

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Автоматизація роботи складу підприємства

Виконав студент 4 курсу групи К-41
Напрямок 6.050101 2 комп'ютерні науки,
Тарасов Анатолій Валерійович

Керівник асистент
Шуптар-Пориваєва Наталія Йосипівна

Консультант д.х.н., професор
Кругляк Юрій Олексійович

Рецензент: д.т.н., професор
Мещеряков Володимир Іванович

Одеса 2019

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	7
1 Аналіз предметної області і постановка завдання	9
1.1 Опис предметної області.....	9
1.2 Вимоги до WMS на підприємстві	12
1.3 Аналіз існуючих аналогів	13
1.4 Постановка завдання	21
2 Проектування системи.....	22
2.1 Програмні засоби розробки	22
2.2 Проектування системи UML-засобами.....	27
2.3 Структура БД.....	29
3 Опис програмного додатку	30
3.1 Інтерфейс користувача	30
3.2 SingleWindow як перспективний підхід	31
3.3 Вікно «Person Manager».....	33
3.4 Відправка вантажу на склад.....	36
3.5 Формування транспортної накладної.....	41
3.6 Відправка вантажу із складу	44
3.7 Планування маршрутом вантажівки	45
3.8 База даних підприємства	48
Висновки	50
Перелік джерел посилання	51

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – База Даних

ГІС – Геоінформаційні системи

ІС – Інформаційна Система

ОС – англ. Operating system, OS

ПЗ – Програмне Забезпечення

СУБД – Системи Управління Базами Даних

Android – Операційна система

API – Application Programming Interface

AWT – Abstract Window Toolkit

CPU – Central Processing Unit

GPU – Graphics Processing Unit

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

Java SE – Java 2 Standard Edition

JDBC – Java DataBase Connectivity

JVM – Java Virtual Machine

PL/SQL – Procedural Language / Structured Query Language

SQL – Structured Query Language

SWT – Standard Widget Toolkit

UML – Unified Modeling Language

WMS – Warehouse Management System

Терміни

Android Studio – середовища розробки під Android

Applets – аплети

IntelliJ – інтегроване середовище розробки програмного забезпечення для багатьох мов програмування

C – мова програмування

Class diagram – діаграма класів

Collaboration diagram – діаграма взаємодії об'єктів

C++ – мова програмування

JavaFx – платформа на основі Java для створення додатків з насиченим графічним інтерфейсом

JetBrains – міжнародна компанія, яка робить інструменти для розробки на різних мовах програмування

Microsoft SQL Server – система керування реляційними БД

MySQL – вільна реляційна система управління базами даних.

Sun Microsystems – американська компанія, виробник програмного і апаратного забезпечення

Swing – бібліотека для створення графічного інтерфейсу для програм на мові Java

Oracle – корпорація розробки ПЗ

Oracle Database – середовище роботи з БД

Oracle SQL Developer – інтегроване середовище розробки на мовах SQL і PL/SQL

Sequence diagram – діаграма послідовності взаємодій

State diagram діаграма – переходу станів

Use Case Diagram – діаграма варіантів використання

ВСТУП

В автоматизованих ІС частину функцій управління і обробки даних виконується комп'ютерами, а частина людиною. Виходячи із сучасних вимог, що пред'являються до якості роботи фінансового ланки великого підприємства, видно, що ефективна робота його цілком залежить від рівня оснащення компанії інформаційними засобами на базі комп'ютерних систем автоматизованого складського обліку [1]¹⁾.

Мета впровадження автоматизаційного управління у роботу підприємства складається з таких факторів, як:

- активне управління складом;
- збільшення швидкості набору товару;
- отримання точної інформації про місце знаходження товару на складі;
- ефективне управління товаром, що має обмежений термін придатності;
- отримання інструменту для підвищення ефективності та розвитку процесів з обробки товару на складі;
- оптимізація використання складських площ;
- оптимізація маршрутів для погрузчиків складу.

Реалізація проектування інформаційної системи обліку руху товару на складі підприємства значно полегшить роботу його співробітників і забезпечить можливість зменшити витрати на управління за рахунок звільнення людських ресурсів, зайнятих різними видами обробки паперових документів, зберігати і аналізувати дані за будь-який проміжок часу, здійснювати пошук потрібної інформації по різними критеріями відбору [2]²⁾.

¹⁾ [1] Информационные системы в управлении складированием. URL: https://www.lobanov-logist.ru/library/all_articles/54584/. (дата звернення 12.02.2019).

²⁾ [2] Інформаційний довідник MAS System. URL: <https://mas-ukraine.com.ua/>. (дата звернення 12.02.2019).

Метою роботи є аналіз діяльності роботи складу підприємства, моделювання основних функцій UML-засобами, створення програмного додатку для автоматизації роботи складу, підключення модуля для обчислення маршрута для погрузчиків складу.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 51
- кількість рисунків – 25
- кількість таблиць – 0
- кількість посилань – 22

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Опис предметної області

Спочатку необхідно розглянути предметну область як деяку сукупність реальних об'єктів і зв'язків між ними. Склад підприємства не займається ніякою комерційною діяльністю, а тільки здійснює процедуру зберігання продукції для сторонніх осіб зацікавлених в цьому.

Склад повинен виконувати наступні функції:

- прийом, облік, зберігання та відвантаження товарів;
- сортування продукції і комплектація споживачам;
- визначення потреби в транспортних засобах і механізованих навантажувальних засобах, а також тарі і робочій силі для відвантаження продукції;
- координація зберігання продукції з наявністю вільних складських площ, підготовка звітів про обсяги продукції, а також участь в розгляді надходять на підприємство претензій.

Склад повинен надати створення умов для збереження продукції, що знаходиться на тимчасовому зберіганні, організацію раціонального зберігання, внутрішнього транспортування.

Крім цього, необхідні умови для упаковки та підготовки продукції до відправлення, забезпечення збереження продукції, забезпечення високого рівня механізації і автоматизації транспортно-складських операцій, застосування комп'ютерних систем і нормативних умов організації і охорони праці.

Склад зобов'язаний вести облік продукції, що знаходиться на тимчасовому зберіганні, облік виконання замовлень по відвантаженню і розвантаженню готової продукції, складання звітів про завантаження складських площ.

Розглянемо типові бізнес-процеси складського обліку на не автоматизованому складі. Такий розгляд проводиться з метою виявити

недоліки існуючої системи складського обліку, а так само показати необхідність автоматизації складу.

Процедура прийняття продукції на склад виконується наступний чином:

- продукція приходить на склад в супроводі експедитора і прибуткової накладної;
- контролер на складі, перевіряє прибуткову накладну, і реєструє її в книзі обліку вхідних документів (накладних);
- після огляду стану продукції її приймають на склад, передаючи експедитору товару виписку (документ) про те, що товар прийнято на зберігання;
- вантажники відвозять товар у вільний місце зберігання, і контролер робить запис в книзі обліку про те, де зберігається знову надійшла продукція.

В ході роботи складу, він потребує інвентаризації, яка включає в себе такі стадії як:

- відповідальний працівник по переобліку продукції, в супроводі книги переобліку, відправляється на склад і вручну оглядає та переписує дані про товар і його кількості;
- після цього дані звіряються в книзі обліку товарів, особами відповідальними за документи звітності на складі і складається відповідний звіт, за даними переобліку продукції.

Послідовність відвантаження товарів:

- одержувач товару подає накладну на відвантаження товару;
- контролер перевіряє цю накладну і реєструє її в книзі обліку вхідних документів;
- далі контролер дає вказівку працівникам складу на пошук потрібної продукції і відвантаження її;
- потім одержувач товару проводить його огляд, на рахунок того чи потрібен товар відвантажили і в потрібній кількості;

- контролер реєструє в книзі обліку факт відвантаження товару;
- далі контролер видає одержувачу вантажу супровідний документ з відвантаження товару;
- далі відбувається безпосередньо відвантаження товару технічними засобами.

Проаналізувавши ситуацію на складі і виявивши всі мінуси, необхідно спроектувати таку систему, яка б автоматизувала наступні операції на складі:

- автоматизувати реєстрацію документів;
- пошук товарів для відвантаження буде проводитися шляхом пошуку відповідного товару в БД і перегляду інформації про місце його зберігання (номер складу);
- формування документів звітності буде здійснюватися через систему автоматично.

В результаті вся робота з паперами буде проводитися з використанням комп'ютерів, не потрібно буде возитися з купою документів. За допомогою ЕОМ на складі автоматизується облік надходження та відвантаження товарів, облік вхідних і вихідних документів, кількісний облік.

У загальному обсязі облікових робіт ці завдання мають значну питому вагу. Їх автоматизація дозволяє скоротити ручні операції, прискорити обробку інформації, підвищити точність обліку. Комп'ютер зберігає в пам'яті інформацію, яка може бути видана на друк, наприклад, детальна інформація про кількість надходження і відвантаження конкретного товару за кожним документом в разі розбіжності величини запасу з даними машинного обліку [3]¹⁾.

Головне призначення автоматизованої системи в даному випадку – підвищити ефективність виконання основних функцій працівників складу. Автоматизація управління процесами на складі, підвищує його оперативність і ефективність. Критеріями вибору технічних засобів є:

¹⁾ [3] Автоматизация бизнес-процессов. URL: <https://www.terrasoft.ua/crm/business-process-automation>. (дата звернення 13.02.2019).

- надійність функціонування системи;
- функціональна повнота системи;
- швидкодія;
- мінімізація витрат на вартість: апаратних засобів, прикладних систем, супроводу системи, розвитку системи.

1.2 Вимоги до WMS на підприємстві

Warehouse Management System (WMS) – система управління, що забезпечує автоматизацію та оптимізацію всіх процесів складської роботи профільного підприємства.

Основне завдання впровадження WMS – збільшення ефективності бізнесу через оптимізацію управління складськими процесами і підвищення прозорості цих процесів.

А нагородою за правильний вибір стає зростання прибутку підприємства завдяки зниженню витрат на утримання складу і підвищення рівня логістичного сервісу [4]¹⁾.

Будь-який замовник може вибрати програму управління складом, яка найбільш відповідає його вимогам: чи йде мова про невеликому складі або логістичному комплексі.

Вимоги до WMS визначаються галуззю бізнесу, специфікою складу, інтенсивністю його роботи, кількістю працівників, технологічними особливостями, складським обладнанням, яке потребує управління і до якого потрібно інтегруватися.

Пастка завдання підбору, в яку розробники програм нерідко втягують замовника, полягає в пропозиції порівняти функціональність різних рішень.

¹⁾ [4] Информационные системы в складской деятельности на примере WMS. URL: <https://novainfo.ru/article/5703>. (дата звернення 13.02.2019).

Небезпека такого підходу в тому, що ви спочатку вибираєте програму і потім адаптуєте до специфіки свого підприємства [5]²⁾.

Це призводить до значного затягування термінів впровадження і підміні мети: замість того щоб оптимізувати роботу складу, ви займаєтеся налагодженням працездатності автоматизованого рішення для управління ним [6]¹⁾.

Часта помилка замовників полягає в тому, що при виборі рішення для управління складськими процесами не береться до уваги подальший розвиток бізнесу. Порівнюється тільки функціональність WMS, терміни впровадження, вартість.

В результаті, коли через деякий час перед постачальником WMS-рішення ставиться завдання, виявляється, що зробити модернізацію з урахуванням нових вимог неможливо. Залишається або жити далі з тим, що є, або міняти систему управління.

Якщо автоматизація управління здійснюється на вже працюючому об'єкті, правильніше проводити запуск системи з зупинкою його роботи, проведенням інвентаризації, обробкою товару і завантаженням даних в систему. Тому краще заздалегідь планувати запуск в період порівняно низьку завантаженість складу [7]²⁾.

1.3 Аналіз існуючих аналогів

Неможливо уявити сучасне підприємство без комп'ютера, а комп'ютерне «залізо» перетворюється в корисного помічника в роботі тільки

²⁾ [5] Информационное обслуживание склада. URL: <http://www.west-pereezd.ru/informacionnoe-obslujivanie-sklada.html>. (дата звернення 14.02.2019).

¹⁾ [6] WMS система управления складом. URL: <http://www.netsoft.com.ua/WMS-Sistema-upravleniya-skladom.html>. (дата звернення 15.02.2019).

²⁾ [7] Обзор программ складского учета. URL: http://sklad-prog.ru/obzor/o_ind.htm. (дата звернення 15.02.2019).

в тому випадку, коли на ньому встановлено відповідне програмне забезпечення, яке адекватно працює.

Але такі програми не безкоштовні, їх потрібно купувати і підтримувати. Тому практично всі підприємства стикаються з необхідністю враховувати витрати на придбання програм.

Спочатку необхідно розглянути декілька існуючих програмних засобів по управлінню роботи складу. Першим аналогом буде програмний модуль «Складской учет товаров» 5.35.011.

Це проста, надійна і зручна програма для автоматизації торгівлі та складського обліку (рис. 1). Призначена для малого і середнього бізнесу, також вона буде прекрасним помічником для індивідуальних підприємців. ПЗ легко встановлюється, не вимагає великих зусиль для впровадження в бізнес, забезпечує швидку віддачу [7]¹⁾.

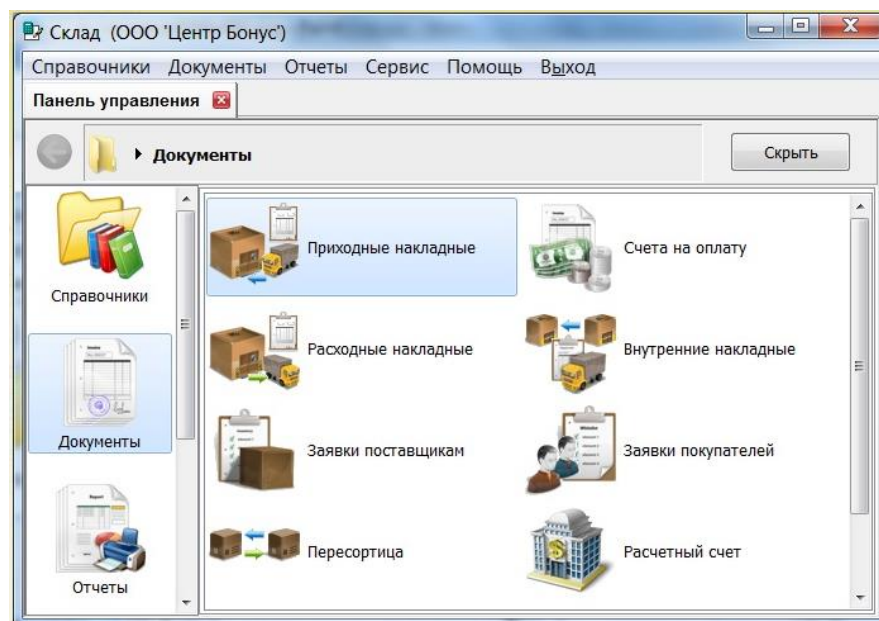


Рисунок 1 – Панель управління програмного модулю «Складской учет товаров» 5.35.011

¹⁾ [7] Обзор программ складского учета. URL: http://sklad-prog.ru/obzor/o_ind.htm. (дата звернення 15.02.2019).

Основні функції програмного продукту:

- аналіз продажів за різними критеріями;
- облік взаєморозрахунків з постачальниками і покупцями і контроль заборгованостей;
- розрахунок собівартості товарів і отриманого прибутку;
- виписка рахунків;
- резервування товарів;
- реєстрація та контроль оплати рахунків, надходжень і витрат товарів;
- формування і висновок на друк всіх необхідних облікових документів;
- друк етикеток і цінників;
- інвентаризація торгових точок;
- обмін даними з віддаленими торговими точками.

Недоліками є ціна та великий перелік функцій, які не є актуальними для оглядаємого підприємства.

Другим аналогом розглянемо програмний модуль «ОК-СКЛАД: Складской учет и торговля» (рис. 2):

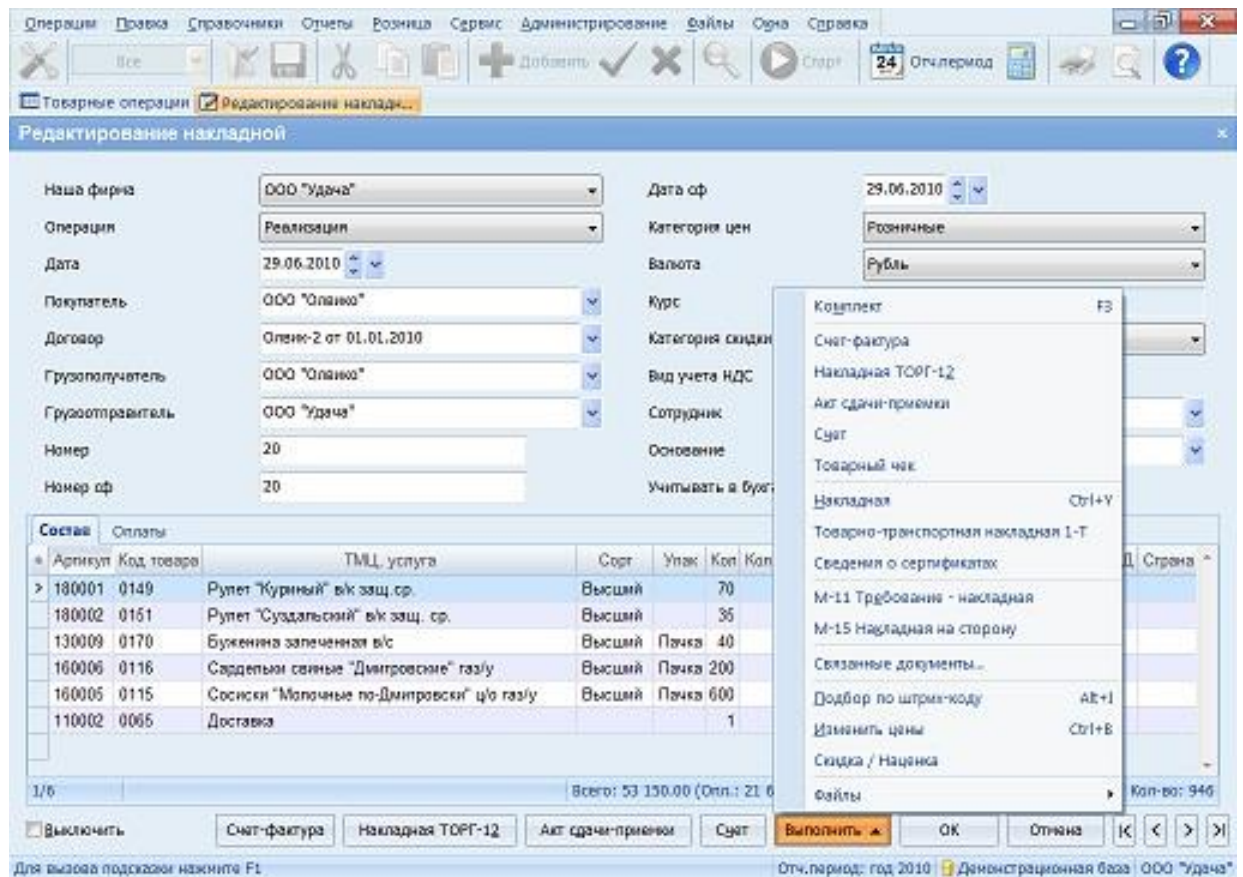


Рисунок 2 – Програмный модуль «ОК-СКЛАД: Складской учет и торговля»

Програма для складського обліку: зручна і потужна програма складського обліку, торгового бізнесу і виробництва. Відрізняється від інших зручним інтерфейсом і простотою розуміння обліку руху товарів на складах, грошових коштів і документів.

Доступність і логічність програми дозволяють використовувати її користувачам, які мають тільки основи роботи з комп'ютером і менеджерської практики. Однак програма високо професійна. У ній повний набір звітів про стан обліку на складах, безготівкових та готівкових грошей, договорів і т.д.

Складський облік можна вести по одному або декількох підприємствах. Кількість складів, кас, розрахункових рахунків не обмежена. У систему «ОК-Склад» можна вводити користувачів з необхідним набором прав доступу до інформації бізнесу.

Будь-які зміни, внесені в програму обліку користувачем, реєструються в журналі. Адміністратор складського обліку завжди зможе виявити, хто змінив дані, які саме і відновити історію подій в програмі.

Повна підтримка роботи сканера штрих кодів в додаткової версії. Дану програму використовують як великі оптові компанії так і роздрібні магазини. Експорт в Excel рахунків, накладних, платежів та інших документів. «ОК-Склад» може працювати з сервером бази даних через Інтернет [8]¹⁾.

Недоліком є дуже висока ціна даного програмного продукту та відсутність виходу поширеної і оновленої версії за останні 7 років. Таким чином, цей додаток можна рахувати старим і нераціональним на даний час.

Програмний модуль «Kataloger» також можна розглянути у ролі аналогу.

Kataloger – безкоштовний універсальний каталогізатор, завдяки якому користувач може систематизувати абсолютно будь-яку інформацію, будь то контактні дані друзів і знайомих (П.І.Б., адреса, телефон, e-mail та ін.), список дисків, книг, монет, велосипедів, одягу і багато іншого.

У програмі є можливість прикріплювати до будь-якого запису в таблиці коментар, а також файли і папки (рис. 3). Додаток не вимагає установки.

¹⁾ [8] ОК-СКЛАД 2.1.5.2310. URL: <https://www.softportal.com/software-10503-ok-sklad.html>. (дата звернення 14.02.2019).

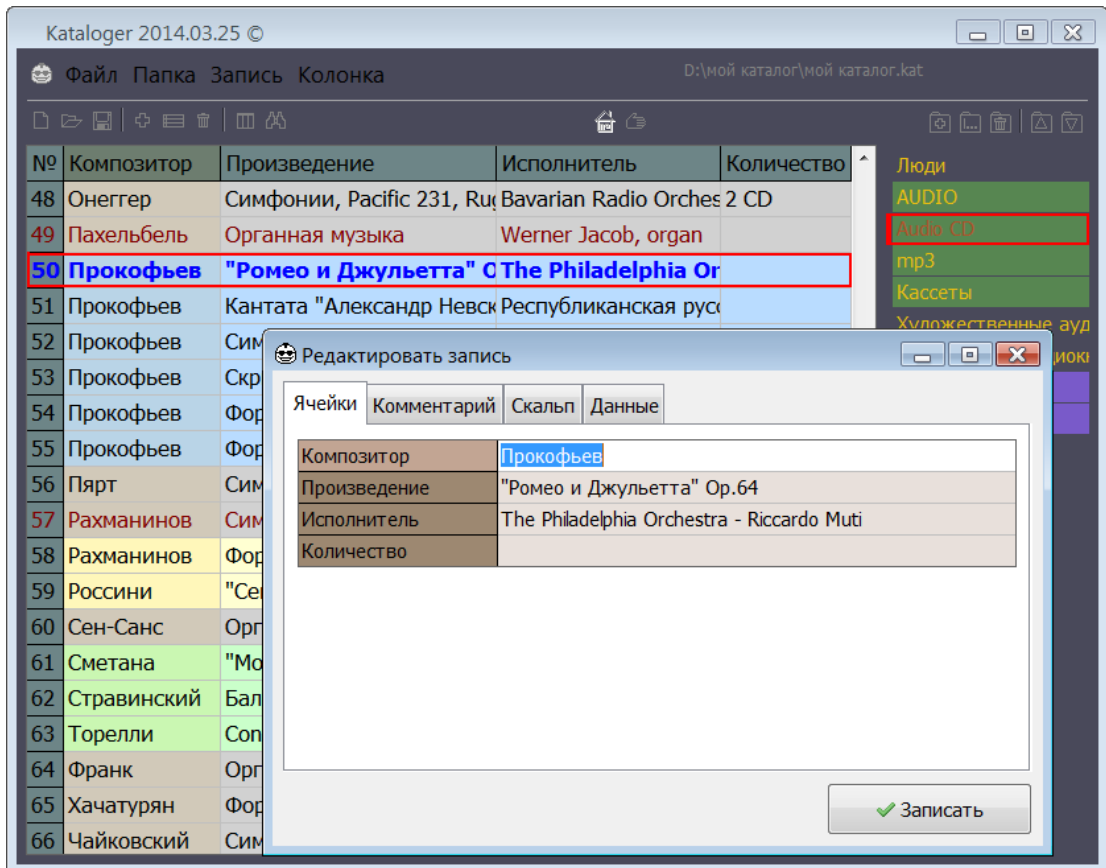


Рисунок 3 – Програмний модуль «Kataloger»

Основні можливості «Kataloger»:

- табличне відображення даних;
- не потрібно спеціальної установки: досить просто скопіювати exe-файл. Всі налаштування зберігаються в файлі «settings.ini» в папці програми);
- простий мінімалістичний зрозумілий інтерфейс;
- розумна сортування;
- можливість налаштування шрифтів (тип, розмір, стиль, колір).

До недоліків даного програмного продукту слід віднести відсутність україно-мовної версії, що сьогодні є обов'язковою вимогою до підприємств на території нашої країни [9]¹⁾.

¹⁾ [9] Kataloger 2019.23. URL: <https://www.softportal.com/software-43426-kataloger.html>. (дата звернення 15.02.2019).

Останнім аналогом розглянемо програмний модуль «ТріумфV2.0».

«ТріумфV2.0» – програма призначена для оперативного обліку товару і грошей (рис. 4).

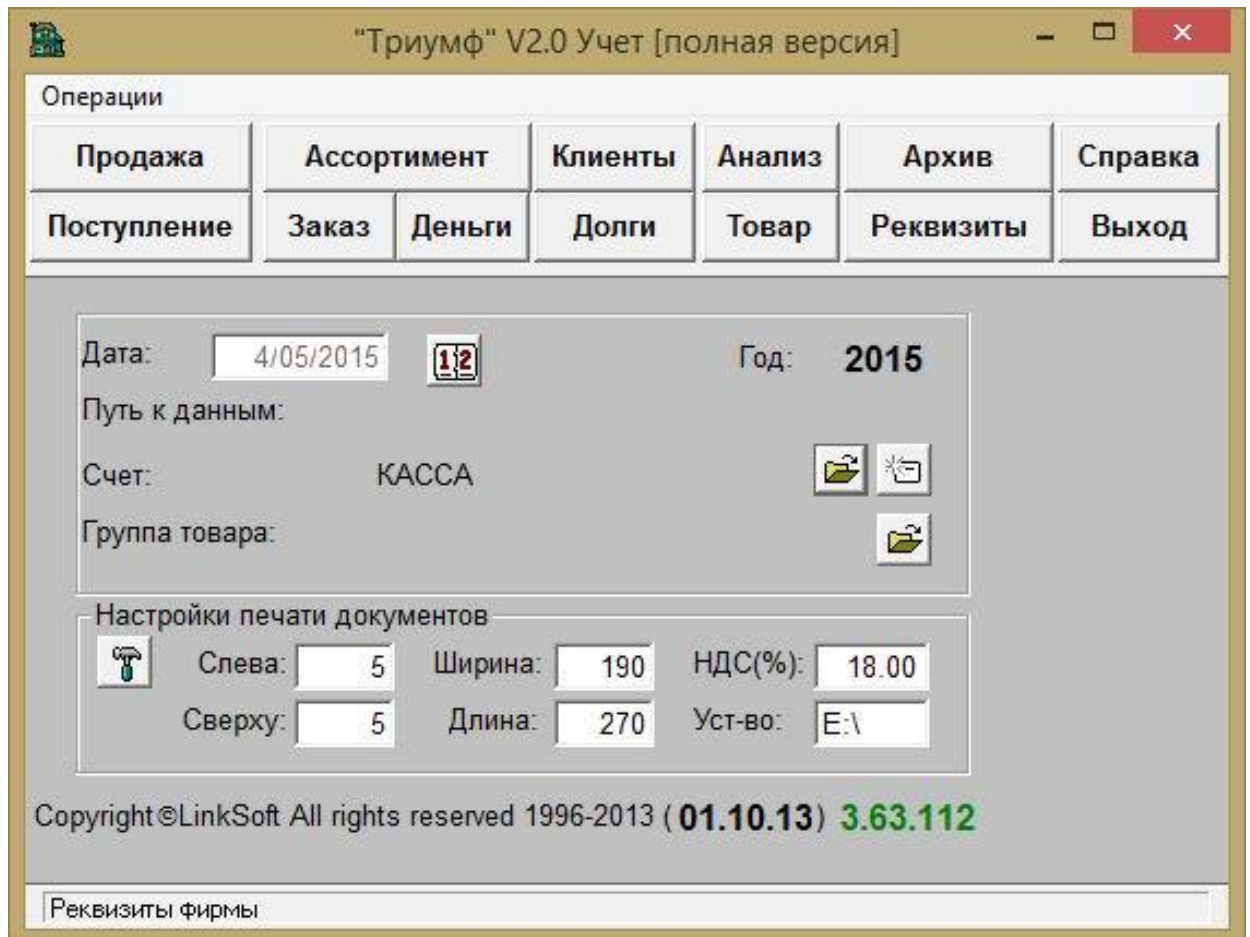


Рисунок 4 – Програмний модуль «ТріумфV2.0»

Можна вести облік в магазині або будь-якій торговельній точці, контролювати борги, залишки товару, вести облік продажів і надходження товару, витрата грошей і надходження грошей. У програмі є можливість прикріплювати до будь-якого запису в таблиці коментар, а також файли і папки.

Додаток не вимагає установки ви зможете контролювати продавців в магазині або на ринку. Складський облік можна вести по одному або

декількох підприємствах. Кількість складів, кас, розрахункових рахунків не обмежена.

Програма, за допомогою якої можна керувати процесом виробництва, починаючи з надходження окремих матеріалів на склад і закінчуючи випуском готового виробу. Склад виробництва буде корисний технологам, логіста, збирачам, фахівцям з постачання, начальникам виробництв, фахівцям складського обліку, бухгалтерам і генеральним директорам підприємств різних сфер діяльності і масштабів бізнесу. Для кожного фахівця в програмі є інформація тому програма охоплює весь виробничий цикл.

Кожен користувач має свій рівень доступу до програми і працює тільки в своїй робочій області. Безліч звітів, які дозволять відобразити оборот виробництва, відомість закупівлі матеріалів з урахуванням поставки його в упаковках та багато інших.

Просте управління і докладна відео інструкція дозволяє швидко навчити роботі з програмою навіть початківця користувача. Програма розробляється з 2003 року. Реалізовано безліч побажань, що дозволили зробити роботу з програмою максимально зручною та ефективною [10]¹⁾.

Деякі можливості програми:

- будь-яка кількість складів матеріалів;
- будь-які технологічні процеси виробництва;
- будь-яка кількість складів готових виробів;
- можливість автоматичного завантаження прайс листа, щоб не набивати довідник вручну;
- можливість градації цін на різних складах;
- заміна відсутнього матеріалу при відправці в виробництво.

Основні можливості:

- перегляд руху товару в надходженні і продажу;

¹⁾ [10] ТриумфV2.0 3.63.117. URL: <https://www.softportal.com/software-45131-triumfv20.html>. (дата звернення 15.02.2019).

- друк цінників, прайс-листів, залишків товарів, нових рахунків-фактур, накладних, прибуткових і видаткових касових ордерів, нових книг продажів і покупок.

За результатами огляду аналогів було прийнято керівництвом складу спроектувати і розробити ІС під свої потреби, що зменшить коштові витрати и максимально охопить ключові дії для данного підприємства, враховуючи відємності оглядів аналогів і бажання керівника підприємства (заказником проєкту).

1.4 Постановка завдання

Для дипломного проєкту поставлені наступні задачі:

- оглянути і проаналізувати предметну область складу підприємства для виявлення основних її бізнес-процесів;
- моделювання основних функцій роботи складу за допомогою UML-засобів;
- розробка програмного додатку для автоматизації роботи складу з використанням БД;
- створення додаткового модулю додатку для розрахунку переміщення погрузчиків та території складу за оптимальним маршрутом з метою економії часу та палива для погрузчиків.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Програмні засоби розробки

Для реалізації поставленої мети були обрані наступні програмні засоби:

- середовище розробки IntelliJ IDEA от JetBrains;
- мова програмування Java, технологія для GUI JavaFX;
- JDBC — бібліотека і драйвер для роботи з різними БД (OracleSB, MySQL і т.д.);
- SQL Developer — для роботи з OracleDB;
- OracleDB.

СУБД Oracle – це єдине інтегроване рішення, для управління інформацією, доступне на різних відкритих платформах і підтримуване безліччю сторонніх додатків. У базі даних зберігаються не тільки реляційні дані, але і документи, XML, мультимедіа інформація про розміщення даних і ГІС інформація. Продуктивність системи оптимізована для перерахованих видів даних, наприклад файли, що зберігаються в базі з використанням функції "Безпека файлів", можуть бути відправлені на зберігання і витягнуті назад швидше, ніж в інших системах. Для всіх видів даних в базі здійснюється індексування і надається SQL-доступ, дозволяючи легко впроваджувати нові види даних в існуючі програми. Також надається звичайний порядок доступу за протоколами, характерним для конкретних видів даних [11]¹⁾.

Oracle SQL Developer – інтегроване середовище розробки на мовах SQL і PL/SQL, з можливістю адміністрування баз даних, орієнтована на застосування в середовищі Oracle Database. Корпорація Oracle надає продукт

¹⁾ [11] СУБД Oracle Database 11g. URL: <https://oracle-patches.com/oracle/prof/3139-D1%81%D1%83%D0%B1%D0%B4-oracle-database>. (дата звернення 01.03.2019).

безкоштовно. Саме середовище написана на мові програмування Java, працює на всіх платформах, де доступна Виконавча Java SE [12]²⁾.

Oracle SQL Developer являє собою графічну версію, яка дає розробникам баз даних зручний спосіб виконання основних завдань. Тут є можливість переглядати, створювати, редагувати і видаляти; запускати оператори SQL і скрипти; маніпулювати і експортувати (вивантажувати) дані; переглядати і створювати звіти [11]¹⁾.

Є можливість підключитися до будь-якої цільової схемою Oracle Database, використовуючи стандартну аутентифікацію Oracle Database. Після підключення виконуються операції над об'єктами в базі даних. Також реально підключитися до схем для MySQL і обраних сторонніх (не Oracle) баз даних, таких як Microsoft SQL Server, а також переглядати метадані і дані в цих базах даних чи перенести ці бази даних в Oracle Database [12]²⁾.

JDBC – це прикладний програмний інтерфейс для виконання SQL-запитів. Він складається з безлічі класів і інтерфейсів, написаних на мові Java. JDBC надає стандартний API для розробників, які використовують бази даних. За допомогою JDBC можна писати програми на мові Java, що використовують БД, легко відсилати SQL-запити майже до всіх реляційних БД.

Використання JDBC API позбавляє від необхідності для кожної СУБД (Informix, Oracle і т.д.) писати свій додаток. Достатньо написати одну єдину програму, яка використовує JDBC API, і ця програма зможе відсилати SQL-

²⁾ [12] Обзор использования Oracle SQL Developer. URL: <http://sql-oracle.ru/obzor-ispolzovaniya-oracle-sql-developer.html>. (дата звернення 01.03.2019).

¹⁾ [11] СУБД Oracle Database 11g. URL: <https://oracle-patches.com/oracle/prof/3139-%D1%81%D1%83%D0%B1%D0%B4-oracle-database>. (дата звернення 01.03.2019).

²⁾ [12] Обзор использования Oracle SQL Developer. URL: <http://sql-oracle.ru/obzor-ispolzovaniya-oracle-sql-developer.html>. (дата звернення 01.03.2019).

запити до необхідної БД. Крім того, це прокладання буде переносимо на різні платформи [13]³⁾.

JDBC розширює і без того багату функціональність Java. Наприклад, можна опублікувати в Інтернет веб-сторінку, яка містить аплет, пов'язаний з БД на сервері.

Ще один приклад: організація за допомогою JDBC може підключити всіх співробітників до однієї БД, навіть якщо ми маємо справу з конгломератом операційних систем на робочих станціях співробітників [14]¹⁾.

Платформа IntelliJ, розроблена JetBrains, стала основою для Android Studio, середовища розробки під Андроїд. Це вже другий реліз з того моменту, як рік тому компанія-розробник прийняла рішення відкрити вихідний код проекту.

IntelliJ IDEA 10 позиціонується як сама інтелектуальне середовище розробки для Java. У новій версії додані поліпшення користувальницького інтерфейсу, досягнуто дворазове прискорення індексування коду при відкритті проекту, перероблена система автодоповнення коду [15]²⁾.

Також в числі найбільш суттєвих доповнень – включення в IntelliJ IDEA засобів розробки мобільних додатків під ОС Android, які раніше були доступні тільки в платній версії. Покращена підтримка технологій програмування і бібліотек.

В цілому, JetBrains, середа IntelliJ IDEA краще розуміє контекст і дозволяє розробляти додатки швидше і простіше, ніж в інших середовищах. IntelliJ IDEA (рис. 5) індексує весь проект, аналізує все, що він містить, і будує синтаксичне дерево. Завдяки цьому в будь-який час, куди б розробник

³⁾ [13] 1.4 Oracle SQL Developer. URL: http://www.askit.ru/custom/oracle10sql/m1/01_04_oracle_sql_developer.htm. (дата звернення 10.03.2019).

¹⁾ [14] Руководство по JDBC v.1. URL: <http://www.javaportal.ru/java/tutorial/tutorialJDBC/intro.html>. (дата звернення 10.03.2019).

²⁾ [15] Руководство по JDBC. Введение.. URL: <https://proselyte.net/tutorials/jdbc/introduction/>. (дата звернення 10.03.2019).

ні помістив курсор, середя знає, де він знаходиться, і що можна в цьому місці зробити з кодом.

Наприклад, IntelliJ IDEA вміє автоматично завершувати вираження, які розробник починає або продовжує набирати з середини. Середовище включає засоби для тестування сумісності додатків з різними версіями платформи і редактор призначеного для користувача графічного інтерфейсу.

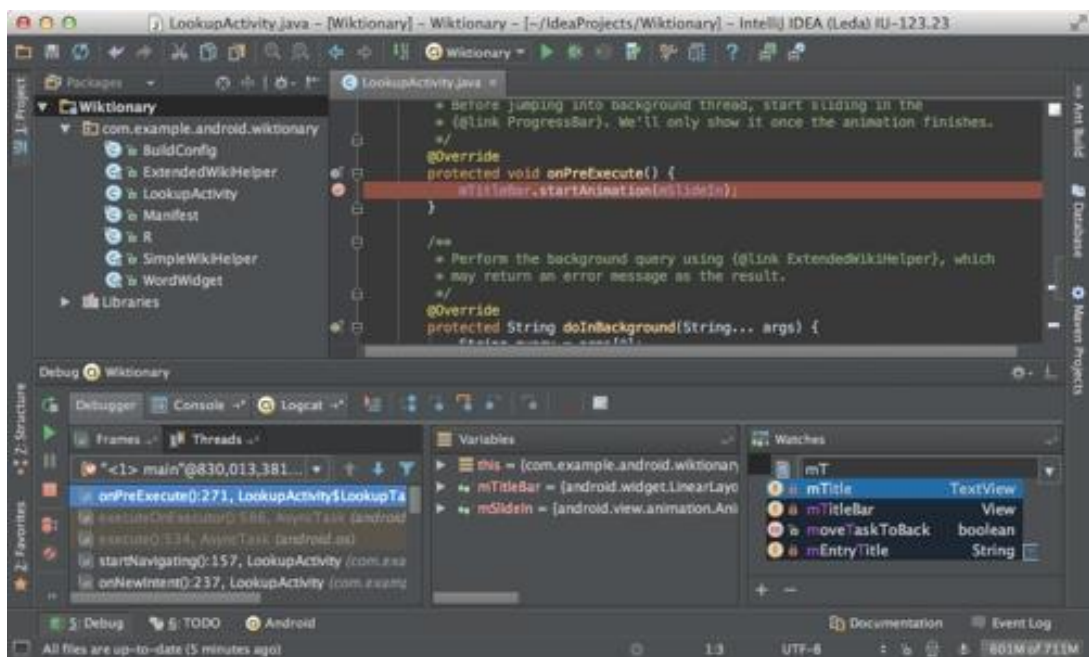


Рисунок 5 – Середя розробки IntelliJ IDEA

Програми на Java можуть бути трансльовані в байт-код, що виконується на віртуальній java-машині (JVM) – програмою, обробній байт-код і передавальній інструкції обладнанню, як інтерпретатор, але з тією відмінністю, що байт-код, на відміну від тексту, обробляється значно швидше. Мова Java зародився як частина проекту створення передового програмного забезпечення для різних побутових приладів.

Java – об'єктно-орієнтована мова програмування, що розробляється компанією Sun Microsystems з 1991 року і офіційно випущений 23 травня 1995 року.

Java став використовуватися для написання аплетів, додатків і серверного програмного забезпечення [16]¹⁾. Реалізація проекту була почата на мові C++, але незабаром виник ряд проблем, найкращим засобом боротьби з якими була зміна самого інструмента – мови програмування.

Стало очевидним, що необхідний платформи-незалежний мова програмування, що дозволяє створювати програми, які не доводилося б компілювати окремо для кожної архітектури і можна було б використовувати на різних процесорах під різними операційними системами [16]¹⁾.

Мова Java потрібна для створення інтерактивних продуктів для мережі Internet. Фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з C і C++. В Java використовуються практично ідентичні угоди для оголошення змінних, передачі параметрів, операторів і для управління потоком виконання коду. В Java додані всі хороші риси C++.

Три ключові елементи об'єдналися в технології мови Java:

- Java надає для широкого використання свої аплеті (applets) – невеликі, надійні, динамічні, які не залежать від платформи активні мережеві додатки, що вбудовуються в сторінки Web. Аплети Java можуть налаштовуватися і поширюватися споживачам з такою ж легкістю, як будь-які документи HTML;
- Java вивільняє міць об'єктно-орієнтованої розробки додатків, поєднуючи простий і знайомий синтаксис з надійним і зручним в роботі середовищем розробки. Це дозволяє широкому колу програмістів швидко створювати нові програми і нові аплеті;
- Java надає програмісту багатий набір класів об'єктів для ясного абстрагування багатьох системних функцій, використовуваних при роботі з вікнами, мережею і для введення-виведення. Ключова риса

¹⁾ [16] Окошко с кнопками на JavaFX. URL: <https://habr.com/ru/post/420003/>. (дата звернення 20.03.2019).

¹⁾ [16] Окошко с кнопками на JavaFX. URL: <https://habr.com/ru/post/420003/>. (дата звернення 20.03.2019).

цих класів полягає в тому, що вони забезпечують створення незалежних від використовуваної платформи абстракцій для широкого спектра системних інтерфейсів [17]²⁾.

2.2 Проектування системи UML-засобами

Моделювання системи проводиться як спуск по рівнях: від концептуальної моделі до логічної, а потім до фізичної моделі програмної системи. Концептуальна модель виражається у вигляді діаграм варіантів використання (use case diagram) [18]¹⁾.

Цей тип діаграм служить для проведення ітераційного циклу загальної постановки завдання разом із замовником.

Логічна модель дозволяє визначити два різних погляди на систему: статичний і динамічний. Ці погляди перетворюються у різні підходи до моделювання системи. Статичний підхід виражається діаграмами класів (class diagram).

Саме діаграми класів є основою для генерації програмного коду цільовою мовою програмування. Можливе дуже гнучке настроювання генерації коду з врахуванням конкретних угод (наприклад, за префіксами імен ідентифікаторів), прийнятих у команді розробників проекту.

У найпростішому випадку варіант використання створюється в процесі обговорення з користувачами тих речей, які вони хотіли б одержати від системи. При цьому кожній окремій функції привласнюється ім'я й записується її короткий текстовий опис.

Це все, що необхідно у фазі аналізу. Знання деяких деталей може знадобитися, якщо передбачається, що даний варіант використання містить

²⁾ [17] Как работает Java машина (JVM). URL: <https://idurdyev.com/kak-rabotaet-jvm>. (дата звернення 22.03.2019).

¹⁾ [18] Общая характеристика языка UML. URL: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2>. (дата звернення 23.03.2019).

важливі архітектурні відгалуження. Більшість варіантів використання може бути деталізована під час конкретної ітерації в процесі проектування [19]²⁾.

Спочатку необхідно визначити основні бізнес-процеси у системі «Склад» (рис. 6). Тут розглянуто взаємодію двох акторів «Склад» і «Вантаж». Крім цього, при розробці системи слід чітко визначити основні варіанти використання для процесу реєстрації і перевезення вантажу по території складу (рис. 7).

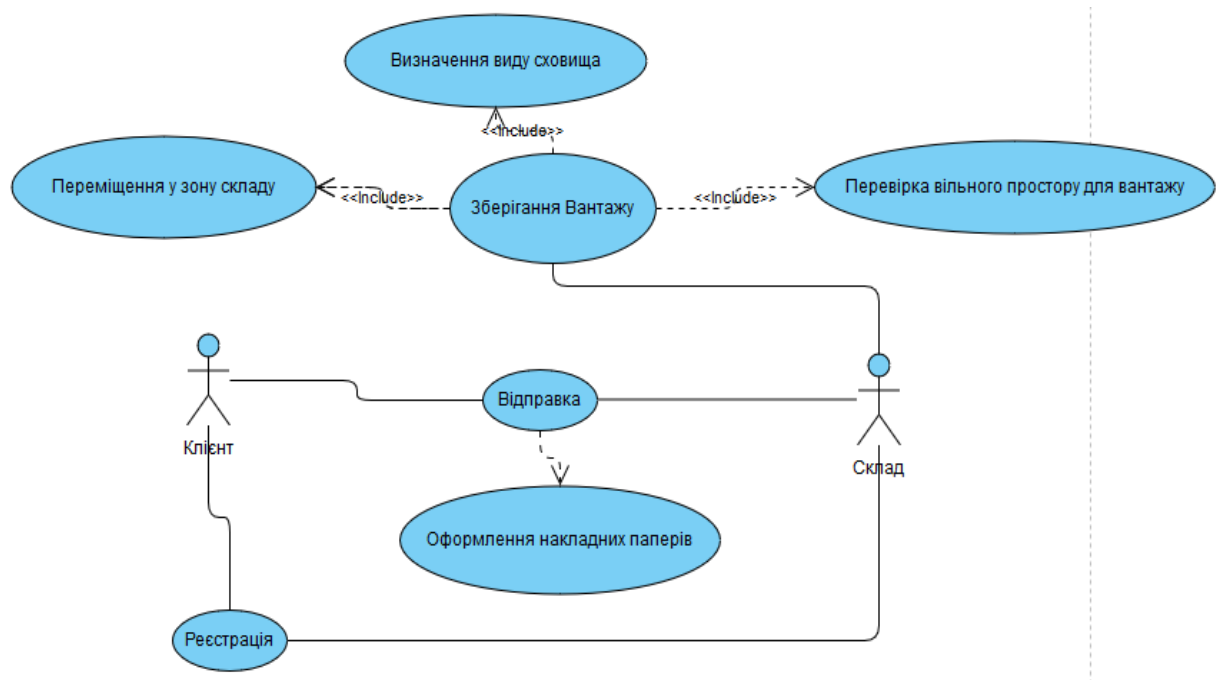


Рисунок 6 – Діаграма Use-Case «Склад»

²⁾ [19] Что такое Юзкейс (Use Case) или "Сценарий Использования" в Тестировании ПО? URL: <https://software-testing.org/testing/chto-takoe-yuzkeys-use-case-ili-scenariy-ispolzovaniya-v-testirovanii-po.html>. (дата звернення 27.03.2019).

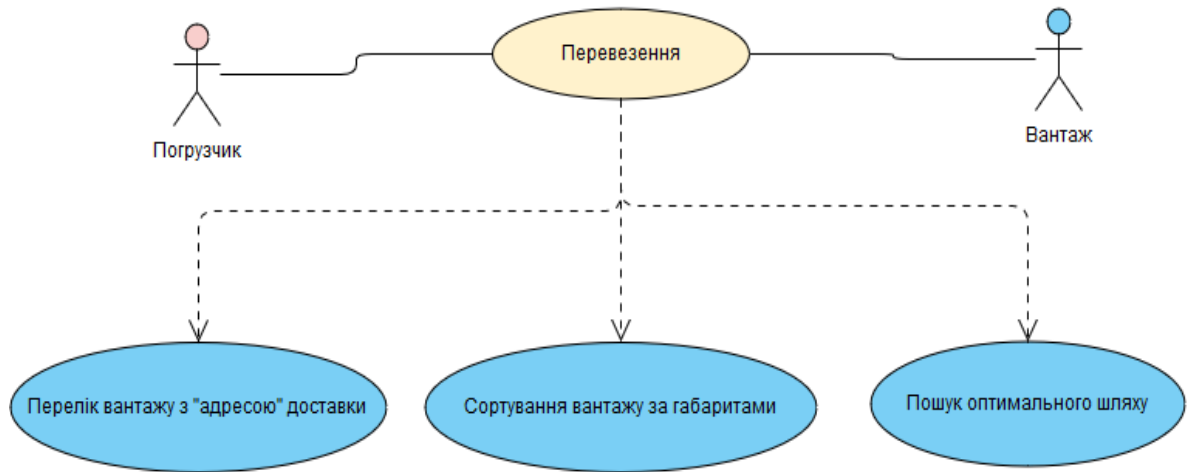


Рисунок 7 – Діаграма Use-Case «Розміщення на складі»

2.3 Структура БД

Так як у роботі складу задіяно багато інформації, наступним етапом розробки системи є проектування структури бази даних (рис. 8):

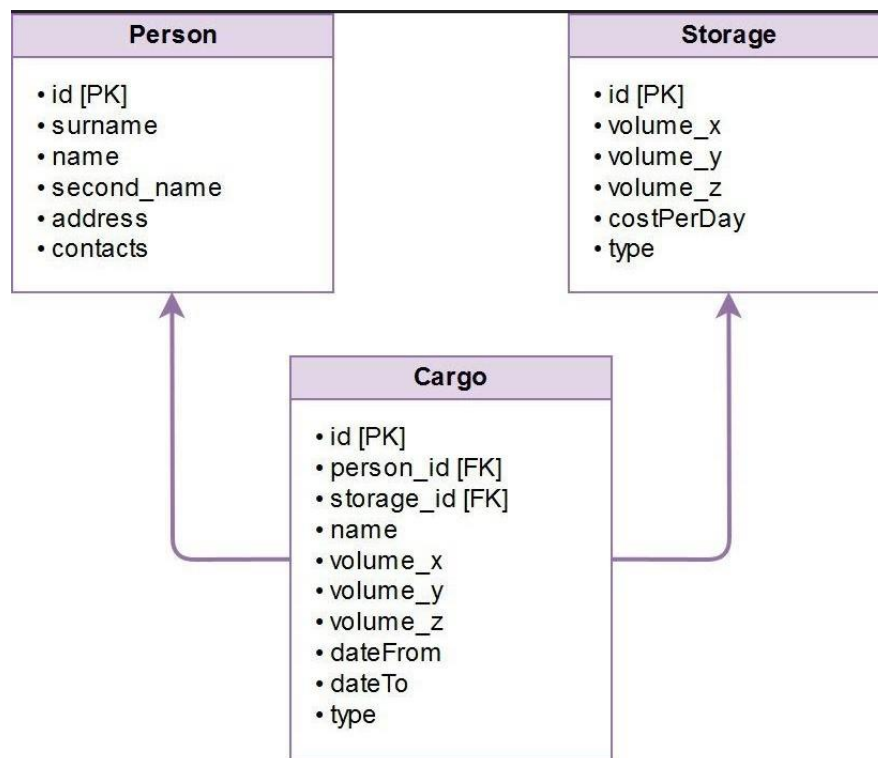


Рисунок 8 – Структура бази даних для системи «Склад»

Буде достатньо побудови трьох таблиць:

- «Person» – тут зберігається інформація щодо клієнтів підприємства;
- «Storage» – різні види сховища на території складу;
- «Cargo» – повна інформація щодо вантажу, з контактами про його і власника.

3 ОПИС ПРОГРАМНОГО ДОДАТКУ

3.1 Інтерфейс користувача

У Java є як мінімум чотири технології для реалізації візуального інтерфейсу користувача, і це AWT, SWT, Swing і нарешті JavaFx. AWT є базовий інтерфейс, але він залежить від системи, і при застосуванні його в платформ він буває дає збої. Зазвичай ця технологія застосовується для зображення нестандартних елементів (трекутник, криві і т.д.).

Як і інші три технології AWT обробляє графіку за рахунок CPU а не GPU, але з усіх запропонованих технологій та ж сама продуктивна за швидкістю. Але для звичайного GUI це не потрібно. SWT це якийсь проміжний етап між AWT і Swing, і належних плодів він не дав.

Swing свого часу був проривним. Він базується на AWT, але тим не менш гарантує стабільність роботи на різних ОС. Так само він більш гнучкий на відміну від AWT, що виразно плюс.

Останньою технологією йде JavaFx. Це – глобальна технологія яка спрямована на дуже гнучкий і незалежний дизайн. Основними відмінностями від попередніх технологій в тому, що тут дизайн будь-якого компонента

задається через CSS, що дає можливість стилізувати компонент як захочеться [20]¹⁾.

Практично повністю перевизначена ієрархія контейнерів і компонентів, що дало абсолютною зв'язність компонентів. Іншими словами, тепер будь-який компонент міг бути представлений як контейнер і адекватно местить в собі інші компоненти, що Swing в свою чергу дозволяв таке дуже обмежена. І хоч JavaFx з усіх є самою повільною (хоча для цього додатка це не суттєво), вона виразно найгнучкіша, тому була обрана саме вона [16]²⁾.

3.2 SingleWindow як перспективний підхід

Для роботи з менеджером потрібно всього вікно для взяття вантажу зі складу та вікна для відправки вантажу на склад. І так як ці два вікна в принципі мали практично ідентичний інтерфейс, то для визначення режиму роботи був введений компонент «ChoiseBox», який міг приймати лише два значення – «Прийом вантажу на склад» або «Взяття вантажу зі складу».

Це було непоганим рішенням, але мало ряд незручностей, а саме пошук і додавання людини в БД. Логічно, що вантаж на складі повинен прив'язуватися ні до порожнього місця і мати лише якийсь ID, а до людини. А людина в свою чергу крім основних даних таких, як ПІБ, повинен мати адресу і контактні дані для можливості зв'язатися в разі потреби. І було незручно з цим оперувати, занадто багато елементів взаємодії (рис. 9).

¹⁾ [20] Руководство по JavaFX. URL: <https://metanit.com/java/javafx/>. (дата звернення 11.03.2019).

²⁾ [16] Окошко с кнопками на JavaFX. URL: <https://habr.com/ru/post/420003/>. (дата звернення 20.03.2019).

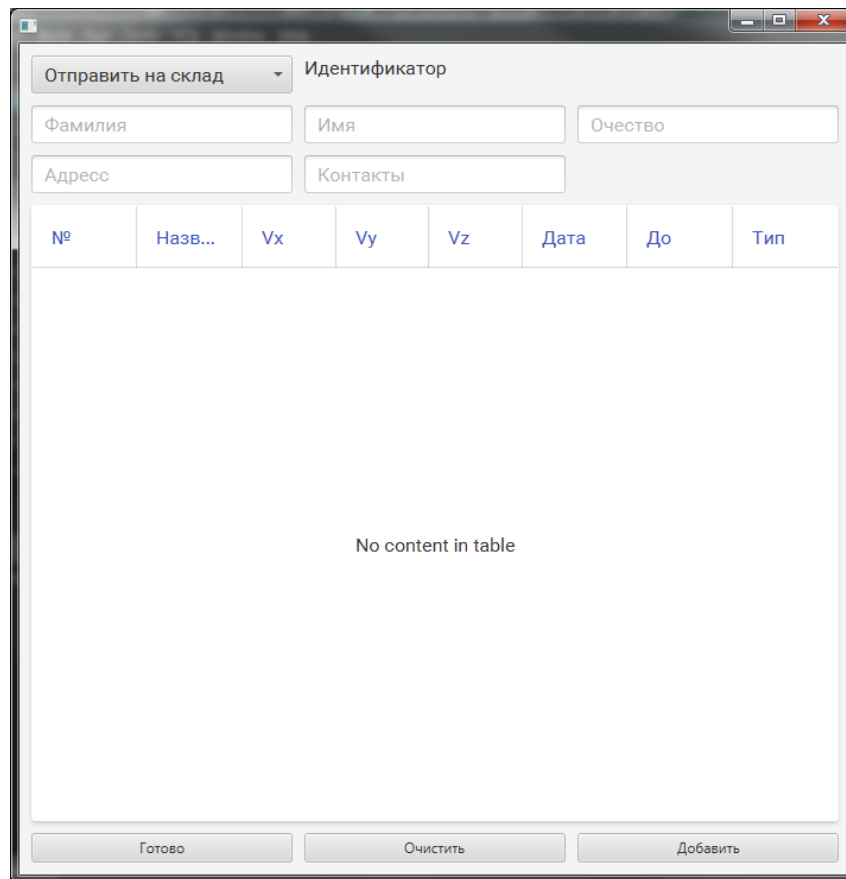


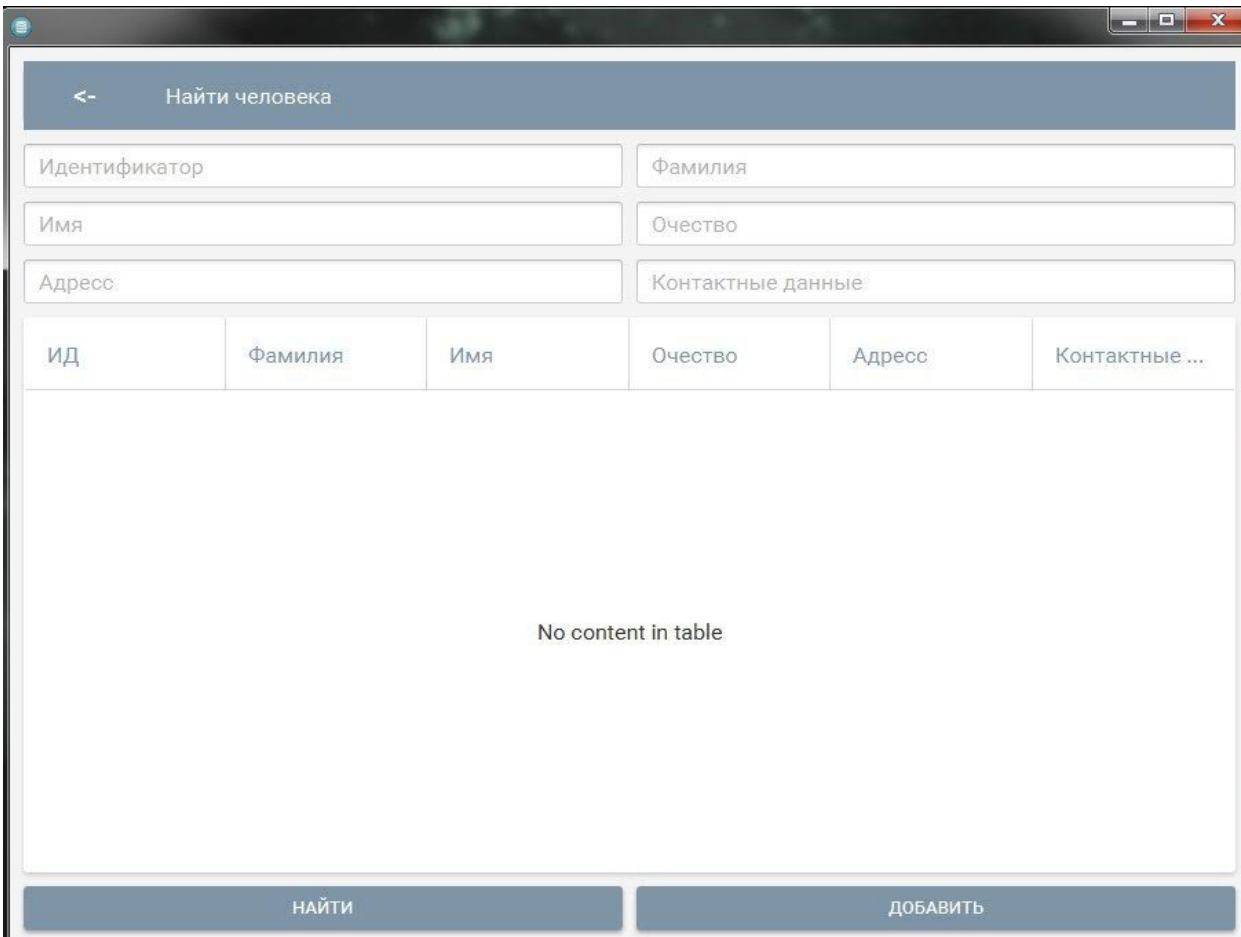
Рисунок 9 – Інтерфейс користувача на першій реалізації додатку

Велика кількість кнопоч і текстових полів не зручно для інтерфейсу і вигідніше розділити його на кілька вікон. Так в Android додатках це замінюється на «Activity».

Ні інтерфейс, ні дизайн не задовольняв вимогам замовника, і тому було прийнято рішення вдатися до Android-стилізації, а саме «SingleWindow» і «Material Design».

Перспектива ідеї «SingleWindow» в тому, що такий підхід можна застосовувати на різних ОС і він легко портіруем. Існує три основні сцени – «PersonManager», «ToStorage» і «FromStorage».

Так, після старту програмного додатку керування складом, для користувача відкривається головна форма (рис. 10).



Найти человека

Идентификатор Фамилия

Имя Очество

Адресс Контактные данные

ИД	Фамилия	Имя	Очество	Адресс	Контактные ...
No content in table					

НАЙТИ ДОБАВИТЬ

Рисунок 10 – Стартове вікно додатку

3.3 Вікно «Person Manager»

Це вікно має шість текстових полів для введення ID, ПІБ, адреса і контактних даних відповідно. Так само є таблиця, де відображуються люди і кнопки «Знайти» і «Додати».

При запуску вікна в таблиці відразу відображаються всі люди з БД, але їх можна відфільтрувати, вводячи дані у відповідні текстові поля (рис. 11).

ИД	Фамилия	Имя	Очество	Адресс	Контактные ...
0	Fan	Nam	Ot	Ad	Con
1	Фам	Имя	Ot	Ад	Кон
2	Marshall	Bruce	Matters	Washington	***

Рисунок 11 – Вікно «Person Manager»

У разі, якщо необхідно додати людину у базу даних (рис. 12), потрібно натиснути на кнопку «Добавить», яка змінює сцену вікна на додавання .

У цьому вікні визначаються всі дані людини:

- ідентифікаційний номер;
- прізвисьце;
- імя;
- по батькові;
- адреса;
- контактна інформація (теленон або адреса електронної скрині).

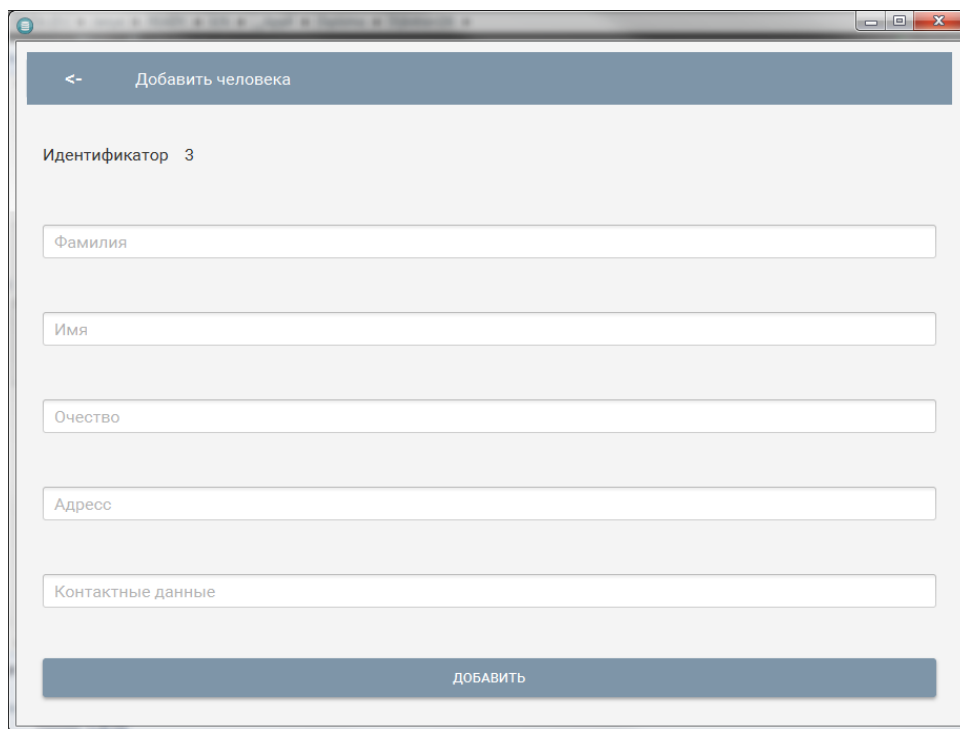


Рисунок 12 – Вікно «Add Person»

Далі людина додається у БД. Наприклад, якщо якийсь параметр не введений, зреагує Alert-вікно з помилкою про незаповнення поле (рис. 13).

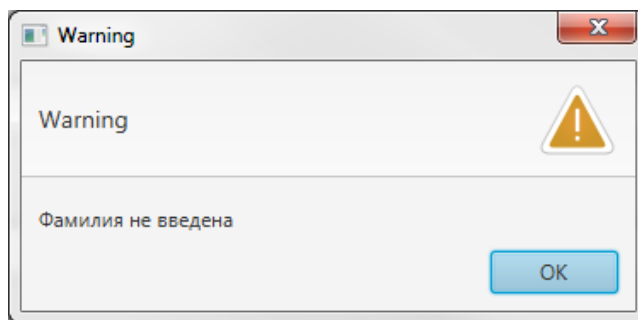


Рисунок 13 – Вікно «Alert-Massege»

Наступний код демонструє запит до БД за допомогою JDBC. Спочатку створюється об'єкт типу «Connection», який пов'язує БД і Java програму, а потім з його допомогою запит:

```
Connection connection = getConnection();
```

```
Statement statement = connection.createStatement();
```

Потім виконується запит і отриману відповідь деконвертується в об'єкт типу «Person», який в свою чергу поміщається в список.

```
ResultSet set = statement.executeQuery  
(builder.toString());  
while (set.next())  
persons.push (buildPerson(set));
```

І в останню чергу необхідно закрити потоки даних з БД, щоб це не навантажувати роботу програми.

```
set.close();  
statement.close();  
connection.close();
```

3.4 Відправка вантажу на склад

Після проходження процедури реєстрації людина буде занесена у БД, після чого вона може залишити вантаж на зберігання. Основними елементами інтерфейсу в цьому вікні є таблиця з доданими вантажами і кнопки «Додати» і «Відправити». Натиснувши на кнопку «додати» вікно змінить сцену на «додавання гвантажу», подібно зі зміною вікна на додавання нової особистості. Потрібно визначити відповідні поля (рис. 14), а саме:

- назву;
- обсяг по $x / y / z$;
- дата зберігання;
- тип вантажу для вибору області складу для зберігання (рис. 15).

Добавить груз

Идентификатор 2

Мясо

10 2 2

24.01.2019

Холодильник

ГОТОВО

Рисунок 14 – Вікно «Додавання Вантажу»

Добавить груз

Идентификатор 1

Название

Объём(x) Объём(y) Объём(z)

До

Обычный
Мезонин
Холодильник

ГОТОВО

Рисунок 15 – Тип сховища для вантажу

Після цього, вантаж поміщається в таблицю вантажів. Потім натиснувши на кнопку «Відправити» буде вироблено обчислення складів, куди будуть поміщені вантажі щодо їх розміру і типу.

У разі, якщо на складі місця не вистачає, з'явиться Alert-вікно (рис. 16), яка оповіщає про те, що якийсь конкретний вантаж неможливо помістити на склад. При цьому інакше, вікно-повідомлення щодо успішного зарахування вантажу (рис. 17).

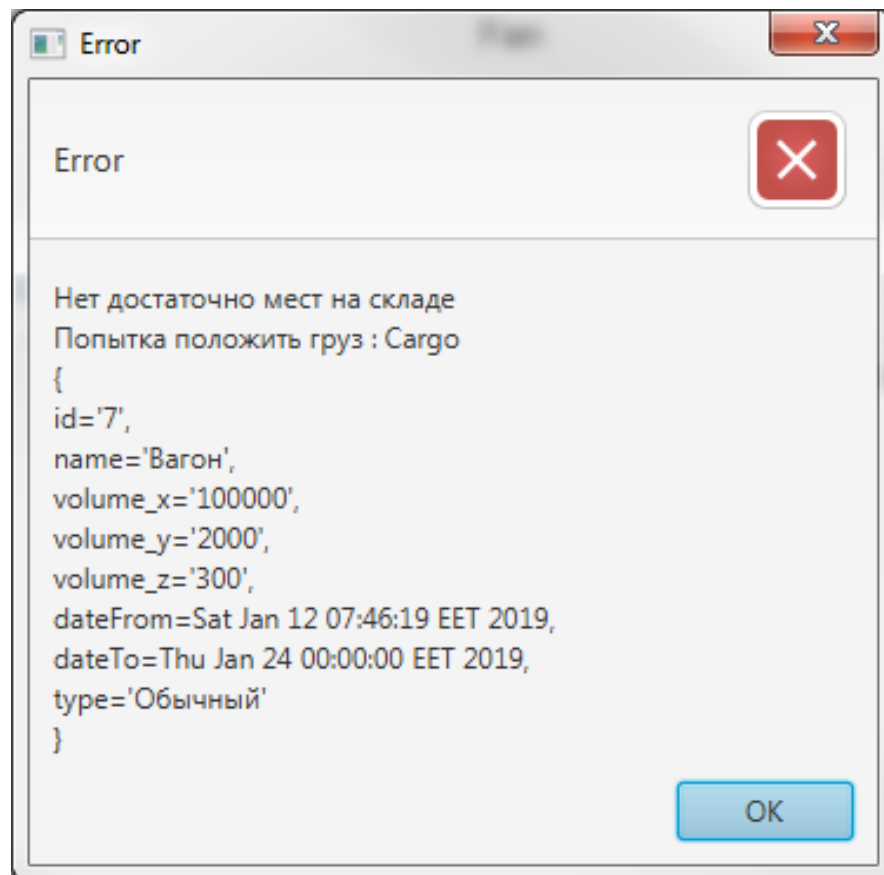


Рисунок 16 – Вікно-повідомлення про неможливість виконання операції

Код для перевірки наявності вакантного місця для вантажу має наступний вигляд:

```
Iterator iterator = cargos.iterator();
while (iterator.hasNext())
{
```

```

Cargo cargo = (Cargo) iterator.next();
boolean put = false;
for
(
    LVertex<Storage> vertex =
storages.getFirst();
    vertex != null;
    vertex = vertex.next()
)
    if (vertex.getItem().trynaPut(cargo))
    {
        put = true;
        break;
    }

if ( ! put)
{
    AlertOutput.getInstance().outExc("Нет
достаточно мест на складе\nПопытка положить груз : " + cargo);
    return;
}
}

```

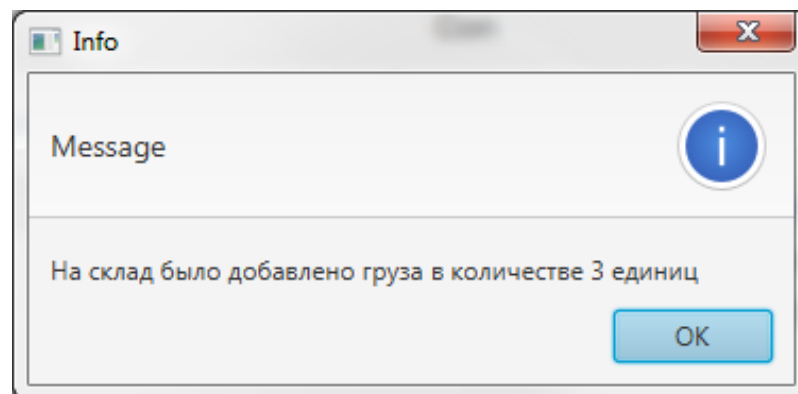


Рисунок 17 – Вікно-повідомлення о вдалої операції щарахування

Код для вікна-повідомлення при успішному зарахуванні вантажу на склад має наступний вигляд:

```

AlertOutput.getInstance().outInfo("на склад было добавлено
груза в количестве " + cargos.size() + " единиц");
cargos.clear();

```

Наступним кроком всі одиниці вантажу успішно будуть зараховані в базу даних (рис. 18). Повни перелік інформації буде розташовано на формі «Зарахування Вантажу на склад».

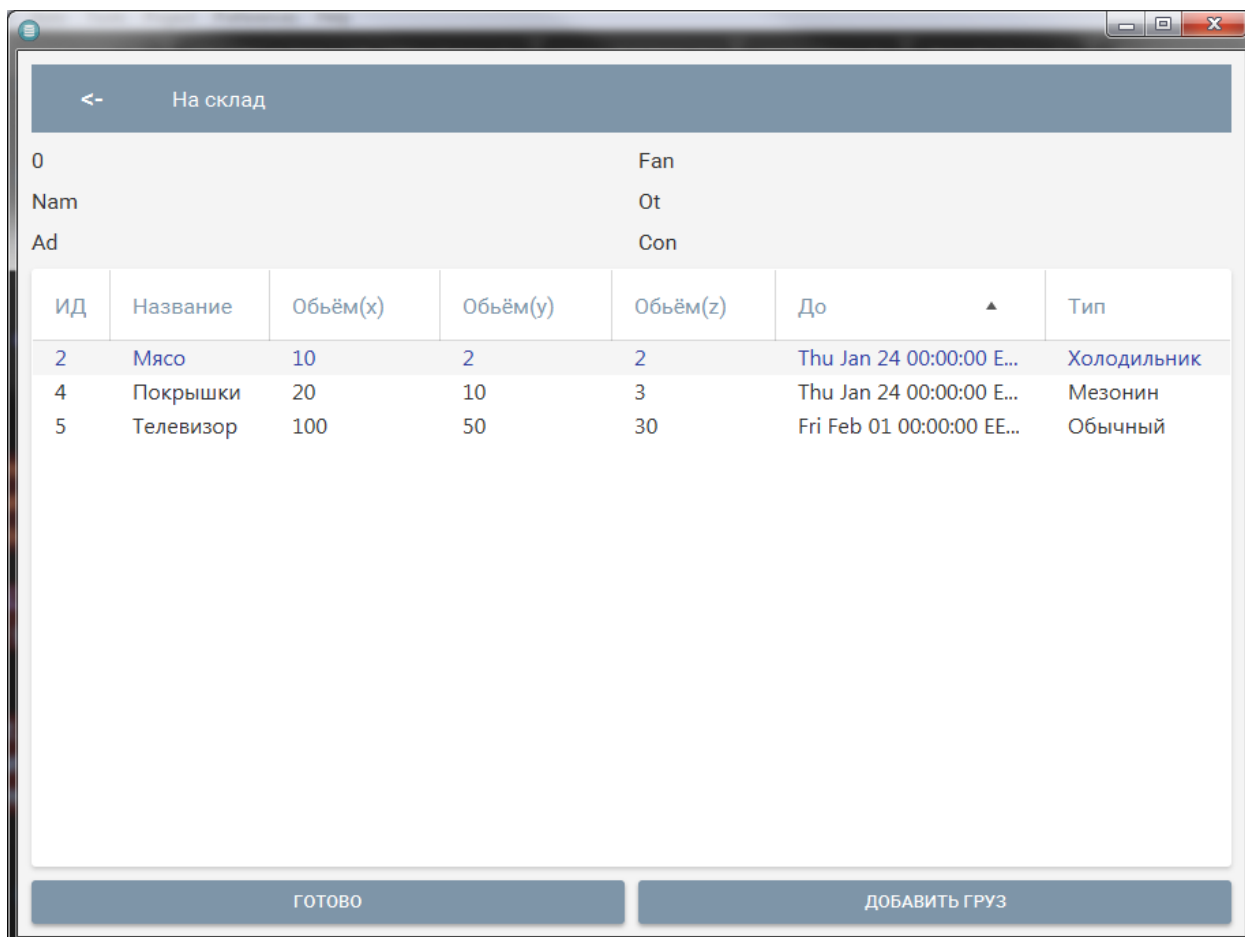


Рисунок 18 – Вікно «Зарахування Вантажу на склад»

Після цього буде сформована накладна (рис. 19) про зарахування вантажу на склад. До цього паперу необхідно додати таку інформацію, як:

- дата формування документу;
- інформація щодо відправника вантажу з контактною інформацією;
- характеристики вантажу, а також тип складського приміщення для зберігання.

Крім цього, з'явиться вікно схеми складу з маршрутом розвезення вантажу по складах для грузового транспорту компанії.

Номентклатура об отправке груза на склад
Номентклатура №38383972216350
Дата : Sun Apr 21 17:38:09 EEST 2019
Отправитель:
Фамилия : Абдулов
Имя : Александр
Отчество : Гаврилович
Адресс : Восменское озеро
Контактные данные : 09545654545
Груз №0
Название : Декорации
Объём(x) : 10
Объём(y) : 20
Объём(z) : 3
На складе до : Mon Apr 22 17:37:59 EEST 2019
Тип : Обычный
Склад : 0

Рисунок 19 – Приклад сформованої накладної для вантажу

3.5 Формування транспортної накладної

Після успішно завершеною транзакції по відправці вантажу на склад або його вилучення необхідно сформувати транспортні папери. Для цього були використані бібліотеки itextpdf, bcprov і pdfbox, які дозволяли формувати універсальний pdf-документ.

Отже, після транзакції весь вантаж був переданий в клас «Nomenclature». Приклад лістингу програми, що описує створення pdf-файлу розглянуто нижче. Для початку необхідно ввести файл збереження pdf-файлу:

```
FileChooser chooser = new FileChooser();
    chooser.setTitle("Сохранить номенклатуру");
    File save = chooser.showSaveDialog(null);
```

Потім створюється об'єкт документа, відкривається введення і ініціалізується потік записи в pdf-файл:

```
Document document = new Document();
PdfWriter.getInstance(document, new
FileOutputStream(save));
document.open();
```

Далі визначаються шрифти. І тут виникла проблема, так як шрифти itextpdf не підтримують кирилицю. Тому довелося прив'язувати нові зовнішні шрифти до цієї бібліотеці. Найпоширеніший шрифт, підтримуваний бібліотекою, називається Helvetica.

```
BaseFont thinBFont = BaseFont.createFont(
"jc/iln_app4/deixtra_db/font/helveticaThin /helveticaThin.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
```

```
BaseFont boldBFont = BaseFont.createFont(
"jc/iln_app4/deixtra_db/font /helveticaBold/ helveticaBold.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
```

```
Font captionFont = new Font(boldBFont);
captionFont.setSize(20);
```

```
Font paragraphFont = new Font(boldBFont);
paragraphFont.setSize(16);
```

```
Font font = new Font(thinBFont);
font.setSize(14);
```

Потім визначається вся потрібна інформація про транзакції, а саме

– ID;

- дата;
- ПІБ, адреса та контактні дані одержувача / відправника;
- назву, обсяг, дата збереження і тип вантажу.

```
Paragraph paragraph = new Paragraph(new Chunk("номенклатура
№" + c_id + "\n\n", paragraphFont));
document.add(paragraph);
```

В кінці необхідно закрити всі потоки виведення і створити файл накладної. Сформований документ можна зручно переглянути та роздрукувати практично на будь-який ОС (рис. 20), що є одним з головних вимог замовника проекту.

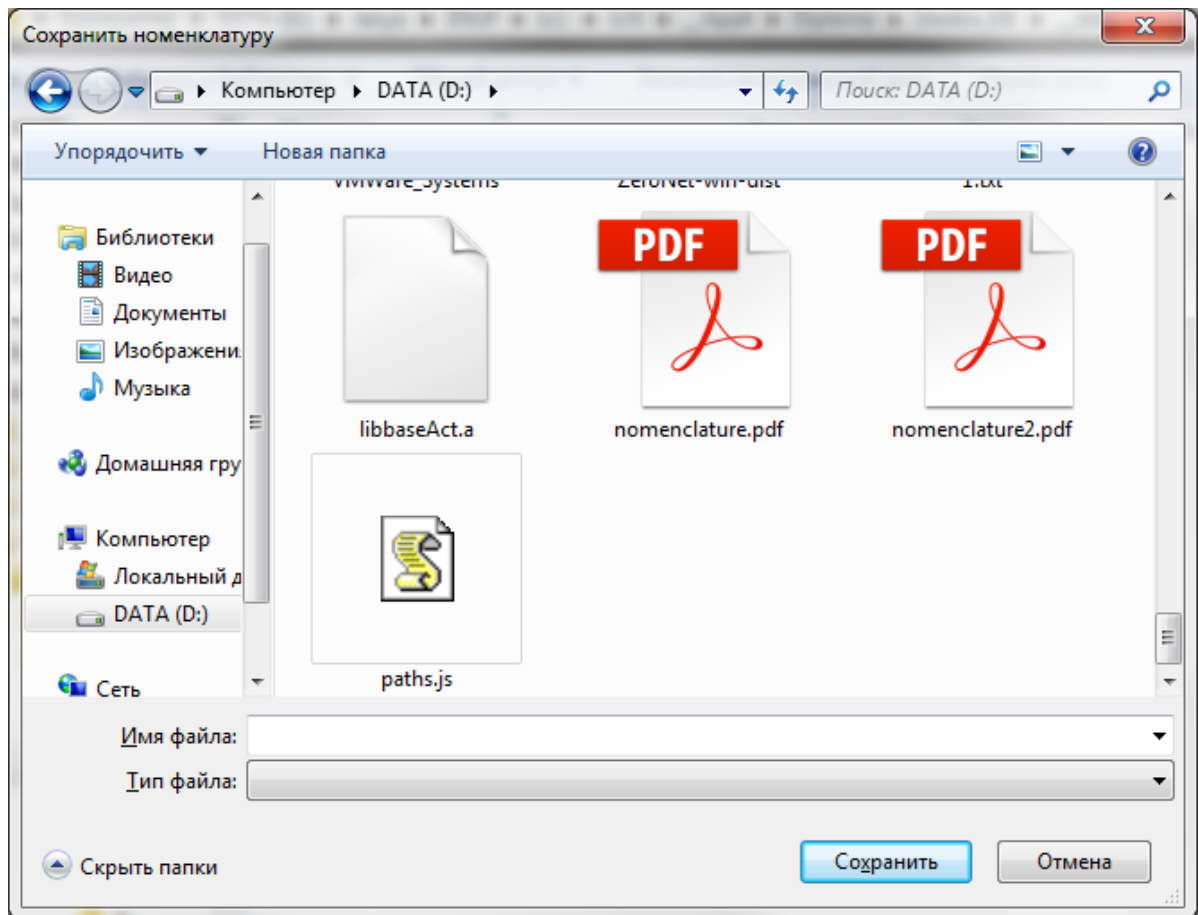


Рисунок 20 – Вікно компоненту FileChooser

Нижче предствалено приклад перегляду сформованного паперу – транспортної накладної за допомогою Acrobat Reader на рис. 21:

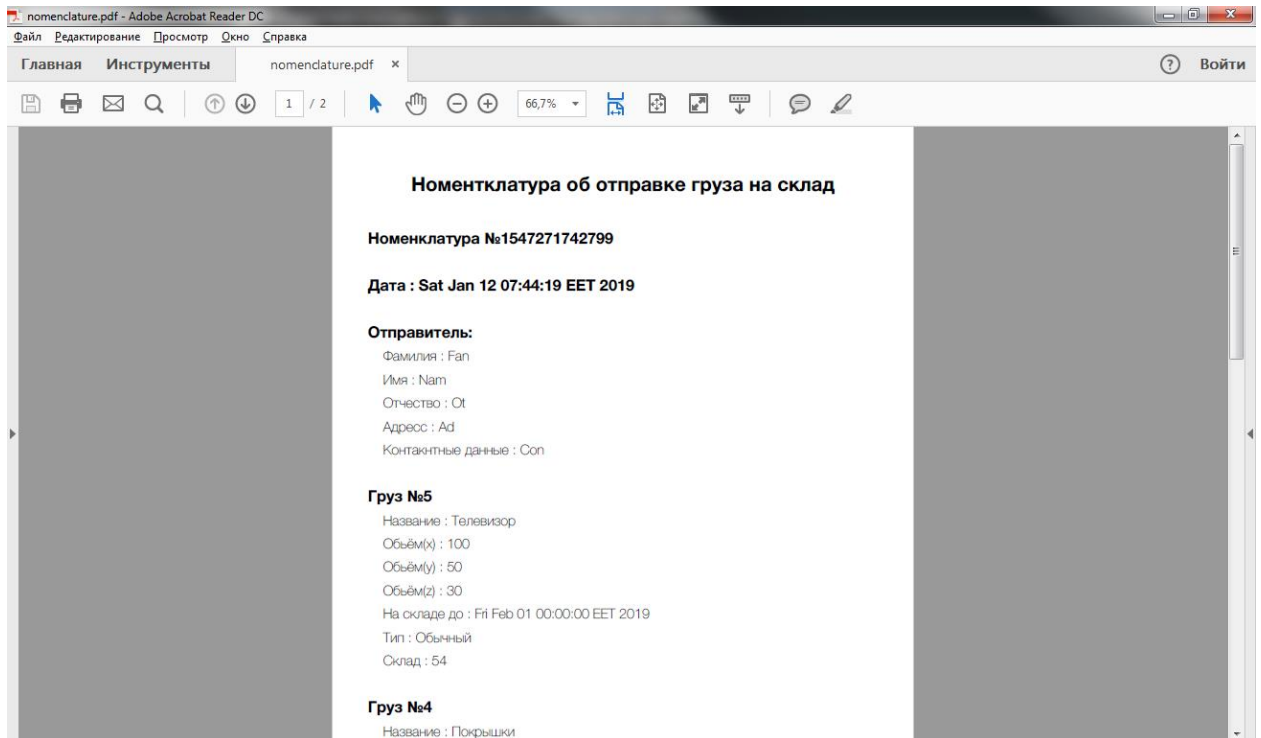


Рисунок 21 – Пример сгенерованної накладної

3.6 Відправка вантажу із складу

З інтерфейсом вікна по відвантаженню вантажу зі складу в порівнянні з зарахуванням істотних змін немає. Тут відсутня кнопка додавання вантажу, таблиця відразу відображає весь вантаж, який числиться на поточному людині.

При вилученні вантажу зі складу так само реагує Alert-повідомлення з інформацією про отримання вантажу зі складу, а саме, повня інформація щодо власника вантажу з його контактної інформацією, дата прибуття на склад і дата відвантаження.

Ці дані зберігаються у БД. Крім цього, формується такі документи, як транспортна накладна і лист-маршрут для навантажувача.

Приклад робочого вікна для переміщення вантажу з приміщення складу наведено на рис. 22.

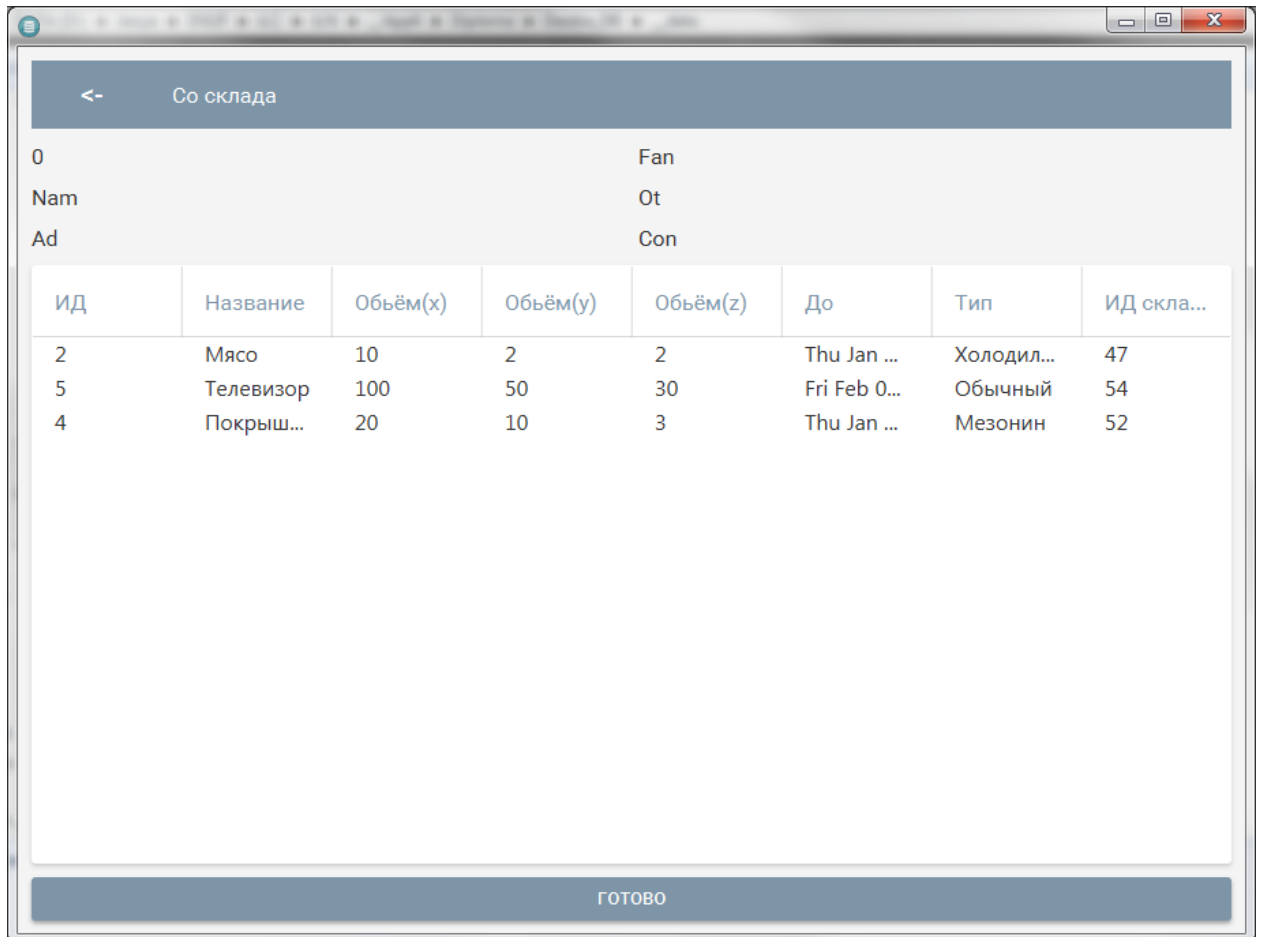


Рисунок 22 – Вікно для отримання вантажу зі складу

3.7 Планування маршрутом вантажівки

Склад займає велику територію з декількома корпусами и невеликими сховищами. Тому гостро виникла необхідність оптимізувати роботу вантажівок щодо переміщення між об'єктами з метою зкорочення часу переміщення і економії палива.

Для цього була створена схема складу (рис. 23). Для неї заздалегіть відзначено точки щодо містя розташування складів з урахуванням їх типу (точки різного кольору). Всі ці дані будуть зберігатись у «JS», а саме координати, тип і ID для кожного складу.

Далі клас «JS» повертає список складів, але в поданні «StorageVertex», який не має не потрібних полів, наприклад, обсяг, який для вирішення даного завдання не потрібен, але він має координати і зв'язку з іншими складами.

Потім «StoragePath», отримавши список складів з «JS» і список складів ззовні (це може бути або вікно зарахування на склад вантажу, або його вилучення), він їх схрещує, створюючи при цьому граф з елементів «CVertex», який грає роль ланки дороги (це може бути як склад, так і перехрестя або просто об'єднання доріг) і «CEdge», який являє собою зв'язок між «CVertex», до того ж він має поле відстань.

Після, за допомогою класу «DijkstraAlgorithm» обчислюється мінімальний шлях через всі задані склади. Кінцевим дією за обсягом вантажу обчислюється тип навантажувача.

Після всіх цих дій на вікні з'являється схема складу з розкресленим шляхом, по якому рухається відповідна модель навантажувача (рис. 24):

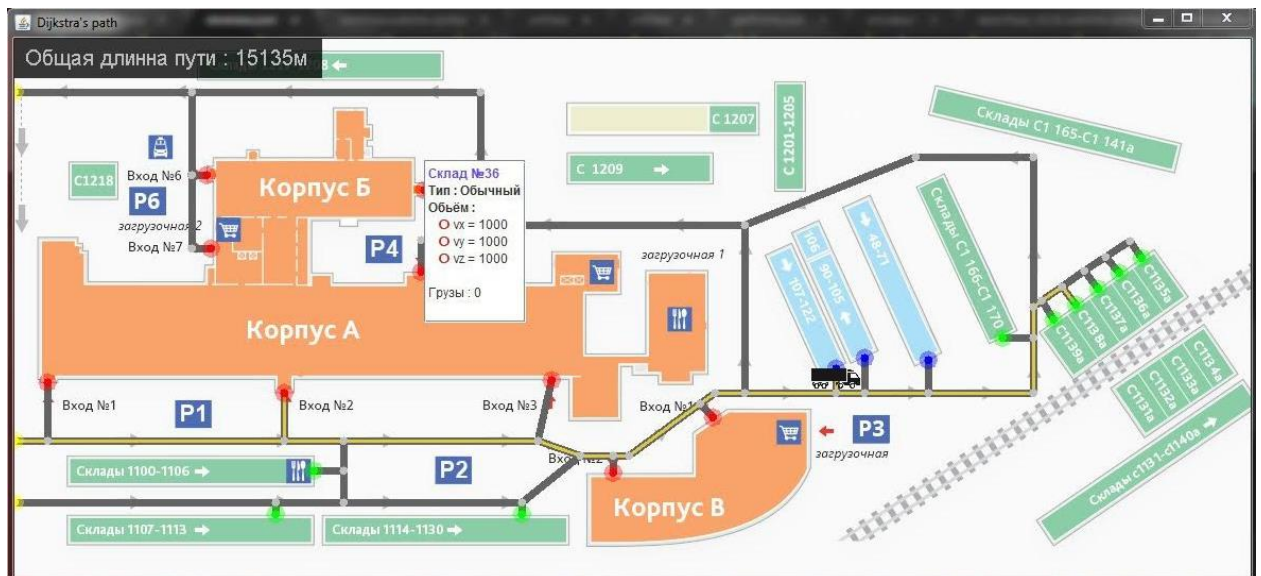


Рисунок 24 – Приклад маршруту переміщення вантажівки по території

Після завершення маршруту на карті можна побачити детальну інформацію щодо доставленого вантажу (рис. 25):



Рисунок 25 – Інформація доставки вантажу на склад

3.8 База даних підприємства

Для бази даних була обрана OracleDB так як вона тісно працює з Java і є найпотужнішою БД. Її архітектура дозволяє швидко працювати з великими масивами даних, в той час як, наприклад, MySQL в цьому програє.

Перш за все, для роботи з OracleDB необхідно створити користувача і дати йому дозволу на з'єднання і створення таблиць:

```
create user Deixtra identified by 0;
grant connect, resource to Deixtra;
```

Далі створюються самі таблиці для людини, складу і вантажу відповідно:

```
create table Person (o_id number (10) not null, surname
varchar (2000), name varchar (2000), secondName varchar (2000),
address varchar (1000), contact varchar (200), constraint
primaryKey primary key (o_id));
```

```
create table Storage (o_id number (10) not null, volume_x
number (10) not null, volume_y number (10) not null, volume_z
number (10) not null, costPerDay number (10, 2) not null, o_type
```

```
varchar (200 ) not null, constraint primaryKeyStorage primary
key (o_id));
```

І останнім етапом це створення кортежів складів. Ось приклад кількох з них:

```
insert into Storage (o_id, volume_x, volume_y, volume_z,
costPerDay, o_type)
values (56, 1000, 1000, 1000, 1.00, 'usual');
```

```
insert into Storage (o_id, volume_x, volume_y, volume_z,
costPerDay, o_type)
values (47, 700, 100, 100, 5.00, 'cold');
```

І в кінці, після всіх вище перерахованих дій здійснюємо збереження: `commit;`

На складі є присутнім три типи складу:

- звичайний;
- мезонін;
- холодильник.

Звичайний призначений для великого і громіздкого вантажу, мезонін для дрібних речей, а холодильник для вантажів, які вимагають низькі температури.

Запит на видалення вантажу з складу:

```
delete from Cargo where o_id =?
```

Запит на додавання вантажу на склад:

```
insert into Cargo (o_id, storage_id, person_id, name,
volume_x, volume_y, volume_z, o_dateFrom, o_dateTo, o_type)
values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
```

ВИСНОВКИ

Реалізація проектування інформаційної системи обліку руху товару на складі підприємства значно полегшить роботу його співробітників і забезпечить можливість зменшити витрати на управління за рахунок звільнення людських ресурсів, зайнятих різними видами обробки паперових документів, зберігати і аналізувати дані за будь-який проміжок часу, здійснювати пошук потрібної інформації по різними критеріями відбору.

Під час роботи над дипломним проектом було виконане:

- аналітичний огляд предметної області, а саме основні етапи роботи складу, маршрути проходження паперів і інше;
- виконан аналіз аналогів програмного забезпечення для об'єкту розробки, на основі якого поставлені вимоги і сформована мета до роботи;
- обрані програмні засоби для реалізації проекту;
- смодельовані основні етапи роботи складу за допомогою UML-засобів;
- розроблена структура БД складу;
- реалізован інтерфейс користувача для оптимального контролю і управління процесом роботи складу і формування транспортних паперів;
- додатково розроблен модуль для визначення оптимального маршруту для вантажівок по території підприємства для зменшення часових витрат, а також економії пального.

Подальшим розширенням функцій дипломного проекту є:

- можливість введення типу вантажівки і ціни на пальне з метою підрахування економії матеріальних коштів;
- розробка модулю для завантаження інших карт з метою автоматизації зчитування координат основних об'єктів;
- формування повного переліку документів крім накладних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Информационные системы в управлении складированием. URL: https://www.lobanov-logist.ru/library/all_articles/54584/. (дата звернення 12.02.2019).
2. Інформаційний довідник MAS System. URL: <https://mas-ukraine.com.ua/>. (дата звернення 12.02.2019).
3. Автоматизация бизнес-процессов. URL: <https://www.terrasoft.ua/crm/business-process-automation>. (дата звернення 13.02.2019).
4. Информационные системы в складской деятельности на примере WMS. URL: <https://novainfo.ru/article/5703>. (дата звернення 13.02.2019).
5. Информационное обслуживание склада. URL: <http://www.west-pereezd.ru/informacionnoe-obslujivanie-sklada.html>. (дата звернення 14.02.2019).
6. WMS система управления складом. URL: <http://www.netsoft.com.ua/WMS-Sistema-upravleniya-skladom.html>. (дата звернення 15.02.2019).
7. Обзор программ складского учета. URL: http://sklad-prog.ru/obzor/o_ind.htm. (дата звернення 15.02.2019).
8. ОК-СКЛАД 2.1.5.2310. URL: <https://www.softportal.com/software-10503-ok-sklad.html>. (дата звернення 14.02.2019).
9. Kataloger 2019.23. URL: <https://www.softportal.com/software-43426-kataloger.html>. (дата звернення 15.02.2019).
10. Триумф V2.0 3.63.117. URL: <https://www.softportal.com/software-45131-triumfv20.html>. (дата звернення 15.02.2019).
11. СУБД Oracle Database 11g. URL: <https://oracle-patches.com/oracle/prof/3139-%D1%81%D1%83%D0%B1%D0%B4-oracle-database>. (дата звернення 01.03.2019).

12. Обзор использования Oracle SQL Developer. URL: <http://sql-oracle.ru/obzor-ispolzovaniya-oracle-sql-developer.html>. (дата звернення 01.03.2019).
13. 1.4 Oracle SQL Developer. URL: http://www.askit.ru/custom/oracle10sql/m1/01_04_oracle_sql_developer.htm. (дата звернення 10.03.2019).
14. Руководство по JDBC v.1. URL: <http://www.javaportal.ru/java/tutorial/tutorialJDBC/intro.html>. (дата звернення 10.03.2019).
15. Руководство по JDBC. Введение.. URL: <https://proselyte.net/tutorials/jdbc/introduction/>. (дата звернення 10.03.2019).
16. Окошко с кнопками на JavaFX. URL: <https://habr.com/ru/post/420003/>. (дата звернення 20.03.2019).
17. Как работает Java машина (JVM). URL: <https://idurdyev.com/kak-rabotaet-jvm>. (дата звернення 22.03.2019).
18. Общая характеристика языка UML. URL: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2>. (дата звернення 23.03.2019).
19. Что такое Юзкейс (Use Case) или "Сценарий Ипользования" в Тестировании ПО? URL: <https://software-testing.org/testing/что-такое-yuzkeys-use-case-ili-scenariy-ispolzovaniya-v-testirovanii-po.html>. (дата звернення 27.03.2019).
20. Руководство по JavaFX. URL: <https://metanit.com/java/javafx>. (дата звернення 11.03.2019).