

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ ВЕДЕННЯ ЕЛЕКТРОННОЇ ЗВІТНОСТІ
СПІВРОБІТНИКІВ ПІДПРИЄМСТВА

RESEARCH AND SOFTWARE IMPLEMENTATION OF
E-REPORTING TOOLS FOR COMPANY EMPLOYEES

Виконав: студент 2 курсу, групи КНТ-122м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

АБЛЯКІМОВ С.Н.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

АБЛЯКІМОВА Сеттара Нарімановича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів ведення електронної звітності співробітників підприємства. Research and Software Implementation of E-Reporting Tools for Company Employees

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “14” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “16” жовтня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Сеттар АБЛЯКІМОВ
(Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Михайло КОЦУР
(Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
89 с., 44 рис., 3 табл., 3 дод., 19 джерел.

PYTHON, DJANGO, ЕЛЕКТРОННА ЗВІТНІСТЬ, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ФРЕЙМВОРК.

Об'єкт дослідження – процес розробки програмного забезпечення для ведення електронної звітності.

Предмет дослідження – програмні засоби підтримки процесу ведення електронної звітності.

Метою роботи є дослідження та розробка програмного забезпечення для ведення електронної звітності співробітників підприємства.

Матеріали, методи та технічні засоби: мова програмування Python, фреймворк Django.

Результати. Розроблено програмне забезпечення підтримки процесу ведення електронної звітності за допомогою фреймворку Django та мови програмування Python.

Практична цінність роботи полягає у розробці програмного забезпечення, що підтримує процес подання електронної звітності та дозволяє організовувати та підтримувати створення, редагування та видалення звітів користувачів.

Висновки. Виконано проєктування програмного забезпечення для ведення електронної звітності. Розроблено програмне забезпечення для ведення електронної звітності. Здійснено тестування розробленого програмного забезпечення для ведення електронної звітності.

Галузь використання – програмні засоби для ведення електронної звітності.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 44 figures, 3 tables, 3 appendixes, 19 sources.

PYTHON, DJANGO, ELECTRONIC REPORTING, PROGRAMMING LANGUAGE, SOFTWARE, FRAMEWORK.

The object of research is the process of developing software for electronic reporting.

The subject of the research is software for supporting the electronic reporting process.

The purpose of this work is to develop the software for electronic reporting of company employees.

Materials, methods and technical tools: Python programming language, Django framework.

Results. The software tools for supporting the electronic reporting process using Python programming language and Django framework have been developed.

The practical value of the work lies in the development of software that supports the process of submitting electronic reports and allows you to organize and support the creation, editing and deletion of user reports.

Conclusions. The software for an electronic reporting has been designed. The software for an electronic reporting has been developed. The developed software for an electronic reporting has been tested.

Scope of use – software tools for conducting electronic reporting.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про електронну звітність та програмні засоби для ведення електронної звітності	11
1.2 Аналіз програмного забезпечення для ведення електронної звітності ...	14
1.3 Висновки за розділом 1	23
2 Матеріали і методи	26
2.1 Вибір мови програмування та інших технологій розробки	26
2.2 Вибір фреймворку для створення програмного забезпечення для ведення електронної звітності співробітників підприємства	29
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства	33
2.4 Висновки за розділом 2	35
3 Опис програми	36
3.1 Структура програмного забезпечення для підтримки діяльності онлайн каси продажу квитків	36
3.2 Функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства	37
3.3 Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства	41
3.4 Розширення функціональності програмного забезпечення для ведення електронної звітності співробітників підприємства за допомогою модулів	46
3.5 Висновки за розділом 3	47
4 Експлуатація, тестування та експериментальне дослідження програми	49
4.1 Призначення й умови застосування програми	49

4.2	Характеристики програмного забезпечення для ведення електронної звітності співробітників підприємства	50
4.3	Інструкція по експлуатації програми.....	50
4.3.1	Звернення до програми.....	50
4.3.2	Вхідні й вихідні дані	51
4.3.3	Повідомлення.....	51
4.4	Виконання програмного забезпечення для ведення електронної звітності співробітників підприємства	51
4.5	Тестування програмного забезпечення для ведення електронної звітності співробітників підприємства	55
4.6	Висновки за розділом 4	56
	Висновки	57
	Перелік джерел посилання	60
	Додаток А Технічне завдання	62
	Додаток Б Фрагмент тексту програми	66
	Додаток В Слайди презентації	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CRM – Customer Relationship Management, система взаємодії з клієнтом;

MVC – Model-View-Controller, модель-вид-представлення;

ОС – операційна система;

ПЗ – програмне забезпечення;

ТЗ – технічне завдання.

ВСТУП

У наш час електронна звітність стає необхідною для підприємств у сучасному світі, де швидкість, точність та доступність інформації є ключовими чинниками успіху. Програмні засоби електронної звітності дозволяють збирати та зберігати дані про співробітників, а також результати їх діяльності [1]-[6]. Програмні системи підтримки процесу подання та збору електронної звітності забезпечують легкість доступу та обробки відповідних даних з будь-якого місця, що полегшує роботу фахівців з управління персоналом. Автоматизована обробка цих даних також допомагає уникнути помилок, що можуть виникнути при ручній обробці [1]-[6]. Програмні засоби ведення електронної звітності співробітників підприємства дозволяють заощадити час та ресурси компанії, оскільки електронна звітність дозволяє значно скоротити час, який потрібен для обробки та аналізу даних [1]-[6]. Такі програмні засоби також зменшують потребу у паперовій документації та фізичному зберіганні даних. Програмні засоби ведення електронної звітності співробітників підприємства допомагають виконувати вимоги стосовно зберігання та обробки особистих даних співробітників, що є важливим аспектом забезпечення відповідності вимогам сучасного законодавства про захист персональних даних [1]-[6]. Програмні засоби ведення електронної звітності співробітників підприємства надають засоби аналітики та стратегічного управління: збір та аналіз даних про персонал дозволяє підприємствам розробляти стратегії управління персоналом, виявляти тенденції та удосконалювати процеси найму, зберігання та розвитку кадрів підприємства [1]-[6].

Проте інтерфейс деяких програмних засобів підтримки електронної звітності може бути складним для розуміння або використання, що впливає на продуктивність та швидкість роботи. Деякі програмні засоби електронної звітності можуть бути складними для інтеграції з іншими програмними системами підприємства, що може ускладнити обмін даними або їхню

взаємодію. Крім того, якщо програмне забезпечення для електронної звітності базується на певних технологіях, воно може стати застарілим у майбутньому, вимагаючи постійного оновлення або переходу на нові платформи, що може бути витратним, що обумовлює необхідність своєчасного оновлення відповідного програмного забезпечення шляхом модифікації існуючих програмних модулів або розробки нового програмного забезпечення [1]-[6].

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та розробки програмного забезпечення для ведення електронної звітності співробітників підприємства.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для ведення електронної звітності;
- розробити модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства;
- здійснити проєктування програмного забезпечення для ведення електронної звітності співробітників підприємства;
- створити програмне забезпечення онлайн для ведення електронної звітності співробітників підприємства;
- виконати тестування розробленого програмного забезпечення для ведення електронної звітності співробітників підприємства.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про електронну звітність та програмні засоби для ведення електронної звітності

Електронна звітність включає в себе операції, пов'язані з отриманням, обробкою та передачею інформації у цифровій формі, спеціалізованими програмами. Це охоплює підготовку та надсилання різноманітних фінансових звітів, податкових декларацій та інших документів про фінанси компанії в електронному вигляді. Електронна звітність спрощує обмін даними між організаціями, регуляторами та іншими учасниками, зменшуючи час, потрібний на обробку інформації, і ризик помилок, що можуть виникнути при ручному введенні даних [1]-[7].

Електронна звітність у сфері бізнесу означає здійснення процесу збору, обробки та передачі фінансових та податкових даних у цифровому форматі для відповідності вимогам законодавства та стандартів звітності. Ця система включає в себе підготовку та представлення різноманітних звітів, таких як рахунки, податкові документи, відомості про доходи та витрати у спеціальному електронному вигляді [1]-[7].

На підприємствах електронна звітність оптимізує процеси обліку, аналізу та звітності, скорочує час підготовки звітів і дозволяє швидше реагувати на зміни у фінансовому становищі бізнесу. Це також забезпечує більш точні та надійні дані, оскільки зменшує ймовірність помилок під час введення даних вручну [1]-[7].

Використання електронної звітності дозволяє підприємствам впроваджувати технологічні рішення для автоматизації процесу збору та обробки даних. Це сприяє підвищенню продуктивності та забезпечує відповідність вимогам законодавства у сфері звітності [1]-[7].

Програмні засоби для електронної звітності компанії включає підтримують декілька процесів, які спрощують збір, обробку та подання

інформації у цифровій формі, внаслідок чого забезпечуються такі переваги [1]-[7]:

- автоматизація обліку: електронна звітність дозволяє автоматизувати багато етапів збору та аналізу даних, зменшуючи ризик помилок та спрощуючи процес [1]-[7];

- точність і надійність: Використання цифрових інструментів для збору та обробки даних забезпечує більш точну та надійну звітність [1]-[7];

- електронне зберігання: Електронна звітність дозволяє зберігати та систематизувати дані у цифровому форматі, що полегшує доступ до інформації [1]-[7];

- підвищена доступність: Електронні звіти можуть надаватися зацікавленим сторонам через відповідний доступ, полегшуючи комунікацію [1]-[7];

- швидкий аналіз: Цифрові інструменти дозволяють швидко аналізувати дані та звітувати, що полегшує прийняття управлінських рішень [1]-[7];

- відповідність законодавству: Електронна звітність сприяє відповідності законодавчим стандартам, спрощуючи процес збору та звітності [1]-[7];

- безпека: Цифрові системи зберігання даних дозволяють контролювати доступ та захищати інформацію від несанкціонованого доступу [1]-[7].

Програмне забезпечення для електронної звітності розроблене для автоматизації збирання, обробки та аналізу різних видів даних, таких як фінансові показники та оперативні дані, необхідні для створення різноманітних звітів.

Програмне забезпечення для електронної звітності також знижують потребу у паперовій документації та фізичному зберіганні даних. Крім того, такі програмні засоби допомагають співробітникам підприємства виконувати вимоги щодо зберігання та обробки особистих даних, що є важливим аспектом

забезпечення відповідності сучасному законодавству про захист особистої інформації [1]-[7].

Таке програмне забезпечення спрямоване на створення звітності, що відповідає галузевим стандартам та нормативним вимогам. Ці інструменти забезпечують безпечне зберігання та обробку інформації, включають заходи безпеки та полегшують електронну передачу повідомлень. Також вони сприяють виконанню вимог законодавства та контролю якості даних, зокрема, забезпечуючи цілісність та точність інформації [1]-[7].

Відзначено, що електронна звітність стає необхідною для підприємств у сучасному світі, де швидкість, точність та доступність інформації є ключовими чинниками успіху. Показано, що програмні засоби електронної звітності дозволяють збирати та зберігати дані про співробітників, а також результати їх діяльності. Відзначено, що програмні системи підтримки процесу подання та збору електронної звітності забезпечують легкість доступу та обробки відповідних даних з будь-якого місця, що полегшує роботу фахівців з управління персоналом. Автоматизована обробка цих даних також допомагає уникнути помилок, що можуть виникнути при ручній обробці. Програмні засоби ведення електронної звітності співробітників підприємства дозволяють заощадити час та ресурси компанії, оскільки електронна звітність дозволяє значно скоротити час, який потрібен для обробки та аналізу даних. Показано, що такі програмні засоби також зменшують потребу у паперовій документації та фізичному зберіганні даних, крім того програмні засоби ведення електронної звітності співробітників підприємства допомагають виконувати вимоги стосовно зберігання та обробки особистих даних співробітників, що є важливим аспектом забезпечення відповідності вимогам сучасного законодавства про захист персональних даних. Програмні засоби ведення електронної звітності співробітників підприємства надають засоби аналітики та стратегічного управління: збір та аналіз даних про персонал дозволяє підприємствам розробляти стратегії управління персоналом,

виявляти тенденції та удосконалювати процеси найму, зберігання та розвитку кадрів підприємства.

1.2 Аналіз програмного забезпечення для ведення електронної звітності

Проаналізуємо програмне забезпечення для ведення електронної звітності [8]-[15].

Програмне забезпечення bitrix24 [13] є онлайн-сервісом для полегшення роботи бізнесу, складається з декількох програм, в які входить програма звітності працівників щодо виконаної роботи (рис. 1.1).

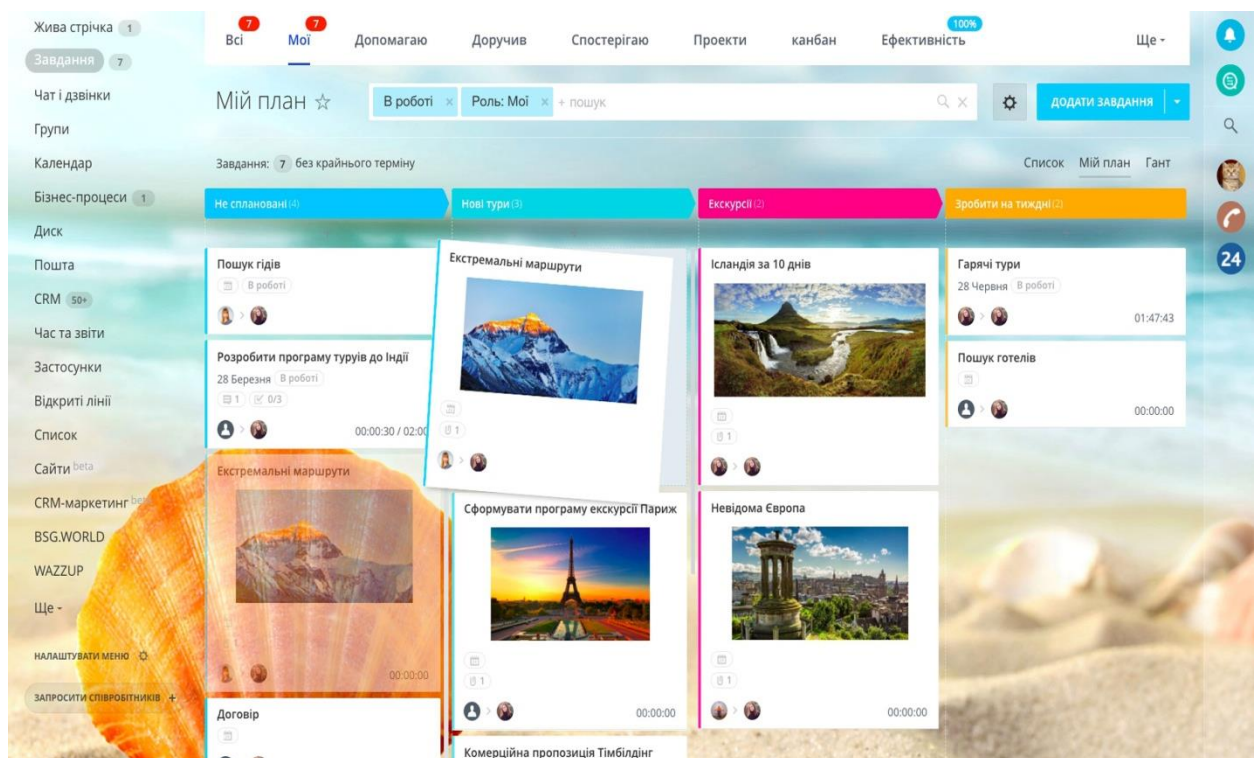


Рисунок 1.1 – Інтерфейс сервісу bitrix24 [13]

Програмне забезпечення Bitrix24 пропонує рішення для ефективного управління бізнесом, надаючи інструменти для комунікації, керування проектами, систему взаємодії з клієнтом (Customer Relationship Management CRM), організації документів та багато іншого. Це об'єднання засобів для

спілкування в команді, планування завдань, керування графіками роботи та інші функції для оптимізації робочих процесів. Програмне забезпечення Bitrix24 має інтегровану CRM для управління клієнтами та продажами, а також доступний онлайн для підприємств будь-якого масштабу [13].

Програмне забезпечення Bitrix24 поєднує інструменти для ефективного управління бізнесом, спілкування у команді та роботи з клієнтами. Функції програмного забезпечення Bitrix24 включають систему CRM для клієнтських взаємодій, планування завдань, інструменти для співпраці та обміну документами через інтегрований чат. Bitrix24 також пропонує аналітику, інструменти для платежів та покупок онлайн [13].

Мобільний додаток програмного забезпечення Bitrix24 працює на iOS та Android, дозволяючи користувачам отримувати доступ до системи в будь-який час та з будь-якого місця. Програмне забезпечення Bitrix24 також спрощує інтеграцію з іншими програмами та сервісами, що забезпечує більшу узгодженість та ефективність у роботі з даними і процесами [13].

У плані використання, програмне забезпечення Bitrix24 має дві версії: хмарні (доступні через веб) та можливість встановлення на власному сервері для корпоративних клієнтів, які мають бажання володіти своїми власними серверами [13].

Програмне забезпечення Bitrix24 є не просто інструментом для керування бізнесом. Програмне забезпечення Bitrix24 поєднує різноманітні функції, дозволяючи працювати в команді, взаємодіяти з клієнтами та вести усі бізнес-процеси в єдиному цифровому середовищі [13].

Програмне забезпечення Bitrix24 володіє системою CRM для взаємодії з клієнтами, інструментами для планування завдань та проектів, співпраці та обміну інформацією в чатах, а також обміну файлами та документами. Це не просто комплект окремих інструментів, а повна екосистема для ефективної роботи та зручного спілкування як зі своєю командою, так і з клієнтами [13].

Також, програмне забезпечення Bitrix24 має кілька функцій аналізу, які дозволяють створювати звіти, проводити аналіз результатів та вести

статистику роботи. Головною перевагою є можливість інтеграції з іншими сервісами та програмами, що спрощує обробку та обмін даними [13].

Незалежно від розміру компанії, програмне забезпечення Bitrix24 пропонує інструменти для автоматизації робочих процесів, підвищення продуктивності та полегшення командної роботи. Програмне забезпечення Bitrix24 є гнучким, оскільки дозволяє використовувати лише потрібні інструменти для підтримки діяльності відповідного бізнесу, без надмірних функцій, які не потрібні користувачу [13].

Перевагами програмного забезпечення bitrix24 є такі [13]:

- широкий спектр послуг;
- наявність цілодобової підтримки.
- програмне забезпечення bitrix24 надає можливості управління клієнтами, відстеження продажів;
- засоби управління завданнями та проектами: наявні програмні інструменти для планування завдань, робочих груп, календарі для спільної роботи членів команди;
- засоби підтримки вбудованого чату та інших засобів спілкування між членами команди розробників, можливість створення групових чатів;
- засоби підтримки документообігу: створення та збереження документів, спільний доступ та редагування документів;
- засоби створення звітів та аналізу даних про роботу користувачів;
- програмні інструменти для підтримки роботи з електронної комерцією, програмні інструменти обробки платежів, створення сайтів;
- засоби керування контактами: взаємодія з клієнтами через різні канали, управління контактами, інтеграція з різними маркетинговими інструментами;
- мобільні додатки: програмне забезпечення bitrix24 підтримує мобільні застосунки для iOS та Android для роботи в системі;
- засоби інтеграція: можливість інтеграції з іншими програмами, сервісами і додатками;

– програмне забезпечення bitrix24 пропонує хмарні версії та можливість самостійного хостингу для корпоративних потреб [13].

Програмне забезпечення Jira є комерційною системою, призначеною для керування програмними проектами та взаємодії з користувачами (працівниками підприємства), з наявним вебінтерфейсом [14].

Програмне забезпечення Jira є платформою для керування проектами та спільної роботи. Вона спрямована на поліпшення організації завдань, відслідковування їх статусу та забезпечення більш ефективної командної праці. Це програмне забезпечення дозволяє створювати, розподіляти та контролювати завдання, встановлювати пріоритети, терміни та відслідковувати прогрес їх виконання [14].

Програмне забезпечення Jira має різні налаштування, які спрощують управління завданнями у всіх типах проектів. Програмне забезпечення Jira дозволяє створювати різноманітні завдання, встановлювати власні поля та налаштовувати робочі процеси для відповідності потребам кожного конкретного проекту чи команди [14].

Основні функції Jira включають створення та відстеження завдань, перегляд календарів і термінів, аналіз ходу проекту у звітах, співпрацю між завданнями та легку інтеграцію з іншими програмами для підвищення продуктивності та ефективності. Jira (рис. 1.2) гнучко налаштовується та корисна для всіх видів галузей та проектів [14].

Програмне забезпечення Jira є платформою, яка сприяє ефективному управлінню проектами командами. Програмне забезпечення Jira призначена для організації завдань, відстеження їх статусу, планування роботи та спілкування в команді, а також звітів про прогрес [14].

Основна мета програмного забезпечення Jira полягає в спрощенні робочих процесів через методи управління проектами, такі як Scrum, Kanban або Waterfall. Jira дозволяє створювати та організовувати завдання, встановлювати терміни виконання, призначати відповідальних та звітувати про прогрес у виконанні кожного завдання та всього проекту [14].

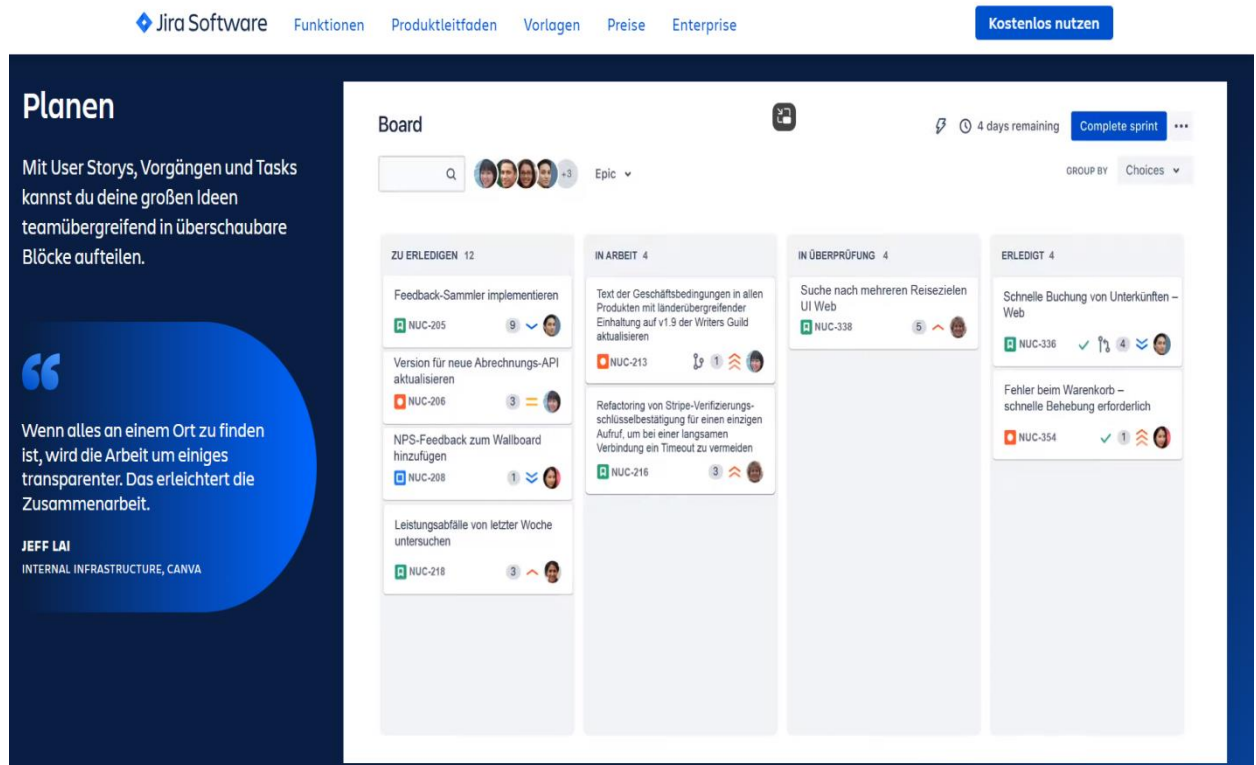


Рисунок 1.2 – Інтерфейс системи Jira [14]

Однією з основних переваг програмного забезпечення Jira є можливість інтеграції з різними інструментами. Вона підтримує різноманітні програми і плагіни, що дозволяють користувачам налаштовувати систему під свої потреби і інтегрувати її з іншими популярними інструментами проекту для сумісності та простоти використання [14].

Крім цього, програмне забезпечення Jira забезпечує можливості звітності про проект та аналізу даних, що допомагає командам вимірювати продуктивність, виявляти можливості для вдосконалення та приймати обґрунтовані управлінські рішення [14].

Програмне забезпечення Jira надає можливість створювати та керувати проектами у будь-якій галузі та пропонує гнучкість налаштування. Користувач може налаштувати різні параметри проекту та відстежувати хід виконання завдань відповідно до своїх потреб. Також, Jira контролює робочий час та відстежує продуктивність, полегшуючи командну роботу, спільні завдання і спілкування між учасниками проекту. Jira використовується в різних галузях

завдяки своїй гнучкості та здатності адаптуватися до різних потреб і специфіки проектів [14].

Перевагами програмного забезпечення Jira є такі [14]:

- простий та зрозумілий інтерфейс користувача [14];
- засоби ефективного планування завдань: програмне забезпечення Jira забезпечує створення, організацію та призначення завдань, встановлення термінів виконання та відстеження прогресу [14];
- достатньо висока швидкість роботи [14];
- чітка структура подання інформації [14];
- підтримка гнучкості в управлінні проектами: програмне забезпечення Jira дозволяє використовувати різні методики управління проектами, такі як Scrum, Kanban, Waterfall тощо [14];
- широкі можливості налаштування: програмне забезпечення Jira дозволяє інтегрувати різні програмні модулі та розширення, а також налаштовувати робоче середовище під конкретні потреби користувача [14];
- можливість інтеграції з іншими програмними інструментами: програмне забезпечення Jira підтримує велику кількість інтеграцій з іншими популярними інструментами для роботи над проектами, що полегшує процеси роботи [14];
- можливість створення звітів і аналіз даних: програмне забезпечення Jira надає можливість отримання звітів та аналізу даних, що допомагає в оцінці продуктивності та прийнятті обґрунтованих управлінських рішень [14];
- гнучкість в конфігурації: можливість налаштовувати різні типи проектів та завдань, додавати нові поля, статуси, типи завдань тощо, відповідаючи специфіці вашого проекту [14];
- керування витратами часу: програмне забезпечення Jira надає функціонал для відстеження витрати часу на виконання завдань, що дозволяє оцінити ефективність роботи та розподілити завдання [14];

– масштабованість: програмне забезпечення Jira працює як для невеликих команд, так і для великих організацій, забезпечуючи гнучкість та масштабованість організації роботи різних груп [14];

– підтримка командної роботи: завдяки різним функціям співпраці та спільного виконання завдань, програмне забезпечення Jira полегшує роботу в команді, забезпечуючи взаємодію та співпрацю між учасниками проекту [14];

– підтримка різних галузей: програмне забезпечення Jira використовується в різних галузях, зокрема, у галузях розробки програмного забезпечення, маркетинг, HR, фінанси, управління продуктом тощо [14].

Програмне забезпечення Mantis Bug Tracker є програмною системою, яка використовується для загальних потреб відстеження запитів, управління проектами, інтерфейс представлено вебсайтом (рис. 1.3) [15].

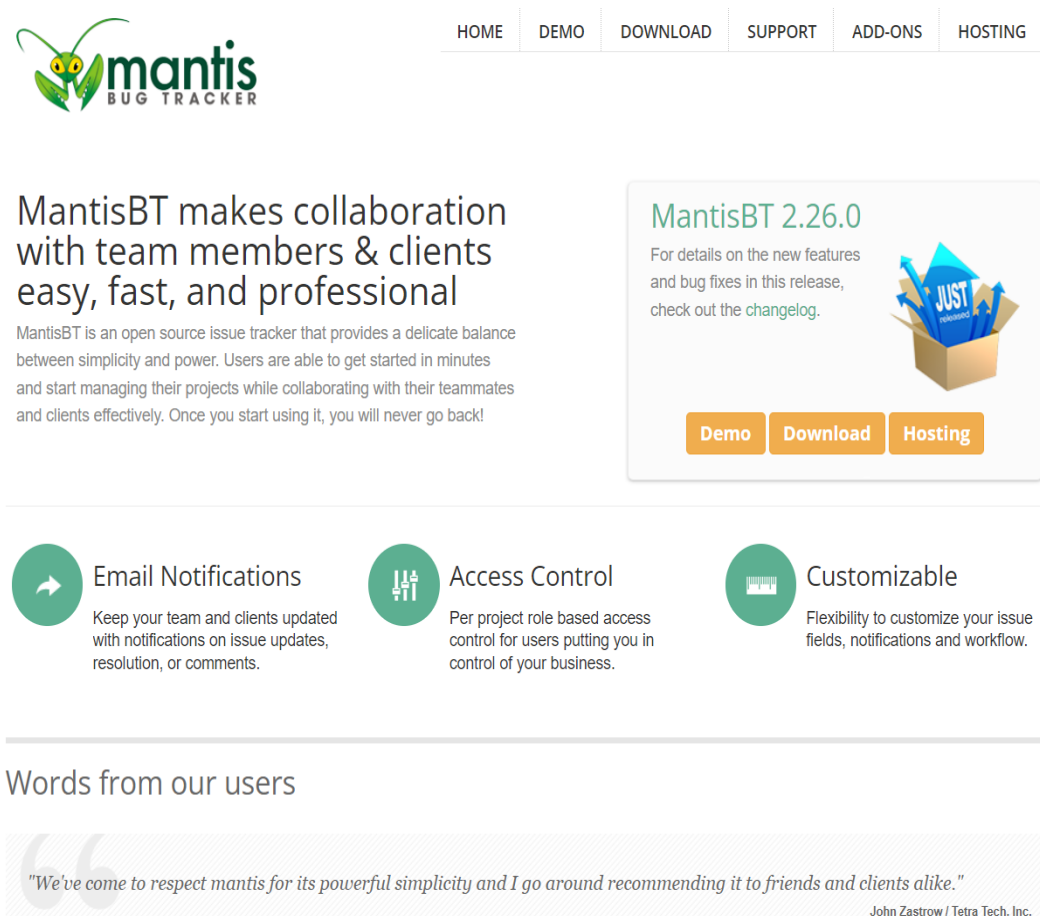


Рисунок 1.3 – Інтерфейс системи Mantis Bug Tracker [15]

Перевагами програмного забезпечення Mantis Bug Tracker є такі [15]:

- більше 10 років на ринку послуг, що може свідчити про відсутність «дитячих помилок»;
- легка інтерфейсна частина [15];
- засоби реєстрації помилок: зручний інтерфейс для додавання дефектів, що дозволяє описати проблему, прикріпити файли або зображення, визначити її пріоритет і серйозність [15];
- відстеження статусу: програмне забезпечення Mantis Bug Tracker надає можливість контролювати стан виявлених помилок, від відкриття до закриття, включаючи різноманітні проміжні етапи виправлення [15];
- призначення відповідальних: програмне забезпечення Mantis Bug Tracker надає можливість визначення осіб, відповідальних за виправлення певних помилок або внесення змін [15];
- коментування та спілкування: інтегровані функції коментування, реалізовані у програмі, допомагають команді вирішувати проблеми шляхом обміну інформацією [15];
- звіти та статистика: можливість генерувати звіти про стан помилок, строків виправлення та інші статистичні дані, що допомагають у керуванні проектом [15];
- програмне забезпечення Mantis Bug Tracker має гнучкі налаштування та можливість розширення через додаткові плагіни, що дає йому привабливість для використання у різних типах проектів та команд [15];
- простота використання: програмне забезпечення Mantis Bug Tracker досить легка для розуміння і використання. Користувачі можуть швидко навчитися використовувати її без значних зусиль [15];
- гнучкість конфігурації: програмне забезпечення Mantis Bug Tracker дозволяє налаштовувати різні параметри для відстеження проблем, створювати власні звіти та додаткові поля, адаптовані під конкретні потреби проекту [15];

– спільна робота команд: програмне забезпечення Mantis Bug Tracker дозволяє кільком користувачам одночасно працювати з однією чи кількома проблемами, обмінюватися коментарями та спілкуватися для ефективного розв’язання задач [15];

– підтримка спільноти: програмне забезпечення Mantis Bug Tracker активно підтримується та оновлюється спільнотою користувачів, що робить її вдосконалення більш гнучким та відповідним сучасним потребам [15];

– легкість у встановленні: програмне забезпечення Mantis Bug Tracker можна швидко встановити і запустити, що дозволяє почати використання майже одразу після встановлення [15].

Порівняльну характеристику програмного забезпечення для ведення електронної звітності наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для ведення електронної звітності

Критерій порівняння	bitrix24 [13]	Jira [14]	Mantis [15]
Зручність використання	–	+	+
Дружній інтерфейс користувача	+	+–	+
Інтуїтивна зрозумілість	–	–	+
Багатофункціональність	+–	+	+
Засоби інтеграції з іншими програмними продуктами	+–	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для ведення електронної звітності. Проте інтерфейс деяких програмних засобів підтримки електронної звітності може бути складним для розуміння або використання, що впливає на продуктивність та швидкість роботи. Деякі програмні засоби електронної

звітності можуть бути складними для інтеграції з іншими програмними системами підприємства, що може ускладнити обмін даними або їхню взаємодію. Крім того, якщо програмне забезпечення для електронної звітності базується на певних технологіях, воно може стати застарілим у майбутньому, вимагаючи постійного оновлення або переходу на нові платформи, що може бути витратним, що обумовлює необхідність своєчасного оновлення відповідного програмного забезпечення шляхом модифікації існуючих програмних модулів або розробки нового програмного забезпечення. Тому актуальною є розробка програмного забезпечення для ведення електронної звітності співробітників підприємства.

При розробці програмного забезпечення для ведення електронної звітності необхідно забезпечити такі функціональні вимоги:

- можливість додавання звіту користувачем щодо виконаної роботи;
- можливість редагування наявного звіту;
- можливість видалення наявного звіту користувача;
- реєстрація нового користувача у програмному забезпеченні для ведення електронної звітності (функція адміністратора);
- можливість зміни облікових даних зареєстрованого користувача;
- можливість додавання нового проєкту для подальшого внесення звітів щодо його реалізації;
- можливість редагування інформації про наявні проєкти;
- можливість перегляду звітів користувачів (виконавців проєктів);
- можливість затвердження звітів користувачів.

1.3 Висновки за розділом 1

Відзначено, що електронна звітність стає необхідною для підприємств у сучасному світі, де швидкість, точність та доступність інформації є ключовими чинниками успіху. Показано, що програмні засоби електронної звітності дозволяють збирати та зберігати дані про співробітників, а також

результати їх діяльності. Відзначено, що програмні системи підтримки процесу подання та збору електронної звітності забезпечують легкість доступу та обробки відповідних даних з будь-якого місця, що полегшує роботу фахівців з управління персоналом. Автоматизована обробка цих даних також допомагає уникнути помилок, що можуть виникнути при ручній обробці. Програмні засоби ведення електронної звітності співробітників підприємства дозволяють заощадити час та ресурси компанії, оскільки електронна звітність дозволяє значно скоротити час, який потрібен для обробки та аналізу даних. Показано, що такі програмні засоби також зменшують потребу у паперовій документації та фізичному зберіганні даних, крім того програмні засоби ведення електронної звітності співробітників підприємства допомагають виконувати вимоги стосовно зберігання та обробки особистих даних співробітників, що є важливим аспектом забезпечення відповідності вимогам сучасного законодавства про захист персональних даних. Програмні засоби ведення електронної звітності співробітників підприємства надають засоби аналітики та стратегічного управління: збір та аналіз даних про персонал дозволяє підприємствам розробляти стратегії управління персоналом, виявляти тенденції та удосконалювати процеси найму, зберігання та розвитку кадрів підприємства.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для ведення електронної звітності. Проте інтерфейс деяких програмних засобів підтримки електронної звітності може бути складним для розуміння або використання, що впливає на продуктивність та швидкість роботи. Деякі програмні засоби електронної звітності можуть бути складними для інтеграції з іншими програмними системами підприємства, що може ускладнити обмін даними або їхню взаємодію. Крім того, якщо програмне забезпечення для електронної звітності базується на певних технологіях, воно може стати застарілим у майбутньому, вимагаючи постійного оновлення або переходу на нові платформи, що може бути витратним, що обумовлює необхідність своєчасного оновлення

відповідного програмного забезпеченням шляхом модифікації існуючих програмних модулів або розробки нового програмного забезпечення. Тому актуальною є розробка програмного забезпечення для ведення електронної звітності співробітників підприємства.

Сформульовано функціональні вимоги до програмного забезпечення для ведення електронної звітності співробітників підприємства.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

При розробці програмного забезпечення для ведення електронної звітності співробітників підприємства було використано мову програмування Python [16], [17].

Мова програмування Python – це мова програмування високого рівня, яка використовується для багатьох завдань. Її популярність зумовлена простотою, можливістю розширення та доступністю [16], [17]. Гнучкий синтаксис мови програмування Python приваблює як початківців, так і досвідчених програмістів, адже вона використовується в різних сферах, включаючи веброзробку, аналіз даних, штучний інтелект та багато інших галузей. Велика кількість бібліотек та підтримка спільноти роблять Python потужним інструментом для програмування [16], [17].

Особливостями мови програмування Python, які також є її перевагами, включають такі аспекти [16], [17]:

- простий синтаксис: синтаксис мови Python схожий на англійську мову, що робить код більш читабельним і зрозумілим [16], [17];
- універсальність: мова Python має стандартну бібліотеку, що надає широкий спектр інструментів для роботи з файлами, мережами, базами даних та іншими функціями [16], [17];
- активна спільнота: спільнота розробників на мові програмування Python сприяє швидкому вирішенню проблем і постійному вдосконаленню мови [16], [17];
- гнучкість: мову програмування Python можна розширити, використовуючи код з інших мов, таких як C/C++, що дозволяє підвищити ефективність чи використовувати системні ресурси [16], [17];
- сфери застосування: Python використовується у веб-розробці, наукових обчисленнях, аналізі даних, штучному інтелекті, створенні ігор, автоматизації завдань та інших галузях [16], [17];

- підтримка ООП: мова програмування Python підтримує парадигму об'єктно-орієнтованого програмування [16], [17];
- просте написання: в Python необов'язково визначати змінні перед використанням, що спрощує процес розробки [16], [17];
- кросплатформенність: Python може працювати на різних операційних системах без значних змін у коді [16], [17].

Якщо потрібно використовувати систему контролю версій для налаштування програмного проекту, часто для цього використовується мова програмування Python. Більшість тем та плагінів для систем управління версіями ґрунтуються на Python [16], [17]. Чимало вебсайтів обмінюються даними, що надходять від інших вебсайтів або сервісів, які використовують Python. Портали новин, які представляють погоду в певному регіоні, часто не мають власних метеостанцій і, зазвичай, використовують дані з інших джерел, які необхідно інтегрувати на своєму вебсайті, для чого часто використовується мова програмування Python [14], [15].

Завдяки своїй доступності та великій кількості бібліотек, мова Python стала вибором для програмних проєктів будь-якого масштабу. Ця мова надихає розробників своєю читабельністю коду та простотою синтаксису, що робить його ідеальним вибором як для початківців, так і для досвідчених програмістів [14], [15].

Python визначається своєю популярністю, активною спільнотою розробників та широким спектром можливостей [16], [17]. Ця інтерпретована мова програмування має об'єктно-орієнтовану парадигму та динамічне виконання. Використання високорівневих структур даних спільно з гнучкою семантикою та динамічними зв'язками приваблює для швидкої розробки програм і інтеграції компонентів [16], [17].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для ведення електронної звітності співробітників підприємства наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для ведення електронної звітності співробітників підприємства

Критерій порівняння мов програмування	Мова програмування		
	Python	JavaScript	PHP
Зручність для розробки ПЗ для ведення електронної звітності співробітників підприємства	+	–	+–
Ефективність	+	+	+–
Зручність синтаксису	+	+–	+–
Велика спільнота	+	+–	+
Розширюваність	+	+–	+

Отже, для реалізації програмного забезпечення для ведення електронної звітності співробітників підприємства обрано мову програмування Python, яка є мовою високого рівня, що використовується для розв’язання багатьох завдань, пов’язаних зі створенням програмних засобів, є досить простою та доступною, надає можливості розширення, має гнучкий синтаксис. Крім того, ця мова програмування використовується в різних сферах, включаючи веброзробку, аналіз даних, штучний інтелект та багато інших галузей, що забезпечує можливість інтеграції розробленої програми в інші програмні продукти.

2.2 Вибір фреймворку для створення програмного забезпечення для ведення електронної звітності співробітників підприємства

В якості фреймворку для створення програмного забезпечення для ведення електронної звітності співробітників підприємства було обрано Django [18], [19].

Django є ефективним вебфреймворком для Python з швидким створенням програм та простим, органічним дизайном. Розроблений досвідченими спеціалістами, він спрощує процес веброзробки, щоб розробники могли зосередитися на своєму програмному коді, не втрачаючи часу на надмірні технічні деталі. Фреймворк Django є безкоштовним та розповсюджується з відкритим вихідним кодом [18], [19].

З фреймворком Django вебпрограма може перейти від ідеї до релізу всього за кілька годин. Фреймворк Django вкладає багато зусиль у веброзробку, тому розробник може фокусуватися на створенні своєї програми, не втрачаючи часу на вирішення технічних проблем. І все це безкоштовно та з можливістю доступу до вихідного коду [18], [19].

Фреймворк Django створено для швидкої розробки програм, щоб візуалізувати їх від ідеї до завершення [18], [19]. У Django є велика кількість додатків, які можна використовувати для типових завдань веброзробки. Він надає інструменти для аутентифікації користувачів, управління контентом, створення карт сайту, RSS-каналів та багато іншого [18], [19].

Django є абсолютно безпечним. Фреймворк серйозно ставиться до захисту та допомагає розробникам уникнути багатьох типових проблем, таких як SQL-ін'єкції, міжсайтові скрипти, міжсайтові запити та перехоплення кліків. Його система аутентифікації користувачів забезпечує безпечний спосіб управління обліковими записами та паролями користувачів [18], [19].

Властивість масштабованості в Django вразливих областях використовується для ефективного збільшення ресурсів і швидкої адаптації до великого обсягу трафіку [18], [19].

Фреймворк Django виявився універсальним використанням для створення різноманітних проектів: від систем управління контентом до соціальних мереж і наукових обчислювальних платформ для бізнесу, уряду та організацій [18], [19].

Основна мета фреймворку Django – спростити складність вебсайтів, що працюють з базами даних. Фреймворк акцентує на використанні компонентів та принципі "підключеності", де менше коду сприяє швидкій розробці, і це не потребує повторення. Python використовується в усіх налаштуваннях, файлах та моделях даних. Django надає додатковий інтерфейс для створення, читання, оновлення та видалення, який генерується динамічно з використанням самоаналізу та шаблонів управління [18], [19].

Незважаючи на власні терміни, такі як «views» для елементів, що формують HTTP-відповіді, основу Django можна розглядати як архітектуру модель-вид-представлення (Model-View-Controller, MVC). Цей фреймворк складається з об'єктно-реляційного відображення з моделями даних, які представляються у вигляді класів на мові Python, та реляційною базою даних («модель»), вебкомпонентами («вид»), системою обробки HTTP-запитів та звичайними маршрутизаціями між брокерами за допомогою Expression URL Dispatcher («диспетчер») [18], [19].

Django може бути впроваджено з використанням різних програмних модулів Python, а також на різних серверах, зокрема, Apache, Nginx. Для фреймворку Django офіційно підтримуються п'ять баз даних на серверному рівні, а також є віддалені серверні модулі [18], [19].

Фреймворк Django також може бути використаний разом із модулями, написаними на мові Java, на будь-якому сервері Java EE.

Для цього потрібно встановити Django-Jython, щоб мати можливість використовувати відповідний драйвер для підключення до бази даних та функції, які дозволяють компілювати Django у відповідний .war файл для розповсюдження [18], [19].

Архітектуру програмного забезпечення на основі фреймворку Django наведено на рис. 2.1.

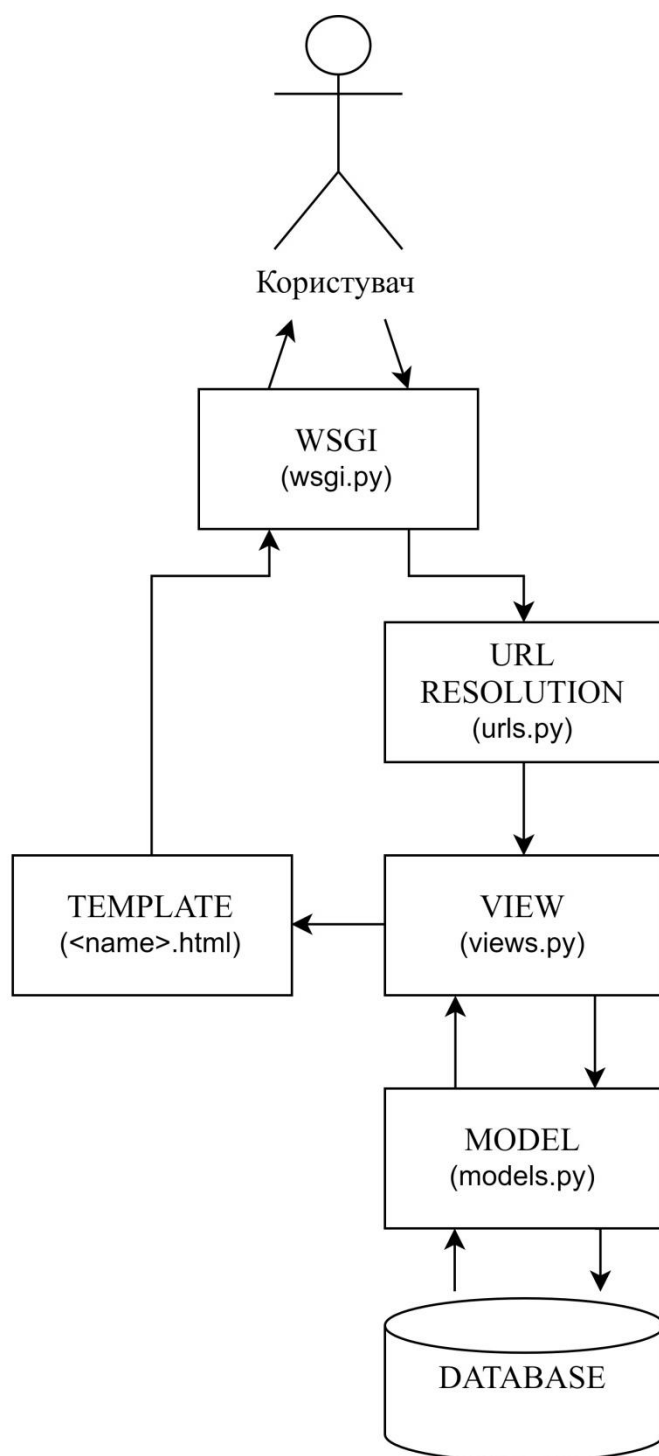


Рисунок 2.1 – Архітектура програмного забезпечення на основі фреймворку Django

Обґрунтування вибору фреймворку для створення програмного забезпечення для ведення електронної звітності співробітників підприємства наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору фреймворку для створення програмного забезпечення для ведення електронної звітності співробітників підприємства

Критерій порівняння фреймворків	Фреймворки		
	Django	Flask	Ruby on Rails
Зручність для створення програми для ведення електронної звітності	+	—	—
Базова мова програмування	Python	Python	Ruby
Висока швидкість розробки	+	+—	+
Велика спільнота розробників	+	+—	+
Наявність великої кількості документації	+	+	+—

Таким чином, для створення програмного забезпечення для ведення електронної звітності співробітників підприємства обрано фреймворк Django, що забезпечує швидке створення програм, є простим та характеризується зручним дизайном. Фреймворк Django спрощує процес веброзробки, він є безкоштовним та розповсюджується з відкритим вихідним кодом, а також містить велику кількість модулів, які можна використовувати для типових завдань розробки програмного забезпечення.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства

З метою можливості розробки програмного забезпечення для ведення електронної звітності співробітників підприємства створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства враховуємо те, що користувачі програми мають можливості додавання звіту користувачем щодо виконаної роботи, редагування наявного звіту, видалення наявного звіту користувача, реєстрації нового користувача у програмному забезпеченні для ведення електронної звітності (функція адміністратора), зміни облікових даних зареєстрованого користувача, додавання нового проєкту для подальшого внесення звітів щодо його реалізації, редагування інформації про наявні проєкти, перегляду звітів користувачів, затвердження звітів користувачів.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для ведення електронної звітності співробітників підприємства можна описати таким чином:

- користувач програмного забезпечення для ведення електронної звітності співробітників підприємства авторизується;
- після авторизації користувача сервер програмного забезпечення ведення електронної звітності співробітників підприємства у відповідь повертає основну форму програми;
- користувач бачить відповідну сторінку і обирає роботу зі звітами;
- користувач обирає як подальший шлях своєї роботи, додавання необхідного звіту;
- користувач здійснює додавання звіту до системи;

– користувач завершує роботу з програмним забезпеченням для ведення електронної звітності співробітників підприємства.

Враховуючи сценарії роботи користувачів при роботі з програмним забезпеченням для ведення електронної звітності співробітників підприємства, модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства подамо у вигляді діаграми прецедентів, зображеної на рис. 2.2.

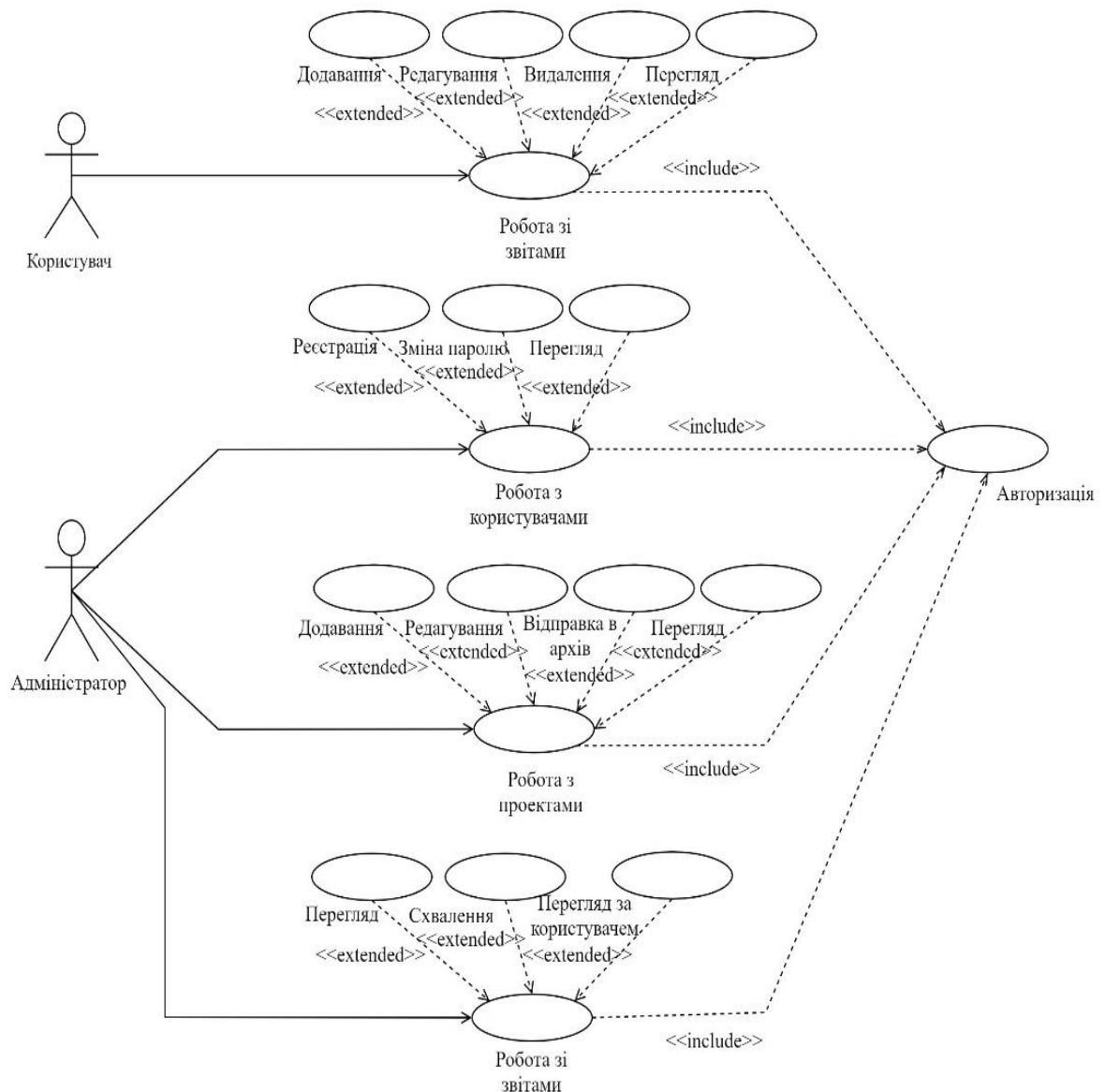


Рисунок 2.2 – Модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для ведення електронної звітності співробітників підприємства.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для ведення електронної звітності співробітників підприємства обрано мову програмування Python, яка є мовою високого рівня, що використовується для розв'язання багатьох завдань, пов'язаних зі створенням програмних засобів, є досить простою та доступною, надає можливості розширення, має гнучкий синтаксис. Крім того, ця мова програмування використовується в різних сферах, включаючи веброзробку, аналіз даних, штучний інтелект та багато інших галузей, що забезпечує можливість інтеграції розробленої програми в інші програмні продукти.

Для створення програмного забезпечення для ведення електронної звітності співробітників підприємства обрано фреймворк Django, що забезпечує швидке створення програм, є простим та характеризується зручним дизайном. Фреймворк Django спрощує процес веброзробки, він є безкоштовним та розповсюджується з відкритим вихідним кодом, а також містить велику кількість модулів, які можна використовувати для типових завдань розробки програмного забезпечення.

Створено модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для ведення електронної звітності співробітників підприємства.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для підтримки діяльності онлайн каси продажу квитків

Загальна діаграма класів програмного забезпечення для ведення електронної звітності співробітників підприємства зображена на рис. 3.1.

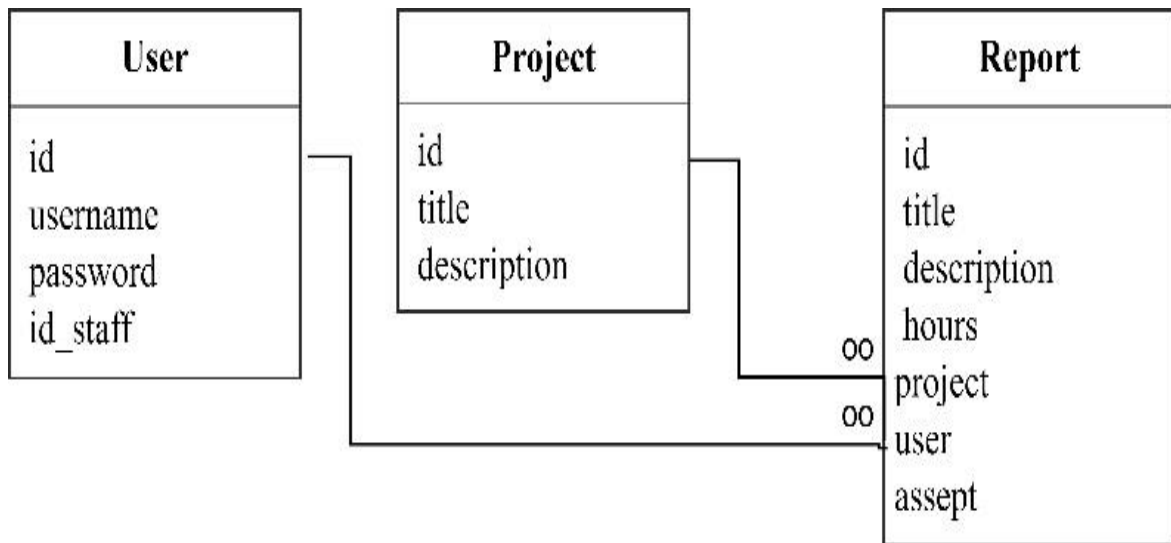


Рисунок 3.1 – Діаграма класів програмного забезпечення для ведення електронної звітності співробітників підприємства

Як видно з діаграми, основними класами програмного забезпечення для ведення електронної звітності співробітників підприємства є: користувач програми (User), проєкт (Project), звіт (Report). Користувач, містить поля номер акаунту в базі, логін, пароль. Клас проєкт, містить поля номер проєкту в базі, ім'я проєкту і його опис. Клас звіт містить наступні поля: номер звіту в базі, ім'я звіту, кількість витрачених годин, проєкт якого стосується звіт, користувач, який його створив.

Діаграма компонентів програмного забезпечення для ведення електронної звітності співробітників підприємства зображена на рис. 3.2.

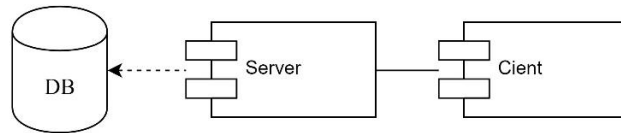


Рисунок 3.2 – Діаграма компонентів програмного забезпечення для ведення електронної звітності співробітників підприємства

Як видно з рисунку, основними компонентами є сервер програмного забезпечення для ведення електронної звітності співробітників підприємства, клієнтська частина, а також база даних з проєктами та звітами.

3.2 Функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства

Загальну схему функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства подамо у вигляді рис. 3.3.

Користувач програмного забезпечення для ведення електронної звітності співробітників підприємства створює запити до програми. Ці запити забезпечуються шляхом взаємодії з кнопками інтерфейсу. На стороні клієнта програма здійснює попередню перевірку даних на коректність, після чого ці дані надсилаються на сервер. Сервер програмного забезпечення для ведення електронної звітності співробітників підприємства отримує та обробляє дані, надіслані користувачем. У випадку необхідності сервер звертається до бази даних програми для отримання або запису певних даних. Дані, згенеровані сервером надсилаються через вебінтерфейс клієнтській частині програми, де вони інтерпретуються у зручний для користувача вигляд.

Схему функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства подамо у вигляді діаграми станів, зображеної на рис. 3.4.

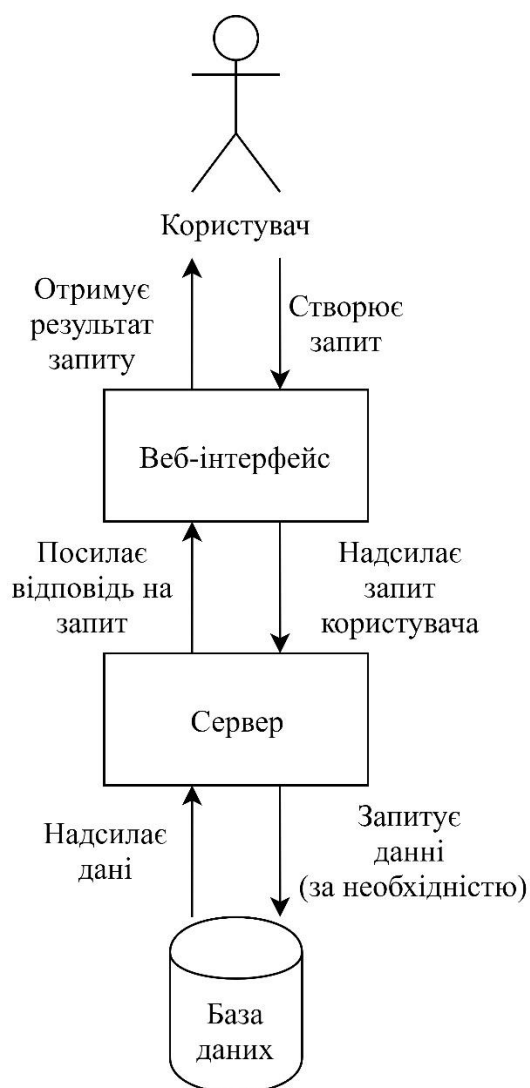


Рисунок 3.3 – Загальна схема функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства

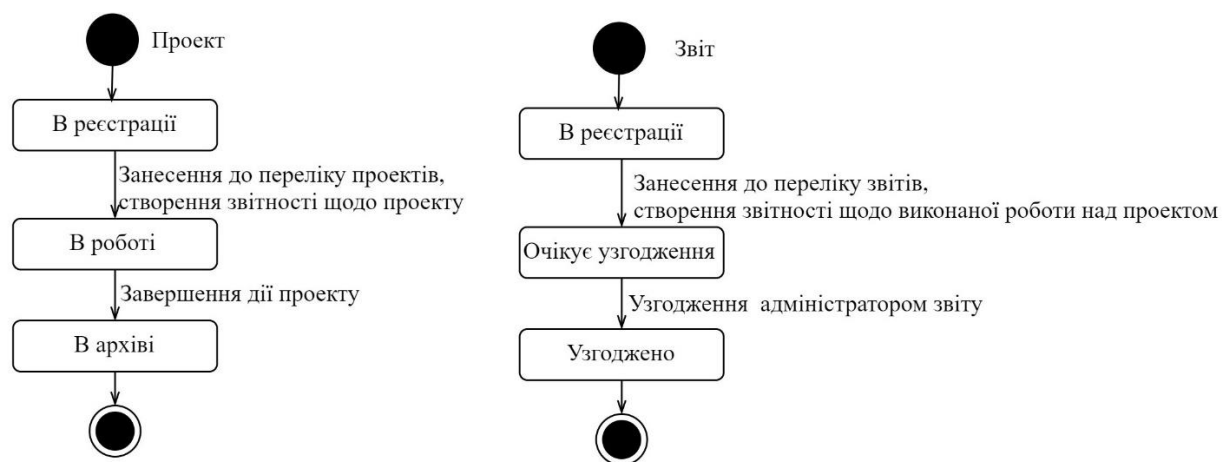


Рисунок 3.4 – Схема функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства у вигляді діаграмі станів

Оскільки основна мета розробки програми полягає в створенні звітів про виконану роботу, було побудовано діаграму станів для основних процесів програми: процесу існування проєкту та процесу існування звіту (рис. 3.4). Коли проєкт додається до програмного забезпечення для ведення електронної звітності співробітників підприємства, він перебуває у стані "реєстрація". Після валідації та відправки на сервер проєкт переходить у стан "в роботі", тобто до нього можна додавати звіти. Після того, як проєкт стає неактуальним і звіти не можуть бути створені для нього, він переходить у стан "в архіві", його можна переглядати, але не можна змінювати.

Звіт виконавця, коли додається до програмного забезпечення для ведення електронної звітності співробітників підприємства, перебуває у стані "реєстрація". Після валідації та відправки на сервер він переходить у стан "очікує узгодження", користувач може вносити зміни або видаляти його, поки адміністратор не підтвердить. Адміністратор програмного забезпечення для ведення електронної звітності співробітників підприємства може переглядати незатверджені звіти, затверджувати їх, але не може видаляти або змінювати. Після затвердження адміністратором звіт переходить у стан "узгоджено", і користувач не може внести зміни або видалити його, також він прихований від перегляду користувача.

Діаграма діяльності програмного забезпечення для ведення електронної звітності співробітників підприємства зображена на рис. 3.5. Найявні такі види діяльності програмного забезпечення для ведення електронної звітності співробітників підприємства як: додавання звіту, видалення звіту, редагування звіту та завершення роботи зі звітом.

Діаграма послідовності програмного забезпечення для ведення електронної звітності співробітників підприємства зображена на рис. 3.6. За допомогою цієї діаграми подано взаємодію авторизованого користувача, системи і бази даних програмного забезпечення для ведення електронної звітності співробітників підприємства.

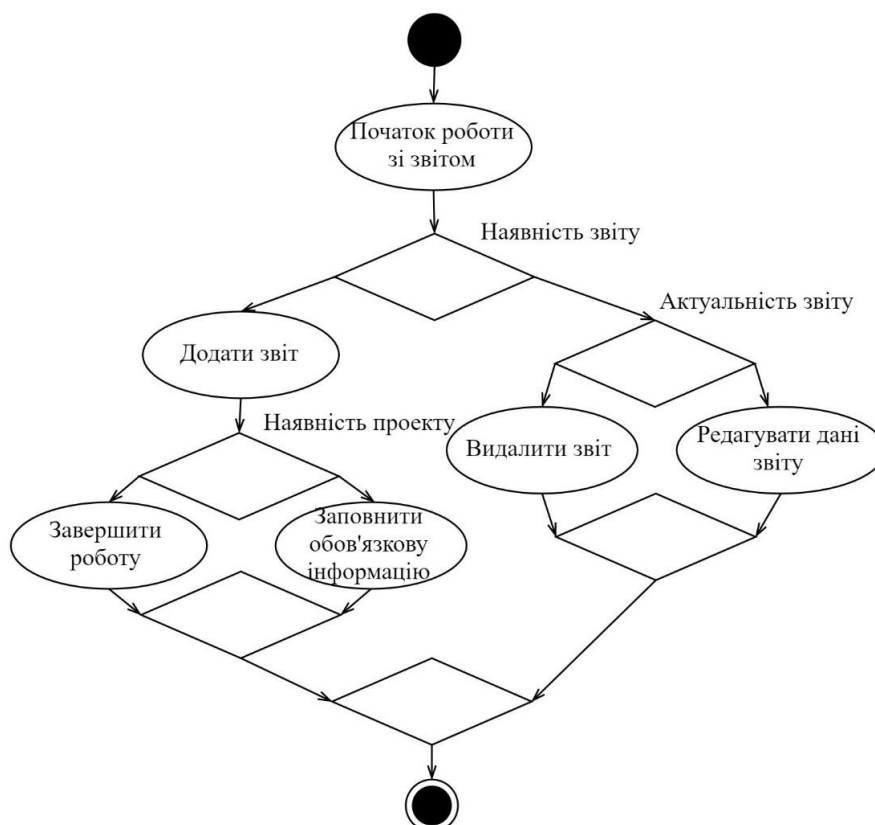


Рисунок 3.5 – Діаграма діяльності програмного забезпечення для ведення електронної звітності співробітників підприємства

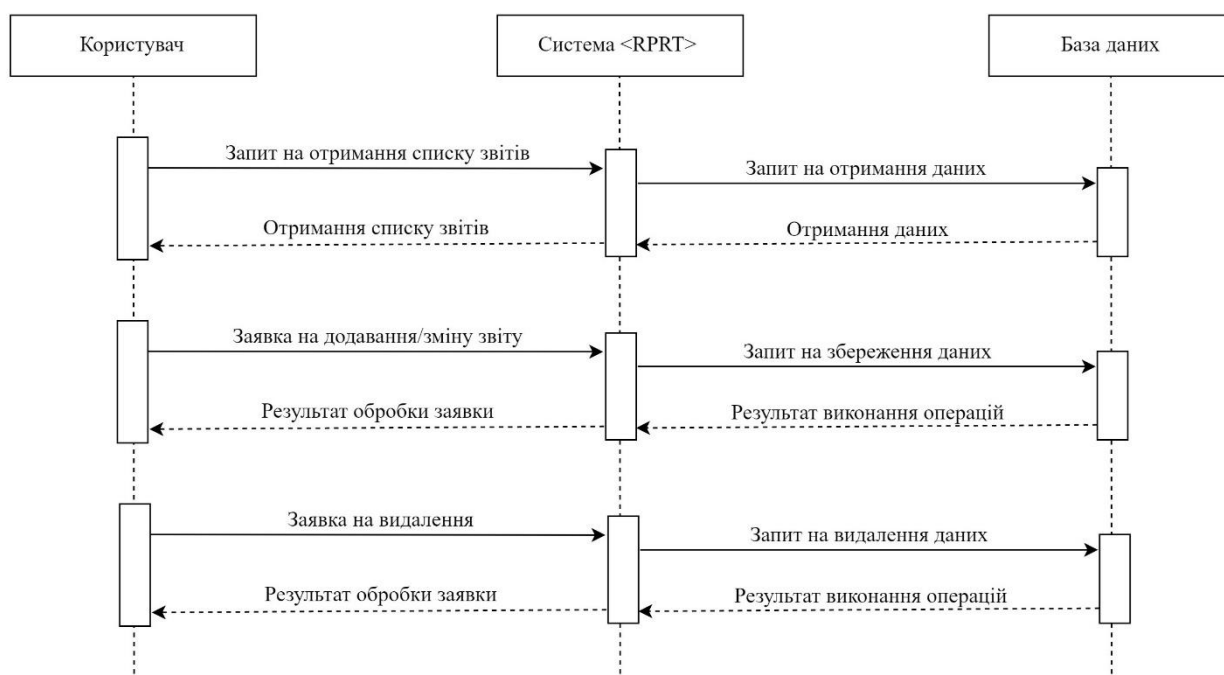


Рисунок 3.6 – Діаграма послідовностей програмного забезпечення для ведення електронної звітності співробітників підприємства

Діаграму кооперацій програмного забезпечення для ведення електронної звітності співробітників підприємства зображено на рис. 3.7.



Рисунок 3.7 – Діаграма кооперацій програмного забезпечення ведення електронної звітності співробітників підприємства

3.3 Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства виконано з урахуванням вимог технічного завдання.

При проєктуванні інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства враховувалися вимоги технічного завдання. Приклади прототипів розроблених інтерфейсних форм наведено на рис. 3.8-3.14.

Прототип головної сторінки програмного забезпечення для ведення електронної звітності співробітників підприємства наведено на рис. 3.8. На головній сторінці в кольоровій палітрі віддано перевагу зеленим відтінкам, які створюють атмосферу комфорту та спокою. На головній сторінці програмного забезпечення для ведення електронної звітності співробітників підприємства використовуються елементи: логотип, кнопки керування (прямокутники з закругленням) і таблиця з інформацією по звітам.

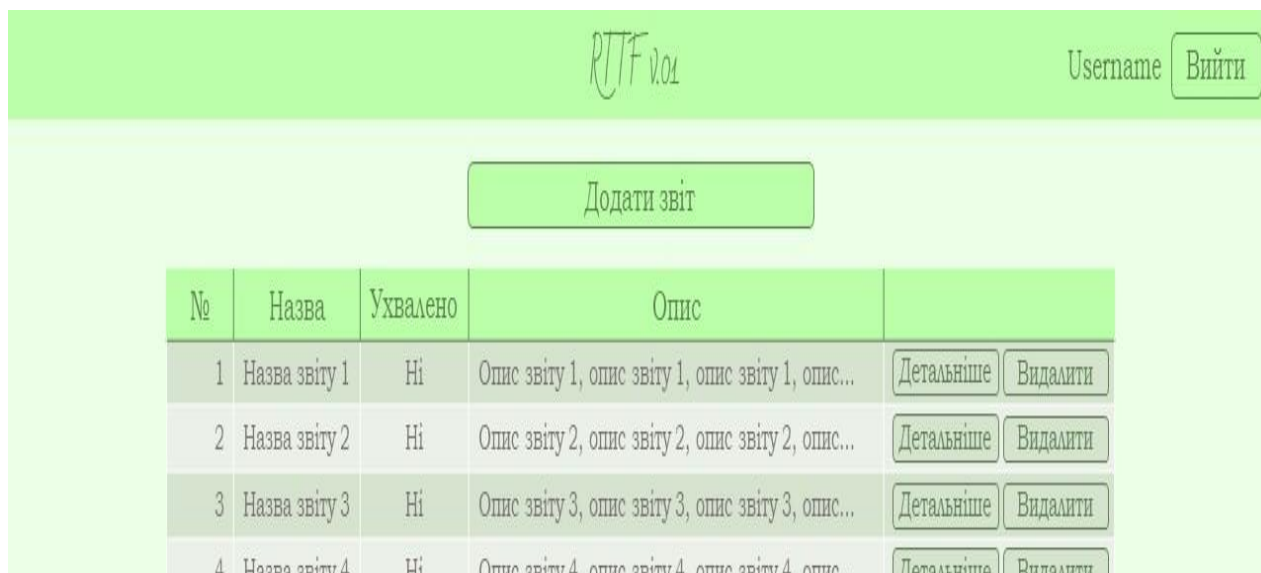


Рисунок 3.8 – Прототип головної сторінки програмного забезпечення для ведення електронної звітності співробітників підприємства

Розроблено макет сторінки авторизації користувача в системі (рис. 3.9). Щоб користувач пройшов авторизацію йому треба ввести логін, пароль і натиснути кнопку «Вхід». Відповідно макет сторінки авторизації містить поле для вводу логіну, паролю і кнопку керування, яка відповідає за «Вхід».

	Лого	
--	------	--

Текст	Логін	
Текст	Пароль	

Кнопка

Рисунок 3.9 – Макет сторінки авторизації

Макет сторінки звіту відображено на рис. 3.10. Ця сторінка відкривається після натискання користувачем на головній сторінки кнопки «Детальніше», відображає усю інформацію щодо звіту і надає можливість її

редагування, а саме: номер звіту, назва, опис, чи узгоджено його. Сторінка містить поле без можливості редагування: відображення номеру звіту, відображення статусу «чи узгоджено»; поля з можливістю редагування: назва звіту, опис звіту; кнопки керування «Повернутися» - перенаправлення до головної форми, «Зберегти зміни» - зберігання внесених змін і «Вихід» - для виходу з аккаунту.

	Лого	
		Вихід

Номер

Чи узгоджено

Текст

Назва звіту

Текст

Опис звіту

Повернутися

Зберегти зміни

Рисунок 3.10 – Макет сторінки звіту

Прототип адміністративної сторінки перегляду звітів програмного забезпечення для ведення електронної звітності наведено на рис. 3.11.

	Лого	
		Вихід

Звіти

Користувачі

Проекти

Архів

Номер	Назва	Чи узгоджено	Детальніше
Номер	Назва	Чи узгоджено	Детальніше
Номер	Назва	Чи	Детальніше

Рисунок 3.11 – Прототип адміністративної сторінки перегляду звітів

Ця сторінка відкривається після натискання адміністратором відповідної кнопки на адміністративній панелі, відображає номер звіту, його назву, статус «Чи узгоджено» і надає можливість переглянути більш детальну інформацію. Відповідно сторінка містить поле без можливості редагування: відображення номеру звіту, відображення назви звіту, відображення статусу «чи узгоджено», кнопку керування «Детальніше» - перенаправлення до перегляду більш детальної інформації про звіт.

Прототип адміністративної сторінки щодо перегляду інформації про користувачів наведено на рис. 3.12.

	Лого		Вихід
--	------	--	-------

Звіти	Користувачі	Проекти	Архів
-------	-------------	---------	-------

Номер	Username	Перегляд звітів	Новий пароль
Номер	Username	Перегляд звітів	Новий пароль
Номер	Username	Перегляд	Новий

Рисунок 3.12 – Прототип адміністративної сторінки щодо перегляду інформації про користувачів

Ця сторінка відкривається після натискання адміністратором відповідної кнопки на адміністративній панелі, відображає номер користувача, його username і надає можливість переглянути звіти користувача, згенерувати новий пароль. Сторінка містить поле без можливості редагування: відображення номеру користувача, відображення username, кнопки керування: «Перегляд звітів» - перенаправлення до перегляду звітів користувача, «Новий пароль» - генерування нового паролю користувача.

Прототип адміністративної сторінки перегляду проектів наведено на рис. 3.13.

	Лого	Вихід
--	------	--

Звіти	Користувачі	Проекти	Архів
-------	-------------	---------	-------

Номер	Назва	Опис	Детальніше	В архів
Номер	Назва	Опис	Детальніше	В архів
Номер	Назва	Опис	Детальніше	В архів

Рисунок 3.13 – Прототип адміністративної сторінки перегляду проектів

Адміністративна сторінка перегляду проектів відкривається після натискання адміністратором відповідної кнопки на адміністративній панелі, відображає номер проекту, його назву, початок опису і надає можливість переглянути більш детальну інформацію чи відправити до архіву. Сторінка містить поле без можливості редагування: відображення номеру проекту, назви, відображення початку опису, кнопки керування: «Детальніше» - перенаправлення до перегляду більш детальної інформації про проект, «В архів» - відправлення проекту до архіву.

Прототип адміністративної сторінки перегляду детальної інформації про проект наведено на рис. 3.14.

	Лого	Вихід
--	------	--

Звіти	Користувачі	Проекти	Архів
-------	-------------	---------	-------

Номер

Текст

Назва проекту

Текст

Опис

Повернутися

Зберегти зміни

Рисунок 3.14 – Прототип адміністративної сторінки перегляду детальної інформації про проект

Адміністративна сторінка перегляду детальної інформації про проект відкривається після натискання користувачем на адміністративній сторінці перегляду проектів кнопки «Детальніше», відображає усю інформацію щодо проектів і надає можливість її редагування. Ця сторінка містить поле без можливості редагування: відображення номеру проекту; поля з можливістю редагування: назва проекту, опис; кнопки керування «Повернутися» - перенаправлення до попередньої форми, «Зберегти зміни» - зберігання внесених змін.

3.4 Розширення функціональності програмного забезпечення для ведення електронної звітності співробітників підприємства за допомогою модулів

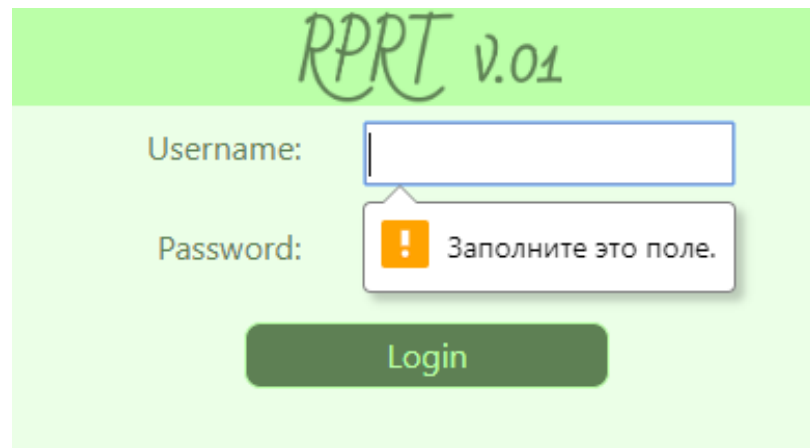
З метою реалізації функціональних вимог, визначених у технічному завданні до розробки програмного забезпечення для ведення електронної звітності співробітників підприємства, було використано модулі.

Для реалізації можливості розмежування користувачів на дві групи (звичайні користувачі, адміністратори) використано бібліотеку `django-contrib-auth`, яка надає набір готових форм, які можна використовувати для сторінки реєстрації користувачів та їх аутентифікації.

При розробці програмного забезпечення для ведення електронної звітності співробітників підприємства використано модулі `django.contrib.auth.forms.UserCreatForm` і `django.contrib.auth.forms.AuthenticationForm`, що зняло необхідність описувати обмеження на поля сторінок, такі як кількість літер в логіні, складність паролю та ін.

Форми забезпечують роботу попередження користувача про некоректні дані і не дозволяють системі обробляти дані, доки вони не будуть відповідати усім вимогам.

На рис. 3.15 наведено дію обмеження на поле, а саме – неприпустимість незаповненого поля.



The image shows a login interface for 'RPRT v.01'. It has a light green header with the text 'RPRT v.01'. Below the header, there are two input fields: 'Username:' and 'Password:'. The 'Username' field is empty. The 'Password' field is also empty, and a tooltip with an orange exclamation mark icon and the text 'Заполните это поле.' (Fill this field.) is pointing to it. Below the fields is a green 'Login' button.

Рисунок 3.15 – Приклад обробки проблемної ситуації

На рис. 3.16 наведено приклад обробки ситуації, пов'язаної з некоректним введенням даних.



The image shows the same login interface for 'RPRT v.01'. The 'Username' field now contains the text 'test'. The 'Password' field is empty. Below the fields, there is a red error message box that says: 'Please enter a correct username and password. Note that both fields may be case-sensitive.' Below the error message is a green 'Login' button.

Рисунок 3.16 – Приклад обробки ситуації,
пов'язаної з некоректним введенням даних

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для ведення електронної звітності співробітників підприємства. Запропоновано структуру програмного

забезпечення для ведення електронної звітності співробітників підприємства, яку подано за допомогою структурних діаграм.

Визначено, що основними класами програмного забезпечення для ведення електронної звітності співробітників підприємства є: користувач програми (User), проєкт (Project), звіт (Report).

Описано функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства за допомогою моделей на основі діаграм поведінки, діаграм кооперації та діаграм послідовності.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для ведення електронної звітності співробітників підприємства. Програма написана на мові Python, яка є мовою високого рівня, що використовується для розв'язання багатьох завдань, пов'язаних зі створенням програмних засобів, є досить простою та доступною, надає можливості розширення, має гнучкий синтаксис. В якості фреймворку для створення програми обрано Django, що забезпечує швидке створення програм, є простим та характеризується зручним дизайном, є безкоштовним та розповсюджується з відкритим вихідним кодом, а також містить велику кількість модулів, які можна використовувати для типових завдань розробки програмного забезпечення.

Програмне забезпечення для ведення електронної звітності співробітників підприємства забезпечує виконання таких функцій:

- можливість додавання звіту користувачем щодо виконаної роботи;
- можливість редагування наявного звіту;
- можливість видалення наявного звіту користувача;
- реєстрація нового користувача у програмному забезпеченні для ведення електронної звітності (функція адміністратора);
- можливість зміни облікових даних зареєстрованого користувача;
- можливість додавання нового проєкту для подальшого внесення звітів щодо його реалізації;
- можливість редагування інформації про наявні проєкти;
- можливість перегляду звітів користувачів (виконавців проєктів);
- можливість затвердження звітів користувачів.

Для експлуатації програмного забезпечення для ведення електронної звітності співробітників підприємства необхідно таке апаратне та програмне забезпечення:

- вільне місце на диску: 400 МБт;
- процесор Intel Pentium, 2 GHz чи потужніший;
- оперативна пам'ять об'ємом 256 Мб;
- веббраузер з оновленнями 2022 року
- функціонуюча операційна система;
- фреймворк Django.

Функціональні характеристики розробленого програмного забезпечення для ведення електронної звітності співробітників підприємства наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для ведення електронної звітності співробітників підприємства наведено у додатку Б.

4.2 Характеристики програмного забезпечення для ведення електронної звітності співробітників підприємства

Програмне забезпечення для п ведення електронної звітності співробітників підприємства може працювати безперервно.

Для роботи програмного забезпечення для ведення електронної звітності співробітників підприємства необхідна наявність встановленої операційної системи та фреймворку Django. Тому для безперебійного функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства необхідно розгорнути вебсервер, наприклад, Denwer, після чого налаштувати програму та базу даних застосунку.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Звертання користувачів до програмного забезпечення для ведення електронної звітності співробітників підприємства здійснюється стандартними засобами операційної системи.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для ведення електронної звітності співробітників підприємства є інформація про користувача, введений користувачем звіт, інформація про звіт, який користувач хоче відредагувати.

Вихідними даними програмного забезпечення для ведення електронної звітності співробітників підприємства є створений або відредагований звіт.

4.3.3 Повідомлення

Користувач програмного забезпечення для ведення електронної звітності співробітників підприємства може отримати різні повідомлення, зокрема повідомлення про необхідність заповнення певного поля, повідомлення про некоректність введених даних, повідомлення про успішне створення звіту.

4.4 Виконання програмного забезпечення для ведення електронної звітності співробітників підприємства

Після запуску програмного забезпечення для ведення електронної звітності співробітників підприємства користувачу виводиться форма для входу в систему (рис. 4.1).



Рисунок 4.1 – Інтерфейс форми для входу у програмне забезпечення для ведення електронної звітності співробітників підприємства

Після входу у систему користувачу відкривається головна сторінка програмного забезпечення для ведення електронної звітності співробітників підприємства (рис. 4.2). Ця сторінка призначена для авторизованих користувачів, які мають намір звітувати щодо виконаної роботи, і відкривається після авторизації користувача.

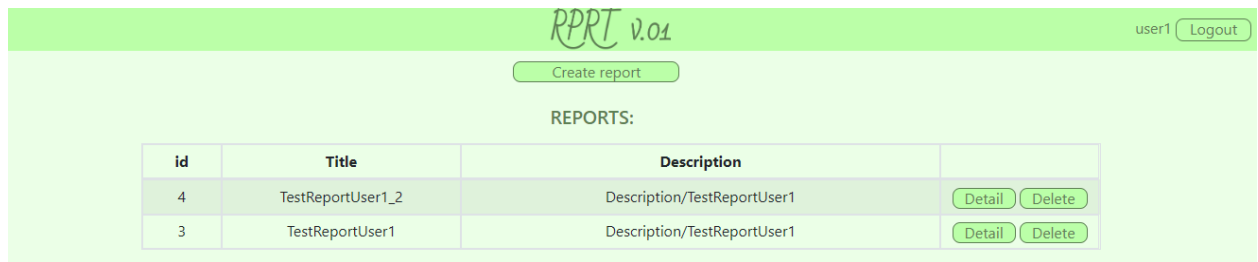


Рисунок 4.2 – Головна сторінка програмного забезпечення для ведення електронної звітності співробітників підприємства

Користувачам на головній сторінці доступно додавання звіту, перегляд стислої інформації про існуючі звіти, перегляд детальної інформації за існуючим звітом, видалення існуючих звітів.

При необхідності створення звіту користувач натискає відповідну кнопку на головній сторінці, внаслідок чого відкривається форма створення звіту (рис. 4.3).

RPRT v.0.1 user1 Logout

Back to reports

ADD REPORT:

Title:

Description:

Hours:

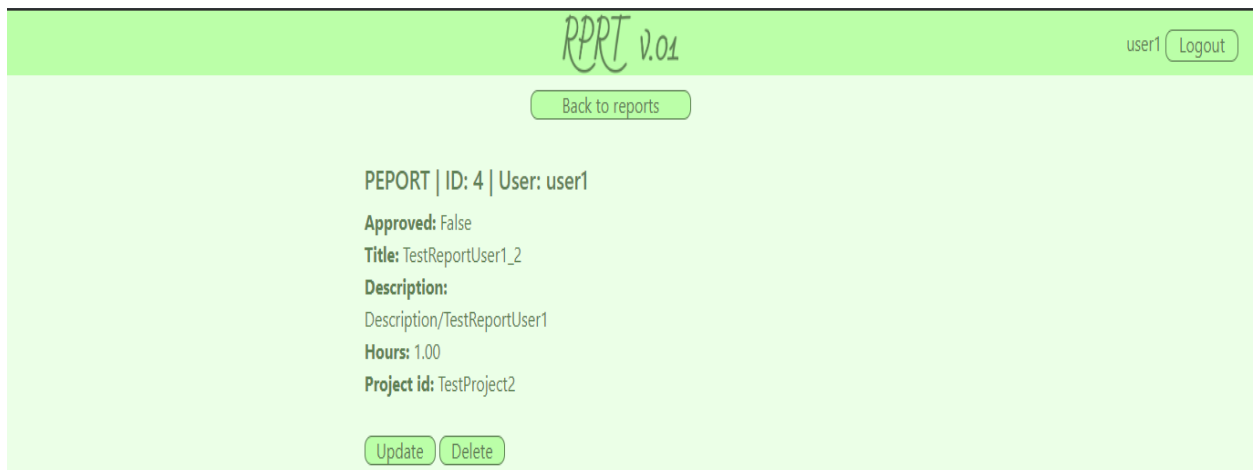
Project:

Approved: ☐ False

Create report

Рисунок 4.3 – Форма для створення звіту

Користувач має можливість переглянути детальну інформацію про звіт (рис. 4.4), натиснувши кнопку Detail на головній формі. Ця сторінка відображає усю інформацію щодо звіту і надає можливість її редагування, а саме: назва звіту, опис звіту, кількість годин на роботу, чи узгоджено звіт та ін. Сторінка містить поле без можливості редагування: відображення номеру звіту, відображення статусу «чи узгоджено».



RPRT v.0.1 user1 Logout

Back to reports

REPORT | ID: 4 | User: user1

Approved: False

Title: TestReportUser1_2

Description: Description/TestReportUser1

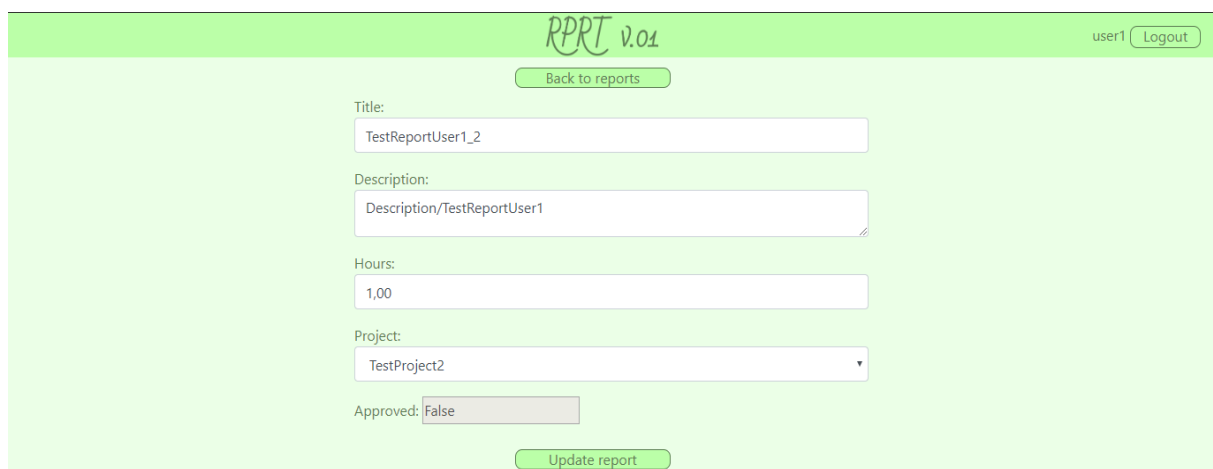
Hours: 1.00

Project id: TestProject2

Update Delete

Рисунок 4.4 – Форма для перегляду детальної інформації про звіт

При натисканні кнопки Update користувач має можливість оновлення звіту (рис. 4.5). Зокрема, користувач може ввести нові дані щодо назви звіту, його опису, кількості відпрацьованих годин та ін.



RPRT v.0.1 user1 Logout

Back to reports

Title: TestReportUser1_2

Description: Description/TestReportUser1

Hours: 1,00

Project: TestProject2

Approved: False

Update report

Рисунок 4.5 – Форма для оновлення звіту

На рисунку 4.6 наведено форму для видалення звіту.

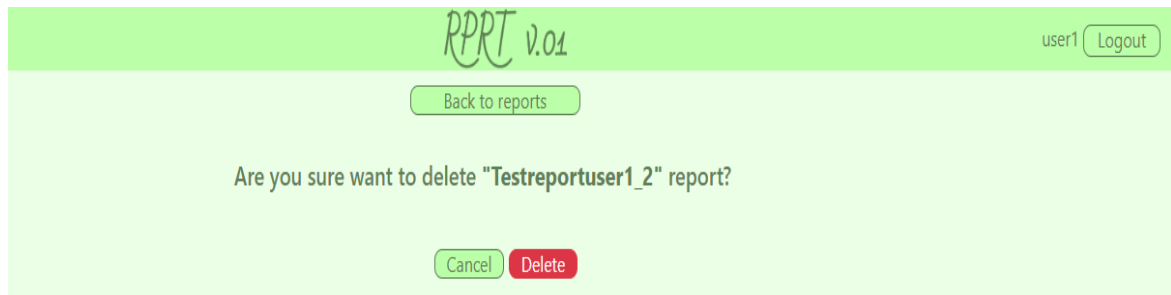


Рисунок 4.6 – Інтерфейс форми для видалення звіту

Адміністратор програмного забезпечення для ведення електронної звітності співробітників підприємства має можливості перегляду звітів та їх узгодження, а також можливості роботи з інформацією про користувачів, зокрема, додавання нових користувачів.

Також після натискання адміністратором відповідної кнопки на адміністративній панелі відображається відповідна форма з детальною інформацією про звіт, де відображено номер звіту, його назву, статус «Чи узгоджено» і надає можливість переглянути більш детальну інформацію. Відповідно сторінка містить поле без можливості редагування: відображення номеру звіту, відображення назви звіту, відображення статусу «чи узгоджено», кнопку керування «Детальніше» - перенаправлення до перегляду більш детальної інформації про звіт.

На рисунку 4.7 наведено інтерфейс форми для погодження звіту.

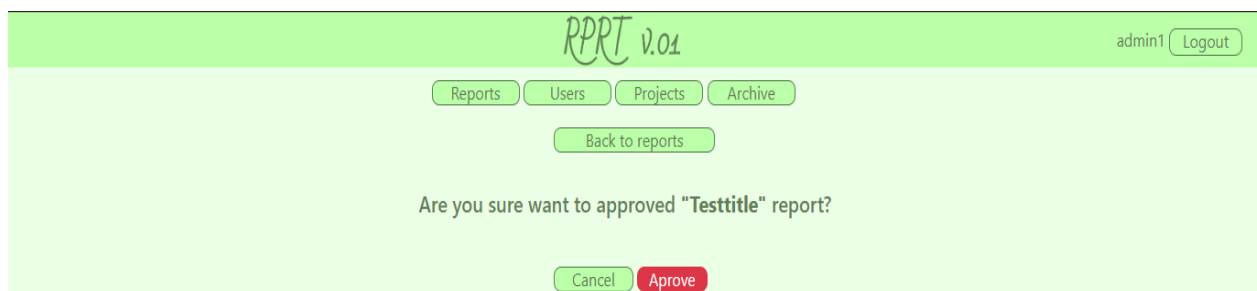


Рисунок 4.7 – Інтерфейс форми для погодження звіту

У випадку необхідності додавання нового користувача адміністратор вводить необхідну інформацію на відповідній формі (рис. 4.8)

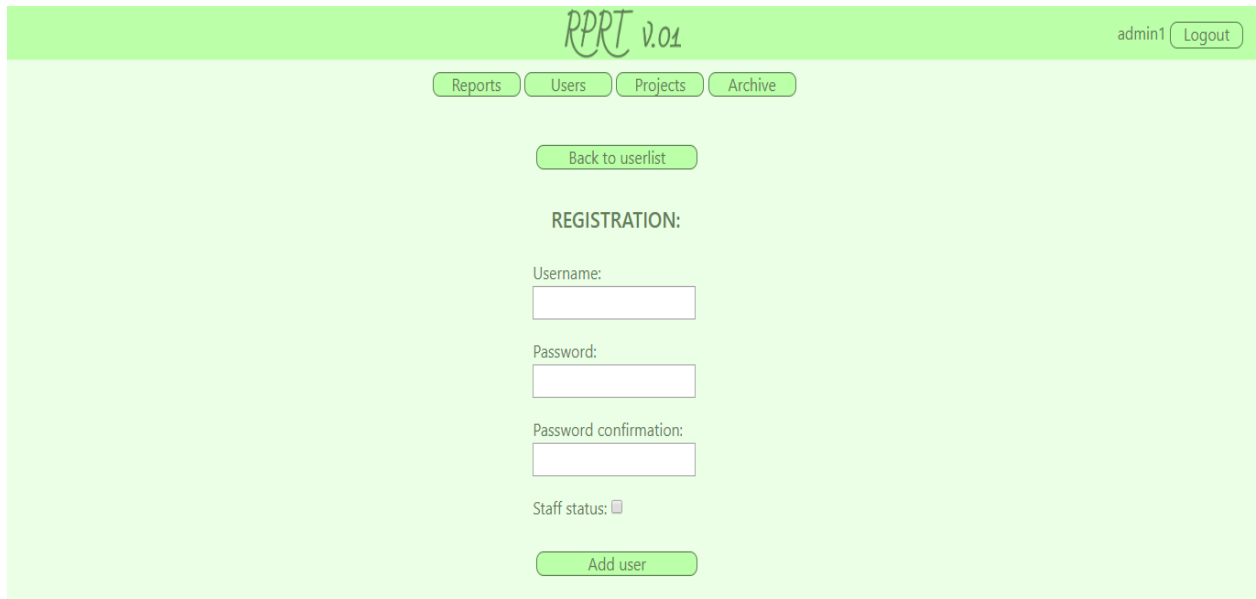


Рисунок 4.8 – Інтерфейс форми для реєстрації користувачів

4.5 Тестування програмного забезпечення для ведення електронної звітності співробітників підприємства

Розроблене програмного забезпечення для ведення електронної звітності співробітників підприємства базується кросплатформенних засобах, тому було виконано перевірку роботи програму на різних операційних системах.

У процесі тестування програмного забезпечення для ведення електронної звітності співробітників підприємства виконано перевірку відповідності функціональних вимог:

- можливість додавання звіту користувачем щодо виконаної роботи;
- можливість редагування наявного звіту;
- можливість видалення наявного звіту користувача;
- реєстрація нового користувача у програмному забезпеченні для ведення електронної звітності (функція адміністратора);

- можливість зміни облікових даних зареєстрованого користувача;
- можливість додавання нового проєкту для подальшого внесення звітів щодо його реалізації;
- можливість редагування інформації про наявні проєкти;
- можливість перегляду звітів користувачів (виконавців проєктів);
- можливість затвердження звітів користувачів.

Також було перевірено інші вимоги ТЗ до розробки програмного забезпечення для ведення електронної звітності співробітників підприємства.

За результатами тестування встановлено, що програмне забезпечення для ведення електронної звітності співробітників підприємства відповідає технічному завданню.

4.6 Висновки за розділом 4

Описано програмне забезпечення для ведення електронної звітності співробітників підприємства. Виконано тестування розробленого програмного забезпечення для ведення електронної звітності співробітників підприємства. Результати тестування програмного забезпечення для ведення електронної звітності співробітників підприємства показали, що розроблена програма може використовуватися за призначенням та відповідає технічному завданню.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для ведення електронної звітності.

Відзначено, що електронна звітність стає необхідною для підприємств у сучасному світі, де швидкість, точність та доступність інформації є ключовими чинниками успіху. Показано, що програмні засоби електронної звітності дозволяють збирати та зберігати дані про співробітників, а також результати їх діяльності. Відзначено, що програмні системи підтримки процесу подання та збору електронної звітності забезпечують легкість доступу та обробки відповідних даних з будь-якого місця, що полегшує роботу фахівців з управління персоналом. Автоматизована обробка цих даних також допомагає уникнути помилок, що можуть виникнути при ручній обробці. Програмні засоби ведення електронної звітності співробітників підприємства дозволяють заощадити час та ресурси компанії, оскільки електронна звітність дозволяє значно скоротити час, який потрібен для обробки та аналізу даних. Показано, що такі програмні засоби також зменшують потребу у паперовій документації та фізичному зберіганні даних, крім того програмні засоби ведення електронної звітності співробітників підприємства допомагають виконувати вимоги стосовно зберігання та обробки особистих даних співробітників, що є важливим аспектом забезпечення відповідності вимогам сучасного законодавства про захист персональних даних. Програмні засоби ведення електронної звітності співробітників підприємства надають засоби аналітики та стратегічного управління: збір та аналіз даних про персонал дозволяє підприємствам розробляти стратегії управління персоналом, виявляти тенденції та удосконалювати процеси найму, зберігання та розвитку кадрів підприємства.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для ведення електронної звітності.

Проте інтерфейс деяких програмних засобів підтримки електронної звітності може бути складним для розуміння або використання, що впливає на продуктивність та швидкість роботи. Деякі програмні засоби електронної звітності можуть бути складними для інтеграції з іншими програмними системами підприємства, що може ускладнити обмін даними або їхню взаємодію. Крім того, якщо програмне забезпечення для електронної звітності базується на певних технологіях, воно може стати застарілим у майбутньому, вимагаючи постійного оновлення або переходу на нові платформи, що може бути витратним, що обумовлює необхідність своєчасного оновлення відповідного програмного забезпечення шляхом модифікації існуючих програмних модулів або розробки нового програмного забезпечення. Тому актуальною є розробка програмного забезпечення для ведення електронної звітності співробітників підприємства.

Сформульовано функціональні вимоги до програмного забезпечення для ведення електронної звітності співробітників підприємства.

Для реалізації програмного забезпечення для ведення електронної звітності співробітників підприємства обрано мову програмування Python, яка є мовою високого рівня, що використовується для розв'язання багатьох завдань, пов'язаних зі створенням програмних засобів, є досить простою та доступною, надає можливості розширення, має гнучкий синтаксис. Крім того, ця мова програмування використовується в різних сферах, включаючи веброботку, аналіз даних, штучний інтелект та багато інших галузей, що забезпечує можливість інтеграції розробленої програми в інші програмні продукти.

Для створення програмного забезпечення для ведення електронної звітності співробітників підприємства обрано фреймворк Django, що забезпечує швидке створення програм, є простим та характеризується зручним дизайном. Фреймворк Django спрощує процес веброботки, він є безкоштовним та розповсюджується з відкритим вихідним кодом, а також

містить велику кількість модулів, які можна використовувати для типових завдань розробки програмного забезпечення.

Новизна роботи полягає в тому, що створено модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для ведення електронної звітності співробітників підприємства.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення для ведення електронної звітності співробітників підприємства.

Запропоновано структуру програмного забезпечення для ведення електронної звітності співробітників підприємства, яку подано за допомогою структурних діаграм.

Визначено, що основними класами програмного забезпечення для ведення електронної звітності співробітників підприємства є: користувач програми (User), проєкт (Project), звіт (Report).

Описано функціонування програмного забезпечення для ведення електронної звітності співробітників підприємства за допомогою моделей на основі діаграм поведінки, діаграм кооперації та діаграм послідовності.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства.

Виконано тестування розробленого програмного забезпечення для ведення електронної звітності співробітників підприємства. Результати тестування програмного забезпечення для ведення електронної звітності співробітників підприємства показали, що розроблена програма може використовуватися за призначенням та відповідає технічному завданню.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is e-reporting [Electronic resource]. – Access mode : <https://complyplatform.com/faq/what-is-e-reporting/>.
2. Electronic Reporting [Electronic resource]. – Access mode : <https://www.esma.europa.eu/issuuer-disclosure/electronic-reporting>.
3. Navigating E-Invoicing Standards in Europe: A Comprehensive Guide [Electronic resource]. – Access mode : <https://www.storecove.com/blog/en/digital-vat-e-reporting-and-e-invoicing-eu/>.
4. Electronic reporting (ER) overview [Electronic resource]. – Access mode : <https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/analytics/general-electronic-reporting>.
5. Introduction to Electronic Reporting [Electronic resource]. – Access mode : <https://dynamicsmanuals.com/2019/09/12/introduction-to-electronic-reporting/>.
6. Introduction to Electronic Case Reporting [Electronic resource]. – Access mode : <https://ecr.aimsplatform.org/cms/resources/blocks/introduction-to-electronic-case-reportingfor-ehr-and-health-it-vendors.pdf>.
7. Factors Affecting the Adoption of Electronic Data Reporting [Electronic resource]. – Access mode : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8664162/>.
8. 15 Best Electronic Document Management Systems in 2023 [Electronic resource]. – Access mode : <https://financesonline.com/best-electronic-document-management-systems/>.
9. The best document management software in 2023 [Electronic resource]. – Access mode : <https://zapier.com/blog/best-document-management-software/>.
10. The Best Document Management Software [Electronic resource]. – Access mode : <https://www.pcmag.com/picks/the-best-document-management-software>.

11. Document Management Software [Electronic resource]. – Access mode : <https://www.capterra.com/document-management-software/>.
12. Best Document Management Software of 2023 [Electronic resource]. – Access mode : <https://www.techrepublic.com/article/document-management-software/>.
13. Bitrix24. Your ultimate workspace [Electronic resource]. – Access mode : <https://www.bitrix24.com/>.
14. Jira [Electronic resource]. – Access mode : <https://www.atlassian.com/de/software/jira>.
15. Mantis Bug Tracker [Electronic resource]. – Access mode : Mantis Bug Tracker.
16. Learn Python [Electronic resource]. – Access mode : <https://www.freecodecamp.org/news/learn-python-free-python-courses-for-beginners/>.
17. Historia de Python [Electronic resource]. – Access mode : https://es.wikipedia.org/wiki/Historia_de_Python.
18. Django makes it easier to build better web apps more quickly and with less code [Electronic resource]. – Access mode : <https://www.djangoproject.com/>.
19. Django Tutorial [Electronic resource]. – Access mode : <https://www.w3schools.com/django/index.php>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу ведення та обробки електронної звітності.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів ведення електронної звітності співробітників підприємства», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

A.2 Призначення розробки

Програмний продукт призначений для ведення електронної звітності співробітників підприємства.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для ведення електронної звітності співробітників підприємства забезпечує виконання таких функцій:

- можливість додавання звіту користувачем щодо виконаної роботи;
- можливість редагування наявного звіту;
- можливість видалення наявного звіту користувача;
- реєстрація нового користувача у програмному забезпеченні для ведення електронної звітності (функція адміністратора);
- можливість зміни облікових даних зареєстрованого користувача;
- можливість додавання нового проєкту для подальшого внесення звітів щодо його реалізації;

- можливість редагування інформації про наявні проекти;
- можливість перегляду звітів користувачів (виконавців проєктів);
- можливість затвердження звітів користувачів.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для ведення електронної звітності співробітників підприємства повинен бути зручним для користувачів.

A.3.3 Вимоги до надійності

Програмне забезпечення для ведення електронної звітності співробітників підприємства повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для ведення електронної звітності співробітників підприємства необхідна наявність персонального комп'ютера у користувача.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для ведення електронної звітності співробітників підприємства необхідно таке апаратне та програмне забезпечення:

- вільне місце на диску: 400 МБт;
- процесор Intel Pentium, 2 GHz чи потужніший;
- оперативна пам'ять об'ємом 256 Мб;

- веббраузер з оновленнями 2022 року
- функціонуюча операційна система;
- фреймворк Django.

A.3.6 Вимоги до маркування і пакування

Програма для ведення електронної звітності співробітників підприємства може бути записана на будь-якому носії інформації. На пакуванні повинна бути назва програми – «Програмне забезпечення для ведення електронної звітності співробітників підприємства».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для ведення електронної звітності співробітників підприємства є інформація про користувача, введений користувачем звіт, інформація про звіт, який користувач хоче відредагувати.

Вихідними даними програмного забезпечення для ведення електронної звітності співробітників підприємства є створений або відредагований звіт.

ДОДАТОК Б
Фрагмент тексту програми

```

class Prjctlst(Bjlstx, View):
    mdl1 = Prjct
    tpl1 = 'prjct/prjct_list.html'
    rsexp = True

class PrjctCreate(Bhcrtgh, View):
    mdfrm = PrjctFrm
    tpl1 = 'prjct/prjct_create.html'
    rsexp = True

class PrjctUpdate(ObjectUpdateMixin, View):
    mdl1 = Prjct
    mdl1_frm = PrjctFrm
    tpl1 = 'prjct/prjct_update.html'
    rsexp = True

class PrjctDetail(ObjectDetailMixin, View):
    mdl1 = Prjct
    tpl1 = 'prjct/prjct_detail.html'
    rsexp = True

class PrjctArchive(Btgfw, View):
    mdl1 = Prjct
    mdl1_arch = PrjctArchive
    tpl1 = 'prjct/prjct_archive.html'
    rdr1 = 'prjcts_list_url'
    rsexp = True

class Prjct(mdlls.Mdll):
    tlt = mdlls.CharField(maxlengt=50, unique=True)
    dscrt = mdlls.TextField(maxlengt=150)

```

```

class PrjctFrm(frms.MdllFrm):
    class Meta:
        mdl1 = Prjct
        fds = ['id','tlt', 'dscrt']
        wds = {
            'id': frms.TextInput(attrs={'class': 'frm-control'}),
            'tlt': frms.TextInput(attrs={'class': 'frm-control'}),
            'dscrt': frms.Textarea(attrs={'class': 'frm-control'}),
        }

class Bjlstx:
    mdl1 = None
    tpl = None

    def get(self, request):
        gntd = Gntd(obj, 200)
        rtghnmbt = request.GET.get('rtgh', 1)
        rtgh = gntd.get_rtgh(rtghnmbt)
        ispgnr = rtgh.has_other_rtghs()
        if rtgh.has_previous():
            prrg = '?rtgh={}'.format(rtgh.previous_rtghnmbt())
        else:
            prrg = ''

        if rtgh.has_next():
            ntrl = '?rtgh={}'.format(rtgh.next_rtghnmbt())
        else:
            ntrl = ''

        context = {
            'rtgh_object': rtgh,
            'ispgnr': ispgnr,
            'ntrl': ntrl,

```

```

        'prrg': prrg,

    }

    return rndf(request, self.tmpl,
                context=context)

class Bhcrtgh:
    mdfrm = None
    tmpl = None

    def get(self, request):
        if not request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))
        frm = self.mdfrm()
        return rndf(request, self.tmpl, context={'frm': frm})

    def post(self, request):
        if not request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))
        bound_frm = self.mdfrm(request.POST)
        if bound_frm.is_valid():
            new_obj = bound_frm.save()
            return redirect(new_obj)
        return rndf(request, self.tmpl, context={'frm':
bound_frm}))

class ObjectDetailMixin:
    mdl1 = None
    tmpl = None

class ObjectUpdateMixin:
    mdl1 = None
    mdl1_frm = None
    tmpl = None

```

```

def get(self, request, id):
    if not request.krstv.sdrtd:
        return HttpResponseRedirect(reverse(crt_rtgh_url'))
    obj = self.mdl1.objects.get(id__iexact=id)
    bound_frm = self.mdl1_frm(instance=obj)
    return rndf(request, self.tmpl, context={'frm':
bound_frm, self.mdl1.__name__.lower(): obj})

```

```

def post(self, request, id):
    if not request.krstv.sdrtd:
        return HttpResponseRedirect(reverse(crt_rtgh_url'))
    obj = self.mdl1.objects.get(id__iexact=id)
    bound_frm = self.mdl1_frm(request.POST, instance=obj)
    if bound_frm.is_valid():
        new_obj = bound_frm.save()
        return redirect(new_obj)
    return rndf(request, self.tmpl,
context={'frm': bound_frm,
self.mdl1.__name__.lower(): obj})

```

```

class Btgfw:

```

```

    mdl1 = None
    tmpl = None
    rdr1 = None
    mdl1_arch = None

```

```

def get(self, request, id):
    if not request.krstv.sdrtd:
        return HttpResponseRedirect(reverse(crt_rtgh_url'))
    obj = self.mdl1.objects.get(id__iexact=id)
    return rndf(request, self.tmpl,
context={self.mdl1.__name__.lower(): obj})

```

```

def post(self, request, id):

```

```

        if not request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))
        obj = self.mdll.objects.get(id__iexact=id)
        obj_arch = self.mdll_arch(id_prjct=obj.id, tlt=obj.tlt,
dscrt=obj.dscrt)
        obj_arch.save()
        obj.delete()
        return redirect(reverse(self.rdr1))

class ZvitList(Bjlstx, View):
    mdll = Zvit
    tpl = 'zvit/zvits_list.html'
    rsexp = True

class ZvitCreate(Bhcrtgh, View):
    mdfrm = ZvitFrm
    tpl = 'zvit/zvit_create.html'
    rsexp = True

class ZvitUpdate(ObjectUpdateMixin, View):
    mdll = Zvit
    mdll_frm = ZvitFrm
    tpl = 'zvit/zvit_update.html'
    rsexp = True

class ZvitDetail(ObjectDetailMixin, View):
    mdll = Zvit
    tpl = 'zvit/zvit_detail.html'
    rsexp = True
    readonly_fds = ('utvd')

class ZvitDelete(ObjectDeleteMixin, View):
    mdll = Zvit
    tpl = 'zvit/zvit_delete.html'
    rdr1 = 'zvits_list_url'
    rsexp = True

```

```

class ReporApprove(Bjprvmxn, View):
    tpl = 'zvit/zvit_approve.html'
    rdrl = 'zvits_list_url'
    mdl1 = Zvit
    mdl1_frm = ZvitFrm
    rsexp = True

def get_sentinel_prjct():
    return Prjct.objects.get_or_create(tlt='deleted')[0]

class Zvit(mdlls.Mdll):
    tlt = mdlls.CharField(maxlengt=50, unique=True)
    dscrt = mdlls.TextField(maxlengt=150)
    hours = mdlls.DecimalField(max_digits=8, decimal_places=2,
validators=[MinValueValidator(0.0)])
    utvd = mdlls.BooleanField(default=False)
    prjct = mdlls.ForeignKey(Prjct, on_delete=mdlls.SET_NULL,
null=True, blank=True)
    krstv = mdlls.ForeignKey(Krstv, on_delete=mdlls.DO_NOTHING,
null=True, blank=True)

class ZvitFrm(frms.MdllFrm):
    class Meta:
        mdl1 = Zvit

        fds = ['tlt', 'dscrt', 'hours', 'prjct', 'utvd']
        wds = {
            'tlt': frms.TextInput(attrs={'class': 'frm-
control'}),
            'dscrt': frms.Textarea(attrs={'class': 'frm-
control'}),
            'hours': frms.NumberInput(attrs={'class': 'frm-
control'}),
            'prjct': frms.Select(attrs={'class': 'frm-control'}),

```

```

        'utvd': frms.TextInput(attrs={'disabled':
'disabled'}}),
    }

```

```

class Bjlstx:

```

```

    mdl1 = None

```

```

    tpl = None

```

```

    def get(self, request, id_krstv=-1):

```

```

        if not request.krstv.santf:

```

```

            return

```

```

        HttpResponseRedirect(reverse('krstv_login_url'))

```

```

        if request.krstv.sdrtd:

```

```

            if id_krstv == -1:

```

```

                obj = self.mdl1.objects.all()

```

```

            else:

```

```

                obj

```

```

            =

```

```

        self.mdl1.objects.all().filter(krstv=id_krstv)

```

```

    else:

```

```

        obj = self.mdl1.objects.all().filter(utvd=False,

```

```

        krstv=self.request.krstv)

```

```

        gntd = Gntd(obj, 200)

```

```

        rtghnmbt = request.GET.get('rtgh', 1)

```

```

        rtgh = gntd.get_rtgh(rtghnmbt)

```

```

        ispgnr = rtgh.has_other_rtghs()

```

```

        if rtgh.has_previous():

```

```

            prrg = '?rtgh={}'.format(rtgh.previous_rtghnmbt())

```

```

        else:

```

```

            prrg = ''

```

```

        if rtgh.has_next():

```

```

            ntrl = '?rtgh={}'.format(rtgh.next_rtghnmbt())

```

```

        else:

```

```

            ntrl = ''

```

```

        context = {
            'rtgh_object': rtgh,
            'ispgnr': ispgnr,
            'ntrl': ntrl,
            'prrg': prrg,
        }
        return rndf(request, self.tmpl,
                    context=context)

class Bhcrtgh:
    mdfrm = None
    tmpl = None

    def get(self, request):
        if not request.krstv.santf:
            return
HttpResponseRedirect(reverse('krstv_login_url'))
        if request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))
        frm = self.mdfrm()
        return rndf(request, self.tmpl, context={'frm': frm})

    def post(self, request):
        if not request.krstv.santf:
            return
HttpResponseRedirect(reverse('krstv_login_url'))
        if request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))

        bound_frm = self.mdfrm(request.POST)
        if bound_frm.is_valid():
            new_obj = bound_frm.save(commit=False)
            new_obj.krstv = self.request.krstv
            new_obj.save()
            return redirect(new_obj)

```

```

        return rndf(request, self.tmpl, context={'frm':
bound_frm})

```

```

class ObjectDetailMixin:

```

```

    mdl1 = None

```

```

    tmpl = None

```

```

    mdl1_frm = None

```

```

    def get(self, request, id):

```

```

        if not request.krstv.santf:

```

```

            return

```

```

        HttpResponseRedirect(reverse('krstv_login_url'))

```

```

        obj = get_object_or_404(self.mdl1, id__iexact=id)

```

```

        return rndf(request, self.tmpl,

```

```

                        context={self.mdl1.__name__.lower(): obj,

```

```

'krstv_object': obj})

```

```

class ObjectUpdateMixin:

```

```

    mdl1 = None

```

```

    mdl1_frm = None

```

```

    tmpl = None

```

```

    def get(self, request, id):

```

```

        if not request.krstv.santf:

```

```

            return

```

```

        HttpResponseRedirect(reverse('krstv_login_url'))

```

```

        if request.krstv.sdrtd:

```

```

            return

```

```

        HttpResponseRedirect(reverse('zvits_list_url'))

```

```

        obj = self.mdl1.objects.get(id__iexact=id)

```

```

        if obj.utvd == True:

```

```

            return

```

```

        HttpResponseRedirect(reverse('zvits_list_url'))

```

```

        bound_frm = self.mdl1_frm(instance=obj)
        return rndf(request, self.tmpl, context={'frm':
bound_frm, self.mdl1.__name__.lower(): obj})

    def post(self, request, id):
        if not request.krstv.santf:
            return
HttpResponseRedirect(reverse('krstv_login_url'))
        if request.krstv.sdrtd:
            return
HttpResponseRedirect(reverse('zvits_list_url'))
        obj = self.mdl1.objects.get(id__iexact=id)

        if obj.utvd == True:
            return
HttpResponseRedirect(reverse('zvits_list_url'))

        bound_frm = self.mdl1_frm(request.POST, instance=obj)

        if bound_frm.is_valid():
            new_obj = bound_frm.save()
            return redirect(new_obj)
        return rndf(request, self.tmpl,
                    context={'frm': bound_frm,
self.mdl1.__name__.lower(): obj})

class ObjectDeleteMixin:
    mdl1 = None
    tmpl = None
    rdrl = None

    def post(self, request, id):
        if not request.krstv.santf:
            return
HttpResponseRedirect(reverse('krstv_login_url'))

```

```

        if request.krstv.sdrtd:
            return
    HttpResponseRedirect(reverse('zvits_list_url'))
    obj = self.mdl1.objects.get(id__iexact=id)

    if obj.utvd == True:
        return
    HttpResponseRedirect(reverse('zvits_list_url'))

    obj.delete()
    return redirect(reverse(self.rdrl))

class Bjprvmxn:
    mdl1 = None
    mdl1_frm = None
    tpl = None

    def get(self, request, id):
        if not request.krstv.santf:
            return
    HttpResponseRedirect(reverse('krstv_login_url'))
    if not request.krstv.sdrtd:
        return
    HttpResponseRedirect(reverse('zvits_list_url'))

    obj = self.mdl1.objects.get(id__iexact=id)
    if obj.utvd == True:
        return
    HttpResponseRedirect(reverse('zvits_list_url'))

    return HttpResponseRedirect(reverse('zvits_list_url'))

class KrstvLogin(FrmView):
    frm_class = AuthenticationFrm
    tpl_name = 'krstv/login.html'

```

```

usprl = '/'

def post(self, request, *args, **kwargs):
    if request.krstv.santf:
        return HttpResponseRedirect(reverse('crt_rtgh_url'))
    return super().post(request, *args, **kwargs)

def get(self, request, *args, **kwargs):
    if request.krstv.santf:
        return HttpResponseRedirect(reverse('crt_rtgh_url'))
    return super().post(request, *args, **kwargs)

def frm_valid(self, frm):
    self.krstv = frm.get_krstv()
    login(self.request, self.krstv)
    return super(KrstvLogin, self).frm_valid(frm)

def frm_invalid(self, frm):
    frm.error = ('\n'.join(frm.get_invalid_login_error()))
    return super(KrstvLogin, self).frm_invalid(frm)

class KrstvList(Bjlstx, View):
    mdl1 = Krstv
    tpl1 = 'krstv/krstv_list.html'
    rsexp = True

class KrstvLogout(View):
    def get(self, request):
        logout(request)
        return HttpResponseRedirect(reverse('krstv_login_url'))

class KrstvRegistration(FrmView):
    frm_class = ExtendedKrstvCreationFrm

```

```

usprl = '/krstv/'
tpl_name = 'krstv/registration.html'

def frm_valid(self, frm):
    frm.save()
    return super(KrstvRegistration, self).frm_valid(frm)

def frm_invalid(self, frm):
    return super(KrstvRegistration, self).frm_invalid(frm)

class ExtendedKrstvCreationFrm(KrstvCreationFrm):
    class Meta:
        mdl1 = Krstv
        fds = ('krstvname', 'swrg1', 'swrg2', 'sdrtd')

class Bjlstx:
    mdl1 = None
    tpl = None

    def get(self, request):
        if not request.krstv.sdrtd:
            return HttpResponseRedirect(reverse('crt_rtgh_url'))
# redirect('crt_rtgh_url', permanent=True)
        obj = self.mdl1.objects.all()

        gntd = Gntd(obj, 200)
        rtghnmbt = request.GET.get('rtgh', 1)
        rtgh = gntd.get_rtgh(rtghnmbt)
        ispgnr = rtgh.has_other_rtghs()
        if rtgh.has_previous():
            prrg = '?rtgh={}'.format(rtgh.previous_rtghnmbt())
        else:
            prrg = ''

        if rtgh.has_next():

```

```
        ntrl = '?rtgh={}'.format(rtgh.next_rtghnmbt())
    else:
        ntrl = ''

    context = {
        'rtgh_object': rtgh,
        'ispgnr': ispgnr,
        'ntrl': ntrl,
        'prrg': prrg,
    }

    return rndf(request, self.tmpl,
                context=context)
```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**Дослідження та програмна реалізація
засобів ведення електронної звітності
співробітників підприємства**

Виконав: Сеттар АБЛЯКІМОВ

ст. групи КНТ-122м

Керівник: Михайло КОЦУР

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для ведення електронної звітності.

Предмет дослідження – програмні засоби підтримки процесу ведення електронної звітності.

Метою роботи є дослідження та розробка програмного забезпечення для ведення електронної звітності співробітників підприємства.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у дипломній кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для ведення електронної звітності;
- розробити модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства;
- здійснити проектування програмного забезпечення для ведення електронної звітності співробітників підприємства;
- створити програмне забезпечення онлайн для ведення електронної звітності співробітників підприємства;
- виконати тестування розробленого програмного забезпечення для ведення електронної звітності співробітників підприємства.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	bitrix24	Jira	Mantis
Зручність використання	–	+	+
Дружній інтерфейс користувача	+	+–	+
Інтуїтивна зрозумілість	–	–	+
Багатофункціональність	+–	+	+
Засоби інтеграції з іншими програмними продуктами	+–	+	+

4

Рисунок В.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	JavaScript	PHP
Зручність для розробки ПЗ для ведення електронної звітності співробітників підприємства	+	–	+–
Ефективність	+	+	+–
Зручність синтаксису	+	+–	+–
Велика спільнота	+	+–	+
Розширюваність	+	+–	+

5

Рисунок В.5 – Слайд 5

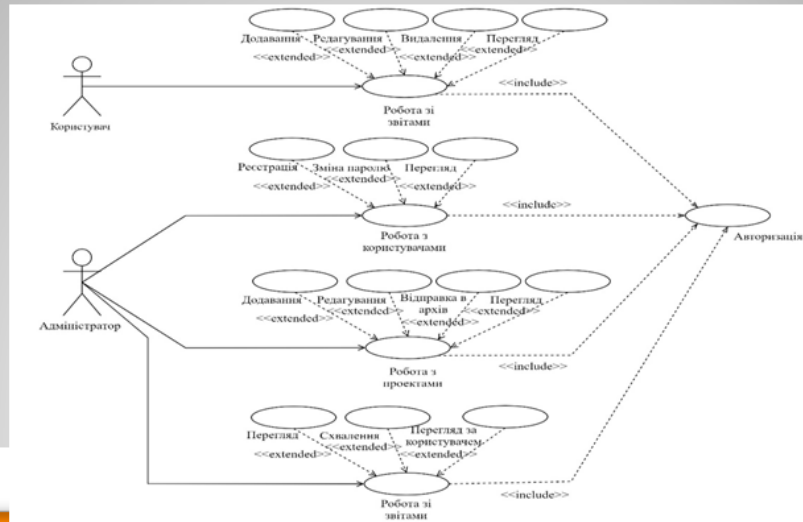
Порівняння фреймворків

Критерій порівняння фреймворків	Фреймворки		
	Django	Flask	Ruby on Rails
Зручність для створення програми для ведення електронної звітності	+	–	–
Базова мова програмування	Python	Python	Ruby
Висока швидкість розробки	+	+–	+
Велика спільнота розробників	+	+–	+
Наявність великої кількості документації	+	+	+–

6

Рисунок В.6 – Слайд 6

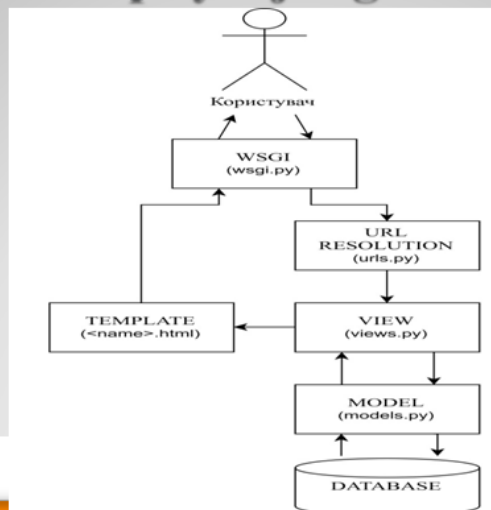
Модель взаємодії користувача з програмним забезпеченням



7

Рисунок В.7 – Слайд 7

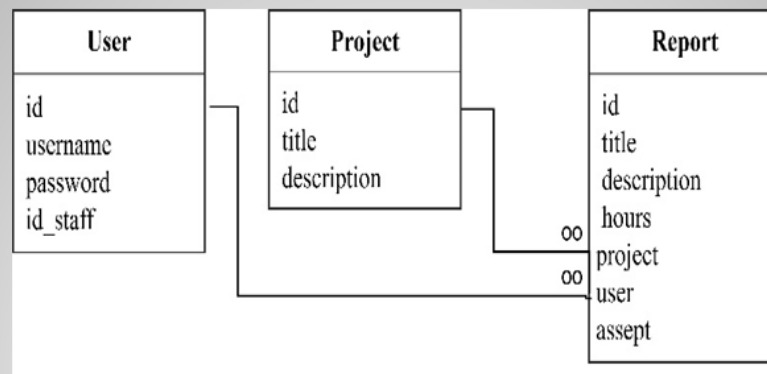
Архітектура програмного забезпечення на основі фреймворку Django



8

Рисунок В.8 – Слайд 8

Діаграма класів ПЗ для ведення електронної звітності співробітників підприємства



9

Рисунок В.9 – Слайд 9

Загальна схема функціонування програми



10

Рисунок В.10 – Слайд 10



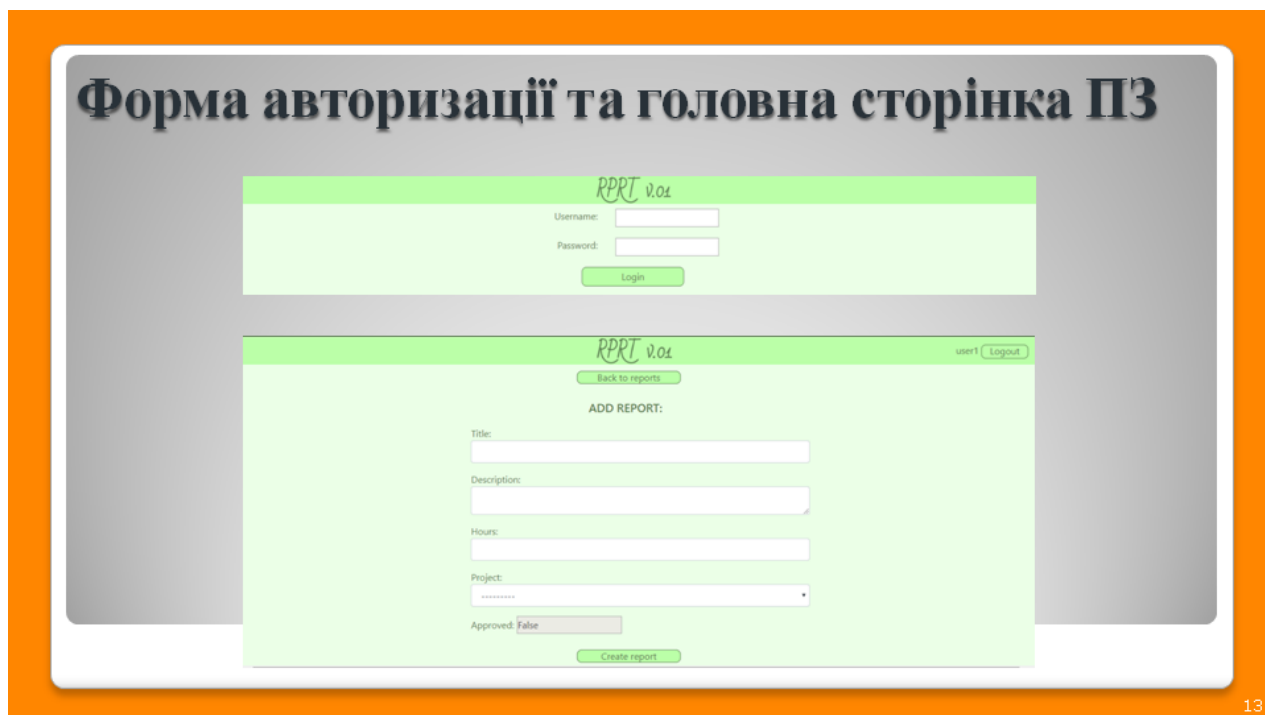
11

Рисунок В.11 – Слайд 11



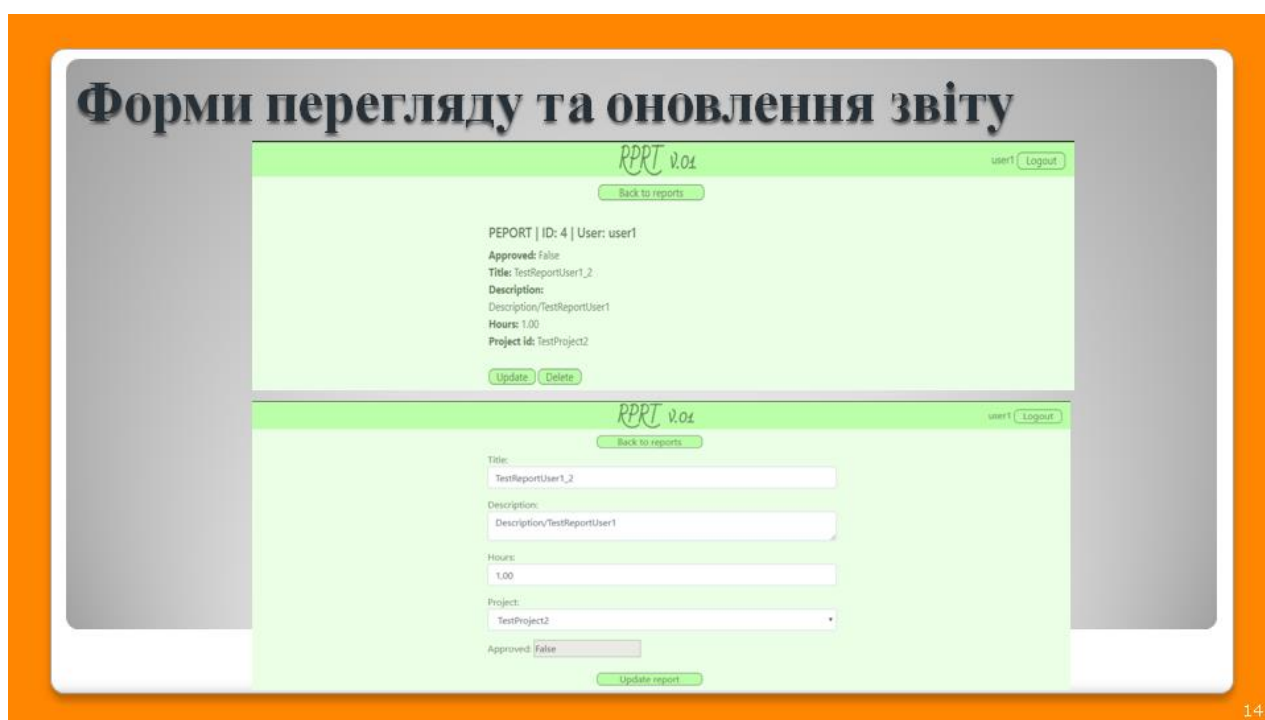
12

Рисунок В.12 – Слайд 12



13

Рисунок В.13 – Слайд 13



14

Рисунок В.14 – Слайд 14

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для ведення електронної звітності.

Новизна роботи полягає в тому, що створено модель взаємодії користувача з програмним забезпеченням для ведення електронної звітності співробітників підприємства, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для ведення електронної звітності співробітників підприємства.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення для ведення електронної звітності співробітників підприємства.

Усі завдання випускної дипломної роботи магістра повністю виконано.

Рисунок В.15 – Слайд 15