

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка компонентів замовлення розробки та використання ПЗ
інформаційної системи біржі спостережень
(тема)

Виконав:

студент II курсу, групи СПРм-22-1

Путілов С.Ю.

(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-наукова

Освітня програма Системне проектування

(повна назва освітньої програми)

Керівник проф. Саваневич В.Є.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ

(підпис)

проф. Гребеннік І.В.

(прізвище, ініціали)

2024 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

12.06.2024



Путілов С.Ю.

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 12 червня 2024 р.

Керівник кваліфікаційної роботи



Саваневич В.Є.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова

Освітня програма Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри СТ

проф. Гребеннік І.В.

" 01 " 04 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Путілову Сергію Юрійовичу
(прізвище, ім'я, по батькові)

- 1. Тема роботи** Розробка компонентів замовлення розробки та використання ПЗ інформаційної системи біржі спостережень
затверджена наказом по університету від " 01 " квітня **2024р. № 259 Ст**
- 2. Термін подання студентом роботи до екзаменаційної комісії** 15 червня 2024р.
- 3. Вихідні дані до роботи (проекту)** Розробити компоненти збору, обробки та розповсюдження даних інформаційної системи. Система повинна являти собою веб-сайт з інтерфейсом доступу до бази даних. Перелік використовуваних програмних засобів: ОС Microsoft Windows 11, MariaDB, Postman, Visual Studio Code, VirtualBox. Технічне забезпечення: IBM-сумісний ПК з ЦП Intel Core i5 та вище.
- 4. Перелік питань, що потрібно опрацювати в роботі**
4.1 Вступ 4.2 Біржа спостережень, аналіз предметної галузі 4.3 Біржа спостережень та аналіз існуючих систем 4.4 Постанова вимог та задач до біржі спостережень 4.5 Системне проектування біржі спостережень 4.6 Проектування функціональної моделі IDEF0 4.7 Проектування моделі потоків даних DFD 4.8 Визначені функціональні вимоги до клієнтської частини ІС 4.9 Розробка діаграми варіантів використання 4.10 Діаграма послідовності дій 4.11 Проектування діаграми класів 4.12 Постанова завдань для розробки ІС 4.13 Опис прийнятих рішень 4.14 Аналіз та обґрунтування обраних технологій проектування 4.15 Вибір РСУБД для проектування бази даних 4.16 Вибір архітектури проектування 4.17 Вибір шаблону проектування 4.18 Реалізація контролерів і моделей 4.19 Створення графічного інтерфейсу користувача 4.20 Тестування запитів серверної частини ІС 4.21 Висновок 4.22 Перелік джерел посилання
- 5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів,**

комп'ютерних ілюстрацій (слайдів) 5.1 Сторінка «Список телескопів», 5.2 Функціональна модель «Біржа спостережень» 5.3 Декомпозиція функціональної моделі «Біржа спостережень» 5.4 Декомпозиція бізнес-функції «Реєстрація / авторизація» 5.5 Декомпозиція бізнес-функції «Функції замовника» 5.6 Декомпозиція бізнес-функції «Функції дослідника» 5.7 Діаграма потоків даних біржі спостережень 5.8 Карта веб-сторінок біржі спостережень 5.9 Діаграма варіантів використання біржі спостережень 5.10 Рисунок 5.11 Діаграма послідовності дій для прецеденту «Проведення сертифікації обладнання» 5.12 Діаграма класів біржі спостережень 5.13 Обрані технології проектування біржі спостережень 5.14 Логічна модель бази даних для інформаційної системи біржі спостережень 5.15 Візуальне представлення шаблону проектування «Публікація–підписка» 5.16 Сторінка «Список обладнання» 5.17 Сторінка «надання інформації про обладнання» 5.18 Сторінка «Створення нового дослідження»

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів атестаційної роботи	Термін виконання етапів роботи	Примітка
1.	Отримання завдання кваліфікаційної роботи	01.04.2024	Виконано
2.	Аналіз предметної області, аналоги та визначення сфери застосування	02.04.2024-04.04.2024	Виконано
3.	Визначення вимог до інформаційної системи, системне проектування біржі спостережень	05.04.2024-15.04.2024	Виконано
4.	Проектування процесів і функцій з використанням UML	16.04.2024-20.04.2024	Виконано
5.	Написання програмного коду інформаційної системи	21.04.2024-20.05.2024	Виконано
6.	Тестування API серверної частини додатку	21.05.2024	Виконано
7.	Розробка «Посібника користувача»	24.05.2024	Виконано
8.	Оформлення пояснювальної записки та документація програмного коду	01.06.2024-11.06.2024	Виконано
9.	Оформлення графічної частини презентаційних матеріалів, комп'ютерних матеріалів для захисту атестаційної роботи	за 4 днів	Виконано
10.	Представлення на рецензування	за 3 дні	Виконано
	Представлення атестаційної роботи в ДЕК	за 2 дні	Виконано

Дата видачі завдання 01.04.2024

Студент Путілов С.Ю.

(підпис)

Керівник роботи проф. Саваневич В.Є.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка з кваліфікаційної роботи містить: 118 с., 2 табл., 35 рис., 4 додатки, 23 джерел

БАЗА ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, MYSQL, АНАЛІЗ ДАНИХ, ІНТЕРФЕЙС, СЕРВЕРЬ БІРЖА СПОСТЕРЕЖЕНЬ, ШАБЛОН ПРОЕКТУВАННЯ, ПЛАГІН, АРХІТЕКТУРА

Об'єктом досліджень атестаційної роботи аналіз вимог та проектування інформаційної системи біржі спостережень.

Предметом дослідження є комплекс програмних та апаратних засобів, що використовуються для аналізу, запису та обробки великих обсягів інформації, що використовується на та взаємодіє з іншими наборами даних через інтерфейс користувача.

Методи дослідження є впровадження аналітичних та практичних засобів тестування інформаційної системи з створенням цілісності даних за допомогою реляційної системи управління базами даних для швидкої та ефективної взаємодії з даними і використанням архітектурного шаблону проектування MVC.

Результатом кваліфікаційної роботи є веб-додаток, що використовує функціонал аналізу, запису та обробки інформації різної величини. Додаток допоможе створити замовлення на певну обробку даних за введеним описом і отримати готові результати після проведення дослідження.

ABSTRACT

The explanatory note on the qualification work contains: 118 pages, 2 tables, 35 figures, 4 appendices, 23 source

DATABASE, INFORMATION SYSTEM, DATABASE MANAGEMENT SYSTEM, MYSQL, DATA ANALYSIS, INTERFACE, OBSERVATION EXCHANGE SERVER, DESIGN PATTERN, PLUG-IN, ARCHITECTURE

The object of research attestation work is the analysis of requirements and design of the information system of the exchange of observations.

The subject of research is a complex of software and hardware tools used to analyze, record and process large volumes of information that is used on and interacts with other data sets through a user interface.

The research methods are the implementation of analytical and practical means of testing the information system with the creation of data integrity using a relational database management system for fast and effective interaction with data and using the MVC architectural design template.

The result of the qualification work is a web application that uses the functionality of analysis, note and processing of information of various sizes. The application will help create an order for certain data processing according to the entered description and receive ready results after conducting the study.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМОВЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Біржа спостережень, аналіз предметної галузі	10
1.2 Біржа спостережень та аналіз існуючих систем	11
1.3 Аналіз інформаційної системи біржа спостережень	17
1.4 Постановка вимог та задач до біржі спостережень	19
2 РОЗРОБКА ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	21
2.1 Системне проектування біржі спостережень	21
2.2 Проектування функціональної моделі IDEF0 для інформаційної системи біржі спостережень	22
2.3 Проектування моделі потоків даних біржі спостережень DFD	33
2.4-Визначені функціональні вимоги до клієнтської частини інформаційної системи біржі спостережень	36
2.5-Розробка діаграми варіантів використання біржі спостережень	38
2.6 Діаграма послідовності дій (Sequence diagram)	39
2.7 Проектування діаграми класів та визначення її мети	41
2.8 Постановка завдань для розробки інформаційної системи біржі спостережень	44
3 ОПИС ПРИЙНЯТИ РІШЕНЬ, РОЗРОБКА БІРЖІ СПОСТЕРЕЖЕНЬ	46
3.1 Аналіз та обґрунтування обраних технологій проектування	46
3.2 Вибір РСУБД для проектування бази даних	48
3.3 Вибір архітектури проектування	51
3.4 Вибір шаблону проектування біржі спостережень.....	53
3.5 Реалізація контролерів і моделей для біржі спостережень.....	54

3.6 Створення графічного інтерфейсу користувача	60
3.7 Тестування запитів розробленої частини інформаційної системи біржі спостережень	67
ВИСНОВОК.....	70
АПРОБАЦІЇ	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	72
ДОДАТОК А	74
ДОДАТОК Б	86
ДОДАТОК В.....	98
ДОДАТОК Г.....	116

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМОВЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ, ТЕРМІНІВ

Біржа спостережень – спеціалізована платформа або система, яка надає можливість спостерігати за подіями або об'єктами.

ІС – інформаційна система

БД – база даних

JSON – текстовий формат обміну даними

РСУБД – реляційна система управління базами даних

UML – уніфікована мова моделювання

ІТ – інформаційні технології

ВСТУП

Розвиток технологій за останні десятиліття приніс значний прогрес у сфері інформаційних технологій (ІТ). Розвиток штучного інтелекту привів до виникнення інноваційних систем, які можуть розпізнавати обличчя, голос і виконувати складні аналітичні завдання з можливістю автоматизації базових процесів. Зростання кількості підключених пристроїв до інтернету створює розширені можливості для збору та аналізу даних у реальному часі. Отже, було обрано предметну область біржа спостережень, що працює з аналізом та збором даних у реальному.

Біржа спостережень – це спеціалізована платформа або система, яка надає можливість спостерігати за подіями або об'єктами. Головна ціль біржі спостережень полягає у зборі, обробці та аналізі отриманої інформації про події чи об'єкти. ІС отримує повідомлення від замовників, які поділяються на категорії, та за необхідності спостерігачі можуть оформити підписку на виконання певного завдання. Результатом спостереження повинна бути зібрана та проаналізована інформація і представлена замовнику в обробленому вигляді для чіткого її розуміння та можливості ефективно взаємодіяти з результатом дослідження.

Кінцевим результатом проектування є створена інформаційна система біржа спостережень, що надає можливість замовляти дослідження даних в певних регіонах на тривалий час. Цей функціонал допоможе комерційним установам або фізичним особам отримувати необхідний аналіз даних для особистих цілей. Уніфікована біржа спостережень може стати лідером серед існуючих систем за умови її постійного вдосконалення та підтримки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Біржа спостережень, аналіз предметної галузі

Біржа спостережень, в якій можна створювати замовлення на дослідження даних, може стати важливим інструментом для аналітиків, дослідників чи компаній. Ця система надає можливість користувачам публікувати завдання для збору, обробки та аналізу даних за певними критеріями. Аналізуючи платформу, можна виділити декілька ключових аспектів ІС:

- можливість замовлення різних досліджень. Система надає користувачам можливість створити дослідження за різними категоріями, починаючи від проведення аналізу фінансових даних до вивчення публічних думок;

- ринок конкуренції та сприяння інноваціям. Біржа спостережень може стати конкурентною системою для аналітиків і дослідників, що сприяє розвитку нових методів та технологій у сфері аналізу даних;

- можливість здійснення замовлень за індивідуальними потребами. Користувачі мають можливість встановити критерії для дослідження з урахуванням особистих потреб;

- забезпечення безпеки даних. ІС забезпечує високий рівень захисту даних для замовників, що дозволить їм відчувати те, що їх замовлення захищені.

Проаналізувавши головні аспекти системи, можна сказати, що вона забезпечує доступ до мінімальних потреб користувачів та бізнесу. Дослідження даних буде проводитися актором системи, який буде приймати замовлення методом підписки на певну категорію задач з можливістю відписатися від неї залежно від ситуацій. Крім того, біржа спостережень забезпечує конфіденційність отриманих результатів, що були досліджені підписником та надані замовнику в обробленому вигляді. Така платформа

може стати важливим інструментом для користувачів, що зможе допомогти прийняти стратегічні рішення чи отримати прогнози, відповідно до свого запиту.

Окрім аналізу самої системи, необхідно проаналізувати ринок конкурентів та переваги для створення інформаційної системи. Це включає в себе дослідження тенденцій, попиту та кількості учасників на ринку. Головною метою є визначення потреб користувачів і розробка ефективних стратегій для подальшого розвитку:

- на ринку спостерігається стабільний ріст попиту на обробку даних і аналітику з візуальним представленням;
- відбувається зростання популярності спостережень завдяки їх швидкості в аналізі даних;
- потрібне розширення сфер застосування подібних бірж спостережень, що буде включати можливість обрати необхідну категорію досліджень;
- конфіденційність даних є ключовим фактором для будь-якої інформаційної системи, оскільки кількість особистих даних може зростати;
- зростає попит на аналіз зображень та відео для виявлення аномалій та підвищення безпеки;
- клієнтам потрібна більша точність досліджень та швидкість обробки даних;
- зі збільшенням обсягу даних потрібно створити ефективні алгоритми, що дозволять швидко отримувати інформацію.

Отже, подібний аналіз ринку допомагає визначити потреби потенційних клієнтів та розробити ефективні стратегії, а також приймати обґрунтовані рішення для подальшого розвитку інформаційної системи.

1.2 Біржа спостережень та аналіз існуючих систем

У 21-му столітті, де величезні обсяги інформації постійно створюються і оброблюються, біржі спостережень стають ключовими системами для збору

і аналізу цих даних. Для аналізу існуючих систем було обрано SVO, що представляє собою віртуальну обсерваторію з можливістю дослідження космічних об'єктів.

SVO – це веб-ресурс для зберігання і відображення даних за кривим блиску небесних об'єктів.

Після авторизації в системі одразу потрапляємо на сторінку «список телескопів». Сторінка представлена у вигляді інтерфейсу користувача з необхідним функціоналом системи, з яким буде взаємодіяти користувач. Дана сторінка подана на рисунку 1.1.

The screenshot shows the SVO interface with the following data in the 'Telescopes' table:

Name	Observatory	Aperture	Longitude	Latitude	Altitude	Fits Count
508	Kolonica	508	0	0	0	
Alica telephoto lens 300 mm	Kolonica		0	0	0	
C11 280/2800	Kolonica	280	0	0	0	
C14 356/3910	Kolonica	356	0	0	0	
Cassegrain 400/2032	Roztoky		0	0	0	
EABA 1540/7485	Cordoba		0	0	0	
Hugo 270/1360	Kolonica	265	0	0	0	
Lichtencknecker	Kolonica		0	0	0	
LX200	Kolonica		0	0	0	
Newton 100/500	Kolonica		0	0	0	

Рисунок 1.1 – Сторінка «Список телескопів»

На сторінці «Список телескопів» представлені такі елементи інтерфейсу:

- пошук супутника за міжнародним номером. Пошук здійснюється за частковим збігом;
- додавання нового телескопа (для авторизованих користувачів);
- список доступних телескопів. Натискання на стрілку змінює колонку та напрямок сортування;

- редагування інформації про телескоп;
- інформація про телескоп.

Сторінка «Управління користувачами» надає можливість адміністратору системи відкривати або блокувати доступ до входу в систему користувачам. Для того, щоб мати доступ до входу в систему, кожен користувач має бути активованим. Інтерфейс сторінки подано на рисунку 1.2.

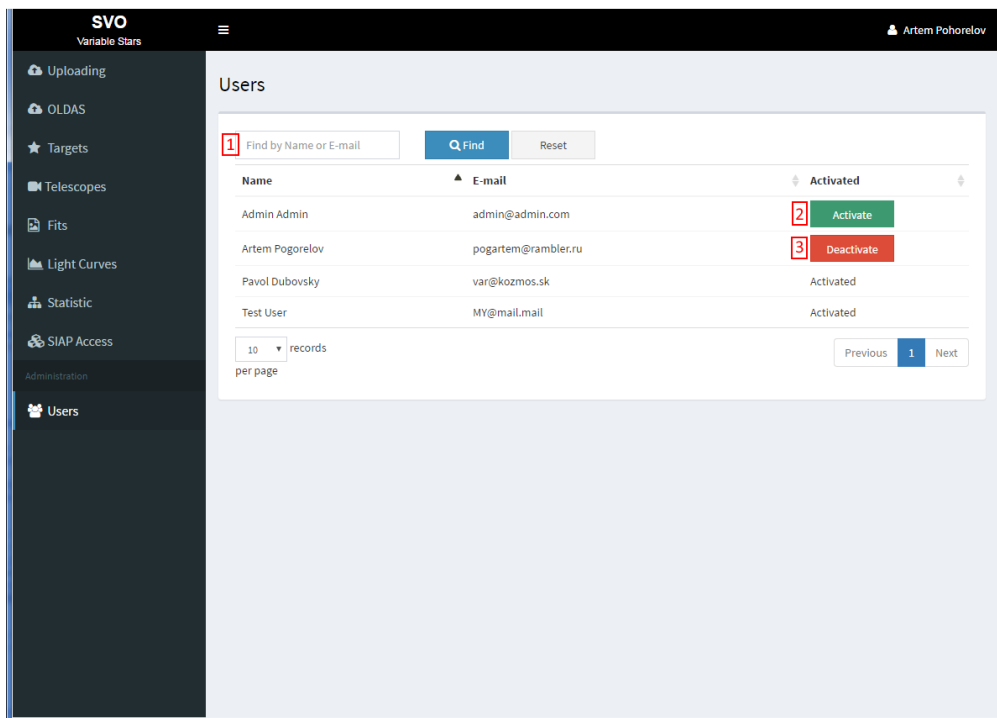


Рисунок 1.2 – Сторінка «Управління користувачами»

Сторінка «Управління користувачами» містить у собі такий функціонал:

- пошук користувача за ім'ям, прізвищем або e-mail;
- можливість деактивації раніше активованого користувача. Користувач більше не матиме можливості входу до системи;
- можливість активації нового або існуючого користувача в системі.

Головні функції будь-якої інформаційної системи – це реєстрація та авторизація, без яких користувачі не зможуть використовувати функціонал системи для своїх цілей. Даний функціонал представлений у вигляді

модального вікна, що не потребує у створенні додаткової веб-сторінки. Модальне вікно реєстрації та авторизації подані на рисунках 1.3 та 1.4.

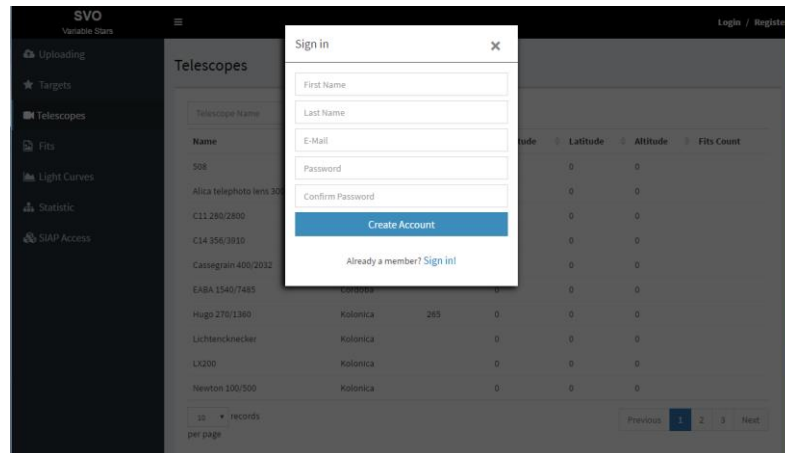


Рисунок 1.3 – Модальне вікно «Реєстрація»

Реєстрація в системі відбувається за відкритою формою, що потребує введення наступних даних:

- ім'я користувача;
- прізвище користувача;
- електронна адреса для реєстрації в системі;
- пароль та підтвердження паролю.

Якщо реєстрація відбулася успішно за введеними даними, то користувач отримає повідомлення «Ви були успішно зареєстровані в системі».

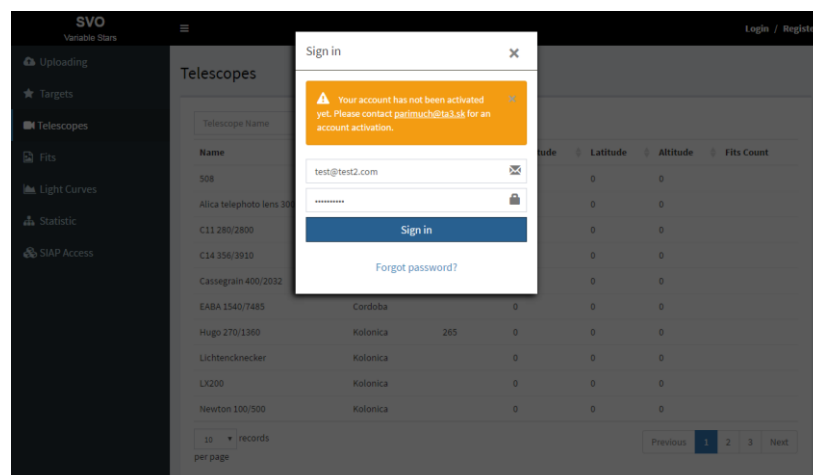


Рисунок 1.4 – Модальне вікно «Авторизація»

У випадку, коли користувач хоче авторизуватися, йому необхідно ввести електронну адресу та пароль. Якщо його особистий профіль не активовано, він отримає повідомлення про те, що потрібно чекати, коли адміністратор активує доступ до системи, що графічно представлено на рисунку 1.4.

Кожній системі необхідно проводити збір та аналіз статистичних даних, які відображають взаємодії користувачів та інших акторів у межах екосистеми додатку, що продемонстрована на рисунку 1.5. На сторінці статистики відображається інформація про загальну кількість телескопів у системі, об'єкти та криві блиску, що зберігаються в системі.

Statistic

Telescopes: 22

Targets: 547

Light Curves: 4885

From: Start Date To: End Date

ID	Observatory	Telescope	Instrument	Filter	Upload Date
1	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
2	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
3	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
4	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
5	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
6	Kolonica	photolense 400mm	Meade DSI Pro	Clear	2017-02-27
7	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
8	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
9	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27
10	Kolonica	Lichtencknecker	Meade DSI Pro	Clear	2017-02-27

10 records per page

Previous 1 2 3 4 5 ... 489 Next

Рисунок 1.5 – Сторінка «Статистика»

Останнім важливим інтерфейсом системи може стати сторінка «Криві блиску» та інші з подібним інтерфейсом і функціоналом. На ній можна переглянути криві блиску по всіх об'єктах, які доступні в системі. При необхідності переглянути детальну інформацію про криву блиску, необхідно

натиснути на запис та отримати відображення інформації. Криві блиску та записи подані на рисунках 1.6 та 1.7.

ID	Target	Observatory	Observer	Telescope	Instrument	Filter	Start Date	End Date
1	V0728 Her	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-07 20:57:07	2006-04-08 02:42:43
2	BE Dra	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-08 23:15:56	2006-04-09 01:56:56
3	IV UMa	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-08 19:50:53	2006-04-08 22:37:55
4	gsc1588	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-18 00:03:55	2006-04-18 01:23:01
5	RZ Com	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-19 20:56:27	2006-04-20 00:58:06
6	EF Dra	Kolonika	Vadila	photolense 400mm	Meade DSI Pro	Clear	2006-04-22 22:11:06	2006-04-23 01:47:20
7	TV UMI	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-22 19:53:53	2006-04-23 02:18:28
8	TV UMI	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-24 21:22:57	2006-04-25 01:37:55
9	V0844 Her	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-25 21:14:06	2006-04-26 00:56:08
10	V0844 Her	Kolonika	DPV	Lichtenknecker	Meade DSI Pro	Clear	2006-04-26 20:05:03	2006-04-27 00:55:19

Рисунок 1.6 – Записи «Криві блиску»

Графічне відображення кривої блиску дозволить провести її аналіз та отримати необхідну інформацію для її використання. На цій сторінці ми можемо переглянути графік кривої блиску за двома параметрами та можливістю обрати певний фрагмент кривої.

Сторінка з детальним зображенням кривої блиску має такі елементи інтерфейсу:

- інформація про об'єкт, що спостерігається;
- завантаження файлу вимірів;
- формат підписів вздовж осі абсцис – JD/DateTime;
- графічне зображення кривої блиску;
- виділення області для збільшення перегляду кривої блиску;
- нотатки кривої блиску;
- завантаження .note файлу.

Даний аналіз функціоналу системи дозволить створити індивідуальні вимоги до інформаційної системи з урахуванням потреб користувачів та бізнесу [7].

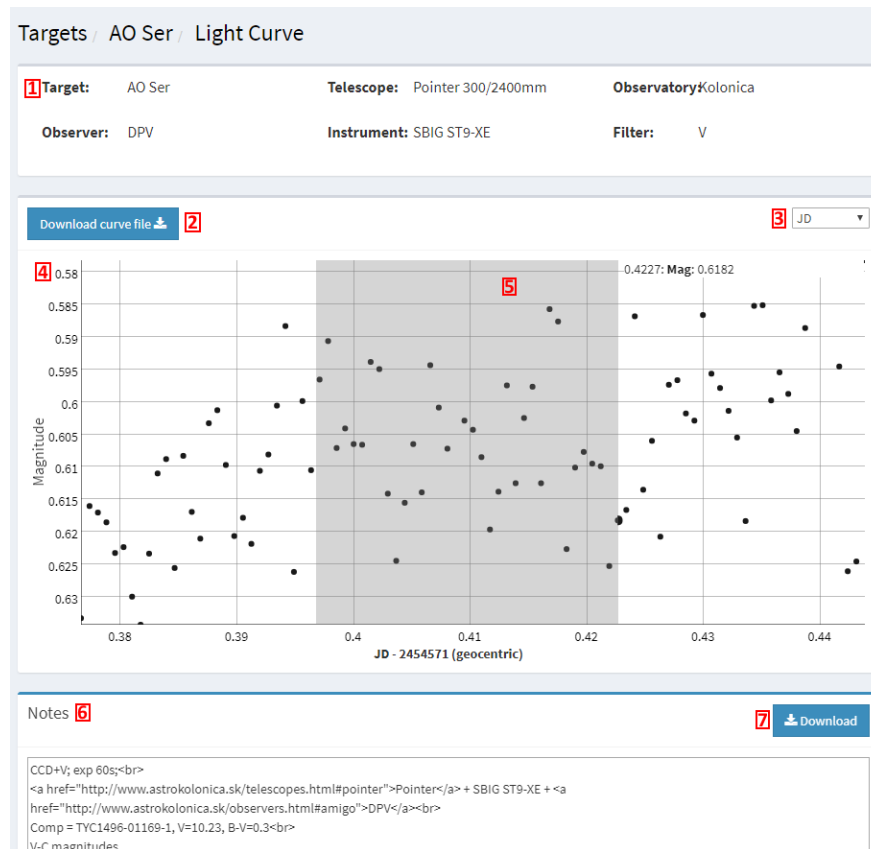


Рисунок 1.7 – Детальне зображення кривої блиску.

1.3 Аналіз інформаційної системи біржа спостережень

В ході аналізу інформаційної системи було отримано впевнені знання предметної області та розуміння необхідного функціоналу системи, які потребують користувачі та бізнес. Розроблена система допоможе користувачам та бізнесу створювати особисті повідомлення для дослідження інформації з можливістю отримання її в обробленому вигляді для комфортного перегляду і аналізу.

Для реалізації системи було обрано шаблон проектування "Видавець – підписник", який має такі переваги:

– розділення UML обов’язків. Шаблон дозволяє розділити обов’язки між видавцем та підписником. Видавець не повинен знати, хто є підписником його повідомлень, що дозволяє зробити систему менш залежною та більш гнучкою для масштабування;

– підтримка реактивності. Використовуючи цей шаблон, можна легко створювати реактивні системи, де при зміні стану одного елементу викликаються певні дії у інших об’єктах системи;

– гнучкість та масштабованість. Додавання нових видавців і підписників не потребує написання додаткового коду та не вимагає змін в існуючому.

Загалом, шаблон проектування є потужним інструментом для організації взаємодії компонентів системи, що є ключовим аспектом при розробці системи з використанням принципів модульності, гнучкості та полегшення комунікації між компонентами інформаційної системи.

Окрім переваг серед вибору шаблону проектування, не менш важливим є визначення головних акторів системи. Актор представляє собою об’єкт, який взаємодіє з системою, виконуючи певні функції або роль у системі. В контексті програмного забезпечення, актором може бути користувач, зовнішній сервіс або програма, яка виконує певні інструкції в ІС.

В розробленій інформаційній системі представлені три головних актори, що взаємодіють з системою та виконують свої ролі:

– видавець (замовник). Особа, що має можливість створити повідомлення за певними критеріями для отримання результатів дослідження;

– підписник (спостерігач). Актор системи, що оформлює підписку на виконання дослідження або за непередбачуваних обставин може відмінити підписку, якщо дана проблема допускається для відміни дослідження;

– адміністратор даних. Перевіряє обладнання спостерігачів, що було відправлене на отримання сертифікації.

Представлені типи акторів обов’язково повинні бути присутні в системі, оскільки без них розроблена біржа спостережень не зможе виконувати головний бізнес-процес. Мета головного бізнес-процесу полягає в

забезпеченні високоякісного та оперативного аналізу даних з візуальним представленням для комфортної взаємодії видавця з результатом дослідження.

1.4 Постановка вимог та задач до біржі спостережень

Візуалізація інформаційної системи є обов'язковою складовою під час проектування будь-якого додатку. Створення діаграм для біржі спостережень має кілька переваг:

- вимоги користувачів. Візуалізація дозволить краще зрозуміти потреби та очікування користувачів;
- дизайн. Методом візуалізації системи можна на ранніх етапах виправити можливі проблеми інтерфейсу і дизайну системи;
- узгодження вимог. Візуалізація дозволить обговорити та узгодити вимоги до інформаційної системи між зацікавленими сторонами, такими як клієнти та розробники для вибору оптимальних рішень;
- комунікація команди. Візуалізація дозволить членам команди зрозуміти загальну інформацію про систему та її компонентів.

Візуалізація інформаційної системи є досить важливою складовою під час проектування додатків, оскільки допоможе розробникам чітко розуміти свої завдання для створення системи.

Наступним кроком після створення візуалізації процесів та інтерфейсу майбутньої системи буде створення інтерфейсу користувача та бізнес-логіки додатку. Інтерфейс користувача визначає спосіб взаємодії користувача з системою, його зручність та ефективність, що була розроблена з урахуванням потреб користувачів. У той же час, бізнес-логіка додатку використовує механізми та алгоритми обробки, які забезпечують функціонування системи.

Розробка інтерфейсу користувача і бізнес-логіки передбачає виконання наступних завдань:

- вибір технологій проектування інформаційної системи;

- проектування бази даних з використанням системи управління базами даних MySQL;
- розробка загального та індивідуального інтерфейсу користувача, що доступний для окремих ролей в системі;
- розробка моделей, що будуть зберігати інформацію та надавати її користувачам;
- розробка бізнес-логіки авторизації та реєстрації в системі;
- розробка функціоналу створення повідомлень з можливістю введення критеріїв дослідження;
- розробка функцій підписки та відписки від виконання замовлення;
- розробка інтерфейсу, що дозволить завантажувати не тільки текстові дані, а також файли різних форматів, як для замовників так і спостерігачів;
- розробка сторінки спостерігача для отримання сертифікації свого обладнання на якій необхідно заповнити форму та отримати відповідь від адміністратора даних;
- розробка індивідуального функціоналу, що доступний тільки для адміністратора даних.

Під час розробки інтерфейсу користувача і бізнес-логіки було вирішено ключові завдання, що спрямовані на створення необхідного функціоналу системи для повноцінної роботи. Вибір технологій та розробка бази даних з розробкою інтерфейсу користувачів, що доступний певним ролям в системі може забезпечити зручність для користувачів у використанні доступних функцій.

Отже, виконання поставлених завдань, а саме розробку інтерфейсу користувача, бізнес-логіки та проектування бази даних на платформі MySQL, було успішно завершено, забезпечивши біржу спостережень необхідними можливостями, що відповідають потребам користувачів.

2 РОЗРОБКА ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Системне проектування біржі спостережень

Системне проектування – це складний і важливий процес розробки будь-якої складної інформаційної системи, що дозволяє чітко сформулювати цілі, визначити оптимальну архітектуру та розробити покроковий план реалізації системи [4]. Під час системного проектування здійснюється аналіз різноманітних аспектів системи, а саме масштабованість, безпека даних, швидкодія, інтуїтивно зрозумілий інтерфейс, послідовність дій та багато іншого. Більшість уваги під час системного проектування приділяється вибору технологій, інструментів розробки та проектуванню архітектури системи за допомогою спеціальних методів і інструментів візуалізації складових та взаємозв'язків системи.

Візуалізація компонентів та інших складових систем, відбувається за допомогою інструменту UML (Unified Modeling Language). UML – уніфікована мова моделювання, що являється загальноприйнятим стандартом для візуального представлення інформаційних систем [12].

Системне проектування з використанням UML включає в себе процес розробки діаграм системи, таких як:

- діаграма класів. Відображає класи системи, атрибути та методи;
- діаграма послідовності. Демонструє послідовність дій між об'єктами системи під час виконання задачі;
- діаграма кооперацій. Демонструє взаємодію об'єктів для виконання функції;
- діаграму варіантів використання. Демонструє відношення між акторами та прецедентами в системі;
- діаграма станів. Відображає різні стани об'єктів системи та можливі переходи станів під час виконання дій.

Отже, уніфікована мова моделювання є важливим інструментом для розробки інформаційних систем під час системного проектування, яка дозволить зробити зрозумілі та надійні системи і можливістю визначити недоліки системи на ранніх етапах проектування.

2.2 Проектування функціональної моделі IDEF0 для інформаційної системи біржі спостережень

Переходячи до етапу проектування функціональних і нефункціональних вимог до інформаційної системи було поставлено запитання, що є метою проектування функціональної моделі? Проаналізувавши, можна сказати, що мета проектування функціональної моделі IDEF0 полягає в створенні чіткого та структурованого опису процесів або організації декількох систем одночасно, для того, щоб полегшити вдосконалення та оптимізацію. Спроектвана модель дозволяє представляти у вигляді ієрархії складні системи та взаємопов'язані функції, що сприяють розумінню основних її компонентів.

Використання функціональної моделі IDEF0 допомагає виявити проблемні місця в процесах та ідентифікувати неефективність, а також визначити можливості для вдосконалення процесів та функцій інформаційної системи [17]. Даний етап працює через визначення входів, виходів, контролю та механізмів, які взаємодіють з процесами, що забезпечує ясність та прозорість взаємодій в ІС та полегшує спілкування між різними частинами проекту.

Отже, головна мета проектування функціональної моделі IDEF0 полягає у створенні інструменту, який допомагає зрозуміти, оптимізувати та документувати складні процеси та функції системи, роблячи їх зрозумілими та більш ефективними для розуміння команди.

Спроектвана функціональна модель для біржі спостережень подана на рисунку 2.1, та містить наступні стрілки, що поділені за категоріями:

Входи:

- введення інформації користувачами;
- введення критеріїв дослідження;
- поповнення рахунку користувача;
- підписка / відписка на дослідження даних;
- оплата дослідження;
- заповнення інформації для сертифікації обладнання;
- завантаження результатів проведеного дослідження;
- отримана сертифікація.

Виходи:

- збереження профілю;
- створене дослідження;
- збільшення балансу користувача;
- запит на сертифікацію;
- отримання коштів за дослідження;
- виставлення рахунку за дослідження;
- результат дослідження.

Механізми:

- ПЗ;
- замовник;
- дослідник;
- адміністратор даних;
- плагіни.

Управління:

- правила проведення сертифікації обладнання;
- нормативні документи;
- керівництво користувача.

На момент проектування функціональна модель має такий вигляд, але в ході її аналізу до проведення декомпозиції, деякі взаємозв'язки можуть змінюватись.

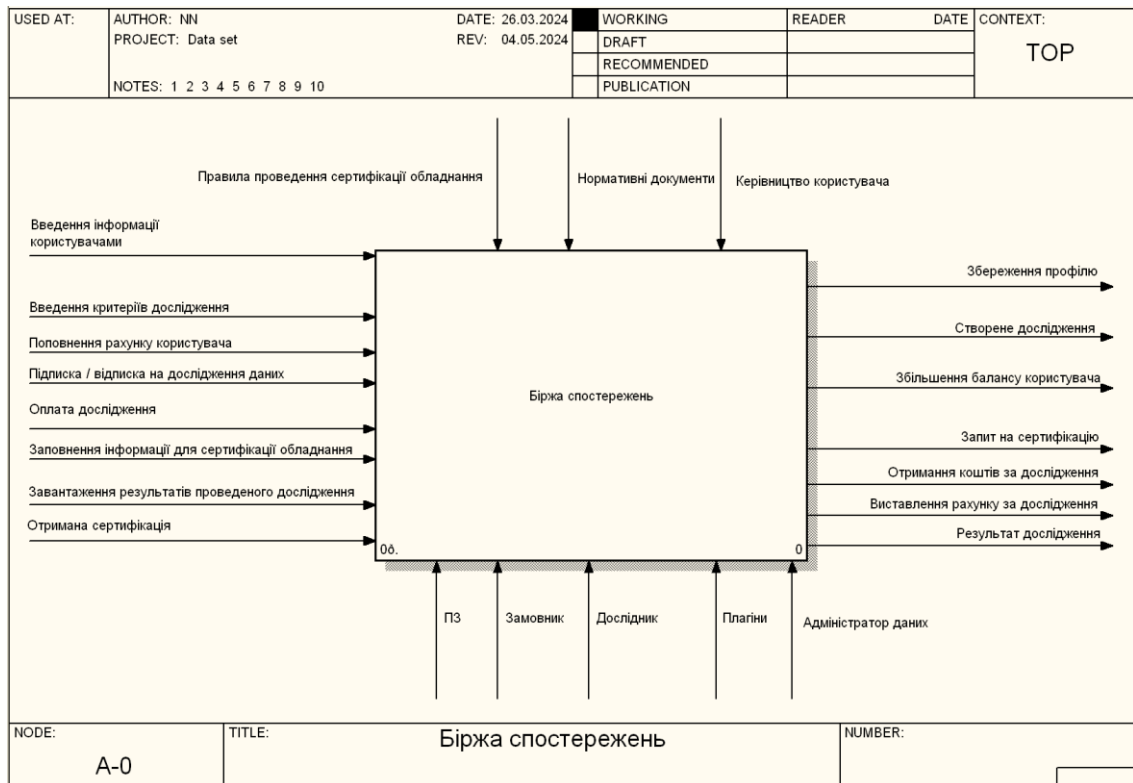


Рисунок 2.1 – Функціональна модель біржі спостережень

Декомпозиція є невід’ємною частиною при проектуванні функціональної моделі, оскільки її мета полягає у деталізації та поділі складних систем або процесів на менші компоненти, що полегшить аналіз та розуміння ІС. Декомпозиція допомагає відобразити складні зв’язки на більш високому рівні та поступово деталізувати їх, що дозволить глибше розуміти систему або процес [3].

Такий підхід дозволяє визначити і дослідити всі елементи системи, визначити функції та зрозуміти, як вони взаємодіють між собою. Саме декомпозиція надає можливість зосередитись на певних аспектах системи, що спрощує вдосконалення системи.

Отже, мета декомпозиції функціональної моделі полягає в тому, щоб полегшити розуміння складних процесів забезпечуючи структурний підхід до аналізу та оптимізації взаємозв’язків в системі та створити чітку основу для ефективного планування та впровадження змін [5]. Діаграма декомпозиції «Біржа спостережень» наведена на рисунку 2.2.

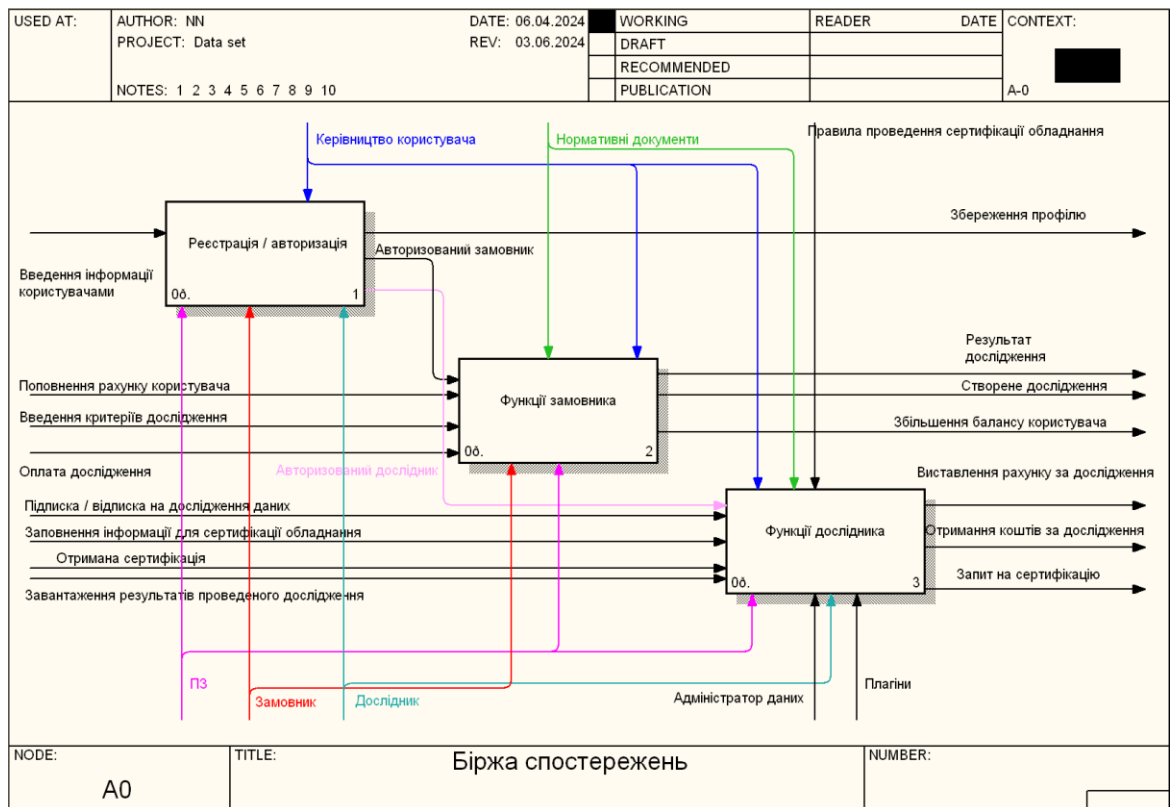


Рисунок 2.2 – Декомпозиція функціональної моделі «Біржа спостережень»

Проведена декомпозиція для біржі спостережень була поділена на три основні функціональні блоки:

- реєстрація / авторизація. Реєстрація та авторизація в інформаційних системах є найважливішою функцією, оскільки надають акторам системи можливість отримати додатковий функціонал для взаємодії з системою. Відвідуючи сторінку реєстрації або авторизації, користувачеві необхідно ввести дані у форму для створення особистого аккаунту або входу в систему;

- функції замовника. Головною метою замовника є створення досліджень, на які будуть підписуватися дослідники та виконувати за вказаними критеріями. Цей процес вимагає ретельного планування та організації з боку замовника, оскільки йому необхідно визначити критерії та специфіку дослідження, а також надати інформацію про очікувані результати. З іншого боку, надання можливості завантажувати файли до інформаційної системи є важливим аспектом для створення замовлень;

– функції дослідника. Дослідник є критично важливим актором у контексті біржі спостережень і відіграє центральну роль у її функціонуванні. Його робота є невід’ємною частиною системи, оскільки без взаємодії з дослідником основна бізнес-функція системи не може існувати. Мета дослідника полягає в отриманні інформації та критеріїв від замовника та здійсненні дослідження даних відповідно до потреб замовника. Після збору та аналізу даних дослідник має подбати про те, щоб надати результати дослідження у зручному форматі, який буде зрозумілий для використання.

Наступним кроком розглянемо декомпозицію бізнес-функції «Реєстрація / авторизація», яка забезпечує користувачам доступ до системи та є першим кроком у встановленні зв’язку між користувачем і інформаційною системою. Розроблена декомпозиція функціонального блоку подана на рисунку 2.3.

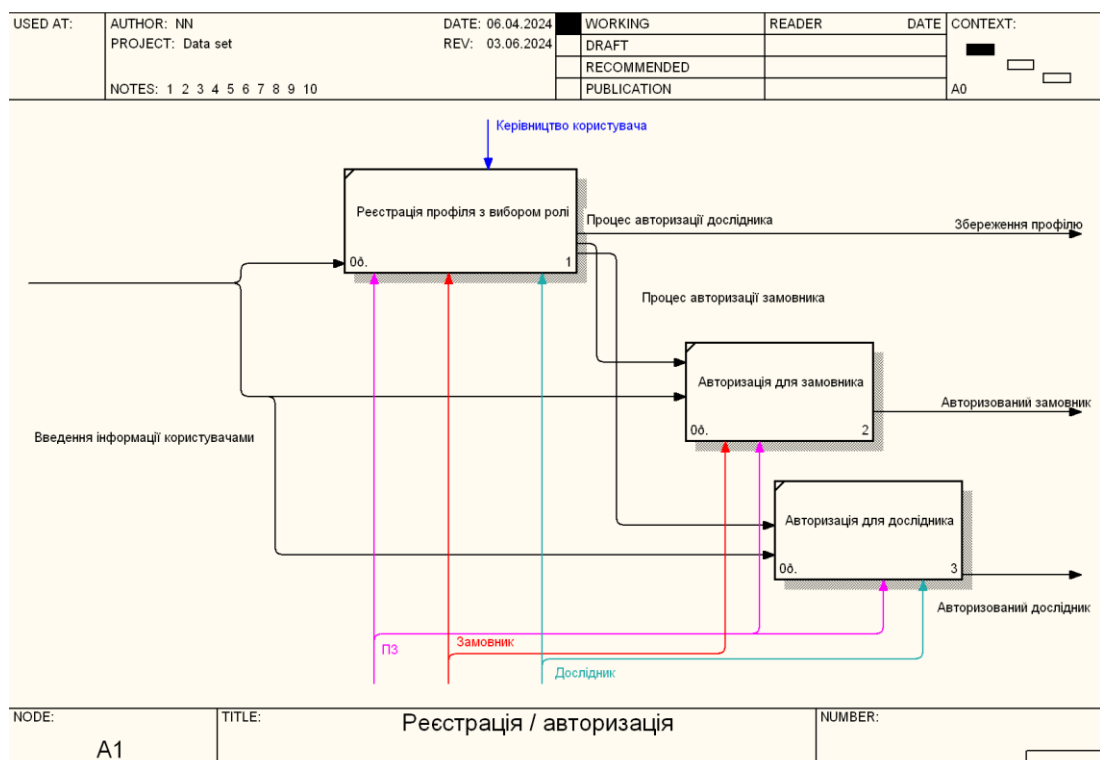


Рисунок 2.3 – Декомпозиція бізнес-функції «Реєстрація / авторизація»

Аспектом реалізації бізнес-функції є створення профілів користувачів з можливістю обрати роль в інформаційній системі. Далі наведено визначені функціональні блоки та їх опис:

– реєстрація профілю з вибором ролі. Реєструючись в інформаційній системі біржа спостережень, важливо мати можливість обрати ролі користувача. Дана опція дозволить користувачам обрати своє місце в системі та отримати доступ до конкретних функцій. Обрана роль буде впливати на завдання та обов'язки користувача, яких необхідно дотримуватись під час взаємодії з системою. Наприклад, адміністратор даних зможе проводити сертифікацію обладнання дослідника, перевіряючи чи відповідає вимогам та стандартам безпеки обладнання, що буде використовуватись для проведення дослідження та аналізу даних;

– авторизація замовника. Пройшовши авторизацію користувач отримує роль «замовник». Отримана роль надає йому права та доступ до певного набору функцій і можливостей системи. Процес авторизації передбачає перевірку користувача, щоб впевнитися, що він вже зареєстрований та має свою роль в системі. Пошук зареєстрованого профілю відбувається за допомогою електронної адреси, а процес авторизації використовує для перевірки електронну адресу та пароль користувача. Отже, даний процес для замовника є складним і вимагає уваги до деталей системи, оскільки він визначає не лише доступ до функцій та можливостей, але і безпеку та захист даних в системі;

– авторизація дослідника. Пройшовши авторизацію користувач отримує роль «дослідник». Отримана роль надає йому права та доступ до певного набору функцій і можливостей системи, які будуть пов'язані з переглядом та взаємодією з певними аспектами досліджень. Роль дослідника зазвичай має обмежений функціонал, але важливий набір для проведення досліджень. Наприклад, дослідник може переглядати критерії обраного дослідження, на яке було оформлено підписку. Перегляд критеріїв дозволяє досліднику

слідкувати за процесом виконання дослідження та завжди мати можливість переглянути інформацію про цілі, методи та результати дослідження.

Наступним кроком було прийняте рішення провести декомпозицію функціонального блоку «Функції замовника». Декомпозиція допоможе детально отримати інформацію про основний функціонал замовника та як функціонал буде використовуватись для досягнення певної мети. Декомпозиція «Функції замовника» наведена на рисунку 2.4.

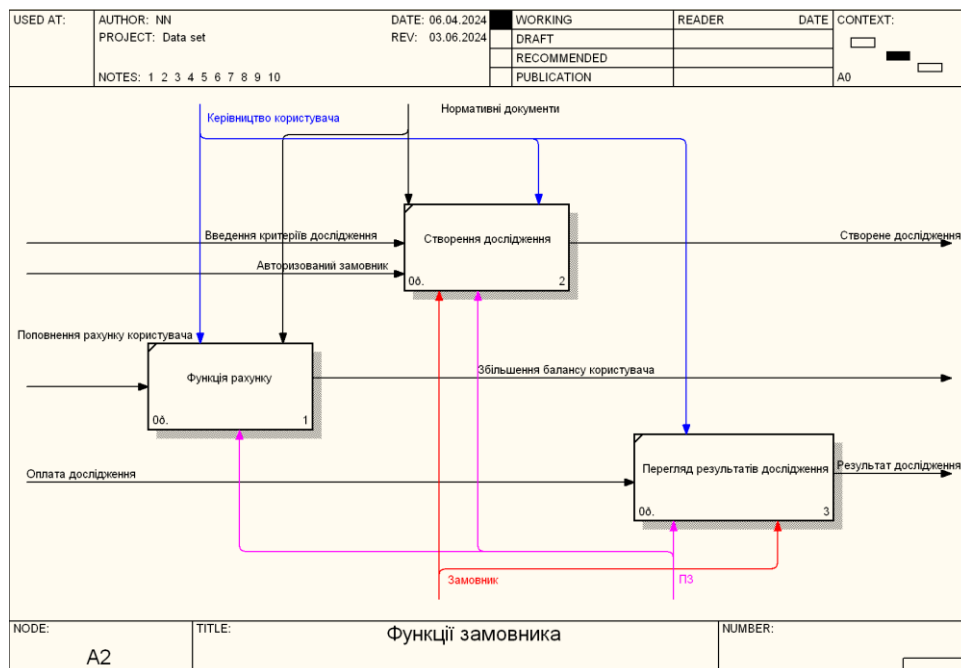


Рисунок 2.4 – Декомпозиція бізнес-функції «Функції замовника»

Аналізуючи результат декомпозиції «функції замовника», можна визначити вимоги до інформаційної системи та розглянути на нижчому рівні, як відбувається взаємодія об'єктів та даних в системі. Декомпозиція містить наступні функціональні блоки нижчого рівня:

– створення дослідження. Мета функції полягає в отриманні інформації та критеріїв дослідження замовником, які необхідно ввести в форму та надіслати до інформаційної системи для занесення інформації до бази даних та паралельно додавання дослідження до списку нових досліджень. Реалізація відбувається з використанням веб-технологій та технологій розробки

серверної частини застосунку з використанням патернів проектування та елементів інтерфейсу користувача;

– функція рахунку. Надає можливість збільшити кількість коштів на особистому рахунку, поповнюючи його через веб-застосунок банку за вказаними реквізитами. Кошти на рахунку користувача є важливим елементом системи, оскільки для отримання результатів дослідження замовнику необхідно провести оплату за виконане дослідження, яке буде надано як результат виконаної роботи в обробленому форматі для подальшої взаємодії;

– перегляд результатів дослідження. Надає доступ до завантаження або відкриття результатів проведеного дослідження. Перегляд результату обробки є кінцевим етапом взаємодії з ІС біржа спостережень. Сторінка результатів представлена у вигляді веб-сторінки з відображенням результатів наукового дослідження, що надає постійний доступ до результату, де замовники можуть досліджувати та вивчати деталі та особливості звіту.

Кінцевим етапом дослідження та аналізу функціональної моделі є проведення декомпозиції «Функції дослідника». Мета декомпозиції полягає у створенні діаграми нижчого рівня для уточнення взаємодії процесів, функцій та потоків даних моделі. Бізнес-функція надає доступ досліднику ефективно взаємодіяти з системою та проводити дослідження даних за вказаними критеріями та вимогами. Початком виконання дослідження завжди є отримання сертифікації свого обладнання, яке перевіряє адміністратор даних та визначає позитивний або негативний результат відповідно наданої інформації. Отримавши сертифікацію обладнання дослідник може обирати для себе дослідження на які буде оформлювати підписку з доступного переліку завдань. Провівши дослідження досліднику обов'язково необхідно провести форматування результатів для створення звіту. Процес форматування результатів є дуже важливим, оскільки він визначає, наскільки зрозумілими і коректними будуть кінцеві висновки.

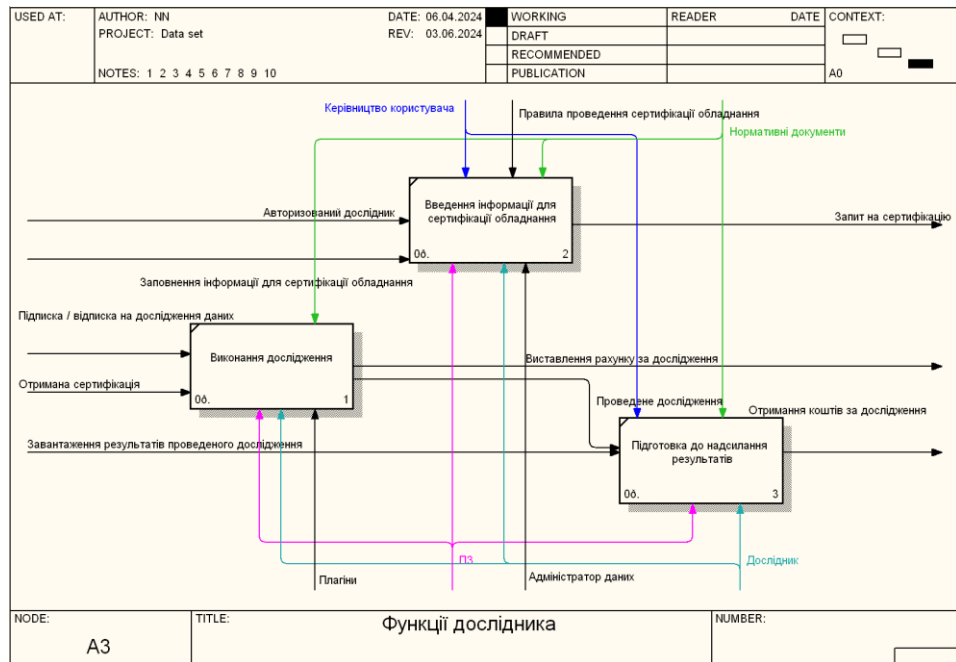


Рисунок 2.5 – Декомпозиція бізнес-функції «Функції дослідника»

На діаграмі декомпозиції «Функції дослідника» можна побачити три функціональні блоки, що відповідають головним можливостям взаємодії для дослідника системи:

– введення інформації для сертифікації обладнання. Початковим етапом взаємодії дослідника з системою є введення інформації про своє обладнання та надсилання запиту на отримання сертифікації, де обладнання повинно відповідати вимогам та нормам, визначеними бізнесом. Отримавши сертифікацію, досліднику стає доступною сторінка всіх замовлень, на які можна оформити підписку, що буде свідчити про те, що було взято дослідження даних на виконання та аналіз. Процес сертифікації буде гарантувати, що обладнання дослідника відповідає всім вимогам та стандартам безпеки;

– виконання дослідження. Маючи сертифікацію свого обладнання, дослідник отримує доступ до нового функціоналу системи. Головною функцією є оформлення підписки або відписки від дослідження даних із можливістю встановити ціну за проведення аналізу даних;

– підготовка до надсилання результатів. Завершальним етапом є підготовка результатів до відправки замовнику. Щоб забезпечити найвищу якість, необхідно приділити особливу увагу правильному оформленню звіту. Оформлення звіту має відповідати стандартам галузі, що включає в себе форматування, шрифт, розмір тексту, інтервали між рядками та загальну структуру документу.

Підсумовуючи та аналізуючи розробку функціональної моделі IDEF0 та її декомпозицій, в ході роботи було визначено головні бізнес-функції системи та визначено вимоги до розробки інформаційної системи біржі спостережень. Встановлено, що розроблена система повинна забезпечувати гнучкість, оскільки біржа спостережень має різні типи користувачів, які мають індивідуальний функціонал в системі, що буде відповідати потребам і вимогам користувачів. Таким чином, забезпечення модульності та гнучкості в розробці біржі спостережень є не лише технічним завданням, але і бізнес-необхідністю, що дозволить в майбутньому оновлювати систему.

На рисунку 2.6 подано діаграму дерева вузлів, що була розроблена на основі функціональної моделі біржа спостережень. Діаграма дерева вузлів є інструментом візуалізації, який використовується для моделювання і зображення процесу у вигляді ієрархічної структури, де кожен вузол відповідає за певний процес або елемент, а зв'язок між вузлами демонструє їх відношення.

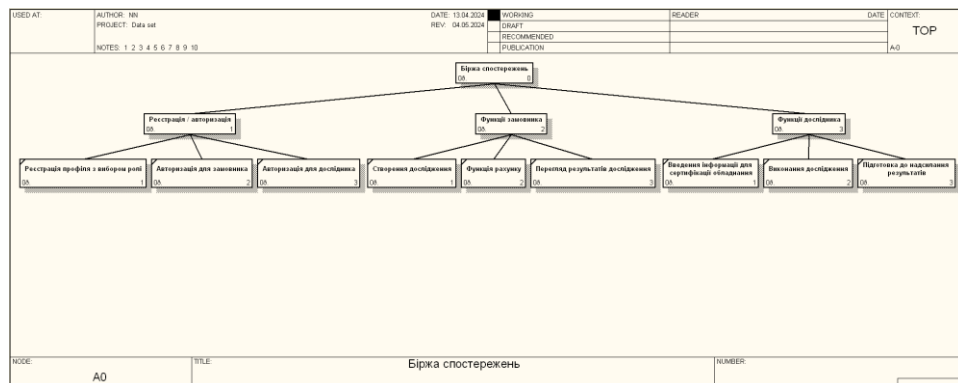


Рисунок 2.6 – Діаграма дерева вузлів біржі спостережень

Діаграма дерева вузлів зазвичай починається з кореневого вузла. З нього виходять гілки, які можуть розділитися на відгілки, утворюючи таким чином дерево вузлів, де кінцева гілка представляє результат процесу. На діаграмі дерева вузлів подано такі вузли:

- біржа спостережень;
- реєстрація / авторизація;
- функції замовника;
- функції дослідника;
- реєстрація профілю з вибором ролі;
- авторизація для замовника;
- авторизація для дослідника;
- створення дослідження;
- функція рахунку;
- перегляд результатів дослідження;
- введення інформації для сертифікації обладнання;
- виконання дослідження;
- підготовка до надсилання результатів.

В ході проектування та аналізу функціональної моделі та діаграми дерева вузлів, було визначено наступні функції системи, що підлягають реалізації:

- реєстрація та авторизація;
- створення нового дослідження;
- поповнення особистого рахунку;
- проведення оплати за дослідження;
- отримання результатів дослідження;
- створення запиту на сертифікацію обладнання;
- підписка та відписка від проведення дослідження;
- виставлення рахунку за дослідження;
- завантаження звітів дослідження;
- перегляд всіх запитів на сертифікацію;

– проведення сертифікації обладнання.

Проектуючи інформаційну систему, аналітики повинні чітко визначити вимоги до застосування, щоб забезпечити потреби користувачів та бізнесу. Від чітко визначених вимог до ІС залежить успіх бізнесу та здатність системи ефективно виконувати поставлені задачі. Детальний аналіз і чітке визначення вимог є ключовими складовими процесу розробки інформаційної системи. Коли вимоги добре визначені і детально описані, це створює міцний фундамент для всього проекту. Цей процес вимагає ретельного вивчення бізнес-процесів, аналізу потреб кінцевих користувачів, а також постійної взаємодії з усіма зацікавленими сторонами для забезпечення повного розуміння та задоволення їхніх вимог.

2.3 Проектування моделі потоків даних біржі спостережень DFD

На підставі функціональної моделі та її декомпозицій було розроблено діаграму потоків даних з використанням нотації DFD. DFD (Data Flow Diagram) – це діаграма, яка використовується для візуалізації потоку даних в інформаційній системі. Діаграма допомагає зрозуміти, як інформаційні потоки та набори даних рухаються між процесами, сутностями та сховищами даних.

Метою DFD діаграми є забезпечення способу моделювання взаємодії потоків даних ІС, що допоможе зрозуміти складні процеси. Однією з ключових переваг DFD є його зрозумілість і простота. Візуалізація потоку даних допомагає виявити вузькі місця, потенційні проблеми з продуктивністю, а також забезпечує засіб для ідентифікації можливих удосконалень. DFD також може бути інструментом для документування системи, допомагаючи зафіксувати поточний стан і використовувати це як основу для подальших розробок чи змін.

Загалом, DFD-діаграма — це ефективний інструмент для моделювання інформаційних систем, дозволяючи зрозуміти їх структуру, функціональність і динаміку даних [22].

На рисунку 2.7 представлена діаграма потоків даних, що була розроблена на основі функціональної моделі біржі спостережень для чотирьох сутностей «неавторизований користувач», «дослідник», «замовник» та «адміністратор даних».

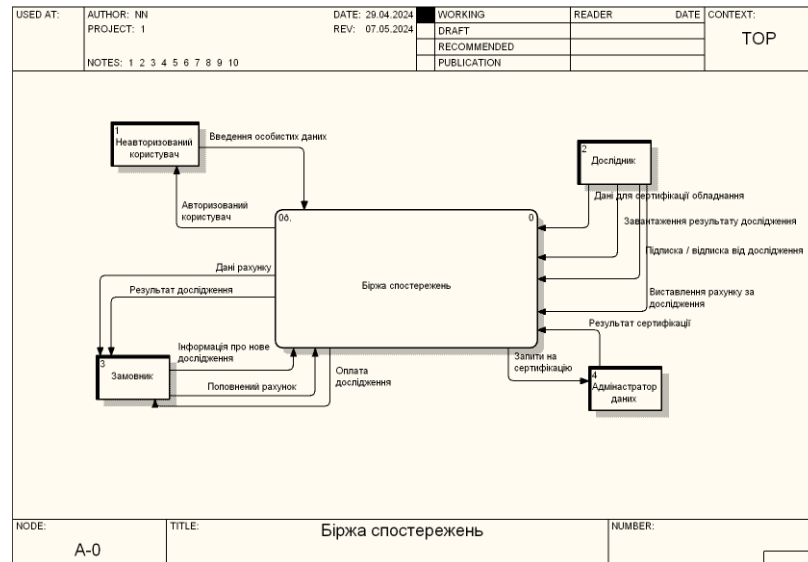


Рисунок 2.7 – Діаграма потоків даних біржі спостережень

Декомпозиція діаграми потоків даних (DFD) дозволяє розбити складну систему на менші компоненти для простого розуміння. Нижній рівень допомагає виявити взаємозв'язки між елементами системи та зрозуміти деталі процесів ІС, забезпечуючи деталізований огляд потоків даних, процесів, зовнішніх сутностей та сховищ даних. Процес декомпозиції передбачає поступовий перехід від верхнього рівня до нижчого, де система представлена в загальних рисах, до нижніх рівнів, не кожен процес деталізується.

Отже однією з ключових переваг декомпозиції DFD є її здатність зменшити складність опису інформаційної системи та надати чітке розуміння. В результаті, декомпозиція сприяє успішній реалізації системи та підвищує ефективність розробки складних процесів.

На рисунку 2.8 представлено декомпозицію діаграми потоків даних біржі спостережень, що містить наступні елементи моделі:

Процеси:

- обробка реєстрації / авторизації;
- поповнення рахунку;
- запит на сертифікацію;
- підписка / відписка;
- створення дослідження.

Сховища даних:

- обладнання;
- користувачі;
- дослідження.

Поток даних:

- введення особистих даних;
- авторизований користувач;
- зареєстрований профіль;
- замовник;
- дослідник;
- дані для авторизації;
- отримана сертифікація;
- сертифікація обладнання;
- дані для сертифікації обладнання;
- результат сертифікації;
- запити на сертифікацію;
- дані рахунку;
- результат дослідження;
- поповнений рахунок;
- доступні дослідження;
- запит на доступні дослідження;
- інформація про нове дослідження;
- підписка / відписка від дослідження;
- виставлення рахунку за дослідження;
- завантаження результату дослідження;

– оплата дослідження;

Результуючи, моделювання діаграми потоків даних дає чітке та зрозуміле графічне представлення функціональності, що сприяє кращому плануванню розробки інформаційної системи.

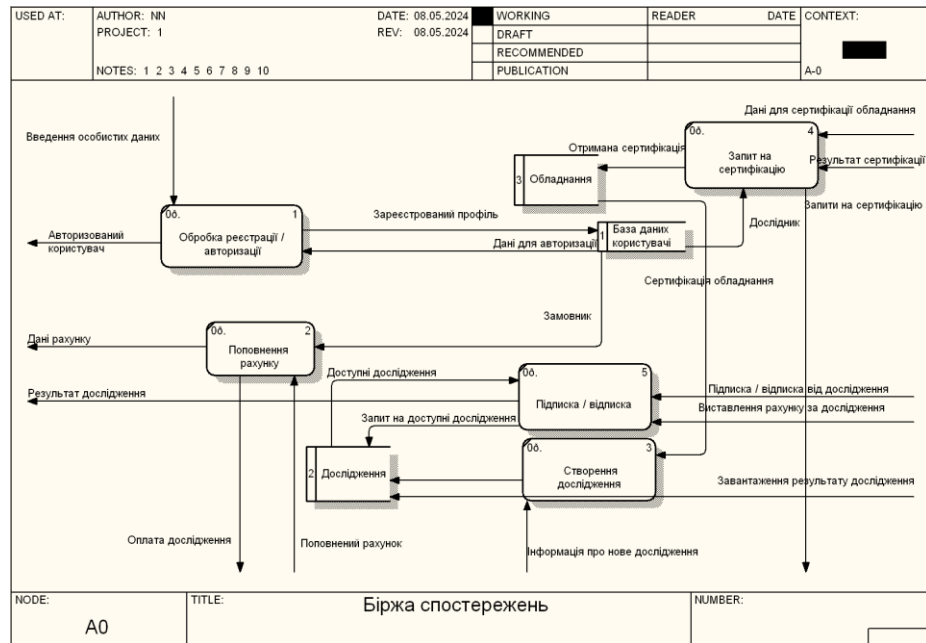


Рисунок 2.8 Декомпозиція діаграми потоків даних біржі спостережень

2.4 Визначені функціональні вимоги до клієнтської частини інформаційної системи біржі спостережень

Інтерфейс користувача був розроблений для біржі спостережень, представляє собою складну структуру, що базується на трьох основних технологіях розробки веб-застосунків: HTML, CSS та JavaScript. Кожна технологія грає важливу роль у створенні динамічного та привабливого інтерфейсу, який забезпечує зручність використання. Карта веб-сторінок біржі спостережень подана на рисунку 2.9.

Розроблені веб-сторінки, що доступні для замовників системи:

– авторизація / реєстрація. Сторінка, яка відображається під час відкриття біржі спостережень, для того, щоб створити новий профіль чи увійти до системи;

– сторінка поповнення балансу. Дає можливість купити віртуальну валюту для можливості оплачувати свої дослідження;

– сторінка створення заказу на дослідження даних. Дозволяє замовникам вказати інформацію свого дослідження та відправити його до системи;

– сторінка створених заказів. Дозволяє переглянути всі замовлення, що були створені в системі;

– перегляд інформації про результат дослідження. Сторінка містить в собі результати дослідження в обробленому вигляді;

Розроблені веб-сторінки, що доступні для підписників (спостерігачів) системи:

– авторизація / реєстрація. Сторінка яка відображається під час відкриття біржі спостережень, для того, щоб створити новий профіль чи увійти до системи;

– сторінка перегляд категорій досліджень. Дозволяє переглянути категорії завдань, які не знаходяться у виконанні;

– сторінка для заповнення інформації про обладнання. Спостерігачу потрібно заповнити доступні поля та надіслати на перевірку;

– сторінка оформлення підписки / відписки на спостереження. Дозволяє спостерігачу оформити підписку на завдання та почати його виконувати, за необхідністю відмінити відписку;

– сторінка для відправлення результату дослідження. Спостерігачу необхідно завантажити результати проведеного дослідження та відправити його.

Розроблені веб-сторінки, що доступні для адміністратора системи:

– авторизація / реєстрація. Дозволяє вибрати відповідну форму для реєстрації нового профілю чи авторизації;

- сторінка запитів на верифікацію обладнання. Адміністратору даних поступають запити на перевірку обладнання спостерігачів;
- сторінка сертифікації обладнання. Якщо обраний запит про верифікацію відповідає вимогам і стандартам, то адміністратор надає результат, що дозволить спостерігачам почати виконувати дослідження.

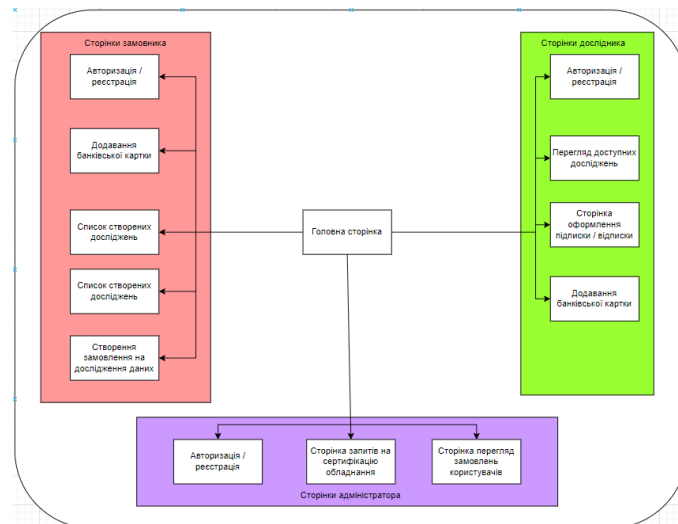


Рисунок 2.9 – Карта веб-сторінок біржі спостережень

2.5 Розробка діаграми варіантів використання біржі спостережень

Використовуючи діаграму варіантів використання можна зобразити функції, що доступні для певного типу користувачів. Варіанти використання описують взаємодію між зовнішніми акторами та системами для досягнення певної мети [24]. Метою для розробки діаграми є визначення функціональності системи та її опису з точки зору користувача, що забезпечує розуміння того, які дії виконуються та хто їх виконує.

Отже, завдяки діаграмі, можна визначити пріоритети розробки, планувати тестування і забезпечити, щоб система відповідала потребам кінцевих користувачів. Це зробить її цінним інструментом на ранніх етапах розробки та узгодженні вимог. Діаграму варіантів використання для біржі спостережень наведено на рисунку 2.10.

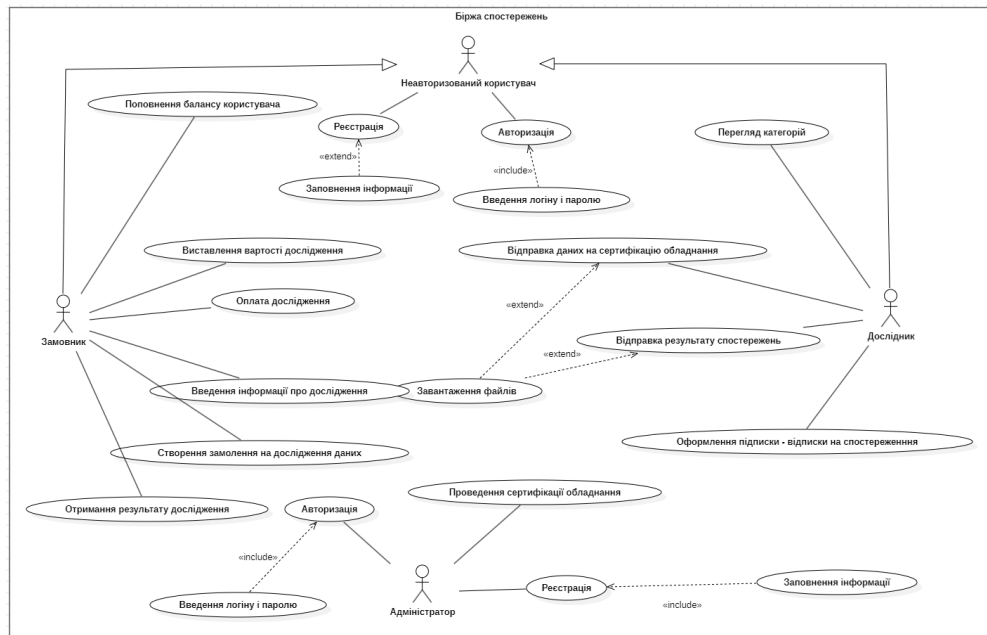


Рисунок 2.10 – Діаграма варіантів використання біржі спостережень.

Прецеденти, які були визначені під час проектування діаграми варіантів використання є критично важливими елементами, що визначають функціональність системи. Кожен визначений прецедент відображає один із варіантів використання ІС. Реалізація прецедентів передбачає врахування різних аспектів, таких як потоки даних, очікувані результати та взаємодії з іншими варіантами використання. Мета розробників забезпечити ефективну обробку запитів і сценаріїв з інтуїтивно зрозумілим інтерфейсом користувача.

Отже, прецеденти, подані у діаграмі варіантів використання є базовими частинами інформаційної системи біржі спостережень. Реалізація функцій є важливим кроком у забезпеченні системи вимогами користувачів та бізнесу.

2.6 Діаграма послідовності дій (Sequence diagram)

Для діаграми варіантів використання було прийняте рішення провести моделювання прецеденту «Проведення сертифікації обладнання» у вигляді діаграми послідовності дій, що виконуються під час даного процесу. Для

даного процесу, діаграма послідовності дій має візуалізувати та документувати послідовність дій, необхідних для виконання даного процесу.

На рисунку 2.11 подано діаграму послідовності дій для обраного прецеденту «Проведення сертифікації обладнання».

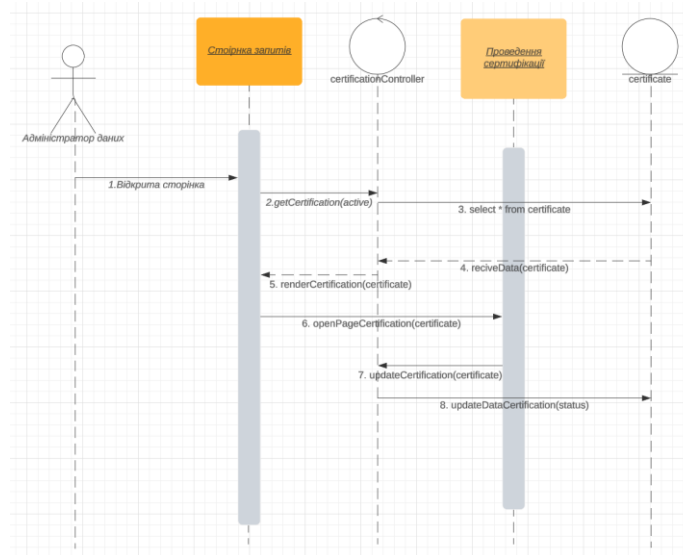


Рисунок 2.11 – Діаграма послідовності дій для прецеденту «Проведення сертифікації обладнання»

Будь-яка діаграма послідовності дій, яка є однією із діаграм уніфікованої мови моделювання (UML), має наступні цілі:

- візуалізація послідовності дій. Діаграма відображає порядок, в якому виконуються дії, що дозволить зрозуміти логіку та потоки даних процесу;
- визначення взаємодії між елементами. Дозволяє побачити, як різні елементи системи взаємодіють між собою під час виконання процесу;
- визначення складних сценаріїв. За допомогою діаграми можна визначити складні прецеденти системи, які можуть включати альтернативні та умовні гілки;
- забезпечення узгодженості. Діаграма допомагає перевірити узгодженість між процесами, компонентами та вимогами, що були визначені, та зменшує ймовірність непорозумінь та помилок під час розробки інформаційної системи.

Таким чином, проектування діаграми послідовності необхідна для створення чіткого, впорядкованого та зрозумілого опису того, як система виконує конкретну задачу ІС.

2.7 Проектування діаграми класів та визначення її мети

Діаграма класів є одним із найбільш популярних інструментів в уніфікованій мові моделювання (UML). Вона представляє статичне представлення структури системи у вигляді візуальної структури системи, фокусуючись на класах, атрибутах, методах і взаємозв'язках між ними. Розроблену діаграму класів для біржі спостережень наведено на рисунку 2.12.

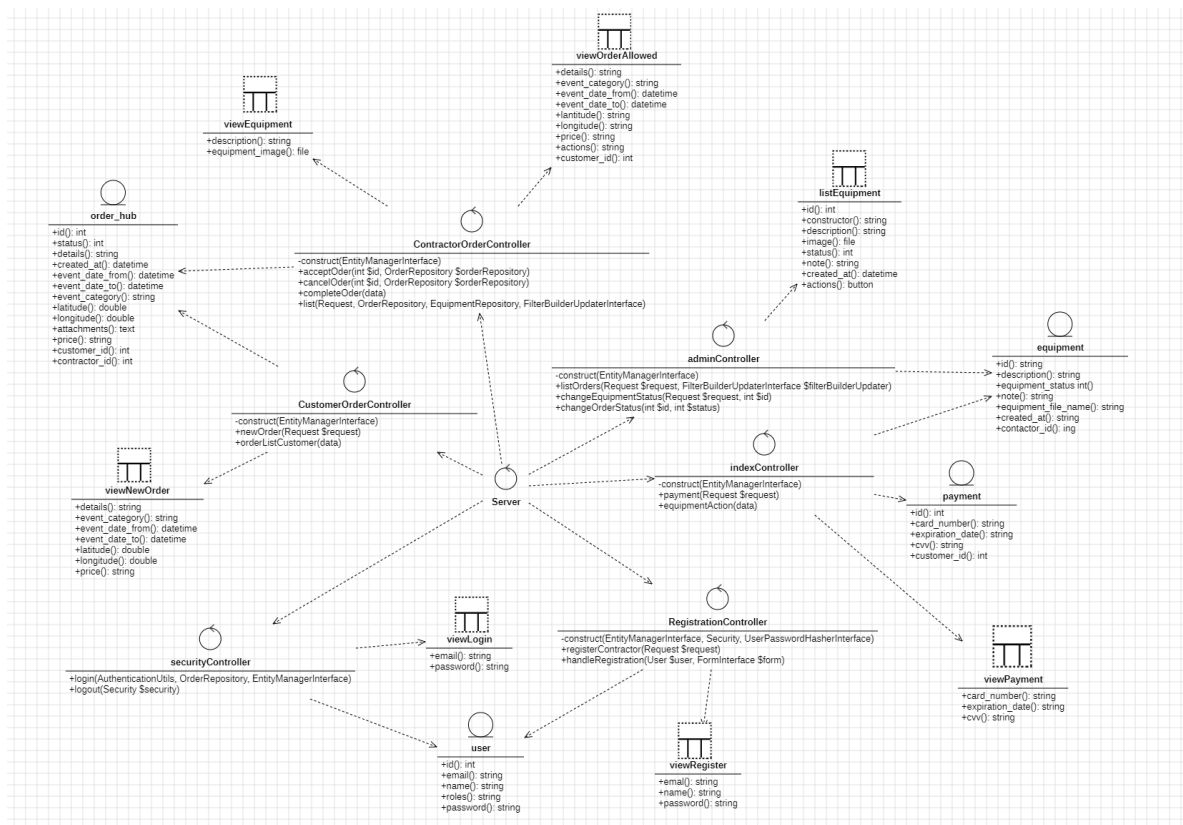


Рисунок 2.12 – Діаграма класів біржі спостережень

Використання діаграми класів у процесі проектування має ряд переваг для розробників, аналітиків та архітекторів програмного забезпечення, які

допомагають ефективніше виконувати свою роботу. Отже, розглянемо переваги, які дійсно відповідають під час проектування інформаційної системи біржі спостережень:

- візуалізація структури системи. Діаграма класів дозволяє створити графічне представлення класів, методів та взаємозв'язків між ними, що допоможе зрозуміти структуру системи на високому рівні;

- спрощення проектування і аналізу. На ранньому етапі можливо визначити потенційні проблеми або недоліки в розроблюваній системі. Наприклад, циклічні залежності, принципи SOLID або неправильне наслідування;

- підтримка ООП принципів. Діаграма класів дозволяє зрозуміти принципи об'єктно-орієнтованого програмування, саме наслідування, поліморфізм та інкапсуляцію;

- підтримка та розширення. Звертаючись до діаграми класів, розробники можуть швидше та легше розроблювати новий функціонал та проводити рефакторинг коду. Надає можливість зрозуміти, де та які класи, методи і атрибути краще додати до інформаційної системи;

- документування ІС. Створення діаграми для внесення її до документації є важливим кроком, оскільки діаграми є частиною технічної документації проекту, що використовується під час розробки, тестування та розширення системи;

- забезпечення узгодженості. Може бути представлена як план проекту, який допомагає команді слідувати одній спільній архітектурі, яка допоможе проектувати систему з використанням певного підходу.

Розробивши діаграму класів створимо таблицю для короткого опису класів, методів та атрибутів, що були використанні під час розробки інформаційної системи біржі спостережень. Опис діаграми класів подано в таблиці 2.1.

Таблиця 2.1 – Опис класів та методів

Контролер / представлення	Методи / атрибути	Опис методів	Модель взаємодії
contactorOrderController	construct	Ініціалізація об'єктів для взаємодії системи	Order Agreement
	acceptOrder	Метод підписки на дослідження даних	
	cancelOrder	Метод на відписку від дослідження	
	completeOrder	Завершення проведеного дослідження	
	list	Перегляд всіх доступних для виконання досліджень	
customerOrderController	construct		Order_ Agreement
	newOrder	Створення нового дослідження	
	orderList	Перегляд всіх створених досліджень користувача	
adminController	construct		Device
	listOrders	Перегляд всіх замовлень на дослідження даних	
	changeEquipmentStatus	Зміна статусу обладнання (проведення сертифікації)	
	changeOrderStatus	Зміна статусу замовлення на дослідження даних	
securityController	login	Метод авторизації в системі	User
	logout	Вихід із інформаційної системи	
registrationController	construct		User
	registerCustomer	Проведення реєстрації в системі для ролі замовника	
	registerContractor	Проведення реєстрації в системі для ролі дослідника	
	handleRegistration	Обробка реєстрації в системі	
indexController	payment	Метод для введення даних картки та оплати за дослідження	Credit_card
	equipmentAction	Перегляд запитів на проведення сертифікації обладнання	
	indexCustomer	Отримання інформації про авторизованого користувача	

Діаграма класів дуже тісно пов'язана з принципами об'єктно-орієнтованого програмування для зображення взаємозв'язків між елементами, оскільки дає змогу зрозуміти архітектуру системи, організацію класів та їх взаємозв'язки. На розробленій діаграмі класів представлено компоненти view, які пов'язані з елементами графічного інтерфейсу користувача. Визначення елементів view на діаграмі класів включає в себе наступні аспекти:

- визначення структури графічного інтерфейсу. Елементи view відображають частини графічного інтерфейсу, такі як кнопки, картинки, текстові поля. Ці елементи дозволяють побачити організацію та взаємодію між собою;

- взаємодія з іншими елементами. Зображення різних типів елементів на діаграмі класів дозволяє визначити взаємозв'язки між моделями даних та контролерами системи [20];

- дизайн інтерфейсу. View дозволять спланувати на етапі розробки, та визначити, які компоненти будуть взаємодіяти з бізнес-логікою системи;

- документація. Опис всіх елементів представлених на діаграмі класів надає чіткого розуміння про використані контролери, моделі та представлення в інформаційній системі.

Отже, визначення різних елементів на діаграмі класів може стати важливим кроком у створенні візуалізації графічного інтерфейсу та бізнес-логіки системи.

2.8 Постановка завдань для розробки інформаційної системи біржі спостережень

Під час проектування процесів системи та їх складових можна поступово створювати завдання для створення інформаційної системи біржі спостережень. Детальне визначення завдань, допоможе зробити процес розробки більш організованим. Для розробки інформаційної системи біржі спостережень було визначено наступні задачі, що підлягають розробці:

- розробка бази даних;
- визначення взаємозв'язків між таблицями бази даних;
- розробка контролерів та моделей на серверній частині застосунку;
- розробка функціоналу доступу до бази даних, та створення API для їх взаємодії;
- проектування дизайну;
- розробка компонентів для забезпечення користувачів графічним інтерфейсом доступу до інформаційної системи;
- розробки маршрутизацію для веб-сторінок;
- створення взаємозв'язків між клієнтською та серверною частиною додатку;
- розробка додаткових сторінок для біржі спостережень.

Отже, визначення та створення завдань для розробки інформаційної системи біржі спостережень дозволяє забезпечити систематичний підхід до розробки, тестування та вдосконалення, результатом чого буде швидкий та ефективний додаток, що задовольнить потреби користувачів і бізнесу.

3 ОПИС ПРИЙНЯТИ РІШЕНЬ, РОЗРОБКА БІРЖИ СПОСТЕРЕЖЕНЬ

3.1 Аналіз та обґрунтування обраних технологій проектування

Провівши повноцінний аналіз вимог та визначивши потреби користувачів, необхідно обрати мову програмування та технології для створення інформаційної системи біржі спостережень. Для розробки було обрано мову програмування Java, з використанням фреймворків Spring Boot і MySQL для написання серверної частини застосунку та React.js для клієнтської.

Використані технології під час проектування та їх цілі продемонстровано на рисунку 3.1.

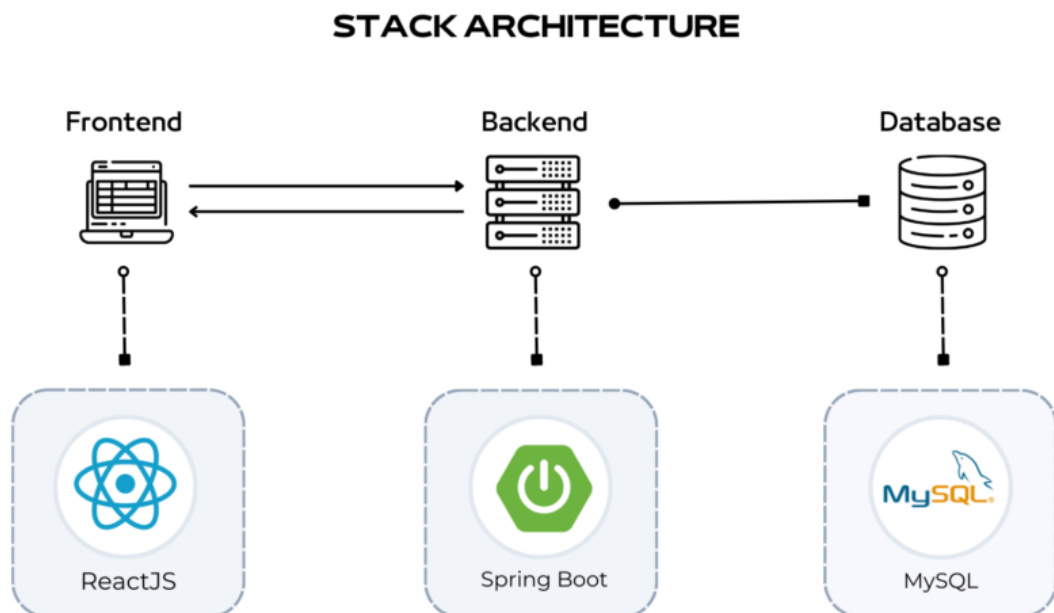


Рисунок 3.1 – Обрані технології проектування біржі спостережень

Вибір стеку технологій для проектування інформаційної системи може впливати на ефективність розробки, продуктивність та масштабованість проекту [6]. Далі розглянуто переваги даного стеку проектування ІС.

Клієнтська частина застосунку:

- висока продуктивність. React використовує віртуальний DOM, що дозволяє швидко оновлювати дані на інтерфейсі користувача без значних ресурсних затрат [2];

- компонентний підхід. Створення компонентів для повторного їх використання, що спрощує підтримку і розвиток додатку [15];

- широка екосистема. React має велику кількість інструментів для спрощення проектування, що дозволяють розширити функціонал фреймворку [1].

Серверна частина застосунку:

- легкий старт. Spring Boot забезпечує розробників легким стартом у створенні проекту завдяки мінімальній кількості конфігурацій та вбудованих механізмів [9];

- модульність. Використовується модульна архітектура, що дозволяє легко створювати нові функції додатку та забезпечувати гнучкість у розробці інформаційних систем різної складності [21];

- безпека даних. Spring Boot включає в себе інструменти для створення безпеки додатка, наприклад, Spring Security, що може легко бути інтегрований з іншими технологіями [13].

База даних:

- відкритий вихідний код. Завдяки великій спільноті розробників відбувається постійна підтримка та оновлення функцій додатку;

- сумісність. MariaDB сумісна з СУБД MySQL, що дозволяє проводити міграції баз даних між ними;

- продуктивність. Висока продуктивність дозволяє швидко оброблювати запити SQL та виконання функцій, за допомогою використання сучасних алгоритмів.

Вибір даного стеку технологій дозволить створювати ефективні та надійні інформаційні системи, що відповідають сучасним вимогам та потребам користувачів.

3.2 Вибір РСУБД для проектування бази даних

Для проектування біржі спостережень було обрано РСУБД MariaDB. MariaDB – це реляційна система управління базами даних, створена на основі MySQL. Вона представлена як відкрита та безкоштовна альтернатива MySQL, що має високу сумісність та дозволяє легко перейти з однієї СУБД до іншої без значних змін у коді [8].

Головні переваги, що були визначені для вибору MariaDB для використання під час створення біржі спостережень:

- відкритий код. MariaDB – це проект з відкритим вихідним кодом, який підтримується великою спільнотою розробників з усього світу;
- сумісність. MariaDB надає високу сумісність із СУБД MySQL, що дозволяє проводити міграції між ними. Перевагою буде робота більшість запитів SQL, бібліотек і інструментів, що були розроблені для MySQL, також працюватимуть і з MariaDB;
- продуктивність. MariaDB забезпечує підвищення продуктивності та масштабованості проектів, завдяки використанню сучасних алгоритмів [11];
- безпека. Забезпечує додатковими функціями безпеки, такі як розширене шифрування даних і підтримка методів автентифікації;
- розширені функції. MariaDB включає в себе додаткові функції, яких немає в MySQL. Серед нових функцій є нові типи даних, функції реплікації та механізми зберігання даних;
- підтримка. Завдяки великій спільноті забезпечується постійна підтримка та розширення функцій та механізмів, що сприяє покращенню бази даних;

– масштабованість. Забезпечує високу масштабованість для розробників, що дозволяє використовувати її для проектів різної складності.

Отже, обрана СУБД повністю задовольняє потреби розробників для її використання, та її переваги для використання в розробці інформаційної системи. Розроблена база даних для інформаційної системи біржі спостережень наведена на рисунку 3.2, на якій можна побачити типи зв'язків між таблицями, поля та їх типи даних, що використовуються.

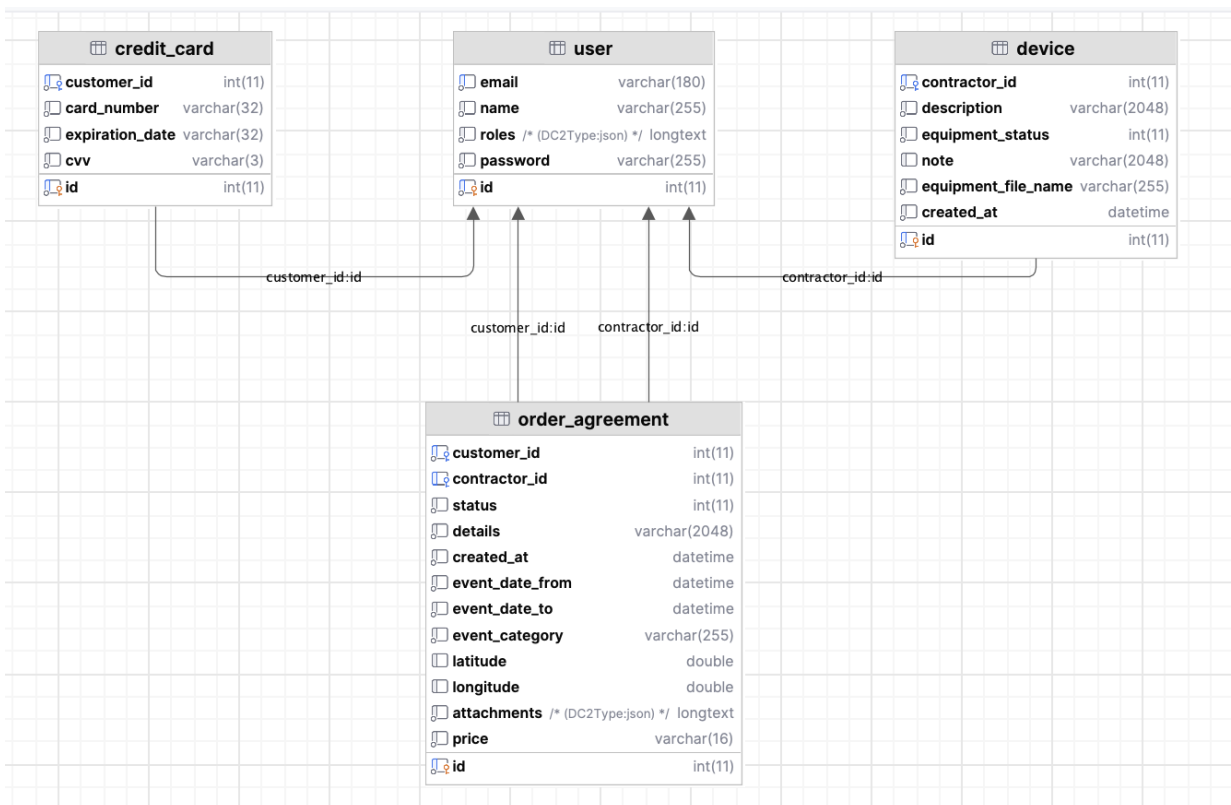


Рисунок 3.2 – Логічна модель бази даних для інформаційної системи біржі спостережень

В таблиці 3.1 наведено більш детальний опис логічної моделі бази даних, яка використовується для проектування біржі спостережень. Логічна модель допомагає визначити структуру даних, включаючи зв'язки між таблицями та обмеження полів. Цей процес дозволяє забезпечити систематичний та структурований підхід до організації та управління даними в системі.

Таблиця 3.1 – опис таблиць бази даних

№	Назва таблиці	Назва поля	Тип даних	Примітка
1	credit_card	id	Int(11)	Унікальний ідентифікатор
		cvv	Varchar(3)	Захисний код картки
		expiration_date	Varchar(32)	Термін придатності
		card_number	Varchar(32)	Номер
		customer_id	Int(11)	Ідентифікатор користувача
2	device	id	Int(11)	Унікальний ідентифікатор
		created_at	Datetime	Дата створення
		equipment_file_name	Varchar(255)	Завантажені файли обладнання
		note	Varchar(2048)	Повідомлення
		equipment_status	Int(11)	Статус обладнання
		description	Varchar(2048)	Опис обладнання
		contractor_id	Int(11)	Ідентифікатор дослідника
3	user	id	Int(11)	Унікальний ідентифікатор
		roles	Longtext	Роль користувача
		email	Varchar(180)	Електронна адреса
		password	Varchar(255)	Пароль
		name	Varchar(255)	Ім'я

Кінець таблиці 3.1

№	Назва таблиці	Назва поля	Тип даних	Примітка
4	order_agreement	id	Int(11)	Унікальний ідентифікатор
		price	Varchar(16)	Ціна за дослідження
		attachments	Longtext	Додаткові вкладення
		longitude	Double	Довгота
		latitude	Double	Широта
		event_category	Varchar(255)	Категорія дослідження
		event_date_to	Datetime	Кінець дати проведення
		event_date_from	Datetime	Початок дати проведення
		created_at	Datetime	Дата створення
		details	Varchar(2048)	Додаткові деталі
		status	Int(11)	Статус
		contractor_id	Int(11)	Ідентифікатор дослідника, що виконує завдання
customer_id	Int(11)	Ідентифікатор замовника		

3.3 Вибір архітектури проектування

Розробка біржі спостережень потребує правильного та ретельного вибору архітектури для швидкого та ефективного створення інформаційної системи.

Написання коду для біржі спостережень є складним і вирішальним завданням, що вимагає вибору архітектури для забезпечення швидкого і ефективного функціонування системи. Після аналізу вимог та функцій для біржі спостережень було обрано архітектуру проектування MVC (Model-View-Controller). Дана архітектура забезпечить чітку структуру коду, полегшить його підтримку, дозволить повторно використовувати фрагменти коду та допоможе розділити логіку програми на три основні компоненти: модель, представлення та контролер. На рисунку 3.2 представлено графічне представлення архітектури MVC та як вона взаємодіє з користувачем.

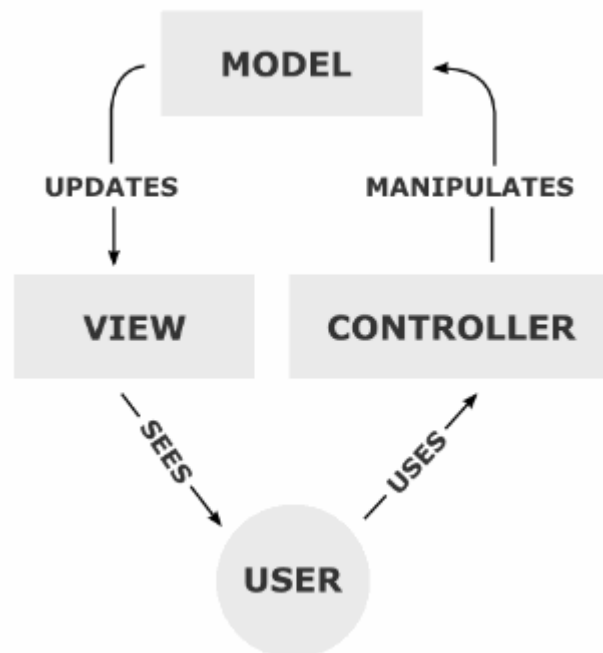


Рисунок 3.3 – Архітектура MVC

MVC розділяє інформаційну систему на три основні компоненти, де кожен з них відповідає за свою логіку та призначення в додатку:

– Model (Модель). Головна мета моделі полягає у взаємодії з даними, бізнес-логіці та правилах роботи з даними. Модель взаємодіє з базою даних для виконання основних операцій з базою даних, саме: вибір, запис, оновлення і видалення;

– View (Представлення). Відповідає за відображення графічного інтерфейсу та даних користувачеві. Представлення отримує дані з моделі та надає їх у обробленому вигляді користувачеві для взаємодії з ними;

– Controller (Контролер). Мета контролера полягає в обробці запитів від користувача, які в свою чергу буде взаємодіяти з моделлю для отримання інформації та передавати отримані дані представленню для відображення їх на інтерфейсі користувача.

Визначення архітектури проектування дозволить зробити чіткий розподіл обов'язків, що зменшить навантаження для тестування, підтримки, масштабованості та гнучкості інформаційної системи. В свою чергу такий підхід до створення структури проекту дозволить створити ефективну і надійну систему, що забезпечить задоволеність користувачів та бізнесу.

3.4 Вибір шаблону проектування біржі спостережень

Провівши короткий огляд шаблонів проектування, що можуть бути використані для розробки біржі спостережень, було обрано «Публікація–підписка». Шаблон реалізовує механізм передачі повідомлень, в якому відправники (публікатори) не здійснюють пряме відправлення повідомлень дослідникам [10]. Для цього було створено сутність в базі даних, що буде зберігати ці створені повідомлення за категоріями, для яких потрібно провести дослідження даних. Це дозволить зробити гнучкий та ефективний підхід до проектування інформаційної системи біржі спостережень.

Основні принципи шаблону проектування «Публікація–підписка», що були реалізовані в біржі спостережень:

– розділення обов'язків. Публікатори відповідають за створення та відправку запитів на аналіз даних, а дослідники (підписники) відповідають за отримання та обробку цих повідомлень. Цей процес зменшує зв'язок між компонентами, що робить їх незалежними один від одного і більш простими для змін та управління [18];

– зберігання повідомлень. Необхідно створити сутність в базі даних, яка буде зберігати всю інформацію, надану про дослідження, зокрема: категорію, дати проведення, довготу, широту та ціну. Дослідники можуть підписуватися або відписуватися від проведення досліджень, маючи можливість підписуватися на ті дослідження, що їх цікавлять.

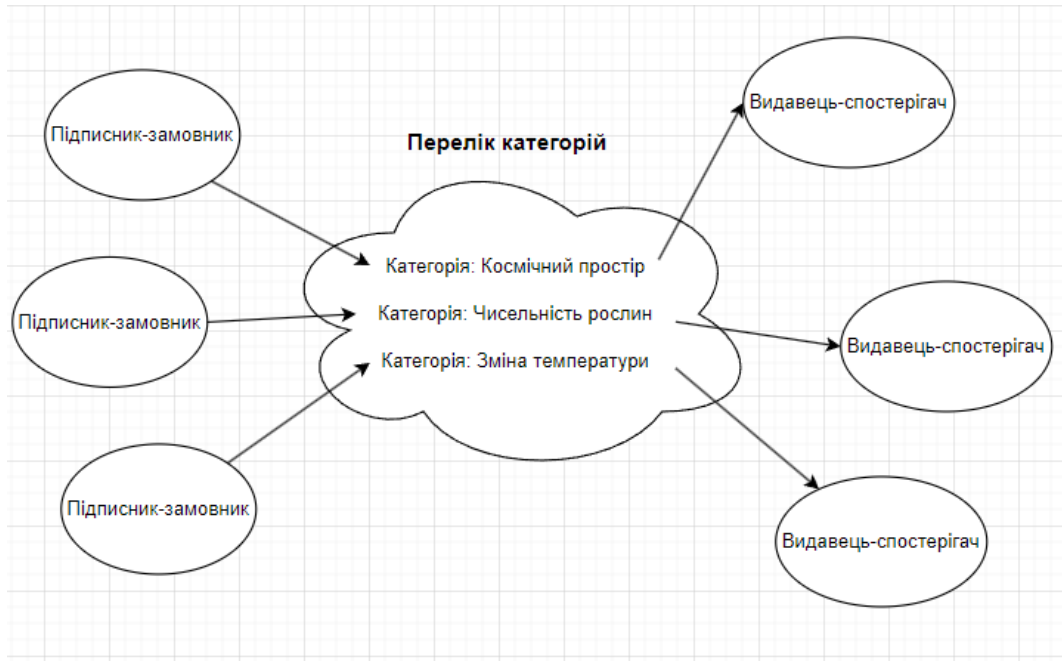


Рисунок 3.4 – Візуальне представлення шаблону проектування «Публікація–підписка»

3.5 Реалізація контролерів і моделей для біржі спостережень

Створення контролерів і моделей системи є критично важливим етапом для складних інформаційних систем. Контролери та моделі забезпечують додаток бізнес-логікою та можливістю взаємодіяти з даними, що були отримані з БД, та передавати інформацію до графічного інтерфейсу користувачів [16].

Створення моделей дозволить розробникам спростити взаємодію з даними, що були отримані з бази даних. Вони дозволяють працювати з даними у вигляді об'єктів, що робить використання коду більш зрозумілим та

ефективним. Розглянемо модель Device, що була розроблена з описом полів та методів, для якої використовували Spring Data JPA. Розроблена модель подана на рисунку 3.5.

```
public class Device {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String status;
    private String researcher;
    // Getters and Setters
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public String getResearcher() {
        return researcher;
    }
    public void setResearcher(String researcher) {
        this.researcher = researcher;
    }
}
```

Рисунок 3.5 – Програмний код моделі Device

Контролери потребують більшої уваги та кількості витраченого часу для їх написання. Розглянемо «AdminController», який використовується для адміністрування даних у системі, таких як дослідження та обладнання. Контролер реалізує чотири основні методи для взаємодії з інформацією в

системі. Методи «listOrders» та «changeOrderStatus» необхідні для перегляду всіх актуальних досліджень у системі та можливості оновлювати поточний статус проведення дослідження. Аналогічно методи «listDevices» та «changeDeviceStatus» відповідають за перегляд всіх поточних запитів на проведення сертифікації обладнання дослідників та можливість проводити сертифікацію для певного обладнання, попередньо переглянувши надану інформацію та визначивши результат проведення сертифікації, що дозволить досліднику почати повноцінно взаємодіяти із системою. Реалізація контролера подана на рисунку 3.6.

```
public class AdminController {
    @Autowired
    private EntityManager entityManager;
    @Autowired
    private DeviceRepository deviceRepository;
    @Autowired
    private OrderRepository orderRepository;
    @GetMapping("/admin/devices")
    public String listDevices(Model model, @RequestParam(defaultValue = "0") int offset) {
        int paginatorPerPage = DeviceRepository.PAGINATOR_PER_PAGE;
        int previous = Math.max(0, offset - paginatorPerPage);
        Pageable pageable = PageRequest.of(offset, paginatorPerPage);
        Page<Device> paginator = deviceRepository.findAll(pageable);
        EquipmentChangeStatusFormType changeStatusForm = new
EquipmentChangeStatusFormType();
        model.addAttribute("equipment", paginator.getContent());
        model.addAttribute("previous", previous);
        model.addAttribute("next", Math.min(paginator.getTotalPages(),
offset + paginatorPerPage));
        model.addAttribute("changeStatusForm", changeStatusForm);
        return "admin/device_list";
    }
    @GetMapping("/admin/orders")
    public String listOrders(Model model, @RequestParam(defaultValue = "0") int offset) {
        int paginatorPerPage = OrderRepository.PAGINATOR_PER_PAGE;
        int previous = Math.max(0, offset - paginatorPerPage);
        Pageable pageable = PageRequest.of(offset, paginatorPerPage);
        Page<Order> paginator = orderRepository.findAll(pageable);
        OrderFormFilter form = new OrderFormFilter();
```

```

        model.addAttribute("orders", paginator.getContent());
        model.addAttribute("previous", previous);
        model.addAttribute("next", Math.min(paginator.getTotalPages(),
offset + paginatorPerPage));
        model.addAttribute("form", form);
        return "admin/order_list";
    }
    @PostMapping("/admin/devices/{id}/status")
    public String changeDeviceStatus(@PathVariable Long id,
EquipmentChangeStatusForm equipmentChangeStatusForm) {
        Device equipment = entityManager.find(Device.class, id);
        if (equipment == null) {
            throw new NotFoundException("Equipment not found");
        }
        equipment.setNote(equipmentChangeStatusForm.getNote());

equipment.setStatus(equipmentChangeStatusFormType.getEquipmentStatus()
);

        entityManager.getTransaction().begin();
        entityManager.persist(equipment);
        entityManager.getTransaction().commit();
        return "redirect:/admin/devices";
    }
    @GetMapping("/admin/orders/{id}/status/{status}")
    public String changeOrderStatus(@PathVariable Long id,
@PathVariable int status) {
        Order order = entityManager.find(Order.class, id);
        if (order == null) {
            throw new NotFoundException("Order not found");
        }
        order.setStatus(status);
        entityManager.getTransaction().begin();
        entityManager.persist(order);
        entityManager.getTransaction().commit();
        return "redirect:/admin/order";
    }
}
}

```

Рисунок 3.6 – Програмний код контролера «AdminController»

Обов'язковим контролером для біржі спостережень є взаємодія дослідника з оформленою підпискою на завдання, саме оновлення статусу дослідження та надання результату. Контролер «ContractorOrderController» реалізує методи взаємодії з дослідженнями замовників. Метод «list» обробляє запит на відображення списку замовлень для відображення на графічному інтерфейсі користувача, враховуючи параметри сторінки та можливість

фільтрації. Метод «acceptOrder» реалізує підписку на виконання дослідження, прив'язуючи себе до об'єкту order. «cancelOrder» реалізує можливість відписки від дослідження даних, скасування своєї підписки доступне за певним переліком причин, що дозволяють використати цей функціонал. Останній метод «completeOrder» оброблює результати дослідження та надсилає інформацію про виконане дослідження. Реалізація даного контролера подана на рисунку 3.7.

```
public class ContractorOrderController {
    @Autowired
    private EntityManager entityManager;
    @Autowired
    private OrderRepository orderRepository;
    @Autowired
    private DeviceRepository deviceRepository;
    @GetMapping("/orderContractor")
    public String list(Model model, @RequestParam(defaultValue = "0")
int offset) {
        int paginatorPerPage = OrderRepository.PAGINATOR_PER_PAGE;
        int previous = Math.max(0, offset - paginatorPerPage);
        Pageable pageable = PageRequest.of(offset, paginatorPerPage);
        Page<Order> paginator =
orderRepository.getOrderQbByContractor(this.getUser(), offset);
        model.addAttribute("orders", paginator.getContent());
        model.addAttribute("previous", previous);
        model.addAttribute("next", Math.min(paginator.getTotalPages(),
offset + paginatorPerPage));
        model.addAttribute("isHasApprovedEquipment",
equipmentRepository.findOneByContractorAndEquipmentStatus(this.getUser
()), EquipmentStatus.APPROVED) == null);
        model.addAttribute("form", new OrderFormFilter());
        return "contractor/order_list";
    }

    @GetMapping("/cancelOrder/{id}")
    public String cancelOrder(@PathVariable Long id) {
        Order order = entityManager.find(Order.class, id);

        if (order == null) {
            throw new BadRequestHttpException("Wrong order id");
        }
    }
}
```

```

        order.setStatus(OrderStatus.CANCELLED);

        entityManager.getTransaction().begin();
        entityManager.persist(order);
        entityManager.getTransaction().commit();

        return "redirect:/orderContractor";
    }

    @PostMapping("/completeOrder/{id}")
    public String completeOrder(@PathVariable Long id,
        @RequestParam("attachments") MultipartFile[] attachments,
        RedirectAttributes redirectAttributes) {
        Order order = entityManager.find(Order.class, id);

        if (order == null) {
            throw new BadRequestHttpException("Wrong order id");
        }

        if (attachments != null && attachments.length > 0) {
            String[] fileNames = new String[attachments.length];

            for (int i = 0; i < attachments.length; i++) {
                MultipartFile attachment = attachments[i];
                String originalFilename =
attachment.getOriginalFilename();
                String newFilename = originalFilename.substring(0,
originalFilename.lastIndexOf('.')) + "-" + System.currentTimeMillis()
+ originalFilename.substring(originalFilename.lastIndexOf('.'));

                try {
                    attachment.transferTo(new
File(this.getParameter("result_directory") + "/" + newFilename));
                    fileNames[i] = newFilename;
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            order.setAttachments(fileNames);
        }
        order.setStatus(OrderStatus.COMPLETED);
        entityManager.getTransaction().begin();
        entityManager.persist(order);
        entityManager.getTransaction().commit();
        return "redirect:/orderContractor";
    }
}

```

Рисунок 3.7 – Програмный код контролера «ContractorOrderController»

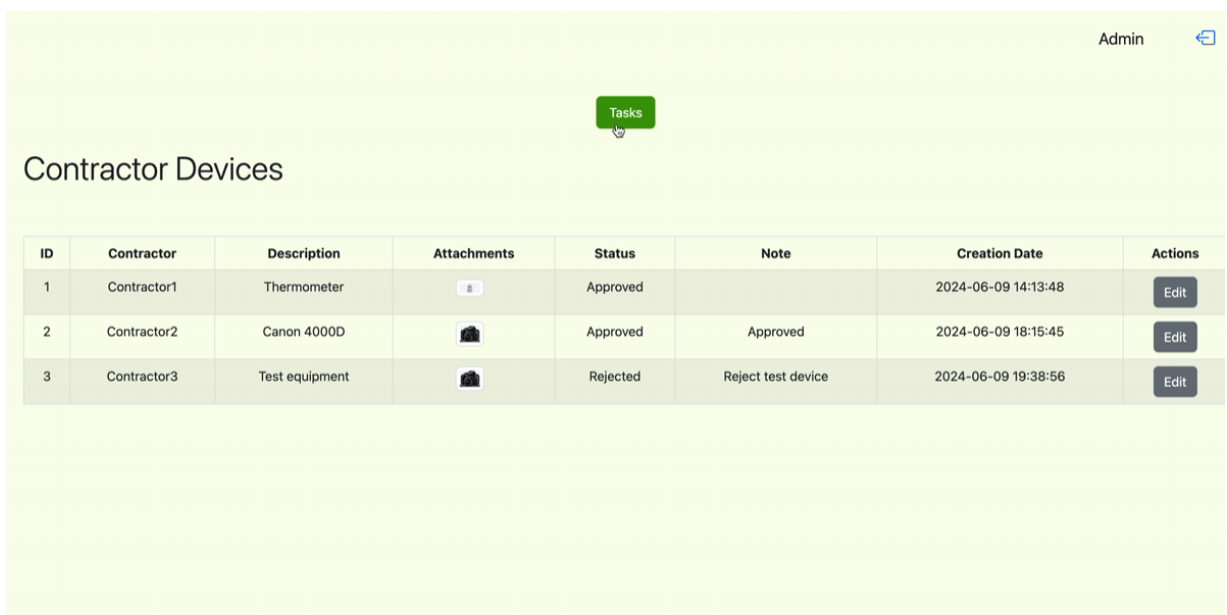
3.6 Створення графічного інтерфейсу користувача

Створення графічного інтерфейсу користувачів відбувалося з використанням технології веб-розробки ReactJS. React – це JavaScript-бібліотека з відкритим вихідним кодом для розробки інтерфейсів.

Інтуїтивно зрозумілий і простий функціонал веб-додатку стане ключовою частиною у створенні успішної інформаційної системи біржі спостережень. Веб-додаток повинен бути зручним та швидко доступним для різних типів користувачів. Використання ReactJS дозволить досягти поставленої мети для створення інтерфейсу користувача, завдяки потужним інструментам розробки [19].

Розробка унікального функціоналу для замовника, спостерігача та адміністратора даних є важливим етапом. Забезпечення таким функціоналом призведе до виконання головного процесу біржі спостережень.

Розроблений компонент «DeviceApproval» дозволяє динамічно взаємодіяти з обладнанням дослідників, переглядати надані зображення та редагувати інформацію про них, забезпечуючи зручний інтерфейс для адміністраторів. На рисунку 3.8 продемонстровано розроблену сторінку.






ID	Contractor	Description	Attachments	Status	Note	Creation Date	Actions
1	Contractor1	Thermometer		Approved		2024-06-09 14:13:48	Edit
2	Contractor2	Canon 4000D		Approved	Approved	2024-06-09 18:15:45	Edit
3	Contractor3	Test equipment		Rejected	Reject test device	2024-06-09 19:38:56	Edit

Рисунок 3.8 – Сторінка «Список обладнання»

Реалізація сторінки відбувалась з використанням бібліотеки ReactJS, HTML та CSS коду. Програмний код компонента подано на рисунку 3.9.

```

const DeviceApproval = ({ equipment, previous, next }) => {
  const [modalData, setModalData] = useState(null);
  const [modalImage, setModalImage] = useState('');
  const openModal = (imageSrc) => {
    setModalImage(imageSrc);
  };
  const handleEditClick = (item) => {
    setModalData(item);
  };
  const closeModal = () => {
    setModalData(null);
  };
  return (
    <div>
      <h1 className="display-6">Devices for approval</h1>
      <nav className="nav nav-pills justify-content-end p-4">
        <a className="nav-link active" aria-current="page"
href="/app_admin_order">Orders</a>
      </nav>
      <thead>
        <tr>
          <th>id</th>
          <th>contractor</th>
          <th>description</th>
          <th>attachments</th>
          <th>status</th>
          <th>note</th>
          <th>created at</th>
          <th>actions</th>
        </tr>
      </thead>
      {previous >= 0 && (
        <a
href={` /app_admin_equipment?offset=${previous}` }>Previous</a>
      )}
      {next < equipment.length && (
        <a
href={` /app_admin_equipment?offset=${next}` }>Next</a>
      )}

      {modalData && (
        <div className="modal fade" id={`editModal-
${modalData.id}`} tabIndex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
          <div className="modal-dialog">

```


Після успішної відправки запиту, досліднику буде відображено повідомлення про те, що обладнання у процесі сертифікації. Повідомлення продемонстровано на рисунку 3.11 та рисунку 3.12 подано програмний код створеного функціоналу.

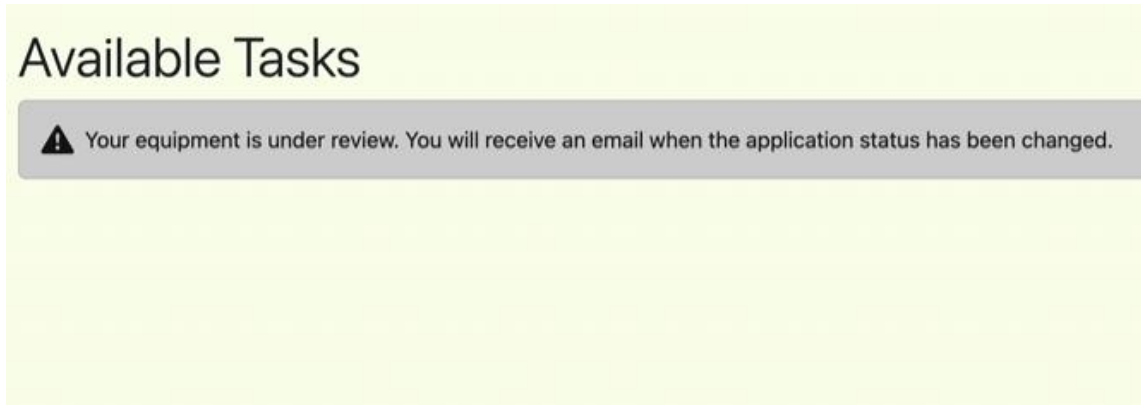


Рисунок 3.11 – Повідомлення після відправленого запиту

```
const DeviceForm = () => {
  // Використання useFormik для управління формою
  const formik = useFormik({
    initialValues: {
      description: '',
      equipment: ''
    },
    onSubmit: values => {
      console.log(values);
      // Обробка відправки форми
    }
  });
  return (
    <div className="container mt-7">
      <div className="row justify-content-center align-items-center">
        <div className="col-md-6">
          <div className="text-center">
            <h2 className="mb-4">Devices</h2>
          </div>
          <form onSubmit={formik.handleSubmit}>
            <div className="container">
              <div className="row">
                <div className="mb-3">
                  <div className="p-3">
                    <label htmlFor="description"
className="form-label">Description</label>

```

```

        <input
            id="description"
            name="description"
            type="text"
            className="form-control"

onChange={formik.handleChange}

value={formik.values.description}
        />
        {formik.errors.description ?
<div className="text-danger">{formik.errors.description}</div> : null}
        </div>
    </div>
    <div className="row">
        <div className="mb-3">
            <div className="p-3">
                <label htmlFor="equipment"
className="form-label">Devices</label>
                <input
                    id="equipment"
                    name="equipment"
                    type="text"
                    className="form-control"

onChange={formik.handleChange}

value={formik.values.equipment}
                />
                {formik.errors.equipment ?
<div className="text-danger">{formik.errors.equipment}</div> : null}
                </div>
            </div>
        </div>
    </div>
    <div className="text-center" style={{
marginTop: '30px' }}>
        <div className="btn-group">
            <button type="button" className="btn btn-primary">
                </div>
        </div>
    </div>
</form>
</div>
</div>
</div>
);
};

```

Рисунок 3.12 – Програмний код сторінки «надання інформації про обладнання»

Біржа спостережень повинна приймати запити на дослідження даних, що дозволить відтворити повноцінне виконання головної мети розробленої інформаційної системи.

Дана форма містить такі поля для заповнення: деталі замовлення, категорія, дата початку, дата закінчення, довгота, широта та ціна. Для відправки дослідження необхідно натиснути кнопку «Create». Інтерфейс даної форми подано на рисунку 3.13.

Рисунок 3.13 – Сторінка «Створення нового дослідження»

Першим кроком для даного процесу є створення нових досліджень на аналіз даних. За нові дослідження, що потраплять до системи відповідають авторизовані замовники, яким необхідні результати. Для даного функціонала було розроблено компонент «NewOrderForm», що відповідає за відображення форми для створення замовлення, який подано на рисунку 3.14.

```
const NewOrderForm = () => {
  const [errors, setErrors] = useState({});
  const handleChange = (e) => {
    const { name, value } = e.target;
    setForm({
      ...form,
      [name]: value
    });
  };
};
```

```

return (
  <div className="container mt-7">
    <div className="row justify-content-center align-items-center">
      <div className="col-md-6">
        <div className="text-center">
          <h2 className="mb-4">Create new order</h2>
        </div>

        {Object.keys(errors).length > 0 && (
          <div className="alert alert-danger">
            {Object.values(errors).map((error, index) => (
              <p key={index}>{error}</p>
            ))}
          </div>
        )}
      </div>
    </div>

    <form onSubmit={handleSubmit}>
      <div className="p-3">
        <label>Details</label>
        <input
          type="text"
          value={form.details}
          onChange={handleChange}
          className="form-control"
        />
      </div>

      <div className="text-center" style={{ marginTop: '30px' }}>
        <div className="btn-group">

          <button type="submit" className="btn btn-primary">

            <i className="bi bi-plus-circle-fill"></i>

            Create
          </button>
        </div>
      </div>
    </form>
  </div>
);
};

```

Рисунок 3.14 – Програмний код компонента «Створення нового дослідження»

3.7 Тестування запитів розробленої частини інформаційної системи біржі спостережень

Тестування відбувалось за допомогою методу «Чорна скринька» та програмного застосунку Postman. Даний метод тестування ПЗ перевіряє функціональність системи без знань структури коду [14]. Тестування серверної частини з використанням Postman є важливим етапом розробки та підтримки API. Postman - це потужний інструмент для розробників, який дозволяє тестувати та коригувати роботу API на різних етапах проектування системи [14]. Метою такого тестування з використанням Postman є:

- перевірка API. Необхідно забезпечити відповідність виконання функцій до специфікацій та очікувань;
- налагодження. Postman дозволять виявити та виправити помилки на серверній частині застосунку;
- перевірка продуктивності. Тестування продуктивності API та проведення оцінки його поведінки і швидкої під час навантаження;
- документація. Створення колекцій запитів для полегшення командної роботи, що забезпечить легку передачу знань між розробниками.

Тестування API було проведено для операцій створення нового профілю користувача та створення нової категорії досліджень.

Створюючи нову категорію, обов'язково необхідно вказувати її назву для коректної роботи API. Але для тестування перевіримо можливість створення нової категорії без її назви. Для цього перейдемо в додаток Postman та створимо новий запит до серверу для перевірки правильності роботи. Надіславши інформацію без вказання назви нової категорії буде отримано повідомлення про те, що назва категорії є обов'язковим параметром. Результат тестування подано на рисунку 3.15.

```

Body Cookies (1) Headers (8) Test Results
400 Bad Request 34 ms 644 B Save as Example
Pretty Raw Preview Visualize JSON
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
{
  "error": {
    "errors": {
      "name": {
        "name": "ValidatorError",
        "message": "Path 'name' is required.",
        "properties": {
          "message": "Path 'name' is required.",
          "type": "required",
          "path": "name",
          "value": ""
        },
        "kind": "required",
        "path": "name",
        "value": ""
      }
    },
    "_message": "Category validation failed",
    "name": "ValidationError",
    "message": "Category validation failed: name: Path 'name' is required."
  }
}

```

Рисунок 3.15 – Результат тестування «Створення нової категорії дослідження без назви»

Реєструючись в системі користувачеві обов’язково необхідно ввести персональні дані для створення особового профілю. Для даного інтерфейсу протестуємо можливість введення тільки імені або прізвища користувача під час створення нового профілю. Проведення тестів продемонстровано на рисунку 3.16.

```

Body Cookies (1) Headers (8) Test Results
Pretty Raw Preview Visualize JSON
1
2
3
{
  "error": "firstName is required"
}

Body Cookies (1) Headers (8) Test Results
Pretty Raw Preview Visualize JSON
1
2
3
{
  "error": "lastName is required"
}

```

Рисунок 3.16 – Тестування інтерфейсу створення нового профілю користувача

Після завершення тестування з використанням програмного застосунку Postman можна продовжувати та розширювати процес тестування не тільки серверної частини біржі спостережень, але й провести тестування інтерфейсу

користувача з використанням інших підходів. Інший підхід може забезпечити перевірку API та гарантувати його надійність і продуктивність виконання запитів.

Можна збільшити процес тестування з використанням інших підходів для перевірки своєї системи:

- автоматизація тестування з використанням Newman. Newman – це інтерфейс Postman, що дозволяє запускати колекції тестів автоматично. Дозволяє виконувати автоматично тести при кожному оновленні коду;

- навантажене тестування. Тестування дозволить зрозуміти, як система реагує на велике навантаження;

- інтеграційне тестування. Дозволить перевірити взаємодію між різними компонентами системи, що дозволить перевірити роботу всіх модулів разом;

- регресійне тестування. Дане тестування гарантує, що нові зміни в коді не впливають негативно на існуючий функціонал системи;

- тестування інтерфейсу користувача. Тестування UI забезпечує, що інтерфейс користувача працює належним чином.

Після початкового тестування з використанням Postman, подальше розширення та автоматизація тестування забезпечать всебічну перевірку вашого API. Використання додаткових інструментів та методів, таких як навантажувальне тестування, інтеграційне тестування та регресійне тестування дозволить створити надійну і безпечну систему, готову до використання в реальних умовах.

ВИСНОВОК

Підсумовуючи розділ про аналіз предметної області, саме для біржі спостережень було отримано необхідну інформацію про її функціональні та нефункціональні вимоги. Функціонал аналогічних систем досить схожий з розроблюваною, оскільки їхня мета полягає у зборі, аналізі та обробці даних.

У розділі розробки вимог було розглянуто технології проектування додатку та визначено переваги деяких із них. Переходячи до системного проектування, було обрано інструмент UML, що допоможе візуалізувати процеси та події в системі. UML — уніфікована мова моделювання, що є загальноприйнятим стандартом для візуального представлення інформаційних систем [12]. Саме вибір уніфікованої мови дозволив розробити детальну візуалізацію процесів та бізнес-функцій для інформаційної системи біржі спостережень.

Втіливши практичні та теоретичні вміння роботи з уніфікованою мовою моделювання, було створено карту веб-сторінок біржі спостережень. Для розробки інтерфейсу користувача було обрано три технології, які використовуються у проектуванні веб-застосунків: HTML, CSS та JavaScript. Кожна з технологій відіграє важливу роль у створенні динамічного та привабливого інтерфейсу, який забезпечує зручність використання.

Отже, поставлені задачі з проектування інформаційної системи біржі спостережень були успішно виконані. У будь-який момент можна поставити нові задачі щодо оптимізації системи, наприклад, провести рефакторинг коду, оптимізувати завантаження веб-сторінок або отримання інформації з бази даних.

АПРОБАЦІЇ

Для апробації була представлена тема «Розробка компонентів замовлення розробки та використання ПЗ інформаційної системи біржі спостережень».

Основною ціллю роботи є розробка компонентів інформаційної системи біржі спостережень, що також включає в себе детальний аналіз розроблених процесів та функцій на теоретичному і практичному рівні. Для підтвердження унікальності дослідження було проведено апробації на наукових конференціях та в публікаціях.

Обрана тема була представлена на конференції «Інформаційні інтелектуальні системи» в м. Харків, 16-18 квітня 2024 року. На цій конференції була представлена доповідь на тему «Розробка компонентів замовлення розробки та використання ПЗ інформаційної системи біржі спостережень». Вона включала опис прийнятих рішень та опис архітектури розробленої інформаційної системи. Було проаналізовано методи оптимізації та покращенні системи обробки даних та визначено актуальність теми для певних категорій користувачів та бізнесу.

Проведена апробація дозволила визначити варіанти подальшого розвитку та покращення ефективності взаємодії користувача із системою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Библиотека React: особенности, перспективы, ситуация на рынке труда. Хекслет. URL: <https://ru.hexlet.io/blog/posts/biblioteka-react-review-article> (дата звернення: 13.05.2024).
2. Importing a Component | Create React App. Create React App. URL: <https://create-react-app.dev/docs/importing-a-component/> (дата звернення: 13.05.2024).
3. Петров Е.Г., Новожилова М.В., Гребеннік І.В., Соколова Н.А. Методи та засоби прийняття рішень у соціально-економічних та технічних системах: Херсон: Олді – плюс, 2003. – 380 с.
4. І.В.Гребеннік, М.Ю.Вишняк, В.Г.Іванов, З.А.Імангулова, Н.І.Калита Елементи системного проектування (за редакцією І.В.Гребенніка): Навч. посібник. – Харків: ХНУРЕ, 2016. – 322 с.
5. Наконечний О. Г., Гребеннік І. В., Романова Т. Є., Тевяшев А. Д., Методи прийняття рішень: Навч. посібник. – Харків: ХНУРЕ, 2016. – 132 с.
6. Гребеннік І.В., Коваленко А.І., Міщеряков Ю.В., Решетнік В.М., Титов С.В. Системне програмування – Х.:ХНУРЕ, 2017 – 374 с. // I. Grebennik, A. Kovalenko, I. Mishcheriakov, V.Reshetnik, S.Titov System programming: Tutorial. Kh.: NURE, 2018 – 374 p.
7. Нефьодов Л. І., Невлюдов І. Ш., Безкоровайний В. В. CALS-технології і системи: навч. посібник. Харків: ХНУРЕ, 2021. 272 с.
8. Documentation for MySQL-server: офіційна документація. URL: <https://www.mysql.com/> (дата звернення: 01.05.2024).
9. Dev.java: The Destination for Java Developers. Dev.java: The Destination for Java Developers. URL: <https://dev.java/> (дата звернення: 26.05.2024).
10. Учасники проектів Вікімедіа. Публікація-підписка (шаблон проектування) – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Публікація-підписка_\(шаблон_проектування\)](https://uk.wikipedia.org/wiki/Публікація-підписка_(шаблон_проектування)) (дата звернення: 26.05.2024).
11. Documentation - MariaDB.org. MariaDB.org. URL: <https://mariadb.org/documentation/> (дата звернення: 26.05.2024).

12. Introduction | v6 | StarUML documentation. Introduction | v6 | StarUML documentation. URL: <https://docs.staruml.io/> (дата звернення: 26.05.2024).
13. Spring Boot :: Spring Boot. Spring | Home. URL: <https://docs.spring.io/spring-boot/index.html> (дата звернення: 26.05.2024).
14. Add API documentation in Postman | Postman Learning Center. Postman Learning Center. URL: <https://learning.postman.com/docs/publishing-your-api/documenting-your-api/> (дата звернення: 26.05.2024).
15. Getting Started | React.js and Spring Data REST. Getting Started | React.js and Spring Data REST. URL: <https://spring.io/guides/tutorials/react-and-spring-data-rest> (дата звернення: 26.05.2024).
16. CRUD Application With React and Spring Boot | Baeldung. Baeldung. URL: <https://www.baeldung.com/spring-boot-react-crud> (дата звернення: 26.05.2024).
17. Учасники проектів Вікімедіа. IDEF0 – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/IDEF0> (дата звернення: 26.05.2024).
18. Путілов С.Ю. Розробка компонентів замовлення розробки та використання програмного забезпечення інформаційної системи біржі спостережень: 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. 2024. С. 250-251
19. useEffect – React. React. URL: <https://react.dev/reference/react/useEffect> (дата звернення: 26.05.2024).
20. UML Class Diagram Tutorial. Lucidchart. URL: <https://www.lucidchart.com/pages/uml-class-diagram> (дата звернення: 26.05.2024).
21. Quick Guide to Spring Controllers | Baeldung. Baeldung. URL: <https://www.baeldung.com/spring-controllers> (дата звернення: 26.05.2024).
22. Bilyk V. Data Flow Diagrams (DFD) Explained. ArtofBA. URL: <https://www.artofba.com/uk/post/data-flow-diagrams-dfd-explained-1> (дата звернення: 26.05.2024).
23. Java Basics | Basics of Java - Javatpoint. www.javatpoint.com. URL: <https://www.javatpoint.com/java-basics> (дата звернення: 26.05.2024).