Зміст

Зміст 2

Перелік умовних позначень, символів, одиниць, термінів, скорочень 3

АНОТАЦІЯ 4

Вступ 5

ПОСТАНОВКА ЗАДАЧІ 6

2. АРХІТEКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ АСОД 7

2.1. DFD0-діаграма 9

2.2. DFD1-діаграма 10

2.3. DFD2-діаграма 12

2.4. Матриця подій 14

2.5. Бізнес правила 16

2.6. Інфологічне проектування 17

2.7. даталогічне проектування 18

ОСНОВНА ЧАСТИНА 20

Висновок 35

Список використаних джерел 36

Додатки 37

Перелік умовних позначень, символів, одиниць, термінів, скорочень

БД –База Даних  
АСОД – Автоматизована Система Обробки Даних

ІС – Інформаційна Система

СКБД – Система Керування Базою Даних

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, ТЕРМІНІВ

АНОТАЦІЯ

В цій роботі описана архітектура та процес розробки інформаційної системи «Залізнична пасажирська станція». Метою для створення такої системи була автоматизація роботи з потягами, товарами, продаж квитків, операції зі співробітниками. Були реалізовані основні потрібні функції, зокрема зі сторони персоналу - додавання нових працівників, редагування особистої інформації, відстеження медоглядів, додавання відпускних, виведення детальної інформації про співробітника. З матеріальної частини – продаж квитків, операції з товарами та контрагентами, зокрема додавання, редагування. Також формування руху до обраного контрагента.

Розроблена інформаційна система має зручний та зрозумілий інтерфейс користувача, що вміє працювати з комп’ютером. Доступ до бази можливий лише за наявності у користувача логіна та пароля(ролі). Кожна роль має різні права в області користування продукту.

Вступ

Розроблена в ході цієї роботи автоматизована система обробки даних має основні функціональні засоби, необхідні для роботи підприємств заданого типу. Програма побудована з двох сторін – персоналу та матеріальної частини. Клієнти кожної з частин мають змогу вносити дані, маніпулювати ними у бажаній формі та виводити. Інтерфейс реалізований задля забезпечення цілісності системи, полегшення процесу роботи кінцевого користувача, приховання процесів, що відбуваються в інформаційній системі. Також інтерфейс обробляє некоректні введені данні та упереджує їх потрапляння до інформаційної системи.

Вступ

# ПОСТАНОВКА ЗАДАЧІ

Автоматизована система обробки даних, що реалізована в даному проекті, призначена для використання як керівниками відділів залізничної станції так і працівниками.

Метою створення є полегшення роботи працівників та забезпечення цілісності даних підприємства. Система надає можливість обробляти дані про співробітників, квитки, товари, потяги.

Область функціонування створеного програмного забезпечення, що охоплює АСОД, вміщує:

* Прийняття на роботу нового працівника
* Зміну посади працівника
* Редагування особистої інформації
* Створення системи відділів та посад
* Можливість ведення записів про медогляд
* Можливість ведення документів
* Призначення індивідуального рафіку роботи
* Надання відпускних
* Звільнення з посади працівника
* Виведення детальної інформації про працівника
* Купівля та продаж білетів
* Бронювання та скасування бронювань
* Додавання потягів та вагонів
* Створення нових складів
* Створення нових документів про рух товару до бригад
* Переміщення товарів між внутріщніми контрагентами
* Переміщення товарів між зовіншніми контрагентами
* Перегляд та надання статусу кожній з операцій над квитками
* Перегляд інформації про рух документів
* Створення нових документів
* Додавання товарів
* Додавання та редагування контрагентів

Для коректності роботи програмного забезпечення накладаються відповідні вимоги то програмних і технічних засобів.

# 2. АРХІТEКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ АСОД

Розроблена в даному проекті АСОД передбачена для використання співробітниками залізничної станції, що працюють з персоналом або товарами.

АСОД вміщує в себе такий набір даних:

* Дані про клієнтів
* Дані про товари
* Дані про персонал
* Дані про білети
* Дані про медогляди
* Дані про відпустки
* Дані про документи
* Дані про графік роботи

Функції контролюючого персонал:

* Перегляд даних про всіх співробітників, їх документи, медогляди, години роботи, відпустки
* Перегляд звіту

Функції менеджера по персоналу:

* Функції контролюючого персонал
* Додавання нових відділів та посад
* Додавання нових працівників
* Додавання та редагування записів про медогляди, документи, особисті дані, години роботи, відпустки
* Звільнення співробітників
* Переведення співробітників на нову посаду

Функції касира:

* Продаж білетів
* Повернення білетів
* Бронювання білетів
* Скасування бронювання білетів
* Перегляд детальної інформації з статусами кожної операції з білетами
* Перевірка доступності білетів

Функції менеджера товарів:

* Додавання замовлень
* Додавання нових товарів
* Додавання нових елементів до документу
* Додавання нового документа про зовнішній рух товару
* Додавання нового документа про внутрішній рух товару

Функції менеджера контрагентів:

* Робота з фізичними особами
* Зміна фізичних осіб
* Зміна Юридичних осіб
* Перегляд інформації про бригади
* Перегляд детальної інформації про кожного контрагента
* Робота з юридичними особами

# 2.1. DFD0-діаграма

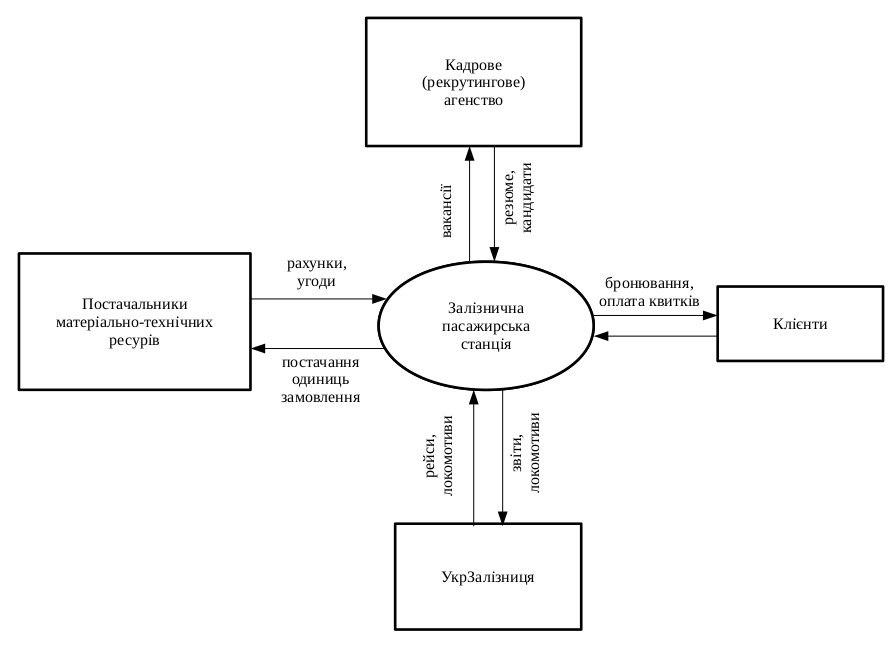


Рис.1 DFD0 - діаграма

# 2.2. DFD1-діаграма

На рис. 2 та рис. 3 зображені діаграми потоків даних першого рівня для підсистем «Відділ кадрів» та «Рух товарів» відповідно.

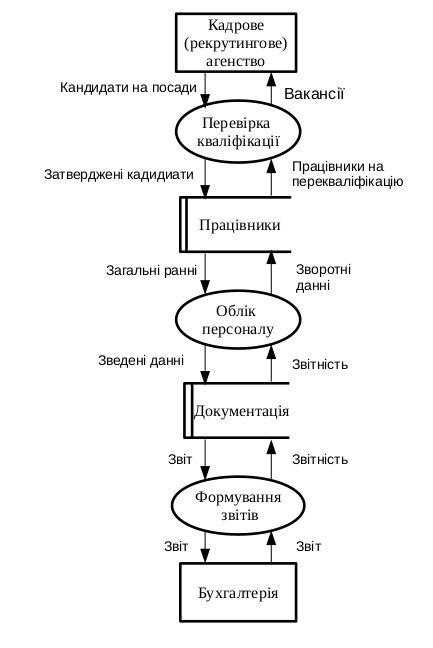


Рис.2 DFD1 – діаграма. «Відділ кадрів»

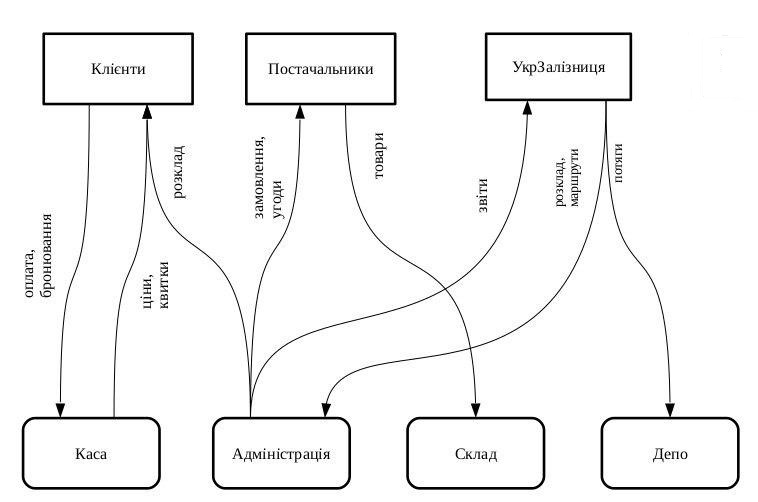


Рис.2 DFD1 – діаграма. «Рух товарів»

# 2.3. DFD2-діаграма

На рис. 4 та рис. 5 зображені діаграми потоків даних другого рівня для підсистем «Робота з клієнтами» та «Рух товарів» відповідно.

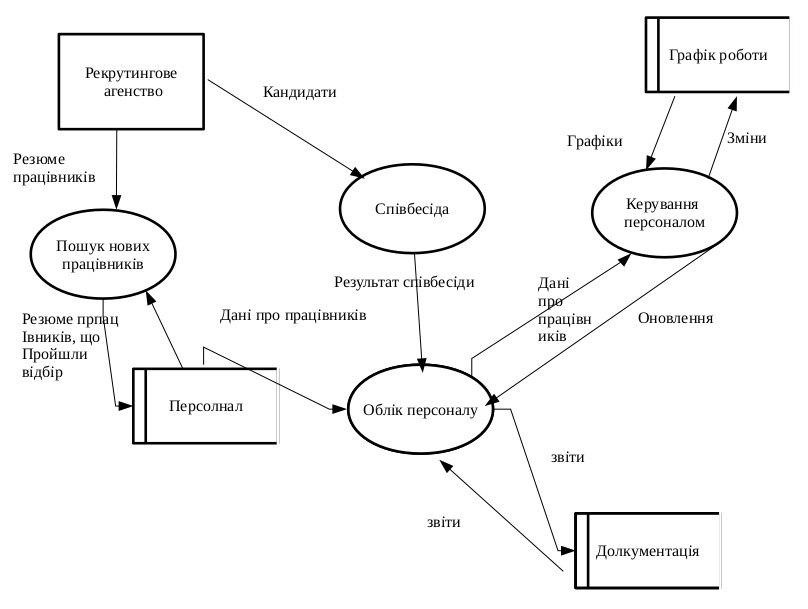


Рис.4 DFD2 – діаграма. «Відділ кадрів»

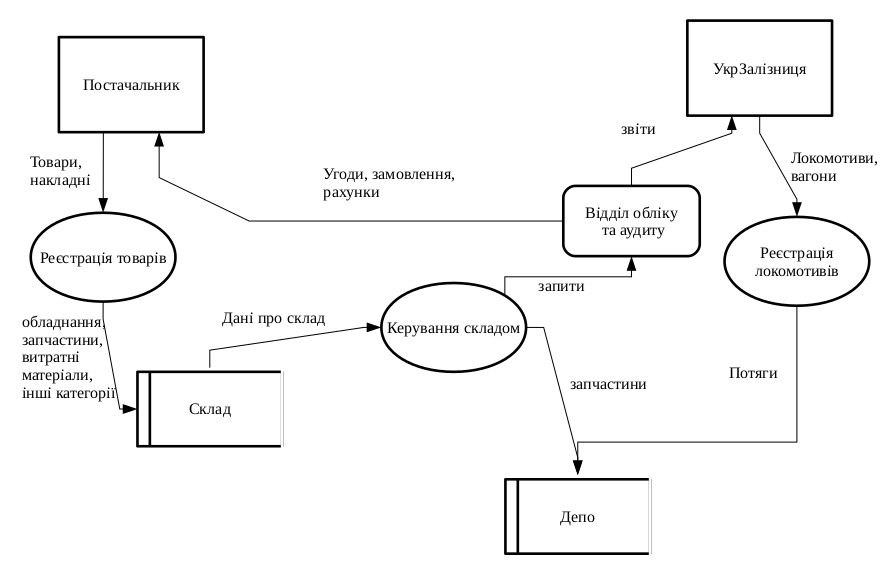


Рис.5 DFD2 – діаграма. «Рух товарів»

# 2.4. Матриця подій

Матриця подій двох підсистем відображають події, що можуть статися та реакцію системи на них

|  |  |  |  |
| --- | --- | --- | --- |
| № | Опис події | Тип | Реакція |
| 1. | Прийом на роботу | N | Якщо кандидат відповідає умовам прийому на роботу, то інформація вноситься до БД «Персонал» та «Працівники» |
| 2. | Звільнення | N | Якщо працівник потребує звільнення, то у БД «Працівники» вноситься інформація про дату закінчення роботи. |
| 3. | Проходження медогляду | N | По проходженні медогляду до БД «Медогляд» додається запис за результатом та датою проходження. |
| 4. | Додавання відділу | N | Відповідний запис заноситься в БД «Відділи» |
| 5. | Додавання посади | N | Відповідний запис заноситься в БД «Посади» |
| 6. | Зміна посади | N | Оновлення попереднього запису в БД «Персонал» та додавання нового засобу |
| 7. | Призначення відпустки | N | Додавання відповідного запису в БД «Відпускні» |
| 8. | Виведення звіту | N | Формується звіт з БД та друкується у форматі PDF. |
| 9. | Зміна графіку роботи | N | Відповідний запис додається до БД «Графік роботи» |
| 10. | Додавання документу | N | Запис про дату та опис документу додається до БД «Документи» |
| 11. | Спроба логіну без пароля | NN | Виведення повідомлення про помилку |

Табл.1 Матриця подій для підсистеми «Персонал»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Опис події | Тип | Реакція |
| 1. | Реєстрація товару | N | Занесення даних про товар в БД |
| 2. | Реєстрація контрагента | N | Занесення даних про контрагента в БД |
| 3. | Редагування даних про товар | N | Якщо товар нявний, то відповідні зміни вносяться в БД. |
| 4. | Продаж квитків | N | Відповідний запис додається до БД |

Табл.2 Матриця подій для підсистеми «Рух товарів»

# 2.5. Бізнес правила

Бізнес правила підприємства:

1. Працівниками залізничної станції можуть бути лише особи, що досягли повноліття, тобто 18 років.
2. Кожен робітник має щорічно проходити медогляд.
3. Не можна в межах одного відділу створювати посади з однаковим іменем.
4. При зміні посади працівник звільнюється у день отримання нової посади.
5. Працівники мають право на 30 денну відпустку протягом року.
6. Не можна створювати товарів з однаковими іменами
7. Після того, як потяг пішов не можна продавати та бронювати квитки на нього
8. Повернення квитків після відправлення потяга можливе

# 2.6. Інфологічне проектування

Із визначеної предметної області було виділено наступний набір основних сутностей:

1. Контрагент
2. Документ
3. Фізична особа
4. Юридична особа
5. Товари
6. Медогляд
7. Відпустка
8. Відділ
9. Посада
10. Квиток
11. Графік роботи
12. Особисті дані співробітника
13. Потяг
14. Квиток
15. Об’єкт зберігання
16. Бригада
17. Склад
18. Рейс
19. Список товарів
20. Документ

# 2.7. даталогічне проектування

Даталогічна модель представлена на рис. 6 та рис. 7 у вигляді фізичної ER-діаграми для підсистеми «Персонал» та «Рух товарів» відповідно

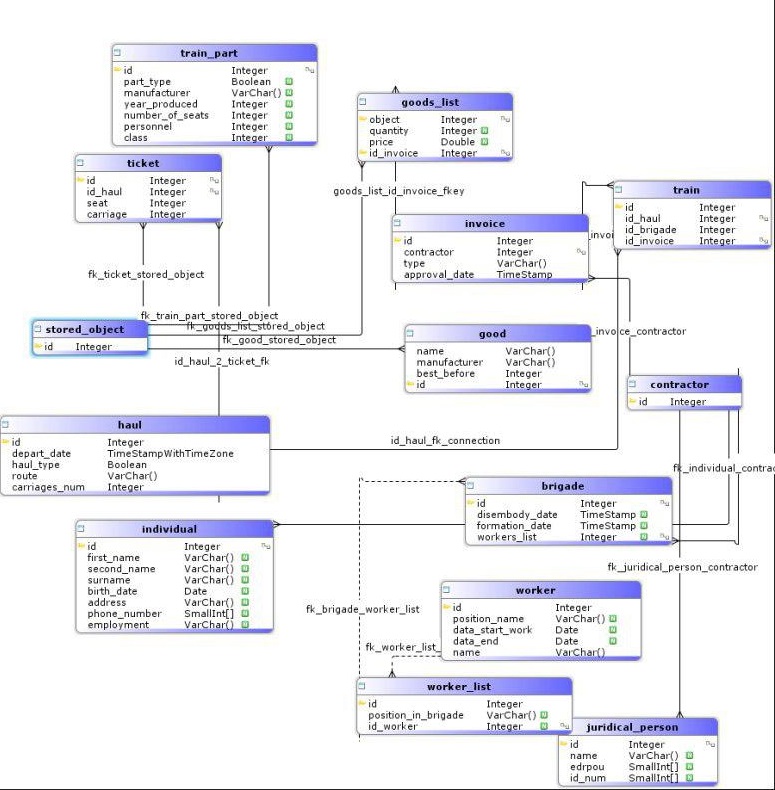


Рис. 6. Фізична ER-діаграма підсистеми «Рух товарів»

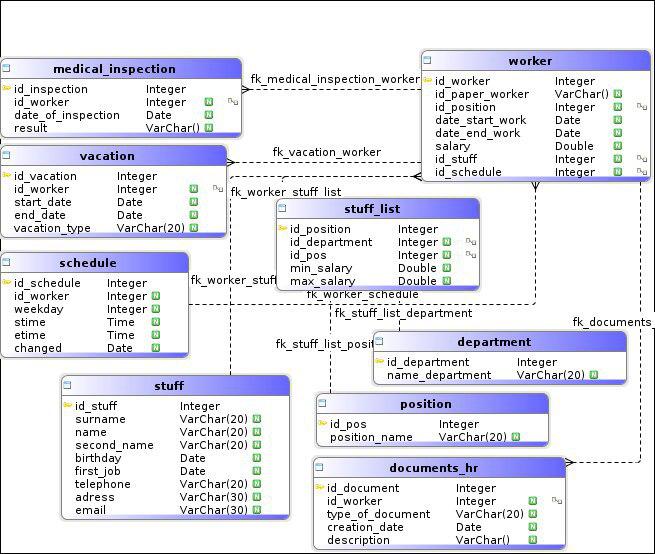
****

Рис. 7. Фізична ER-діаграма підсистеми «Персонал»

# Опис програми

**ІНТЕРФЕЙС ВІДДІЛУ КАДРІВ**

При запуску програми відкривається вікно авторизації (рис. 3.1). У ньому необхідно ввести ім’я користувача і пароль. При спробі зайти до бази без їх введення, чи при невірному введенні, буде виведене повідомлення про помилку (рис. 3.2)

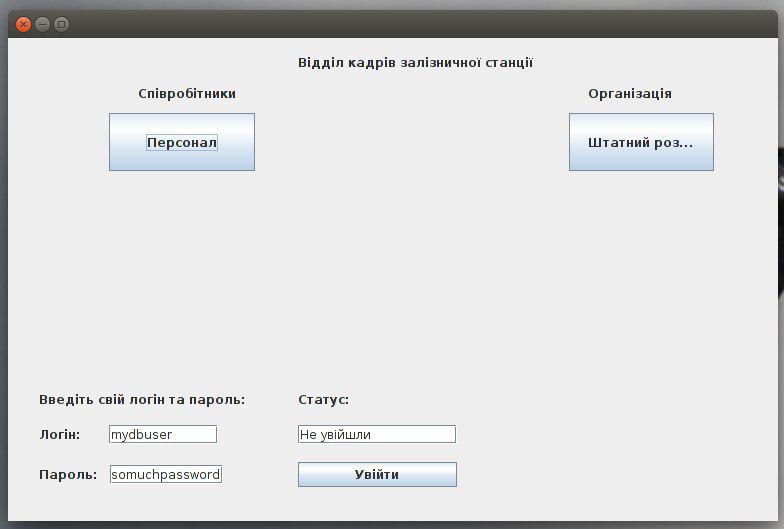


Рис.3.1 Вікно авторизації

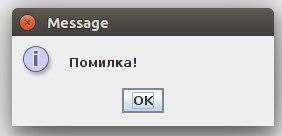


Рис.3.2 Повідомлення про помилку

При успішній авторизації, користувач зможе зайти до обраної частини інтерфейсу – «Персонал» (рис. 3.3)- використовується для перегляду, додавання та зміни інформації про працівників або «Штатний розклад»(рис. 3.4) - використовується для для перегляду, додавання та зміни інформації про відділи та посади. Надалі будемо розглядати роботу інтерфейса з ролі менеджера персоналу. Роль наглядача за персоналом буде відрізнятися відсутністю права на редагування, внесення або видалення будь якої інформації. Кожне вікно інтерфейсу має кнопку «Назад» для повернення до попереднього вікна. Варто зазначити, що при таких переходах зберігається цілісність системи – до бази не буде додано неповного запису.

У інтерфейсі «Штатний розклад» користувач має можливість додати новий відділ або посаду у існуючий відділ(рис. 3.5). в основному вікні інтерфейсу «Штатний розклад» також передбачена можливість оновити данні таблиці, що складається з полів: відділ, посада, мінімальна зарплатня, максимальна зарплатня.

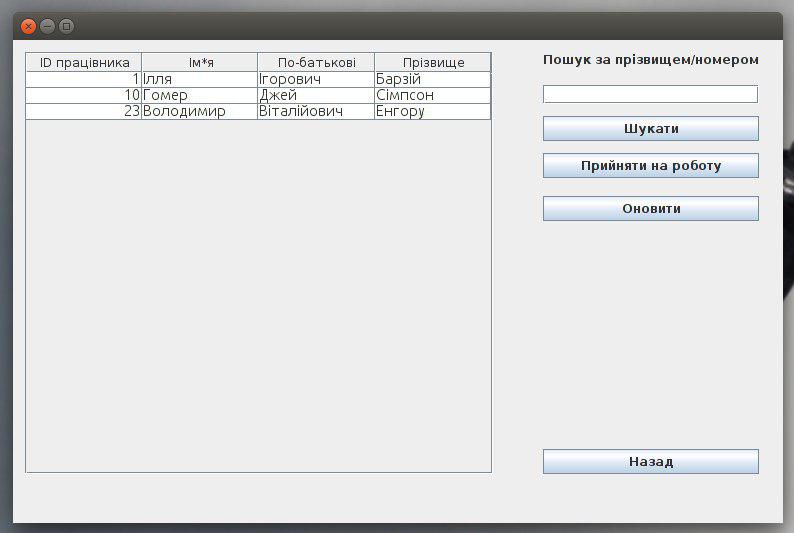


Рис.3.3 Інтерфейс частини «Персонал»

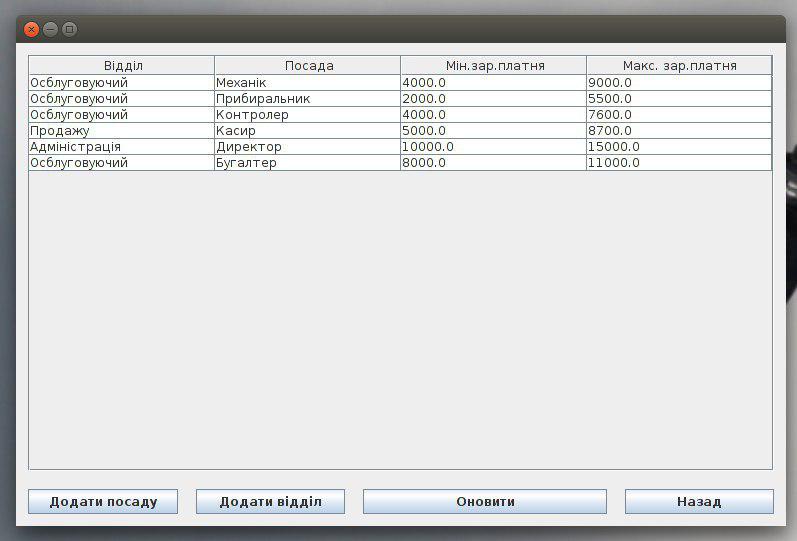


Рис.3.4 Інтерфейс частини «Штатний розклад»

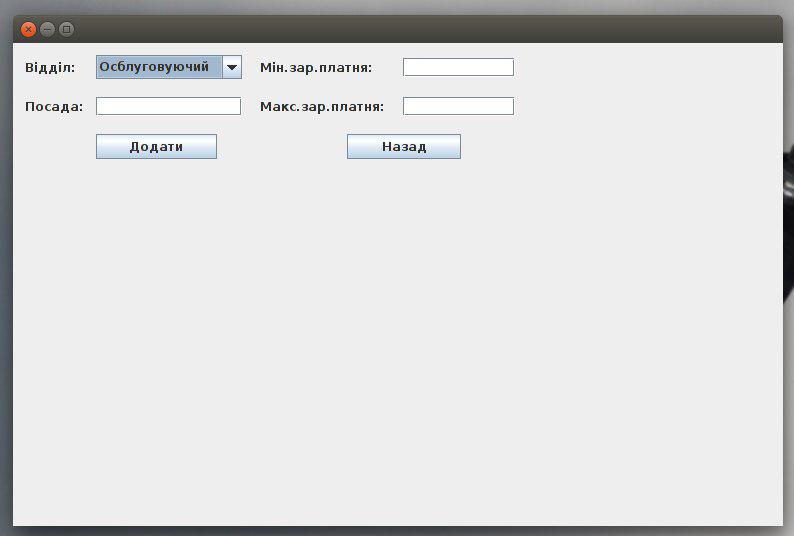


Рис.3.5 Вікно додавання нової посади

Більша частина функцій міститься в інтерфейсі «Персонал». Зліва таблиця, що вміщує всіх робітників підприємства. Реалізований пошук за номером працівника або за прізвищем/ім\*ям/по батькові а також оновлення списку.

При необхідності додати нового співробітника, варто натиснути відповідну кнопку. Користувач перейде до послідовності вікон для введення даних про співробітника(рис. 3.5, рис. 3.7, рис. 3.8). При виході з даної послідовності вікон, нова інформація не буде додана до бази. При введенні всіх полів вірно, запис додається до бази та відображається у таблиці інтерфейсу «Персонал», до я кого повертається користувач.

Можливо також не вводити частину особистих даних у разі, коли робітник вже працював на цьому підприємстві. Можливо присвоєння працівнику особистого номеру, що є унікальним, але не є порядковим.

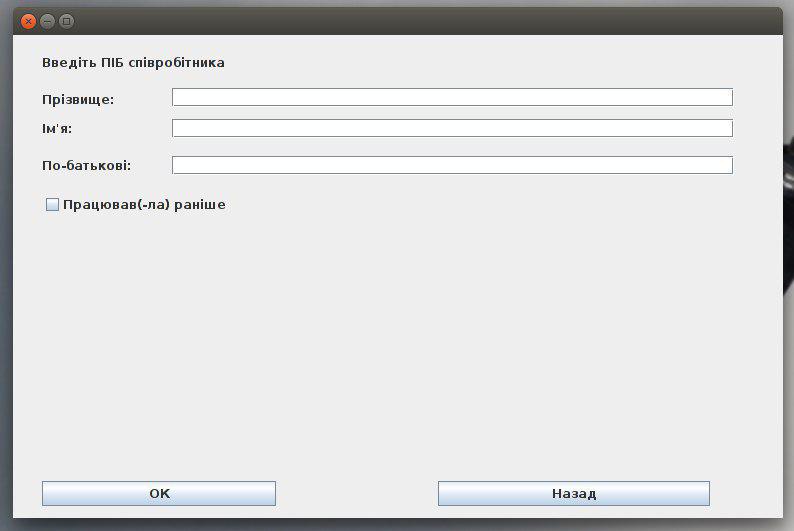


Рис.3.6 Перше вікно додавання працівника

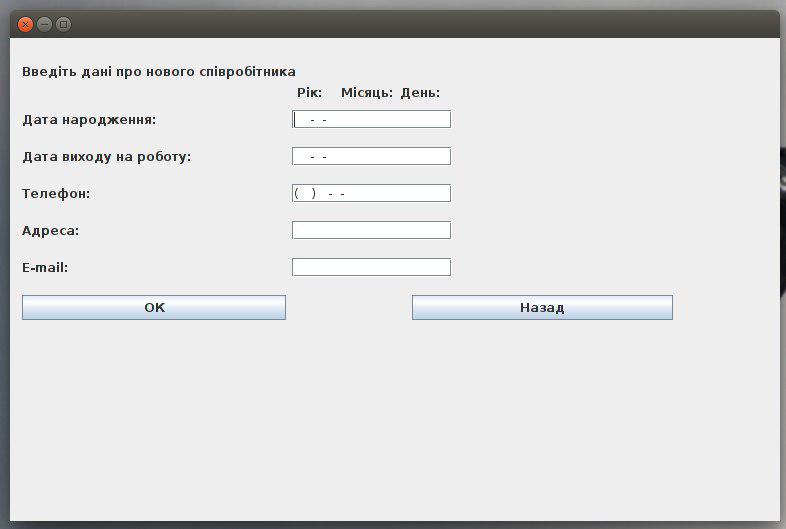


Рис.3.7 Друге вікно додавання нової посади – особисті дані

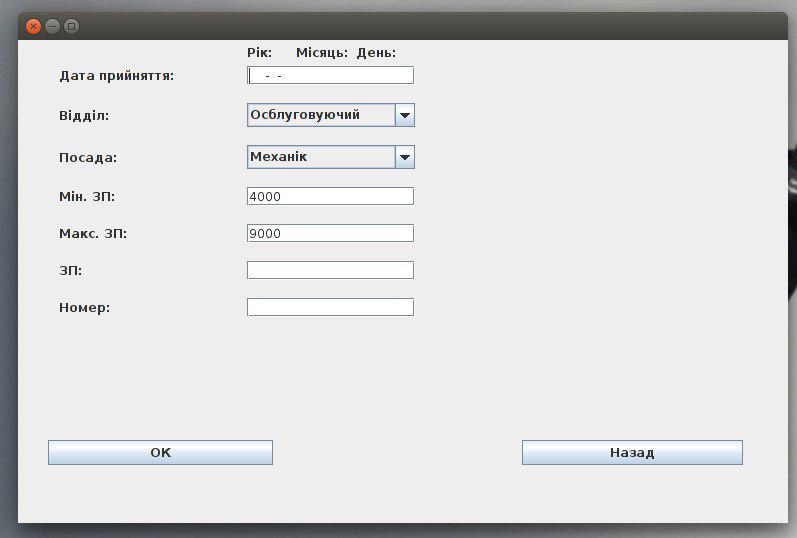


Рис.3.8 Трете вікно додавання нової посади – робочі дані

При натисканні на рядок з таблиці відкривається детальна інформація про працівника(рис 3.9). З цього вікна можна редагувати особисті дані, звільнити працівника, змінити його посаду, додати запис про медогляд(рис 3.10), документ, призначити відпустку, змінити графік роботи а також вивести детальну інформацію про працівника у файл PDF. Деякі поля недоступні для редагування з міркувань цілісності бази даних.

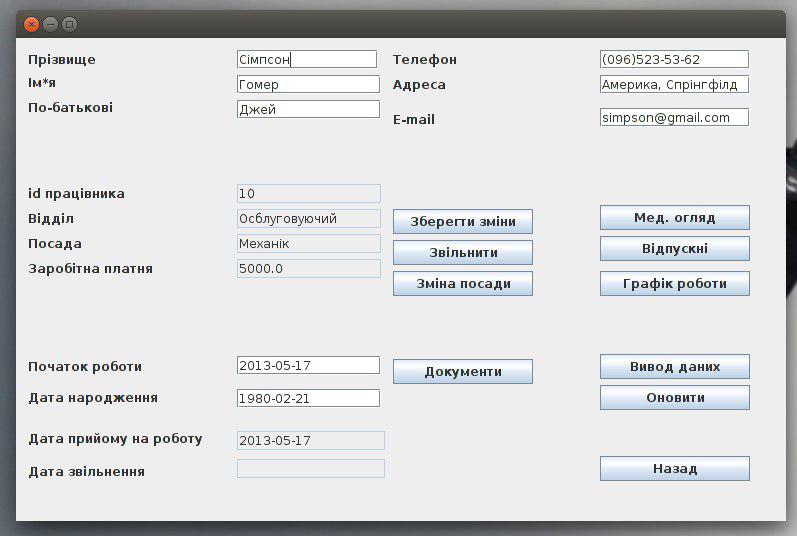


Рис.3.9 Детальні дані про працівника

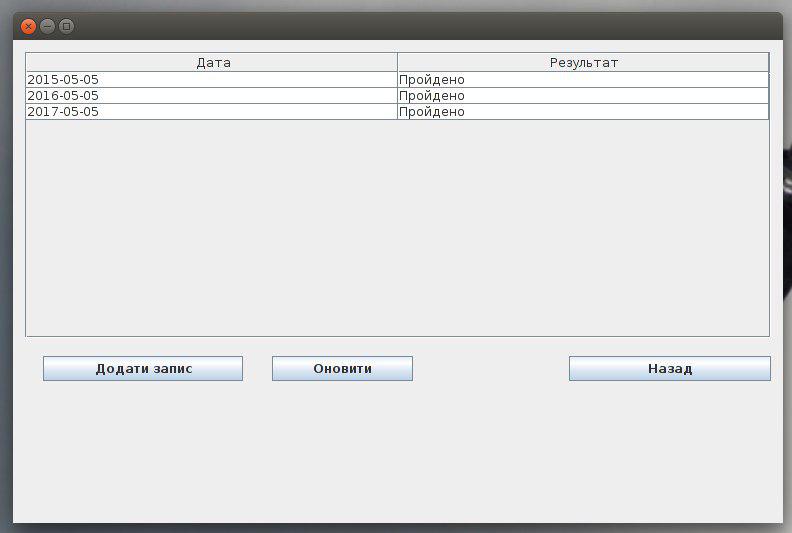
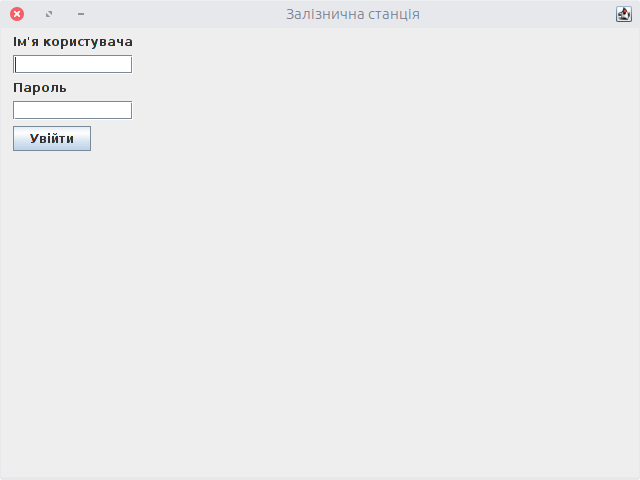


Рис.3.10 Вікно медоглядів працівника

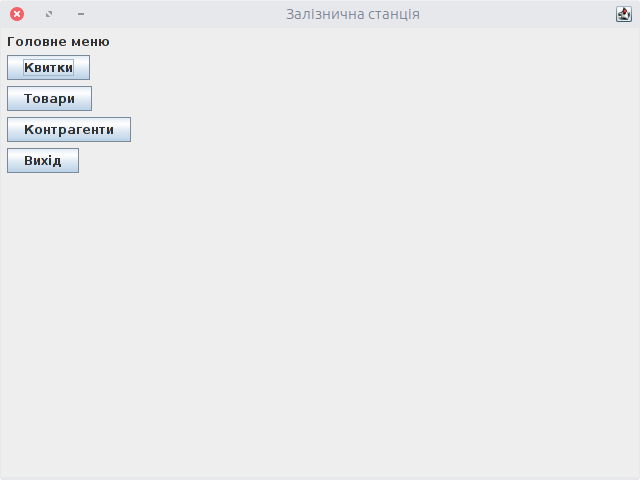
**ІНТЕРФЕЙС ВІДДІЛУ РУХУ ТОВАРІВ**

Розглянемо декілька прикладів, що допоможуть зрозуміти, як працює та як можна використати результати роботи програми.

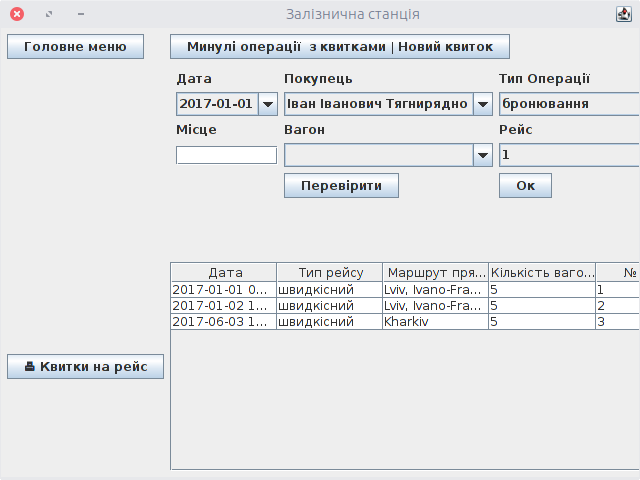
## Частина руху товарів



Перший екран входу. Для того, щоб увійти як певний користувач необхідно ввести правильні пароль та логін. Зверніть увагу, що вони відрізняються від тих ,що вказані для самої СКБД



Головне меню. У цьому меню відбувається вибір підрозділу, з яким буде працювати оператор. У зображеному випадку моливо вибрати усі три варіанти через те, що використовується режим адміністратора.

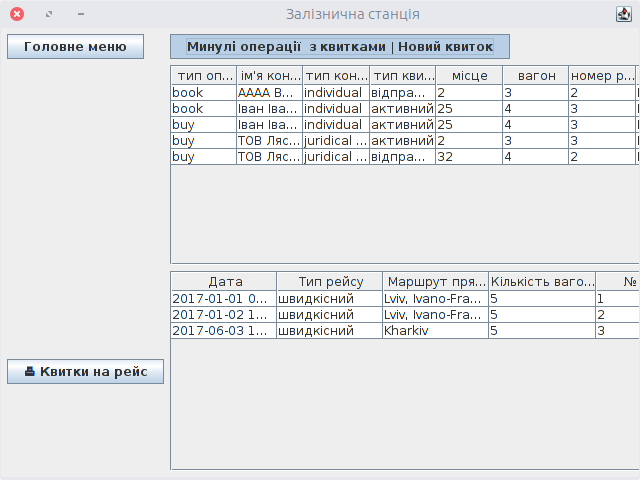


Режим роботи з квитками. Розглянемо спочатку нижню панель. Демонструється список рейсів та необхідна інформація про них. У випадку збільшення кількості рейсів з’явиться повзунок, що довзолить переміщуватись по таблиці.

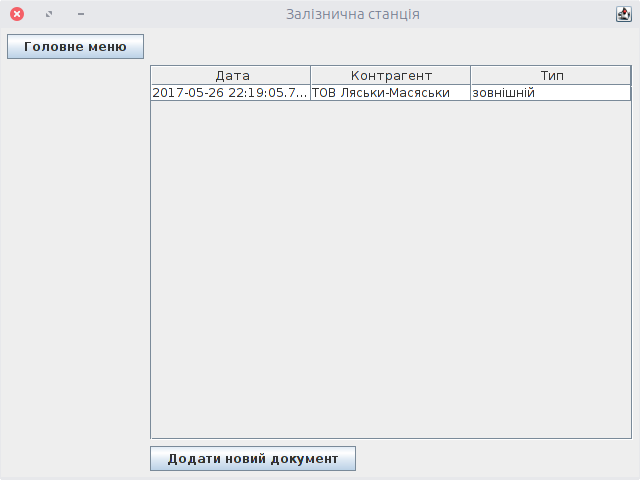
У лівому нижньому кутку знаходиться кнопка друку звіту про продані квитки на певний рейс для подальшого використання на станції.

Верхня половина містить частину інтерфейсу, яка відповідає за опрацювання квитків. Підписи до кожного опля вказані над ним. Якщо усі поля заповнені, то можна виконати або перевірку операції, або спробувати виконати її.

Найвищий – перемикач на минулі операції з квитками, що класифіковані.

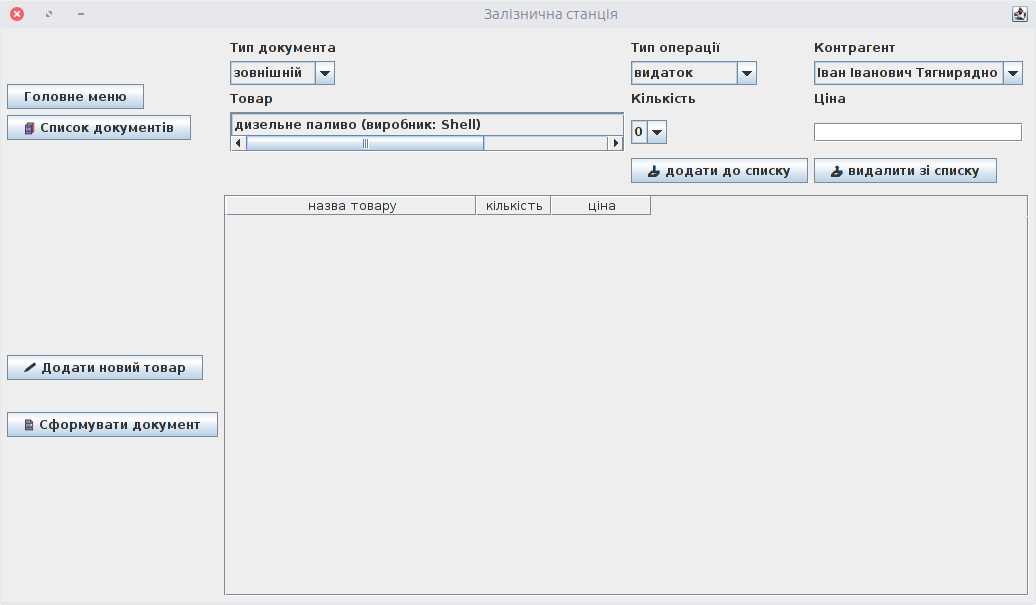


У цьому режимі користувач переглядає усі попередні операції з квитками та може розрізняти їх за активними та пасивними, тобто такими, що можуть бути використані ще раз в операції



Перший екран роботи з документами. Тут зображена таблиця усіх документів, що наявні в базі даних, їх тип, контрагент та дата.

За кнопкою знизу переходимо до додавання нового документа



Форма додавання нового документа. Панель угорі дозволяє додавати довільний товар до загального списку товарів, що будуть направлені на документ.

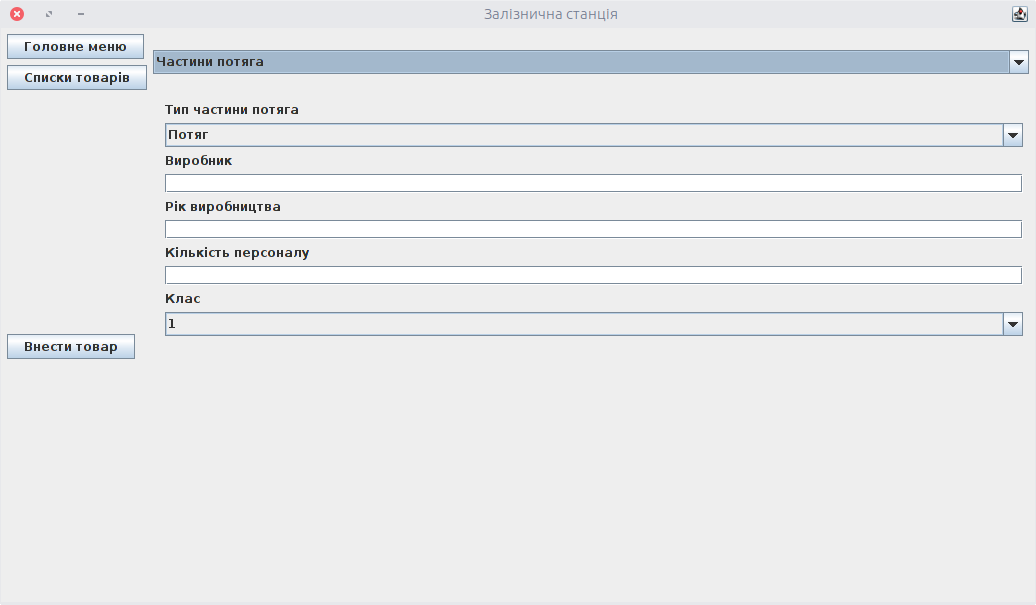
Рух товарів поділяється на внутрішній та зовнішній. Кожен окремий рядок списку є видатком чи надходженням.

Контрагент спільний для усього списку, так само, як і тип документа руху товару.

Кількість на видаток не перевищує тієї, що є на складі за документами. Кількість на надходження не обмежена.

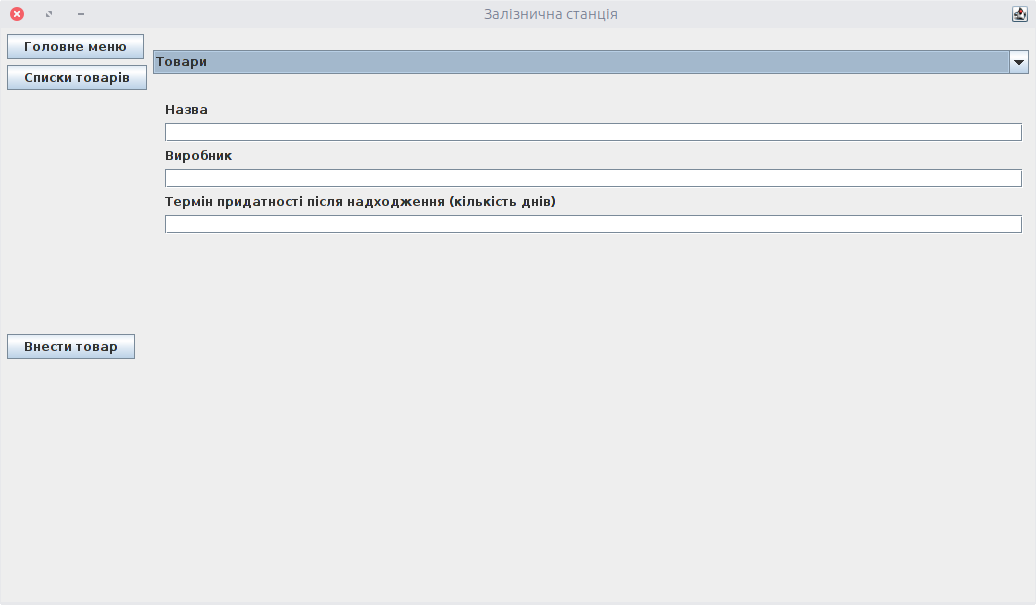
Якщо товару немає серед бажаних, то його можна додати натиснувши «додати новий товар»

Якщо після додавання товарів до списку чек готовий, то його можна зберегти, натиснувши «сформувати документ»

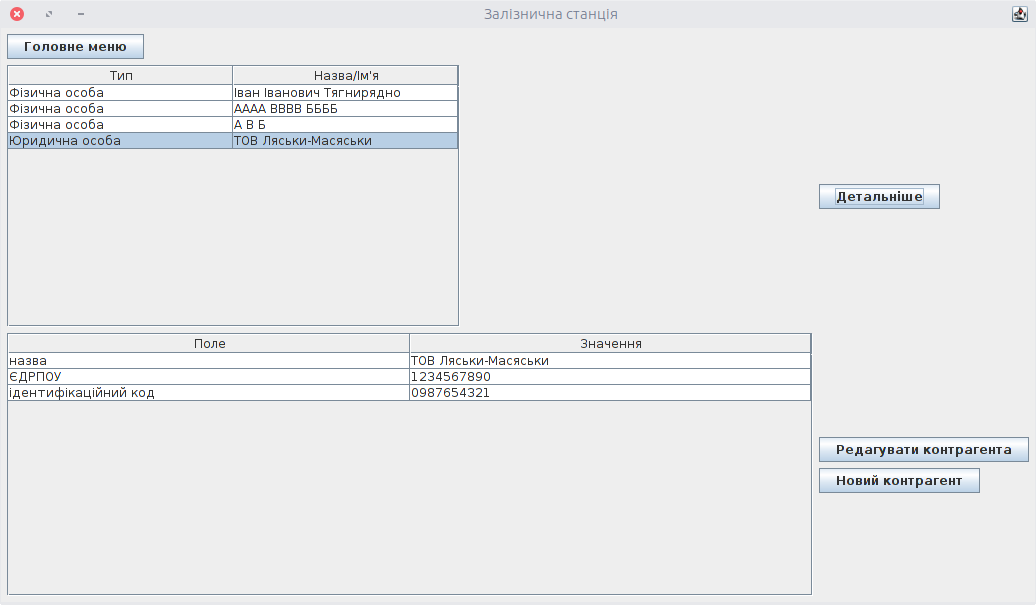


Додавання нової частини потяга до бази даних. Вказавши усі необіхдні форми, користувач натискає «внести товар» і він потряпляє до загальних списків.

Кожна частина потяга має необхдіну кількість персоналу та клас.

Виробник виступає специфікатором назви, бо не є контрагентом і жодних функцій не виконує.

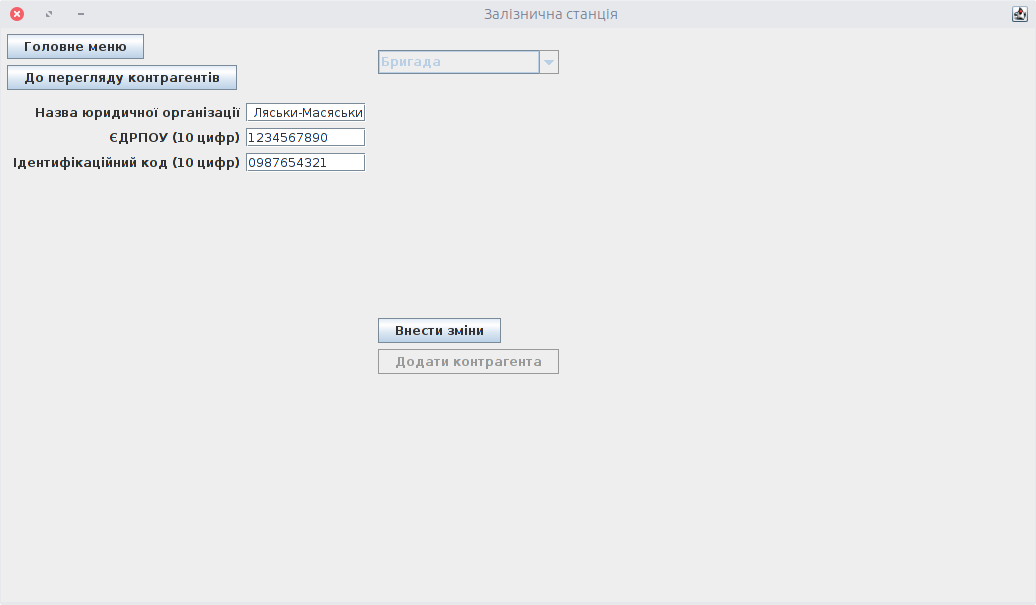
Додавання Товару відбувається за подібною схемою, тільки з меншою кількістю деталей.



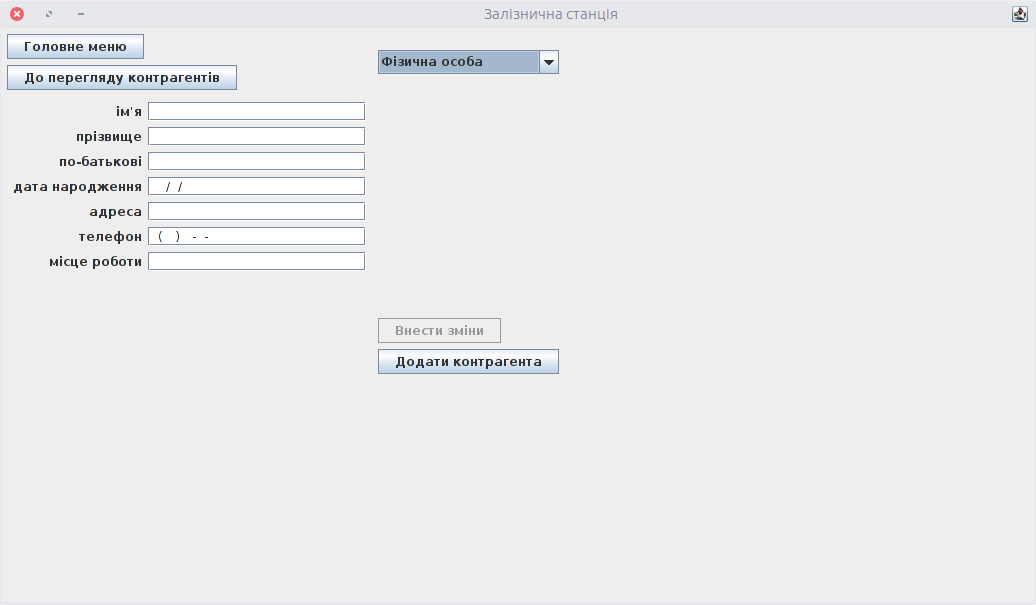
Робота з контрагентами. На цій панелі користувач має можливість переглянути загальний список контрагентів з іменами. На нижній панелі знаходиться список деталних властивостей контрагента, що обирається та за кнопкою «Детальніше» демонструються.

Обраний контрагент може бути відредагований за натисканням кнопки «Редагувати контрагента».

Оператор має можливість додати нового контрагента за натисканням на кнопку «додати нового контрагента»

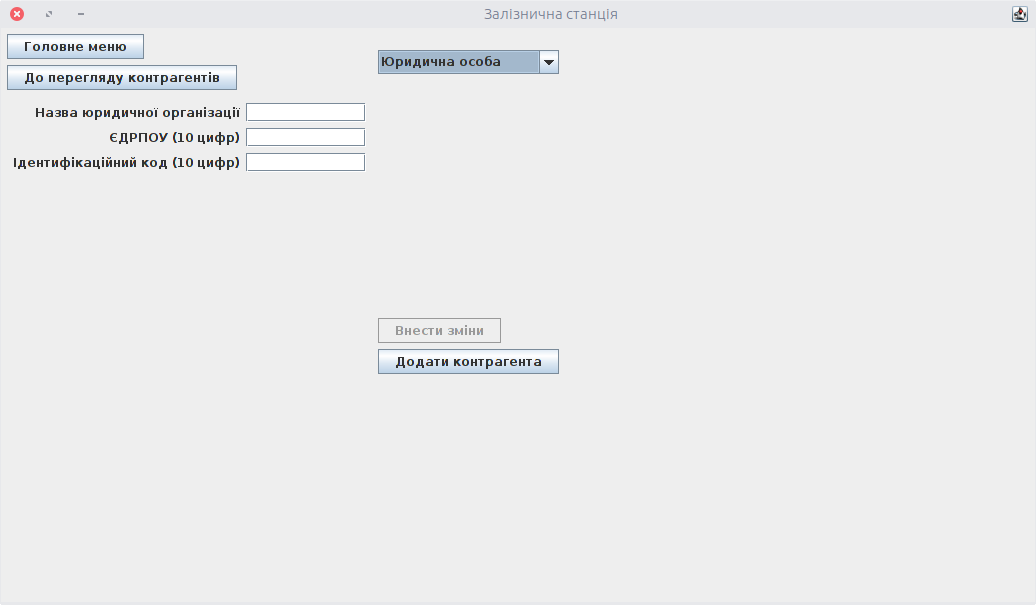


Приклад редагування контрагента



Додавання нового фізичної особи – контрагента

Усі поля заповнюються відпвідно до назв

А потім користувач натискає на кнопку «Додати контрагента»

Додавання нової юридичної особи-контрагента

Аналогічно з випадком фізчиної особи всі поля заповнюються

# Висновок

Отже, при виконанні даного курсового проекту була обміркована, спроектована та створена база даних засобами PostgerSQL та розроблений користувацький інтерфейс засобами NetBeans та Eclipse мовою програмування Java. Реалізований продукт забезпечує виконання основних функцій системи «Залізнична пасажирська станція».

Враховані можливості роботи з різними правами доступу (ролями). Серед них:

Функції контролюючого персонал:

* Перегляд даних про всіх співробітників, їх документи, медогляди, години роботи, відпустки
* Перегляд звіту

Функції менеджера по персоналу:

* Функції контролюючого персонал
* Додавання нових відділів та посад
* Додавання нових працівників
* Додавання та редагування записів про медогляди, документи, особисті дані, години роботи, відпустки
* Звільнення співробітників
* Переведення співробітників на нову посаду

Функції інспектора товару:

* Перегляд списків товарів, контрагентів, замовлень, потягів.

Функції касира:

* Продаж білетів

Функції менеджера товарів:

* Додавання та редагування контрагентів
* Додавання замовлень

В ході проектування Бази Даних була передбачена обробка виключних ситуацій. Інтерфейс програми є зручним та зрозумілим.

# Список використаних джерел

1. Конспект лекцій з курсу «Бази даних»

2. Дейт.К. «Введение в системі баз данных»

3. [Інтернет ресурс]: stackoverflow.com

4. [Інтернет ресурс]: postgresql.ru/manual/

5. [Інтернет ресурс]: <http://www.javaknowledge.info>

# Додаток

**Додаток1**. Процедура та функції для відділу кадрів

Функція підрахунку часу роботи співробітника на посаді по годинах:

CREATE OR REPLACE FUNCTION calctime(id\_w integer)

RETURNS double precision AS

$BODY$

DECLARE

hours double precision;

startd date;

iterdate date;

endd date;

cur date;

dayi int;

chd date;

beg time;

en time;

delta double precision;

del interval;

BEGIN

—- need to get the date of appliance

select date\_start\_work into startd from worker where id\_worker=id\_w;

select date\_end\_work into endd from worker where id\_worker=id\_w;

iterdate := startd;

—- now calculating the total amount of hours

hours := 0;

LOOP

delta := 0;

dayi := extract(isodow from iterdate); —-getting weekday

select stime, etime into beg,en from schedule WHERE changed > iterdate AND id\_worker = id\_w AND weekday = dayi ORDER BY changed ASC;

IF beg > en THEN

del := beg - en;

ELSE

del := en - beg;

END IF;

delta := extract(hour from del );

IF delta IS NOT NULL THEN

hours := hours + delta;

END IF;

iterdate := iterdate + 1;

if endd IS NOT NULL THEN

cur := endd;

ELSE

cur := current\_date;

END IF;

IF iterdate = cur THEN

EXIT; —-finished iterating

END IF;

END LOOP;

RETURN hours;

END;

$BODY$

LANGUAGE plpgsql IMMUTABLE

Функція для підрахунку отриманої зарплатні за час роботи на посаді:

Додаток А

CREATE OR REPLACE FUNCTION calcsalary(id\_w integer)

RETURNS double precision AS

$BODY$

DECLARE

monthly float;

sal double precision;

startd timestamp;

endd timestamp;

iterdate date;

dayi int;

beg time;

en time;

delta double precision;

delt double precision;

del interval;

BEGIN

—- need to get the date of appliance

select date\_start\_work into startd from worker where id\_worker=id\_w;

select date\_end\_work into endd from worker where id\_worker=id\_w;

select salary into monthly from worker where id\_worker=id\_w;

—-iterdate := startd;

—- now calculating the total amount of hours

IF endd IS NOT NULL THEn

del := endd - startd;

ELSE

del := localtimestamp - startd;

END IF;

delta := extract(month from del );

delt := extract(day from del);

raise notice 'Days: %',del;

raise notice 'Months: %',delta;

sal := delta \* monthly + monthly \* delt / 30 ;

RETURN sal;

END;

$BODY$

LANGUAGE plpgsql IMMUTABLE

Функція для перевірки заповненсті полів даних працівника:

CREATE OR REPLACE FUNCTION if\_worker\_full(id\_w integer)

RETURNS boolean AS

$BODY$

BEGIN

RETURN if\_stuff\_list\_full(id\_w) AND if\_stuff\_full(id\_w) AND if\_schedule\_full(id\_w) AND EXISTS(SELECT \* FROM worker WHERE (id\_paper\_worker IS NOT NULL) AND (id\_position IS NOT NULL) AND (date\_start\_work IS NOT NULL) AND (salary IS NOT NULL) AND (id\_stuff IS NOT NULL) AND id\_position = id\_w);

END;

$BODY$

LANGUAGE plpgsql VOLATILE

**Додаток2**. Процедура та функції для відділу руху товарів  
Функція додавання квитка:

-- Function: private.insert\_ticket(integer, integer, integer, text, text)

-- DROP FUNCTION private.insert\_ticket(integer, integer, integer, text, text);

CREATE OR REPLACE FUNCTION private.insert\_ticket(

v\_id\_haul integer DEFAULT 0,

v\_seat integer DEFAULT 0,

v\_carriage integer DEFAULT 0,

v\_contractor\_name text DEFAULT ''::text,

v\_operation\_type text DEFAULT ''::text)

RETURNS void AS

$BODY$DECLARE

v\_id\_cont integer;

v\_id\_invoice integer;

v\_id\_ticket integer;

v\_prev\_id\_cont integer;

v\_prev\_op\_type text;

v\_prev\_op\_quantity integer;

v\_prev\_ticket\_price integer;

BEGIN

-- if there exists such ticket, then use its id, if not, then create it and return id

IF EXISTS (SELECT id

FROM private.ticket

WHERE (id\_haul = v\_id\_haul

AND carriage = v\_carriage

AND seat = v\_seat))

THEN SELECT id INTO v\_id\_ticket

FROM private.ticket

WHERE id\_haul = v\_id\_haul

AND carriage = v\_carriage

AND seat = v\_seat;

ELSE INSERT INTO private.ticket

(id, id\_haul, carriage, seat)

VALUES

(DEFAULT, v\_id\_haul, v\_carriage, v\_seat)

RETURNING id INTO v\_id\_ticket;

END IF;

-- get id of contractor which we should use to insert new invoice

SELECT id INTO v\_id\_cont

FROM private.contractor\_types\_and\_names

WHERE name = v\_contractor\_name;

-- setting previous operation type, quantity, contractor

v\_prev\_op\_type = '';

v\_prev\_op\_quantity = 0;

v\_prev\_id\_cont = 0;

v\_prev\_ticket\_price = 0;

SELECT i.type, gl.quantity, i.contractor , gl.price

INTO

v\_prev\_op\_type, v\_prev\_op\_quantity, v\_prev\_id\_cont, v\_prev\_ticket\_price

FROM private.invoice as i,

private.goods\_list as gl

WHERE i.id = gl.id\_invoice

AND gl.object = v\_id\_ticket

ORDER BY i.id DESC

LIMIT 1;

-- depending on operation type we should take different actions

CASE v\_operation\_type

WHEN 'return'

THEN CASE

WHEN (v\_prev\_op\_type = 'buy'

AND v\_prev\_op\_quantity = 1

AND v\_prev\_id\_cont = v\_id\_cont)

THEN BEGIN -- create new identifer for this invoice

INSERT INTO private.invoice

(contractor, type)

VALUES

(v\_id\_cont, 'buy') RETURNING id into v\_id\_invoice;

INSERT INTO private.goods\_list

(object, quantity, price, id\_invoice)

VALUES

(v\_id\_ticket, -1, v\_prev\_ticket\_price, v\_id\_invoice);

END;

ELSE

RAISE EXCEPTION 'The ticket couldn''t be returned because it wasn''t bought by this customer before';

END CASE;

WHEN 'cancel booking'

THEN CASE

WHEN (v\_prev\_op\_type = 'book'

AND v\_prev\_op\_quantity = 1

AND v\_prev\_id\_cont = v\_id\_cont)

THEN BEGIN -- create new identifer for this invoice

INSERT INTO private.invoice

(contractor, type)

VALUES

(v\_id\_cont, 'book') RETURNING id into v\_id\_invoice;

INSERT INTO private.goods\_list

(object, quantity, price, id\_invoice)

VALUES

(v\_id\_ticket, -1, 0, v\_id\_invoice);

END;

ELSE

RAISE EXCEPTION 'This ticket wasn''t booked by this customer before, therfore, couldn''t cancel the booking';

END CASE;

WHEN 'buy'

THEN CASE

WHEN (v\_prev\_id\_cont IS NULL

OR (v\_prev\_id\_cont = v\_id\_cont AND v\_prev\_op\_type = 'book')

OR (v\_prev\_op\_type = 'book' AND v\_prev\_op\_quantity = '-1')

OR (v\_prev\_op\_type = 'buy' AND v\_prev\_op\_quantity = '-1'))

AND (SELECT depart\_date FROM private.haul WHERE id = v\_id\_haul) > now()

THEN BEGIN -- create new identifer for this invoice

INSERT INTO private.invoice

(contractor, type)

VALUES

(v\_id\_cont, 'buy') RETURNING id into v\_id\_invoice;

INSERT INTO private.goods\_list

(object, quantity, price, id\_invoice)

VALUES

(v\_id\_ticket, 1, 100, v\_id\_invoice);

END;

ELSE RAISE EXCEPTION 'This ticket couldn''t be bought';

END CASE;

WHEN 'book'

THEN CASE

WHEN (v\_prev\_id\_cont IS NULL

OR (v\_prev\_op\_type = 'book' AND v\_prev\_op\_quantity = '-1')

OR (v\_prev\_op\_type = 'buy' AND v\_prev\_op\_quantity = '-1'))

AND (SELECT depart\_date FROM private.haul WHERE id = v\_id\_haul) > now()

THEN BEGIN -- create new identifer for this invoice

INSERT INTO private.invoice

(contractor, type)

VALUES

(v\_id\_cont, 'book') RETURNING id into v\_id\_invoice;

INSERT INTO private.goods\_list

(object, quantity, price, id\_invoice)

VALUES

(v\_id\_ticket, 1, 0, v\_id\_invoice);

END;

ELSE RAISE EXCEPTION 'This ticket couldn''t be booked';

END CASE;

ELSE RAISE EXCEPTION 'operation not recognized';

END CASE;

END;$BODY$

LANGUAGE plpgsql VOLATILE