Victoria Riht	Operator	rihtv	2014-05-27 17:13:11.0	
6	Ivan Pupkin	Operator	pupkini	2014-05-27 17:13:11.0	
22	Karina Levko	Operator	levkok	2014-05-27 21:17:56.0	
27	Anna Vetkova	Operator	vetkova	2014-05-27 21:26:16.0	

Рисунок 4.7 – Редагування персоналу

Вибравши фільтр за посадою «Operator», змінимо співробітника з номером 27 (Anna Vetkova). Отримуємо повідомлення про успішне редагування (рис. 4.8).



Рисунок 4.8 – Успішна зміна даних працівника

Є можливість додати нового працівника (Додаток В). Натиснувши на кнопку «Add employee» відкривається діалог з полями для заповнення. Представлено на рисунку 4.9.

Add new employee

Role:

Name:

Login:

Password:

Рисунок 4.9 – Створення працівника

Натиснувши кнопку «Save», діалог закривається, виводиться повідомлення про успішне додавання. Таблиця працівників оновлюється. Представлено на рисунку 4.10.

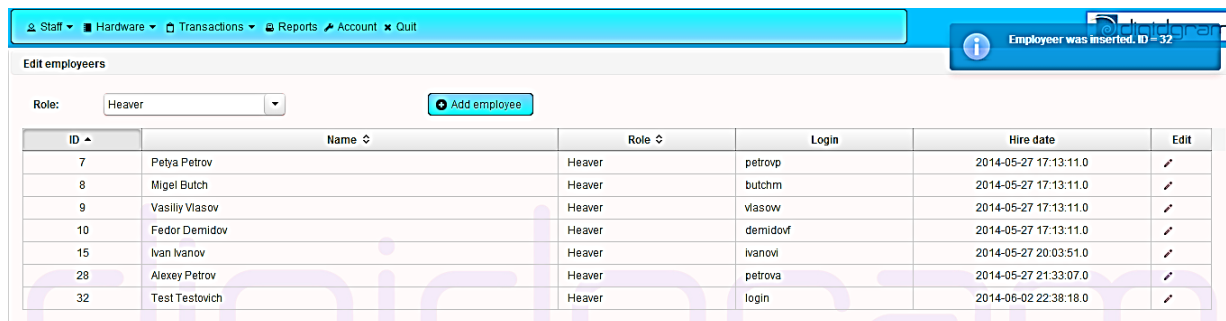


Рисунок 4.10 – Успішне створення працівника

4.5.2 Облік товарів

Перегляд інформації про товар. Вибираємо пункт «Hardware» -> «Show» основного меню, потрапляємо на сторінку з таблицею, яка містить список всіх товарів підприємства. Дана сторінка представлена на рисунку 4.11.

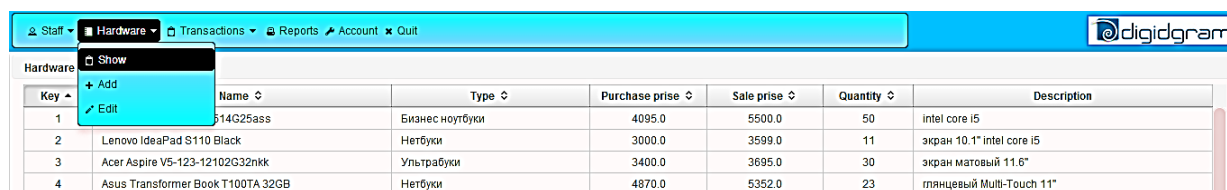


Рисунок 4.11 – Перегляд товарів

Додавання товару. Вибираємо пункт «Hardware» -> «Add», потрапляємо на сторінку додавання товару, заповнимо поля. Натиснувши кнопку «Save», товар зберігається, виводиться повідомлення про успішне додавання. Представлено на рисунку 4.12.

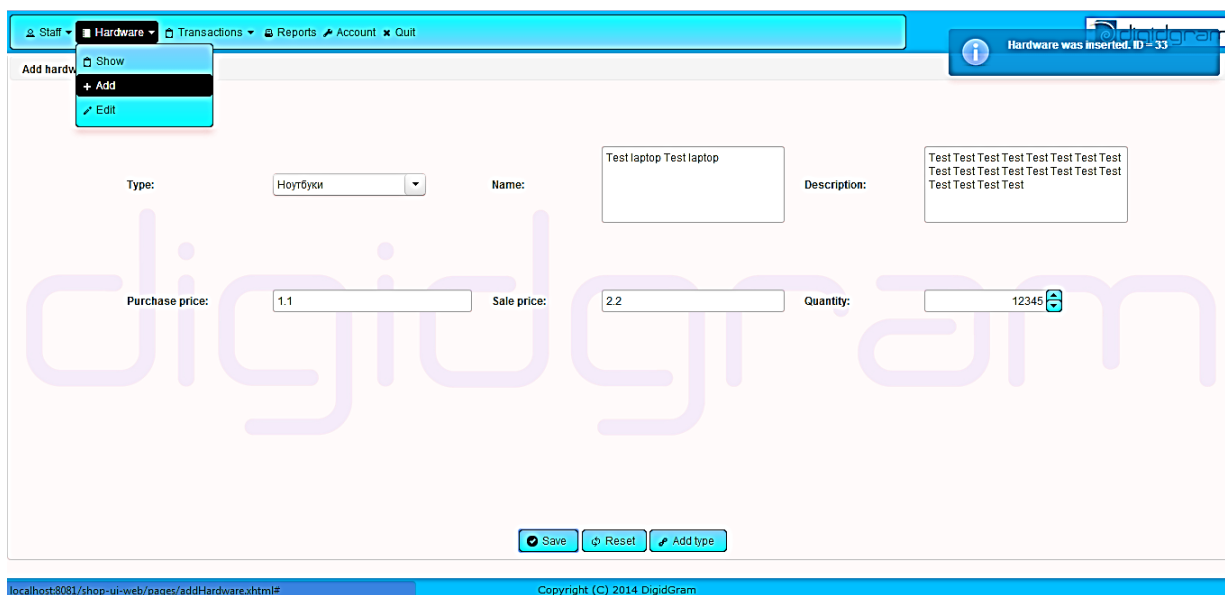


Рисунок 4.12 – Додавання товару

Також на даній сторінці можна додати новий тип товару. Натискаємо кнопку «Add type», відкривається діалогове вікно додавання типу. Заповнимо усі поля, натиснемо кнопку «Save». Отримуємо повідомлення про успішне додавання типу. Представлено на рисунку 4.13.



Рисунок 4.13 – Додавання типу товару

Редагування товарів. Вибираємо пункт «Hardware» -> «Edit» основного меню, потрапляємо на сторінку редагування. Виберемо фільтр по типу і

Key	Name	Type	Sale price	Quantity	Description
18	Mackie HD1531 (D000529)	Аудио	23798.0	100	Мощность 900 Вт
20	AV-ресивер Yamaha RX-V775 Black	AV-ресиверы	13798.0	5	Количество акустических каналов: 7.2, Выходная мощность на канал: 160 Вт
28	Magnat MC 1 Black	Аудио	18000.0	10	Количество каналов: 2.0 Общая выходная мощность (RMS): 80 Вт Поддержка носителей: CD Караоке, Нет Интерфейсы: AUX (3.5 мм), Выход на наушники (3.5 мм), Оптический вход, Цифровой коаксиальный выход

Рисунок 4.16 – Перегляд інформації про операцію

Створення операції. Вибираємо пункт «Transaction» -> «Sales» (або «Purchases») -> «Create» основного меню. Потрапляємо на сторінку створення угоди. Заповнимо усі поля і додамо пару товарів. Натиснувши кнопку «Save transaction», отримуємо повідомлення про успішне створення операції. Представлено на рисунку 4.17.

Key	Name	Quantity	Cost
33	Test laptop Test laptop EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST	5	11.5
6	Samsung PS-64F8500	22	659978.0
Totals:		27	659989.5

Рисунок 4.17 – Створення операції

Перевіримо, що від загальної кількості кожного обладнання віднята продана кількість. У звітності підприємства прибуток за рік і за місяць збільшилися.

Облік угод продемонстрований на типі «Sale» – продажу. Також всі операції перевірені на типі «Purchase» – поставки.

4.5.4 Звітність роботи підприємства

Вибравши пункт «Reports» основного меню, потрапляємо на сторінку звітів про роботу підприємства, яка представлена на рисунку 4.18.

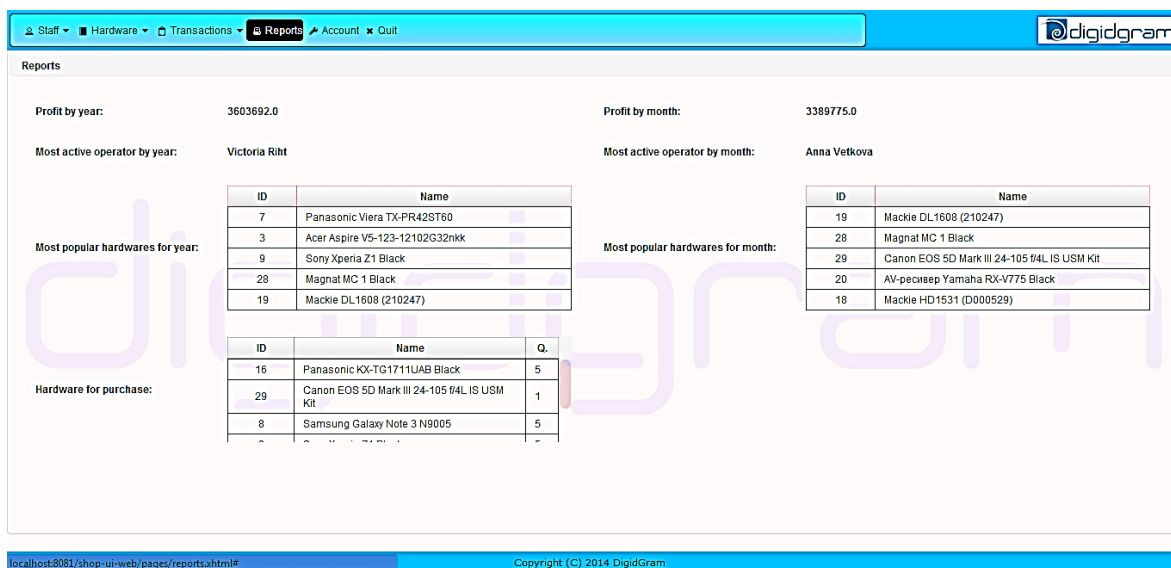


Рисунок 4.18 – Сторінка звітів

4.5.5 Зміна особистих даних

Вибравши пункт «Account» основного меню, потрапляємо на сторінку редагування особистих даних, яка представлена на рисунку 4.19.

The screenshot shows the 'Account' page with the following form fields:

- Name:**
- Login:**
- Password:**

At the bottom of the form, there are two buttons: **Save** and **Reset**.

Рисунок 4.19 – Редагування особистих даних

Змінивши дані, натискаємо кнопку «Save», виводиться повідомлення про успішну зміну особистих даних – рисунок 4.20.



Рисунок 4.20 – Успішне редагування особистих даних

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Вступ

Метою розділу «Охорона праці та безпека в надзвичайних ситуаціях» є розгляд методів і засобів забезпечення безпечних умов праці на робочих місцях користувачів комп'ютерних та інформаційних технологій при розробці предмета праці (програмного забезпечення), а також його експлуатації.

У даному розрахунковому завданні необхідно провести аналіз системи "Людина – Машина – Середовище" та визначити потенційно небезпечні та шкідливі виробничі чинники, їх вплив на виробниче середовище та на людину, розглянути характеристики приміщення і шкідливі виробничі фактори, які мають місце в даному приміщенні.

5.2 Аналіз умов праці в офісі на робочому місці оператора ЕОМ

У офісі, розміри якого складають 5,3×3,75×2,85 м (довжина × ширина × висота) розташовано два робочих місця, кожне з яких оснащено персональним комп'ютером. Технічні характеристики комп'ютерів наступні: процесор – Intel Core i5 з тактовою частотою процесора 1,7ГГц, об'єм жорсткого диску HDD 500Гб, об'єм оперативної пам'яті складає 4Гб, відео карта GeForce GT 720 з пам'яттю 1Гб, рідкокристалічний монітор з діагоналлю 24". Площа офісу складає 19,875 м², а об'єм – 56,64375 м³. Норма площі та об'єму на одного працівника, за вимогами ДСанПиН 3.3.2-

007-98[9], складає 6 м² та 20 м³. Таким чином, для даного приміщення, на кожного працюючого відводиться 9,9375 м² площі та 28,321875 м³, що відповідає вимогам.

Працівники, офіс та персональні комп'ютери утворюють систему «Людина-Машина-Середовище».

Елемент «Людина» розглядається з таких сторін:

– Л1 – це людина, яка виконує управління «машиною», переважно для виконання основної задачі системи — виробництва кінцевого продукту, а також забезпечення можливості цього виробництва. Цей елемент є об'єктом вивчення інженерною психологією та іншими ергономічними дисциплінами;

– Л2 – це людина, яка розглядається з точки зору безпосереднього впливу на навколишнє середовище (за рахунок тепло - та вологовиділення, споживання кисню та ін.);

– Л3 – це людина, яка розглядається з точки зору її фізіологічного стану під впливом чинників, що впливають на неї у виробничому процесі. Цей елемент вивчається з позиції охорони праці та суміжних з нею дисциплін. Стан цього елемента системи є вхідним пунктом процесу декомпозиції, що проводиться.

Елемент «Машина» можна поділити на три елементи:

- М1 – елемент, який виконує основну технологічну функцію;
- М2 – елемент, який відповідає за функції аварійного захисту (заземлення типу TN-C-S, ізоляція, відключення з-за перегріву);
- М3 – елемент, який впливає на виробниче середовище та людину як джерело небезпечних та шкідливих виробничих факторів, за рахунок виділення тепла, шуму та електромагнітного випромінювання.

Під елементом «Середовище» розуміється приміщення для розробки програмного забезпечення – офіс.

Елемент «Предмет праці» представляє інформаційна система.

Загальна схема «Людина-Машина-Середовище» для розробки програмного забезпечення представлена на рисунку 5.1.

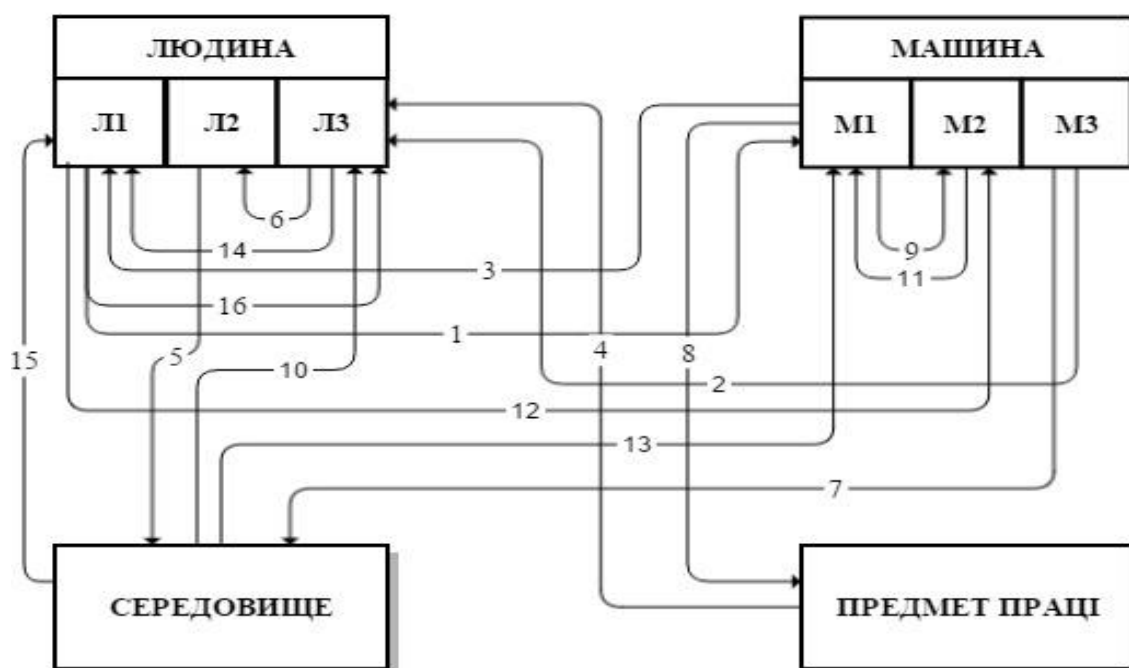


Рисунок 5.1 – Схема «Людина-Машина-Середовище»

Пояснення зв'язків системи «Людина-Машина-Середовище» представлені у таблиці 5.1.

Таблиця 5.1 – Зв'язки системи «Людина-Машина-Середовище»

№	Напрямок зв'язку	Опис зв'язку
1	Л1 – М1	Налаштування та керування машини людиною задля створення продукту праці (програмного забезпечення).
2	М3 – Л3	Вплив машини на людину, а саме випромінювання монітору, виділення теплоти системного блоку, підвищення рівню шуму.

Продовження таблиці 5.1

№	Напря́м зв'язку	Опис зв'язку
3	М1 – Л1	Комп'ютер надає людині необхідну інформацію для виробництва програмного забезпечення та забезпечує її надійну обробку.
4	Продукт праці – Л3	Вплив продукту праці на психологічний стан людини – розумове перенапруження, стреси.
5	Л2 – Середовище	Вплив людини на виробниче середовище завдяки споживанню кисню, виділення теплоти.
6	Л3 – Л2	Психологічний стан людини впливає на інтенсивність впливу людини на середовище. У стресовому стані людина буде споживати більше кислороду.
7	М3 – Середовище	Комп'ютер впливає на виробниче середовище завдяки електромагнітному випромінюванню, виділенню теплоти та створення шуму.
8	М1 – Продукт праці	Вплив комп'ютера на продукт праці – створення, зберігання, компіляція коду програмного забезпечення та його запуск.
9	М1 – М2	Передача інформації для аналізу стану машин для контролю аварійних ситуацій.

Продовження таблиці 5.1

№	Напрямок зв'язку	Опис зв'язку
10	Середовище – ЛЗ	Підвищений рівень шуму у приміщенні впливає на психологічний стан людини, поганий мікроклімат приміщення та погане освітлення впливають на фізіологічний стан людини.
11	М2 – М1	Аварійні управляючі дії (при відключенні ПК зупиняється виконання основної технологічної функції).
12	Л1 – М2	Вплив людини на механізми захисту та аварійного відключення персонального комп'ютера.
13	Середовище – М1	Стан середовища впливає на технічний стан машини (ПК нормально функціонує при оптимальних значеннях мікроклімату).
14	ЛЗ – Л1	Вплив стану людини на якість роботи (вдома знижує працездатність).
15	Середовище – Л1	Вплив виробничого середовища на якість роботи людини (підвищена або знижена температура повітря на робочому місці, підвищений шум, недостатнє освітлення).
16	Л1 – ЛЗ	Вплив виконуваних дій на фізіологічний стан людини.

Зв'язки у системі «Людина-Машина-Середовище» сприяють виникненню небезпечних та шкідливих виробничих факторів. Згідно ГОСТ 12.0.003-74[10] небезпечні виробничі фактори діляться на чотири категорії: фізичні, хімічні, біологічні, психофізіологічні.

Фізичні фактори:

- підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;
- підвищений рівень шуму;
- недостатня освітленість робочого місця;
- підвищена або знижена температура повітря;
- електромагнітне випромінювання;
- підвищена або знижена вологість повітря.

Хімічні та біологічні фактори у офісному приміщенні не виявлені.

Психофізіологічні фактори:

- перенавантаження органів зору;
- розумові перенавантаження;
- нервово-емоційні перенавантаження;
- монотонність праці;
- перенапруга опорно-рухової системи у зв'язку з постійним перебуванням у сидячому положенні.

Проаналізувавши умови праці, домінуючим небезпечним виробничим фактором була визначена небезпека ураження електричним струмом при неправильному виборі перетину проводів.

5.3 Промислова безпека у офісному приміщенні

Умови, які можуть створювати підвищену і особливу небезпеку ураження електричним струмом: підвищена вологість та температура, струмопровідний пил, струмопровідні підлоги, можливість одночасного дотику до заземлених металоконструкцій будівлі і металевих поверхней електроприладів. Хімічно-активні речовини відсутні.

У приміщенні комп'ютерного офісу застосовані такі технічні заходи:

– для захисту людей від ураження електричним струмом передбачена система заземлення типу TN, оскільки використовується електрична мережа напругою до 1000 В з глухо заземленою нейтраллю, згідно ДБН В.28-27-2006[18];

– всі наявні розетки забезпечені попереджувальними написами, мережеві шнури і вилки ізолювані діелектричними матеріалами, що знаходяться в справному стані;

– контроль та профілактика пошкоджень ізоляції. Контроль стану ізоляції, опір якої має становити не менше 500кОм, повинен проводитися не рідше одного разу на рік між фазою і нулем між фазами;

– згідно НПАОП 0.00-4.12-05[11] проводяться наступні інструктажі з охорони праці: вступний, первинний, повторний, позаплановий, цільовий. Факт інструктажу фіксується в журналі інструктажу, де реєструється такі дані: хто проводив, з ким проводили, дата проведення інструктажу.

– застосування пристроїв захисного аварійного відключення. Час спрацювання відключення 0,1 - 0,2 с.

Для запобігання займанню електропроводів слід правильно добирати переріз кабелів за значенням струму. Розрахуємо потрібний перетин кабелів для даного виробничого приміщення.

У відповідності з рекомендаціями НПАОП 40.1-1.21-98 [12] розрахунок перетину проводів ведуть за формулою 5.1:

$$S_e = \frac{I_{max}}{J_{ek}}, \quad (5.1)$$

де I_{max} – розрахунковий струм лінії при нормальній роботі мережі, одиниця вимірювання – А. $I_{max} = \frac{P}{U_{\phi}}$, де U_{ϕ} – фазна напруга мережі, що

дорівнює 220В, P – потужність ЕОМ. Оскільки маємо дві ЕОМ потужністю 450 Вт кожна, то $P = 900$ Вт;

$J_{ек}$ – економічна щільність струму, А/мм², яка визначається в залежності від матеріалу та часу використання максимального навантаження.

У якості провідників застосовуються мідні кабелі з гумовою та пластмасовою ізоляцією.

Значення економічної щільності струму беремо з таблиці 5.2.1

Таблиця 5.2.1 – Економічні щільності струму, які рекомендуються НПАОП 40.1-1.21-98

Найменування провідників	$J_{ек}$, А/мм ² , за тривалості використання максимуму навантаження в рік, год.		
	1000–3000	3000–5000	5000–8700
Голі проводи та шини: мідні.	2,5	2,1	1,8
Кабелі з паперовою та проводи з гумовою й поліхлорвініловою ізоляцією;			
мідні;	3,0	2,5	2,0
алюмінієві.	1,6	1,4	1,2
Кабелі з гумовою та пластмасовою ізоляцією:			
мідні;	3,5	3,1	2,7
алюмінієві.	1,9	1,7	1,6

Отже економічна щільність струму в нашому випадку = 3,5 А/мм², а

розрахунковий струм $I_{max} = \frac{450 \cdot 2}{220} = \frac{900}{220} \approx 4,1$ А. Підставивши значення

I_{max} та $J_{ек}$ у формулу 5.1 отримаємо:

$$S = \frac{4.1}{3.5} \approx 1.1714 \text{ мм}^2$$

Отриманий переріз провідника округлюють до найближчого більшого стандартного перерізу (табл. 5.2.2).

Таблиця 5.2.2 – Стандартні значення перерізу провідника

Переріз, мм ²	1	1,5	2,5	4	6	10	16	25	35	50	70	95	120	150	185
-----------------------------	---	-----	-----	---	---	----	----	----	----	----	----	----	-----	-----	-----

Отже переріз для нашого провідника повинен бути 1,5 мм².

5.4 Виробнича санітарія в офісі

Роботи в офісному приміщенні відносяться до робіт категорії 1а – легка фізична робота, яка виконується сидячи. Згідно із ДСН 3.3.6.042-99 [17] для такої категорії припустимі такі параметри вологості, температури та швидкості руху повітря:

- Оптимальна відносна вологість 40-60 %, припустима на робочих місцях 55 (при 28°C) – 75, в залежності від пори року.
- Оптимальна температура – 22-25° С, допустима температура: верхня межа – 25° -28° С, нижня межа – 21-22° С.
- Швидкість руху повітря: оптимальна – $\leq 0,1$ та припустима на робочих місцях $\leq 0,1 - 0,2$ м/с. Забезпечуються місцевим кондиціонером.

В приміщенні використовується сумісне освітлення: природне та штучне. Згідно з ДБН В.2.5-28-2006[19] розряд зорових робіт, що проводяться у приміщенні – III В. Нормативні значення штучного освітлення $E=300-500$ лк, природного – КПО $\geq 1,2$ %. Штучне освітлення виконано як загальне, за допомогою світильників з люмінесцентними лампами.

Шум в приміщенні відповідає нормативним значенням – 50 дБА згідно з ДСН 3.3.6-037-99[15] .

5.5 Пожежна безпека в офісі

Як відомо, пожежа може виникнути при взаємодії горючих речовин, кисню та джерела займання. Дане приміщення за класом пожежонебезпеки та вибухонебезпеки відповідно до НАПБ Б.03.002-2007[21] відноситься до категорії В, так як застосовуються тверді горючі речовини. Дана будівля має ступінь вогнестійкості – II, так як має несучу та огорожувальну конструкцію з бетону, згідно ДБН В.1.1.7-2002.[16]

Зазвичай основною причиною виникнення пожеж стає коротке замикання, перевантаження електромережі та порушення правил пожежної безпеки.

Для пожежогасіння потрібно використовувати вуглекислотні вогнегасники. На дві ЕОМ потрібно 2 вогнегасника ВВК-1,4 з розрахунку 1 вогнегасник на 3 ЕОМ по НАПБ Б.03.001-2004[21], але не менше двох. Додатковий вихід не потрібний тому, що розміри кімнати достатні для швидкої евакуації робітників з приміщення, згідно ДБН В.1.1.7-2002. [16]

Організаційно-технічні заходи з пожежної безпеки включають в себе наступне:

- інструктаж з пожежної безпеки;
- розробку заходів щодо дій працівників на випадок виникнення пожежі та організації евакуації;
- застосування плакатів наочної агітації з пожежної безпеки.

Автоматична система сигналізації виявляє пожежу за допомогою сигнальних датчиків та передає сигнал тривоги в пожежну охорону. У

приміщенні повинно бути 1 датчик, бо за ДБН В.2.5-56-10[13] на 75м² встановлюється 1 тепловий датчик.

Висновки до розділу 5

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

У ході виконання дипломної роботи була вирішена задача автоматизації обліку товарів підприємства. Сформульована постановка завдань обліку товару на складі, розроблена концептуальна модель даних, логічна і фізична структури даних, спроектована та розроблена інформаційна система.

У пояснювальній записці приведена схема даних та процес її проектування, проведений аналіз та вибір СУБД, а також засобів розробки програмного забезпечення, спроектована архітектура, завдяки якій є можливість розділення системи на незалежні модулі.

Розроблена інформаційна система дозволяє швидко і зручно проводити операції з продажу та закупівлі товарів, формувати звіти, а також керувати даними з БД. Таким чином, в роботі вирішені всі поставлені завдання, а саме:

- автоматизація обліку співробітників підприємства;
- автоматизація обліку товарів;
- автоматизація обліку операцій товарообігу;
- автоматизація звітності роботи підприємства.

Перевагами розробленої інформаційної системи є можливість обробки запитів від багатьох користувачів, незалежність від операційної системи та розширюваність завдяки архітектурі програмного забезпечення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Информационные технологии в маркетинге [Текст] : учеб. / под ред. Г. А. Титаренко. – М. : ЮНИТИ, 2000. – 335 с. – (Textbook). – ISBN 5-238-00154-1
2. Гринченко, Н. Н. Проектирование баз данных [Текст]: учеб. пособие / Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров. – М.: Горячая Линия – Телеком, 2004. – 240 с.
3. Clements, P. Documenting Software Architectures: Views and Beyond [Text] / P. Clements, F. Bachmann, L. Bass and others. – 2nd Edition. – Addison-Wesley Professional, 2010. – 334. – (Textbook). – ISBN 978-0-13-248861-7.
4. Гамма, Э. Приемы объектно-ориентированного проектирования Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – Питер. : Питер Бук, 2001. – 368 с.
5. Бежанишвили Л. Программирование на языке PL/SQL базы данных ORACLE (часть 1): учеб. пособие / Л. Бежанишвили. – Тбилиси: Издательский дом "Технический университет", 2009. – 58 с.
6. PL/SQL в Oracle. Первые шаги [Электронный ресурс] / Режим доступа: [www/URL: http://www.firststeps.ru/sql/oracle/](http://www.firststeps.ru/sql/oracle/) – 10.04.2014. – Загл. с экрана.
7. iBATIS, Hibernate, and JPA: Which is right for you? Object-relational mapping solutions compared [Электронный ресурс] / Режим доступа: [www/URL: http://www.javaworld.com/article/2077875/open-source-tools/ibatis--hibernate--and-jpa--which-is-right-for-you-.html](http://www.javaworld.com/article/2077875/open-source-tools/ibatis--hibernate--and-jpa--which-is-right-for-you-.html) – 17.04.2015 г. – Загл. с экрана.
8. Программирование на Java. Hibernate [Электронный ресурс] / Режим доступа: [www/URL: http://javaxblog.ru/category/article/hibernate-article/](http://javaxblog.ru/category/article/hibernate-article/) – 12.05.2015. – Загл. с экрана.

9. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин

10.ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация

11.НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці

12.НПАОП 40.1-1.21-98 Правила безпечної експлуатації електроустановок споживачів

13.ДБН В.2.5-56:2014 Системи протипожежного захисту

14.ГОСТ 12.1.001-89 ССБТ. Ультразвук. Общие требования безопасности

15.ГОСТ 12.1.003-83 ССБТ. Шум. Общие требования безопасности

16.ДБН В.1.1.7-2002 Пожежна безпека об'єктів будівництва

17.ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих

18.ГОСТ 12.1.030-81 ССБТ. Электробезопасность.Защитное заземление. Зануление

19.ДБН В.2.5-28:2015 Природне і штучне освітлення

20.ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения

21.НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою

ДОДАТОК А

Комп'ютерна презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

Інформаційна система автоматизації контролю та
обліку товарів підприємства

Керівник проекту: доц. Сафонова С.О.
Виконав: Рябоконт Д.А.
студент гр. КІ-146д

Рисунок А.1 – Слайд №1

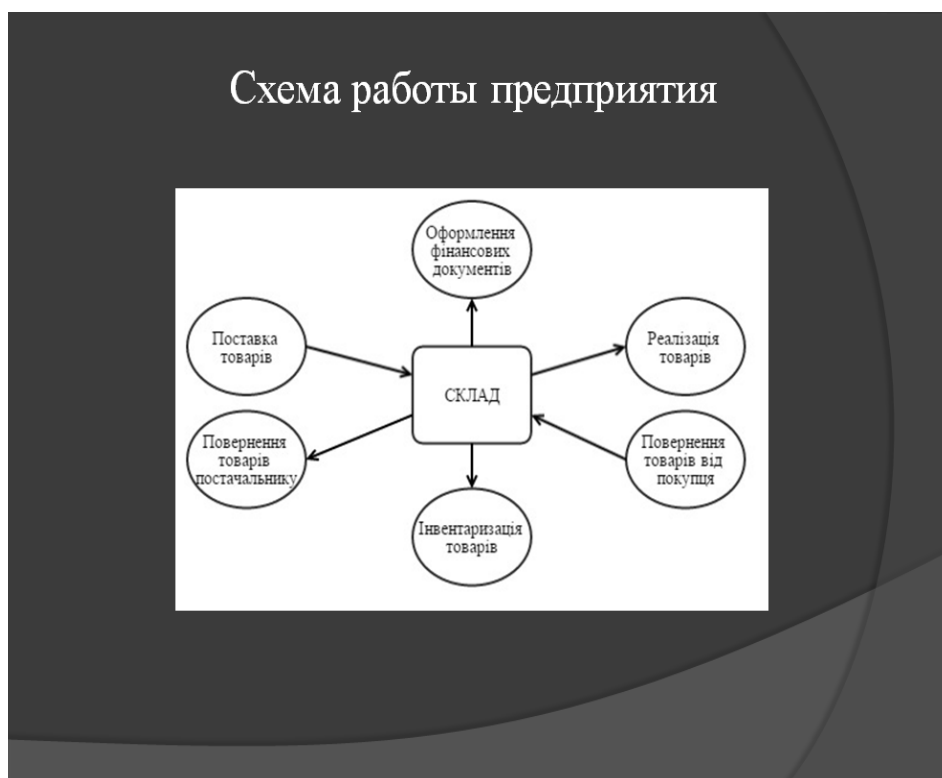


Рисунок А.2 – Слайд №2

Спецификация требований

Сущности:

- поставщик;
- покупатель;
- товар;
- персонал;
- транзакция.

Атрибуты сущностей:

- поставщик / покупатель : имя, адрес, телефон, электронная почта, расчетный счет;
- товар: бренд, наименование, цена продажи, цена закупки, количество, тип, описание;
- тип товара: название, подтипы, родительский тип;
- работник: имя, должность;
- транзакция: номер, тип операции, поставщик/покупатель, адрес доставки, сумма, список товаров.

Рисунок А.3 – Слайд №3

Спецификация требований

Приложение должно автоматизировать следующие операции:

- Учет сотрудников предприятия:
 - просмотр информации о персонале;
 - добавление новых сотрудников;
 - редактирование информации о сотрудниках.
- Учет товаров:
 - просмотр информации о товарах;
 - добавление новых товаров;
 - редактирование товаров;
 - добавление новых типов товара.
- Учет транзакций:
 - просмотр всех поставок и продаж;
 - оформление сделок по поставке и продаже.
- Отчетность работы предприятия:
 - подсчет прибыли за год/месяц;
 - определение самых популярных товаров;
 - определение самого активного сотрудника.
- Изменение личных данных:
 - изменение имени, логина и пароля.

Рисунок А.4 – Слайд №4

Концептуальное проектирование БД

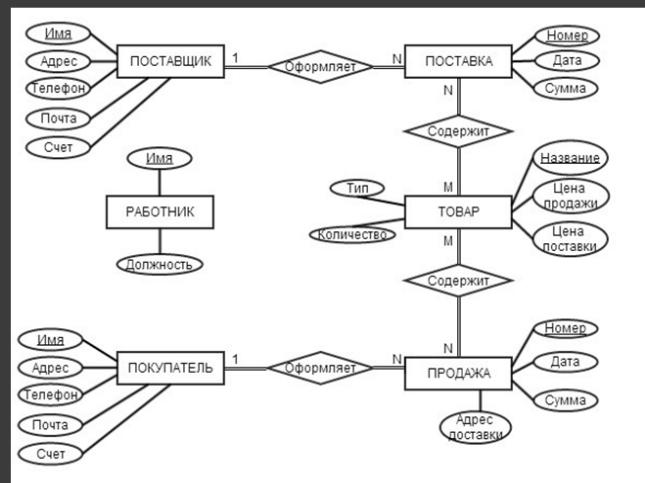


Рисунок А.5 – Слайд №5

Логическое проектирование

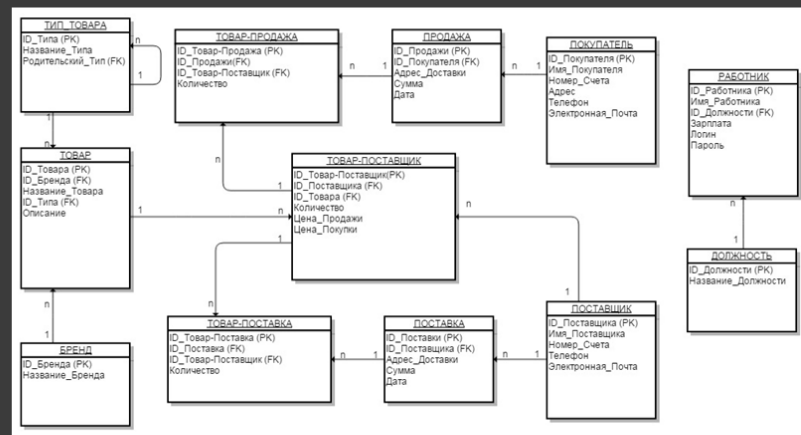


Рисунок А.6 – Слайд №6



Рисунок А.7 – Слайд №7

Использование MyBatis в качестве ORM

Для выполнения данной работы в качестве ORM – инструмента был выбран MyBatis.

MyBatis не обязывает обеспечивать полное соответствие бизнес объектов и реляционной модели данных.

Пример:

CAR
Model
Color
Manufacturer
Price

```

<resultMap id="result" class="Car">
  <result property="model" column="MODEL"/>
  <result property="color" column="COLOR"/>
  <result property="price" column="PRICE"/>
</resultMap>
  
```

MyBatis обеспечивает полный контроль над запросами к БД.

Рисунок А.8 – Слайд №8

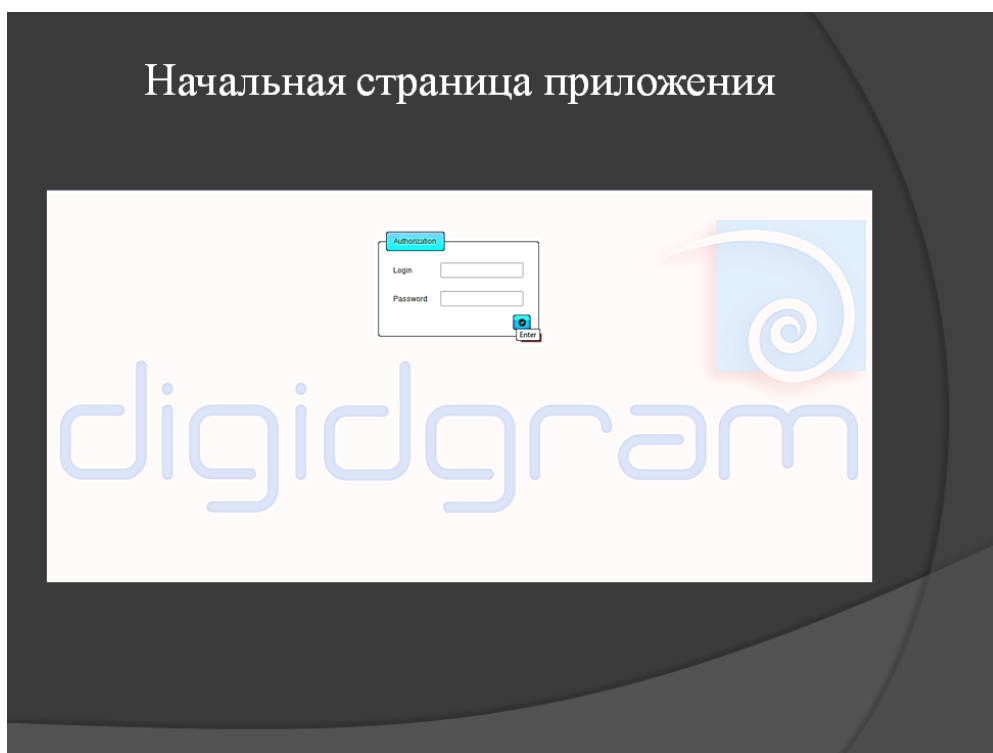


Рисунок А.9 – Слайд №9

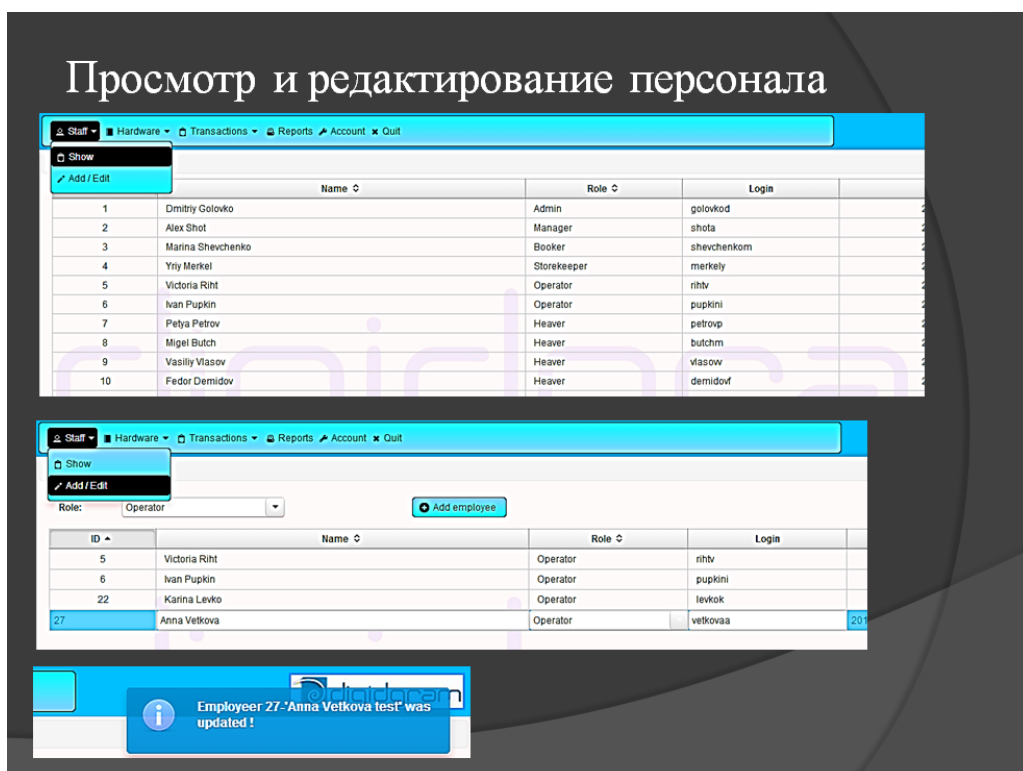


Рисунок А.10 – Слайд №10

Просмотр и редактирование товаров

The screenshot displays the 'Hardware' section of the digigram system. It features a navigation bar with 'Staff', 'Hardware', 'Transactions', 'Reports', 'Account', and 'Out'. The main content is divided into two sections: 'Hardware' and 'Edit goods'.

Hardware Table:

Key	Name	Type	Purchase price	Sale price	Quantity	Description
1	14Q25ass	Бизнес ноутбук	4095.0	5500.0	50	intel core i5
2	Lenovo IdeaPad S110 Black	Нетбук	3000.0	3599.0	11	экран 10.1" intel core i5
3	Acer Aspire V5-123-12102G32nkk	Ультрабук	3400.0	3695.0	30	экран матовый 11.6"
4	Asus Transformer Book T100TA 32GB	Нетбук	4870.0	5352.0	23	матричный Multi-Touch 11"

Edit goods Form:

Type:

Key	Name	Type	Purchase price	Sale price	Quantity	Description	Edit
1	Acer Aspire S7-191-73514Q25ass	Бизнес ноутбук	4095.0	5500.0	50	intel core i5	✓
2	Lenovo IdeaPad S110 Black	Нетбук	3000.0	3599.0	11	экран 10.1" intel core i5	✓
3	Acer Aspire V5-123-12102G32nkk	Ультрабук	3400.0	3695.0	30	экран матовый 11.6"	✓
4	Asus Transformer Book T100TA 32GB	Нетбук	4870.0	5352.0	23	матричный Multi-Touch 11"	✓
26	Asus Taichi 31 (TAICH31-CR022H)	Ультрабук	1330.0	1499.99	20	Экран 2 x 13.3" (1920x1080) LED IPS, матовый / Intel Core i5-3337U (1.8 ГГц) / RAM 4 Гб / SSD 1Тб / Intel HD Graphics 4000 / Flex IO / LAN / Wi-Fi / Bluetooth 4.0 / веб-камера 5 Мп + фронтальная 5 Мп / Windows 8 / 1.55 кг	✓
33	Test laptop Test laptop EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST	Ноутбук	1.2	2.3	12	Test Test	✓

A notification bubble in the top right corner states: 'Hardware 33 Test laptop Test laptop EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST was updated!'.

Рисунок А.11 – Слайд №11

Добавление товаров

The screenshot shows the 'Add goods' form in the digigram system. It includes a navigation bar and a notification bubble: 'Hardware was inserted. 33'. The form has the following fields:

- Type:
- Name:
- Description:
- Purchase price:
- Sale price:
- Quantity:

At the bottom, there are three buttons: 'Save', 'Reset', and 'Add type'.

Рисунок А.12 – Слайд №12

Поиск и просмотр сделок

From date: 01.06.2014 To date: 01.06.2014

Key	Collaborator	Total sum
18	ООП Петр Пулюк	2628790.0
19	Rebud Corporation	2773000.0
22	ООП "Sonus"	209400.0
23	ООП Петр Пулюк	3695.0
24	ЧП БУБу	128750.0
25	Rozetca	180250.0

Transaction ID: 18 Type: SALE Collaborator: ООП Петр Пулюк Contact phone: 0731580777
 Total sum: 2628790.0 Delivery address: test delivery address Mail: top.pp@mail.com Payment account: 1987654321

Key	Name	Type	Sale price	Quantity	Description
18	Mackie HD1531 (D000529)	Аудио	23798.0	100	Мощность 900 Вт
20	AV-ресивер Yamaha RX-V775 Black	AV-ресиверы	13798.0	5	Количество акустических каналов: 7.2. Выходная мощность на канал: 160 Вт
28	Magnat MC 1 Black	Аудио	18000.0	10	Количество каналов: 2.0 Общая выходная мощность (RMS): 80 Вт Поддержка носителей: CD Карaoke; Неги интерфейс: AUX (3.5 мм), Выход на наушники (3.5 мм), Оптический вход, Цифровой коаксиальный выход.

Рисунок А.13 – Слайд №13

Создание сделок

Transaction was created. 26

Collaborator: ООП Петр Пулюк Hardware type: ТВ-техника Quantity: 22
 Cost = 659978.0

Delivery address: test Hardware: Samsung PS-64F8500
 Max quantity = 42

Key	Name	Quantity	Cost
33	Test laptop Test laptop EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST EDIT TEST	5	115
6	Samsung PS-64F8500	22	659978.0
Totals:		27	659989.5

Delete selected hardware Save transaction

localhost:8081/shop-ua-web/pages/createTransaction.html Copyright (C) 2014 DigidGram

Рисунок А.14 – Слайд №14

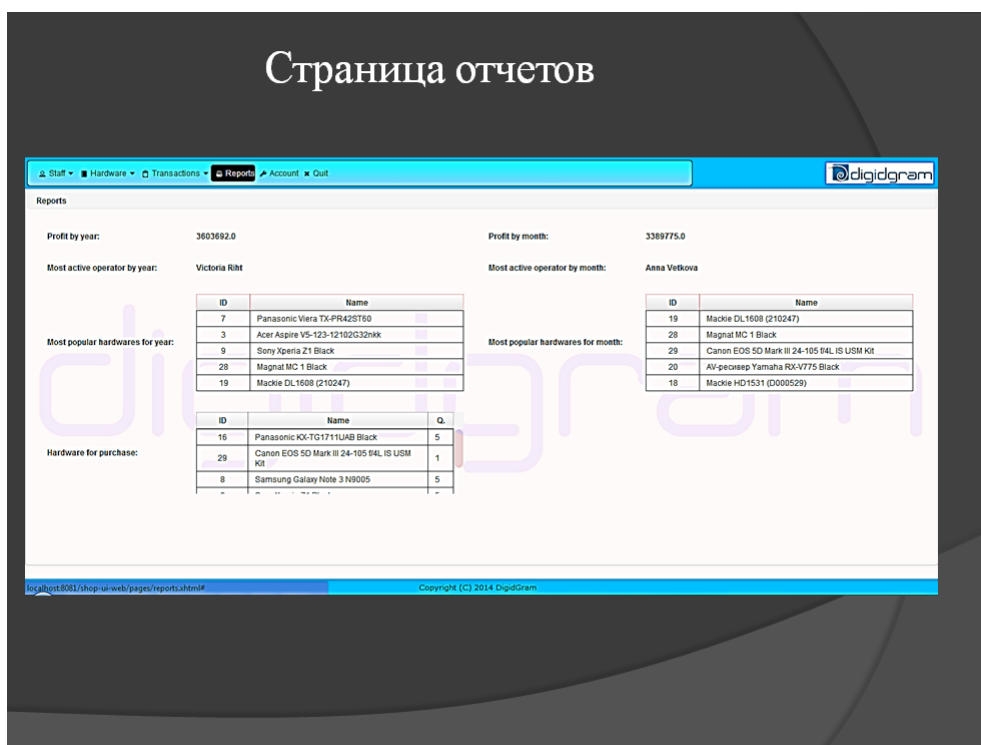


Рисунок А.15 – Слайд №15

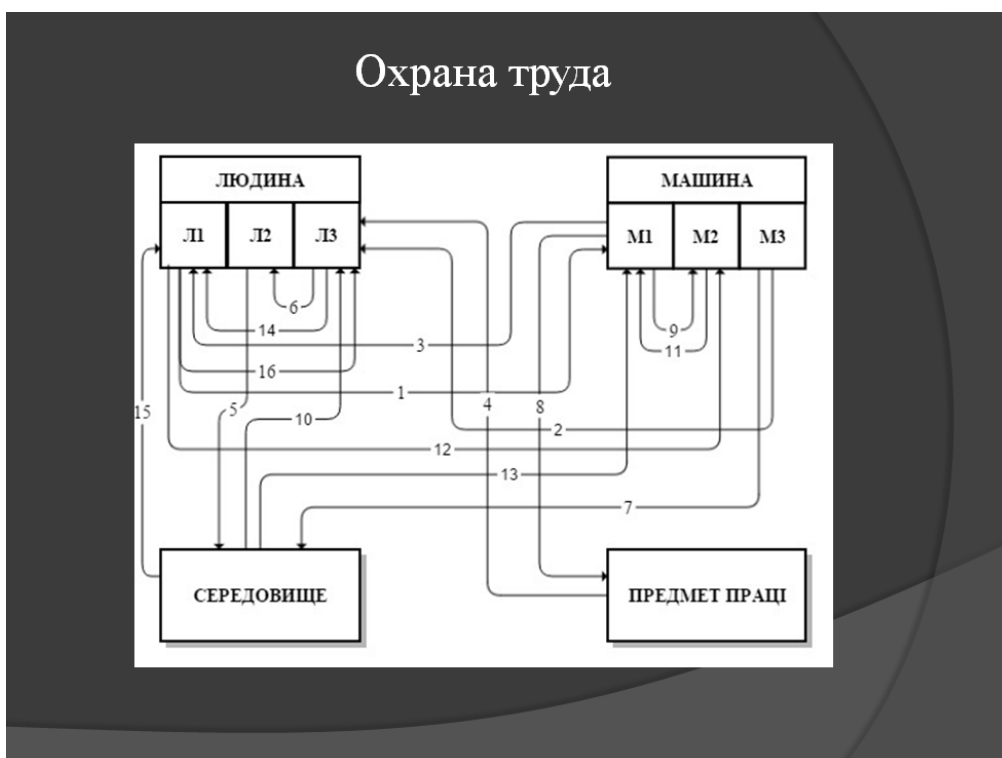


Рисунок А.16 – Слайд №16

Выводы

В ходе выполнения дипломной работы была решена задача автоматизации учета товаров предприятия. Сформулирована постановка задач учета товара на складе, разработана концептуальная модель данных, логическая и физическая структуры данных, спроектирована и разработана информационная система.

Разработанная информационная система позволяет быстро и удобно проводить операции по продаже и закупке товаров, формировать отчеты, а также управлять данными БД. Решены все поставленные задачи, а именно:

- автоматизация учета сотрудников предприятия;
- автоматизация учета товаров;
- автоматизация учета операций товарооборота;
- автоматизация отчетности работы предприятия.

Преимуществами разработанной информационной системы является возможность обработки запросов от многих пользователей, независимость от операционной системы и расширяемость благодаря архитектуре программного обеспечения.

Рисунок А.17 – Слайд №17

ДОДАТОК Б

Лістинг програми Main.java

```
1  package com.subd.shop.mainConsole;
2  import java.util.List;
3  import java.util.Locale;
4  import org.springframework.context.support.ClassPathXmlApplicationContext;
5  import com.subd.shop.persistence.mybatis.dao.EmployerDao;
6  import com.subd.shop.persistence.mybatis.dto.EmployerDto;
7  public class Main {
8  public static void main(String[] args) {
9      Locale.setDefault(Locale.ENGLISH);
10     ClassPathXmlApplicationContext context = new
11     ClassPathXmlApplicationContext(
12     new String[] { "applicationContext.xml" }, true);
13     EmployerDao dao = (EmployerDao) context.getBean("employerDao");
14     List<EmployerDto> dtos = dao.getAll();
15     //System.out.println(type.toString());
16     // List<EmployerRoleDto> roles = new
17     ArrayList<EmployerRoleDto>(roleDao.selectAll());
18     //     for(EmployerRoleDto item : roles){
19     //         System.out.println(item.getId() + " " + item.getName() + " " +
20     item.getCreatedBy() + " " + item.getCreateDate());
21     //     }
22     System.out.println("Bye !");
23     System.exit(0);
24 }
25
26 }
```

ДОДАТОК В

Лістинг програми Employee.java

```
1  package com.subd.shop.bom;
2  import com.subd.shop.enumeration.EmployeeRoleEnum;
3  public class Employeeer extends Entity {
4  private String name;
5  private String login;
6  private String password;
7  private EmployeeerRole role;
8  public Employeeer() {
9  }
10 public boolean isAdmin() {
11     return
12     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.ADMIN).equals(r
13     ole.getId());
14 }
15 public boolean isManager() {
16     return
17     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.
18     MANAGER).equals(role.getId());
19 }
20 public boolean isBooker() {
21     return
22     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.B
23     OOKER).equals(role.getId());
24 }
25 public boolean isStorekeeper() {
26     return
27     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.S
28     TOREKEEPER).equals(role.getId());
29 }
30 public boolean isOperator() {
31     return
32     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.
33     OPERATOR).equals(role.getId());
34 }
35 public boolean isHeaver() {
36     return
37     EmployeeerRoleAsFormattedStringMapper.getIntVal(EmployeeerRoleEnum.
38     HEAVER).equals(role.getId());
39 }
40 public String getName() {
41     return name;
42 }
43 public void setName(String value) {
44     name = value;
45 }
46 public String getLogin() {
```

```
47         return login;
48     }
49     public void setLogin(String value) {
50         login = value;
51     }
52     public String getPassword() {
53         return password;
54     }
55     public void setPassword(String value) {
56         password = value;
57     }
58     public EmployeeRole getRole() {
59         return role;
60     }
61     public void setRole(EmployeeRole value) {
62         role = value;
63     }
64
```