

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ І. СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №3-4

з дисципліни “Компоненти програмної інженерії”

“Проектування та створення архітектури програмного забезпечення і
реляційної бази даних Національного реєстру”

Тема: Єдиний реєстр спеціальних бланків нотаріальних документів

Виконали

студенти 3 курсу
групи КП-71

Песчанський Даниїл
Кравчук Аркадій

Київ 2020

Мета роботи: Уточнення даних, інтерфейсів, алгоритму функціонування, архітектури програми. Розроблення технічного проекту, яке охоплює проектування системи, що включає проектування архітектури системи і детальне проектування.

Завдання на лабораторну роботу: Проаналізувати та спроектувати архітектуру програмного забезпечення для реалізації Національного реєстру відповідно до індивідуального варіанта завдання. Задokumentувати результати проектування.

СТРУКТУРА СИСТЕМИ

Структурно система складається з ізольованих один від одного рівнів, представлену у вигляді ієрархічної структури (див рис. 1), кожна частина якої взаємодіє лише із рівнем, що знаходиться нижче за ієрархією. Рівні взаємодіють між собою за допомогою програмних інтерфейсів (API), які надаються кожним рівнем.



Рис 1. Архітектура системи Реєстру спеціальних бланків нотаріальних документів

Опис компонентів системи Реєстру спеціальних бланків нотаріальних документів:

- Користувацький інтерфейс

Представляє собою веб-додаток, завдяки якому користувач має змогу взаємодіяти з системою реєстру спеціальних бланків нотаріальних документів. Містить сторінки передбачені функціональними вимогами, які були перелічені при формуванні вимог до програмного забезпечення функціонального реєстру.

- Серверна частина

Сервер, який має доступ до БД реєстру і до якого має доступ веб-додаток. Структурно складається із наступних програмних модулів:

- модуль авторизації;
- модуль для роботи із спеціальними бланками нотаріальних документів у реєстрі ;
- модуль роботи із профілями Реєстраторів та Адміністраторів;
- модуль для роботи із логами про діяльність Реєстраторів.

- БД реєстру

Представляє собою сервер СКБД. Зберігає дані про спеціальні бланки нотаріальних документів, користувачів (Адміністраторів, та Реєстраторів) та логи змін, внесених до записів реєстраторами.

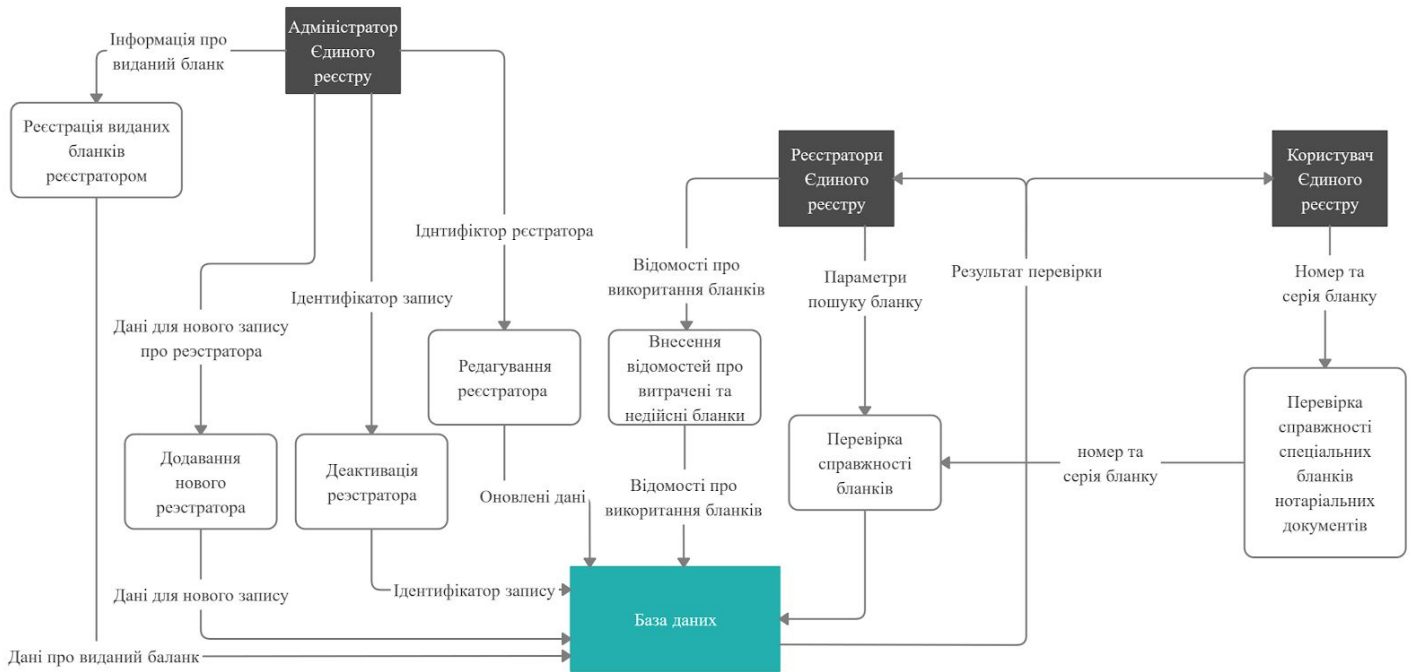
ФУНКЦІЇ І ЗАВДАННЯ, ЯКІ ВИРІШУЮТЬСЯ СИСТЕМОЮ

Система призначена для вирішення наступних завдань:

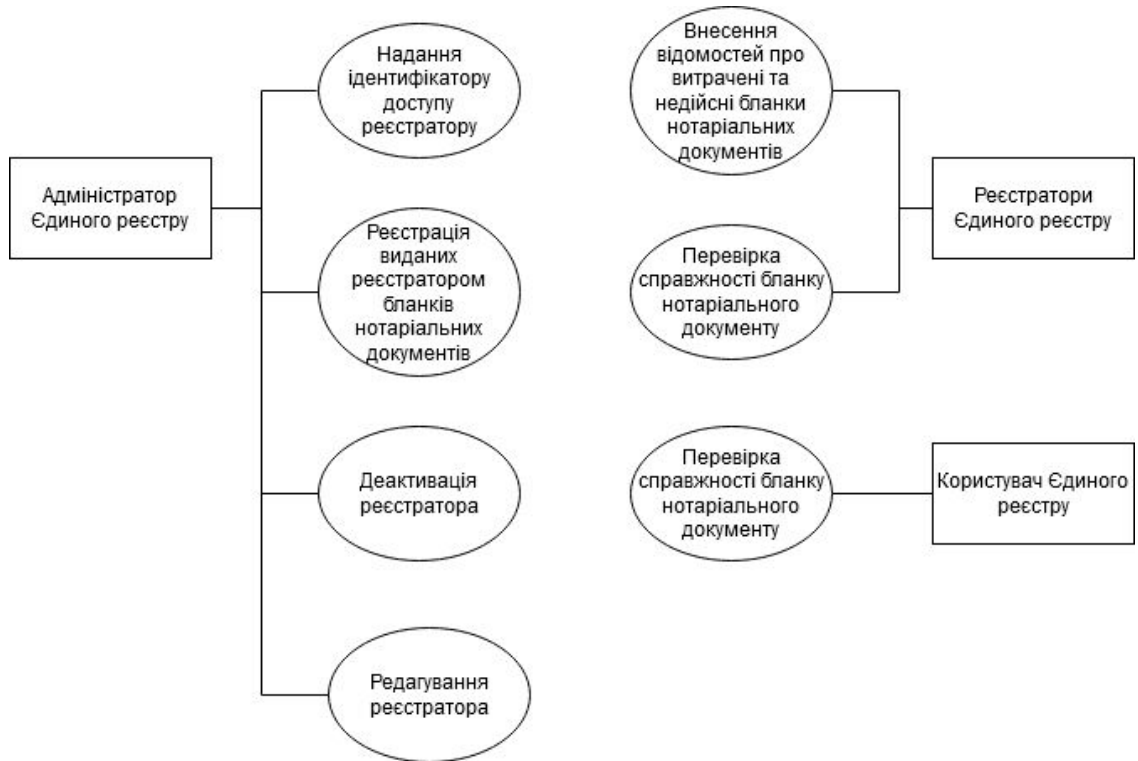
- Реєстрація відомостей про видачу нотаріусам спеціальних бланків (внесення, виправлення або видалення інформації про операції прийому-передачі бланків);
- Реєстрація відомостей про витрачання спеціальних бланків, а також відомостей про недійсні бланки (внесення, виправлення або видалення звітів про витрачання нотаріальних бланків);
- Перевірка справжності спеціальних бланків за даними РНБ для Користувачів;
- Статистичний аналіз відомостей РНБ щодо обліку постачання та звітності витрачання спеціальних бланків (формування та друк статистичних відомостей);
- Отримання інформації щодо кожного виданого та/або використаного спеціального бланка для Реєстраторів.

БІЗНЕС-ЛОГІКА ПРОГРАМИ

Загальні сценарії – нижче наведено схему інформаційних потоків, що відображає вхідні і вихідні дані, їх джерела, пункти призначення і зберігання.



Сценарії використання – нижче наведено модель запланованих функцій системи і околосистемних об'єктів, яка служить в якості угоди між замовником і розробниками.



СТРУКТУРА ДАНИХ

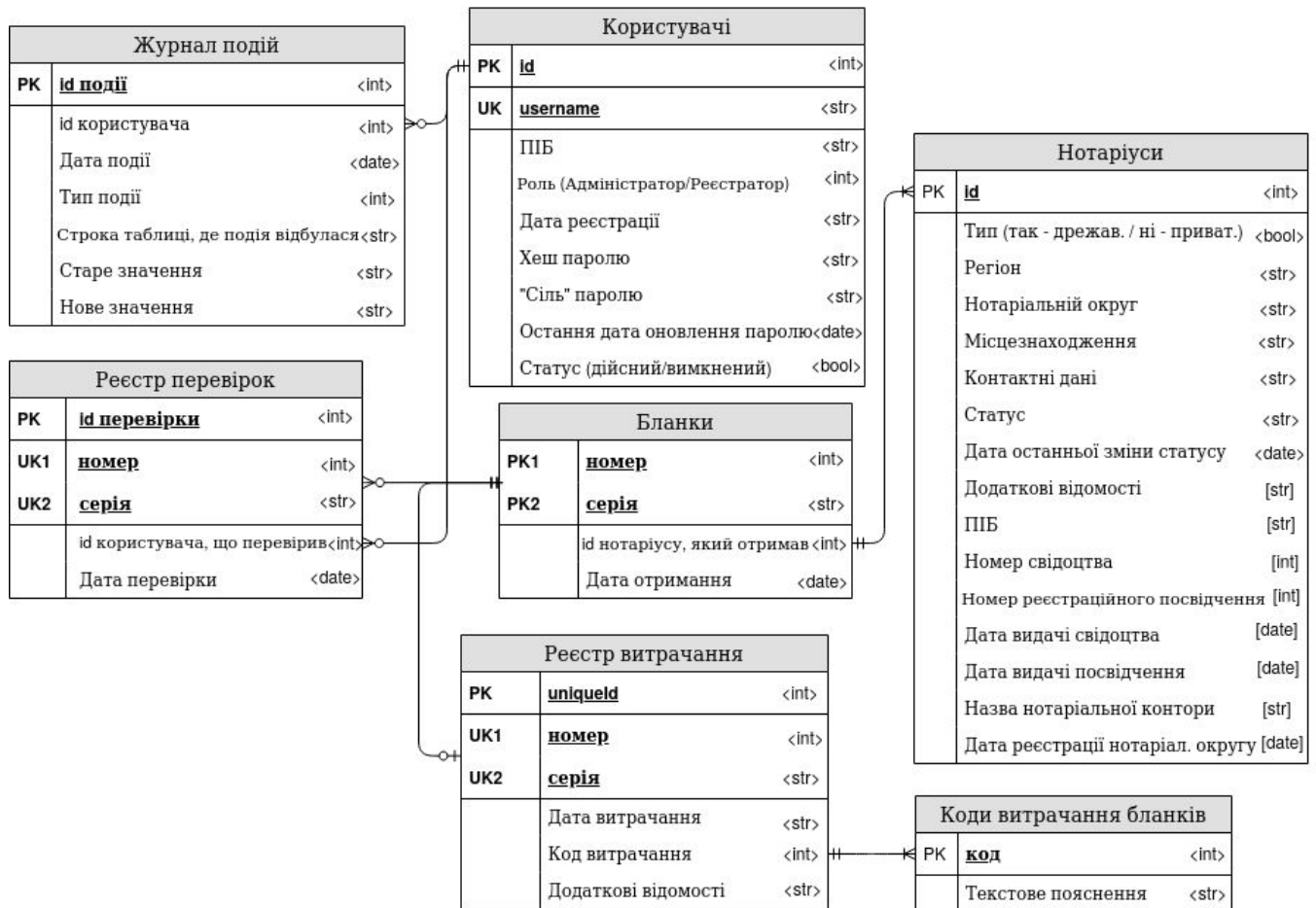


Рис 2. ER-діаграма для БД розроблюваного реєстру

Опис схеми БД Реєстру:

Таблиці:

- Користувачі – визначає усіх користувачів в системі, їх поточний стан та роль, відповідно до якої надаються певні функціональні можливості. Також містить дані, за якими проводиться авторизація та аутентифікація.
- Бланки – визначає сутність бланку за унікальним поєднанням серії та номеру бланку. Також зберігається дата видачі спеціального бланку нотаріусу. Має строге відношення один до одного з таблицею нотаріусів.
- Реєстр витрачання – визначає необхідну інформацію про витрачання бланків нотаріусом. Має відношення один до

одного з таблицею бланки, але відношення є не строгим, адже бланки можуть бути зареєстровані в системі, але ще не використані. Також має зовнішній ключ: код витрачання, який посилається на відповідну таблицю.

- Коди витрачання бланків – визначає текстову відповідність для номера витрачання бланку. Використовується під час виводу інформації для полегшення сприйняття інформації.
- Нотаріуси – відображає сутність нотаріусу у системі. В цій таблиці будуть розташовані дані як про приватних, так і про державних нотаріусів. Тип записаного нотаріусу буде визначатися окремою булевою змінною. Поля, які не потрібні для даного типу нотаріуса будуть мати відсутнє значення (null). Спільні поля для двох типів нотаріусу мають зліва позначення типу даних у трикутних дужках <...>. До державного типу нотаріусу відносяться такі поля, як: назва нотаріальної контори та дата реєстрації нотаріального округу. Відповідно усі інші поля, у яких тип даних позначається [...] – відносяться до приватного нотаріусу. Якби розроблюваний реєстр мав би можливість інтегрування з існуючими реєстрами, то тоді б дану таблицю не довелося б зберігати, а дані можна було б отримувати з відповідного реєстру.
- Реєстр перевірок – зберігає інформацію про проведені перевірки спеціальних бланків адміністраторами та реєстраторами. Фіксується дата перевірки, та зовнішні ключі: серія і номер бланку та ідентифікатор користувача. Передбачена законодавством (див. пункт

3.3 – 3.6):

<https://zakon.rada.gov.ua/laws/show/z1043-09>.

- Журнал подій – призначена для зберігання змін даних в таблицях реєстру. Під зміною даних мається на увазі операції видалення, додавання та зміни записів у Реєстрі.

Типи полів:

На схемі визначені позначені у дужках <...>. int означає цілочисельне значення, bool означає булеве значення, str означає строкове значення, date означає значення типу дати. Значення у дужках [...] позначає можливу відсутність даних.

Зв'язки між таблицями:

позначені стрілками, які визначають відношення: (не-) строго один до одного, (не-) строго один до багатьох.

Первинні та унікальні ключі:

На схемі визначені позначками “PK” та “UK” відповідно.

Тип СКБД:

Реляційна база даних PostgreSQL 12.2.

SQL-скрипти бази даних та запитів до неї

create_tables.sql

```
CREATE TABLE IF NOT EXISTS blank
(
    num            int NOT NULL,          -- номер
    series         varchar(2) NOT NULL,   -- серія
    notarius_id    bigint NOT NULL,       -- id нотаріуса, що отримав
б л а н к и
    date_receiving date NOT NULL,         -- дата отримання
    CONSTRAINT blank_pkey PRIMARY KEY (num, series ),
    CONSTRAINT foreign_key_to_notarius FOREIGN KEY (notarius_id )
        REFERENCES notarius (id) MATCH FULL
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)
```

```

CREATE TABLE TABLE IF NOT EXISTS code_usages_blank
(
    code                int NOT NULL,    -- код
    text_representation text NOT NULL,    -- текстове пояснення
    CONSTRAINT code_usages_blank_pkey PRIMARY KEY (code)
)

CREATE TABLE IF NOT EXISTS users
(
    id                  serial NOT NULL, -- унікальний ідентифікатор
користувача
    name                text NOT NULL,    -- ПІБ користувача
    role                int NOT NULL,     --
роль (Адміністратор/реєстратор)
    date_registration   date NOT NULL,    -- дата реєстрації
    username            text NOT NULL,    -- логін користувача
    pwd_hash            text NOT NULL,    -- хеш паролю
    pwd_salt            text NOT NULL,    -- "сіль" паролю
    date_last_update    date NOT NULL,    -- остання дата оновлення
ідентифіка
    status              boolean NOT NULL, -- статус (дійсний/вимкнений)
    CONSTRAINT users_pkey PRIMARY KEY (id)
    CONSTRAINT uniq_username UNIQUE(username)
)

CREATE TABLE IF NOT EXISTS notarius
(
    id                  serial NOT NULL, -- унікальний
ідентифікатор
    type                boolean NOT NULL, --
тип (приватний/державний)
    status              text NOT NULL,    -- статус
    date_status_update  date NOT NULL,    -- дата набуття статусу
    num_certificate     text,             -- номер свідоцтва
    num_card            bigint,           -- номер посвідчення
    name                text,             -- ПІБ нотаріуса
    name_organization   text,             -- назва нотаріальної
контори
    region              text NOT NULL,    -- регіон
    contacts            text NOT NULL,    -- контактні дані
    notarius_region     text NOT NULL,    -- нотаріальний округ
    additional_info     text,             -- додаткові відомості
    date_issue_certificate date,          -- дата видачі свідоцтва
    date_issue_card     date,             -- дата видачі
посвідчення
    date_reg_region     date,             -- дата реєстрації
нотаріального округу
    location            text NOT NULL,    -- місцезнаходження
    CONSTRAINT notarius_pkey PRIMARY KEY (id)
)

CREATE TABLE IF NOT EXISTS usages_register
(
    id                  serial NOT NULL, -- унікальний ідентифікатор
витрачання
    num_blank           int NOT NULL,     -- номер бланку
    series_blank        text NOT NULL,    -- серія бланку
    date_usage          date NOT NULL,    -- дата витрачання

```

```

        code_usage      int NOT NULL,      -- код витрачання
        additional_info text,              -- додаткові відомості
        CONSTRAINT usages_register_pkey PRIMARY KEY (id),
        CONSTRAINT blank_ukeys UNIQUE(num_blank, series_blank),
        CONSTRAINT foreign_key_to_code_usage FOREIGN KEY (code_usage)
            REFERENCES code_usages_blank (code) MATCH FULL
            ON UPDATE RESTRICT
            ON DELETE RESTRICT
    )

CREATE TABLE IF NOT EXISTS verifications_register
(
    id                serial NOT NULL, -- id перевірки
    num_blank         int NOT NULL,    -- номер бланку, що
    перевіряється
    series_blank      text NOT NULL,   -- серія бланку, що
    перевіряється
    user_id           bigint NOT NULL, -- id користувача, що
    перевірів
    date_verification date NOT NULL,   -- дата перевірки
    CONSTRAINT verifications_register_pkey PRIMARY KEY (id),
    CONSTRAINT blank_ukeys UNIQUE(num_blank, series_blank)
)

CREATE TABLE IF NOT EXISTS journal_actions
(
    id                serial NOT NULL, -- id події
    user_id           bigint NOT NULL, -- id користувача
    action_date       date NOT NULL,   -- дата події
    action_type       int NOT NULL,    -- тип події
    row_affected      text NOT NULL,   -- строка таблиці де подія
    відбулася
    old_value         text NOT NULL,   -- старе значення
    new_value         text NOT NULL,   -- нове значення
    CONSTRAINT journal_actions_pkey PRIMARY KEY (id),
    CONSTRAINT foreign_key_to_user_id FOREIGN KEY (user_id )
        REFERENCES users (id) MATCH FULL
        ON UPDATE RESTRICT
        ON DELETE RESTRICT
)

```

Приклади запитів до бази даних

Приклад внесення даних про отримання бланків нотаріусом:

add_blanks.sql

```
INSERT INTO blank(num, series, notarius_id, date_receiving) VALUES (234135,
"AB", 135135, CURRENT_TIMESTAMP);
```

Внесення інформації про використання бланка:

use_blanks.sql

```
INSERT INTO usages_register(num_blank, series_blank, date_usage, code_usage, additional_info) VALUES (234135, "AB", CURRENT_TIMESTAMP, 20, NULL);
```

Пошук бланків з певним кодом витрачання:

search_blanks.sql

```
SELECT num_blank, series_blank FROM usages_register WHERE code_usage=32;
```

Перевірка витрачання бланку за серією та номером:

verification_blanks.sql

```
INSERT INTO verifications_register(num_blank, series_blank, user_id, date_verification) VALUES (234135, "AB", 758, CURRENT_TIMESTAMP);  
SELECT * FROM usages_register WHERE num_blank = 234135 AND series_blank = "AB";
```

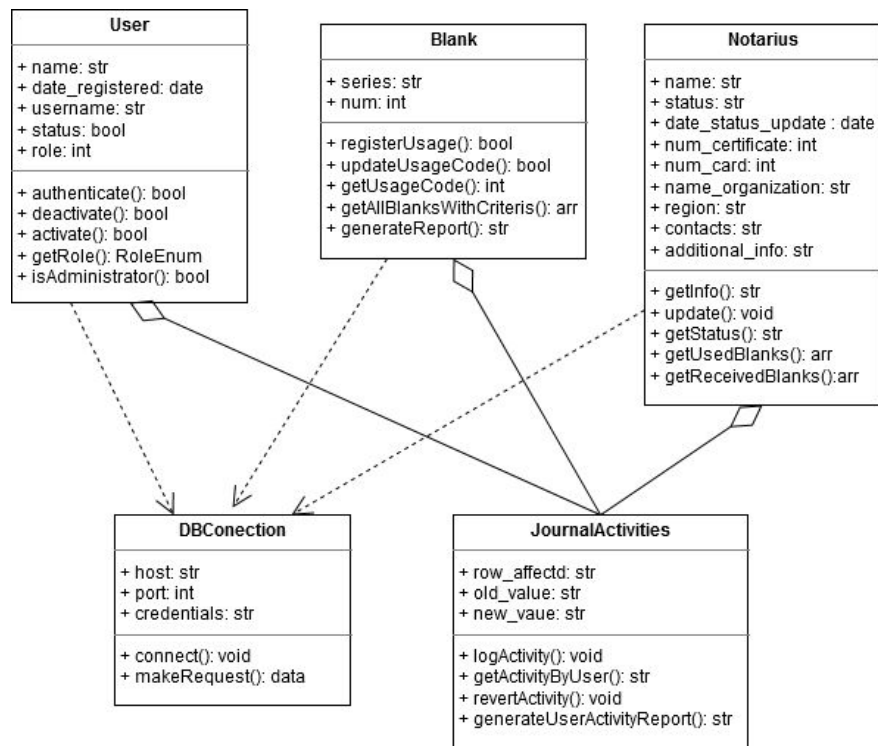
Деактивація користувача за юзернеймом:

deactivate_user.sql

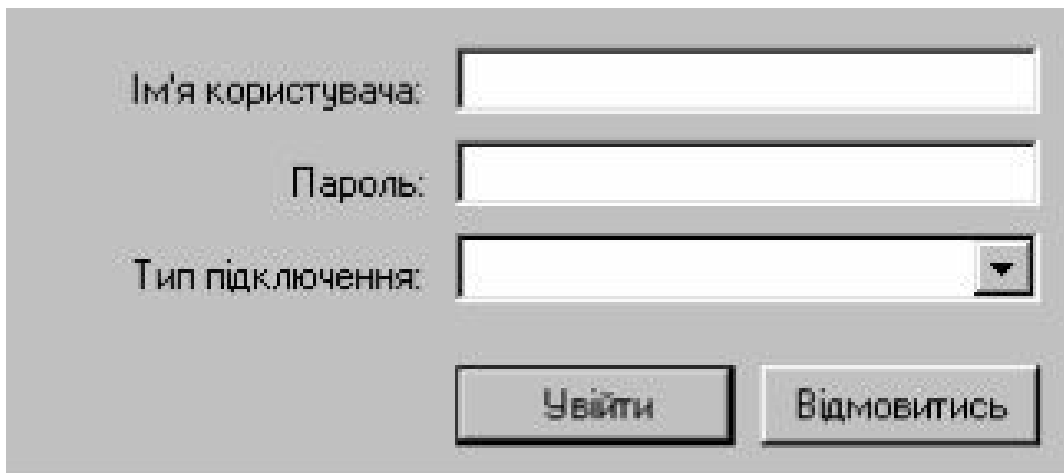
```
UPDATE users SET status = false WHERE id = (SELECT id FROM users WHERE username = "some_username");
```

Класи об'єктно-орієнтованого програмування

Класи об'єктно-орієнтованого програмування, які будуть використані при розробці ПЗ Реєстру зазначені на діаграмі нижче:



АРХІТЕКТУРА КОРИСТУВАЧЬКОГО ІНТЕРФЕЙСУ ПРОГРАМИ (основний функціонал)



Ім'я користувача:

Пароль:

Тип підключення:

рис. 2.1 вікно «Вхід в систему»

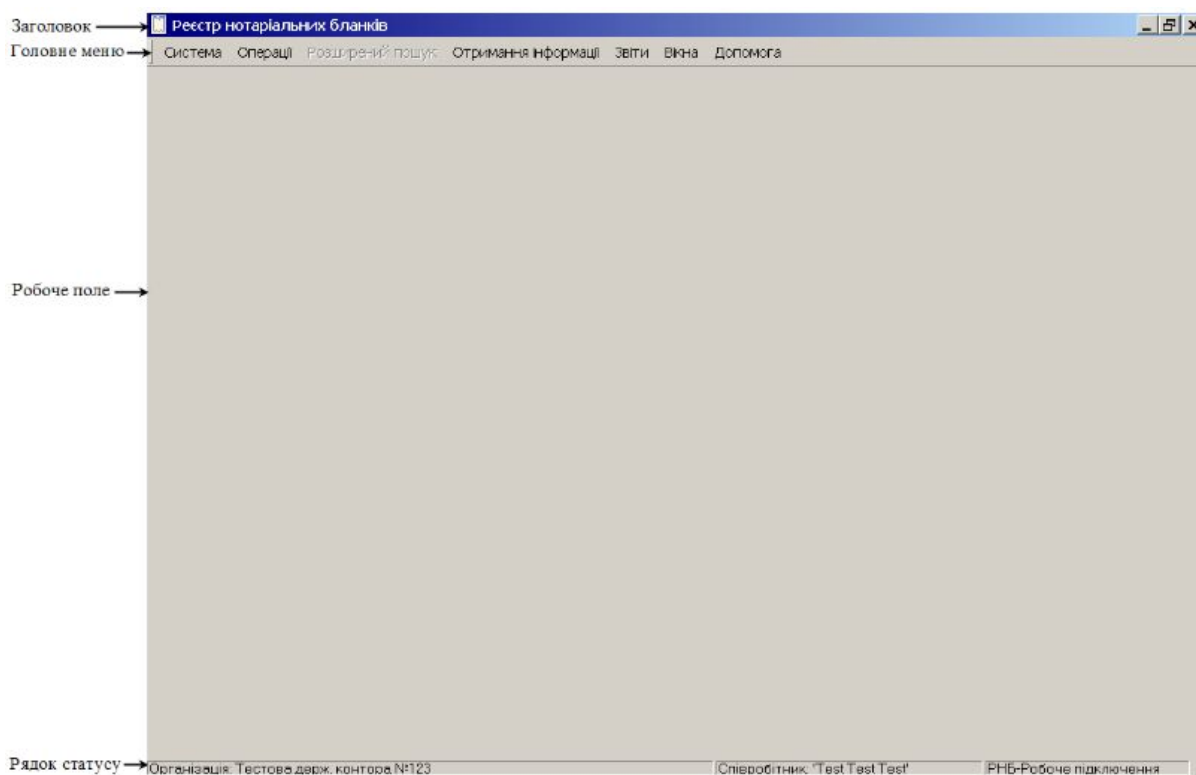


рис. 2.2 вікно «Головне вікно програми»

Система Операції Розширений пошук Отримання інформації Звіти Вікна Допомога

Інформація про бланк

Серія: Номер:

☐ Звільнення від оплати

Результати перевірки

Дата та час перевірки:

Ідентифікаційний номер перевірки:

Бланк: Дата отримання:

Бланк отримав:

Код витрачання:

Додаткові відомості:

Дата витрачання:

Бланк витрачений:

Інформація про нотаріуса

Свідцтво: Посвідчення: Прізвище: Ім'я: По батькові:

Вибір нотаріуса

Тип організації:

Регіон:

Назва організації:

Нотаріус:

Адреса нотаріуса

Регіон:

Район:

Населений пункт:

Вулиця:

Результат пошуку

Статус: Дата набуття статусу:

Нотаріус:

Регіон: Нотаріальний округ:

Назва організації:

Контактні дані:

Додаткові відомості:

Організація: Тестова держ. контора №123 | Співробітник: "Test Test Test" | РНБ-Робоче підключення

рис. 2.3 вікно «Перевірка – Реєстр нотаріальних бланків»

Результати перевірки

Дата та час перевірки:

Ідентифікаційний номер перевірки:

Бланк: Дата отримання:

Бланк отримав:

Код витрачання:

Додаткові відомості:

Дата витрачання:

Бланк витрачений:

рис 2.6 вікно «Результат перевірки справжності нотаріального бланка»

Вибір нотаріуса - Реєстр нотаріальних бланків

За вказаними параметрами пошуку знайдено інформацію щодо декількох нотаріусів.
Для продовження операції оберіть одного з них:

Свідцтво	Посвідч...	Статус	Нотаріус	Назва організації	Регіон
1029	27	діє	Пікуль Ярослав Григорович	Приватний нотаріус	Тернопіль
2767	27	діє	Суслін Євген Олександрович	Приватний нотаріус	м.Севасті
2612	27	діє	Ковтун Клара Вадимівна	Приватний нотаріус	Чернігів
2441	27	діє	Писаковська Ольга Іванівна	Приватний нотаріус	Кіровоград
1700	27	діє	Тихонова Лариса Володимирівна	Приватний нотаріус	Луганськ
1225	27	діє	Цекеева Ірина Олександрівна	Приватний нотаріус	Запоріжжя
2463	27	діє	Адамова Людмила Антонівна	Приватний нотаріус	Харківськ
1416	27	діє	Корпало Ганна Василівна	Приватний нотаріус	Львівськ
1406	27	діє	Міняйло Ірина Петрівна	Приватний нотаріус	Черкаськ
1383	27	прип...	Суржук Наталія Дмитрівна	Приватний нотаріус	Чернівець
1198	27	діє	Кузьміч Надія Володимирівна	Приватний нотаріус	Волинськ
2929	27	діє	Хомін Віталій Ярославович	Приватний нотаріус	Закарпат
2588	27	діє	Кравцова Лариса Андріївна	Приватний нотаріус	Сумська
2896	27	діє	Артемова Леся Яківна	Приватний нотаріус	Хмельниць
2658	27	діє	Овдіюк Діна Іванівна	Приватний нотаріус	Рівненськ
2818	27	прип...	Федорук Любов Петрівна	Приватний нотаріус	Херсонськ

✓Прийняти ✗Закрити

рис. 2.4 вікно «Вибір нотаріуса – Реєстр нотаріальних бланків»

Результат пошуку

Статус: діє Дата набуття статусу: . . .

Нотаріус: Білорус Надія Василівна

Регіон: Кіровоградська обл. Нотаріальний округ: Бобринецький районний

Назва організації: Приватний нотаріус

Контактні дані: 27200, Бобринецький р., м. Бобринець, вул. Леніна, 39, тел.:(05257)35271

Додаткові відомості:

Свідцтво: 672 Посвідчення: 13

Дата видачі: Свідцтва: 14.03.1994 Посвідчення: 05.02.1996

рис 2.5 вікно «Результат пошуку інформації про нотаріуса»

Назва організації: держ. контора №123 (діє) Нотаріус: Test Test Test (діє) Додаткові відомості: ...

№	Дата	Код	№	Дата	Код	№	Дата	Код	№	Дата	Код	№	Дата	Код
1			2			3			4			5		
6			7	. . .	0	8			9			10		
11			12			13			14			15		
16			17			18			19			20		
21			22			23			24			25		

рис 2.8 вікно «Внесення звітів зі знайденими бланками»

ЗАСОБИ РОЗРОБКИ І СКЛАД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Розроблюване програмне забезпечення складатиметься з:

- Серверної частини (server-side) – складається з переліку модулів(наведених у описі структури системи), які в сукупності виконують основну роботу системи.
- Клієнтської частини (client-side) – включає в себе обмін даними з серверною частиною та візуалізацію отриманих від серверу даних.
- Тести серверної та клієнтської частини – автоматизовані тести основного функціоналу системи.

Розроблювана система буде веб-орієнтованою. Як засоби розробки буде використано мову програмування Python 3.6 для серверної частини, та мова програмування JavaScript із фреймворком React. В якості СКБД виступатиме PostgreSQL.

ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Технічне забезпечення системи Реєстру складається із серверного обладнання для розгортання розробленого програмного забезпечення.

Даний сервер має забезпечувати швидку роботу системи Реєстру, тому він має відповідати наступним вимогам:

- Статична IP адреса в мережі Інтернет.
- Пропускна здатність інтернет-з'єднання: не менше 100 Mbps.
- ОС: Linux Debian 10.3 або Windows Server 2019.
- CPU: 4 ядра, тактова частота, не менше: 3.5 GHz.
- RAM: не менше 32 GB.
- HDD space: не менше 256 GB.
- Предстановлене ПЗ: Docker, Docker-compose, Kubernetes.

ЧИСЕЛЬНІСТЬ, КВАЛІФІКАЦІЯ І ФУНКЦІЇ ПЕРСОНАЛУ

Персонал: Кравчук Аркадій, Песчанський Даниїл.

Чисельність персоналу: 2 людини.

Кваліфікації персоналу: інженери програмного забезпечення комп'ютерних систем, які є студентами третього курсу ФПМ КПІ ім. Ігоря Сікорського.

Функції персоналу: вчасне і якісне завершення процесу створення та тестування ПЗ відповідно до вимог, які містяться у ТЗ системи, розгортання системи на виділеному сервері.

Розподіл робіт між учасниками:

- Кравчук Аркадій: розробка користувацького інтерфейсу та архітектури БД, тестування серверної частини.
- Песчанський Даниїл: розробка серверної частини та архітектури БД, тестування користувацького інтерфейсу.