

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ центр післядипломної освіти _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський) _____

Програмна система обліку комерційних пропозицій виробничого підприємства _____
(тема)

Виконав:
студент 4 курсу, групи ПЗПП-22-2 _____

_____ Махлай К. О. _____
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

Освітня програма Програмна інженерія _____
(повна назва освітньої програми)

Керівник доц. Лановий О.Ф. _____
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

_____ З.В.Дудар _____
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ центр післядипломної освіти
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ перший (бакалаврський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ Освітньо-професійна
 Освітня програма _____ Програмна Інженерія
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Махлай Костянтину Олександровичу
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмна система обліку комерційних пропозицій
 виробничого підприємства _____

Затверджена наказом по університету від 17.06. 2024р. № 588 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 14.06.2022

3. Вихідні дані до роботи Розробити систему обліку комерційних пропозицій
 виробничого підприємства, що дозволяє виконувати розрахунки обладнання,
 генерувати комерційні пропозиції а також вести архів.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи,
 архітектура та проектування програмного забезпечення, опис прийнятих
 програмних рішень, тестування розробленого програмного забезпечення,
 висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	20.05.2024	<i>виконано</i>
2	Створення специфікації ПЗ	22.05.2024	<i>виконано</i>
3	Проектування ПЗ	24.05.2024	<i>виконано</i>
4	Розробка ПЗ	28.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	05.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2024	<i>виконано</i>
8	Попередній захист	09.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	06.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	10.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	11.06.2024	<i>виконано</i>

Дата видачі завдання 16 травня 2024р.

Студент (ка) _____

(підпис)

_____ Махлай К. О.

Керівник роботи _____

(підпис)

_____ доц. Лановий О.Ф.

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра: 70 с., 23 рисунків, 2 таблиці, 12 джерел.

МОВА ПРОГРАМУВАННЯ PYTHON, РЕЛЯЦІЙНА БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, POSTGRESQL, DJANGO, API, GOOGLE SHEET, GOOGLE DOCS.

Метою кваліфікаційної роботи є розробка Web додатку на базі Django додатку «Система обліку комерційних пропозицій виробничого підприємства» з використанням реляційних баз даних, API інтерфейсів та логіки об'єктно-орієнтованого програмування.

В ході створення додатку були використані середа розробки GetBrain PyCharm, мова програмування Python, ПЗ для полегшення розгортання додатку Docker, Web framework Django, база даних PostgreSQL.

В результаті роботи був створений додаток, який дозволяє взаємодіяти та зберігати проекти будівництва очисних споруд, а саме проекти, технологічні вузли, позиції та данні стосовно обладнання. Також програма дозволяє шукати, фільтрувати інформацію, отримувати статистику підбору обладнання, генерує зводні специфікації до проектів, дозволяє виконувати автоматичні розрахунки собівартості обладнання з використанням Google sheets API, та завантажувати комерційні пропозиції з використання Google docs API

PYTHON PROGRAMMING LANGUAGE, RELATIONAL DATABASE, DATABASE MANAGEMENT SYSTEM, POSTGRESQL, DJANGO, API, GOOGLE SHEET, GOOGLE DOCS.

The purpose of the diploma project is to develop a Web application "Manufacturing Enterprise Commercial Offers Accounting Software System" based on the Django application with usage of relational databases, API interfaces and object-oriented programming logic.

During the application creation were used the JetBrains PyCharm development environment, Python programming language, Docker application deployment software, Django Web framework, PostgreSQL.

Developed application allows to create and save projects for the wastewater treatment plant, as well as projects, technological nodes, technological positions and equipment related data. The program also allows to search and filter information, gets statistics of definite equipment, generates summary specifications for projects, provides automatic equipment cost calculations using the Google sheets API, and generate and download commercial offers using the Google docs API.

Я, Махлай Костянтин Олександрович, студент гр. ПЗПП 22-2, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система обліку комерційних пропозицій виробничого підприємства», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання. Я ознайомлений(а) з діючим положенням «Про протидію

академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП.....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	11
1.1. Аналіз предметної галузі.....	11
1.2. Аналіз існуючих аналогів.....	13
1.3. Виявлення проблем.....	21
1.4. Постановка задачі.....	22
2. ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ	24
3. АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ.....	27
3.1. Моделювання бізнес процесів.....	28
3.2. Концептуальне моделювання.....	33
3.3. Побудова логічної моделі бази даних.....	36
4. ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ.....	36
4.1 Загальні відомості.....	36
4.2 Виклик і завантаження.....	37
4.3 Опис програмної реалізації.....	40
5. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	49
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	54
Додаток А.....	56
Додаток Б.....	62
Додаток В.....	64
Додаток Г.....	65
Додаток Г.....	67

ВСТУП

У сучасному світі, де стрімкий розвиток інтернету та програмних застосунків веде до трансформації традиційних процесів у цифровий формат, спостерігається значне зростання використання автоматизованих систем у різних сферах діяльності. Якщо ще 20 років тому значна частина підприємств обходилася без спеціалізованого програмного забезпечення, то нині такі підприємства є великою рідкістю. Особливо актуальним це стає у таких галузях, як проектування очисних споруд, які використовують власні виробничі потужності.

Кваліфікаційна робота зосереджений на розробці веб застосунку для детального вивчення та управління процесами обліку у сфері проектування. Обрана тема включає систематизацію даних про обладнання, використовуване у проектах, що є надзвичайно перспективним напрямком для впровадження систем автоматизації. Важливість цього дослідження обумовлена інтеграцією сучасних технологічних рішень у процеси планування та реалізації проектних завдань на підприємствах, зокрема в очисних спорудах.

Під час виконання кваліфікаційної роботи було виконано ER-моделювання, а також концептуальне та логічне моделювання реляційної бази даних, за якою була здійснена її фізична реалізація. В результаті розроблено додаток, що дозволяє взаємодіяти з даними про структуру проекту, включаючи обладнання, його постачальників, характеристики, комплектацію та відомості про застосування в окремих проектах. Програма також забезпечує функції пошуку, фільтрації інформації та отримання статистики, підтримує автоматизацію розсилки листів.

Для реалізації проекту були використані середовище розробки JetBrains PyCharm, веб-фреймворк Django, мова програмування Python та система управління базами даних PostgreSQL. Додаток розроблено таким чином, що

його інтерфейс є інтуїтивно зрозумілим, а функціонал максимально покриває потреби підприємства в області контролю проходження обладнання за проектами.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Предметна галузь підприємства, що виробляє обладнання для очисних споруд каналізації, пов'язана з екологічними технологіями та водоочисними системами. Очисні споруди каналізації є необхідним елементом інфраструктури міст та населених пунктів, які забезпечують відведення стічних вод від населення та промислових підприємств.

Підприємства, що виробляють обладнання для очисних споруд каналізації, мають важливу роль у забезпеченні якісної та ефективної роботи систем очищення стічних вод. Вони забезпечують виробництво різноманітного обладнання, такого як насоси, фільтри, реактори, аератори, трубопроводи та інші компоненти, необхідні для побудови та експлуатації очисних споруд.

Однією з ключових проблем, з якими стикаються підприємства, є постійна необхідність розробки та впровадження нових технологій та обладнання, які б дозволили забезпечити ефективність та якість очищення стічних вод. Крім того, підприємства повинні враховувати різноманітні фактори, такі як вплив кліматичних умов, географічні особливості регіону, технологічні вимоги та інші фактори.

Однак, не дивлячись на технічну складність та високу вимогливість предметної галузі, підприємства, що виробляють обладнання для очисних споруд каналізації, є важливими учасниками на ринку екологічних технологій та забезпечують ефективну та безпечну роботу систем очищення стічних вод.

Зазвичай такі підприємства поставляють не саме обладнання, а комплексне рішення для тієї чи іншої потреби. Такі рішення складаються з

великої кількості обладнання. Ключовим моментом є те, що такі рішення оформлюються в проекти з детальними розрахунками того чи іншого рішення, ефективності та вартості.

Саме в розробці пропозицій приймають участь спеціалісти з різних галузей, такі як:

- технологи (виконують розрахунки технологічних процесів, ефективність і доцільність застосування тієї чи іншої технології...);
- проектувальники (розробляють рішення про розміщення того чи іншого технологічного рішення, підключення обладнання, обв'язка обладнання комунікаціями...);
- спеціалісти з обладнання (роблять технологічні й технічні розрахунки обладнання, готують технічну документацію на обладнання).

На першому етапі розробки технологи роблять основні технологічні розрахунки технологічного вузла. Після чого вони передають свої розрахунки до проектувальників які базуючих на них мають розробити проект для будівництва чи модернізації. В процесі розробки проектної документації проектувальником кожному технологічному вузлу, та кожній позиції присвоюється технологічний номер за проектом.

Після присвоєння номерів технологічним позиціям проектувальники створюють запит на підбір обладнання до профільних фахівців з обладнання того чи іншого призначення. При формуванні запиту обов'язково вказується позиція за проектом, а також вимоги до обладнання.

Після підбору обладнання спеціаліст з підбору обладнання повертає усю необхідну документацію проектувальнику. При частими є ситуації за яких необхідно виконувати повторній підбір обладнання, робити запити на роз'яснення чи уточнення даних. Зважаючи на кількість позицій у кожному проекті так кількість місць де ця інформація дублюються, занесення та ведення

обладнання у проекті стає справньою проблемою. До того ж з урахуванням сучасного темпу проектування зміни відбуваються з великою частотою, що призводить до утворення “блоків” саме на етапі додавання та контролю обладнання.

1.2 Аналіз існуючих аналогів

Для виконання завдань аналогічних поставленим найбільш часто використовують наступні ПЗ, які здебільшого частково покривають необхідний функціонал:

- microsoft excel;
- google sheets;
- autodesk revit;
- jira.

Excel - це програма для обробки даних та створення електронних таблиць, яка широко використовується для аналізу, візуалізації та обчислення різних видів інформації. Є одним з найпоширеніших застосунків, зокрема через свою розповсюдженість. Дозволяє з легкістю додавати, сортувати та обробляти дані. Але вагомим її мінусом є доволі незручний і не завжди вдалий колективний доступ до документа. Робити якусь автоматизацію можна, для цього навіть є вбудований редактор коду, але по перше інтеграція з іншими сервісами викликає труднощі, оскільки скрипти пишуться на VBA і орієнтовані здебільшого на роботу у межах одного комп'ютеру чи мережі.

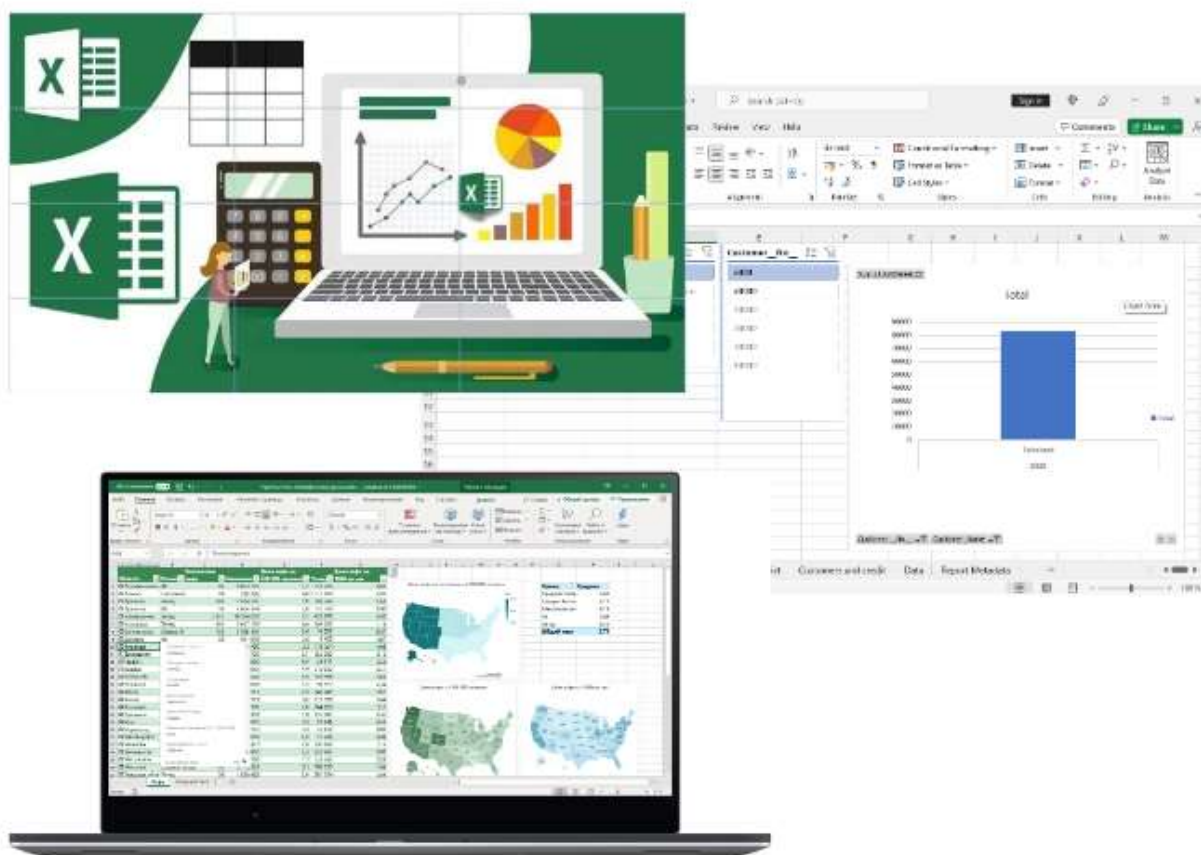


Рисунок 1.1 – Робочий екран Excel

До переваг можна віднести:

- зручність використання: простий та інтуїтивно зрозумілий інтерфейс полегшує роботу з таблицями для користувачів будь-якого рівня;
- розширені Функції та Формули: має широкий спектр вбудованих функцій та формул для автоматизації обчислень та аналізу даних;
- Візуалізація Даних: забезпечує можливість створювати графіки, діаграми та інші візуальні елементи для кращого розуміння даних;
- сумісність та Розповсюдженість: широко використовується в різних галузях та секторах, та має велику спільноту користувачів;
- можливості Фільтрації та Сортування: забезпечує зручні можливості для фільтрації та сортування даних;
- вбудований редактор коду.

До недоліків можна віднести наступне:

- обмежена робота з великими обсягами даних: при великій кількості даних може виникати обмеження швидкодії та продуктивності;
- не ідеальна для колективної роботи: ускладнює колективну роботу, оскільки може виникати проблема з контролем версій та синхронізацією даних;
- відсутність централізованого управління даними: відсутність централізованого контролю може вести до розсіяності даних та ризику втрати інформації;
- не ефективний для складних звітів та аналітики: для складних звітів та аналітики може виявитися не настільки потужним, як спеціалізовані інструменти;
- брак контролю доступу: не завжди забезпечує ефективний контроль доступу до окремих елементів таблиці;
- складнощі в інтеграції з іншими системами за межами Microsoft Office.

Google Sheets - це інтерактивна електронна таблиця, яка входить до складу Google Workspace (раніше G Suite). Забезпечує можливість створення, редагування та спільне використання таблиць в хмарному середовищі. Являє собою найближчий аналог Microsoft Excel, в якому виправлений недолік - робота з декількома редакторами одночасно.

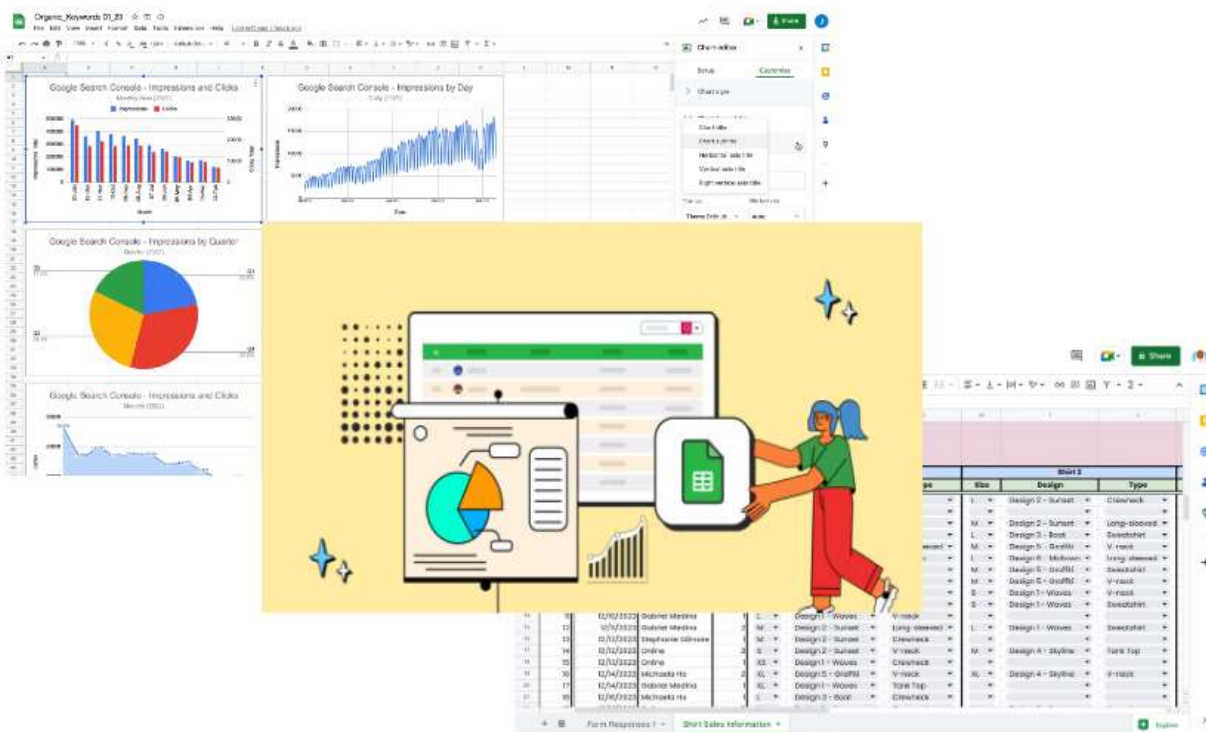


Рисунок 1.2 – Робочий екран Google Sheets

Попри це не набула широкого розповсюдження для такого типу задач оскільки функціонал, зазвичай не повний і не зручний. Окремо варто відмітити можливість написання повноцінних скриптів і функцій за допомогою AppsScript від Google, що відкриває широкі можливості для розробки і інтеграції

До переваг можна віднести:

- хмарна система та спільне використання: можливість колективної роботи, оскільки дані автоматично синхронізуються в хмарному середовищі та доступні для спільного використання;
- зручний інтерфейс та інтуїтивне використання: має інтуїтивно зрозумілий і простий інтерфейс, що полегшує використання користувачам будь-якого рівня;
- розширені можливості функцій та формул: надає широкий спектр функцій та формул, що забезпечують автоматизацію обчислень;

- можливості візуалізації даних: надає можливості створення графіків, діаграм та інших візуальних елементів для кращого розуміння даних;
- інтеграція з іншими засобами google: інтегрується з іншими засобами Google, такими як Gmail, Google Drive, що полегшує обмін та зберігання даних.

До недоліків можна віднести наступне:

- бажаним до гнучкості індивідуальних налаштувань: для користувачів, які шукають високий рівень гнучкості та індивідуальних налаштувань, Google Sheets може видаватися менш гнучким у порівнянні із локальними аналогами;
- брак повних засобів захисту даних: може виникати питання щодо повноцінності захисту даних, особливо при роботі з конфіденційною інформацією;
- обмежені можливості роботи з великими обсягами даних: для великих обсягів даних може виявитися менш ефективним порівняно з деякими локальними аналогами.

Autodesk Revit - спеціалізоване програмне забезпечення для BIM (Building Information Modeling), що забезпечує проектування, моделювання, аналіз та документування будівельних об'єктів. Revit широко використовується у сфері архітектури та будівництва для створення детальних 3D моделей з повною візуалізацією та автоматизованими засобами для аналізу будівельних процесів.

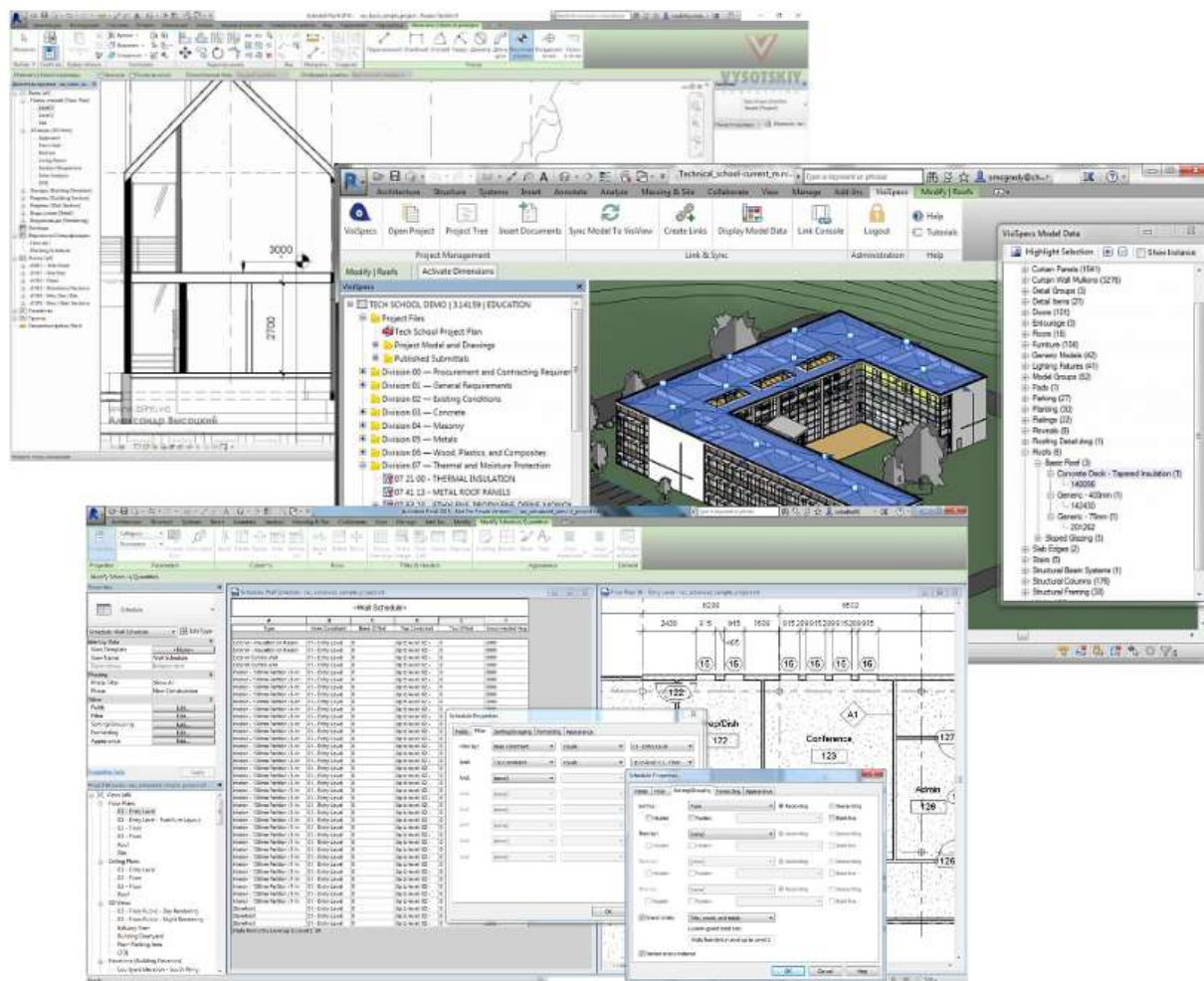


Рисунок 1.3 – Робочий екран Revit

Насправді Revit - надзвичайно потужний інструмент для проектування і в нього є безліч можливостей, але в даному контексті ми розглянемо його як програмний засіб що дозволяє вести обладнання крізь проект та проектні специфікації.

До переваг Revit слід віднести:

- інтегрованість проектування: Можливість працювати одночасно над архітектурними, інженерними та конструкційними аспектами проекту;
- висока точність моделювання: Забезпечення точності розмірів та координації між різними частинами проекту;

- автоматизація документації: Автоматичне оновлення креслень, специфікацій та звітів при зміні моделі;
- ефективне управління ресурсами: Підвищена ефективність планування та використання матеріалів завдяки інтегрованим інструментам аналізу.

Недоліками же є:

- складність освоєння: Потребує значних навичок та тренінгів для ефективного використання;
- вимоги до апаратного забезпечення: Високі вимоги до комп'ютерного обладнання для роботи з великими проектами;
- обмежена сумісність: Потенційні проблеми при обміні даними з іншими програмами, якщо вони не підтримують формати файлів Revit;
- неможливість повноцінного контролю за проходженням обладнання по стадіям проектування.

Jira – це відомий інструмент для управління проектами, розроблений Atlassian. Він широко застосовується в IT-індустрії для управління завданнями, справами та помилками, а також для спрощення процесу спільної розробки.

Цей інструмент не дуже часто застосовується для ведення проектів поза IT сектором, але незважаючи на це має ряд ключових переваг що могли б бути використані в проектуванні.

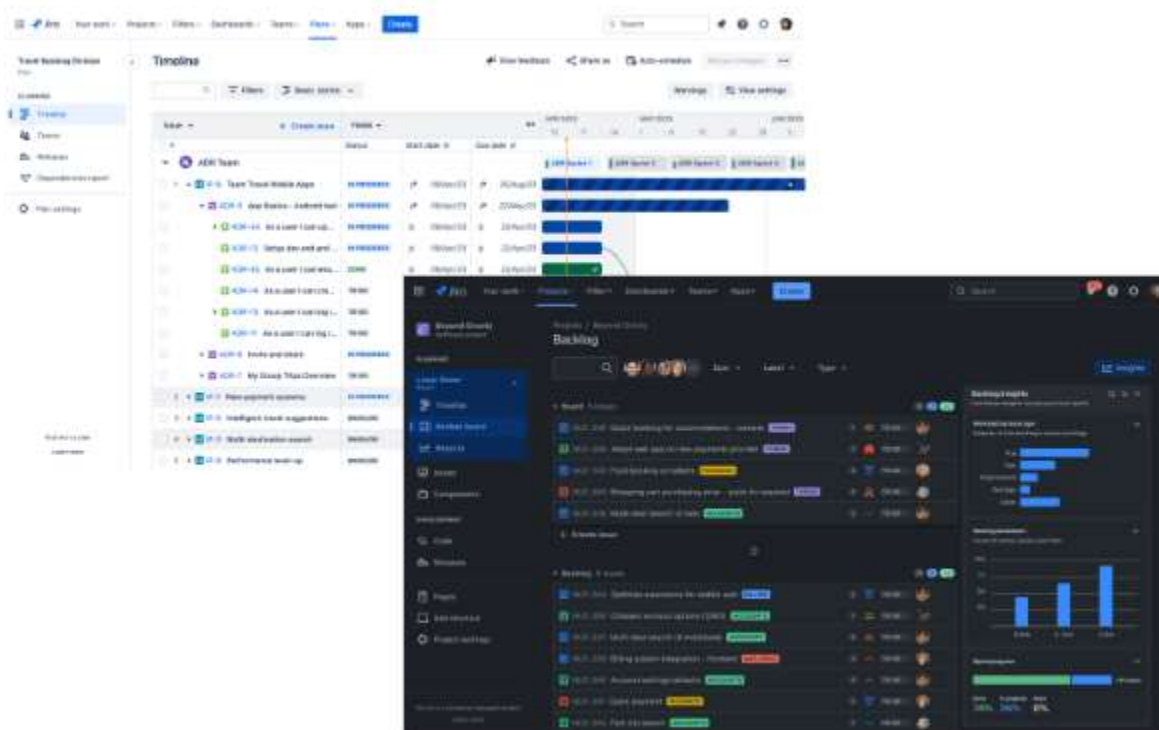


Рисунок 1.4 – Робочий екран Jira

А саме до таких переваг слід віднести:

- гнучкість конфігурації: Можливість налаштувати Jira під специфіку будь-якого проекту або команди;
- інтеграція з іншими інструментами: Легко інтегрується з різними програмами для розробки, комунікації та моніторингу;
- підтримка агільних методологій: Забезпечує підтримку Scrum та Kanban, сприяючи ефективній організації робочих процесів;
- контроль проходження завдань та процесів (є чітке розуміння хто, що робить, та які завдання є).

Недоліками можна вважати:

- складність інтерфейсу: Інтерфейс може здатися перевантаженим для нових користувачів;

- висока вартість: Підписка може бути дорогою, особливо для великих команд;
- значний час налаштування: Потрібен час та зусилля для оптимізації системи під конкретні потреби команди або проекту.

1.3 Виявлення проблем

Усі з наведених варіантів мають як сильні так і слабкі сторони. Так Excel та Google sheet добре впораються з розрахунками та обробкою табличних даних, але не в повній мірі задовольняють потреби бізнесу. Зокрема недоліком цих систем є необхідність зберігати великий масив даних у одному файлі, або ж робити купу пов'язаних таблиць (по суті реляційну базу даних) не дуже зручно у користуванні. Ще одним недоліком цих рішень є брак контролю доступу та розподілення прав в залежності від типу доступу, а також практично повна відсутність контролю версій та проходження процесів. До того ж, хоча більшість процесів і може бути автоматизована, але гнучкість таких рішень під великим сумнівом.

В то й час яке Jira максимально ефективно виконує функції контролю версій дозволяючи відслідковувати весь flow процесу проектування таким чином щоб можна було відслідкувати кожний процес і разі потреби вжити заходів.

Revit же в свою чергу надає користувачеві дуже потужний інструмент для формування специфікацій до проекту, та їх оновлення. Але цей застосунок вимагає максимально повного виконання проекту перед тим як можна буде сформувавши специфікацію.

Також є один суттєвий недолік який об'єднує усі ці рішення, це то що усі ці рішення є програмним продуктом сторонніх компаній, що дуже сильно блокує потенціал до розширення оскільки при інтеграції чи кастомізації ми






можемо орієнтуватися лише на певний перелік налаштувань на установок, що значно звужує можливості для подальшого масштабування. У той час я власне програмне забезпечення, хоча у капітальних затратах перевищує використання готових рішень, у перспективі надасть значну економію часу та засобів на подальше розширення.

1.4 Постановка задачі




З вищесказаного можна зробити висновок, що кожен з розглянутих застосунків покриває лише частину потреб бізнесу. Таким чином необхідно створити програмний додаток, який увібрав би в себе всі найкращі риси кожного з розглянутих додатків і таким чином задовольнив би потреби бізнесу.

На основі цього можна скласти порівняльну таблицю (див. Таблиця 1.1.).

Таблиця 1.1 - Порівняння основних конкурентів

					
	Excel	G-Sheet	Revit	Jira	Our
Складні розрахунки	+	+	+	-	+
Комплексні розрахунки	+	+	+	-	+
On-line	-	+	-	+	+
Формування звітів	+	+	+	+	+

Продовження таблиці 1.1.

					
	Excel	G-Sheet	Revit	Jira	Our
Формування специфікацій	-	-	+	-	+
Система відстеження задач	-	-	-	+	+
Можливість заповнення шаблонів комерційних пропозицій	+-	+-	-	-	+
Можливість інтеграції зі сторонніми застосунками	+-	+	-	+	+
Багатокористувацький режим	+-	+	-	+	+
Можливості для кастомного розширення функціоналу	-	-	-	+	+

Таким чином, як видно з таблиці 1.1 додаток повинен включати в себе найсильніші сторони розглянутих застосунків в конкретному контексті задач підприємств. Гнучкість та контроль над задачами як у Jira, автоматизацію формування документації та зручність її перегляду як у Revit. А також обчислювальні потужності як у Excel чи G-Sheet.

2. ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Необхідно програмний продукт для внутрішньо виробничих потреб підприємства, що займається виробництвом обладнання для очисних споруд каналізації і їх проектуванням. Крім того система має забезпечити швидке та доступне управління розрахунками та системою формування комерційних пропозицій. Система дозволяє вести облік яке обладнання, коли і в якій кількості було запропоновано якому замовнику, а також отримувати статистику за запитам.

Логічно система має бути розділена на наступні функціональні сервіси:

- секція проекту - функціональний розділ що дозволяє зберігати в базі дані про проект та його структуру, статуси конкретних позицій та іншу релевантну інформацію;
- секція архіву - дозволяє оперативно перевіряти входження конкретного типу обладнання по проектам, перевіряти дані;
- секція розрахунків - призначений для розрахунків технічних характеристик то собівартості обладнання;
- секція комерційних пропозицій - дозволяє формування на підставі виконаних розрахунків комерційних пропозицій.

Система забезпечуватиме наступні функції для кожної секцій:

Секція проекту:

- а) данні об'єкта (місто, замовник, організація);
- б) данні щодо місця встановлення обладнання (технологічний вузол);
- в) данні стосовно запропонованого обладнання (маркування, технічні характеристики, собівартість, номер позиції за технологічною схемою..);
- г) перегляд та редагування структури проекту;

- д) можливість створювати запит та повторний запит на підбір обладнання;
- е) архівувати обладнання;
- ж) видаляти обладнання;
- и) отримати специфікацію обладнання;
- к) кошторис проекту;
- л) отримати детальну інформацію за кожною позицією:
 - 1) вихідний запит;
 - 2) чат по кожній позиції;
 - 3) технічні характеристики;
 - 4) собівартість.

Секція архіву:

Архів в свою чергу має бути розділений на дві підгрупи:

- комерційний архів - архів, що містить дані для формування кошторисів і не перенавантажений технічними даними;
- технічний архів - архів по кожному типу обладнання, що включає в себе повний перелік технічних характеристик обладнання і має можливість їх сортування, групування та експорту.

Кожен з наведених типів архіву має включати в себе наступний функціонал:

- сортувати обладнання за об'єктами, вагою, назвою, вартістю, тощо;
- здійснювати пошук інформації про обладнання за його маркуванням або частиною маркування;
- здійснювати фільтрацію інформації за об'єктами, технологічними вузлами.

Секція розрахунків

- можливість виконувати розрахунки усіх типів обладнання;
- виконувати переоцінку у разі необхідності;

- експортувати розрахунки;
- можливість швидко та просто оновлювати змінні для розрахунків (вартість комплектуючих, робіт, сировини...);
- можливість швидко та просто оновлювати алгоритми розрахунків у разі змін;
- можливість зв'язування розрахунків з обладнанням за проектами;
- експорт даних;
- створення та експорт комплексних кошторисів.

Секція комерційних пропозицій:

- створення комерційної пропозиції з описом та технічними характеристиками обладнання з архіву;
- простий та швидкий механізм редагування та зміни шаблонів комерційних пропозицій;
- можливість експорту комерційних пропозицій.

3. АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ

3.1. Моделювання бізнес процесів

Опис функціонального призначення ІС, що створюється наведено на рисунку 3.1.

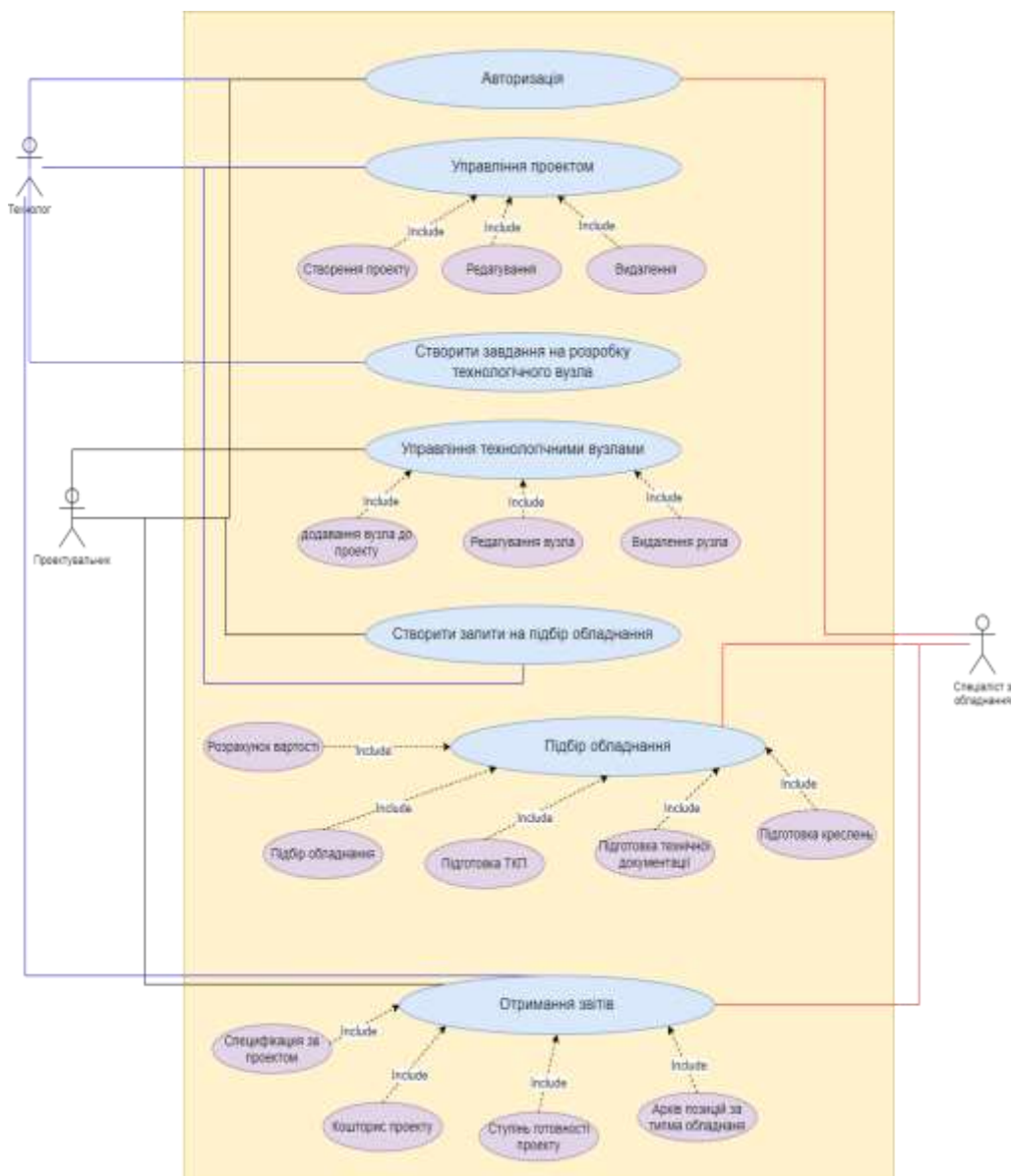


Рисунок 3.1 – Загальна USE-CASE діаграм

3.2. Концептуальне моделювання

Опис концептів програмної області, їх властивостей та зв'язків між ними зобразимо у вигляді загальної діаграми класів представлено на рисунку 3.2.

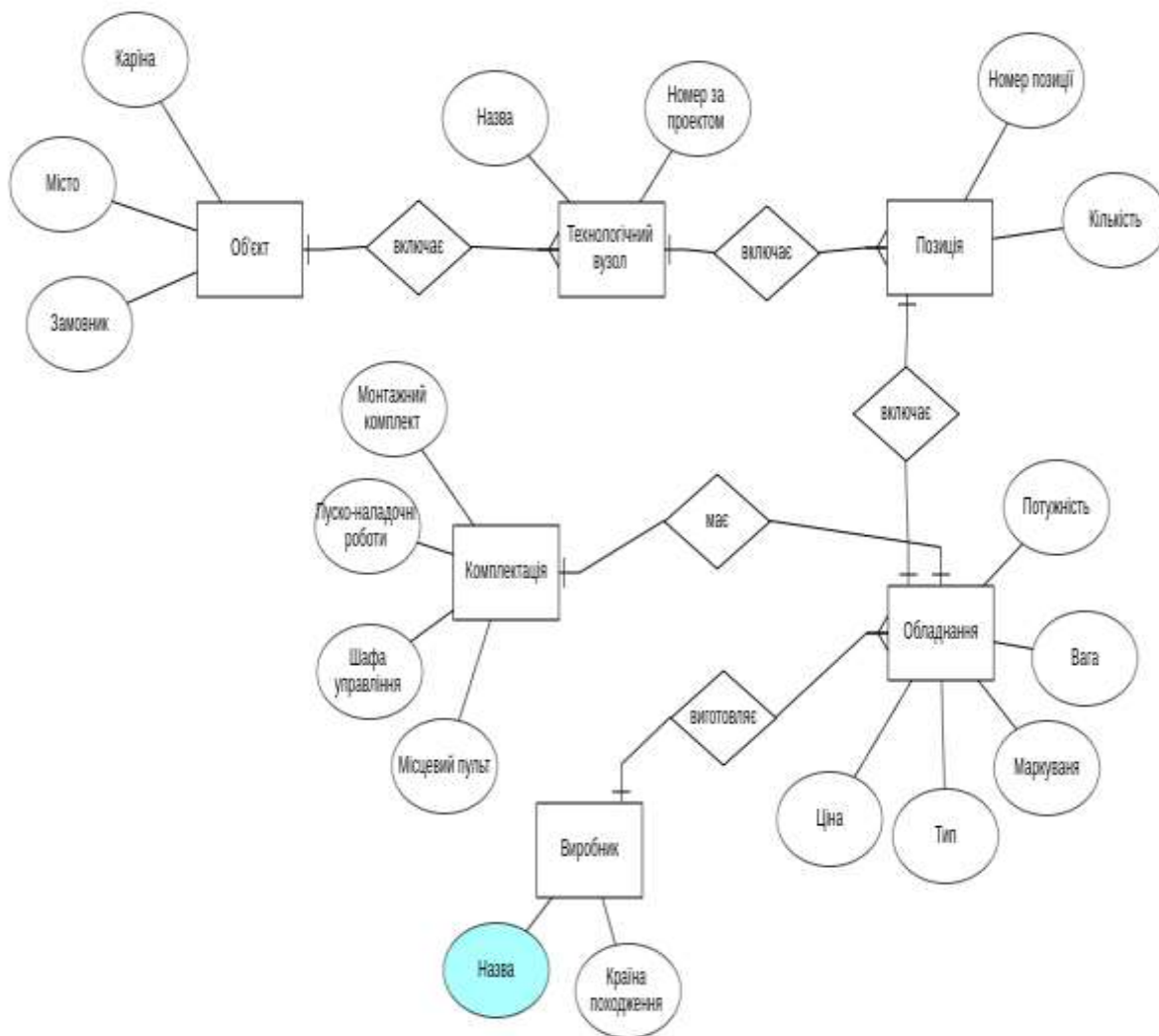


Рисунок 3.2 – Загальна діаграма класів

Зазначимо перелік характеристик для кожного з наведених понять. Поняття об'єкт має наступні властивості: країна, місто, замовник. Поняття технологічний вузол включає в себе найменування, номер вузла за проектом. Позиції мають зазначені атрибути: номер позиції за проектом, кількість, заархівно ван чи ні. Поняття обладнання - маркування, ціна, тип обладнання,

вага, потужність. Модель виробник – назва та країна походження. Комплектація включає наступні елементи — шафа управління, локальний пульт, ПНР роботи, монтажний комплект.

Для спрощення роботи технологів під час роботи з інформацією вони повинні мати можливість:

- створювати та редагувати новий проект;
- робити запит до проектувальників на розробку технологічних вузлів;
- переглядати специфікацію обладнання;
- робити вибірку та фільтрацію обладнання за типами обладнання, спорудами тощо.

Технологи для спрощення аналізу інформації з проекту повинні мати можливість отримання наступної статистики:

- специфікацію обладнання;
- стан підбору обладнання (підібране чи ні);
- переглядати кошторис проекту.

Для спрощення роботи проектувальників під час роботи з інформацією вони повинні мати можливість:

- створювати та редагувати технологічні вузли;
- робити запит до спеціалістів з обладнання на підбор обладнання;
- переглядати специфікацію обладнання;
- робити вибірку та фільтрацію обладнання за типами обладнання, спорудами тощо.

Проектувальники для спрощення аналізу інформації з проекту повинні мати можливість отримання наступної статистики:

- специфікацію обладнання;
- стан підбору обладнання (підібране чи ні);

Для спрощення роботи спеціалістів з обладнання під час роботи з інформацією вони повинні мати можливість:

- переглядати запити на підбір обладнання;
- повертати запити для уточнення інформації;
- переглядати специфікацію обладнання;
- робити вибірку та фільтрацію обладнання за типами обладнання, спорудами тощо.

Спеціалісти з підбору обладнання для спрощення аналізу інформації з проекту повинні мати можливість отримання наступної статистики:

- специфікацію обладнання;
- кошторис проекту.

Запропоновану автоматизацію зображено за допомогою діаграми видів діяльності (див. Рис. 3.3).

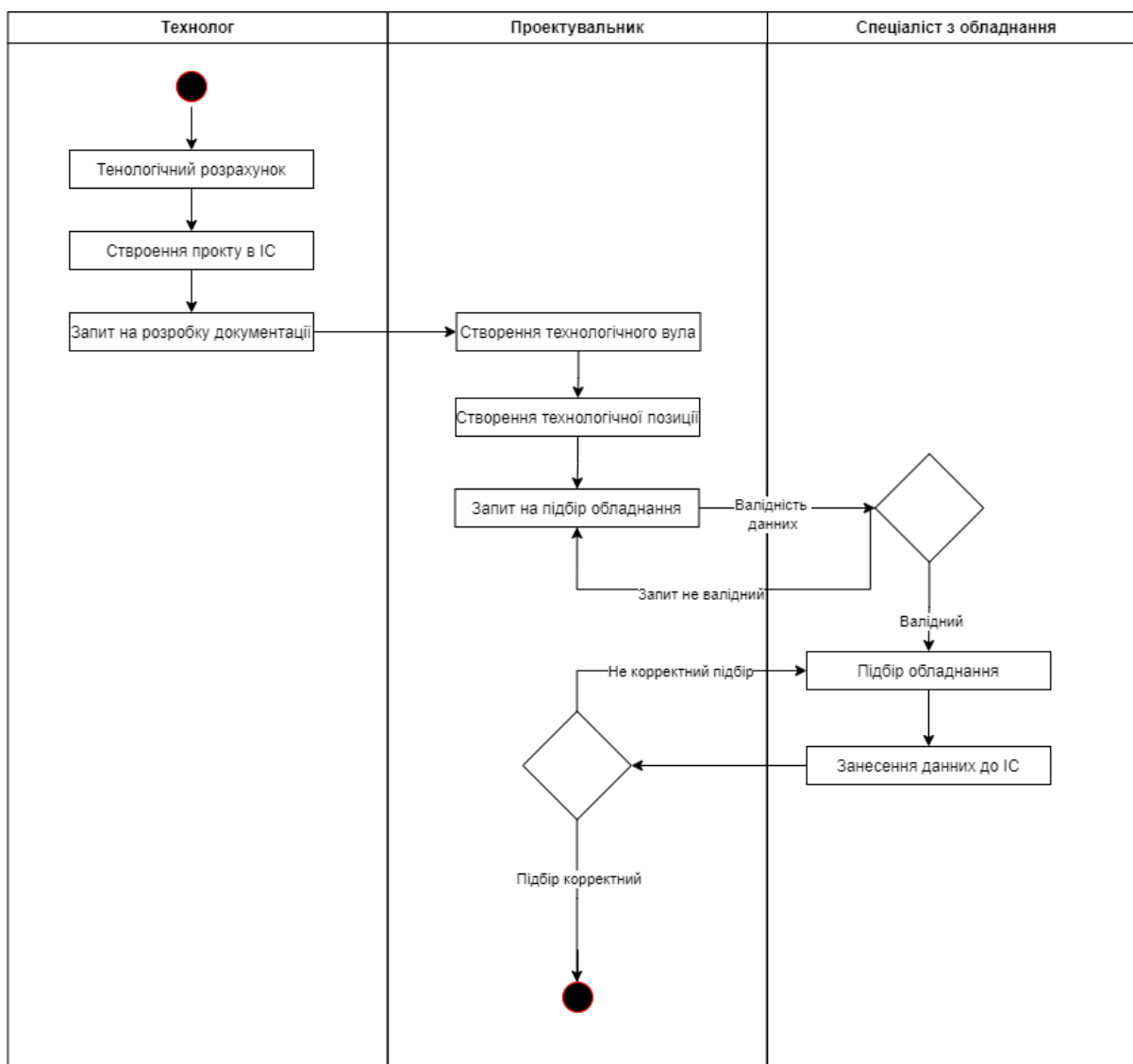


Рисунок 3.3 – Діаграма видів діяльності

Опис існуючого документообігу складається з наступних документів:

а. «Специфікація»:

1. назва проекту;
2. назва технологічного вузла;
3. позиція за проектом;
4. кількість;
5. маркування;
6. вага;

7. потужність.

б. «Кошторис»:

1. назва проекту;
2. назва технологічного вузла;
3. позиція за проектом;
4. кількість;
5. маркування;
6. вартість.

В проблемній області існує ряд обмежень, які можна віднести до обмежень цілісності стосовно зв'язків:

- кожен об'єкт може мати кілька технологічних вузлів, але технологічний вузол може мати лише один об'єкт;
- кожен технологічний вузол може мати безліч позицій обладнання, але кожна позиція може бути прикріплена лише до одного технологічного вузла;
- позиція може мати кілька одиниць обладнання, але при цьому лише обрана як основне обладнання, а всі інші архівні.

З урахуванням ліцензійного ПЗ, що є в на підприємстві необхідно дотримуватися наступних обмежень:

- всі електронні таблиці формувати у форматах сумісних з редактором Microsoft Excel;
- всі текстові документи формувати у форматах сумісних з редактором Microsoft Word;
- використовувати СУБД SQLite, PostgreSQL.

Лінгвістичні відносини, що існують в ПО, повинні нормалізуватися документацією з організації та термінологією, що прийнята в галузі роботи виробничих підприємств, наприклад:

- ТКП (техніко-комерційна пропозиція) – документ що містить технічний опис та характеристики обладнання, комплектацію, умови та терміни поставки, вартість;
- Специфікація — таблиця що містить базові технічні характеристики обладнання з вказанням місця встановлення (унікальний ідентифікатор за проектом);
- Кошторис - таблиця що ціни обладнання з вказанням місця встановлення (унікальний ідентифікатор за проектом).

3.3 Побудова логічної моделі бази даних

Реляційна база даних є системою зберігання та організації доступу до взаємопов'язаних елементів інформації [12]. Ця база даних базується на реляційній моделі, яка представляє інформацію у вигляді таблиць. Кожен рядок у таблиці має унікальний ідентифікатор - ключ, а стовпці таблиць містять атрибути даних, що дозволяє встановлювати зв'язки між елементами даних [7]. Реляційна модель бази даних є найбільш ефективною для вирішення задачі кваліфікаційної роботи.

Нормальна форма є вимогою до структури таблиць в теорії реляційних баз даних для усунення зайвих функціональних залежностей між атрибутами.

В даному проекті при нормалізації виконується приведення таблиць до третьої нормальної форми. Відношення знаходиться у третій нормальній формі, коли воно знаходиться у першій та другій нормальній формі, тобто всі атрибути атомарні, кожний не ключовий атрибут повністю функціонально залежить від первинного ключа відношення, та між не ключовими атрибутами немає транзитивних залежностей.

В усіх відношеннях всі атрибути є атомарними, тому вони знаходяться в 1 НФ.

При зведенні до другої нормальної форми було доведено, що всі відношення не містять неповних функціональних залежностей (див Рис. 3.4), тобто в складі потенційного ключа відсутня менша підмножина атрибутів від якої можна також вивести дану функціональну залежність.

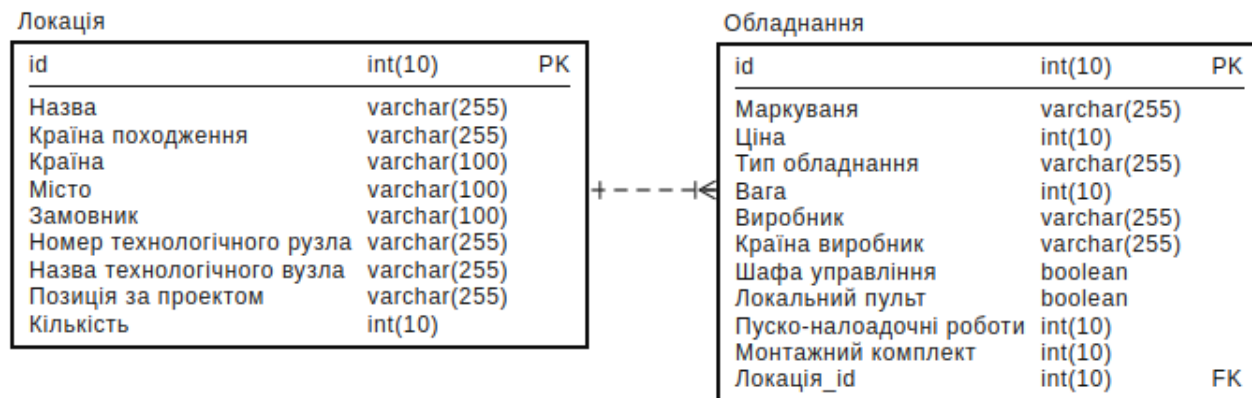


Рисунок 3.4 – ER-діаграма за нотацією Баркера (2НФ)

При зведенні до третьої нормальної форми була проведена перевірка на відсутність транзитивних функціональних залежностей. І доданий унікальний ідентифікатор id що виступає первинним ключем.

Окрім того була проведена додаткова оптимізація структури та винесені деякі групи даних в окремі таблиці. Виконано реструктуризацію таблиці “Локація” з її розділення на “Об’єкт”, “Технологічний вузол”, “Позиція”. Згідно з даною структурою у кожного “Об’єкта” може безліч “Технологічних вузлів”, в той час як кожен вузол може бути асоційований лише з одним “Об’єктом”. Аналогічний взаємозв’язок між “Технологічним вузлом” та “Позицією”. Кожна позиція відноситься лише до одного технологічного вузла у той час як технологічний вузол може мати багато позицій.

На даному етапі проектування БД доцільно за основу взяти загальну діаграму класів з п. 3.2. Було розроблена ER-діаграма за нотацією Баркера (див. Рис 3.5).

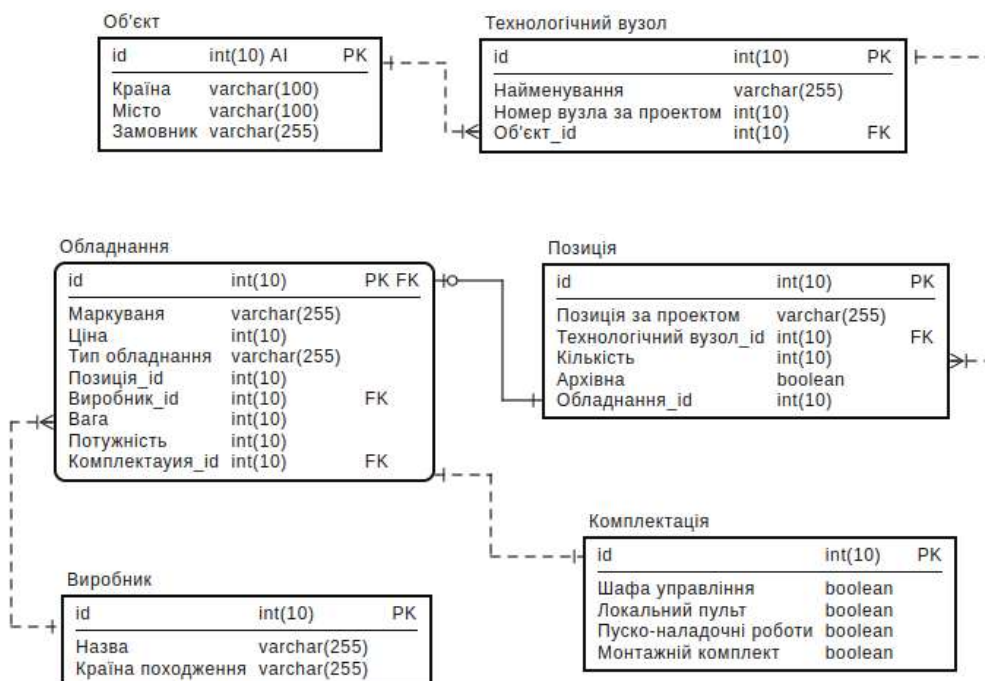


Рисунок 3.5 – ER-діаграма за нотацією Баркера (ЗНФ)

До того ж була виконана реструктуризація таблиці “Обладнання” з винесенням виробника і країни походження в окрему таблицю. До іншої таблиці з назвою комплектація увійшли параметри, що прямого відношення до обладнання не мають а лише відображають послуги та додаткові аксесуари такі як: локальні пульти керування, шафа управління та інше.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Загальні відомості

В сучасному світі активний розвиток та впровадження технологій, пов'язаних із контейнеризацією, стає необхідністю для забезпечення ефективності та гнучкості розробки програмного забезпечення. Додаток, розроблений на базі Django, PostgreSQL та Celery в середовищі Docker, відзначається використанням передових інструментів для оптимізації та полегшення розробки.

Середа розробки PyCharm була обрана для своєї високої продуктивності та розширених можливостей, що сприяють зручному та ефективному програмуванню. Мова програмування Python була обрана за свою універсальність та платформенну незалежність, що дозволяє легко розгортати додаток на різних пристроях та операційних системах.

Для створення візуального інтерфейсу використовується Django templates, який забезпечує високу швидкість розробки та можливість створення багатofункціональних веб-додатків [1, 2, 4]. Бізнес-логіка та взаємодія з клієнтською частиною реалізована за допомогою Django-контролерів у Django-проекті [3, 2].

При взаємодії з інтерфейсом django template генерує відповідні запити, які обробляються backend. Бізнес-логіка включає в себе обробку запитів, валідацію даних, взаємодію з базою даних та інші бізнес-операції [].

Збереження та отримання даних з бази даних здійснюється за допомогою Django ORM як ORM-фреймворку. Django ORM дозволяє взаємодіяти з базою даних PostgreSQL та виконувати операції CRUD над об'єктами сутностей.

Загальна взаємодія компонентів системи, заснована на використанні Docker, Django, PostgreSQL, Celery, спрямована на забезпечення ефективності, надійності та безпеки усіх аспектів розробленого додатку.

4.2 Виклик і завантаження

Для встановлення проекту в середовищі Linux необхідно виконати наступні кроки:

- Встановіть Git на комп'ютер, якщо він ще не встановлений. Відкрийте термінал та введіть наступну команду:
sudo apt-get install git
- Завантажте код застосунку з репозиторію GitHub. Відкрийте термінал та введіть наступну команду:
git clone https://github.com/KostiaMakh/ProjectsHandler.git
- Перейдіть до папки з кодом застосунку. Відкрийте термінал та введіть наступну команду:
cd <folder_name>
- Помістіть у корньову директорію файл .env (файл що зберігає налаштування проекту й ключі доступу) та папку devtools (зберігаються файлові ключі):

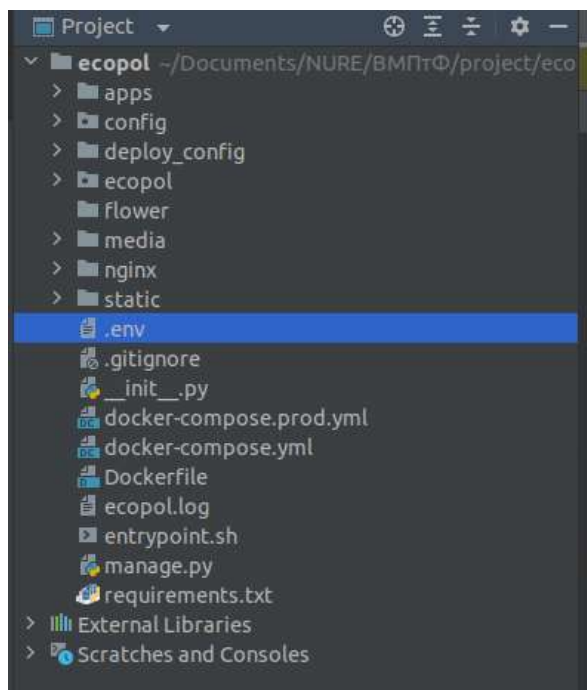


Рисунок 4. 1. - Розміщення файла .env

- Відкрийте термінал:
- Запустіть Docker-compose виконавши наступну команду (це може зайняти якийсь час):
docker-compose up —build
- У разі успішної збірки і запуску контейнерів Docker (див. Рис 4.2) має з'явитися наступне повідомлення:

```

kostiamakh@kostiamakh-MS-7A38: ~/Documents/NURE/BM...
Flower_1 | Did you mean one of these? deploy_  ecopol  flower  media
Flower_1 | worker  config
web_1 | Operations to perform:
web_1 | Apply all migrations: account, admin, auth, contenttypes, project
s, sessions
web_1 | Running migrations:
web_1 | No migrations to apply.
web_1 | Your models in app(s): 'projects' have changes that are not yet r
eflected in a migration, and so won't be applied.
web_1 | Run 'manage.py makemigrations' to make new migrations, and then r
e-run 'manage.py migrate' to apply them.
ecopol_flower_1 exited with code 2
web_1 | Watching for file changes with StatReloader
web_1 | Performing system checks...
web_1 |
web_1 | System check identified no issues (0 silenced).
web_1 | December 20, 2023 - 13:46:43
web_1 | Django version 4.1.7, using settings 'ecopol.settings'
web_1 | Starting development server at http://0.0.0.0:8000/
web_1 | Quit the server with CONTROL-C.
celery_1 | [2023-12-20 13:46:43,913: INFO/MainProcess] mingle: all alone
celery_1 | [2023-12-20 13:46:43,919: INFO/MainProcess] celery@a0d884c17915 rea
dy.rd

```

Рисунок 4. 2. - Успішно піднятий docker-compose

- Відкрийте браузер та перейдіть за адресою <http://0.0.0.0:8000/> для перевірки роботи застосунку.

Це все, що потрібно зробити, щоб завантажити застосунок з GitHub та запустити його у Docker.

Для початку використання додатку необхідно створити супер користувача. Для того щоб це зробити необхідно виконати наступні кроки:

- у терміналі (нова вкладка) запустити команду `docker ps`. Ця команда поверне список усіх запущених контейнерів docker;
- обираємо id контейнера з ім'ям web;
- відкриваємо контейнер командою `docker exec -t container_ID bash`;
- після того як система зайде у середовище контейнера необхідно викликати наступну команду `python3 manage.py createsuperuser`;
- ввести логін (адресу електронної скриньки);
- ввести пароль;
- підтвердити пароль;

- super user створено;

Після виконання пунктів вище можна переходити до користування застосунком.

4.3 Опис програмної реалізації

Програмна реалізація включає в себе розробку клієнтської з використання HTML, CSS, JavaScript, та python на серверній частині [8, 9]. На серверній частині використаний фреймворк Django [10]. В якості адаптеру до бази даних був використано Django ORM [11]. Приклад програмного коду надано в Додатку А.

Розглянемо головні елементи інтерфейсу додатку.

Форма авторизації (див. Рис 4.3) до застосунку. Система вимагає обов'язкової авторизації користувачів оскільки це корпоративний додаток для внутрішнього користування.

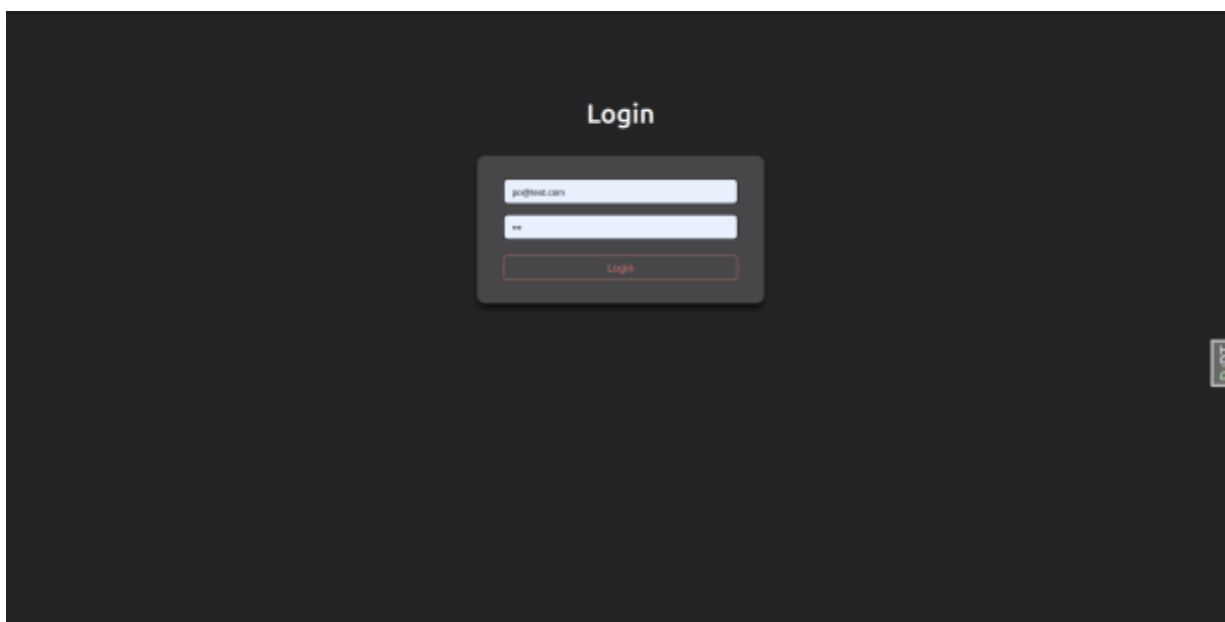


Рисунок 4.3 — Форма авторизації

Головна сторінка застосунку (див. Рис 4.4). На цій сторінці наведено перелік усіх функціональних блоків додатку: проектів, архівів, налаштувань, тощо.

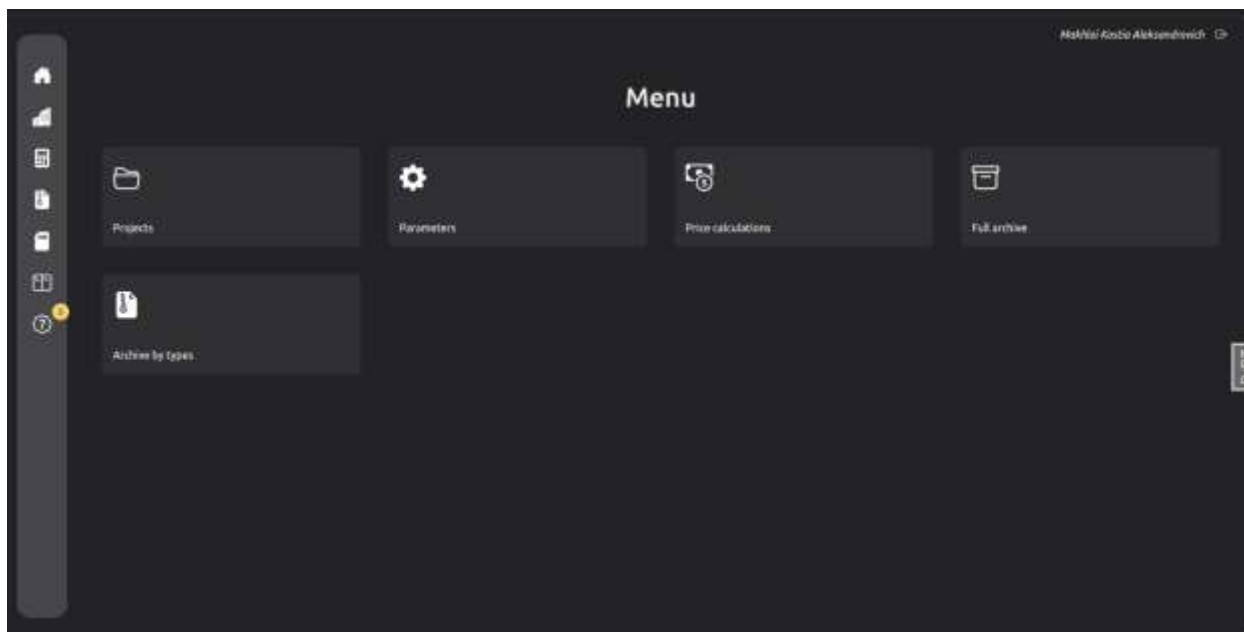


Рисунок 4.4 — Головна сторінка

Користувач має можливість переглядати перелік та деталі проектів (див. Рис 4.5), а якщо у нього є відповідні права то може створювати нові або редагувати існуючі.

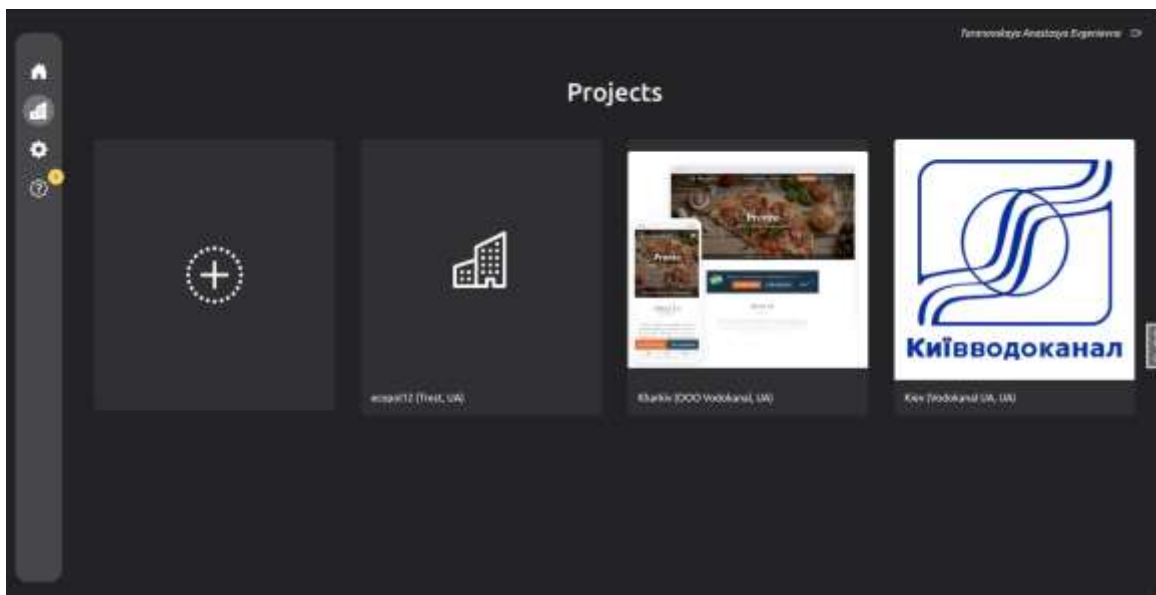
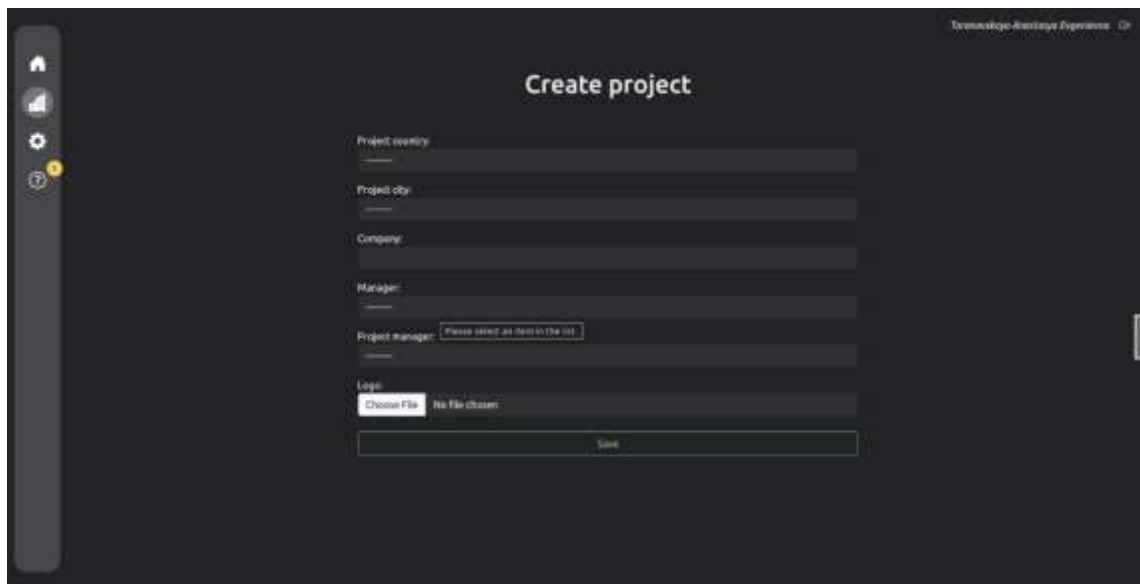


Рисунок 4.5 — Головна сторінка

Форма створення нового проекту наведена на рисунку 4.6.

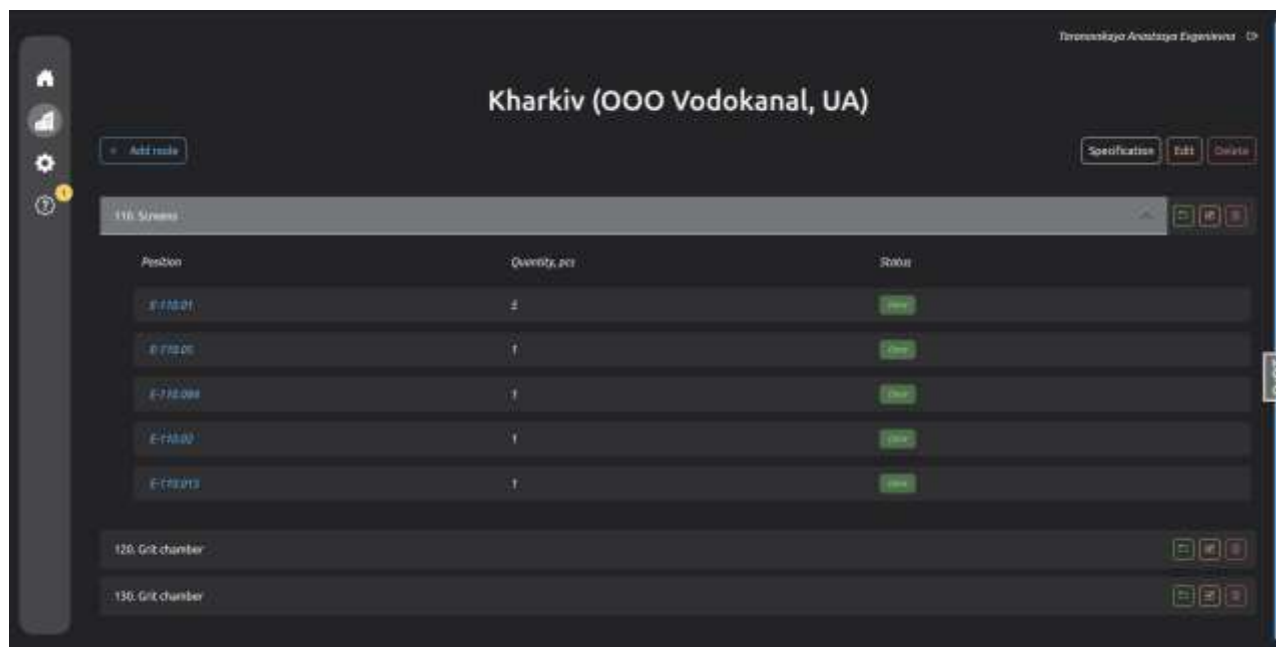


The screenshot shows a dark-themed 'Create project' form. The form contains the following fields and elements:

- Project name:** A text input field.
- Project obj:** A text input field.
- Company:** A text input field.
- Manager:** A text input field.
- Project manager:** A dropdown menu with the placeholder text 'Please select an item in the list'.
- Logo:** A file selection area with a 'Choose File' button and the text 'No file chosen'.
- Save:** A large button at the bottom of the form.

Рисунок 4.6 — Форма створення та редагування проекту

На Рис 4.7 представлена сторінка проекту. Тут відображаються перелік технологічних вузлів даного проекту а також перелік позицій. Данні згруповано по технологічним вузлам, для зручності користування та сприйняття.



The screenshot shows the project details page for 'Kharkiv (OOO Vodokanal, UA)'. The page includes a sidebar with navigation icons and a main content area with the following elements:

- Project Name:** Kharkiv (OOO Vodokanal, UA)
- Buttons:** 'Add node', 'Specification', 'Edit', and 'Delete'.
- Table:** A table with columns 'Position', 'Quantity', and 'Status'. The table is grouped by technology nodes.

Position	Quantity	Status
E-11801	2	Done
E-11802	1	Done
E-11803	1	Done
E-11804	1	Done
E-11805	1	Done
120. Glt chamber		
130. Glt chamber		

Рисунок 4.7 — Деталі проекту

До проекту можна додати відомості про обладнання (форма додавання обладнання див. Рис 4.8), а також переглянути деталі по підбраному обладнанню (див. Рис 4.9), а також відреагувати чи видалити обладнання.

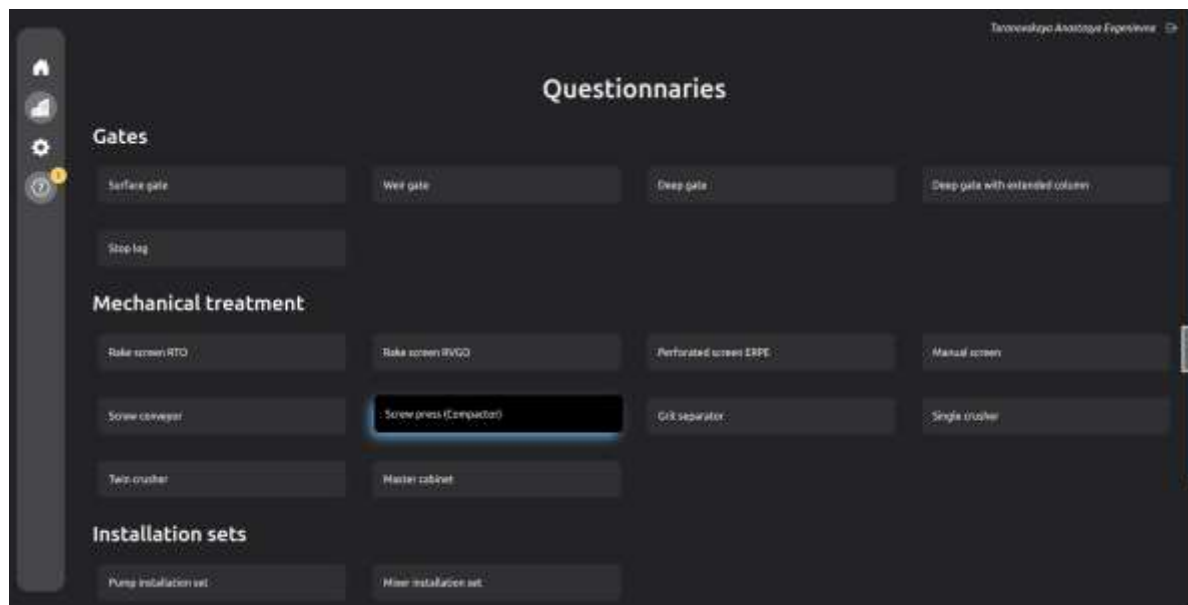


Рисунок 4.8 — Меню вибору типу обладнання

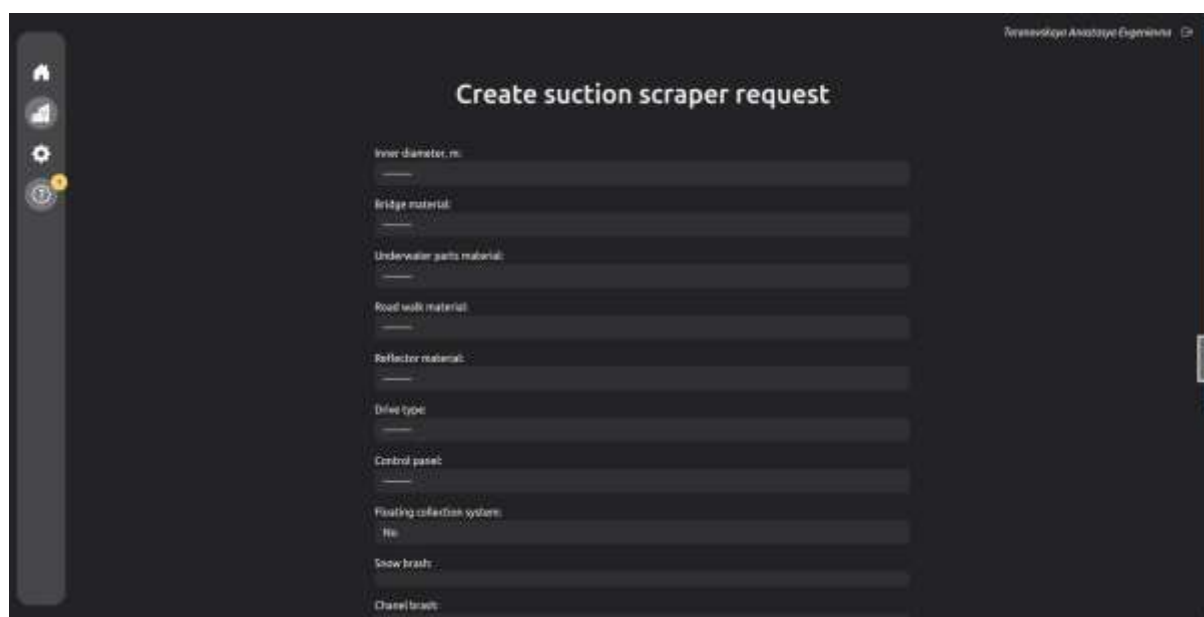


Рисунок 4.9 — Форма створення запиту на підбір обладнання

Користувач також має можливість продивлятися, а якщо є відповідні права то і редагувати запити (див. Рис 4.10). При цьому система веде архів усіх змін, що вносяться. Таким чином при зміні якогось параметру з запиту у чаті (справа) системою буде автоматно додано повідомлення про те що на що було замінено коли і ким.

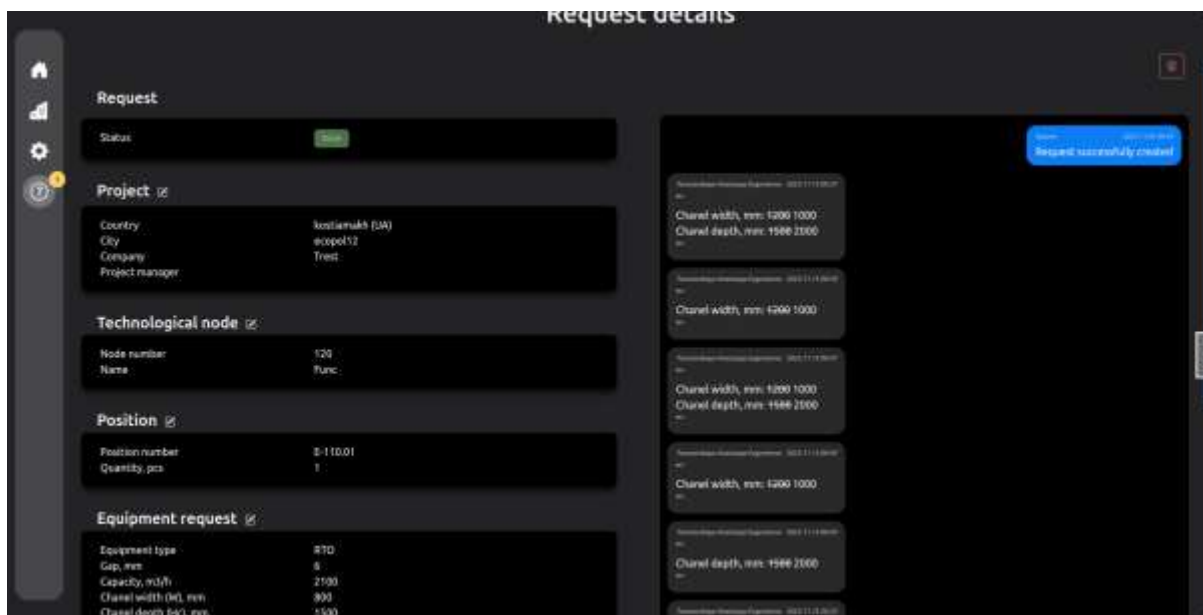


Рисунок 4.10 — Деталі обладнання

Користувач може переглядати лист запитів (див. Рис 4.11) на підбір обладнання, а якщо є відповідні допуски то може виконувати розрахунок, редагувати, чи видаляти запит.

Status	Node	Position	Equipment type	Author	Last updates
new	120. koostlenak6	E-110.084	Master control cabinet	Технічний Аналіз Експертів	04.07.2024 05:51:58
new	120. Func12	E-110.015	Surface gate	Технічний Аналіз Експертів	29.11.2022 06:04:39
new	120. Func12	E-110.01	Control press	Технічний Аналіз Експертів	17.10.2022 06:30:11

Рисунок 4.11 — Запити на підбір обладнання

Також користувач може переглянути специфікацію по проекту (див. Рис 4.12). Дане подання відображає технічні характеристики обладнання по проекту.

Position	Qty, pcs	Mark	Material	Power, kW	IP	Weight, kg	Description
120. Func							
E-110.01	1	PTO 800.1500.1200.6	AlSi 316	0.37	IP 66	650	Решетка тонкой очистки гравельного типа. Максимальная производительность - 2000 м³/ч; прозор - 6 мм; диаметр канала - 800 мм; глубина канала - 1500 мм; высота выгрузки отбросов - 1200 мм; вес решетки в сборе - 650 кг; привод - 0,37 кВт; IP 66; 380 В; 50 Гц; материал исполнения - AlSi 316; в комплекте с ЦУ и ВПК.
E-110.03	1	-	-	-	-	-	-
E-110.02	1	-	-	-	-	-	-
E-110.09	1	PTO 600.1200.650.10	AlSi 201	0.37	IP 55	450	Решетка тонкой очистки гравельного типа. Максимальная производительность - 2000 м³/ч; прозор - 10 мм; диаметр канала - 600 мм; глубина канала - 1200 мм; высота выгрузки отбросов - 850 мм; вес решетки в сборе - 450 кг; привод - 0,37 кВт; IP 55; 380 В; 50 Гц; материал исполнения - AlSi 201; в комплекте с ЦУ и ВПК.
E-110.54	1	PTO 500.800.850.8	AlSi 201	0.37	IP 55	340	Решетка тонкой очистки гравельного типа. Максимальная производительность - 2000 м³/ч; прозор - 8 мм; диаметр канала - 500 мм; глубина канала - 800 мм; высота выгрузки отбросов - 850 мм; вес решетки в сборе - 340 кг; привод - 0,37 кВт; IP 55; 380 В; 50 Гц; материал исполнения - AlSi 201; в комплекте с ЦУ и ВПК.
E-110.084	1	ЗГВП 1-150-1000	AlSi 304	1.5	IP 66	230	Пресс винтовой промышленный. Производительность по отбросам - 1 м³/ч; длина загрузкиного вала - 1000 мм; высота выгрузки отбросов - 1500 мм; вес пресса в сборе - 230 кг; привода - 0,83 кВт; 3-5 бар; привод - 1,5 кВт; IP 66; 380 В; 50 Гц; материал (корпус, вал, труба выгрузки) - AlSi 304; без ЦУ.
E-110.0145	1	-	-	-	-	-	-
E-110.10	1	-	-	-	-	-	-
E-110.085	1	ЗД02	AlSi 304	8.0	IP 55	630	Дробилка отбросов. Максимальная производительность - 1 м³/ч; вес дробилки в

Рисунок 4.12 — Специфікація проекту

Під час формування специфікації виконується наступний SQL запит:

```

SELECT «questionnaire_customsoftdeletemodel».»id»,
      «questionnaire_customsoftdeletemodel».»is_deleted»,
      «questionnaire_customsoftdeletemodel».»deleted_at»,
      «questionnaire_technologicalnode».»customsoftdeletemodel_ptr_id»,
      «questionnaire_technologicalnode».»created_at»,
      «questionnaire_technologicalnode».»updated_at»,
      «questionnaire_technologicalnode».»node_project_id»,
      «questionnaire_technologicalnode».»node_number»,
      «questionnaire_technologicalnode».»name»,
      «questionnaire_technologicalnode».»is_archive»
FROM «questionnaire_technologicalnode»
INNER JOIN «questionnaire_customsoftdeletemodel»
  ON      («questionnaire_technologicalnode».»customsoftdeletemodel_ptr_id»
«questionnaire_customsoftdeletemodel».»id»)
WHERE    (NOT      «questionnaire_customsoftdeletemodel».»is_deleted»      AND
«questionnaire_technologicalnode».»node_project_id» = '98')
ORDER BY «questionnaire_technologicalnode».»node_number» ASC

```

Окрім того користувач може переглянути загальний архів обладнання за проектом (див. Рис 4.13). На цій сторінці доступні, для зручності користування, є наступні фільтри та сортування (Рис 4.14). Також тут передбачена можливість експорту звіту у форматі csv.

ID	Country	City	organization	Mode	Position	Mark	Total price (no TAX)	Equipment price (no TAX)	CP price (no TAX)	Material	Drive IP	Manager	Author
178	Ukraine	Kharkiv	«ІНТЕЛ»	«ІНТЕЛ»	PTD	1500.2500.1200.4	753 700	465 010	48 690	АІСІ 304	ІР 55	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
177	Ukraine	Kharkiv	«ІНТЕЛ»	110. Eco	PTD	500.500.850.6	450 030	340 240	89 790	АІСІ 304	ІР 66	Logosov Petr Aleksandrovich	Mikhail Stepan
176	Ukraine	ecopoll2	Vedkarski	«ІНТЕЛ»	PRFD	1500.2000.050.20	1 318 060	1 318 060	0	АІСІ 321	ІР 55	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
175	Ukraine	Kharkiv	Vedkarski	primo-stalen 83	E-110.23	PTD 1500.1800.850.8	726 350	437 660	88 690	АІСІ 304	ІР 55	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
174	Ukraine	ecopoll2	«ІНТЕЛ»	110. Eco	E-110.23	PRFD 1391.1800.1500.1200.40	414 739 066	414 739 003	0	АІСІ 304	ІР 66	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
173	Ukraine	Kharkiv	ООО Vedkarski	120. Oil-chamber	E-120.02	PTD 1200.1500.1200.8	676 055	587 365	88 690	АІСІ 304	ІР 55	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
172	Ukraine	Kharkiv	«ІНТЕЛ»	«ІНТЕЛ»	E-110.23	PTD 1200.1500.850.8	609 080	379 290	89 790	АІСІ 304	ІР 66	Logosov Petr Aleksandrovich	Mikhail Kostin Aleksandrovich
171	Ukraine	Kharkiv	Vedkarski	110. Eco	PTD	1200.1200.850.8	417 260	417 260	0	АІСІ 201	ІР 55	Logosov Petr Aleksandrovich	Mikhail Stepan
170	Ukraine	ecopoll2	Vedkarski	«ІНТЕЛ»	E-110.23	PTD 1500.3000.1200.4	602 260	513 570	88 690	АІСІ 201	ІР 55	Logosov Petr Aleksandrovich	Mikhail Stepan
169	Ukraine	Kharkiv	«ІНТЕЛ»	110. Eco	E-110.23	PTD 1000.1300.1200.4	515 945	426 155	89 790	АІСІ 201	ІР 66	Logosov Petr Aleksandrovich	Mikhail Stepan
168	Ukraine	Kharkiv	«ІНТЕЛ»	110. Eco	E-110.23	PTD 1000.1300.1200.4	515 945	426 155	89 790	АІСІ 201	ІР 66	Logosov Petr Aleksandrovich	Mikhail Stepan

Рисунок 4.13 — Архів обладнання

The screenshot displays a web application interface for an equipment archive. The main area shows a table with the following columns: ID, Country, City, Organization, Role, Position, Mark, Total price (Ukr. UAH), Equipment price (Ukr. UAH), CP price (Ukr. UAH), Material, IP, and Storage. The table contains several rows of data. On the right side, there is a 'Filters' sidebar with a search icon and a close button. The filters are organized into sections: City (with a search field containing 'Kyiv'), Country (with a search field containing 'Ukraine'), Author (with a search field containing 'Mikhail Kostin Aleksandrovich'), Manager (with a search field containing 'Logunov Deyr Aleksandrovich'), Mark (with a search field containing '3C7-2-90'), and Material (with a search field containing 'AISI 304').

Рисунок 4.14 — Фільтри та сортування архіву обладнання

Для фільтрації та сортування в проєкті була застосована бібліотека `django-filters`. Код інтеграції бібліотеки з проєктом наведено у додатку А.

1. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Перевірка коректної роботи програмного забезпечення є важливою частиною розробки, оскільки вона допомагає виявити дефекти та помилки на ранніх етапах, що забезпечує високу якість кінцевого продукту. А також уникнути ситуацій коли при зміні коду в одній частині програми проблеми виявляються зовсім в іншій. Це також сприяє кращому розумінню вимог до системи ще на стадії розробки. В процесі розробки і тестування важливо використовувати інструменти для відстеження та управління помилками, дефектами, завданнями та іншими аспектами проекту.

Серед популярних інструментів є JIRA, Bugzilla, Trello, але вибір залишається за кожним. У цьому випадку використовуються xlsx файли для зберігання інформації для перевірок. Оскільки оновлення додатку буде рідкісним, цього достатньо для аналізу тестування та відстеження змін. До того ж ці програмні продукти мають сенс коли на проекті працює велика кількість розробників та тестувальників.

Під час розробки та тестування були виділені та застосовані наступні види тестування:

- функціональне тестування: Перевірка основних функцій додатку, таких як реєстрація, авторизація, створення запиту на обладнання, зміна статусів, розрахунок обладнання, ведення листування;
- тестування користувацького інтерфейсу (UI Testing): Перевірка правильності роботи та зручності користування інтерфейсом, включаючи навігацію, розташування елементів та відповідність дизайну;

- тестування зручності використання (Usability Testing): Оцінка інтуїтивної зрозумілості та комфорту використання додатку, враховуючи логіку взаємодії та рівень задоволеності користувачів;
- тестування інсталяції (Installation Testing): Випробування процесу розгортання проекту у різних середовища.

Для проведення функціонального тестування та тестування інсталяції було створено чек-лист і написано тестові кейси.

Зміст розробленого чек-листа представлено в таблиці 5.1.

Таблиця 5.1 — чек-лист (виконана самостійно)

№	Назва тесту	Результат (+/-)
1	Розгортання проекту	+
2	Перший запуск	+
3	Створення бази даних	+
4	Вхід до панелі адміністратора	+
5	Створення / редагування / видалення користувачів	+
6	Вхід у панель користувача як проектувальник	+
7	Вхід у панель користувача як спеціаліст з обладнання	+
8	Відображення елементів інтерфейсу на всіх екранах	+
9	Працездатність кнопок та інтерактивних елементів	+
10	Коректна навігація між вкладками	+
11	Зручність та інтуїтивність інтерфейсу	+
12	Відповідність дизайну інтерфейсу вимогам	+
13	Зрозумілість логіки взаємодії з застосунком	+
14	Легко доступність всіх функцій додатку	+
15	Швидкодія додатку без затримок	+

В додатках Б - В викладено приклади розроблених тест-кейсів.

Тестування зручності використання (Usability Testing) та інтерфейсу користувача (UI Testing) показали, що додаток відповідає вимогам зручності як у десктопному так і ц мобільному варіанті відображення. Зручне розташування блоків інтерфейсу, логічна навігація та відповідність дизайну забезпечують комфортне використання додатку. Інтуїтивна зрозумілість та високий рівнів уподобань користувачів підтверджують успішність реалізації функціоналу та дизайну додатку.

ВИСНОВКИ

Базуючись на дослідженнях, що були проведені в рамках кваліфікаційної роботи, можна зробити висновок, що розробка даного веб-застосунку значно оптимізує процеси менеджменту та розрахунків обладнання при веденні складних комплексних проектів. Використання сучасних технологій та програмних рішень, таких як Django і PostgreSQL, дозволяє забезпечити високий рівень автоматизації, покращує точність та доступність даних про обладнання, що використовується у проектах, а також дозволяє оперативна контролювати поточний стан проекту.

Функціонал, що надає розроблений програмний застосунок дозволяє зменшити витрати часу на ручну обробку інформації та забезпечує кращу координацію та комунікацію між учасниками проекту. Такий підхід дозволяє оперативно відстежувати життєвий цикл обладнання, від запиту до використання у конкретних проектах, та відповідно реагувати на зміни у проектних завданнях. Окрім того, допоміжні сервіси такі як ведення архіву, збереження листувань, статусів, фільтрацій даних та автоматизація розсилки листів, вносять вагомий вклад у підвищення ефективності загальних процесів управління проектами.

Завдяки зручному інтерфейсу та масштабованому функціоналу, система має значний потенціал до масштабування і стане незамінним інструментом для підприємств, які прагнуть інтегрувати цифрові рішення в свої процеси. Це не тільки сприяє зростанню продуктивності, але й забезпечує більшу адаптивність компаній до швидко змінних умов ринку, а також підвищує їхню конкурентоспроможність на рівні з найсучаснішими світовими стандартами управління проектами.

Таким чином, кваліфікаційна робота демонструє, як технологічні інновації можуть трансформувати традиційні підходи до управління у сфері

проектування очисних споруд, роблячи їх більш ефективними, масштабованими та орієнтованими на майбутнє.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. The Django admin site. Django documentation. URL: <https://docs.djangoproject.com/en/3.0/ref/contrib/admin/> (дата звернення 30.05.2024).
2. Security in Django. Django documentation. URL: <https://docs.djangoproject.com/en/3.0/topics/security/> (дата звернення 04.06.2024).
3. Celery - Distributed Task Queue. Celery documentation. URL: <https://docs.celeryq.dev/en/stable/> (дата звернення 09.06.2024).
4. Writing views. URL: <https://docs.djangoproject.com/en/4.2/topics/http/views/> (дата звернення 18.05.2024).
5. UML Diagrams in Software Engineering Research URL:: <https://www.mdpi.com/2504-3900/74/1/13>. (дата звернення: 16.05.2024).
6. Діаграми UML URL: <https://planerka.info/item/diagrammy-kommunikacij-uml/>. (дата звернення: 08.06.2024).
7. Методичні вказівки до курсового проектування з дисципліни «Бази даних» для студентів усіх форм навчання спеціальності 121 – «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмна інженерія») / Упоряд. О.О. Мазурова, М.С.Широкопетлева, Ю.Ю.Черепанова, – Харків: ХНУРЕ, 2022. – 61 с.
8. Jalolov T. S. ADVANTAGES OF DJANGO FEMWORKER //International Multidisciplinary Journal for Research & Development. – 2023. – Т. 10. – №. 12.
9. Nagaraj P. et al. Automated E-Commerce Price Comparison Website using PHP, XAMPP, MongoDB, Django, and Web Scrapping //2023 International Conference on Computer Communication and Informatics (ICCCI). – IEEE, 2023. – С. 1-6.

10. Chen X. Design and Application of Web-Based Financing Management Information System for Infrastructure Projects in Higher Vocational Colleges //Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023). – Atlantis Press, 2023. – C. 1535-1539.
11. Tao S., Ding L. Design of Remote maintenance and management system for charging piles based on Django //2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT). – IEEE, 2021. – C. 604-609.
12. Le Q. H., Diaz M. Developing Modern Database Applications with PostgreSQL: Use the highly available and object-relational PostgreSQL to build scalable and reliable apps. – Packt Publishing Ltd, 2021.

ДОДАТОК А

Фрагмент коду програми

models.py

```
class BaseModel(CustomSoftDeleteModel):
```

```
    created_at = models.DateTimeField(auto_now_add=True, blank=True, null=True)
```

```
    updated_at = models.DateTimeField(auto_now=True, blank=True, null=True)
```

```
class Meta:
```

```
    app_label = "questionnaire"
```

```
    abstract = True
```

```
class Project(BaseModel):
```

```
    project_country = models.ForeignKey('Country', on_delete=models.PROTECT,  
verbose_name='Country')
```

```
    project_city = models.ForeignKey('City', on_delete=models.PROTECT, verbose_name='City')
```

```
    company = models.CharField(max_length=256, blank=True, null=True,  
verbose_name='Company')
```

```
    manager = models.ForeignKey(  
        User,  
        on_delete=models.SET_NULL,  
        null=True,  
        related_name='manager_project',  
        verbose_name='Manager')
```

```
    project_manager = models.ForeignKey(  
        User,  
        on_delete=models.SET_NULL,  
        null=True,
```

```

        related_name='projects',
        verbose_name='Project manager'
    )
    logo = models.ImageField(upload_to=get_file_path, blank=True, null=True,
verbose_name='Logo')
    is_archive = models.BooleanField(default=False)

class Meta:
    verbose_name = "Project"
    verbose_name_plural = "Projects"

def __str__(self):
    return f'{self.project_city} ({self.company}, {self.project_country.country_code})'

@property
def project_name(self):
    return f'{self.project_city} ({self.company}, {self.project_country.country_code})'

def delete(self, cascade=None, *args, **kwargs):
    TechnologicalNode.objects.filter(node_project=self.pk).delete()
    super().delete(cascade=None, *args, **kwargs)

class TechnologicalNode(BaseModel):
    node_project = models.ForeignKey('Project', on_delete=models.CASCADE,
related_name='nodes')
    node_number = models.PositiveIntegerField(default=0, verbose_name='Node number')
    name = models.CharField(max_length=256, verbose_name='Node name')
    is_archive = models.BooleanField(default=False)

class Meta:

```

```

unique_together = ('node_project', 'node_number', 'name',)
verbose_name = "Technological node"
verbose_name_plural = "Technological nodes"
ordering = ('node_number',)

```

```

def __str__(self):
    return f'{self.node_number}. {self.name}'

```

```

def delete(self, cascade=None, *args, **kwargs):
    Position.objects.filter(technological_node__pk=self.pk).delete()
    super().delete(cascade=None, *args, **kwargs)

```

```

class Position(BaseModel):
    technological_node = models.ForeignKey(
        'TechnologicalNode',
        on_delete=models.CASCADE,
        blank=True, null=True,
        related_name='positions'
    )
    position_number = models.CharField(max_length=64, verbose_name='Position number')
    quantity = models.PositiveIntegerField(default=1, verbose_name='Quantity, pcs')
    is_archive = models.BooleanField(default=False)

```

```

class Meta:
    unique_together = ('technological_node', 'position_number',)
    verbose_name = "Position"
    verbose_name_plural = "Positions"

```

```

def __str__(self):
    return f'{self.position_number} - {self.quantity} pcs.'

```

```

def last_request(self):
    last_request = PositionRequest.objects.filter(equipment_position=self).first()
    return last_request

```

```

def delete(self, cascade=None, *args, **kwargs):
    PositionRequest.objects.filter(equipment_position__pk=self.pk).delete()
    super().delete(cascade=None, *args, **kwargs)

```

filtes.py

```

def get_archive_marks():
    marks = EquipmentArchive.objects.all().order_by('-mark').values_list('mark', flat=True)
    marks_list = [[val, val] for val in marks]
    marks_list.insert(0, ['', '-----'])
    return marks_list

```

```

class RTOScreenFilter(django_filters.FilterSet):
    author = ModelMultipleChoiceFilter(queryset=User.objects_oto.all())
    manager = ModelMultipleChoiceFilter(queryset=User.objects_managers.all())
    # mark = ChoiceFilter(choices=get_marks())
    material = ChoiceFilter(choices=equipment_general.MATERIALS)
    equipment_details__equipment_type =
ChoiceFilter(choices=equipment_types.EQUIPMENT_TYPES)

```

```

class Meta:
    model = EquipmentArchive

    fields = {
        'mark': ['contains'],

```

```
}

```

```
class FullArchiveFilter(django_filters.FilterSet):
    country = ModelMultipleChoiceFilter(
        queryset=Country.objects.all(),
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    city = ModelMultipleChoiceFilter(
        queryset=City.objects.all(),
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    author = ModelMultipleChoiceFilter(
        queryset=User.objects_oto.all(),
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    manager = ModelMultipleChoiceFilter(
        queryset=User.objects_managers.all(),
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    mark = MultipleChoiceFilter(
        choices=get_archive_marks(),
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    material = MultipleChoiceFilter(
        choices=equipment_general.MATERIALS,
        widget=forms.SelectMultiple(attrs={'class': 'form-control'})
    )
    equipment_details__equipment_type = ChoiceFilter(
        choices=equipment_types.EQUIPMENT_TYPES,
        widget=forms.Select(attrs={'class': 'form-control'}),

```

)

class Meta:

model = EquipmentArchive

fields = [

 'city',

 'country',

 'author',

 'manager',

 'mark',

 'material',

 'equipment_details__equipment_type'

]

ДОДАТОК Б

Тест-кейс №1

Інформація про тест-кейс			
Ідентифікатор тесту:	Тест-кейс №1		
Опис функції:	Створення проекту		
Власник тесту:	Махлай Костянтин Олександрович		
Дата створення:	20.06.2024		
Мета тесту:	Перевірити коректність реалізації створення проекту, додавання обладнання, додавання позицій		
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити веб-застосунок	Користувач має доступ до сайту, який відкритий	Пройдено
2	Має відповідні права	Має права на створення нових проектів	Пройдено
3	Має обліковий запис та хоче авторизуватися	Перенаправляється на сторінку профілю	Пройдено
Створення нового проекту			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити сторінку з проектами	З'являється список існуючих проектів і кнопка "додати проект"	Пройдено
2	Натискання на "додати проект"	З'являється форма для створення проекту	Пройдено
3	Заповнити обов'язкові поля «Місто», «Країна», «Назва»	Кількість символів у полях вказана правильно	Пройдено
4	Створити проект	Після створення проекту система редіректить на сторінку проекту	Пройдено
5	Натискання на "додати технологічний вузол"	З'являється форма для створення проекту	Пройдено
6	Заповнити обов'язкові поля «Номер», «Назва»	Кількість символів у полях вказана правильно	Пройдено
7	Створити технологічний вузол	На сторінці з деталями проекту з'являється технологічний вузол	Пройдено
8	Натискання на "додати технологічний позицію"	З'являється форма для технологічно вузлу	Пройдено
9	Заповнити обов'язкові поля «Номер», «Назва»	Кількість символів у полях вказана правильно	Пройдено

10	Натискання на “додати позицію	На сторінці з деталями проекту з’являється технологічний вузол з статусом “додати запит”	Пройдено
----	-------------------------------	--	----------

Результати тестування

Тестувальник:
Махлай К. О.

Дата прогону тесту:
05.07.2024

Результат тесту (P/F/V):
ПРОЙДЕНО (P)

ДОДАТОК В

Тест-кейс №2

Інформація про тест-кейс			
Ідентифікатор тесту:	Тест-кейс №2		
Опис функції:	Створення запиту на підбір обладнання		
Власник тесту:	Махлай Костянтин Олександрович		
Дата створення:	20.06.2024		
Мета тесту:	Перевірити коректність реалізації створення запиту на підбір обладнання		
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити веб-застосунок	Користувач має доступ до сайту, який відкритий	Пройдено
2	Має відповідні права	Має відповідні права	Пройдено
3	Має обліковий запис та хоче авторизуватися	Перенаправлення на сторінку профілю	Пройдено
4	У системі є створенні проекти	У системі є зареєстровані проекти	Пройдено
Створення запиту на підбір обладнання			
№	Опис випадку	Очікуваний результат	Висновок
	Увійти в систему з правами “проектувальник”	Перехід на головну сторінку	Пройдено
1	Відкрити сторінку з проектами	З’являється список існуючих проектів та обрати один	Пройдено
2	Відкрити сторінку з деталями проекту	З’являється список технологічних вузлів та перелік позицій	Пройдено
	Натиснути “Створити запит”	Система переводить на сторінку з переліком доступних типів обладнання	Пройдено
	Обрати тип обладнання	Система переводить на сторінку з формою	Пройдено
3	Заповнити обов’язкові поля	Кількість символів у полях вказана правильно, типи даних коректні	Пройдено
	Створити запит	Система створює запит і переводить на сторінку з деталями запису. Статус запиту “new”	Пройдено
4	Вийти з системи	Переводить на сторінку логіну	Пройдено

	Увійти в систему з правами “спеціаліст з обладнання”	Система переводить на головну сторінку	
5	Перейти на сторінку “запити”	Відображається перелік проектів і позицій по яким був запит.	Пройдено
6	Натиснути на запит	Система переводить на сторінку з деталями запиту	Пройдено
7	Натиснути кнопку “Розрахувати”	Система виконує розрахунки, додаються технічні характеристики, вартість, статус змінюється на “done”	Пройдено
8	Натиснути кнопку “завантажити пропозицію”	Завантажується doc документ з техніко-комерційною пропозицією.	Пройдено

Результати тестування

Тестувальник:
Махлай К. О.

Дата прогону тесту:
05.07.2024

Результат тесту (P/F/V):
ПРОЙДЕНО (P)

ДОДАТОК Г

Результат перевірки роботи на плагіат



 Дата звіту 7/16/2024
 Дата редагування ---


 Звіт не був оцінений.

метадані

Заголовок
ПЗПIn-22-2_Махлай_К_О

Автор
Махлай Костянтин Олександрович

Науковий керівник / Експерт
Вадим Юрійович Нечволод

підрозділ
Харківський національний університет радіоелектроніки

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просямо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		38

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Заверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

5724

Кількість слів

45594

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	2024_Б_ПІ_ПЗПIn-22-2_Бабій_В_Г_скорочений 7/12/2024 Kharkiv National University of Radio Electronics (Харківський національний університет радіоелектроніки)	91	1.59 %
2	http://openarchive.nure.ua/bitstream/document/11192/1/2019_M_ST_Ilyunina_M.pdf	60	1.05 %
3	2024_Б_ПІ_ПЗПIn_22_2_Гавриленко_В_В 7/9/2024 Kharkiv National University of Radio Electronics (Харківський національний університет радіоелектроніки)	27	0.47 %

ДОДАТОК Д

Слайди презентації



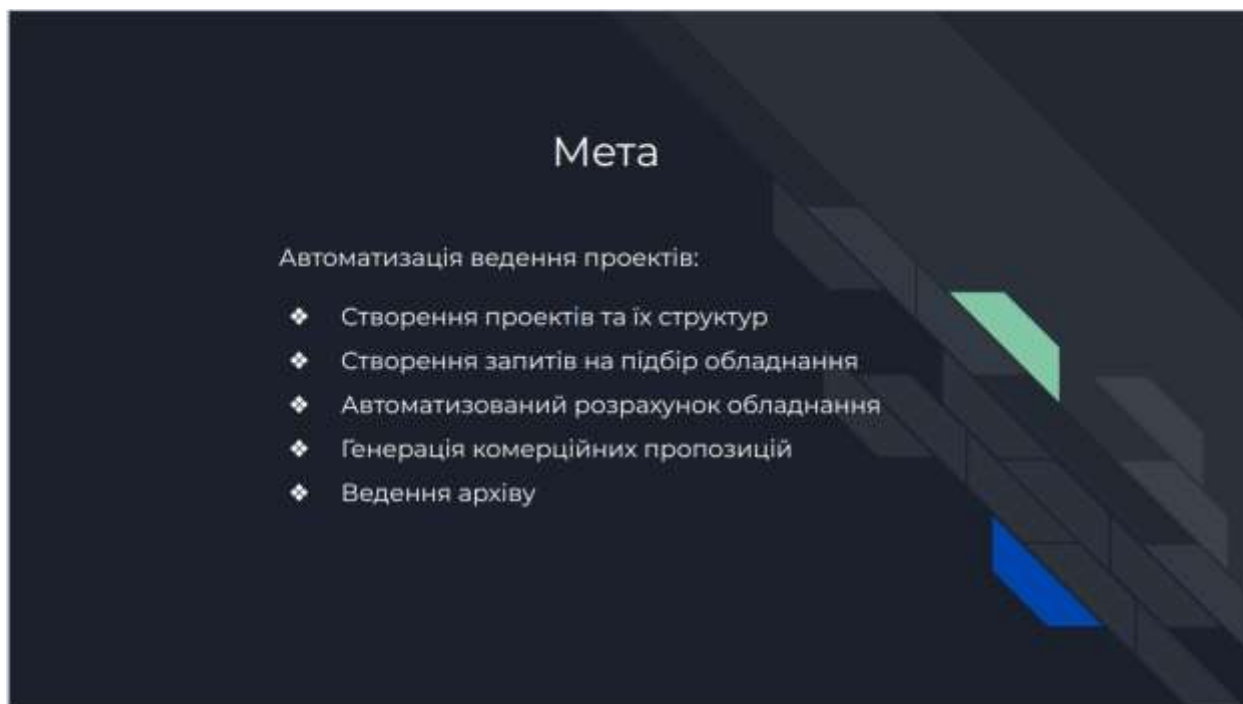
Міністерство освіти і науки України Харківський
національний університет радіоелектроніки

Кваліфікаційна робота бакалавра

Програмна система обліку комерційних
пропозицій виробничого підприємства

Виконав: студент гр. ПЗПп-22-2
Махлай К. О.

Науковий керівник: доц. Лановий О.Ф.



Мета

Автоматизація ведення проектів:

- ❖ Створення проектів та їх структур
- ❖ Створення запитів на підбір обладнання
- ❖ Автоматизований розрахунок обладнання
- ❖ Генерація комерційних пропозицій
- ❖ Ведення архіву

Аналіз ринку

Microsoft Excel

Переваги:

- Зручний інтерфейс
- Розширені функції та формули
- Візуалізація даних
- Широка сумісність
- Вбудований редактор коду (VBA)

Недоліки:

- Обмежена робота з великими обсягами даних
- Проблеми з колективною роботою
- Відсутність централізованого управління даними
- Не підходить для складної аналітики
- Складнощі в інтеграції з іншими системами



Аналіз ринку

Google Sheets

Переваги:

- Хмарна система та спільне використання
- Зручний інтерфейс
- Розширені функції та формули
- Можливості візуалізації даних
- Інтеграція з іншими засобами Google

Недоліки:

- Менша гнучкість налаштувань
- Брак повного захисту даних
- Обмежені можливості роботи з великими обсягами даних



Аналіз ринку

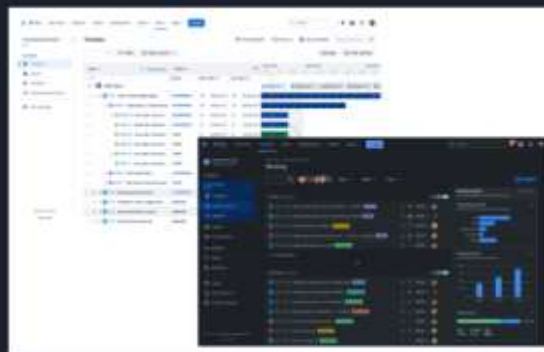
Jira

Переваги:

- Гнучкість конфігурації
- Інтеграція з іншими інструментами
- Підтримка агільних методологій
- Контроль проходження завдань та процесів

Недоліки:

- Складний інтерфейс
- Висока вартість
- Тривалий час налаштування



Аналіз ринку

Autodesk Revit

Переваги:

- Інтегроване проектування
- Висока точність моделювання
- Автоматизація документації
- Ефективне управління ресурсами

Недоліки:

- Складність освоєння
- Вимоги до апаратного забезпечення
- Обмежена сумісність



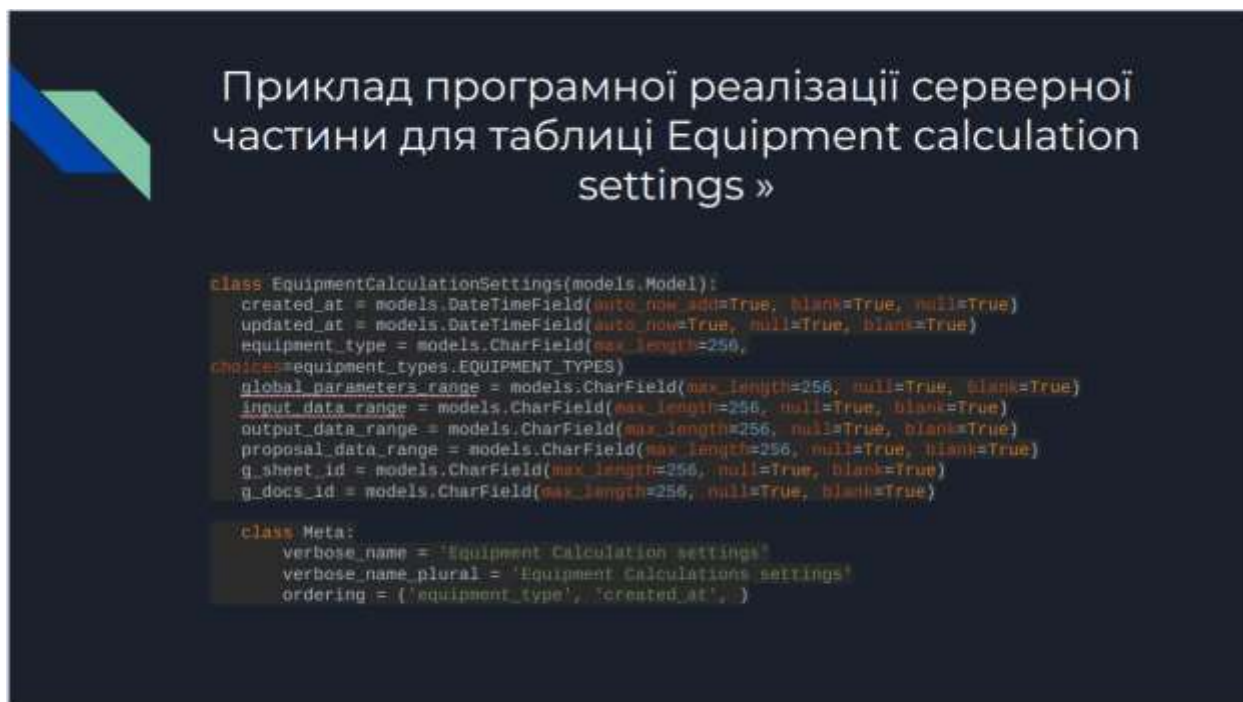
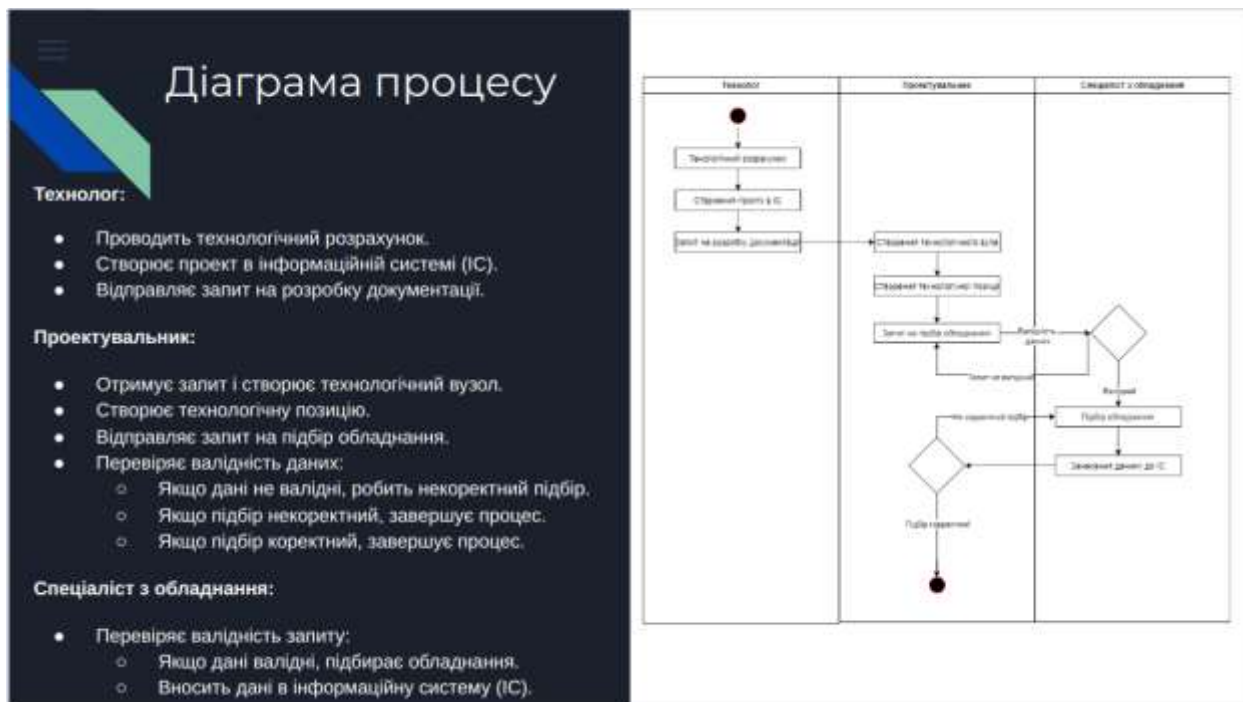
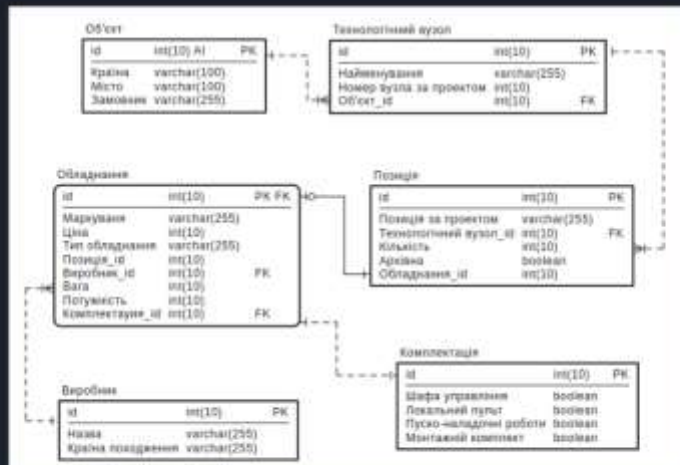


Схема БД



Интерфейс користувача

Technological Assets Register

New request

Статус	Назва	Решення	Розмірний тип	Акція	Історія змін
Відкрито	100 Автосканер	В 110,204	Модель заводу завод	Technological Assets Register	20.01.2024 08:59:59
Відкрито	102 Ручка	В 110,211	ЗаРучка ручка	Technological Assets Register	20.01.2024 08:59:59
Відкрито	103 Ручка	В 110,211	ЗаРучка ручка	Technological Assets Register	27.06.2023 06:08:30

Кнопки: Відкрито, Замовлено, Вибрано, Вибрано

Меню: Головна, Нові запити, Історія змін, Вибрано, Вибрано

