НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра автоматизації та систем неруйнівного контролю

(повна назва кафедри)

# **КУРСОВА РОБОТА**

з технології розроблення програмного забезпечення

(назва кредитного модуля)

на тему: Автоматизована система керування складом аптеки

Студентів 2 курсу групи ПК-11

Котусенко А.В.

Попов В.О.

Дрот М.О.

Алексеєнко Г.І.

Керівник ас., Паздрій О.Я.\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Залікова оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц., к.т.н. Цибульник С.О. . л

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ас., Паздрій О.Я. . .

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кондратенко Д.Ю. .

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2023 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра автоматизації та систем неруйнівного контролю

Дисципліна (кредитний модуль) Курсова робота з технології розроблення програмного забезпечення

Викладач доц., к.т.н. Цибульник С.О.

Курс 2 Група ПК-11 Семестр 3

# **Завдання**

на курсову роботу студентів

Котусенко А.В.; Попов В.О.; Дрот М.О; Алексеєнко Г.І..

на тему

Автоматизована система керування складом аптеки щ   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Строк здачі студентом виконаної роботи (проекту)

2. Вихідні дані до роботи (проекту): розробити додаток за обраною темою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Зміст пояснювальної записки (відповідно до пунктів РСО) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Титульний аркуш. Бланк завдання та календарний план. Вступ. 1. Технічне завдання. 2. Архітектура проекту. 3. Програмний код. 4. Результати розробки. 5. Програмна документація. 6. Керівництво користувача. 7. Результати тестування. Висновки Література. Додатки.

4. Перелік графічного матеріалу з точним зазначенням обов’язкових креслень та їх формату\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Дата видачі завдання на курсову роботу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Календарний план**

(відповідно до РСО)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Назва пунктів курсової роботи (проекту) | Строк виконання роботи (проекту) | Бали, нараховані за виконання календарного плану | Підпис викладача |
| 1 | Формування вимог до програмного забезпечення | 20.09.2021р. |  |  |
| 2 | Створення технічного завдання на розробку | 27.09.2021р. |  |  |
| 3 | Архітектурне проектування з забезпеченням нефункціональних вимог (безпека, надійність, розширюваність, тощо) | 11.10.2021р. |  |  |
| 4 | Документування архітектури | 18.10.2021р. |  |  |
| 5 | Програмування основного функціоналу (графічний інтерфейс користувача, бізнес-логіка, тощо) | 15.11.2021р. |  |  |
| 6 | Оформлення програмної документації | 22.11.2021р. |  |  |
| 7 | Тестування | 29.11.2021р. |  |  |
| 8 | Виправлення багів | 06.12.2021р. |  |  |
| 9 | Написання детального керівництва користувача | 13.12.2021р. |  |  |
| 10 | Оформлення пояснювальної записки до курсової роботи | 20.12.2021р. |  |  |

Керівник роботи . ас., Паздрій О.Я. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Студент: Дрот М.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Студент: Попов В.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Студент: Котусенко А.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Студент: Алексеєнко Г.І. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

підпис

Кількість балів за виконання роботи \_\_\_\_\_\_\_\_\_\_\_

Кількість балів за дотримання етапів календарного плану \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сумарна кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сумарна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підпис

ЗМІСТ

[КУРСОВА РОБОТА 1](#_Toc124031064)

[Завдання 2](#_Toc124031065)

[Календарний план 3](#_Toc124031066)

[ЗМІСТ 4](#_Toc124031067)

[ТЕХНІЧНЕ ЗАВДАННЯ 6](#_Toc124031068)

[**1.Вступ** 6](#_Toc124031069)

[**1.1.Найменування програми:** Автоматизована система керування складом аптеки. 6](#_Toc124031070)

[**1.2.Призначення розробки:** 6](#_Toc124031071)

[**2.Вимоги до програми** 7](#_Toc124031072)

[**2.1.Вимоги до функціональних характеристик(Рис.№1)** 7](#_Toc124031073)

[**2.2 Вимоги до надійності** 8](#_Toc124031074)

[**2.2.1.Вимоги до забезпечення надійного функціонування програми:** 8](#_Toc124031075)

[**2.2.2. Час відновлення програми після відмови:** 8](#_Toc124031076)

[**2.2.3. Відмови програми із-за некоректных дій оператора:** 8](#_Toc124031077)

[**3 Умови експлуатації** 8](#_Toc124031078)

[**3.1. Кліматичні умови експлуатації програми** 8](#_Toc124031079)

[**3.2. Вимоги до кваліфікації і чисельності персоналу** 8](#_Toc124031080)

[**3.3. Вимоги до складу і параметрів технічних засобів** 8](#_Toc124031081)

[**3.4. Вимоги до інформаційної сумісності:** 9](#_Toc124031082)

[**3.4.1. Вимоги до інформаційних структур і методів рішення** 9](#_Toc124031083)

[**3.4.2. Вимоги до початкових код і мов програмування** 9](#_Toc124031084)

[**3.4.3. Вимоги до програмних засобів, використовуваних програмою** 9](#_Toc124031085)

[**3.4.4. Вимоги до захисту інформації і програм** 9](#_Toc124031086)

[**3.5. Спеціальні вимоги** 9](#_Toc124031087)

[**4. Вимоги до програмної документації** 9](#_Toc124031088)

[**4.1. Попередній склад програмної документації** 9](#_Toc124031089)

[**5. Техніко-економічних показників** 9](#_Toc124031090)

[**5.1. Економічні переваги розробки програми** 9](#_Toc124031091)

[**6. Стадії і етапи розробки програми** 10](#_Toc124031092)

[**6.1. Стадії розробки програми:** 10](#_Toc124031093)

[**6.2. Етапи розробки програми** 10](#_Toc124031094)

[**6.3. Зміст робіт по етапах** 10](#_Toc124031095)

[**7. Порядок контролю і приймання** 11](#_Toc124031096)

[**7.1. Видів випробувань** 11](#_Toc124031097)

[**7.2. Загальні вимоги до приймання роботи\** 11](#_Toc124031098)

[АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 13](#_Toc124031099)

[**Рівень 1(Контекстна діаграма системи):** 13](#_Toc124031100)

[**Рівень 2(Діаграма контейнера):** 15](#_Toc124031101)

[**Рівень 4(Код):** 16](#_Toc124031102)

[**Рівень 3(Діаграма компонентів):** 17](#_Toc124031103)

[ПРОГРАМНИЙ КОД 19](#_Toc124031104)

[ПРОГРАМНА ДОКУМЕНТАЦІЯ 20](#_Toc124031105)

[РЕЗУЛЬТАТИ ТЕСТУВАННЯ 27](#_Toc124031106)

[**ВИПРАВЛЕННЯ БАГІВ** 29](#_Toc124031107)

[КЕРІВНИЦТВО КОРИСТУВАЧА 30](#_Toc124031108)

[**АКАУНТ «ВХОДУ»** 30](#_Toc124031109)

[**АКАУНТ ПРОДАВЦЯ** 31](#_Toc124031110)

[**«СКЛАД»** 31](#_Toc124031111)

[**«ЗАМОВЛЕННЯ»** 32](#_Toc124031112)

[**«ВИЙТИ»** 33](#_Toc124031113)

[**АКАУНТ АДМІНІСТРАТОРА** 34](#_Toc124031114)

[**«СКЛАД»** 34](#_Toc124031115)

[**«ЗАМОВЛЕННЯ»** 35](#_Toc124031116)

[**«ПРАЦІВНИКИ»** 37](#_Toc124031117)

[**«ВИЙТИ»** 38](#_Toc124031118)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 40](#_Toc124031119)

# **ТЕХНІЧНЕ ЗАВДАННЯ**

**Перелік умовних позначень та скорочень**

|  |  |
| --- | --- |
| Позначення | Опис |
| БД | База даних |
| ОС | Операційна система |
|  |  |

## **1.Вступ**

Робота виконується в рамках проекту автоматизації управління складом лікарняної аптеки.

ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### **1.1.Найменування програми:** Автоматизована система керування складом аптеки.

### **1.2.Призначення розробки:**

Аптека  — медично-санітарний заклад, що виготовляє і відпускає ліки за рецептами, продає готові лікарські засоби, лікарські мінеральні води, перев'язувальні матеріали, предмети догляду за хворими та інші медичні вироби. Для виконання такої роботи потрібно багато часу і трудовитрат, щоб пришвидшити цей процес можна впровадити заходи автоматизації складського приміщення. Завдяки автоматизованому процесу, можна вести облік наявності товару, його термін придатності та надходження. Об'єктом дослідження є склад лікарняної аптеки. Предметом дослідження є інформаційний потік, який складається з находження, витрати, зберігання та збуту лікарських засобів. Метою роботи є розробка Автоматизованої системи складського обліку лікарських засобів.

Система автоматизації аптеки буду використана для утворення електронних накладних та подальше збереження їх у базу даних (БД). Після впровадження автоматизації, звітувати про надходження, витрати й залишки стане простіше.

Опис предметної області і її особливостей. На склад своєчасно потрібно постачати усі життєво необхідні ліки для певної кількості відділень аптек. Також на складі зобов’язані бути всі заходи для збереження вимог складування.

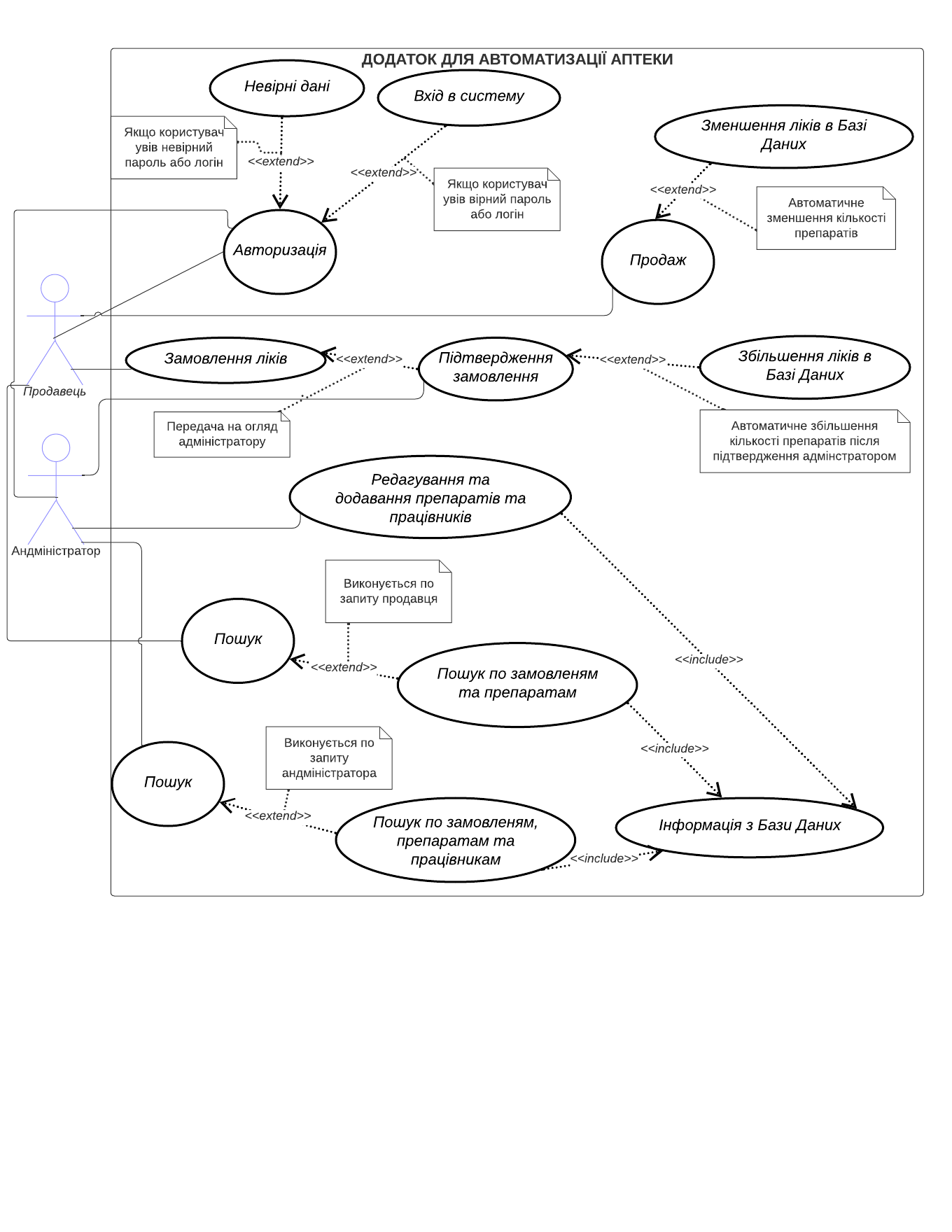
Отримання, розміщення та видачу засобів аптеки, виконує особа з відповідним рівнем допуску – фармацевт. Важливо вести облік, щоб на складі були лише придатні ліки й в достатній кількості.

Працівник складу та аптеки, зможуть: контролювати отримування, зберігання, постачання у відділення; ідентифікація товару на складі, з метою швидкого пошуку товару на складі.

## **2.Вимоги до програми**

### **2.1.Вимоги до функціональних характеристик(Рис.№1)**

* Авторизація працівників аптеки.
* Перегляд(швидкий пошук) наявності лікарських засобів на складі та при необхідності в загальній базі даних всіх аптек.
* При пошуку можливість вибору пошуку в локальній чи глобальній БД;
* Можливість ведення обліку надходжень та продажу ліків.
* При покупці товару інформація про покупку проходить через програму;



**Рис.№1** Діаграма претендантів

### **2.2 Вимоги до надійності**

#### **2.2.1.Вимоги до забезпечення надійного функціонування програми:**

Програма повинна:

* мати активний протокол захисти БД проти шахраїв.;
* контролювати доступ працівників та адміністраторів до програми за допомого індетифікації(логін та пароль);

#### **2.2.2. Час відновлення програми після відмови:**

Час відновлення програми при стабільному зв'зку з сервером, не повинен перевищувати 5хв, щоб не затримувати клієнтів та не зробити помилки при продажі.

#### **2.2.3. Відмови програми із-за некоректных дій оператора:**

Програма повинна повідомляти користувача при введенні невірного паролю або логіну, невірної інформації про товар, після перевірки з глобальної БД або введенні некоректних значень.

**3 Умови експлуатації**

**3.1. Кліматичні умови експлуатації програми**

* відповідна вологість(45-60%) та температура (+5 – +30 0С) експлуатації електронно обчислюваної машини.

**3.2. Вимоги до кваліфікації і чисельності персоналу**

* для використання цієї програми достатньо одної персони, а більша кількість персоналу потрібна в залежить від розміру складу;
* для використання програми персонал повинен ознайомитися з інструкцією по використанню.

**3.3. Вимоги до складу і параметрів технічних засобів**

* для коректного функціювання програми потрібно комп’ютер з ОС Windows 10 та новіше.
* мінімальні вимоги до програмних і апаратних засобів для нормального функціонування програмного продукту: тактова частота процесора - 1 400 Гц; обсяг оперативної пам'яті - 1 Гб; обсяг вільного дискового простору - 10 Гб; дозвіл монітора - 1080 x 920.

### **3.4. Вимоги до інформаційної сумісності:**

#### **3.4.1. Вимоги до інформаційних структур і методів рішення**

База даних працює під управлінням Microsoft SQL Server.

#### **3.4.2. Вимоги до початкових код і мов програмування**

Додаткові вимоги не пред'являються.

#### **3.4.3. Вимоги до програмних засобів, використовуваних програмою**

Системні програмні засоби, використовувані програмою, мають бути представлені ліцензійною локалізованою версією операційної системи Windows 2022 Server або Windows 2022 і Java18.

#### **3.4.4. Вимоги до захисту інформації і програм**

Запровадити активний протокол захисти проти шахраїв та ідентифікацію особи за допомогою логіна та паролю.

### **3.5. Спеціальні вимоги**

Запровадити конроль за умовами зберігання та придатністю ліків.

## **4. Вимоги до програмної документації**

### **4.1. Попередній склад програмної документації**

**В ході розробки програми повинні бути підготовлені:**

* текст програми,
* опис програми,
* програма і методика тестування,
* керівництво користувача.

**5. Техніко-економічних показників**

**5.1. Економічні переваги розробки програми**

Орієнтовна економічна ефективність не розраховуються. Аналогія не проводиться зважаючи на унікальність вимог, що пред'являються, до розробки.

**6. Стадії і етапи розробки програми**

**6.1. Стадії розробки програми:**

**Розробка має бути проведена в три стадії:**

1. розробка технічного завдання;
2. робоче проектування;
3. впровадження.

**6.2. Етапи розробки програми**

* На стадії розробки технічного завдання має бути виконаний етап розробки, узгодження і затвердження справжнього технічного завдання.
* На стадії робочого проектування мають бути виконані перераховані нижче етапи робіт :
  1. розробка програми;
  2. розробка програмної документації;
  3. випробування програми.
* На стадії впровадження має бути виконаний етап розробки підготовка і передача програми

**6.3. Зміст робіт по етапах**

* На етапі розробки технічного завдання мають бути виконані перераховані нижче роботи:

1. постановка завдання;
2. визначення і уточнення вимог до технічних засобів;
3. визначення вимог до програми;
4. визначення стадій, етапів і термінів розробки програми і документації на неї;
5. узгодження і затвердження технічного завдання.

* На етапі розробки програми має бути виконана робота по програмуванню (кодуванню) і відладці програми.
* На етапі розробки програмної документації має бути виконана розробка програмних документів відповідно до вимог до складу документації.
* На етапі випробувань програми мають бути виконані перераховані нижче види робіт :

1. розробка, узгодження і твердження і методики випробувань;
2. проведення приймально-здавальних випробувань;
3. коригування програми і програмної документації за результатами випробувань.

* На етапі підготовки і передачі програми має бути виконана робота з підготовки і передачі програми і програмної документації в експлуатацію на об'єктах Замовника.

**7. Порядок контролю і приймання**

**7.1. Видів випробувань**

Приймально-здавальні випробування повинні проводитися на об'єкті Замовника в обумовлені терміни.

Приймально-здавальні випробування програми повинні проводитися згідно розробленої Виконавцем і узгодженої Замовником Програми і методик випробувань.

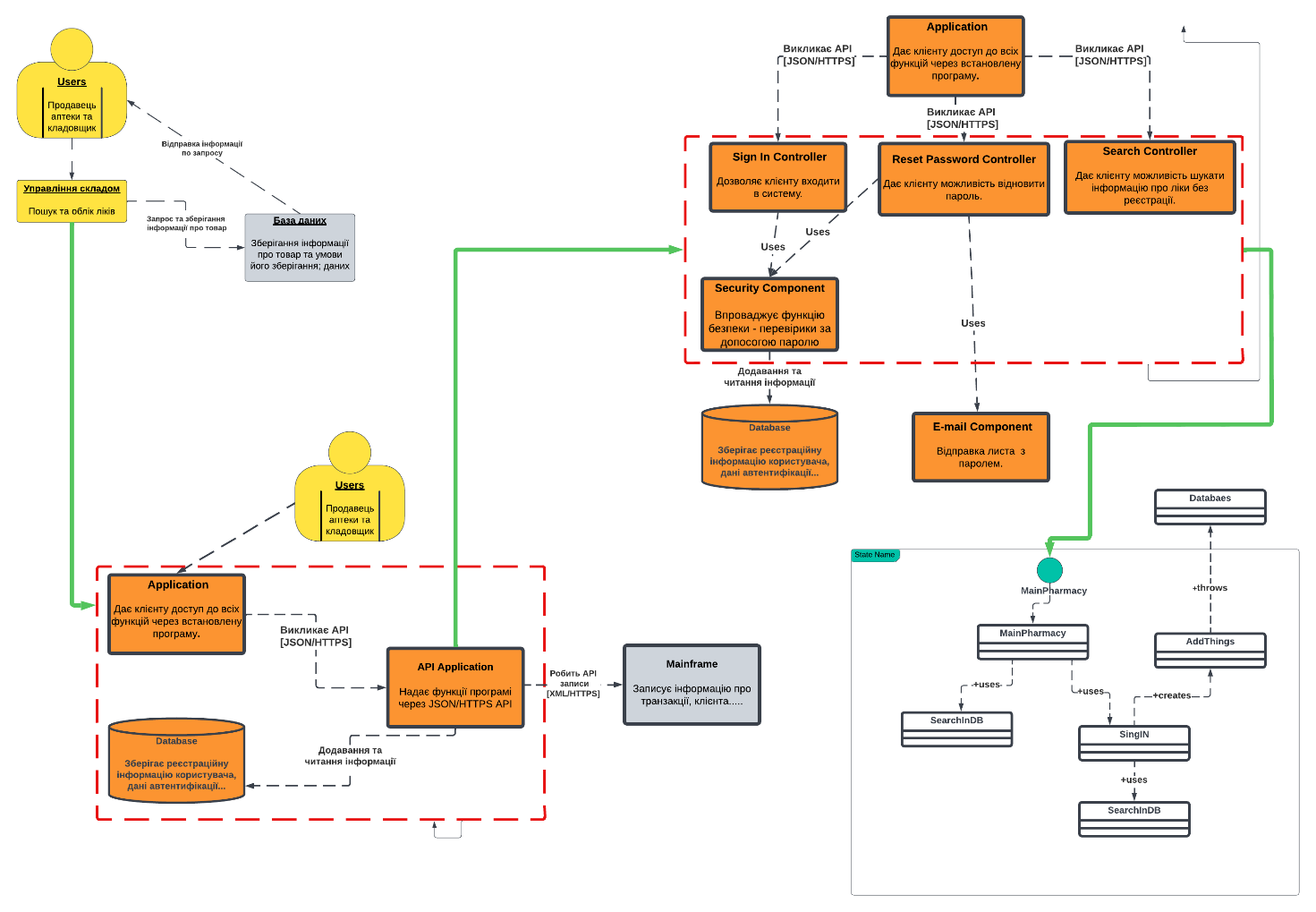
Хід проведення приймально-здавальних випробувань Замовник і Виконавець документують в Протоколі проведення випробувань

**7.2. Загальні вимоги до приймання роботи\**

На підставі Протоколу проведення випробувань Виконавець спільно із Замовником підписує Акт приймання-здачі програми в експлуатацію.

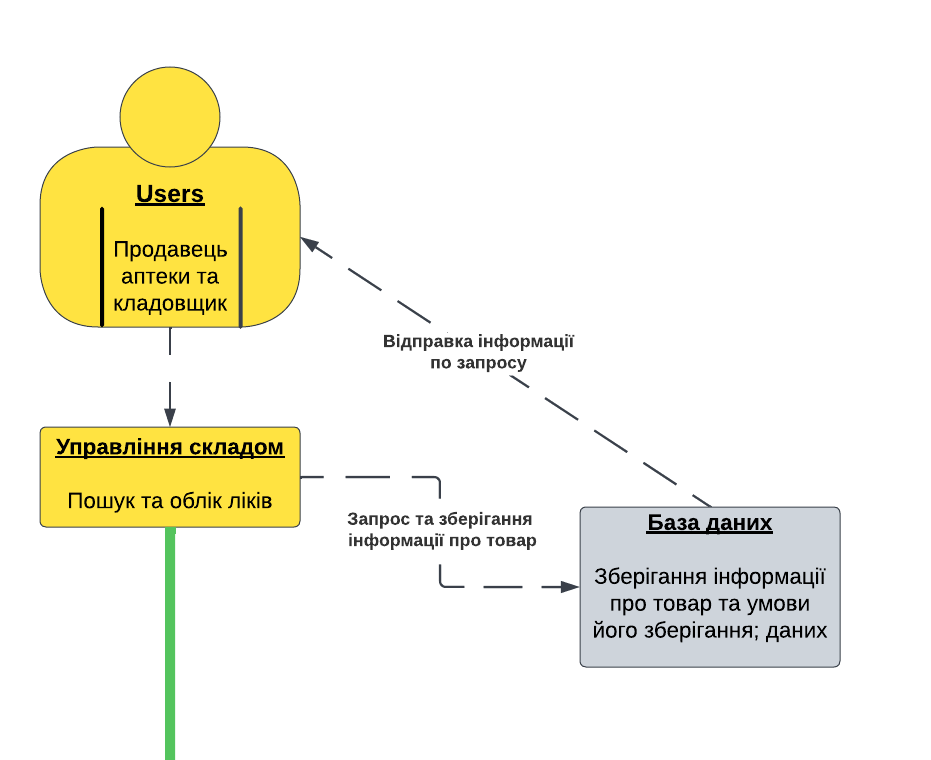
# [**АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**](#_5._Архітектура_програмного)

Для оформлення архітектури даного програмного забезпечення використаємо модель С4, яка формується в 4рівні:



## **Рівень 1(Контекстна діаграма системи):**

Тут зображено взаємодію програми з користувачем. Користувач має можливість вести облік товарів(зберігати, додавати та змінювати інформацію), також шукати інформацію про ліки в базі даних.



**Архітектурні цілі та обмеження**

Будь-яка аптека – складне комерційне підприємство, ключова особливість якого – продукт. Аптекам та аптечним складам доводиться працювати з величезною номенклатурою: препарати надходять у різних дозуваннях та лікарських формах, різниця у цінових пропозиціях від різних дистриб'юторів та виробниках, необхідність відстежувати та переоцінювати залишки на складі, тощо. як у будь-якому великому retail'і, але додатково присутній державний контроль за ціноутворенням та специфічні моделі властиві саме фарм. бізнесу – всі ці процеси можна автоматизувати. Сфера застосування даного програмного забезпечення, внутрішній облік товарів в аптеці. Також контроль умов зберігання та продажу ліків.

Архітектура призначена для дизайнерів, програмістів та людей, які можливо будуть вдосконалювати програму (вони повинні володіти знаннями в розробці програмного забезпечення та мовою проuрамування Java). Дана архітектурна дасть змогу дізнатися всю детальну інформацію.

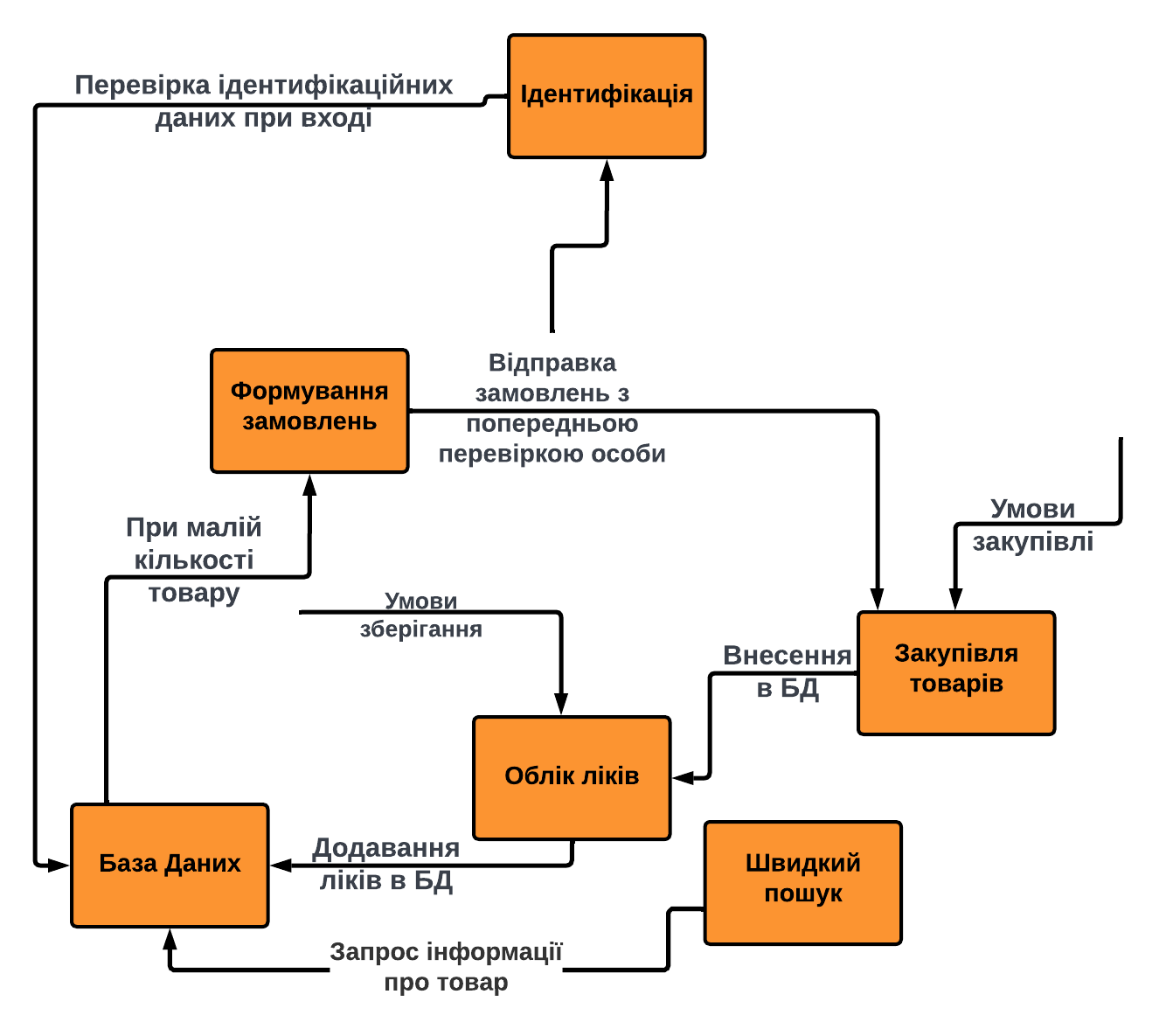
Для користувачів інформаційної системи недостатньо, щоб база даних просто відбивала об'єкти реального світу. Важливо, щоб такий відбиток був однозначним і несуперечливим.

**Логічна архітектура**

Діаграма, уточнює функції програми. На цій діаграмі присутні п’ять блоків:

1. **Блок формування списку замовлень;**
2. **Блок закупівлі товарів до аптеки;**
3. **Блок обліку товарів на складі;**
4. **Блок ідентифікації;**
5. **Блок Бази Даних;**
6. **Блок швидкого пошуку.**

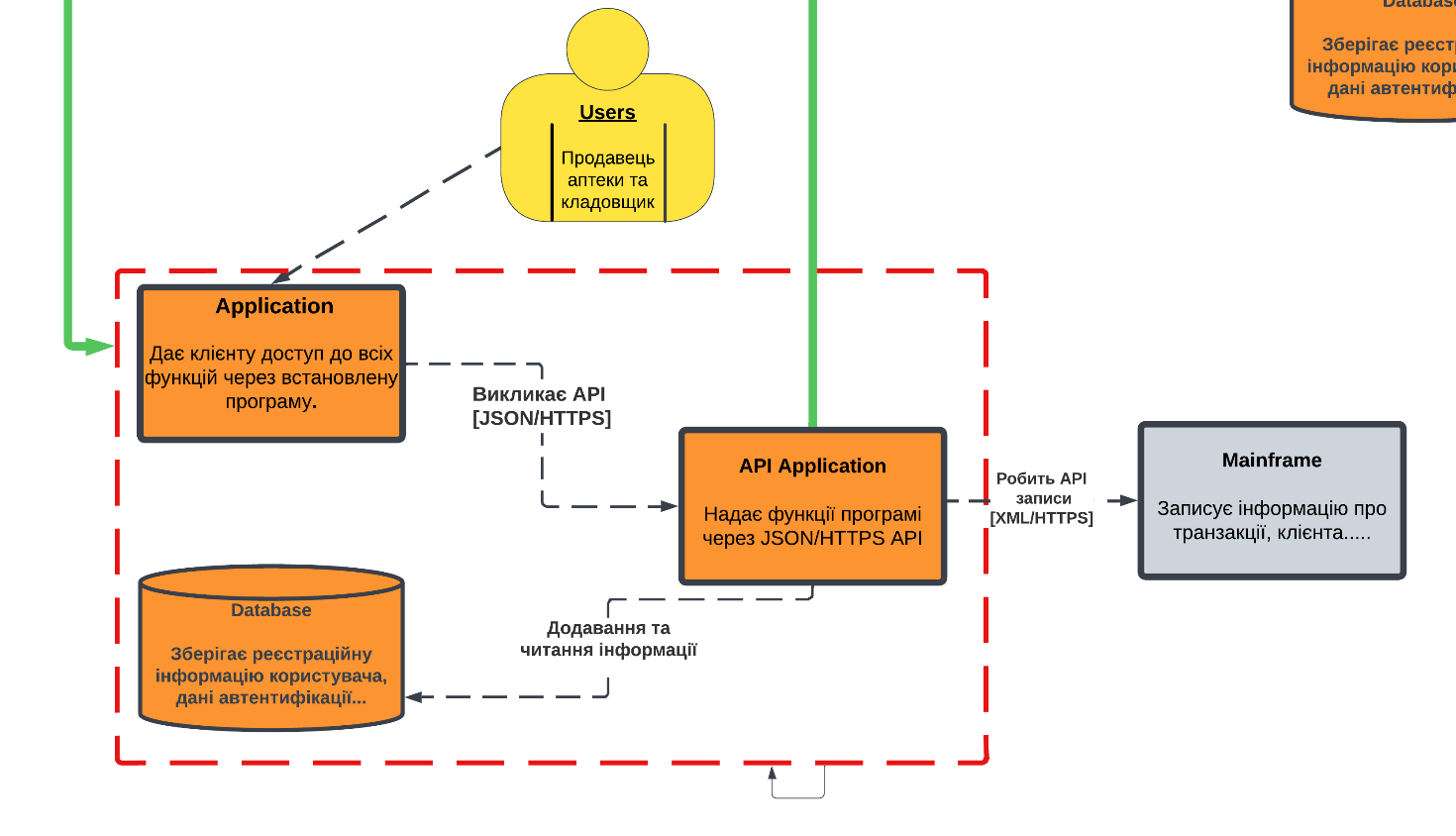
Умови закупівлі та збереження товарів надаються як список усіх необхідних умов для правильної закупівлі та безпечного зберігання товарів на складі. Перед закупівлею проходить формування списку необхідних товарів, під час якого перевіряється залишок товару на складі а також його резерв. Після цього інформація перевіряється знову і надається дозвіл на закупівлю. Коли товари будуть привезені, вони розміщуються на складі, а потім постачаються у відділення аптеки у необхідній кількості.



**Архітектура процесу**

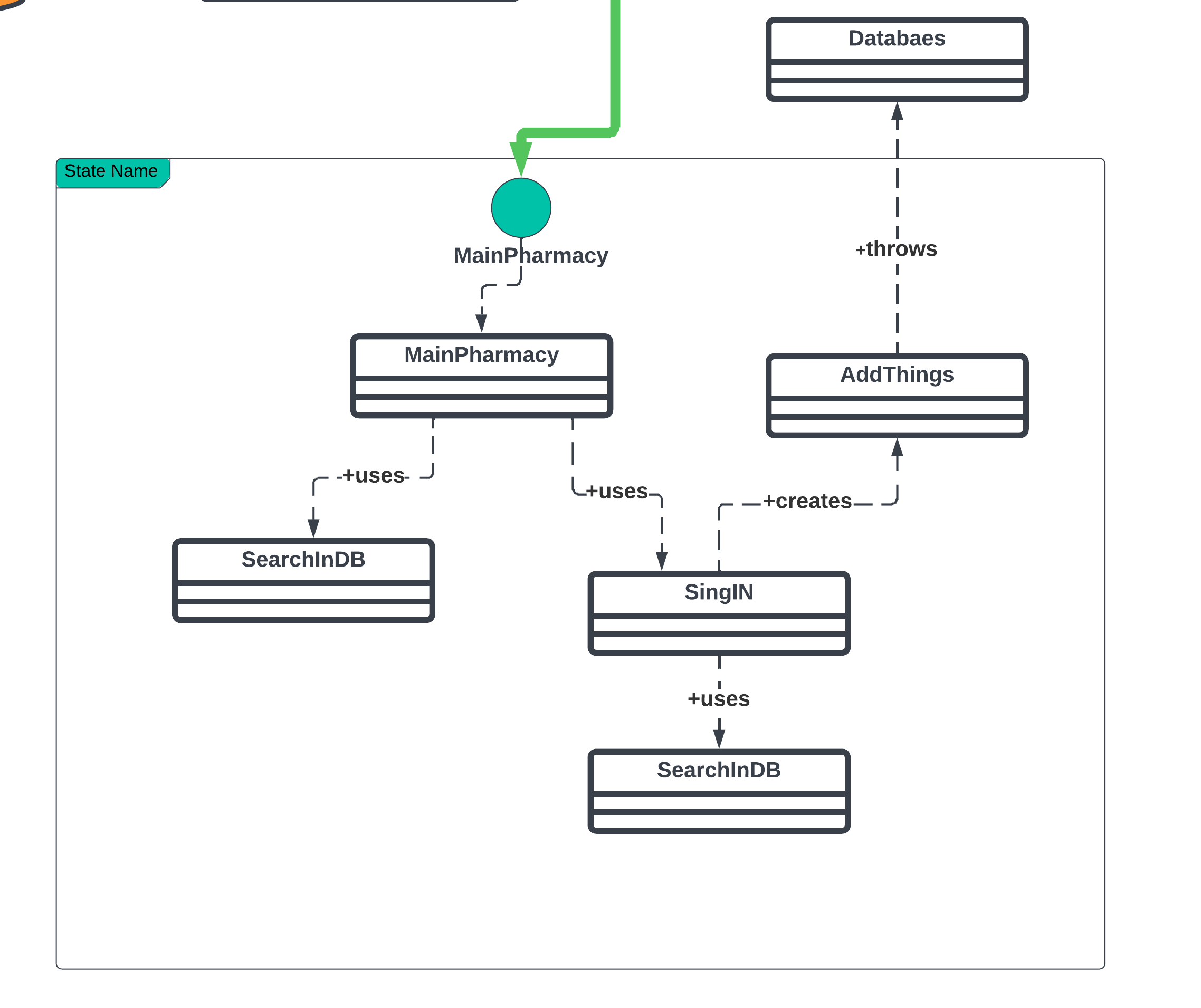
## **Рівень 2(Діаграма контейнера):**

Контейнерна діаграма дає нам уявлення про роботу БД та самої програми. Користувач запрошує інформацію в програми, яка в свою чергу запрошує інформацію в Database через API[JSON/HTTPS].



**Архітектура розробки**

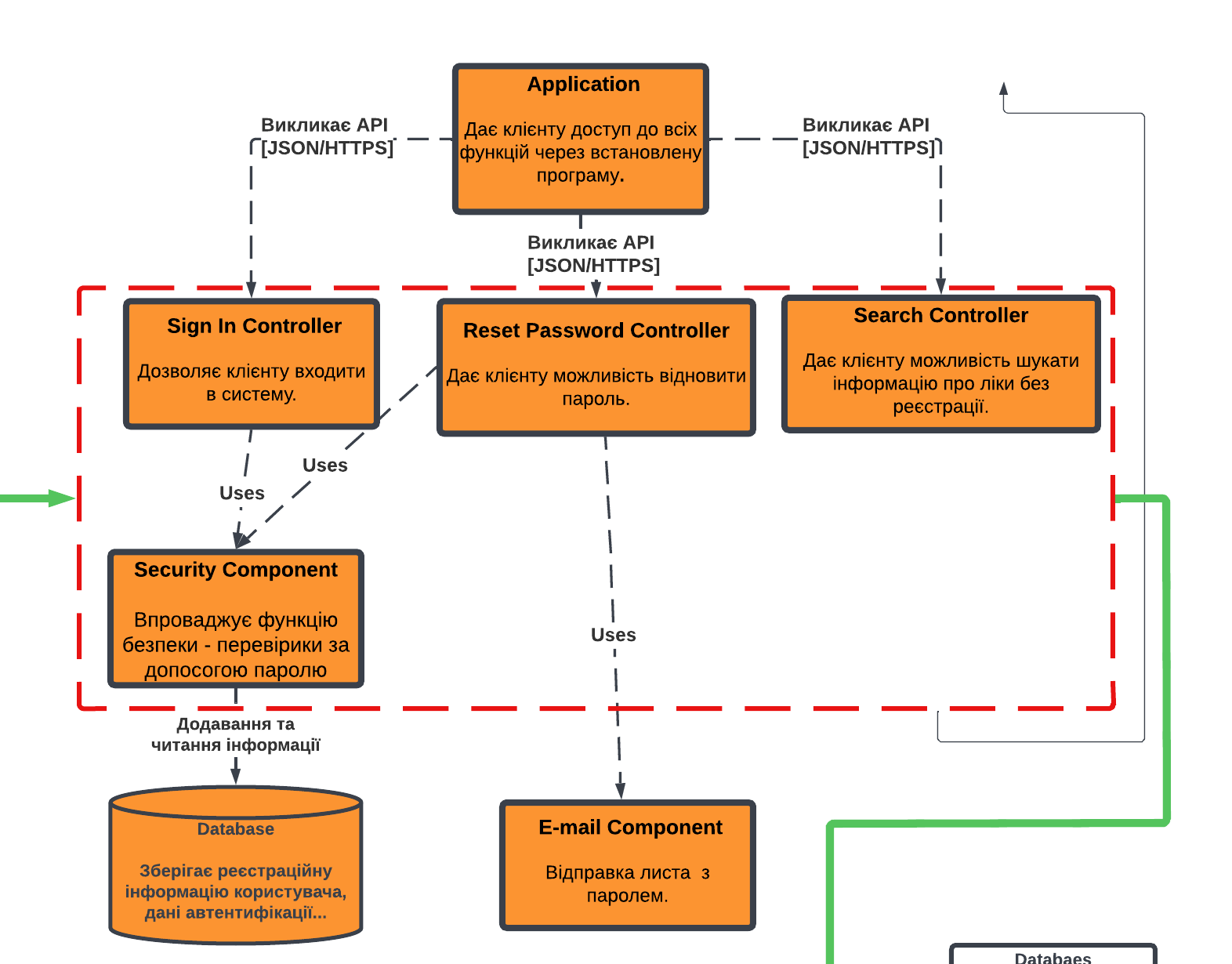
## **Рівень 4(Код):**



**Фізична архітектура**

## **Рівень 3(Діаграма компонентів):**

Компонентна діаграма, спочатку програми користувач має вибір: увійти, відновити пароль або шукати ліки. Якщо користувач вибирає вхід, то введені дані перевіряються через базу даних, якщо відновлення паролю, то він відправляється йому на пошту.



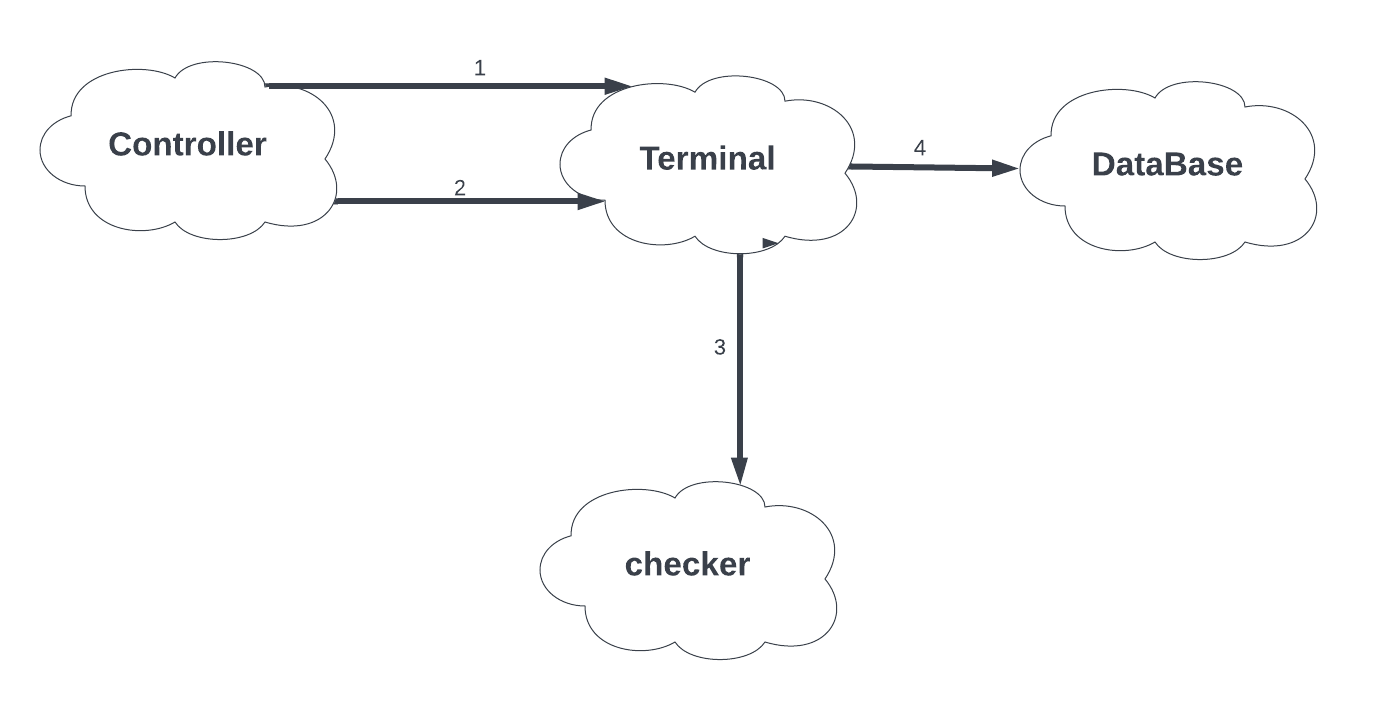
**Сценарії**

Зображено фрагмент сценарію при введені інформації в поле входу.

1.Контролер виявляє натискання кнопки.

2. Контролер приймає дані та передає терміналу.

3.Термінал передає їх на перевірку,

4.Після збігу даних, відкривається потрібний відділ.

# **ПРОГРАМНИЙ КОД**

**Клас Main**

package com.metanit.Main;  
  
import com.metanit.Controller.LoginController;  
import javafx.application.Application;  
import javafx.event.EventHandler;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
import javafx.scene.Scene;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.stage.StageStyle;  
  
  
 public class Main extends Application{  
 private double xOffset;  
 private double yOffset;  
  
 public static void main(String[] args) {  
  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) throws Exception{  
  
  
 Parent root = FXMLLoader.*load*(LoginController.class.getResource("LoginController.fxml"));  
 Scene scene = new Scene(root, Color.*TRANSPARENT*);  
 primaryStage.setScene(scene);  
 scene.setOnMousePressed(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 xOffset = primaryStage.getX() - event.getScreenX();  
 yOffset = primaryStage.getY() - event.getScreenY();  
 }  
 });  
 scene.setOnMouseDragged(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 primaryStage.setX(event.getScreenX() + xOffset);  
 primaryStage.setY(event.getScreenY() + yOffset);  
 }  
 });  
  
 primaryStage.initStyle(StageStyle.*UNDECORATED*);  
 primaryStage.initStyle(StageStyle.*TRANSPARENT*);  
 primaryStage.setResizable(false);  
  
 primaryStage.show();  
 }  
  
  
  
}

**Клас LoginController(Частина)**

package com.metanit.Controller;  
  
import java.io.IOException;  
import java.net.URL;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.ResourceBundle;  
  
import com.metanit.Animation.Shake;  
import com.metanit.CLASS.UserLOGIN;  
  
import com.metanit.DATABASE.DatabaseHandler;  
  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
  
  
public class LoginController {  
  
 private static String *LOGIN* = null ;  
 private static String *PROFESSION* = null;  
 private static String *NAME* = null;  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button ButEnter;  
  
 @FXML  
 private ToggleGroup Kindofuser;  
 @FXML  
 private Button ButtonHide;  
 @FXML  
 private Button ButtonClose;  
  
 @FXML  
 private TextField LogField;  
  
 @FXML  
 private PasswordField PaswordField;  
  
 @FXML  
 private ToggleButton TogButAdmin;  
  
 @FXML  
 private ToggleButton TogButSeller;  
  
 @FXML  
 private Label UncorrectLog;  
  
 private double xOffset;  
 private double yOffset;  
  
  
  
 @FXML  
 void initialize() {  
 //кнопка входу  
 ButEnter.setOnAction(actionEvent -> {  
 String loginText = LogField.getText().trim();  
 String paswordText = PaswordField.getText().trim();  
 UncorrectLog.setText("");  
//вибір продавця  
 if (TogButSeller.isSelected()) {  
 if (!loginText.equals("") && !paswordText.equals("")) {  
 DatabaseHandler dbHandler = new DatabaseHandler();  
 UserLOGIN user = new UserLOGIN(loginText,"", paswordText, "Seller");  
 ResultSet result = dbHandler.getUser(user);  
  
 int counter = 0;  
 while (true) {  
 try {  
 if (!result.next()) break;  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
  
 counter++;  
 try {  
 this.*NAME* = (result.getString("FirstandLastname"));  
 this.*PROFESSION* ="Seller";  
 this.*LOGIN* = (result.getString("login"));  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
  
  
 }  
  
 if (counter >= 1){  
  
 ButEnter.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader(ControllerSell.class.getResource("ControllerSell.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 Scene scene = new Scene(root, Color.*TRANSPARENT*);  
 stage.setScene(scene);  
 scene.setOnMousePressed(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 xOffset = stage.getX() - event.getScreenX();  
 yOffset = stage.getY() - event.getScreenY();  
 }  
 });  
 scene.setOnMouseDragged(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 stage.setX(event.getScreenX() + xOffset);  
 stage.setY(event.getScreenY() + yOffset);  
 }  
 });  
 stage.initStyle(StageStyle.*UNDECORATED*);  
 stage.initStyle(StageStyle.*TRANSPARENT*);  
 stage.setResizable(false);  
 stage.showAndWait();  
  
 }  
  
 else {  
 UncorrectLog.setText("Введено невірний логін або пароль!");  
 LogField.setText("");  
 PaswordField.setText("");  
 Shake();  
 }  
  
 } else {  
 UncorrectLog.setText("Будь ласка, введіть логін та пароль!");  
 LogField.setText("");  
 PaswordField.setText("");  
 Shake();  
 }  
 }

**Увесь код можна знайти за посиланням: https://github.com/Kotuss/PHARMACY.git**

# **ПРОГРАМНА ДОКУМЕНТАЦІЯ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Вхідні Дані** | **Повертає** | **Призначення** |
| ControllerSell | | | | |
| ShowOrderSeller() |  | Таблицю з даними | Вивод даних про замовлення до таблиці |
| Show() |  | Таблицю з даними | Вивод даних про препарати до таблиці |
| showPersonDetails() | Данні з таблиці препаратів | Данні з таблиці | Отримання данних з таблиці |
| showOrderDetails() | Данні з таблиці замовлень | Данні з таблиці | Отримання данних з таблиці |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| Timenow() |  | Поточну дату та час | Відображення дати та часу в додатку в реальному часі |
| ControllerAdmin | | | | |
| showItemDetails() | Данні з таблиці препаратів | Данні з таблиці | Отримання данних з таблиці |
| showOrderDetails() | Данні з таблиці замовлень | Данні з таблиці | Отримання данних з таблиці |
| showPersonDetails() | Данні з таблиці працівників | Данні з таблиці | Отримання данних з таблиці |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| ShowTable() |  | Таблицю з даними | Вивод даних про препарати до таблиці |
| ShowTableOrders() |  | Таблицю з даними | Вивод даних про замовлення до таблиці |
| ShowTableWorkers() |  | Таблицю з даними | Вивод даних про працівників до таблиці |
| AddWorkers() |  | Оновлена таблиці з замовленнями | Додати замовлення до таблиці замовлень |
| Timenow() |  | Поточну дату та час | Відображення дати та часу в додатку в реальному часі |
| Clear() |  |  | Очистка даних з полів для вводу |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ControllerRedactPerson | | | | |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| ControllerRedactItems | | | | |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| Method | Вхідні Дані | Повертає | Призначення |
| ControllerSell | | | | |
| ShowOrderSeller() |  | Таблицю з даними | Вивод даних про замовлення до таблиці |
| Show() |  | Таблицю з даними | Вивод даних про препарати до таблиці |
| showPersonDetails() | Данні з таблиці препаратів | Данні з таблиці | Отримання данних з таблиці |
| showOrderDetails() | Данні з таблиці замовлень | Данні з таблиці | Отримання данних з таблиці |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| Timenow() |  | Поточну дату та час | Відображення дати та часу в додатку в реальному часі |
| ControllerAdmin | | | | |
| showItemDetails() | Данні з таблиці препаратів | Данні з таблиці | Отримання данних з таблиці |
| showOrderDetails() | Данні з таблиці замовлень | Данні з таблиці | Отримання данних з таблиці |
| showPersonDetails() | Данні з таблиці працівників | Данні з таблиці | Отримання данних з таблиці |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| ShowTable() |  | Таблицю з даними | Вивод даних про препарати до таблиці |
| ShowTableOrders() |  | Таблицю з даними | Вивод даних про замовлення до таблиці |
| ShowTableWorkers() |  | Таблицю з даними | Вивод даних про працівників до таблиці |
| AddWorkers() |  | Оновлена таблиці з замовленнями | Додати замовлення до таблиці замовлень |
| Timenow() |  | Поточну дату та час | Відображення дати та часу в додатку в реальному часі |
| Clear() |  |  | Очистка даних з полів для вводу |
| ControllerRedactPerson | | | | |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| ControllerRedactItems | | | | |
| CheckFields() | Значення для перевірки | Вірно, якщо значення задовільняють потреби; Невірно, якщо ні. | Перевірки значення вірні |
| isDigit() | Значення для перевірки | Вірно, якщо значення є числом, Невірно, якщо ні. | Перевірки значення чи є числом |
| DatabaseHandler | | | | |
| getUser() | Введений логін та пароль | Значення присутності | Перевірка наявності логін та вірності паролю |
| CheckName() | Введене ім'я | Значення присутності | Перевірка імені, чи воно вже існує пристворенні препарату |
| CheckUser() | Введений логін | Значення присутності | Перевірка логіну чи він вже існує пристворенні працівника |
| SELLItemsSeller() | ID та кількість продажи | Оновлену таблицю препаратів | Змешити кількість препарату після його продажу |
| UpdateItemsSeller() | ID та кількість поставки | Оновлену таблицю препаратів | Збільшення кількість препарату після його поставки |
| UpdateOrderSeller() | Оновленна інформація про замовлення | Оновлену таблицю замовлення | Редактування невиконаних замовлень |
| UpdateOrders() | ID замовлення | Оновлений статус замовлення | Оновити статус замовлення після підтвердження Адміном |
| UpdateItems() | Оновленна інформація про препарат | Оновлену таблицю препаратів | Редагувати препарат |
| DeleteOrder() | ID замовлення | Оновлену таблицю замовлення | Видалення замовлення |
| DeleteWorkers() | ID працівника | Оновлену таблицю працівників | Видалення працівника |
| DeleteItems() | ID препарату | Оновлену таблицю препаратів | Видалення препарату |
| AddOrder() | Інформація про замовлення | Оновлену таблицю замовлення |  |
| AddItems() | Інформація про препарат | Оновлену таблицю препаратів |  |
| AddPerson() | Інформація про працівника | Оновлену таблицю працівників |  |
| getItemToTableViewOrder() |  | Таблицю для виводу | Інформація з таблиця замовення конкретного продавця |
| getItemToTableViewOrderAdmin() |  | Таблицю для виводу | Інформація з таблиця замовення адміністратору |
| getItemToTableView() |  | Таблицю для виводу | Інформація з таблиця препаратів |
| getWorkerToTableView() |  | Таблицю для виводу | Інформація з таблиця працівників |

# **РЕЗУЛЬТАТИ ТЕСТУВАННЯ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Опис | Кроки тестування | Очікуваний результат | Фактичний результат | Результат тест-кейсу |
| 1 | Вибір опції для входу, як Продавець або Адміністратор | 1. Натиснути на кнопку "Адміністратор" або “Продавець” | Відбувається візуальне натиснення кнопки для інформування споживача | Відбувається візуальне натиснення кнопки для інформування споживача | Працює |
| 2 | Вхід до системи, як Адміністратор | 1. Вести логін і пароль  2. Натиснути на кнопку "УВІЙТИ" | Відбувається перехід до меню Адміністратора | Відбувається перехід до меню Адміністратора | Працює |
| 3 | Відображення препаратів в таблиці | 1.Автоматично після входу в систему або при натисненні на кнопку «СКЛАД» | Відбувається відображення препаратів в таблиці | Відбувається відображення препаратів в таблиці | Працює |
| 4 | Додавання препарата | 1. Натиснути кнопку "ДОДАТИ" | Відбувається відкриття вікна для додавання товару | Відбувається відкриття вікна для додавання товару | Працює |
| 5 | Вибір препарату | 1. Натискання на рядок в талиці | Вибір товару | Вибір товару | Працює |
| 6 | Видалення препарату | 1. Натиснути кнопку "ВИДАЛИТИ" | Відбувається видаленя препарату | Відбувається видаленя препарату | Працює |
| 7 | Редагування препарату | 1. Натиснути кнопку "РЕДАГУВАТИ" | Відбувається редагування препарату | Відбувається редагування препарату | Працює |
| 8 | Перехід до меню ЗАМОВЛЕННЯ | 1. Натиснути кнопку "ЗАМОВЛЕННЯ" | Відбувається перехід до меню ЗАМОВЛЕННЯ та відображення таблиці замовлень | Відбувається перехід до меню ЗАМОВЛЕННЯ та відображення таблиці замовлень | Працює |
| 9 | Виконання замовлення | 1. Натиснути на кнопку "ВИКОНАТИ" | Відбувається зміна статусу замовлення та додавання кількості товару на склад | Відбувається зміна статусу замовлення та додавання кількості товару на склад | Працює |
| 10 | Відмінити замовлення | 1. Натиснути на кнопку "ВІДМІНИТИ" | Відбувається видалення замовлення | Відбувається видалення замовлення | Працює |
| 11 | Перехід до меню ПРАЦІВНИКИ | 1. Натиснути кнопку "ПРАЦІВНИКИ" | Відбувається перехід до меню ПРАЦІВНИКИ та відображення таблиці працівників | Відбувається перехід до меню ПРАЦІВНИКИ та відображення таблиці працівників | Працює |
| 12 | Додавання працівника | 1. Натиснути кнопку "ДОДАТИ" | Відбувається відкриття вікна для додавання працівника | Відбувається відкриття вікна для додавання працівника | Працює |
| 13 | Видалення працівника | 1. Натиснути на кнопку "ВИДАЛИТИ" | Відбувається видалення працівника | Відбувається видалення працівника | Працює |
| 14 | Пошук препаратів | 1. Натиснути на кнопку "ПОШУК" | Відбувається пошук препарату | Відбувається пошук препарату | Працює |
| 15 | Вихід з меню АДМІНІСТРАТОРА | 1. Натиснути кнопку "ВИХІД" | Відбувається відкриття вікна ВХОДУ | Відбувається відкриття вікна ВХОДУ | Працює |
| 16 | Вхід до системи, як Продавець | 1. Вести логін і пароль  2. Натиснути на кнопку "УВІЙТИ" | Відбувається перехід до меню Продавця | Відбувається перехід до меню Продавця | Працює |
| 17 | Відображення препаратів в таблиці | 1.Автоматично після входу в систему або при натисненні на кнопку «СКЛАД» | Відбувається відображення препаратів в таблиці | Відбувається відображення препаратів в таблиці | Працює |
| 18 | Замовлення препарату | 1. Натиснути кнопку "ЗАМОВИТИ" | Відбувається замовлення препарату | Відбувається замовлення препарату | Працює |
| 19 | Перехід до меню ЗАМОВЛЕННЯ | 1. Натиснути кнопку "ЗАМОВЛЕННЯ" | Відбувається перехід до меню ЗАМОВЛЕННЯ та відображення таблиці замовлень | Відбувається перехід до меню ЗАМОВЛЕННЯ та відображення таблиці замовлень | Працює |
| 20 | Видалити замовлення | 1. Натиснути на кнопку "ВИДАЛИТИ" | Відбувається видалення замовлення | Відбувається видалення замовлення | Працює |
| 21 | Редагувати замовлення | 1. Натиснути на кнопку "РЕДАГУВАТИ" | Відбувається редагування замовлення | Відбувається редагування замовлення | Працює |
| 22 | Вихід з меню Продавця | 1. Натиснути кнопку "ВИХІД" | Відбувається відкриття вікна ВХОДУ | Відбувається відкриття вікна ВХОДУ | Працює |
|  | Проведено тестів 22/22; Дата:18.12.2022 | | Середовище тестування: Windows 10 | |  |

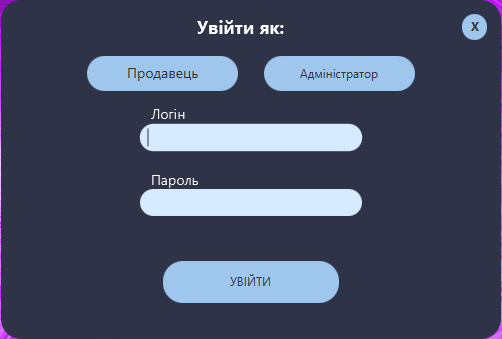
## **ВИПРАВЛЕННЯ БАГІВ**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Опис багу | Результат виправлення | Результат |
| 1 | Некоректне виведення повідомлення при введенні неправильного логіну чи паролю при вході як адміністратор | Виводиться червоний текст який сповіщує про неправильність введенного логіну чи паролю | Виконано |
| 2 | При додавання товару некоректна робота перевірки даних | Відбувається перевірка на вміст полів при додаванні препарату. | Виконано |
| 3 | Продавець може видаляти замовлення інших працівників | Відбувається виведення продавцю тільки його замовлень | Виконано |
|  | Виправленно багів: 3 Дата проведення виправленнь: 18.12.2022 | | | |

# **КЕРІВНИЦТВО КОРИСТУВАЧА**

## **АКАУНТ «ВХОДУ»**

На самому початку нас зустрічає вікно входу в програму, користувач обирає свою професію і вводить пароль та логін:

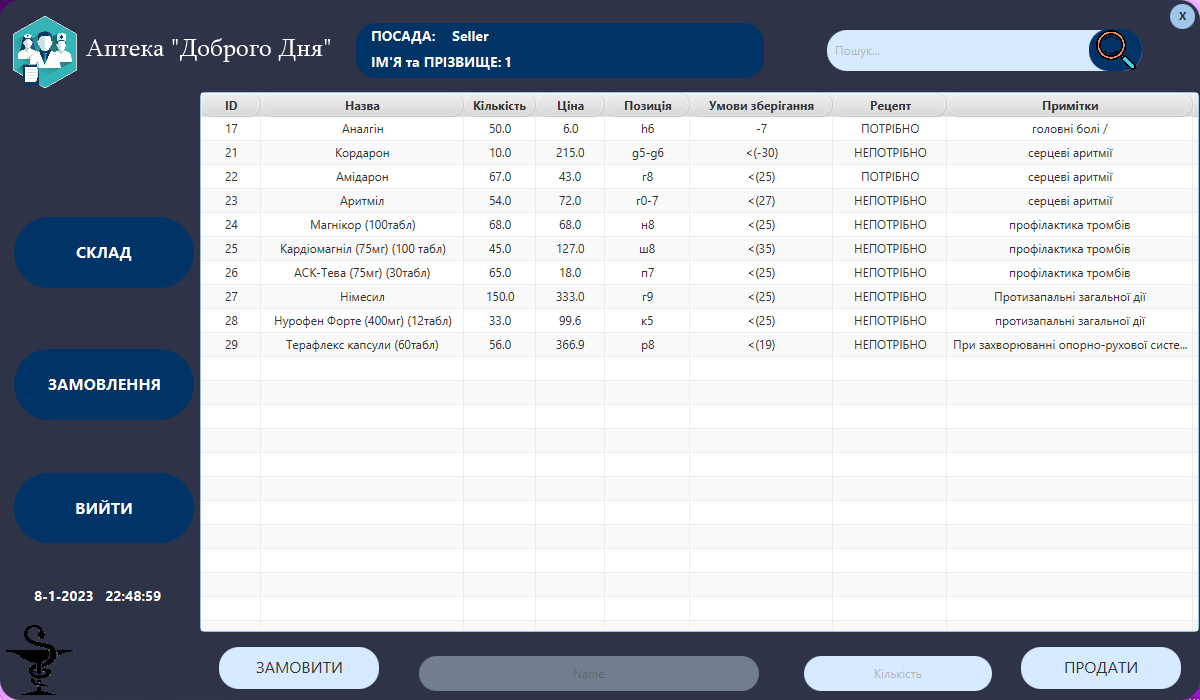
****

Логін та пароль головного Адміністратора: (Admin / Admin45);

## **АКАУНТ ПРОДАВЦЯ**

### **«СКЛАД»**

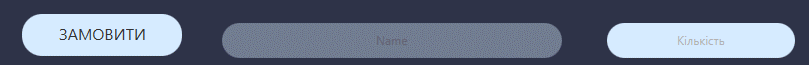
Після входу на екрані відображається склад аптеки, де продавець має можливість :

****

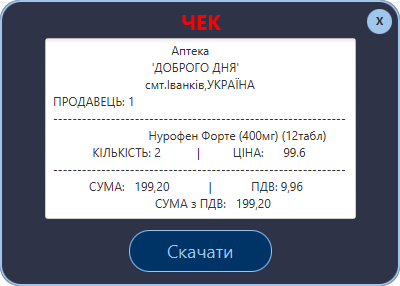
* Обрати препарат;



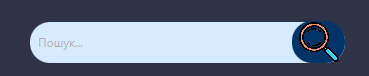
* Замовити певну кількість товару;



* Продати певну кількість та скачати чек:



* Швидко знайти препарат за унікальним кодом, назвою, позицією або додатками;



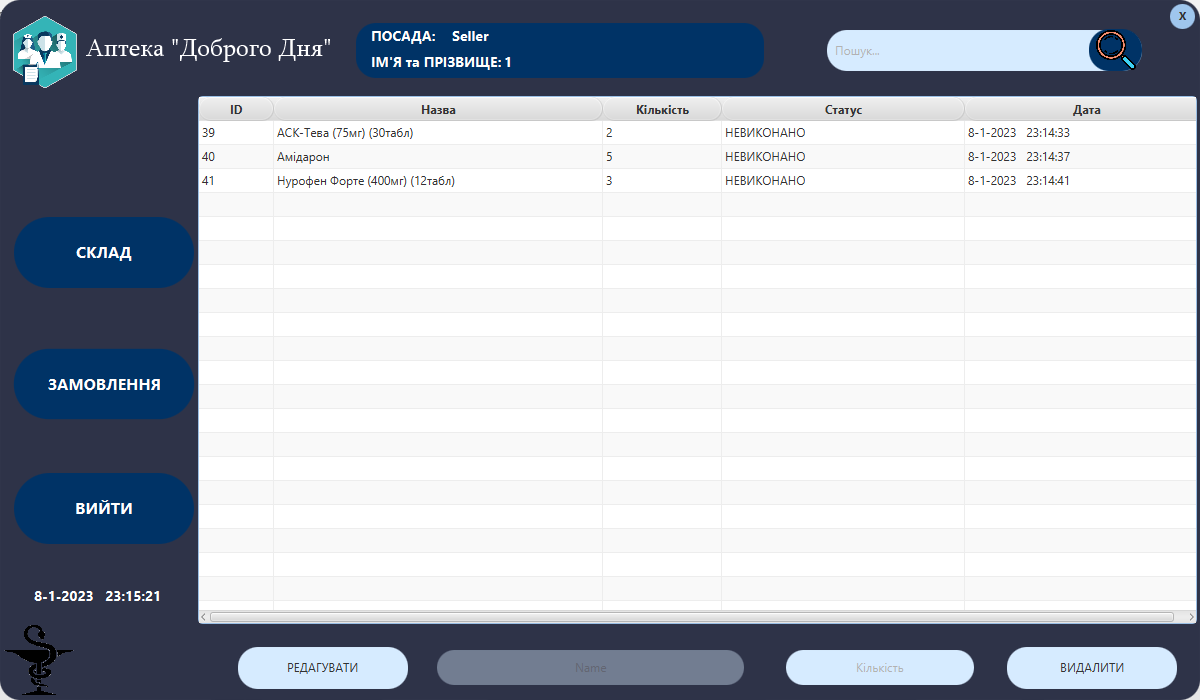
* Отримати інформацію про дані свого акаунту;



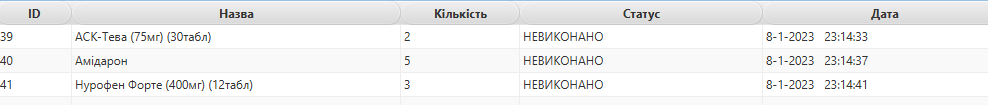
* Після натискання клавіші «СКЛАД», відбувається перезагрузка, якщо ви знаходитесь в даному вікні.

### **«ЗАМОВЛЕННЯ»**

Після натиснення клавіші «Замовлення», відбувається відкриття таблиці замовлень та відображаються замовлення, які створені з даного акаунта продавця;



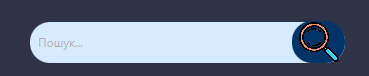
* Обрати замовлення;



* Відредагувати або видалити замовлення;



* Швидко знайти замовлення за унікальним кодом або назвою;



* Отримати інформацію про дані свого акаунта;



* Після натискання клавіші «ЗАМОВЛЕННЯ», відбувається перезагрузка, якщо ви знаходитесь в даному вікні.

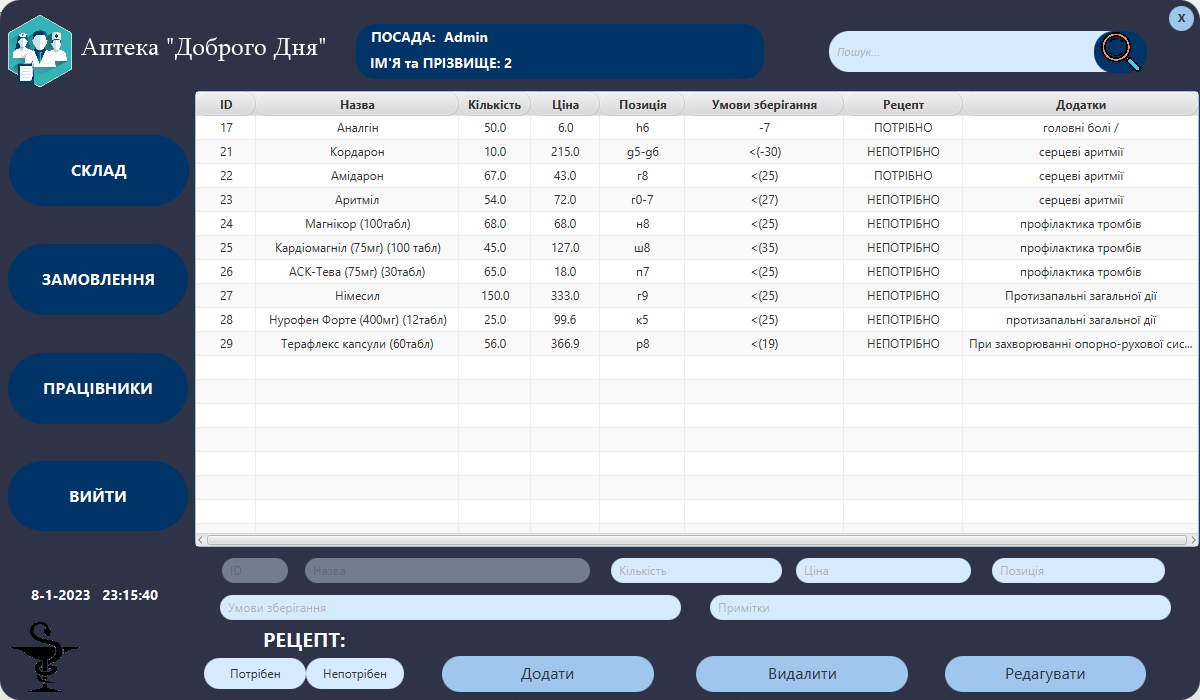
**«ВИЙТИ»**

Після натиснення клавіші «Вийти», відбувається вихід з акаунт та повернення до вікна входу;

**АКАУНТ АДМІНІСТРАТОРА**

### **«СКЛАД»**

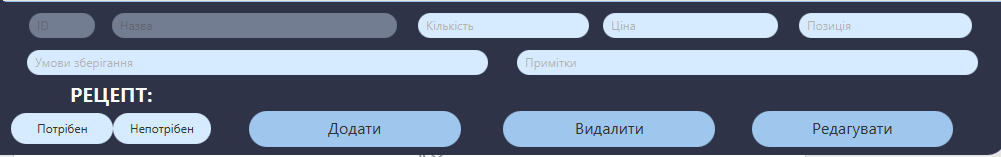
Після входу на екрані відображається склад аптеки, де адміністратор має можливість :



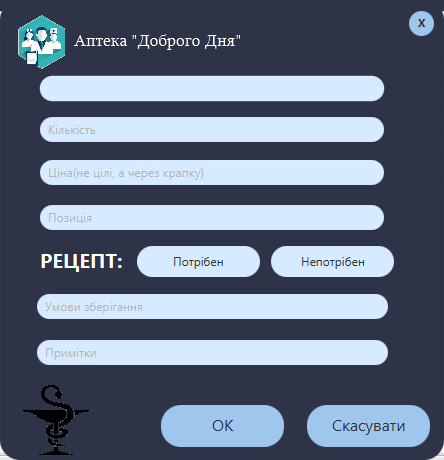
* Обрати препарат;



* Додати, Видалити, Редагувати препарат;

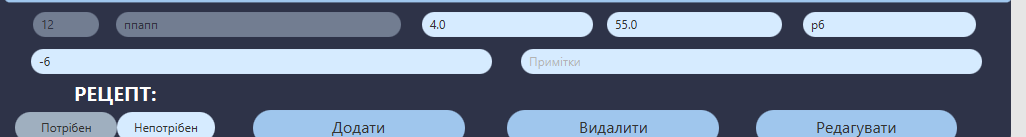


ДОДАТИ (відкривається нове вікно): Щоб додати новий товар, потрібно ввести значення та натиснути «ОК», якщо відмінити «Скасувати»:

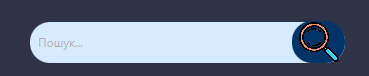


ВИДАЛИТИ: Вибрати препарат та натиснути «ВИДАЛИТИ»;

РЕДАГУВАТИ: Вибрати препарат та змінити необхідні дані, які з’явилися в полях:



* Швидко знайти препарат за унікальним кодом, назвою, позицією або додатками;



* Отримати інформацію про дані свого акаунту;



* Після натискання клавіші «СКЛАД», відбувається перезагрузка, якщо ви знаходитесь в даному вікні.

### **«ЗАМОВЛЕННЯ»**

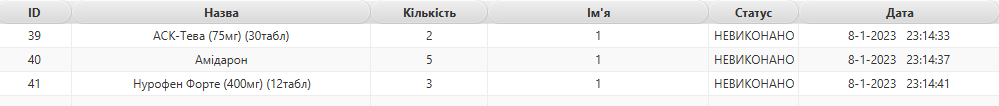
Після натиснення клавіші «Замовлення», відбувається відкриття таблиці замовлень та відображаються замовлення, які створені з даного акаунта продавця;



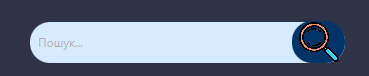
ВИКОНАТИ: Вибрати препарат та натиснути «ВИКОНАТИ», після цього замовлення буде відправлено замовнику;

ВІДМІНИТИ: Вибрати препарат та натиснути «ВІДМІНИТИ», після цього замовлення не буде відправлено замовнику;

* Обрати замовлення;



* Швидко знайти замовлення за унікальним кодом або назвою;



* Отримати інформацію про дані свого акаунта;



* Після натискання клавіші «ЗАМОВЛЕННЯ», відбувається перезагрузка, якщо ви знаходитесь в даному вікні.

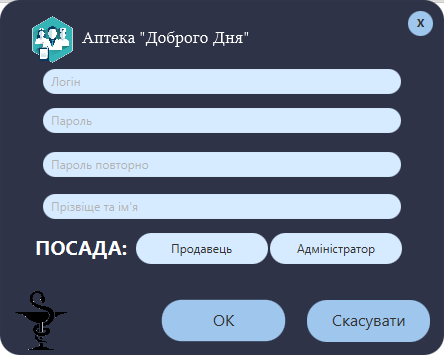
Після натиснення клавіші «Вийти», відбувається вихід з акаунт та повернення до вікна входу;

### **«ПРАЦІВНИКИ»**

Після натиснення клавіші «ПРАЦІВНИКИ», відбувається відкриття таблиці замовлень та відображаються замовлення, які створені з даного акаунта продавця;

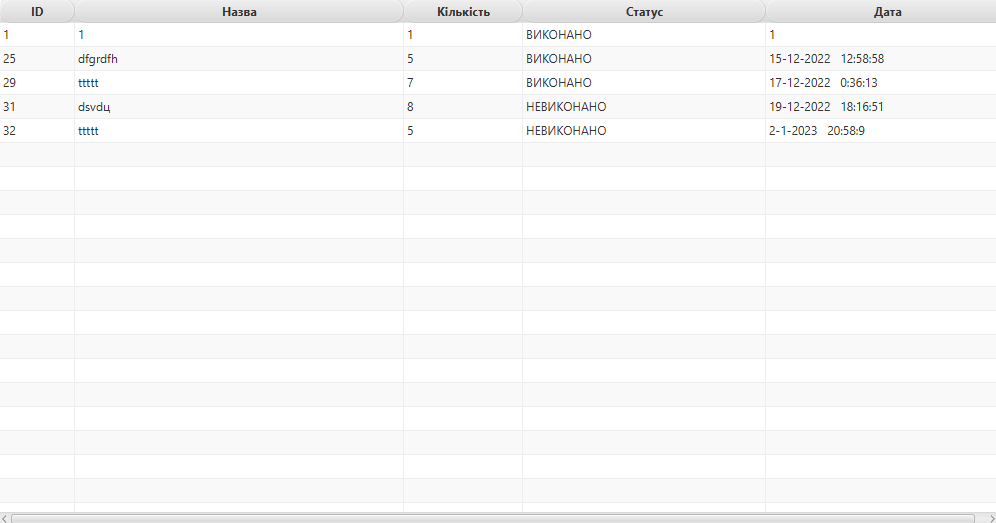


ДОДАТИ (відкривається нове вікно): Щоб додати новий товар, потрібно ввести значення та натиснути «ОК», якщо відмінити «Скасувати»:

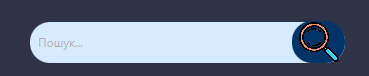


ВИДАЛИТИ: Вибрати працівника та натиснути «ВИДАЛИТИ», після цього дані про працівника буде видалено з програми;

* Обрати замовлення;



* Швидко знайти замовлення за унікальним кодом або назвою;



* Отримати інформацію про дані свого акаунта;



* Після натискання клавіші «ПРАЦІВНИКИ», відбувається перезагрузка, якщо ви знаходитесь в даному вікні.

### **«ВИЙТИ»**

Після натиснення клавіші «Вийти», відбувається вихід з акаунт та повернення до вікна входу;

**ВИСНОВОК**

В процесі виконання даної курсової роботи, розробили програму для керування складом медичних препаратів. Робота полягала у створення програми для полегшення роботи працівників складу аптеки.

Згідно з вимогами замовника , програма повинна містити інформацію про препарат (назву, ціну, кількість, умови зберігання, розташування на складі, потрібність рецепту для продажу, додатки); можливість додавати, редагувати та видаляти препарати; швидкий пошук; також реалізовано можливість створювати замовлення та редагувати їх.

Було розрозблено методи обробки вийняткових ситуацій, що дозволить зменшити кількість непередбачуваних порушень в роботі програми.

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

**1.** Технология разработки программных продуктов. Практикум: учеб.

пособие для студ. учреждений сред. проф. образования / А. В.Рудаков, Г. Н. Федорова. — 4-е изд., стер. — М. : Издательский центр «Академия»; 2014. — 192 с.

**2.** Конспект лекцій з дисципліни «Аналіз вимог до програмного забезпечення» для студентів напряму підготовки 6.050103 «Програмна інженерія». Укладач: канд. фіз.-мат. наук, доцент Божуха Л.М. Дніпродзержинськ: ДДТУ, 2015. 94 с.

**3.** Gamma E., Helm R., Johnson R. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Boston, Addison-Wesley Professional, 1994. 416 p.

**4.** Технології програмування та створення програмних продуктів: конспект лекцій /укладач О. В. Алексенко. Суми, Сумський державний університет, 2013. 133 с.

**5.** Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol, O'Reilly Media, 2020. 432 p.

**6.** Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship. London, Pearson, 2008. 464 p.

**7.** Martin R.C. The Clean Coder: A Code of Conduct for Professional Programmers. London, Pearson, 2011. 256 p.

**8.** Карпенко М.Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем : навч. Посібник. Харків, ХНУМГ ім. О. М. Бекетова, 2017. 93 с.