Міністерство освіти і науки України

Державний університет «Одеська політехніка»

Інститут комп’ютерних систем  
Кафедра інформаційних систем

**КУРСОВА РОБОТА**

з дисципліни «Технології створення програмних продуктів»

за темою

Інформаційна системи управління складовими процесами поштової та кур’єрської служби «Skladobase»

Пояснювальна записка до етапів визначення вимог до програмного продукту та планування процесів розробки програмного продукту

Виконали:

студенти 3-го курсу

групи АІ-194

Буряченко О.О.

Оганесян М.А.

Перевірив:

Блажко О. А.

Одеса-2021

**Анотація**

В курсовій роботі розглядається процес створення програмного продукту «Інформаційна системи управління складовими процесами поштової та кур’єрської служби «Skladobase»» на етапах визначення вимог до програмного продукту та планування процесів розробки.

Робота виконувалась в команді з декількох учасників: Буряченко О.О.

Оганесян М.А.

В робота пов`язана з такими матеріальними потребами споживача як необхідність швидко шукати ресурси, ефективно взаємодіяти зі списками збереженеих ресурсів . Аналіз вказаних потреб визначив інформаційну потребу - в актуальних даних про розташування і кількість необхідних ресурсів, зменшенні кількості помилок при обліку ресурсів.

При визначені ступеня готовності існуючих програмних продуктів до вирішення інформаційної потреби проаналізовано наступні програмні продукти:

* «1С: Торгівля і склад»
* ERP «Галактика»

Поточну версію пояснювальної записки до результатів роботи розміщено на *GitHub*-репозиторії за адресою: https://github.com/Alehandrissimus/skladobase

**Перелік скорочень**

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП– програмний продукт

UML – уніфікована мова моделювання

СУ – система управління

**Зміст**

|  |  |
| --- | --- |
|  | стор. |
| 1 Вимоги до програмного продукту | 6 |
| 1.1 Визначення потреб споживача |  |
| 1.1.1 Ієрархія потреб споживача |  |
| 1.1.2 Деталізація матеріальної потреби |  |
| 1.2 Бізнес-вимоги до програмного продукту |  |
| 1.2.1 Опис проблеми споживача |  |
| 1.2.1.1 Концептуальний опис проблеми споживача |  |
| 1.2.1.2 Опис цільової групи споживача |  |
| 1.2.1.3 Метричний опис проблеми споживача |  |
| 1.2.2 Мета створення програмного продукту |  |
| 1.2.2.1 Проблемний аналіз існуючих програмних продуктів |  |
| 1.2.2.2 Мета створення програмного продукту |  |
| 1.2.3 Назва програмного продукту |  |
| 1.2.3.1 Гасло програмного продукту |  |
| 1.2.3.2 Логотип програмного продукту |  |
| 1.3 Вимоги користувача до програмного продукту |  |
| 1.3.1 Історія користувача програмного продукту |  |
| 1.3.2 Діаграма прецедентів програмного продукту |  |
| 1.3.3 Сценаріїв використання прецедентів програмного продукту |  |
| 1.4 Функціональні вимоги до програмного продукту |  |
| 1.4.1. Багаторівнева класифікація функціональних вимог |  |
| 1.4.2 Функціональний аналіз існуючих програмних продуктів |  |
| 1.5 Нефункціональні вимоги до програмного продукту |  |
| 1.5.1 Опис зовнішніх інтерфейсів |  |
| 1.5.1.1 Опис інтерфейсів користувача |  |
| 1.5.1.1.1 Опис INPUT-інтерфейсів користувача |  |
| 1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача |  |
| 1.5.1.2 Опис інтерфейсу із зовнішніми пристроями |  |
| 1.5.1.3 Опис програмних інтерфейсів |  |
| 1.5.1.4 Опис інтерфейсів передачі інформації |  |
| 1.5.1.5 Опис атрибутів продуктивності |  |
| 2 Планування процесу розробки програмного продукту |  |
| 2.1 Планування ітерацій розробки програмного продукту |  |
| 2.2 Концептуальний опис архітектури програмного продукту |  |
| 2.3 План розробки програмного продукту |  |
| 2.3.1 Оцінка трудомісткості розробки програмного продукту |  |
| 2.3.2 Визначення дерева робіт з розробки програмного продукту |  |
| 2.3.3 Графік робіт з розробки програмного продукту |  |
| 2.3.3.1 Таблиця з графіком робіт |  |
| 2.3.3.2 Діаграма Ганта |  |
| Висновки до курсової роботи |  |

**1 Вимоги до програмного продукту**

**1.1 Визначення потреб споживача**

* + 1. **Ієрархія потреб споживача**

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

* фізіологія (вода, їжа, житло, сон);
* безпека (особиста, здоров'я, стабільність),
* приналежність (спілкування, дружба, любов),
* визнання (повага оточуючих, самооцінка),
* самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено одну ієрархію потреби споживача, яку хотілося б задовольнити, використовуючи майбутній програмний продукт.



Рис. 1.1 – Приклад ієрархії потреби споживача

1.1.2 Деталізація матеріальної потреби

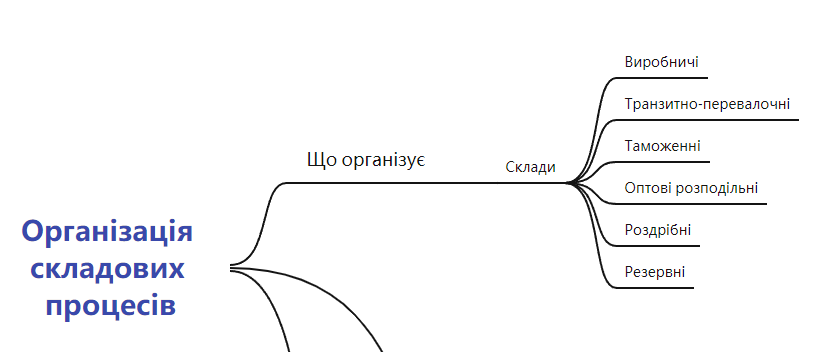


Рис 1.2 – Гілка Mindmap-дерева

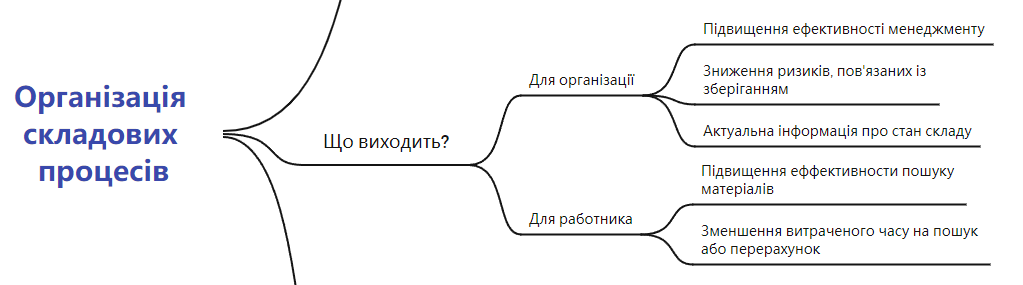


Рис 1.3 – Гілка Mindmap-дерева



Рис 1.4 - Гілка Mindmap-дерева

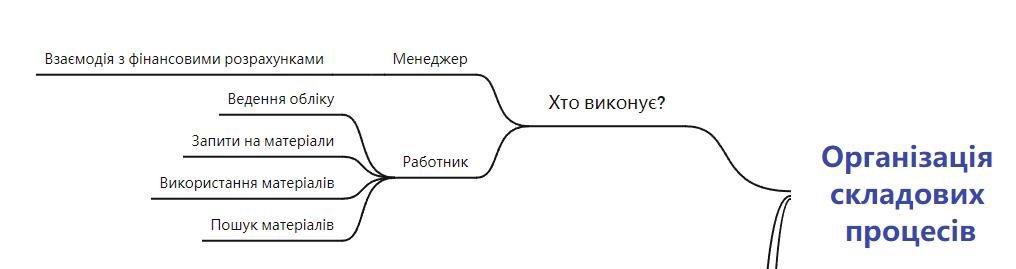


Рис 1.5 – Гілка Mindmap-дерева

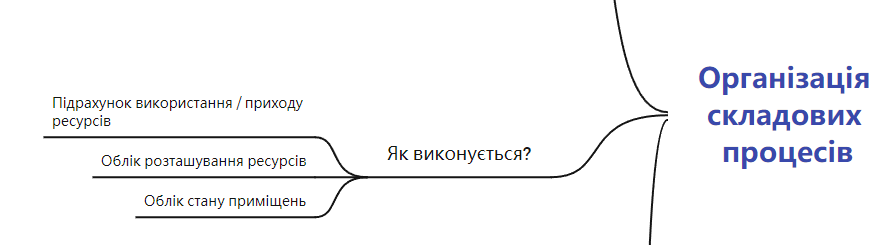


Рис 1.6 – Гілка Mindmap-дерева

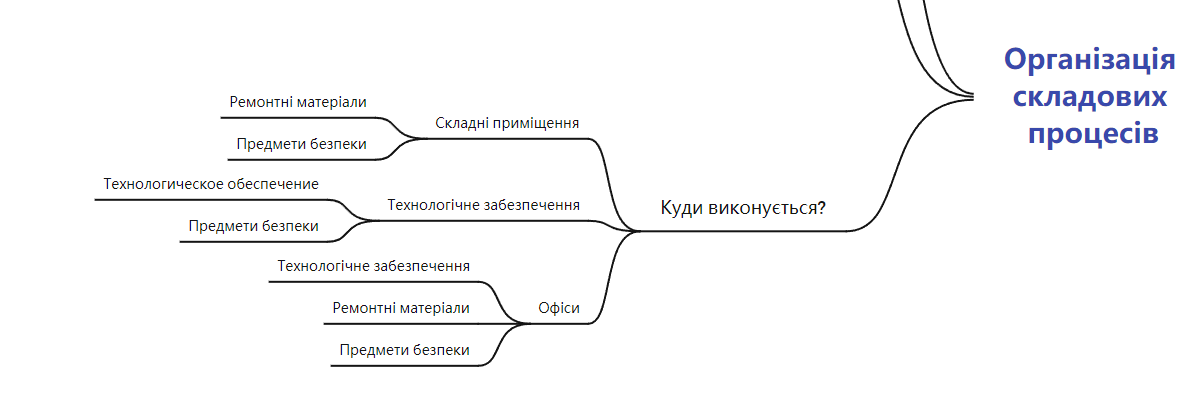


Рис 1.7 – Гілка Mindmap-дерева

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.1 Концептуальний опис проблеми споживача

Велика частина складів, орієнтованих на зберігання техніки та експлуатаційних матеріалів компанії, зовсім не автоматизовані, а працездатність і ефективність даного типу складів досить важлива для компанії. Зі збільшенням кількості ресурсів на складі пошук цих ресурсів стає все скрутнішим і контроль кількості ресурсів в наявності відповідно, теж. Витрачається велика кількість часу на пошук ресурсів візуально і виростає кількість помилок в підрахунку загальної кількості ресурсів.

1.2.1.2 Опис цільової групи споживачів

Цільова група споживачів нашого программного продукту – це будь-які підприємства, які взаємодіють з підконтрольним ними складом. Але безпосередньо з програмним продуктом взаємодіють 2 основні групи людей: робітники і менеджери. Програмний продукт не має гендерних особливостей. Для ефективної роботи з продуктом, безпосередньому споживачу необхідно мати мінімальні навички в сфері взаємодії з програмними інтерфейсами і комп'ютерним забезпеченням.

1.2.1.3 Метричний опис проблеми споживача

Нехай мінімальна швидкісна ефективність взаємодії зі складом буде визначатися як:

Де EF - коефіцієнт швидкості взаємодії зі складом без СУ в порівнянні з ситуацією з використанням СУ, А – довжина складу, В – ширина складу, N – кількість стелажів.

Або

Де Smax1 – максимальний шлях від інтерфейсу до ресурсу з використанням СУ, Smax2 - максимальний шлях від інтерфейсу до ресурсу без використання СУ.

При використанні СУ на маленьких складах (Н <1-3), коефіцієнт буде > 0.6 і використання СУ буде нераціонально.

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

Були розглянуті слідуючі WPS – системи:

Впровадження і використання даних систем вимагає значних вкладень, так як необхідно додаткове обладнання, купувати ліцензії, наймати додатковий персонал в штаб для обслуговування цих систем, що тягне за собою вагомі фінансові витрати.

1.2.2.2 Мета створення програмного продукту

Підвищити легкість пошуку, виявлення та оновлення інформації про ресурси на складі за допомогою автоматизованої інформаційної системи управління логістичними процесами поштової та кур’єрської служби, полегшити орієнтування персоналу на складі.

1.2.3 Назва програмного продукту

1.2.3.1 Гасло програмного продукту

Назва: SKLADOBASE Perfomance Solutions

1.2.3.2 Логотип програмного продукту



Рис. 1.8– Логотип програмного продукту

1.3 Вимоги користувача до програмного продукту

1.3.1 Історія користувача програмного продукту

* Як працівник складу, я хочу мати доступ до інформації про розташування необхідних мені ресурсів складу.
* Як працівник складу, я хочу мати можливість вносити даний в систему управління про прихід або догляд тих чи інших ресурсів.
* Як технічний фахівець, я хочу мати можливість змінювати і модифікувати складові системи управління.
* Як відповідальний за фінанси, я хочу мати доступ до всіх фінансових розрахунків.
* Як відповідальний за фінанси, я хочу мати можливість при необхідності змінювати умови фінансових розрахунків.

1.3.2 Діаграма прецедентів програмного продукту

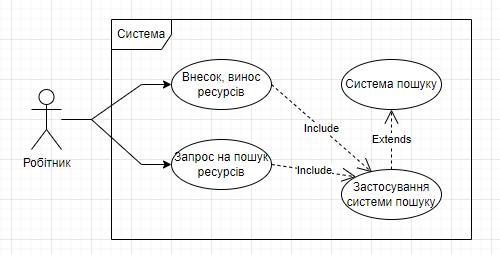


Рис 1.9 - Діаграма прецедентів програмного продукту

1.3.3 Сценарії використання прецедентів програмного продукту

1. «Внеcок, винос ресурсів»

Умова початку: дія з боку Робітника

Зацікавлені актори: Робітник

Гарантії успіху: оновлена інформація про ресурси на складі

Успішний сценарій:

1. Робітник заповнює форму на внесення/винесення
2. ПП робить зміни
3. ПП повертає повідомлення про успіх

Альтернативний сценарій:

2.1 ПП виявляє помилку в процесі валідації або обробки запиту

2.2 ПП повертає повідомлення і переходить до кроку 1 основного успішного сценарію

1. «Запит на пошук ресурсів»

Умова початку прецеденту: дія с боку Робітника

Актори що зацікалвенни в виконанні даного прецеденту: Робітник

Гарантії успіху: Робітник получив інформацію о необхідних для нього ресурсів

Успішний сценарій:

1. Робітник заповнює форму на пошук ресурсу
2. ПП шукає необхідні ресурси
3. ПП передає інформацію о знаходженні ресурсу у вигляді графічного позначення на схемі складу
4. Робітник забирає ресурс

Альтернативный сценарій:

2.1 ПП виявляє помилку в процесі валідації або обробки запиту

2.2 ПП повертає повідомлення і переходить до кроку 1 основного успішного сценарію

1. «Отримання фінансових данних»

Умова початку прецеденту: дія з боку Менеджера

Акторі, що зацікалвенни в виконанні даного прецеденту: Менеджер

Гарантії успіху: менеджер получив необхідні данні

Успішний сценарій:

1. Менеджер вибирає необхідний фінансовий шаблон
2. ПП знаходить шаблон фінансового разразунку та проводить необхідні обчислення
3. ПП виводить фінансову інформацію у вигляді ексель-таблиці
4. Менеджер отримує необхідну інформацію

Альтернативний сценарій:

2.1 ПП виявляє помилку в процесі валідації або обробки запиту

2.2 ПП повертає повідомлення і переходить до кроку 1 основного успішного сценарію

1. «Модифікація фінансових розрахунків»

Умова початку прецеденту: дія з боку Менеджера

Акторі, що зацікалвенни в виконанні даного прецеденту: Менеджер

Гарантії успіху: успішна зміна фінансових даних

Успішний сценарій:

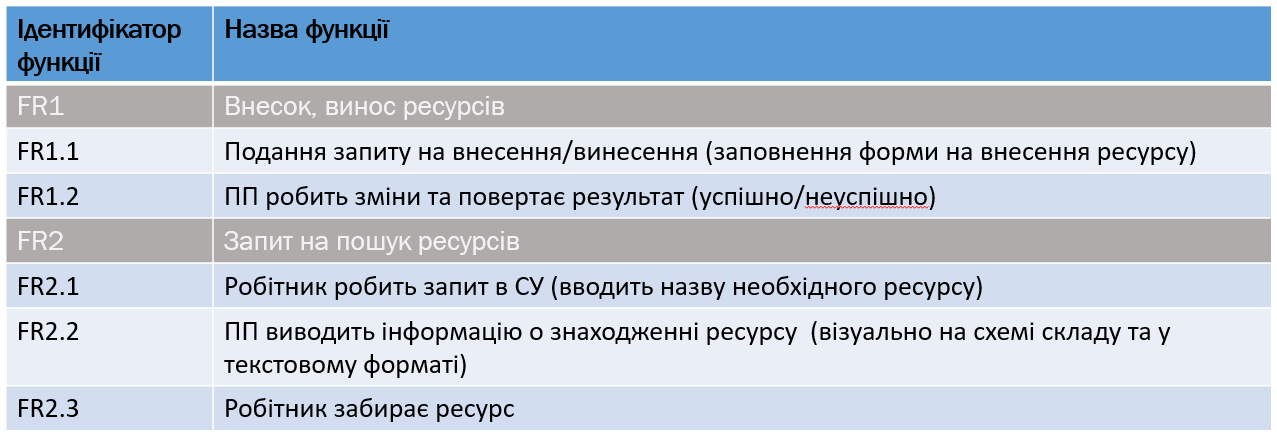
1. Менеджер вибирає необхідну фінансову операцію
2. Менеджер змінює фінансові змінні
3. ПП зберігає заредактований фінансовий шаблон
4. ПП повертає повідомлення про успіх

Альтернативний сценарій:

3.1 ПП виявляє помилку в процесі валідації або зберігання

3.2 ПП повертає повідомлення і переходить до кроку 1 основного успішного сценарію

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

1.4.2 Функціональний аналіз існуючих програмних продуктів

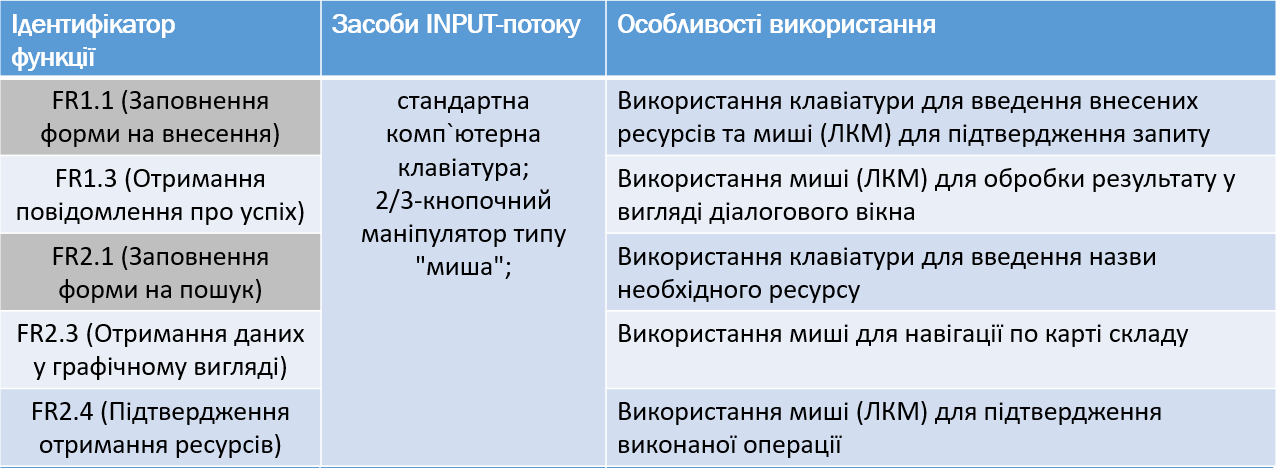
1.5 Нефункціональні вимоги до програмного продукту



1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейсів користувача

1.5.1.1.1 Опис INPUT-інтерфейсів користувача



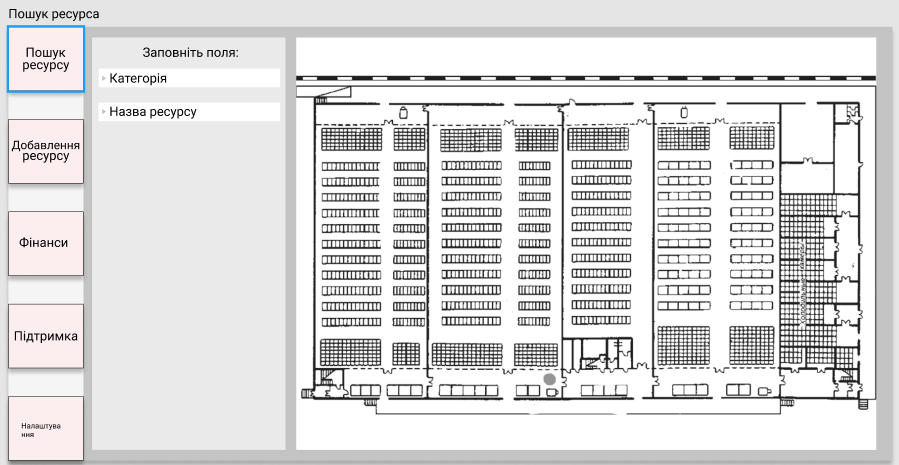
1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача

Рис 1.11 - OUTPUT-потік функції «Внесок, винос ресурсів»

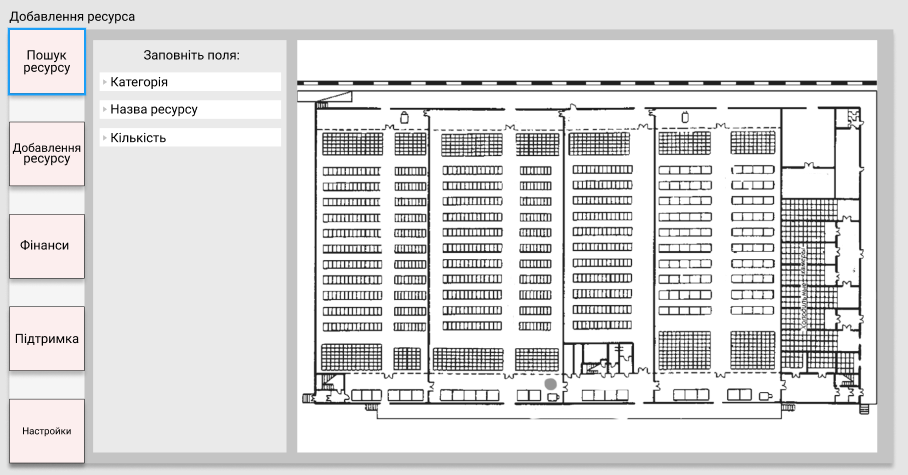
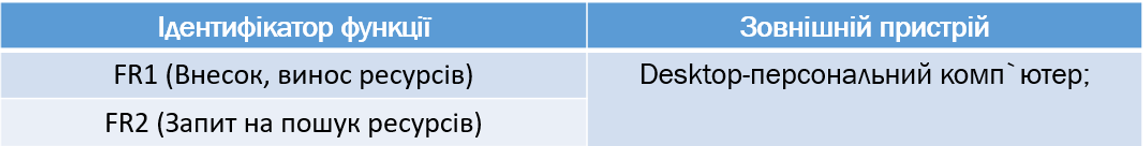


Рис 1.12 - OUTPUT-потік функції «Запит на пошук ресурсів»

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

1.5.1.3 Опис програмних інтерфейсів

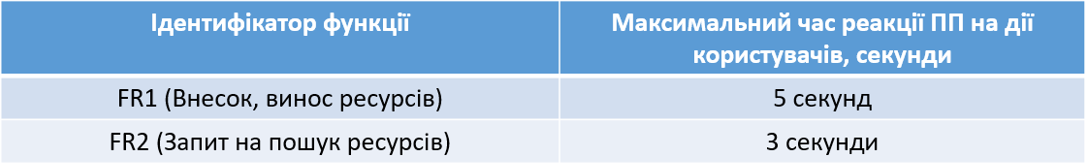
Версії операційних систем та програмних бібліотек, які знадобляться при реалізації більшості функцій ПП.

* Windows/Linux
* JDBC
* PostgreSQL

1.5.1.4 Опис інтерфейсів передачі інформації

Інтерфейси передачі інформації, які знадобляться при реалізації більшості функцій ПП.

* Провідні інтерфейси: Enthernet
* Безпровідні інтерфейси : Wi-fi

1.5.1.5 Опис атрибутів продуктивності

**2 Планування процесу розробки програмного продукту**

**2.1 Планування ітерацій розробки програмного продукту**

З метою забезпечення вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, визначено функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП. Результати представлено в таблиці 2.1

Таблиця 2.1 – опис функцій з наданням унікальних ієрархічних ідентифікаторів

2.2 Концептуальний опис архітектури програмного продукту

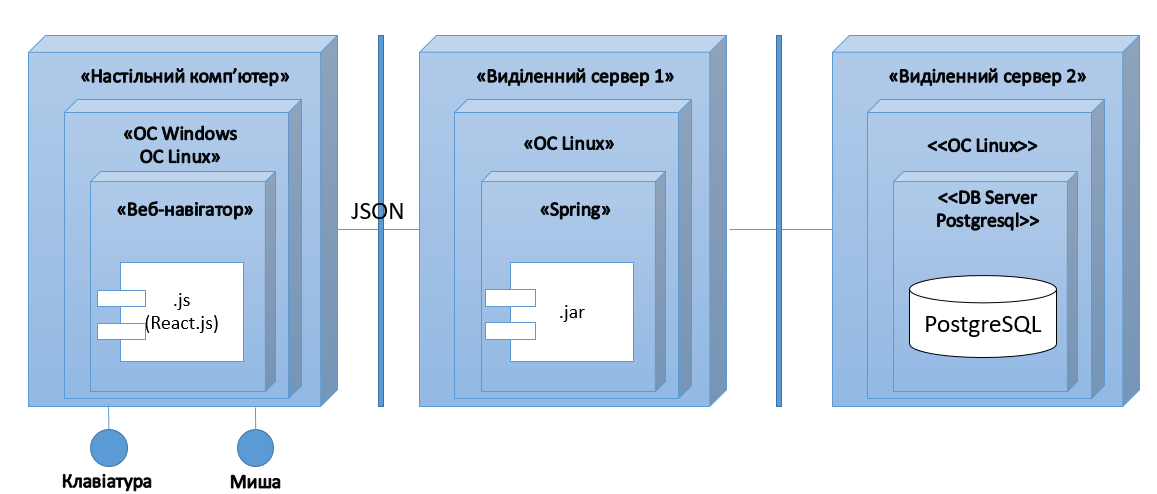
ПП є full-stack додатоком з усіма рівнями: Presentation Level, Business Level, Access Level. Presentation Level - React, Business Level - Java Spring, Access Level – PostgreSQL.

Рис 2.1 - Концептуальний опис архітектури програмного продукту

2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

Визначення нескорегованого показника UUCP:





Визначення UUCP:

UUCP = A + UC

UUCP = (1 + 2) + (15+10+5+5+10)= 48

Визначення технічної складності проекту:



Визначення TCF:

TCF = 0,6 + (0,01 \* (STi \* Вага\_i))

TCF = 0,6 + (0,01 \* (20)) = 0,8

Визначення технічної кваліфікації робітників:



Визначення EF:

EF = 1,4 + (- 0,03 \* (SFi \* Вагаi)) = 0,935

Остаточне значення UCP

UCP = UUCP \* TCF \* EF

UUCP = 48

TCF = 0,8

EF = 0,935

UCP = 35,904

Трудомісткість проекту:

35,904 \* 20 = 718,08 люд.-год

2.3.2 Визначення дерева робіт з розробки програмного продукту

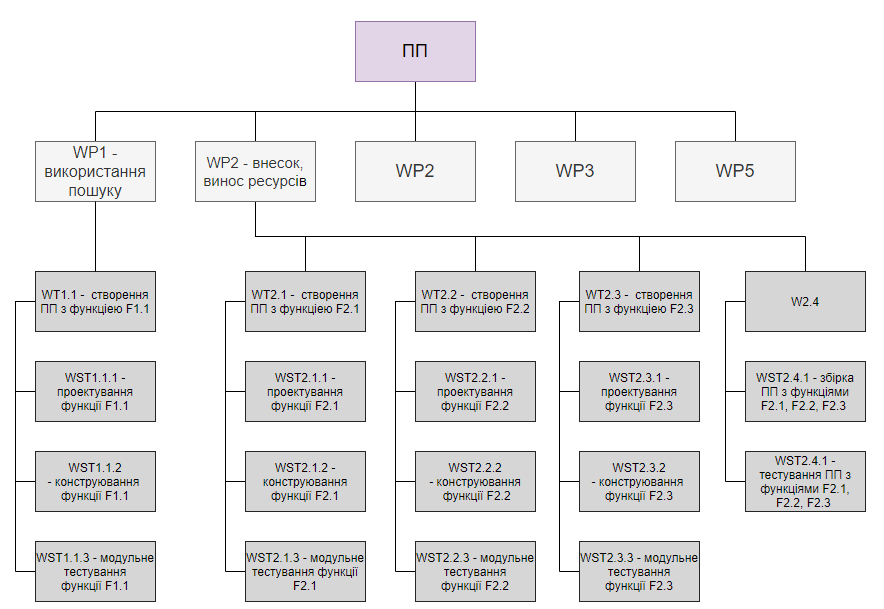


Рис 2.2 – Древо робіт з розробки ПП

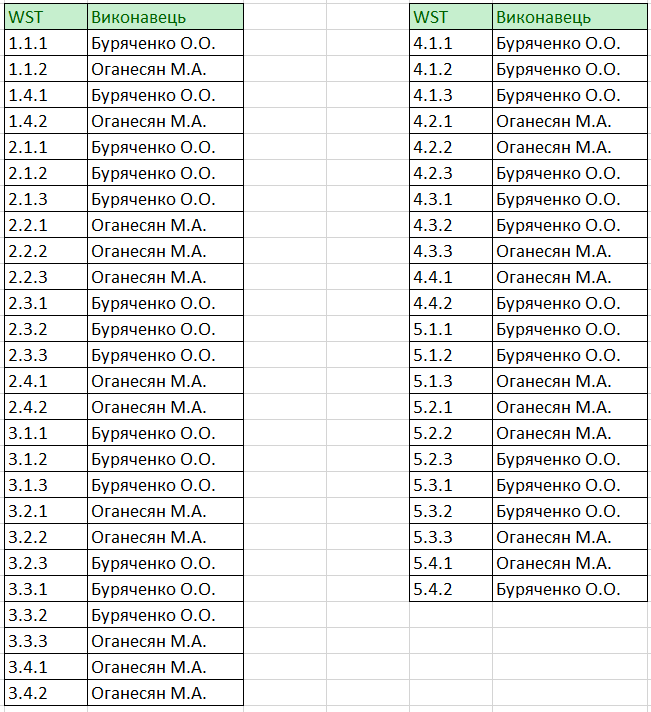


Рис 2.3 – Таблиця з закріпленням виконавців за задачами

2.3.3 Графік робіт з розробки програмного продукту

2.3.3.1 Таблиця з графіком робіт

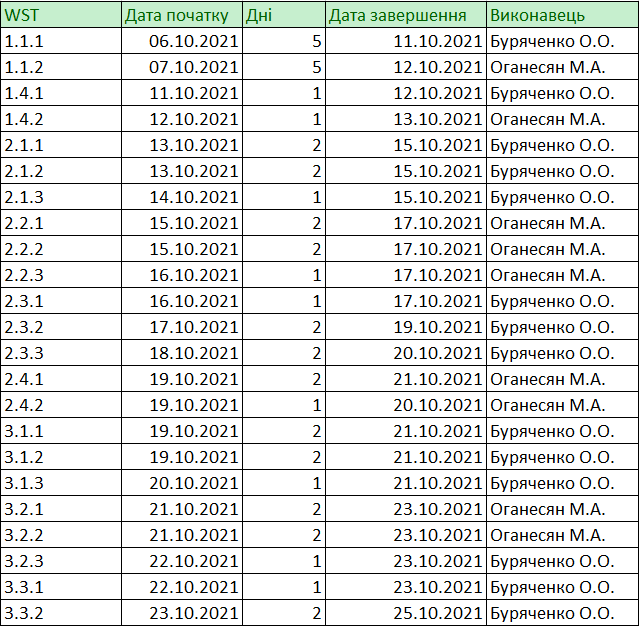


Рис 2.4 – Таблиця з датами старту та закінчення етапів розробки

2.3.3.2 Діаграма Ганта

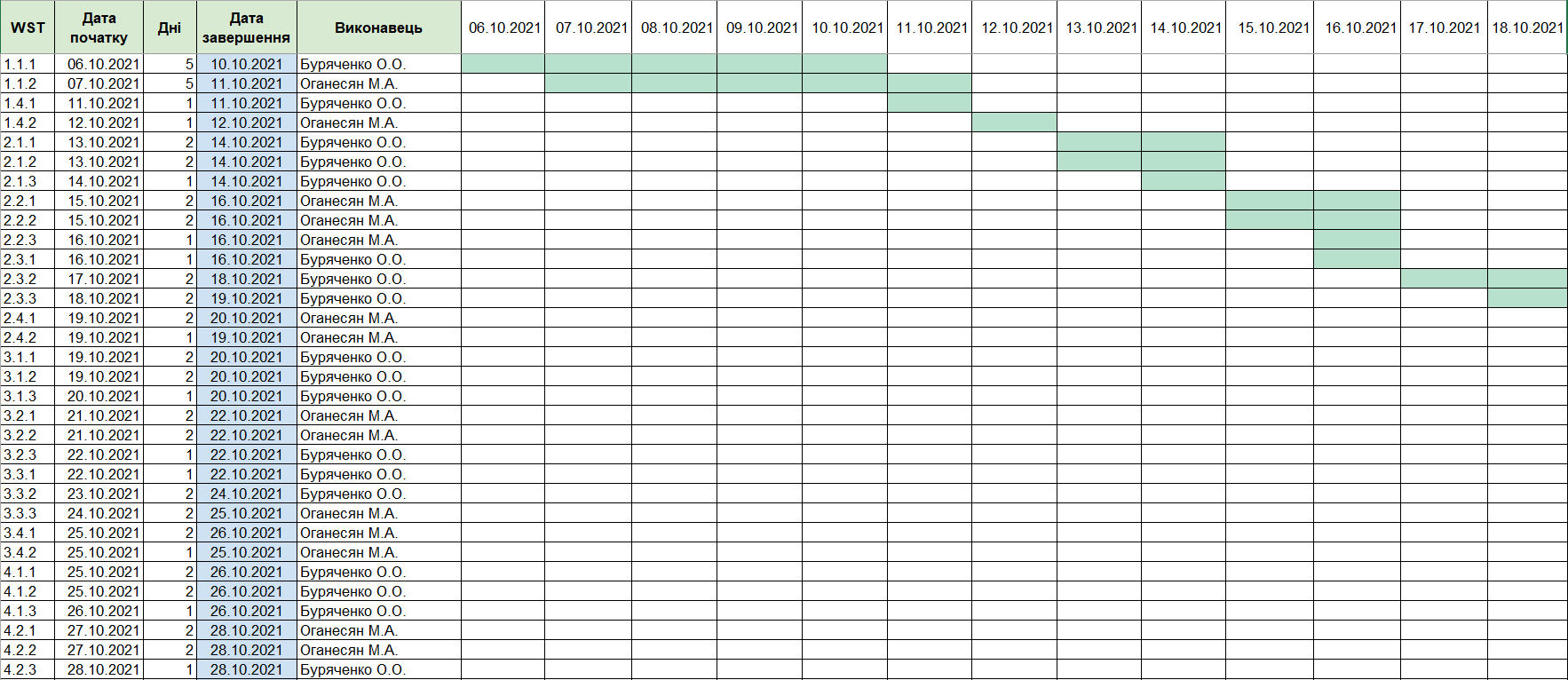


Рис 2.5 – Діаграмма Ганта

Повна діаграма Ганта доступна за посиланням:

<https://docs.google.com/spreadsheets/d/12spCJx0l8pyBhB_pLKgwzR0mc5s36q5d/edit?usp=sharing&ouid=114375752334105036404&rtpof=true&sd=true>

1. **Проектування програмного продукту**

3.1 Концептуальне та логічне проектування структур даних програмного продукту

3.1.1 Концептуальне проектування на основі UML-діаграми концептуальних класів

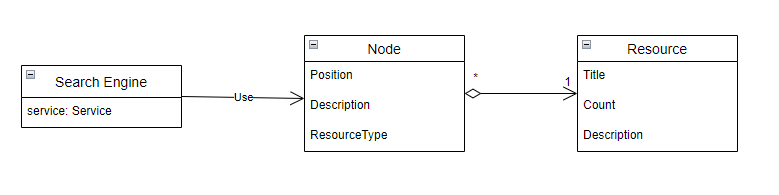


Рис 3.1 – UML-діаграма концептуальних класів

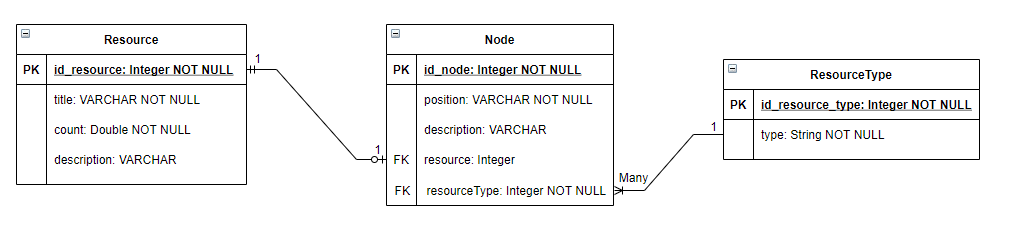
3.1.2 Логічне проектування структур даних

Рис 3.2 – Діаграма структур даних

3.2 Проектування програмних класів

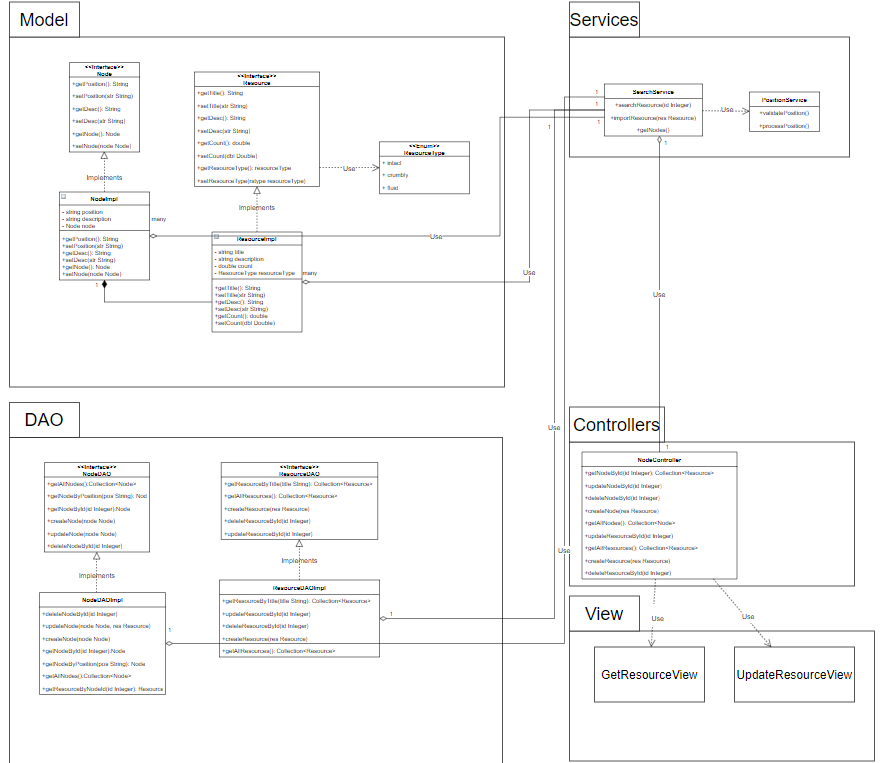


Рис 3.3 – Діаграма програмних класів

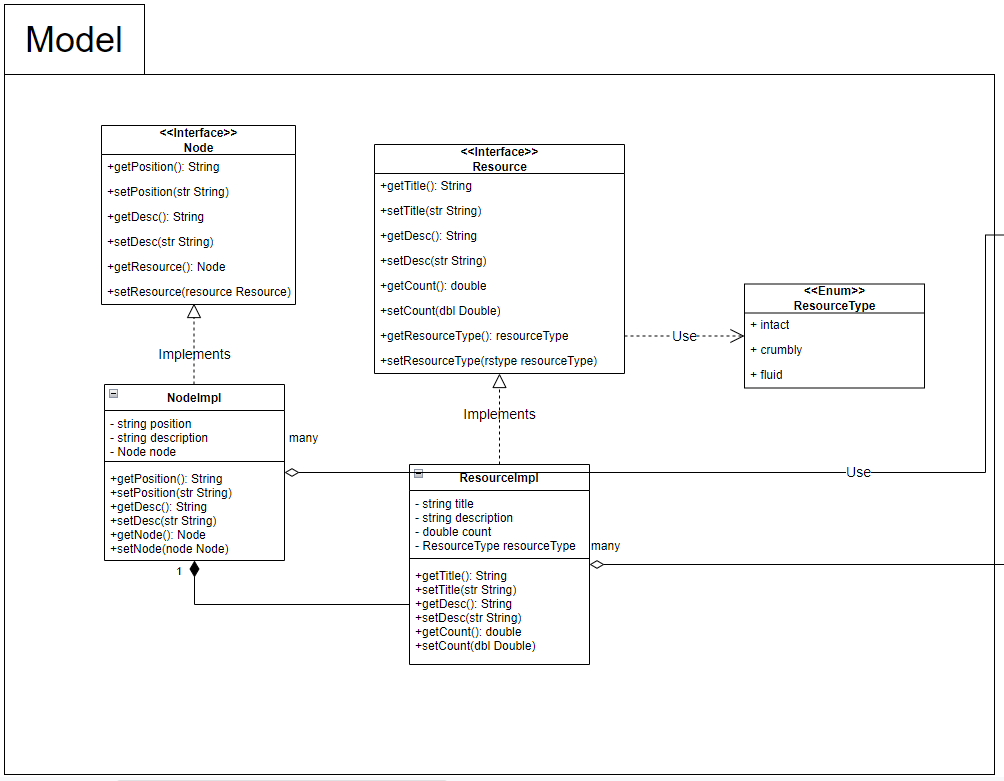


Рис 3.4 – Наближена діаграма програмних класів (Model)

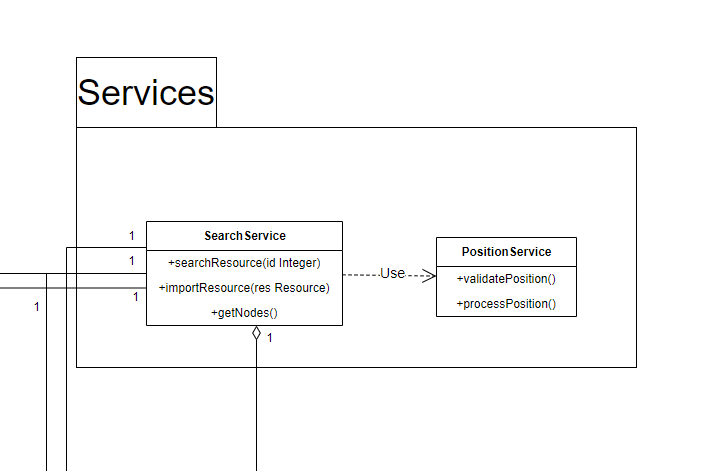


Рис 3.5 – Наближена діаграма програмних класів (Services)

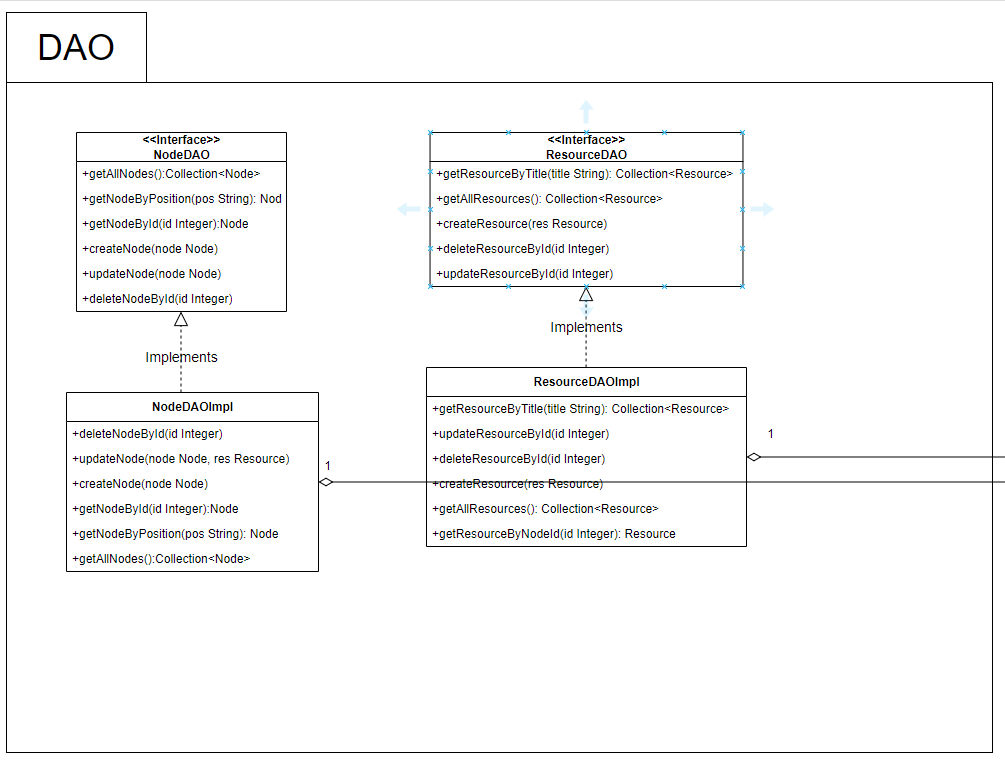


Рис 3.6 – Наближена діаграма програмних класів (DAO)

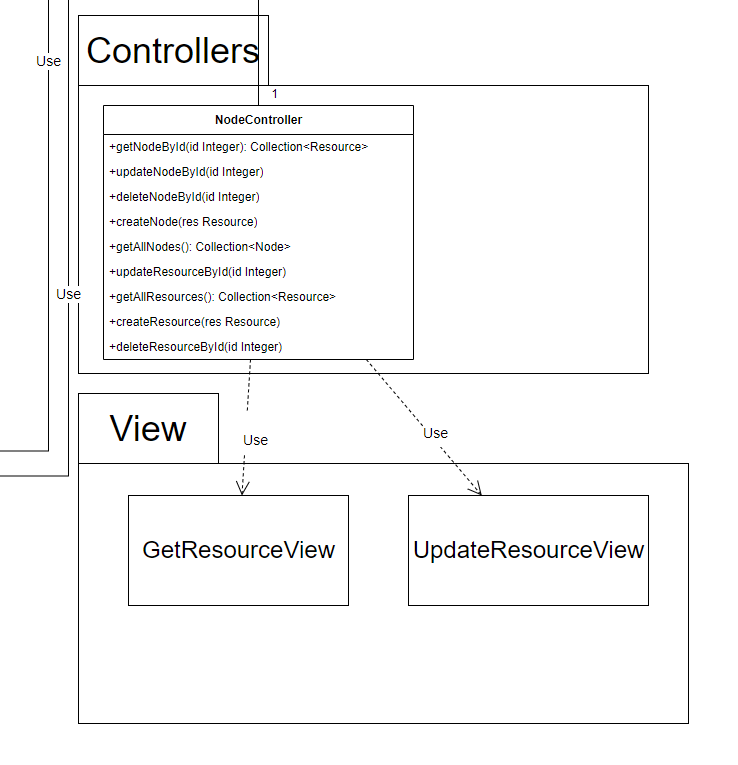


Рис 3.7 – Наближена діаграма програмних класів (Controllers, View)

Діаграма вийшла об'ємною, розглянути її детальніше можна за посиланням:

<https://drive.google.com/file/d/1MQ0GmXezpByrebtcfWHKTe2iedpXX_41/view?usp=sharing>

3.3 Проектування алгоритмів роботи методів програмних класів

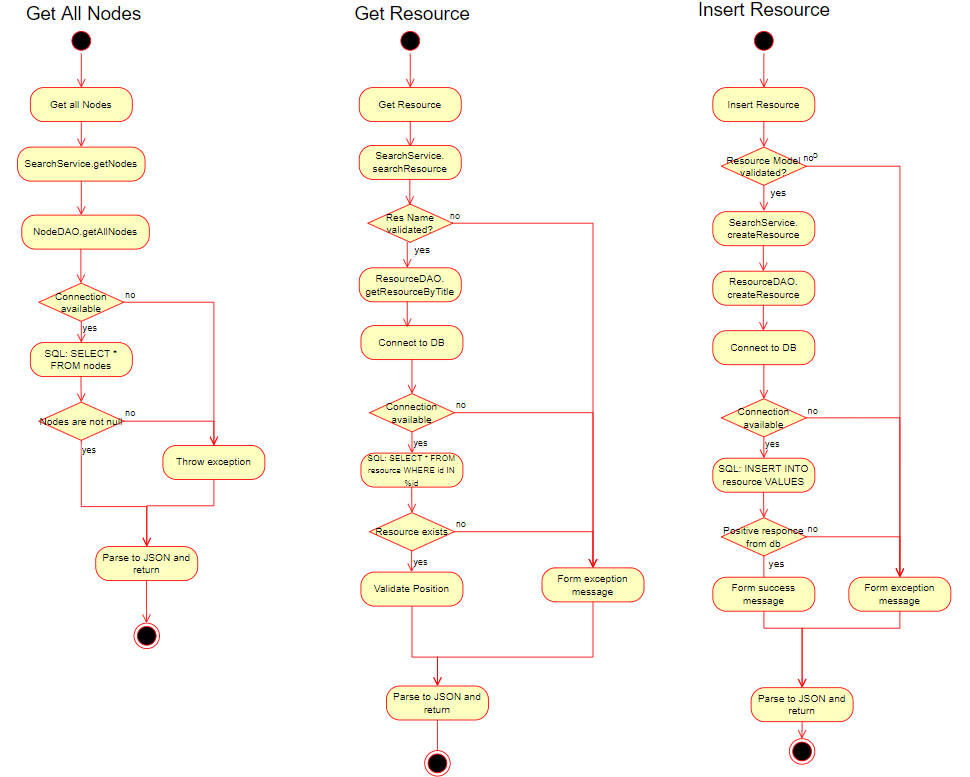


Рис 3.8 – Алгоритми роботи методів

3.4 Проектування тестових сценаріїв верифікації роботи програмних модулів

3.4.1 Проектування тестових сценаріїв верифікації функціональних вимог

При реалізації функціональної вимоги FR1 створено програмний клас NodeController з функцією updateResourceByNode, який має наступну деталізовану специфікацію.

1) Вхідні параметри методу updateResourceByNode :

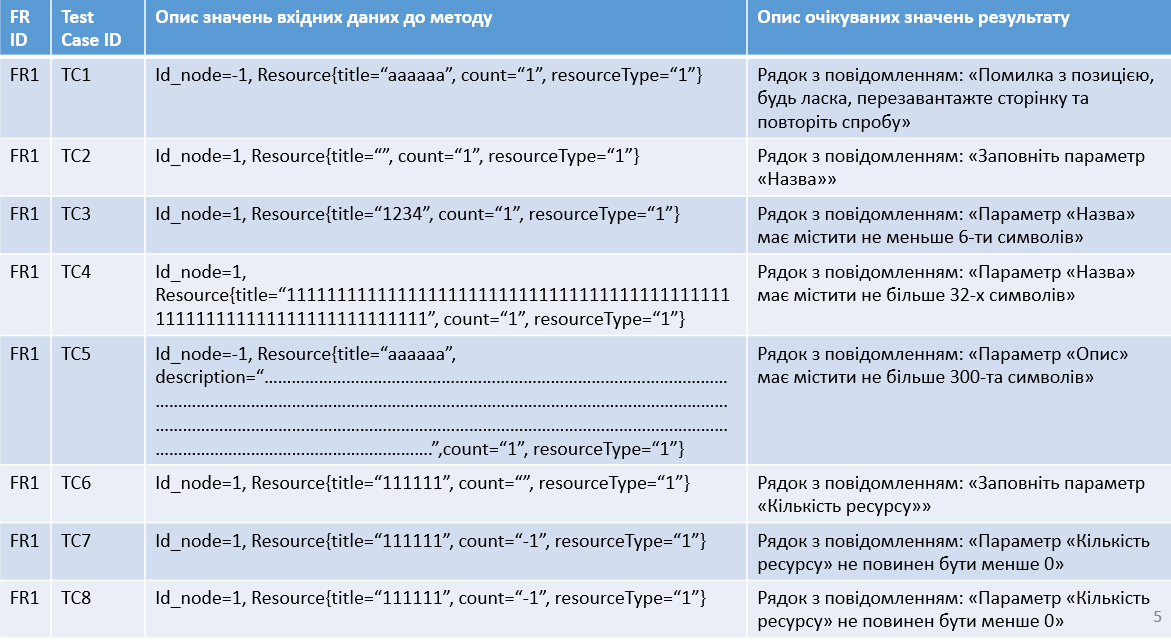
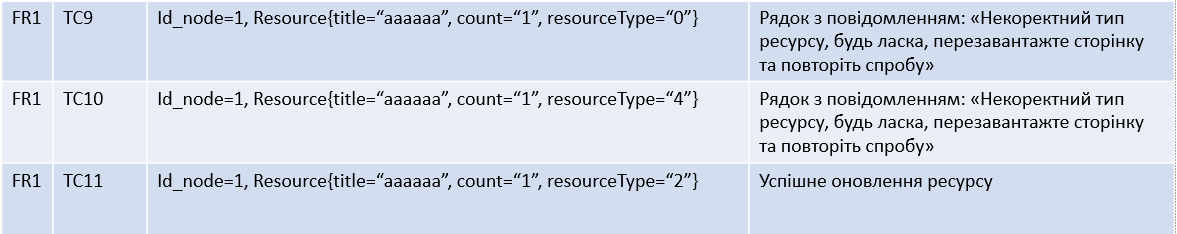
* + Id\_node – порядковий номер Node (Умови: Обов’язковий, Цілочисельний, Не менше 0)
  + Resource – зібрана модель Resource (Умови: Обов’язковий, Не null)
    - Title – Назва ресурсу (Умови: Обов’язковий, Без спец-символів, Мінімум 6 символів, Максимум 32 символа)
    - Description – Опис ресурсу (Умови: Не обов’язковий, Максимум 300 символів)
    - Count – Кількість ресурсу (Умови: Обов’язковий, З плаваючою комою, Не менше 0)
    - ResourceType – Тип ресурсу (Умови: Обов’язковий, Цілочислельній, Не менше 1, Не більше 3)

Рис 3.9 – Тестові сценарії веріфікації функціональних вимог методу updateNode

При реалізації функціональної вимоги FR2 створено програмний клас NodeController з функцією getResourceByNode, який має наступну деталізовану специфікацію.

1) Вхідні параметри методу getResourceByNode :

* + Id\_node – порядковий номер Node (Умови: Обов’язковий, Цілочисельний, Не менше 0)

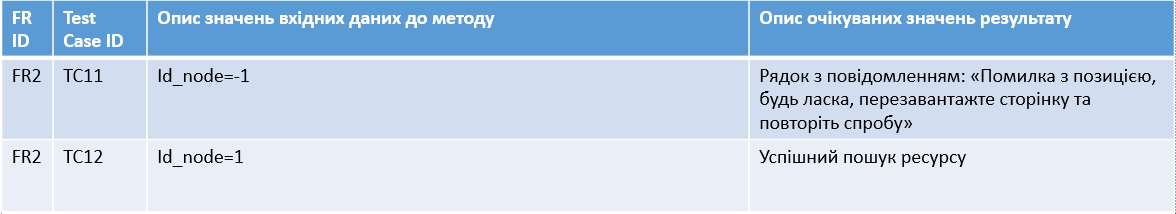


Рис 3.10 - Тестові сценарії веріфікації функціональних вимог методу getResourceByNode

3.4.2 Проектування тестових сценаріїв верифікації нефункціональних вимог

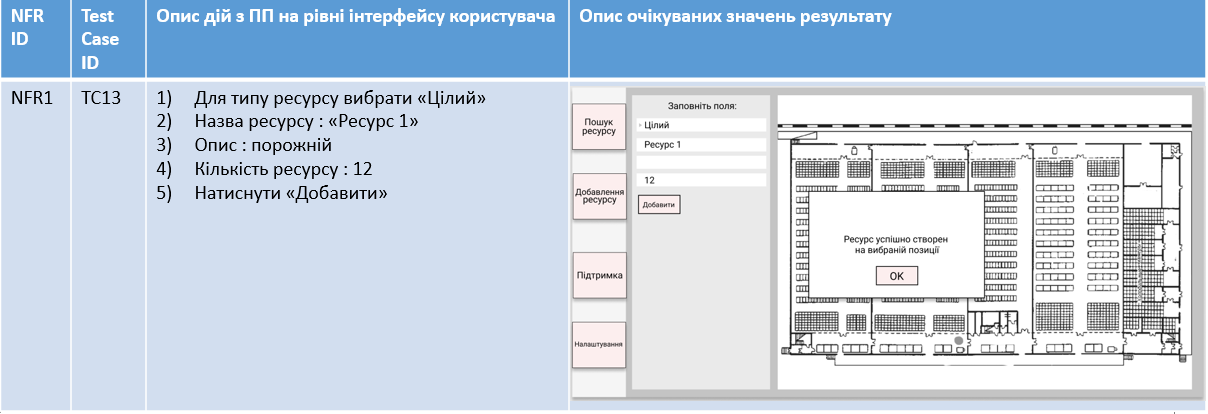
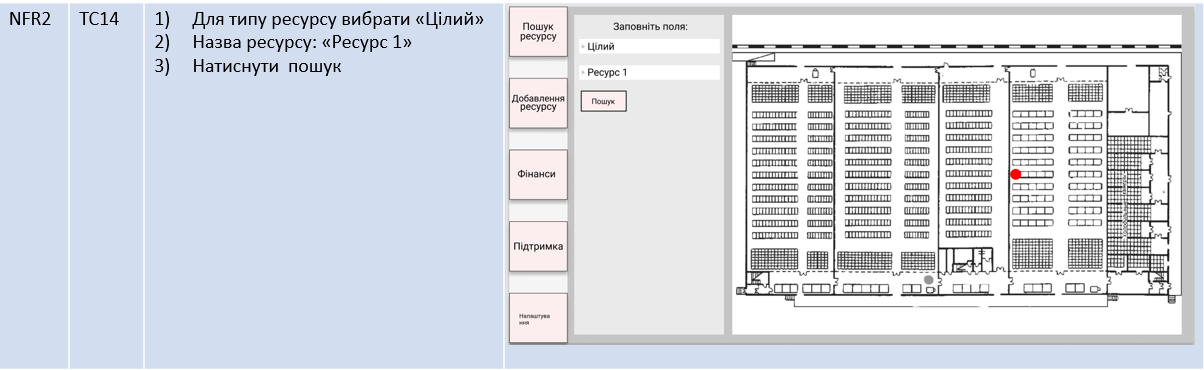
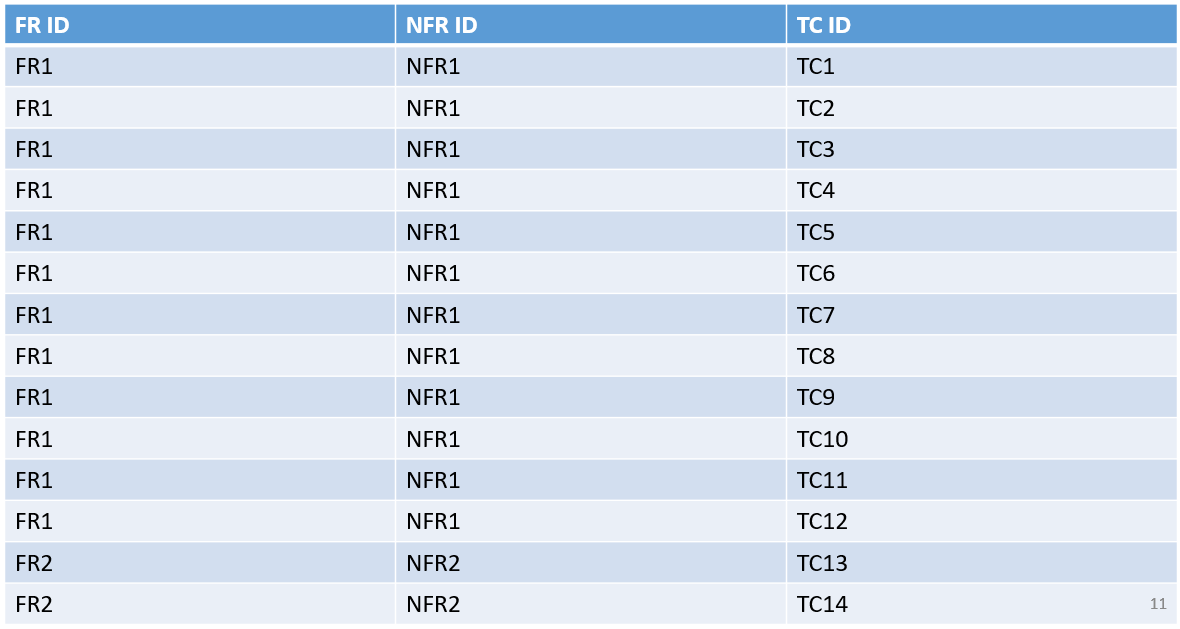


Рис 3.11 - Тестові сценарії веріфікації нефункціональних вимог

3.4.3 Створення матриці відповідності вимог до програмного продукту

Рис 3.12 – Матриця відповідності вимог

**4 Конструювання програмного продукту**

4.1 Особливості конструювання структур даних

Таблиці БД:

* CREATE TABLE resource\_type

(id\_resource\_type int PRIMARY KEY NOT NULL,

resource\_type VARCHAR(30) NOT NULL);

* CREATE SEQUENCE s\_resource

INCREMENT BY 1

START WITH 1;

* CREATE TABLE resource

(id\_resource int PRIMARY KEY DEFAULT nextval('s\_resource') NOT NULL,

title VARCHAR(50) NOT NULL,

count float NOT NULL,

description VARCHAR(300),

resource\_type int REFERENCES resource\_type(id\_resource\_type));

* CREATE SEQUENCE s\_node

INCREMENT BY 1

START WITH 1;

* CREATE TABLE node

(id\_node int PRIMARY KEY DEFAULT nextval('s\_resource') NOT NULL,

position VARCHAR(30) NOT NULL,

description VARCHAR(300),

resource int REFERENCES resource(id\_resource));

4.2 Особливості конструювання програмних модулів

4.2.1 Конструювання програмної структури

За основу бек-енд частини, написаної на Java, був взятий дизайн патерн розробки ПО – MVC.

Опис архітектури ПП:

* 1.1) інтегрованого середовища розробки – Intellij IDEA 2020.2.1
* 1.2) Рівень бізнес-логіки (Business Level) – Spring IOC 2.4.1
* 1.3) Рівень зберігання даних (Access Level) - PostgreSQL 9.6

Місце розташування – сервер на локальному середовищі.

4.2.2 Конструювання програмних класів (за наявністю об`єктно-орієнтованого програмування)

POJO Класи:

Модель Ресурсу - ResourceImpl

Рис 4.1 – ResourceImpl Model

Модель Вузлу – NodeImpl

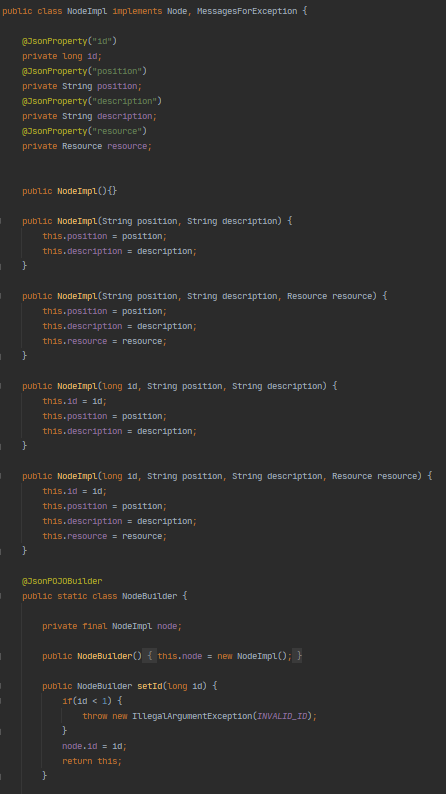
 

Рис 4.2 – ResourceImpl Model

4.2.3 Конструювання алгоритмів методів програмних класів або процедур/функцій

Get All Nodes:

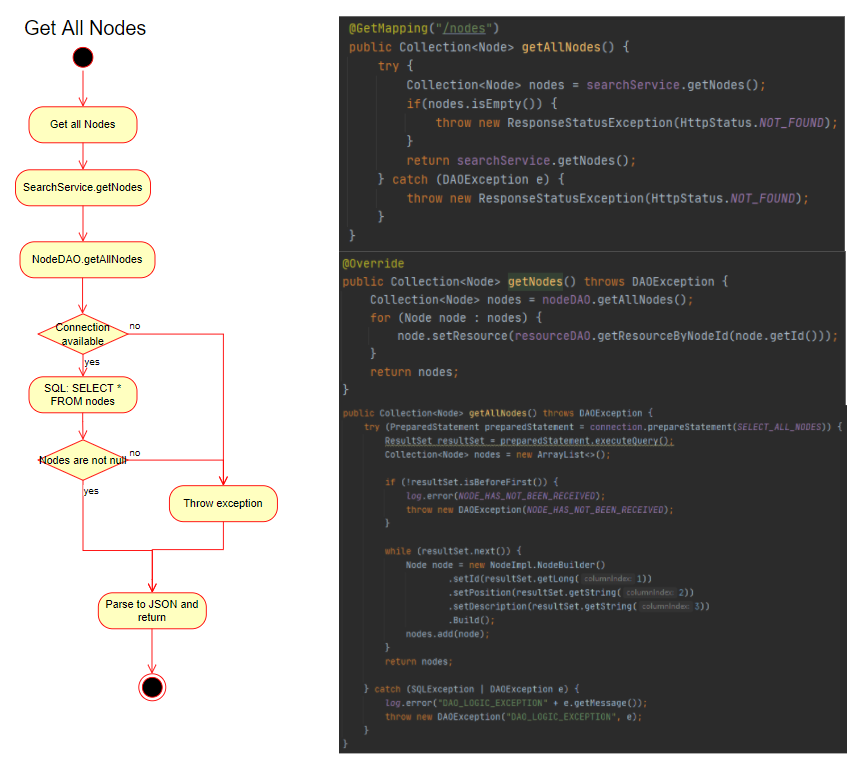


Рис 4.3 - Алгоритм роботи GetAllNodes та програмна реалізація

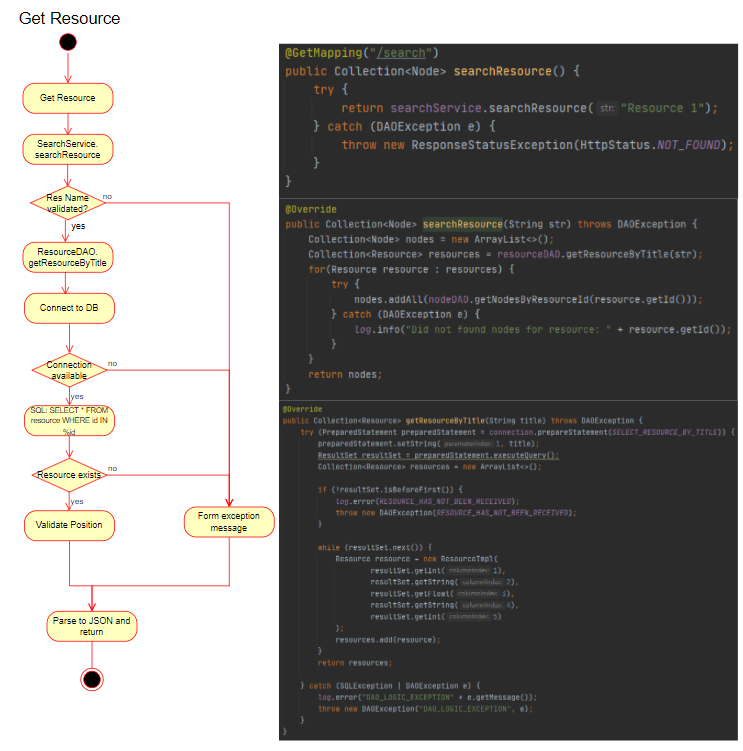


Рис 4.4 – Алгоритм роботи GetResource та програмна реалізація

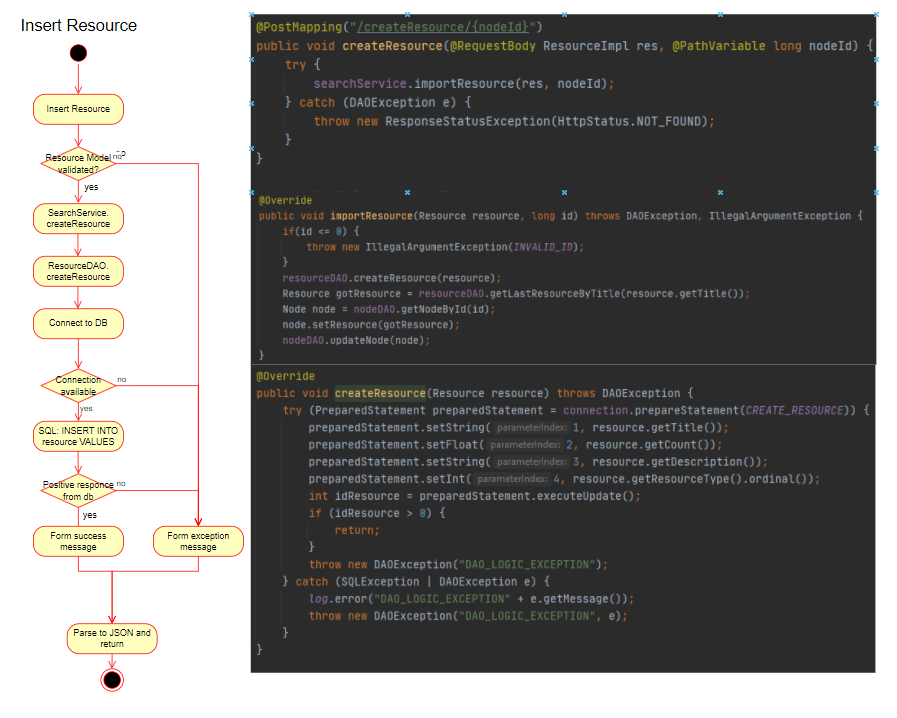
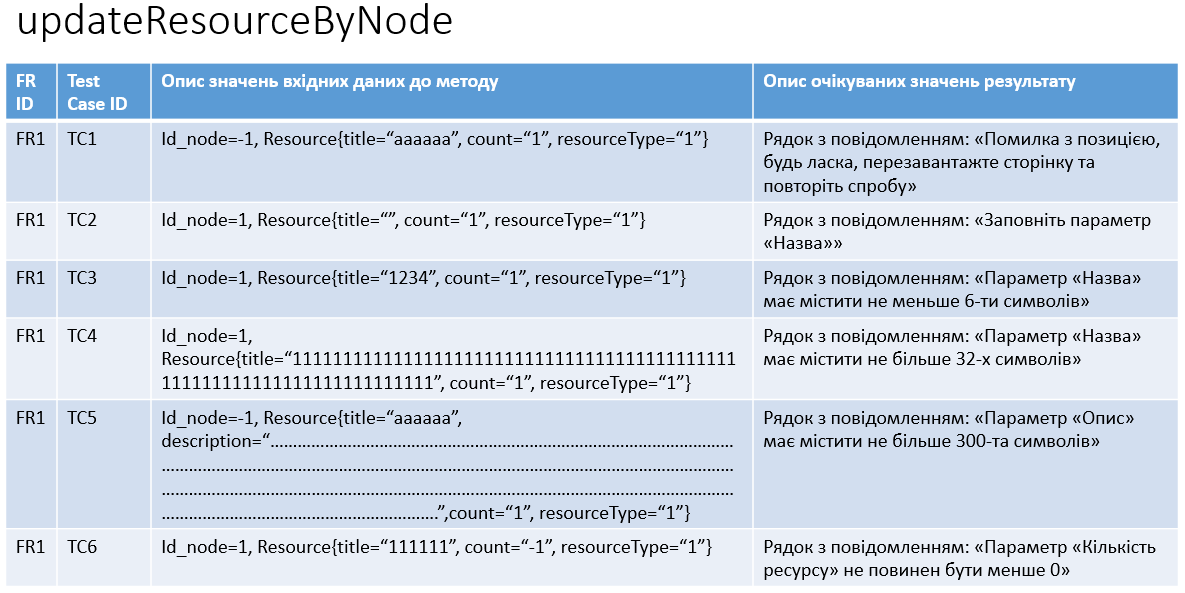
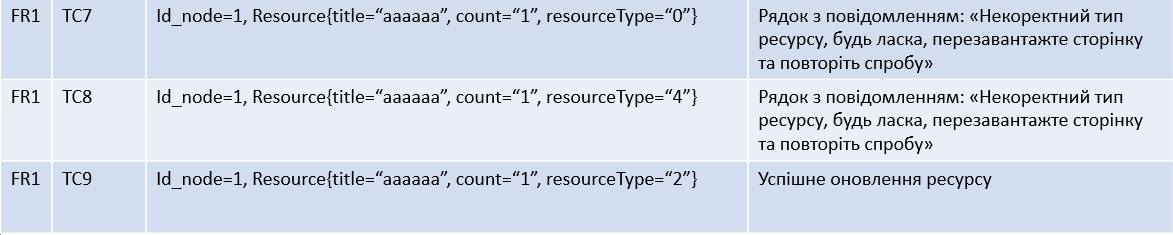


Рис 4.5 - Алгоритм роботи ImportResource та програмна реалізація

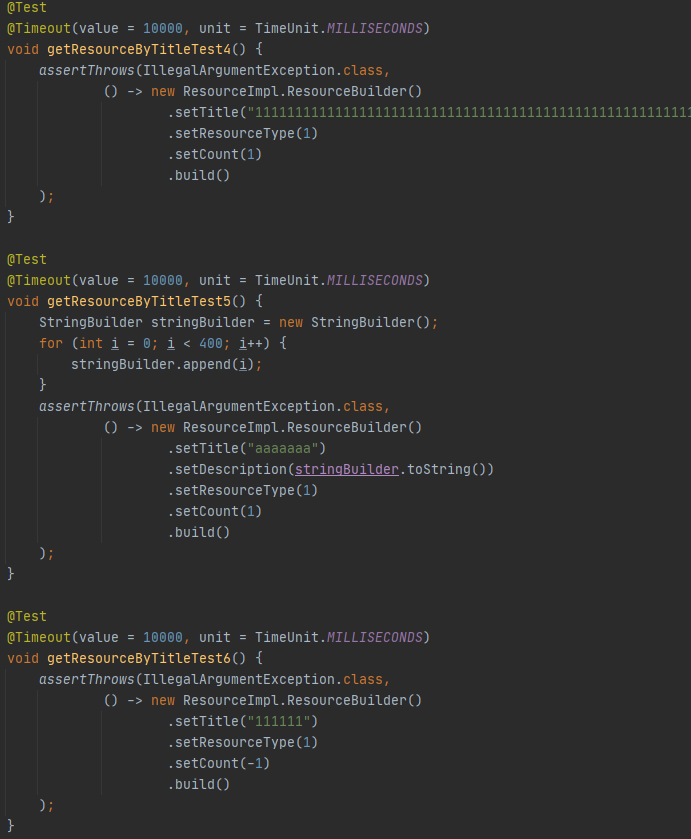
* 1. Модульне тестування програмних модулів

Тестові сценарії нефункціональних вимог:



Программні тести:



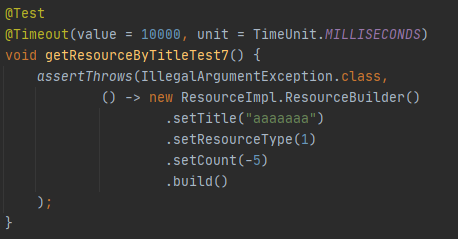


Рис 4.6 – Программні тести updateResourceByNode



Рис 4.7 – Программні тести getResourceByNode

**5 Верифікація програмного продукту**

5.1 Тестування апаратно-програмних інтерфейсів програмного продукту

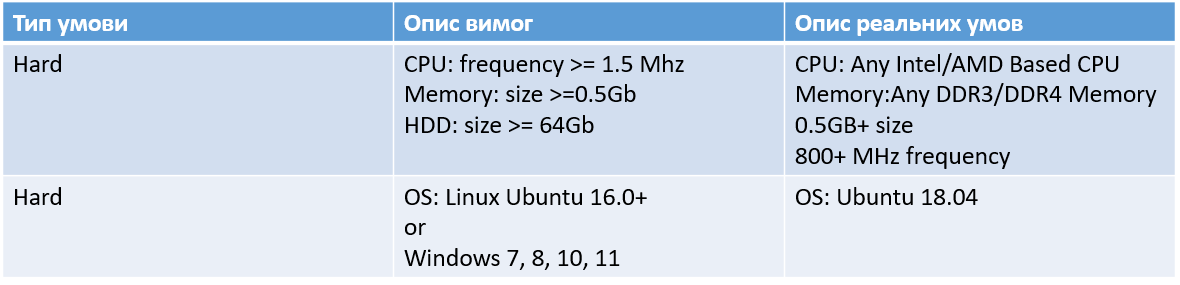
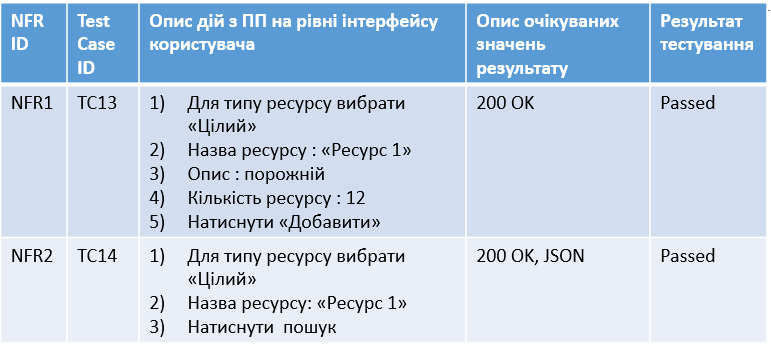


Рис 5.1 - Опис умов перевірки апаратно-програмних інтерфейсів ПП.

5.2 Тестування інтерфейсу користувача програмного продукту



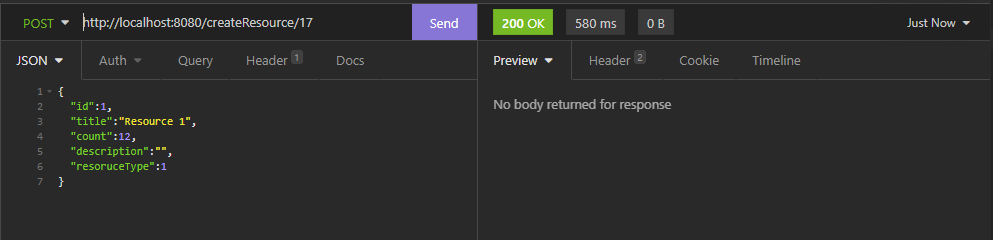


Рис 5.2 – Результат запиту за даними з ТС13

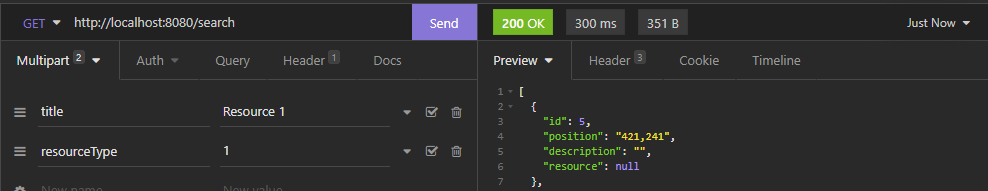
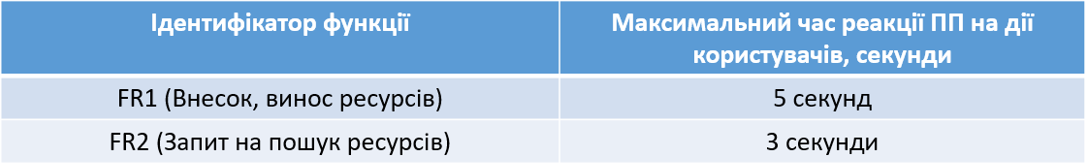
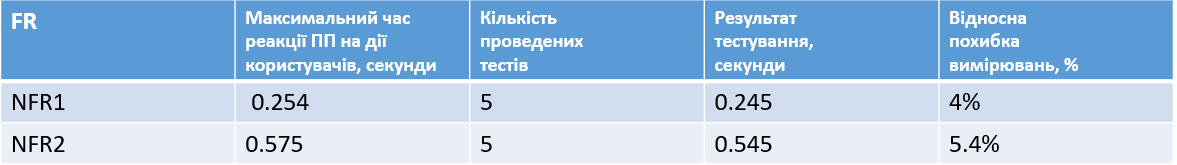


Рис 5.3 – Результат запиту за даними з ТС14

5.3 Тестування часу реакції програмного продукту на дії користувача



Похибка NFR1 = ((0,255-0,245)/0,251)\*100 = 4%

Похибка NFR2 = ((0,575-0,545)/0,551)\*100 = 5.4%

**6 Розгортання та валідація програмного продукту**

6.1 Інструкція з встановлення системного програмного забезпечення

Кроки з встановлення системного програмного забезпечення:

1) Встановити Java JRE v8.0+

2) Встановити PostgreSQL v12.0+

6.2 Інструкція з використання програмного продукту

6.3 Результати валідації програмного продукту