НУБіП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

**Факультет інформаційних технологій**

**УДК 004.9**

| **«ПОГОДЖЕНО»** | **«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»** |
|---|---|
| Декан факультету інформаційних технологій | Завідувач кафедри комп'ютерних наук |
| Глазунова О.Г., д.п.н., професор | Голуб Б.Л., к.т.н., доцент |
| _____ | _____ |
| ___ _____ 2021 р. | ___ _____ 2021р. |

# МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему** «Експертна система управліннв командною роботою розробників»

Спеціальність 121 «Інженерія програмного забезпечення»
(код і назва)

Освітня програма Програмне забезпечення інформаційних систем
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми**

кандидат технічних наук, доцент _____ Голуб Б.Л.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

**Керівник магістерської кваліфікаційної роботи**

_____ _____ Міловідов Ю.О.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

**Виконав**

_____ Мороз Є.С.
(підпис) (ПІБ студента)

**КИЇВ-202**

НУБіП України

**НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет інформаційних технологій**

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри** _____

к.т.н., доцент _____ Гслуб Б.Л

(науковий ступінь, вчене звання) (підпис) (ПІБ)

“ ____ ” _____ 20____ року

**З А В Д А Н Н Я**

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ**

Мороза Євгенія Олексійовича

(прізвище, ім’я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма програмне забезпечення інформаційних систем

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи «Експертна система управління командною роботою розробників»

затверджена наказом ректора НУБіП України від “ 29 ” жовтня 2020 р. № 1636 «С»

Термін подання завершеної роботи на кафедру 2021.11.30

Вихідні дані до магістерської кваліфікаційної роботи:

1) дані про розробників, проекти, задачі та кількість витраченого часу, зібрані за допомогою системи управління проектами;
2) технічна документація;
3) існуючі рішення;

Перелік питань, що підлягають дослідженню:

| № з/п | Питання, що підлягає дослідженню | Строк виконання | Примітка |
|---|---|---|---|
| 1. | Аналіз предметної області. | 21.09.2020-24.10.2020 | |
| 2. | Дослідження технології OLAP | 30.11.2020-31.12.2020 | |
| 3. | Проектування системи | 01.02.2021-06.03.2020 | |
| 4. | Дослідження технології Data Mining | 11.03.2021-10.04.2021 | |
| 5. | Розробка системи та алгоритмів для аналізу даних | 22.04.2021-22.05.2021 | |
| 6. | Дослідження отриманих результатів | 1.09.2021-13.11.2021 | |
| 7. | Попередній захист | 15.11.2021-10.12.2021 | |
| 8. | Захист | 14.12.2021 | |

**Дата видачі завдання** “ 12 ” листопада 2021 р.

**Керівник магістерської кваліфікаційної роботи** _____ Мілєвідов Ю. О.

( підпис ) (прізвище та ініціали)

**Завдання прийняв до виконання** _____ Мороз Є. О.

(підпис ) (прізвище та ініціали студента)

# ЗМІСТ

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ** 4

**ВСТУП** 5

**1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ** 8

1.1 Постановка завдання 8

1.2 Технічне завдання 14

1.3 Аналіз існуючих рішень 17

**2. Теоритичне підгрунтя** 21

2.1 Експертна система аналізу даних 21

2.2 Технології OLAP 22

2.3 Технологія Data Mining 24

2.4 Вибір технологій для створення системи 27

**3. РОЗРОБКА СИСТЕМИ** 31

3.1 Архітектура системи 31

3.2 Система 38

3.3. Сховище даних 48

**4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ** 52

4.1 Механізм вилучення, обробки та передачі даних 52

4.2 Використані методи Data Mining 56

4.3 Розрахунок KPI 63

**ВИСНОВОК** 66

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 68

НУБіП України

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- BI - Business intelligence.
- KPI – Key Performance Indicators, ключовий показник ефективності(успішності).
- OLAP – online analytical processing, аналітична обробка в реальному часі.
- SQL – structured query language, мова структурованих запитів.
- БД – база даних.
- ІС – інформаційна система.
- СД – сховище даних.
- ПО – предметна область.
- СБД – система баз даних.
- ЕС – експертна система.
- СППР – система підтримки прийняття рішень.

НУБіП України

# ВСТУП

**Актуальність**

Всесвітня павутина являє собою надзвичайно перспективний засіб комунікації, що може пояснити його сучасний надзвичайний розвиток. Сьогоднішні засоби зв'язку дозволили об'єднати розрізнені комунікаційні системи у глобальну мережу.

Завдяки стрімкому розвитку технологій представники багатьох професій без особливих складнощів можуть переходити працювати з офісу додому. Причин цьому може бути безліч: створення спеціалізованих віддалених команд, ремонт в офісі. Ситуація, що склалася в світі, не є винятком. Пандемія нового вірусу вже сильно вдарила по економіці і може завдати ще більшої шкоди, якщо не бути підготовленим до цього.

Навіть така велика компанія як Google, відправляє своїх працівників на роботу в віддаленому режимі. Багато світових компаній, в тому числі Amazon і Facebook, дотримуються такої ж тактики. Віддалений формат роботи не означає, що компанія повинна відмовитися від таких активностей, як наради, навчання або тімбілдинг. Навпаки, потрібно найшвидше налагодити процеси роботи в команді за допомогою сучасних методів та інструментів для управління командою.

**Мета дослідження**

Метою даної роботи є дослідження використання сучасних методів та інструментів по управлінню командною роботою розробників під час розробки проектів.

**Об'єкт дослідження**

Об'єктом дослідження є система моніторингу та аналізу успішності методів та інструментів на прикладі розробки проектів.

**Предмет дослідження**

Предметом дослідження є система моніторингу та аналізу успішності методів та інструментів на прикладі розробки проектів.

НУБіП України

**Завдання**:

- Провести аналіз предметної області
- Сформувати вимоги
- Моделювання системи
- Побудувати структуру сховища даних
- Заповнити сховище даними з джерел
- Провести аналіз на основі даних
- Сформулювати висновки на основі отриманих результатів

**Методи дослідження**

Під час проведення даного дослідження буде створено сховище даних, яке заповниться даними про проект, поставлену задачу, дані програміста котрий виконує цю задачу та часові проміжки, та факти в яких будуть збиратися данні по кількості виконаних задач та витраченого часу. За допомогою технології OLAP, яка дає можливість аналізу великого обсягу даних в режимі реального часу, буде проведено аналіз на основі отриманих даних. Якщо системи регламентованої звітності дозволяють відповісти на питання "Скільки команда витратила часу на розробку проекту?", то OLAP-системи дають відповідь на питання: "Який розробник працює найефективніше працює і чому".

**Наукова новизна**

Вдосконалення алгоритмів аналізу роботи в команді розробників з самостійно розробленою підсистемою аналітики витраченого часу та кількості виконаних задач, кожного з учасників.

**Апробація результатів дослідження**

1. Мороз Є. О. Експертна система управління командною роботою розробників // Збірник матеріалів XI Міжнародної науково-практичної конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта». – Київ. – 2020. – ст. 117-118 (тези доступні за посиланням: https://drive.google.com/file/d/1-Y7Q05n4Gy7u6YIqtIP6YZuzZEcO4y97/view).

2. Мороз Є. О. Експертна система управління командною роботою розробників // Збірник матеріалів конференції «Теоретичні та прикладні аспекти розробки комп'ютерних систем» . – Київ. – 2021. – ст. 38-39 (тези доступні за посиланням:https://drive.google.com/file/d/1e77WfOOL5T86sbVmPKbcerO676kOvBV4/view).

**Структура магістерської роботи**

Пояснювальна записка складається з вступу, висновку, використаних джерел та чотирьох розділів:

1. Системний аналіз предметної області. В даному розділі було проаналізовано предметну область, з різними нюансами, для того, щоб отримати більш гнучку та ефективну теоритичну структуру. Для кращого розуміння взаємодії учасників, в процесі роботи в команді, було побудовано діаграму прецедентів. На основі отриманих вимог, було сформовано технічне завдання для розроблюваної системи. Наступним етапом було проаналізовано наявні рішення, для покращення розуміння та виявлення різних шляхів рішень під час вирішення задачі.

2. Теоритичне підгрунтя. В даному розділі спочатку, було визначено, що таке експертна система, для вирішення яких задач потрібна та на основі, яких рішень або даних базується. Наступним етапом, було обрано дві технології, OLAP та Data Mining, які допомагають підготувати дані та проаналізувати їх.

3. Розробка системи. В цьому розділі була визначена та сформована архітектура розроблюваної системи з побудованими діаграмами, пакетів, компонентів та топологією системи. Представлено систему у вигляді зображень та описаних вузлів системи, та описано структуру сховища даних.

4. Результати дослідження. В даному підрозділі, описані механізми, які допомагають в отриманні, обробці та передачі даних. Описаний процес використання методів Data Mining, та опис отриманих результатів у вигляді

НУБіП України

звітів. Також описаний та представлений у вигляді діаграм, процес розрахунку KPI для команди розробників, на основі отриманих даних.

# 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Постановка завдання

Дистанційна робота - це робота яка виконується віддалено, від місць де використовуються результати цієї роботи. За таких умов роботи працівник так само працює в штаті компанії, але не ходить в офіс. Робота виконується зі зручної локації. Він може виконувати роботу з будь-якого зручного місця, вдома, в кафе. Деякі галузі, наприклад фриланс, такому явищу, як дистанційна робота, зобов'язані самим своїм існуванням.

Неможливо уявити собі, як фрилансер із маленького міста міг би надати свої послуги замовникам, які знаходяться в іншій країні, якщо б не існувало сервісів для пошуку фрилансу, які зводять замовників із виконавцями у віддаленому режимі. Тому в першу чергу, важливо не поєднувати поняття «віддалена робота» та «фріланс». Це різні речі, адже штатний співробітник компанії цілком може працювати віддалено. Фрілансер, в свою чергу, сам собі керівник, який може сам будувати графік, систему роботи та виконувати проекти від різних замовників, але працює теж дистанційно.

Можливість роботи з дому забезпечує людям ряд дуже важливих переваг:

Робота у власному ритмі. Більшість працівників, якы працюють у выддаленому режимы, можуть самі спланувати власний час та працювати в такому темпі, якийзручний саме для них.

Заощадження часу. В середньому людина витрачає на поїздку (та підготовку до неї) майже дві години на день, за місяць це становить майже 44 години, а за рік набіжить ще більше.

Можливість подорожувати. В наш час, технології розвинулись достатнього, щоб можливість працювати, була практично в будь-якому місці, де є доступ до Інтернету. Skype, Slack, CRM та до інших інструменті, які

дозволяють комунікувати з клієнтом або організувати роботу команди хоч з іншого кутка землі.

- Легше сконцентруватися. Якщо ви працюєте маленькій або в великій команді, в офісі буде набагато більше стресових ситуацій або різноманітних ситуацій які відволікають, ніж вдома.
- Висока мобільність. З Інтернетом у нашій країні проблем уже немає практично ніде, і якщо у вас є ноутбук, працювати можна не лише з дому, але й з будь-якого місця, де є доступ в інтернет (готель, аеропорт, кафе).

Незважаючи на безліч переваг, віддалена робота може здатися раєм далеко не всім. Перша проблема, з якою можна зіткнутися – відсутність самоконтролю. Якщо людина не вміє зосередитися та регулярно відволікається на всілякі дрібниці, це може помітно знизити продуктивність її роботи.

Друга проблема – відсутність живого спілкування. Якщо людина звикла до постійних балачок з колегами і посиденьок за чашкою кави, то робота вдома на самоті їй може швидко набриднути.

Третя проблема – відсутність робочого простору. Якщо людина живе в однокімнатній квартирі з сім'єю і домашніми улюбленцями, то перебувати в тиші та спокої досить важко. Ізольоване робоче місце потрібно продумати заздалегідь, щоб ніщо і ніхто не відволікав Вас від роботи.

До недоліків можна віднести також сидячий спосіб життя. Підтримувати себе у формі важко, не виходячи з дому. Тому необхідно виробити корисну звичку, і регулярно виходити на свіже повітря або ж займатися яким-небудь активним видом спорту.

Головною проблемою є вибір правильного способу дистанційного управління командою, для кожної компанії є свої індивідуальні потреби, але є основні принципи, яких потрібно дотримуватись під час управління проектами та які повинна реалізовувати розроблювана система.

1. Запуск проекту.

На цьому етапі необхідно чітко сформулювати основні цілі проекту та надати його короткий опис, включаючи високорівневий план проекту, зібрати команду, запропонувати основні правила комунікації та зазначити подальші дії

команди. Тому в розроблюваній системі, менеджер проекту зможе створювати проект, та прикріпляти до нього розробників.

2. Планування

Планування є найважливішою частиною в управлінні проектами. Після запуску проекту до створюються конкретні задачі, за якою закріплюють відповідного члена команди.

При віддаленій роботі над проектом особливо важливо, щоб усі учасники проекту були ознайомлені з переліком та статусами задач в режимі реального часу та розуміли зону своєї відповідальності. Після створення проекту менеджер повинен мати можливість створювати задачі, з можливістю установки пріоритету, статусу, рівня складності задачі, додавання опису та потрібних матеріалів для реалізації та закріпити відповідального за цю задачу.

3. Забезпечення єдиного інформаційного простору

Під час дистанційного виконання проектів потрібно надати доступ до проектної інформації та дати можливість комунікації.

Використання систем управління проектами покращує рівень розуміння для членів команди та підвищує їх рівень залученості. Ознайомлення з бізнес-цілями, показниками ефективності та пов'язаними даними надає можливість людям краще зрозуміти значення своєї ролі при розробці проекту та побачити вплив їхнього особистого внеску, що, підвищує мотивацію.

Тому потрібно надати можливість залишати коментарі під час виконання задачі, для кращого розуміння ситуації по її виконанню, та можливість прикріпляти до задачі файли, які допомагають під час реалізації.

4. Реалізація проекту та контроль

Коли команда знаходиться в офісі і це сприймається як належне, можна швидко дізнатися статус відповідної задачі, перемовившись с колегою біля кулеру або у їдальні, ни підійти до робочого місця члена команди, або зібрати всю команду на декілька хвилин у конференц-залі, то під час роботи з дому можливість такого контактування обмежена.

НУБіП України

Необхідно постійно відстежувати основні показники продуктивності, контролювати завантаженість співробітників та трудовитрати на виконання тих чи інших завдань, водночас не перевантажуючи колег надмірними комунікаціями.

Тому дистанційне управління проектами вимагає ще більш ретельного та прозорого підходу до встановлення пріоритетів, контролю виконання завдань, термінів та процесів. Тому система повинна надавати можливість встановлення пріоритету, статусу виконання та рівня складності задач, також потрібно надати пожливість внесення часу, яке було витрачено на задачу, щоб менеджер розумів, якою задачою працівник займався в певний період часу.

Дана інформація про час та статуси потрібна для того, щоб була можливість отримати звітність по кожному працівнику по кількість витраченого часу, кількості виконаних задач та рівня його ефективності.

Діаграми прецедентів потрібна, щоб візуалізувати поведінку системи, підсистеми або класу, тобто, щоб кожен користувач міг зрозуміти, як використовувати систему, а розробник в свою чергу, реалізував відповідний елемент.

Прецеденти складаються з великої кількості сценаріїв (кожен крок описує взаємодію між користувачем і системою), які об'єднані між собою певною метою користувача. Кожна складна система має декілька десятків прецедентів, кожен з яких можна розгорнути, також в кілька десятків сценаріїв. Для кожного з прецедентів можна виділити головні сценарії, що описують допоміжні сценарії, які в свою чергу описують альтернативні послідовності, та найважливіші послідовності.

Є два типи прецедентів: бізнес-прецедент (описує функціональність верхнього рівня та призначена для замовника програмної системи) і системні прецеденти (вони описують функціонал для нижнього рівня та будується для розробника програмної системи). Кожен прецедент повинен мати ім'я, яке унікальне в рамках пакета.

НУБіП України

При створенні імені для прецедентів використовують короткі дієслівні фрази в активній формі, які означають деякий поведінку. Також, якщо визначити відносини узагальнення між прецедентами можна їх організувати.

Узагальнення яке відбувається між прецедентами аналогічне узагальненним відносинам між класами. Тобто дочірній прецедент наслідує поведінку свого батьківського прецеденту, може замінювати його або доповнювати його поведінку, а також, може замінювати усюди, де з'являється його батьківський прецедент.

Використання (Uses) використовується, коли існує будь-якої фрагмент поведінки системи, який повторюється більше ніж в одному прецеденті.

Розширення, це май теж саме, що і узагальнення. Під час побудови моделі може розширювати прецедент, також може доповнити поведінку основного прецеденту, але для цього повинні бути визначені точки для розширення. При цьому прецедент, який розширюється може доповнити поведінку основного прецеденту тільки в певних точках розширення. Розширення може використовуватись, коли точніше описується зміна деякої нормальної поведінки.

Варто пам’ятати, що словесний опис будь-якого з прецедентів, навіть самого елементарного, повинен включати в себе, мінімум 3 розділу, в які входять: попередня умова, основний потік, постумова. Однак, в загальних випадках опис прецеденту може включати в себе тільки один або декілька альтернативних потоків, які пов'язані з різними кроками основного потоку, це посилання на інші прецеденти, вказівки конкретних акторів, залучених до прецедент, тобто розділ "Дійові ".

Згідно діаграми зображеній на Рис. 1.1.1, ми бачимо, що з розробленою системою будуть взаємодіяти чотири актори:

- Клієнт. Даний актор надає замовлення на розробку проектів.
- Менеджер проекту. Головним обов’язком є взаємодія з клієнтом, розподілення задач між програмістами та контроль виконання.

НУБіП України

- Програміст. Після отримання задач, програміст виконує їх та передає їх на перевірку.
- Аналітик. Дивлячись на роботу в команді за обраною методологією та системою, даний актор проводить аналіз слабких місць в проекті, тобто які недоліки є в ТЗ по виконанню проекту, що можуть вплинути на подальшу продуктивність команди. Проводить аналіз слабких місць в команді, та в залежності від цього проводить певні збори, мітинги щоб запропоновувати рекомендації по управлінню командою.

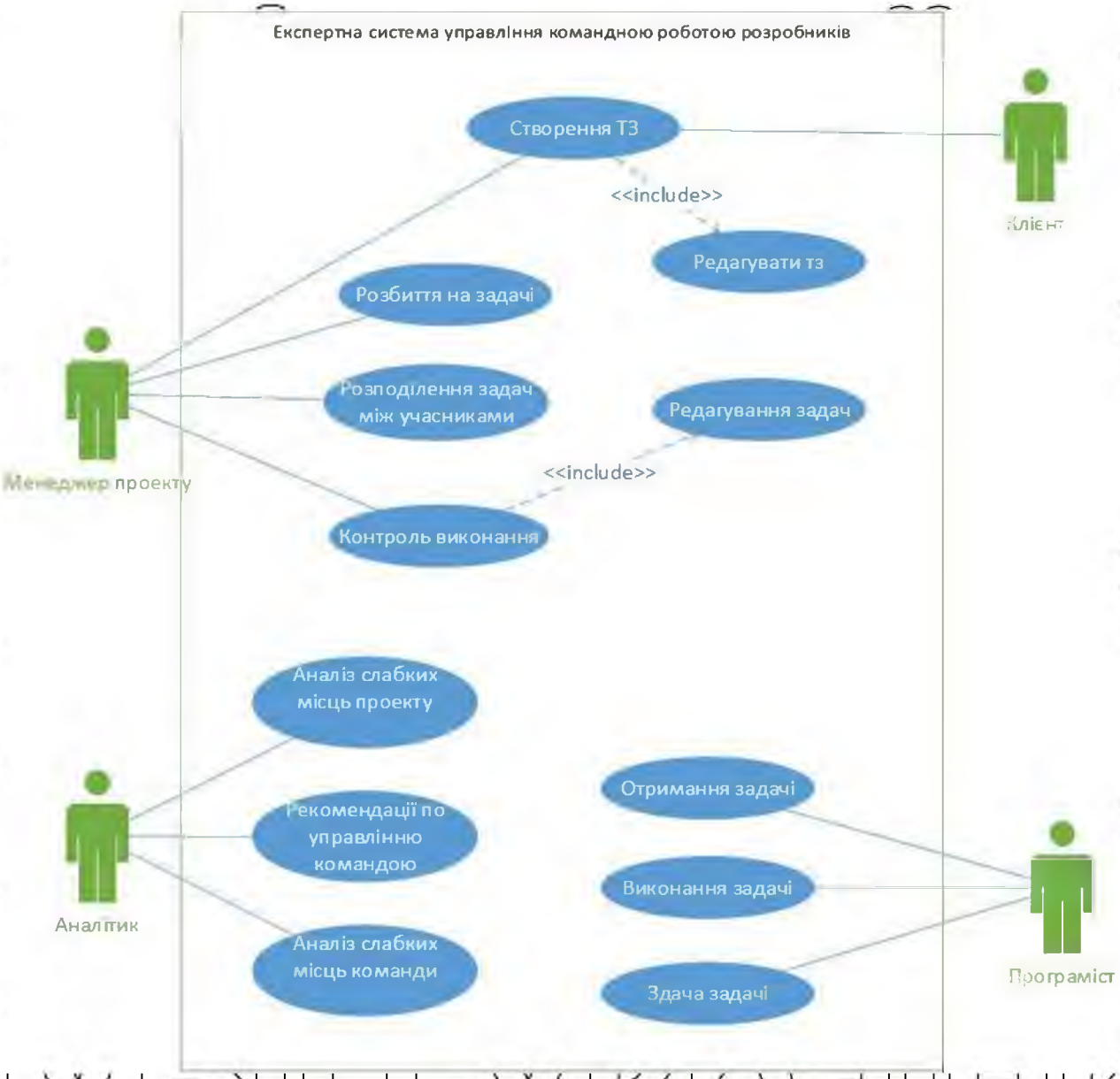



*Рис. 1.1.1 Діаграма прецедентів*

## 1.1 Технічне завдання

Створення та обслуговування системи — це комплексний процес. На початковому етапі потрібно зібрати інформацію про майбутню систему, саме технічне завдання містить ці відомості. Воно дуже важливе, адже визначає всі наступні етапи і має великий вплив на вигляд системи після завершення її розробки та на успіх проекту загалом.

На основі технічного завдання здійснюється наступний крок — оцінка проекту, тобто його рівень складності і відповідно розрахунок терміну на його розробку. На цьому неоднозначному і спірному етапі можна зіткнутися з проблемою точної оцінки часу, який буде витрачений на розробку. Тому, потрібно якомога детальніше розписати технічне завдання з самого початку, щоб зміни в подальшому, були мінімальними, тим точніше будуть прогнози. Чим чіткіше описати кінцевий результат проекту, тим простіше буде до нього наблизитись.

На початку створення технічного завдання, визначимо цільову аудиторію, на яку розраховано сайт:

1. Менеджери проектів – матимуть можливість налаштування процесів роботи команди у віддаленому режимі.
2. Розробники – матимуть швидкий доступ до необхідної інформації про проект, етапи виконання, дедлайни та ін.

Вимоги до інтерфейсу:

1. Простий та зручний у користуванні.

Мова сайту: українська.

Опис функціоналу який повинна забезпечувати система:

- створення форми реєстрування користувачів з можливістю тривалого зберігання даних, яка буде містити такі поля:
  - ім`я;
  - електронна пошта;
  - пароль;
  - підтвердити пароль;
  - створення форми авторизації користувача з полями:
  - електронна пошта;

- пароль;
- створення форми створення проекту з можливістю тривалого зберігання даних, яка буде містити такі поля:
  - назва;
- створення сторінкі відображення проектів з даними;
- створення форми створення задач з можливістю тривалого зберігання даних, яка буде містити такі поля:
  - назва;
  - дата;
  - час;
  - вибір статусу;
  - вибір пріоритету;
  - вибір рівня задачі;
  - вибір проекту;
  - файли;
  - детальний опис;
- створення сторінки відображення задач, з можливістю фільтрування по фластивостям, пошуку та сортування задач;
- створення детальної сторінки відображення задачі, з можливістю редагування, додавання коментарів та файлів;
- створення сторінкі профілю з відображенням даних поточного профілю, та можливістю їх редагування;
- створення сторінки «Журнал робіт», на якій буде можливість за допомогою фільтра відобразити кількість використаного часу розробника в певний проміжок часу.
- Створення сторінки «Статистика виконання задач», на якій буде можливість за допомогою фільтра побудувати статистику з кількістю загального та середнього витраченого часу на різні рівні задачи розробника та отримати рівень виконання задач.
- Створення сторінки «Графік виконання задач», на якій буде можливість за допомогою фільтра побудувати графік кількость виконаних задач за обраний проміжок часу.
- створення головної сторінкі(ознайомчої), де будуть описані переваги данного ПЗ, у вигляді тексту та картинок;

Після визначення основного функціоналу системи, потрібно побудувати базу даних.

База даних - це засіб збереження та організації великих об'ємів інформації про об'єкти предметної області. БД повинна відображати поточні дані про

предметну область, збирати, зберігати інформацію та надавати користувачам з різними категоріями допуску швидкий доступ до даних. Для цього дані в базі мають бути структуровані відповідно до деякої моделі, що відображає основні об'єкти ПО, їх властивості і зв'язки між ними. БД є частиною складної системи або системою баз даних (СБД).

При проектуванні важливо визначити їх атрибути та деякі правила, що обмежують можливість введення користувачем неправильних значень.

Основні кроки проектування БД:

- визначити інформаційні потреби БД;
-проаналізувати об’єкти, що потрібно промоделювати в БД.
- визначити атрибути, які ідентифікують кожен об’єкт.
-визначити правила, які будуть підтримувати цілісність даних.
- встановити зв’язки між об’єктами, провести нормалізацію таблиць.
-вирішити питання щодо надійності даних та їхнього збереження.

Після побудови структури бази даних потрібно її наповнити якісними даними, наступним етапом є створення сховища даних на основі створеної бази даних.

Сховище даних – це особлива форма організації бази даних, яка призначена для зберігання в узгодженому вигляді агрегованої інформації, що будується на основі баз даних різних систем та зовнішніх джерел.

Сховище даних будується на основі клієнт-серверної архітектури, СУБД і утиліт підтримки прийняття рішень. Дані, які потрапляють у сховище, мають доступ тільки для читання.

Ключовим компонентом під час організації сховищ даних є OLAP . Ця технологія має основу на побудові і візуалізації багатовимірних кубів даних, та має можливість довільно маніпулювати даними, які містяться в кубі, це дозволяє представити дані для аналізу в будь-якому розрізі.

Властивості сховища даних:

- предметна орієнтація (інформацію організовано відповідно до кожного з основних аспектів діяльності);

- інтегрованість даних (в сховище дані потрапляють з різних джерел і відповідно агрегуються);
- інваріантність та стабільність у часі (записи в DW, являють собою відбитки даних, зроблені у певний час, та ніколи не змінюються);
- зменшує кількість збиткової інформації (для завантаженням у сховище дані сортуються тільки у певній послідовності, відповідно фільтруються, а також формується деяка підсумкова інформація).

В нашому випадку керівнику команди потрібно знати кількість виконаних задач за деякий період, та кількість витраченого часу. При цьому команда розробляє не один, а декілька проектів.

Тому, потрібно задати правильні питання, на які відповість наше СД:

- Скільки витрачається часу на одну задачу певного рівня?
- Скільки виконує задач один програміст за певний проміжок часу?

## 1.3 Аналіз існуючих рішень

Для того щоб зрозуміти які ріщення вже існують, та чим можна індивідуалізувати розроблювану систему, було проведено аналіз статей, патентів та інших наукових робіт пов’язаних з організацією, управлінням командною роботою розробників та розробки системи.

В посібнику Романовський О.Г. розкриває основні визначення, завдання, історія дослідження, основні компоненти, принципи та інструменти тимбілдингу, основні відмінності команди та групи, а також її значення для успішної діяльності бізнес-лідера, психолога-лідера та лідера у науці. Розглянуто, як правильно використовувати особливості створення ефективної команди та розвивати основні якості, які необхідні в процесі тимбілдингу. Навчитись володіти базовими навичками тимбілдингу: вмінням створювати команди та використовувати особливості рольової структури команди, володіти інструментами тимбілдингу та його компонентами [1].

В монографії Н. В. Доценко досліджено залежність між чисельністю команди та її успішністю. За надбаннями вчених Доценка Н.В, Сабадоша Л.Ю,

Чумаченка І. В. було виділено, що процес формування команди складається з чотирьох етапів. Також було виділено моделі формування, за допомогою яких можна створити команду [2].

У даній книзі Джефф Сазерленд узагальнив власний управлінський і консультативний досвід зі створення Scrum-команд, а також досвід своїх послідовників в різних країнах. Автор переконаний, що Scrum пропонує надто серйозні конкурентні переваги, щоб його не використовувати, і дає докладні рекомендації по організації роботи. Примітний той факт, що, поряд з бізнесом, Scrum ефективний і в особистому житті, і в вирішенні глобальних проблем[3]. Також Кен Швабер та Джефф Сазерленд розробили посібник з користування Скраму. Визначення полягає в описі ролей, церемоній та артефактів Скраму, а також правил, що забезпечують зв'язок між ними. На мою думку це більш уточнена версія попередньої книги, яка побудована на визначеннях та дає можливість більш детально розібрати структуру, яку потрібно побудувати при організації роботи в команді [4].

В.А. Рач в своїй роботі «Управління проектами та розвиток виробництва», розглядає проект з позиції двох світоглядних метрик інтуїтивної та наукової. Розкрито механізм їх конвергентного взаємодії. Запропоновано модель системи наук, в якій доданий новий елемент цивілізації, в якій враховано появу нової відповідних науково технічних областей [5].

В науковому дослідженні за спеціальністю «Управління проектами та програмами», було розкрито більш загальні відомості про управління в організаційно-технічних системах, а саме які є стандарти по управлінню проектами. Які знання повинен мати спеціаліст по управлінню проектами, щоб відповідати міжнародним вимогам до компетенцій проектних менеджерів, що затверджені міжнародною асоціацією з управління проектами (IPMA)[6].

В своєму посібнику О. Петрик розглянув основні принципи і методи проектування програмного забезпечення (ПЗ), які допоможуть при побудові системи. Їх практичного застосування з використанням сучасних

НУБіП України

інструментальних засобів побудови моделей, що використовуються на різних етапах життєвого циклу ПЗ та характеризують різні властивості ПЗ[7].

В посібнику по управлінню проектами розглянуто комплекс питань, в сукупності складових сутність дисципліни - «управління проектами» (Project Management). Висвітлено всі елементи управління проектами, представлені в чотирьох частинах: знайомство зі світом управління проектами; розробка проекту; функції управління проектами; підсистеми управління проектами[8].

В патенті «Методи та система управління проектами для комп'ютерного коду та візуалізації» розглядаються системи та методи для обробки та передачі інформації та даних, таких як ділова інформація, а особливо до систем, програмного забезпечення, обладнання, продуктів та процесів для використання підприємствами, приватними особами та іншими організаціями для збору, обробки, розподілу, аналізу та візуалізації інформація, включаючи, але не обмежуючись ними, бізнес-аналітику, візуалізацію даних, зберігання даних та видобуток даних [9].

Робертас Дамашевичюс в своєму патенті представив дуже цікаву ідею у вигляді системи, яка дозволяє учасникам команд отримувати винагороду за завданнями проекту, де вони накопичують бали за різними критеріями, включаючи те, наскільки їх робочий продукт відповідає раніше встановленим критеріям, а також можливість отримати додаткові бали за засвоєння різних навчальних матеріалів, пов'язаних із завданнями. Інші фактори, такі як надання допомоги іншим членам команди, оцінка продукту роботи тощо, також отримують бали[10].

В даному патенті про систему підтримки командної роботи предетавлено винахід, який відноситься до методу колективної роботи та комп'ютерної системи, за допомогою якої можна зібрати оцінки групи щодо особливостей різних цілей та зробити резюме. Метод швидкий, демократичний, забезпечує анонімність, придатний для роботи в інформаційній мережі, багатовимірність, виправлення похилого ставлення та придатний для роботи незалежно від місця. Система включає сервер (1), робочу станцію основного користувача (3), ряд

робочих станцій для оцінювачів (4-7) та програмне забезпечення, що використовується веб-браузером, до якого основний користувач та оцінювачі можуть підключитисяБ але в даній системі не має можливості проаналізувати роботу одного з учасників команди або команди розробників[11].

В патенті по управлінню задачами програмних проектів на основі API , я почерпнув багато інформації для подальшої своєї розробки, тому що в ньому представлено програмне рішення для управління, сортування та ранжування списків завдань та інтеграції управління завданнями та часом, таким чином, що завдання можуть бути автоматично або вручну призначені заданим часовим блокам. Користувачі можуть контролювати взаємозв'язок між обсягом завдань та доступним часом, за який їх потрібно виконати, але після виконання задач або підчас, менеджер проекту немає можливості побудувати звітність та проаналізувати роботу команди або розробника, та їх ефективність [12].

# 2. Теоритичне підгрунтя

## 2.1 Експертна система аналізу даних

Експертна система (ЕС) - це система штучного інтелекту, яка використовує отримані знання, щоб забезпечити високоефективне рішення задач у вузькій професійній області. Експертні системи відносяться до систем підтримки прийняття рішень (СППР), основаних на знаннях. В таких системах використовують різнії методи для аналізу даних. Сьогодні дані системи розвиваються у межах певного напрямку, якій одержав назву Data Mining –це отримання або підготовка даних.

Експертні системи, можна поділити на:

- Діагностичні

- ЕС моніторингу
- ЕС прогнозування
- ЕС планування

Експертні діагностичні системи відносять об'єкти до визначених класів. Застосовуються в налузях, від встановлення несправностей у технічних системах (технічна діагностика) до розпізнавання захворювань живих організмів, а також соціальних і природних аномалій.

Експертна система моніторингу інтерпретує даних у реальному часу та сигналізує про вихід тих чи інших параметрів за припустимі межі.

Експертна система прогнозування представляють ймовірні наслідки із певних ситуацій. У таких системах часто використовуються параметричні моделі, у яких значення параметрів підлаштовуються під аналізовану ситуацію. Останнім часом для рішення задачі, часто використовують інші підходи, зокрема нейрономп'ютерний підхід, різні алгоритми пошуку логічних закономірностей у структурах багатомірних даних.

Експертна система для планування належить до об'єктів, які здатні виконувати певні функції планування. У таких системах використовуються моделі поведінки реальних об'єктів для того, щоб послідовно вивести результати запланованої діяльності.

Призначення експертних систем бувають загального призначення та спеціалізованими (предметно орієнтованими та проблемно орієнтованими).

По складності виконання задач, їх поділяють на:

- Прості, до 1000 простих правил;
- Середні, від 1000 до 10 000 структурованих правил;
- Складні, більше 10 000 структурованих правил;

Експертні системи, також можна за їх ступінь інтеграції з іншими програмами.

Автономні ЕС використовуються для рішення "експертних" задач в режимі консультації, коли не потрібно залучати додаткові методи обробки даних

Гібридні ЕС поєднують стандартні пакети прикладних програм. Вони становлять інтелектуальні надбудови і виконують функції моніторингу стосовно відомого програмного забезпечення.

**2**.2 Технології OLAP

Для того щоб розроблювана система мала можливість проводити аналіз над отриманими даними однією з технологій було обрано OLAP.

OLAP - це аналітична обробка даних, вона дає можливість одночасно аналізувати інформацію з декількох систем баз даних. Також можна сказати, що це обчислювальний метод, який дає можливість користувачам легко отримувати потрібні дані, та дані які використовуються для запитів, щоб проаналізувати їх з різних точок зору. Вона будується на великих об'ємах даних, місце де зберігаються ці дані називають сховищем даних.

Наступним етапом він збирає необхідні дані зі сховища даних та проводить необхідний аналіз предметної області. В результаті отриманого аналізу можна прийняти рішення, в нашому випанку, щоб покращити продуктивність учасників команди, покращити рівень виконання задач, тощо. Тому дана технологія використовується у бізнес-аналітиці для запитів допомоги в аналізі тенденцій, прогнозуванні продажів, фінансовій звітності, плануванні, бюджеті тощо.

Тепер ми побачимо, як працює OLAP - спочатку збираємо дані з декількох джерел та зберігаємо в сховищах даних, які наступним етапом очищаються та будуються у вигляді кубів даних. Куб, тому що його класифікують за трьома вимірами, їх можна класифікувати за багатовимірними розмірами, архітектура OLAP предетавлена на Рис. 2.2.1.

НУБіП України

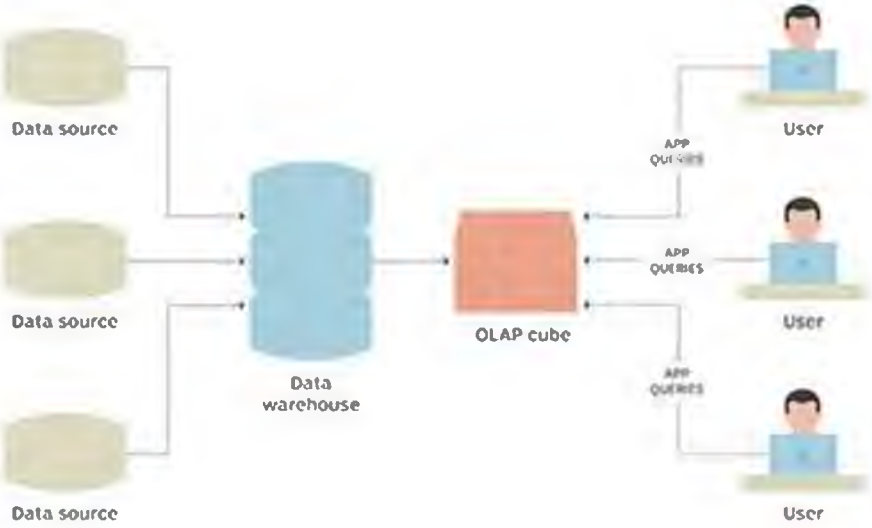



*Рис. 2.2.1 Архітектура OLAP*

Отже, кожен куб OLAP має дані, які класифікуються за деякими вимірами наприклад, розробник, часовий період, рівень задач тр та ін., які отримані за допомогою багатовимірних таблиць у сховищах даних. Параметри можуть приймати значення, такі як імена розробників, рівні задач, витрачений час на задачу, вони повинні буги організовані ієрархічно та мають можливість провести аналіз конкретних значень. Куби OLAP попередньо відсортовані за запитами, які найчастіше повторюються в різних розмірах, які покращують час виконання запитів над реляційними базами даних. Даний механізм сортування полегшує проведення аналізу протягом часу.

OLAP технологія створена для спрощення роботу в галузі, яка потребує звітності щодо продажів, звітності з управління, маркетингу, управління бізнес-процесами, фінансової звітності, складання бюджету та прогнозування тощо.

OLAP можна використовувати для виконання п'яти типів аналітичних операцій проти багатовимірних баз даних:

Згортання – також знайомия, як деталізація або консолідація, використовується для узагальнення даних після проведених операцій разом із розмірністю.

Детальний аналіз – використовується для аналізу в більш глибоких розмірах даних. Наприклад, створення з "періоду часу" до "років" і "місяців" і

"днів" і так далі, щоб побудувати графік кількості виконання задач за певний проміжок часу в нашому випадку.

Фрагмент - використовується для проведення аналізу потрібно взяти один рівень інформації для відображення, наприклад "кількість виконаних задач за останні 3 дні".

Кістки - використовується для проведення аналізу, щоб відібрати дані з різних вимірів для аналізу, наприклад "кількість виконаних задач певним розробникос за останні 3 дні".

Зворотний - використовується  під час проведення аналізу, який може отримати новий вид даних, обертаючи осі куба даних.

2.3 Технологія Data Mining

Data Mining – це автоматизований пошук даних, який оснований на аналізі великих масивів інформації. Для цілі береться ідентифікація тенденцій і шаблонів, яка при звичайному аналізі неможлива. Для сегментації даних та оцінки ймовірності наступних подій використовуються складні математичні алгоритми.

До основних задач цієї технології відносять:

Класифікація

Кластеризація

Прогнозування

Асоціація

Візуалізація

Відхилення та аналіз зв’язків

Оцінювання

Аналіз зв’язків

Підведення підсумків

Класифікація це найпростіша і поширена задача в Data Mining. Після розв’язання цієї задачі виявляються ознаки, які в свою черху характеризують групи об’єктів даних, які досліджуються – класи; за такими ознаками нові

об’єкти можна віднести до того чи іншого класу. Для розв’язання такої задачі використовуються, такі методи: найближчого сусіда (Nearest Neighbor), k-найближчого сусіда (k-Nearest Neighbor); байєсовські мережі (Bayesian Networks); індукція дерев рішень; нейронні мережі (neural networks).

Кластеризація, це логічне продовження класифікації, але ця задача більш складна. Особливістю кластеризації є те, що класи об’єктів спочатку не визначені і результатом кластеризації є розбиття об’єктів на групи.

Асоціація. Дана задача відрізняється від двох попередніх задач. Data Mining полягає в тому, що тут не  на основі властивостей аналізованого об’єкта відбувається пошук закономірностей здійснюється, а між кількома подіями, які відбуваються одночасно.

Послідовність. Вона дає можливість знайти тимчасові закономірності між транзакціями. Задача послідовності близька до асоціації, але її метою є встановлення закономірностей не між подіями, що наступають одночасно, а між подіями, які пов’язаними в часі.

Прогнозування. Під час опрацювання данної задачі на основі особливостей історичних даних оцінюються пропущені або ж майбутні значення цільових чисельних показників. Мета розв’язання задачі з визначення відхилень, аналіз даних, які найбільше відрізняються від інших даних.

Задача оцінювання, це передбачити неперервні значення ознаки. Задача аналізу  зв’язків  це знайти залежність в наборі даних.  В результаті візуалізації ,створюється графічне відображення аналізованих даних.

Головною цінністю Data Mining , є спрямованість даної технології, шлях від сирих даних до отримання конкретних результатів, від постановки завдання до готової системи, з підтримкою якої можна приймати рішення. Велика кількість понять, які об’єднав Data Mining, та різноманітні методи, які підтримують дану технологію, напочатку можуть нагадати мозаїку, частини якої мало пов’язані між собою. Щоб зв’язати в одне ціле закономірності, додатки, дані, інформацію, рішення, розглянемо два потоки «Дані – інформація – знання і рішення»

На рис. 2.3.1 показано зв'язок понять «дані», «інформація» і «рішення», яка виникає в процесі прийняття рішень.

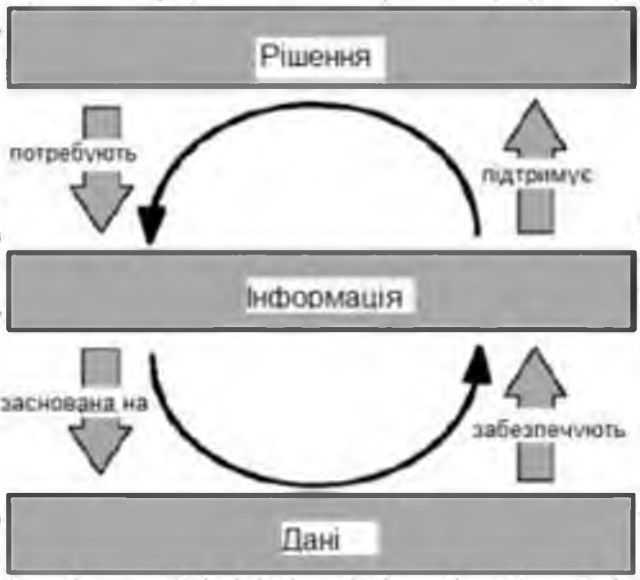



Рис. 2.3.1 Рішення інформація та дані.

Як видно з рисунку, це циклічний процес. Прийняття рішень потребує інформації, яка заснована на даних. Дані забезпечують інформацію, яка підтримує рішення і т.д.

Як висновок, можна сказати, що Data Mining, це круговорот взаємодоповнюючих і дослідницьких кроків, які допомагають ідентифікувати, вирішити та визначити нову задачу, а результати процесу аналізу даних становляться базою для нового бізнес-задач.

Тому в аналізі даних дуже важливодотримуватися двох принципі, це визначати чіткі запити для пошуку та розбиратись в своїх даних.

**2.4** Вибір технологій для створення системи

Для розробки даної систем необхідно було обрати певний стек технологій, які б допомагало в управлінні командною роботою для розробників. Щоб окремі підсистемипрацювали як однієї цілої системи, та користувачі могли використовувати програмний продукт в повній мірі, та реалізувати можливість використання системи в складі декількох користувачів, які могли б

працювати незалежно один від одного та вносити зміни – на сервері необхідно розгортувати базу даних.

Для реалізації даного програмного забезпечення, використано такий стек технологій та інструментарій:

мова програмування PHP;

фреймворк Laravel;

база даних MySQL;

веб-додаток phpMyAdmin;

сервер OpenServer;

мова програмування Java Script;

фреймворк Bootstrap;

Так як було обрано реалізацію системи у вигляді веб-додатку, мовою програмування було обрано PHP, яка є класичною для розробки в веб.

PHP, це скриптова мова програмування,як призначена для генерації HTML-сторінок на стороні сервера. PHP інтерпретується веб-сервером в HTML-код, який передається в браузер.

Розглядаючи різні напрямки використання данної мови програмування, можна виділити основні переваги:

Традиційність;

Простота;

Ефективність;

Безпека;

Гнучкість;

Для того щоб полегшити процес розробки та покращити якість розроблюваної системи, було обрано один з найпопулярніших фреморків, Laravel.

Laravel, це PHP фреймворк з виразним і елегантним синтаксисом. Він дозволяє спрощувати певні рішення при вирішеній актуальних завдань, таких як аутентифікація, маршрутизація, сесії і кешування. Laravel, об'єднує все найкраще, що є в інших PHP фреймворках.

Основні переваги Laravel:

підвищена продуктивність;

аутентифікація;

відкритий код і велике ком'юніті;

міграції баз даних;

MVC-архітектура;

швидкість розробки.

Фундаментальною частиною нашої системи є база даних, для цього обрана база даних MySQL з якою гарно взаємодіє обрана мова програмування та веб-інтерфейс phpMyAdmin для адміністрування бази даних.

MySQL - це система управління реляційними базами даних. В реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому є первага в швидкості та гнучкості.

Таблиці в даній базі даних зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL це як частина системи MySQL її можна характеризувати як мову структурованих запитів, яка використовується для доступу до баз даних.

Для розробки інтерфейсу системи використано JavaScript, Bootstrap, HTML, CSS.

JavaScript – це об'єктно-орієнтована мова програмування, вона дає можливість взаємодіяти з користувачем на стороні клієнта, керувати браузером, обмінюватися даними з сервером асинхронно та змінювати структуру та зовнішній вигляд веб-сторінки.

Bootstrap – це CSS фреймворк, він дає можливість адаптивно відображеати контент веб-сторінки на різних розширеннях екрану. За основу цього фреймворку взята модульна сітка яка складається з 12 колонок, вона вміє змінювати розташування контенту в залежності від заданих параметрів для елементів та розміру екрану пристрою користувача нашої системи.

НУБіП України

Також, Bootstrap має безліч стилів оформлення елементів HTML структури таких як: кнопоки, форми, спливаючі вікна, навігаційна панелі і т.д.

Одним з методів аналізу було обрано алгоритм 1-R. Даний алгоритм потрібен для формування елементарних правил для класифікації об'єкта. Він будує правила за значенням однієї незалежної змінної, тому називається 1R(rule)-алгоритм.

Ідея роботи даного алгоритму заключається у тому, що з будь-якого можливого значення кожної незалежної змінної формується правило, яке класифікує об'єкти з навчальної вибірки, при цьому в заключній частині правила вказується значення залежної змінної, яке найчастіше зустрічається у об'єктів із вибраним значенням незалежної змінної. У цьому випадку помилкою правила є кількість об'єктів, що мають те ж значення змінної, що розглядається, але не відносяться до обраного класу. Таким чином, для кожної змінної буде отримано набір правил для кожного значення. Оцінивши ступінь помилки кожного набору, вибирається змінна, на яку побудовані правила з найменшою помилкою.

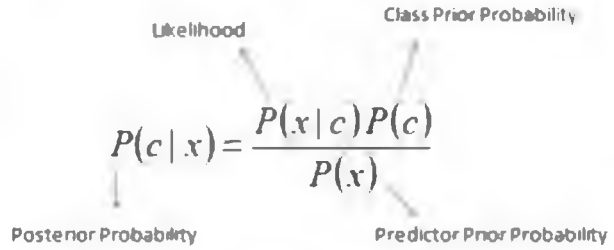
Для рішення задачі класифікації з врахуванням значень декількох незалежних змінних було використано метод Наївного Байєса.

Даний метод потрібен для формування елементарних правил для класифікації об'єкта. Він будує правила за значенням кількох незалежних змінних, використовуючи формулу Байєса до розрахунку ймовірності.

Моделі на основі Наївного Байеса досить прості і вкрай корисні при роботі з великими наборами даних. При своїй простоті Наївного Байеса здатний перевершити навіть складні алгоритми класифікації.

Теорема Байєса дозволяє розрахувати апостеріорну ймовірність P(c|x) на основі P(c), P(x) та P(x|c).

НУБіП України



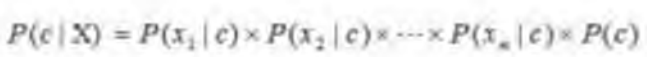




*Рис. 2.4.1 Формула Байеса*

На Рис. 2.4.1 передставлено :

- P (c | x) - апостеріорна ймовірність даного класу c (тобто даного значення цільової змінної) при даному значенні ознаки x.
- P (c) - апріорна ймовірність цього класу.
- P (x | c) - ймовірність даного значення ознаки при даному класі
- P (x) - апріорна ймовірність даного значення ознаки.

Також однією з технологій аналізу даних було обрано KPI.

KPI (ключові показники ефективності) – це показники які показують діяльності підрозділу, допомагають при організації у досягненні стратегічних та тактичних цілей. Використання ключових показників ефективност дає підприємству можливість оцінити свій поточний стан та допомогти в оцінці успішності реалізації обраної стратегії.

KPI дозволяють виробляти контроль ділової активності співробітників, підрозділів та компанії в цілому. За допомогою KPI можна створити досконалу та ефективну систему мотивації та стимулювання співробітників компанії, тому KPI та мотивація персоналу стали нерозривними поняттями.

В залежності від стратегії компанії розрізняють різні KPI. Здебільшого їх застосовують для визначення результативності роботи адміністративно-управлінського персоналу.

Ключові показники ефективності можна розділити на:

- запізнілі – відображають результати діяльності після закінчення періоду;
- випереджаючі - дають можливість керувати ситуацією в межах звітного періоду з метою досягнення заданих результатів після його закінчення

# 3. РОЗРОБКА СИСТЕМИ

## 3.1 Архітектура системи

Архітектура програмного забезпечення, це спосіб структурувати програбну або обчислювальну систему, абстрактно представити елементи системи на певній фазі її розробки. Зазвичай система складається з декількох рівнів абстракції та має багато фаз для її роботи, кожна з яких має індивідуальну архітектуру.

НУБіП України

Дослідження архітектури системи потрібно, для того, щоб визначити як найкраще розбити систему на підсистеми, як підсистеми визначаються та взаємодіють між собою яким способом відбувається передача інформації, та як ці підсистеми розвиваються поодинці.

Архітектура потрібно будувати таким чином, щоб найкраще відповідати визначиним вимогам до системи що створюється, згідно принципу "форма відповідає функції, тобто архітектура це набір елементів які мають певну форму, властивості, обмеження які накладаються на елементи, та їх обґрунтування. Обґрунтування відображає мотиви вибору того чи іншого архітектурного рішення, елементів та обмежень.

Основною архітектурою розроблюваної системи є «Модель-Вид-Контролер», яка зображена у вигляді діаграми компонентів на Рис. 3.1.1.

Діаграми компонентів відображає залежність між програмними компонентами, які виникають на етапі компіляції або в процесі роботи програми. На діаграмі компонентів відображають входження об'єктів та класів в програмні компоненти системи, наприклад, модулі, бібліотеки і т.д. На головній діаграмі компонентів зазвичай відображають головні пакети для системи. Компоненти вихідного коду - це програмні файли, які знаходяться всередині пакетів. Компонента, це вихідний код,він може бути бінарним кодод або run-time об'єкт.

На Діаграмі компонентів також відображають залежності між компонентами програми та їх типи, які виникають під час виконання програми або в процесі компіляції, зазвичай це зв'язок файлів коду з бібліотеками DLL.

НУБіП України

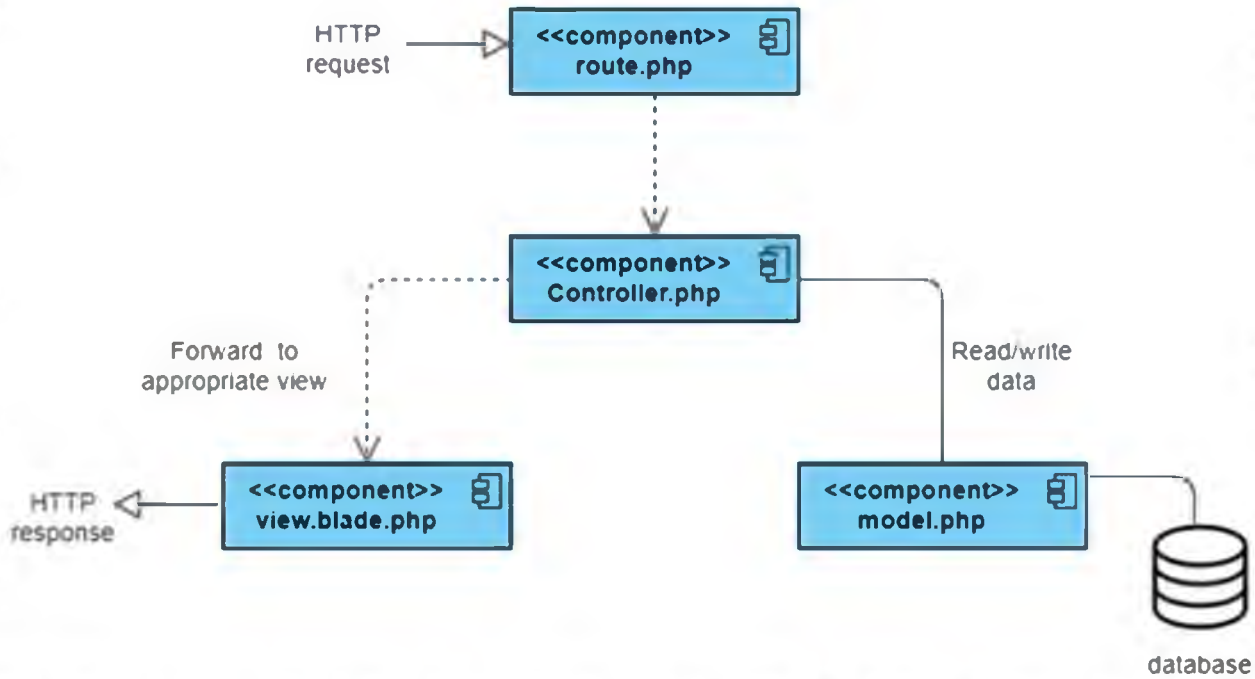



*Рис. 3.1.1 Діаграма компонентів*

Розберемо по частинам схему роботи MVC. Шаблон MVC має три основних компоненти: Модель, Представлення, та Контролер.

Модель. Модель є фундаментальною частиною системи, яка відповідає за алгоритми всередині системи, процес розрахунків та внутрішню організацію системи.

Представлення. Модуль який відпвідає за виведення інформації. Зазвичай це шаблонізатор або альтернативні рішення, основною ціллю яких є тільки інформації у форматі HTML на основі будь-яких готових даних.

Контролер. Модуль який управляє введеннямтаі виведенням даних.

Даний модуль стежить за передачу в систему данич, будь-яким способом, наприклад, через форму, рядок запиту, cookie і т.д. та на основі цих даних вирішує:

Передавати ці дані у модель чи ні;

Відобразити помилку та відправити запит на повторне введення;

Крім того, контролер обов'язково повинен визначати тип отримних даних від моделі (чи це остаточний результат, та немає повідомлення про помилку) та передавати інформацію в модуль представлення.

НУБіП України

Модель. Модуль, який відповідає за розрахунок будь-яких алгоритмів на даних, які були отримані від користувача. Наступним етапом цього модуля є передача в контролер, без вмісту інформації, що відноситься до безпосереднього висновку (тобто має бути представлений у внутрішньому форматі програми).

Дана система, буде розроблюватись, як Web-додаток. Веб-додаток від різняється від локальних додатків які мають тільки один шару , він розмішений на комп'ютері користувача. Такі додатки мають поділ на логічні частини(шари), кожен з шарів має свою роль. Зазвичай додатки мають поділ на три основних шари:

уявлення;

бізнес-логіка;

база даних;

Веб-додаток - програма, яка виконується, використовуючи веб-браузер у якості свого клієнта. Він може бути різного рівня складності, як дошка оголошень або лендінг з формою зворотнього зв'язку, та такий складний, як банківський сервіс або ігровий додаток.

Логіка таких додаткім має розмоділ між клієнтом та сервером та зберігає свої дані, зазвичай, на сервері, а обмінюється інформацією за допомогою мережі.

В результаті основною перевагою, такого додатку є не залежність користувача від конкретної операційної системи, тому що веб-додатки є кросплатформенними сервісами.

В архітектурі розроблюваної системи, у якості робочого місця використовується браузер. Таке рішення не вимагає, спеціальних налаштувань на стороні користувача. Він лише може надіслати заповнену форму реєстрації або авторизації, та очікувати відповіді від додатку.

В якості серверної частини системи використовується одна із сучасних мов програмування PHP.

НУБіП України

Обрана архітектура програмного забезпечення представлена на Рис.3.1.2, у вигляді діаграми пакетів на якій представлені два рівні:

рівень бізнес-логіки, та реалізація доступу до даних, що включає компоненти які реалізують доступ до даних;

Користувацький рівень який включає логіку представлення та компоненти інтерфейсу.

Діаграмою пакетів є діаграма, що містить пакети класів та залежності між ними. Тобто, пакети та залежності між ними, це елементи діаграми класів, тобто діаграма пакетів, це лише форма діаграми класів. Однак виходячи з практики є різні причини побудови таких діаграм.

Залежність між двома елементами може бути тільки в тому випадку, якщо зміни одного елемента, якимось чином можуть внести зміни в іншому. У класів, причини для залежності можуть бути різними: один клас може посилати повідомлення для іншого; один клас включає частину даних з іншого класу; один клас посилається на інший як параметр операції. Повідомлення, яке посилає клас може стати не коректним, якщо він змінив інтерфейс.

В найкращому випадку, тільки зміни в інтерфейсі класу повинні мати вплив на інші класи. Правильне проектування великих систем включає в себе мінімізацію залежностей, яка зменшує кількість зміни та потребує менше зусиль на їх внесення.

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

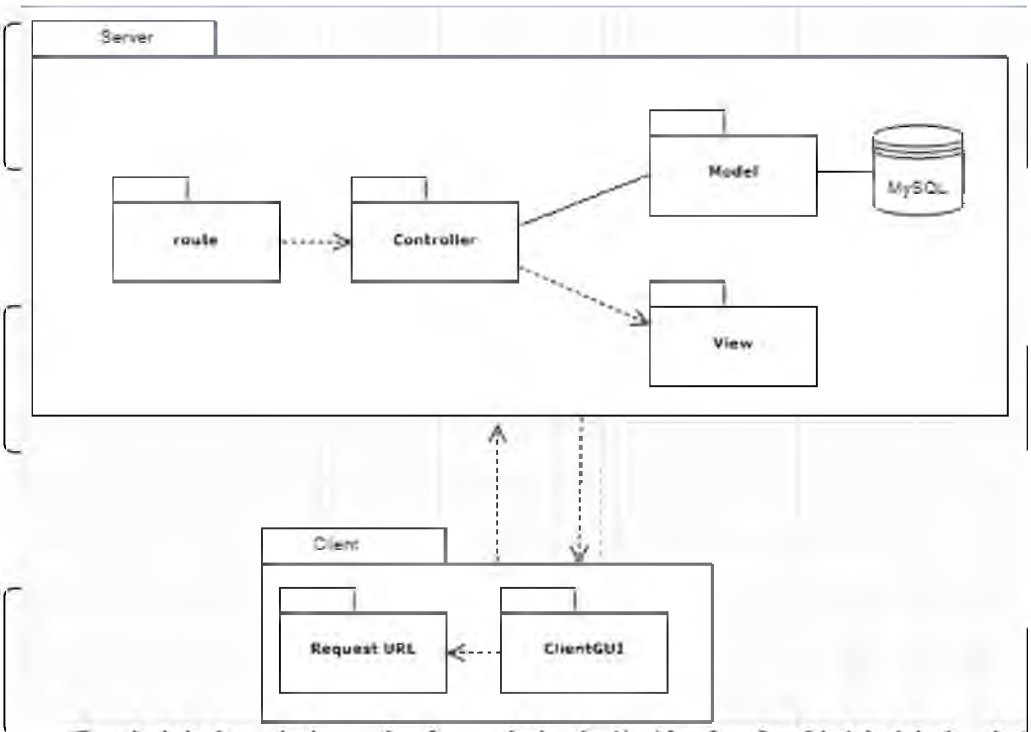



*Рис. 3.1.2 Діаграма пакетів*

Найбільш важливим аспектом розробки системи є те, як взаємодіють між собою окремі компоненти так учасники системи. Конфігурація системи міжкомпонентного зв'язку дуже впливає на порядок підключення учаників до системи, та які з'єднують окремі компоненти. Організація внутрішніх комунікацій системи називається топологією. Топологію розроблюваної системи представлено на Рис.3.1.3.

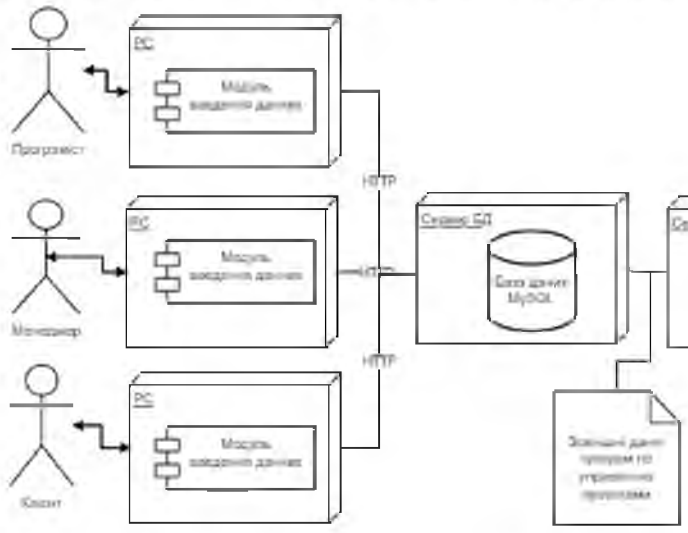



Рис. 3.1.3 Топологія системи

На цій діаграмі, ми можемо спостерігати, що система для аналізу ефективності роботи в команді складається з клієнта котрий надає інформацію згідно якої потрібно розробити проект, менеджер приймає ТЗ, розбиває на задачі та розподіляє між програмістами.

Програміст отримує задачу, виконує та відправляє на перевірку. Данні які надають ці об’єкти заносяться в БД, після чого частина даних передається в СД. Тому що для аналізу аналітик бере тільки певні данні. Аналіз даних відбувається за допомогою OLAP- технології.

Одним з основних етапів життєвого циклу розроблюваної системи є проектування бази даних. Перщим етапом проектування є створення інформаційно-логічної моделі даних відповідно до предметної області, яка основується на описі предметної області, отриманому під час її обстеження. Розробка логічної моделі даних має велику роль під час розробки. Логічна модель даних – модель відображає сутності та зв’язки між ними.

В процесі проектування бази потрібно визначити структуру бази, тобто склад таблиць, структуру таблиць та зв'язки між ними. Процес проектування можна розділити на два підходи. Перший підхід, це визначення основних задач, для розв'язку яких створюється база, таяких даних потребують ці задачі. Другий підхід, це визначення предметну область , здійснити аналіз даних та встановити типові об'єкти предметної області. Кращим варіантом проектування бази даних, це поєднати ці підходи

Після того, як визначили основні дані, які будуть використовуватись в системі, наступним етопом є створення структуру таблиць, сюди входять такі етапи:

1. Визначети поля для таблиць баз даних;
2. Визначити типи даних для полів таблиць;
3. Визначити ключові поля;
4. Визначити зв’язки між таблицями

НУБіП України

Даний етап проектування відбувався за допомогою програми ERwin до складу якої входять засоби для побудови згідно стандарту IDEF1x. Erwin дозволяє розробити логічну модель баз даних за стандартом IDEF1x і на її основі створити фізичну модель даних, яка підійде майже для всіх СУБД. Також можна зробити зворотне перетворення, фізичної моделі в логічну.

За допомогою даної програми ми отримали логічну модель баз даних у вигляді діаграми прив'язки сутї ER. ER-модель – це семантична модель даних, яка потрібна для того, щоб спростити процес проектування бази даних. З ER-моделі можуть бути породжені всі види бази даних: реляційні, ієрархічні, мережні, об'єктні. ER-модель основана на таких поняттяч, сутність, зв'язок та атрибут.

Для великих баз даних побудова ER-моделі дозволяє уникнути помилок проектування, які дуже складно виправити, особливо якщо база даних вже в експлуатації або на стадії тестування. Помилки які були допущені при розробці структури бази даних можуть призвести до зміни програмного коду, який керує цією базою даних. В результаті чого, буде неефективно витрачено, час, засоби та людські ресурси.

ER-модель, дає можливість представити базу даних як наочних графічних діаграм. ER-модель відображає процес, який відбувається у деякій предметній області. Діаграма "сутність"-"зв'язок" - це діаграма, яка представляє в графічному вигляді сутності, атрибути та зв'язки.

Для побудови логічної моделі даних розроблюваного програмного забезпечення, виділено три основних таблиці: задачі (objectives), проекти (projects) та користувачі (users) які представленні на Рис. 3.1.4.

НУБіП України

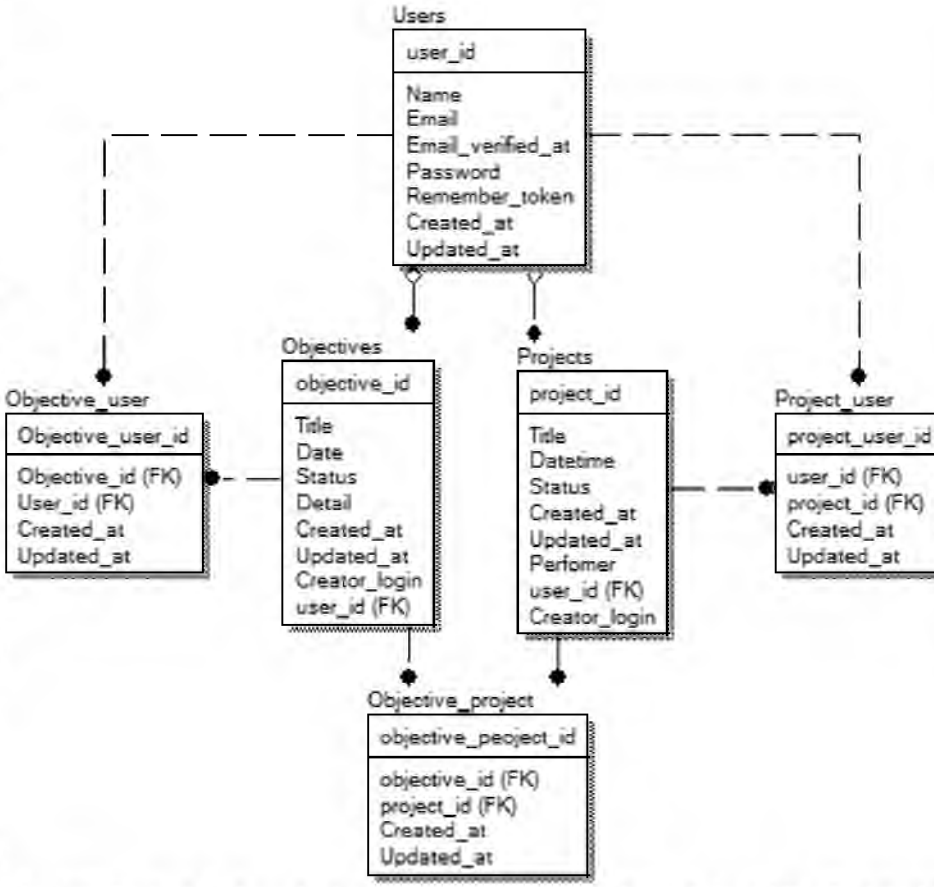



*Рис. 3.1.4 ER-діаграма.*

3.2 Система

З урахуванням вимог технічного завдання проектується відповідний інтерфейс користувача (Рис. 3.2.1-3.2.13), відповідно цьому інтерфейсу відбувається програмування окремих модулів розроблюваної системи. І як результатом розробки правильних алгоритмів, буде отримано якісний продукт.

НУБіП України

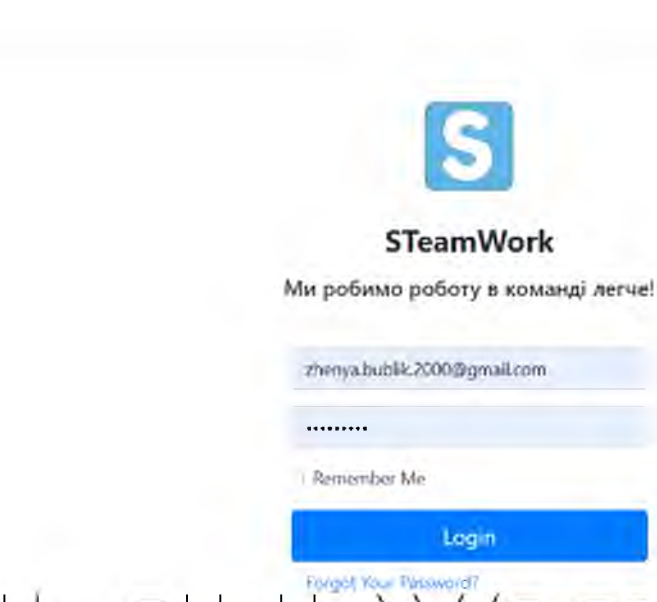



*Рис. 3.2.1 Форма авторизації.*

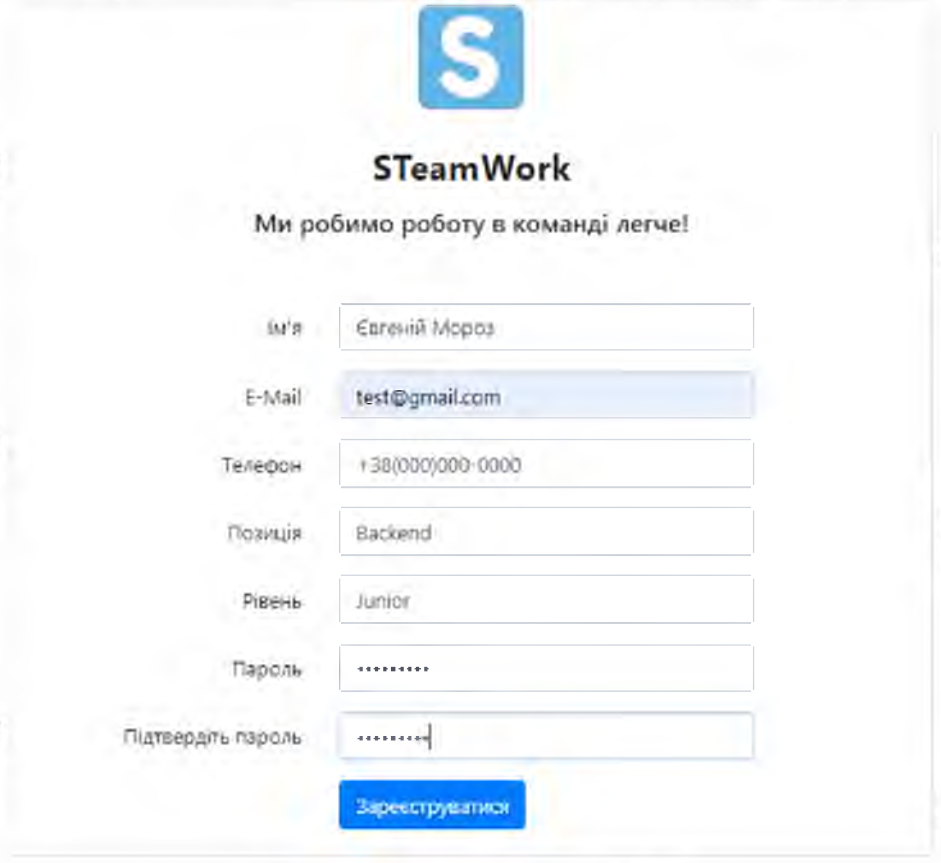



*Рис. 3.2.2 Форма реєстрації.*

На рис. 3.2.1-3.2.2 зображено розроблену форму аторизації та реєстрації, відповідно технічному завданню. Форма авторизації містить в собі два поля для аутентифікації користувача, це пошта та пароль. Форма реєстрації складається з

імені, пошти, телефону, позиції розробника, його рівня, паролю, та поля для повторного вводу пароля.

Серверна частина авторизації та реєстрації реалізується за допомогою вбудованої команди фреймворка laravel команди php artisan make:auth. При виконанні данної команди створюється контроллер, модель та відповідна міграція, що створює таблицю в базі даних з відповідними полями. Логотип веб-додатку розроблений за власним дизайном та за допомогою онлайн сервісу Desygner який надає всі необхідні інструменти для створення логотипів, та все можливі шаблони.

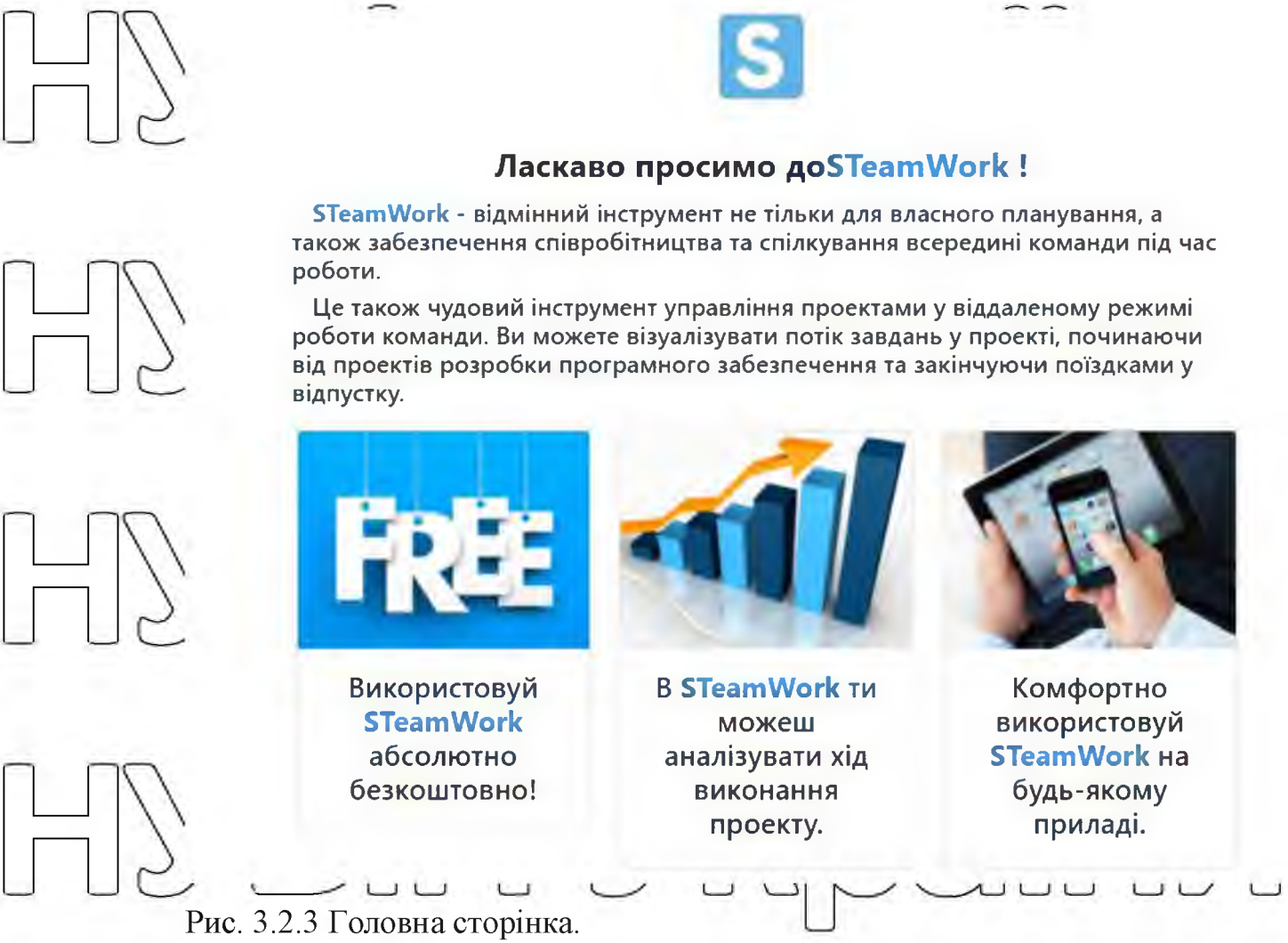



Рис. 3.2.3 Головна сторінка.

На рис. 3.2.3 представлено головну сторінку, куди користувач потрапляє після успішної авторизації в системі, та може ознайомитись з головними перевагами даної системи. Візуальна частина всіх сторінок створена за допомогою компонентів bootstrap, які дозволяють адаптивно відображати

контент на будь-якому розширенні екрану за системою сітки bootstrap.

Позначення у веб-додатку, які зображенні у вигляді іконок, створенні за допомогою font awesome. Font Awesome - це шрифт, створений за допомогою іконок, але це не набір картинок.

Переваги Font Awesome:

Векторний - це означає, що він дуже гарно виглядає неважливо, в якому розмірі він відображається.

Легко налаштувати – можна легко змінювати його, як і будь-який інший шрифтом. Можна змінити колір додати анімацію та інше.

Крос-браузерність - тобто шрифт працює практично в будь-якому браузері.

Також після авторизації в верхній навігаційній панелі з'являється аватар користувача, якщо він завантажений, ім'я та випадаючий список з двома кнопками профіль, який веде на сторінку профілю користувача та вихід, який призводить до виходу з системи.

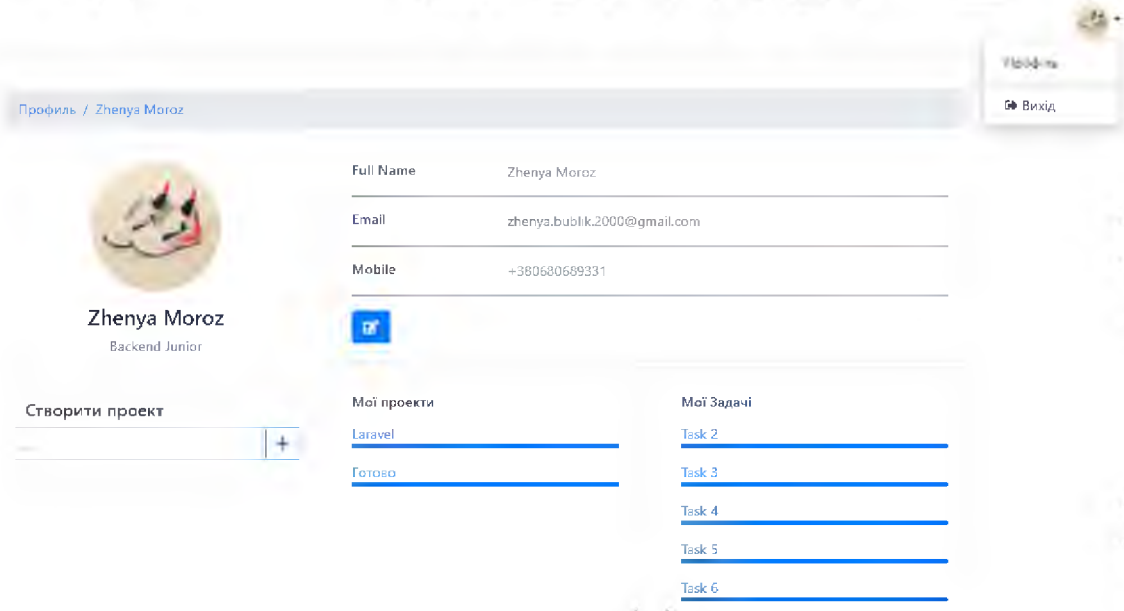



*Рис. 3.2.4 Сторінка профілю користувача.*

На рис. 3.2.4 зображено сторінку профілю користувача. На ній виводиться контактна інформація користувача, ім'я, пошта, позиція, рівень, список проектів в яких приймає участь користувач, список його задач, з можливістю переходу на

детальну сторінку, можливість створення проекту, для цього достатньо лише ввести назву проекту.

Також є можливість редагування профілю користувача яка зображена на рис. 3.2.5, редагування профілю відбувається у випадаючому вікні, в яком є відповідні поля для зміни інформації про профіль.

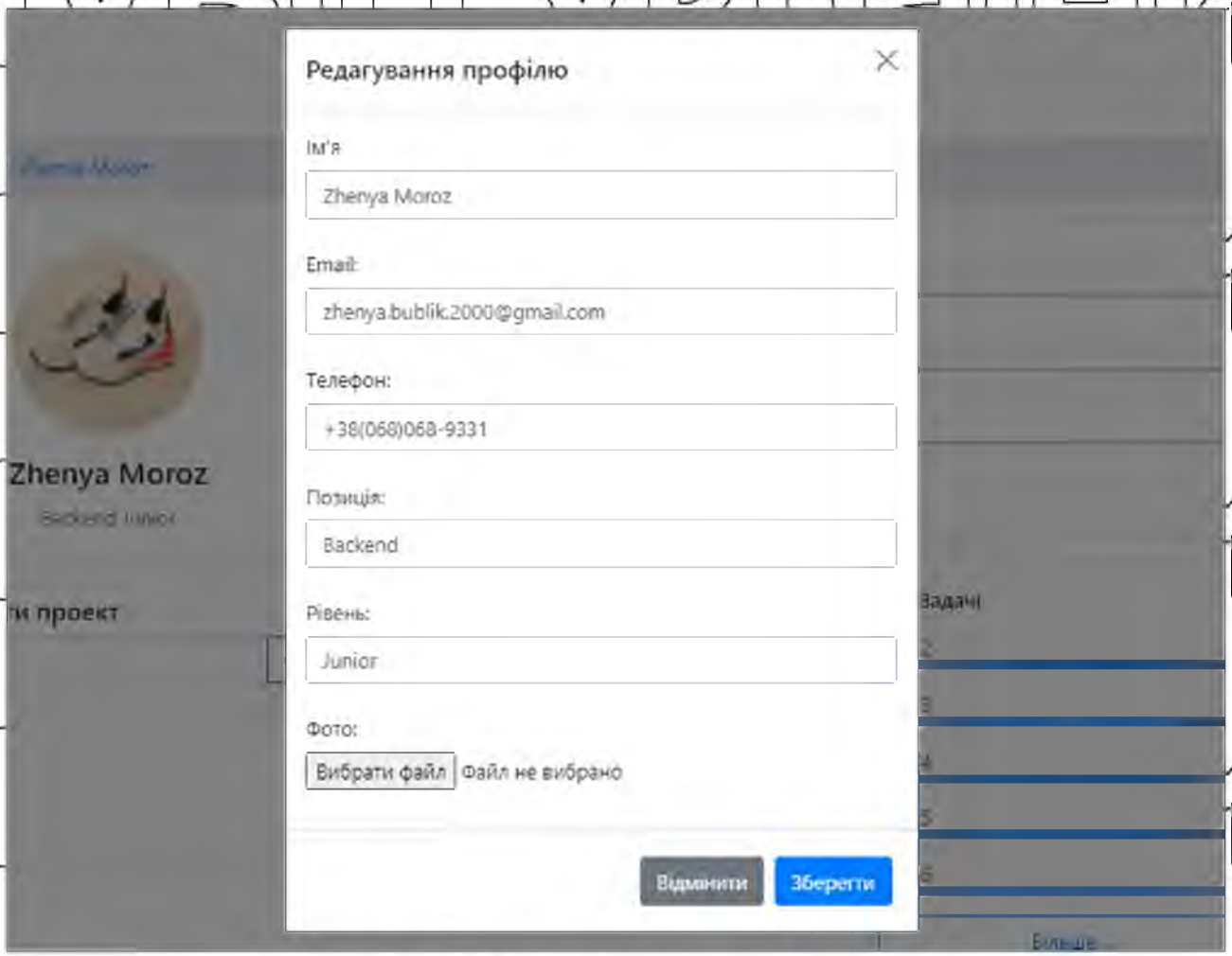



*Рис. 3.2.5 Редагування профілю.*

НУБіП України

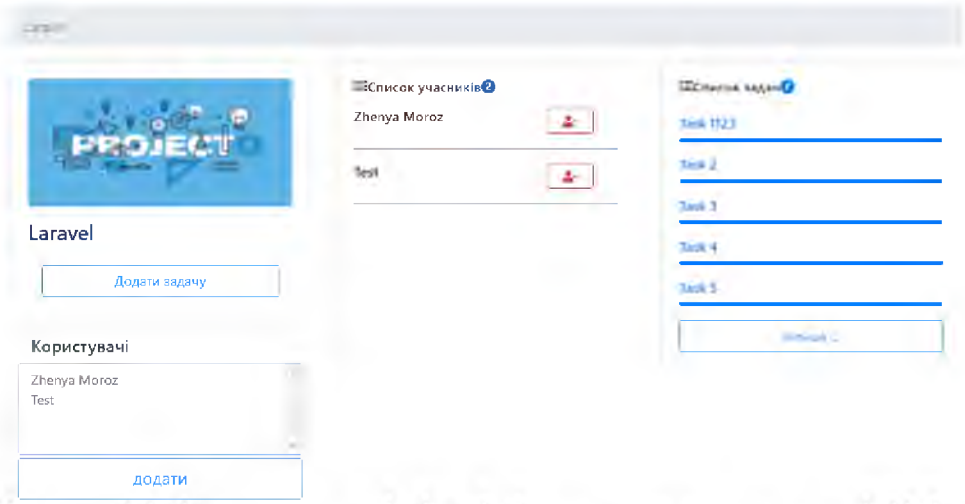
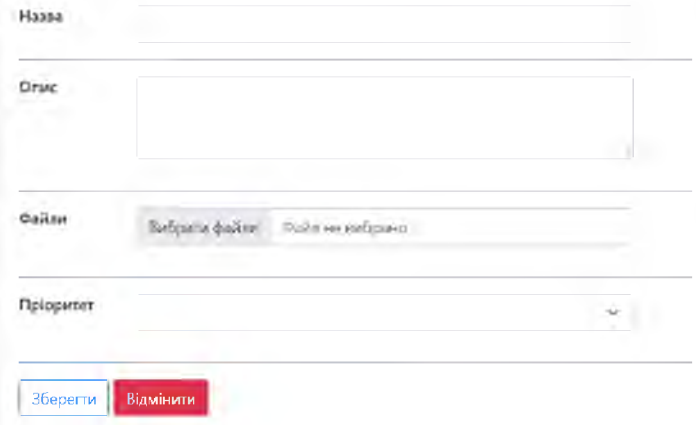



*Рис. 3.2.6 Детальна сторінка проекту.*

На рис. 3.2.6 зображена детальна сторінка проекту на якій відображається список прикріплених до проекту учасників у вигляді списку з можливістю видалення з проекту, список задач з можливістю переходу на детальну сторінку або цілого списку

Проект відображається у вигляді картечки bootstrap. Він містить стандартну картинку при створені проекту, без можливості її зміни та назву проекту. Також є зручна кнопка створення задачі з ховер ефектом, яка веде на сторінку з формою створення задачі. Форма створення задачі зображена на рис 3.2.7, має поля для вводу таких даних: назва, опис, прикріплення файлів та виставлення пріоритетів.




НУБіП України

*Рис. 3.2.7 Сторінка створення задачі.*

На рис. 3.2.8 зображено детальну сторінку задачи, яка містить таку інформацію про задачу:

Назва;

Автор;

Виконавець;

Прикріплені фали;

Статус задачі;

Пріоритет задачі;

Рівень задачі.

На сторінці також є список задач користувача з можливістю переходу на їх детальні сторінки, можливість створення задачі та можливість редагування інформації по задачі.

Також є можливість ведення журналу часу, тобто запис часу користувачем, який він потратив на виконання задачі, та можливість залишати коментарі в задачі.

НУБіП України

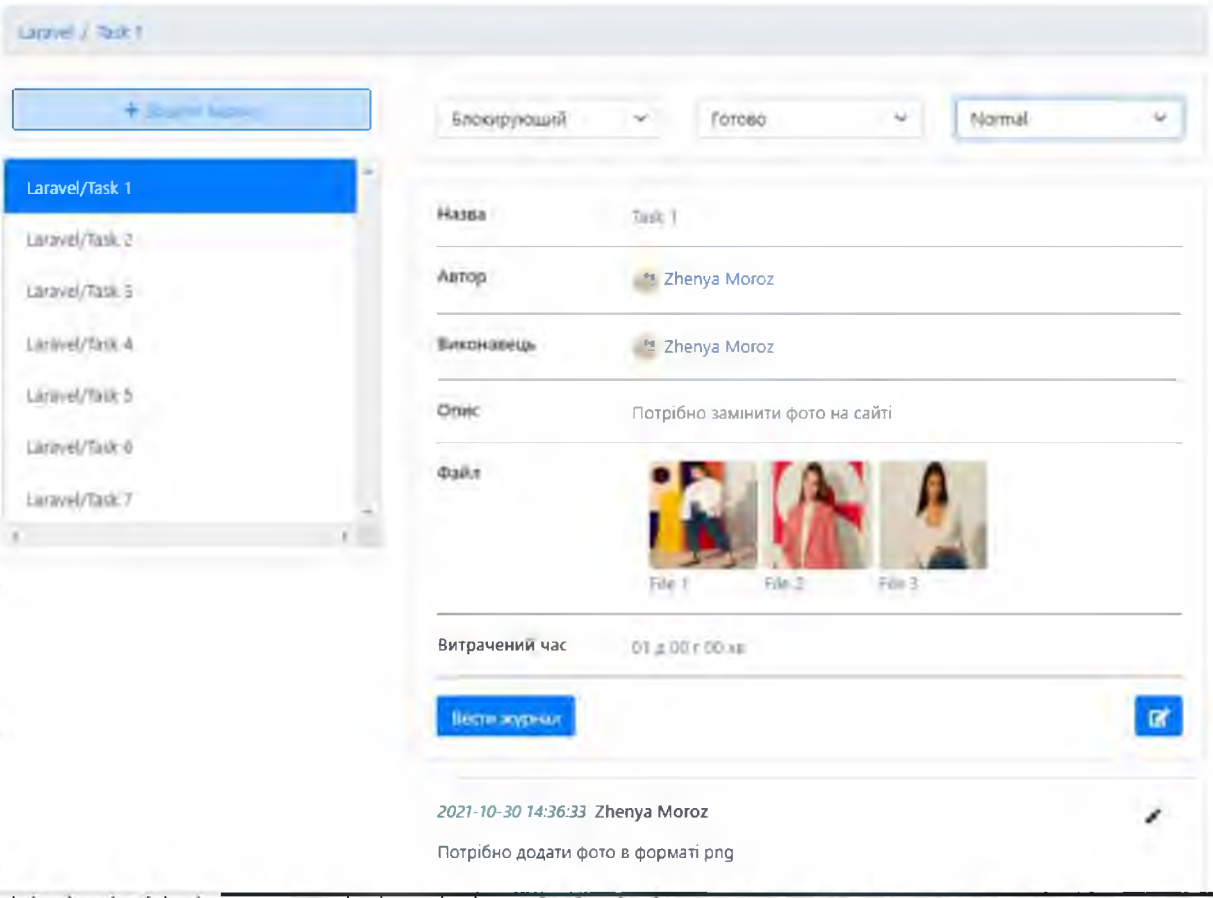



*Рис. 3.2.8 детальна сторінка задачі.*

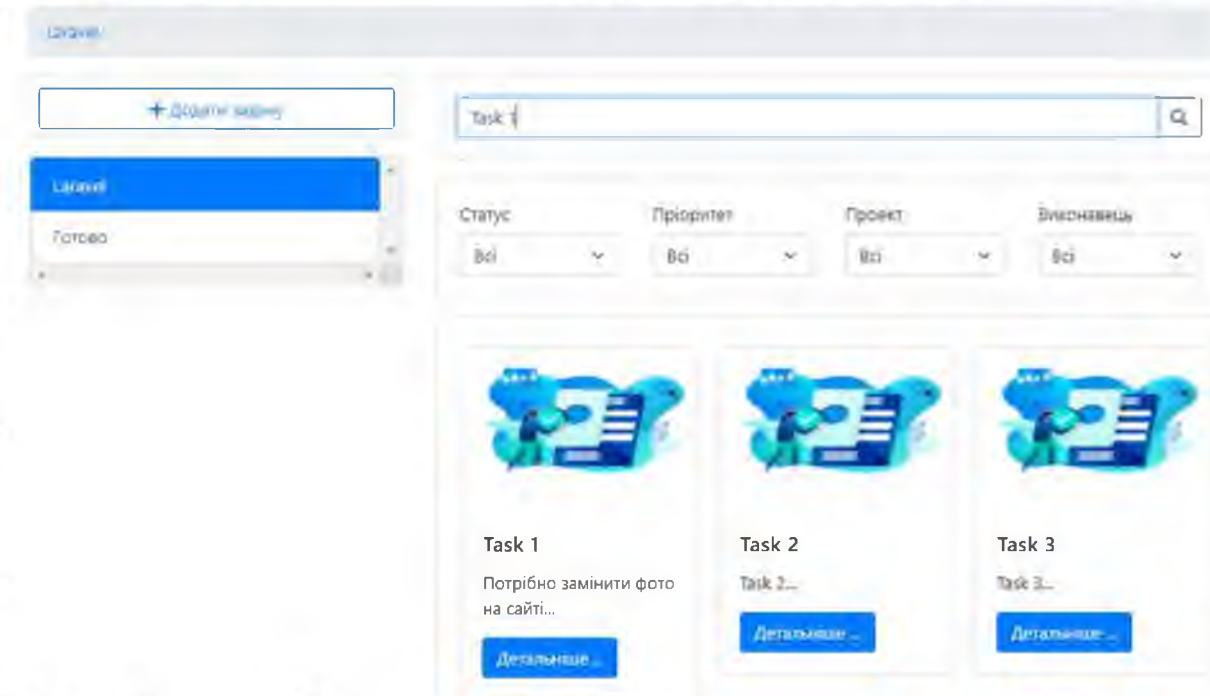


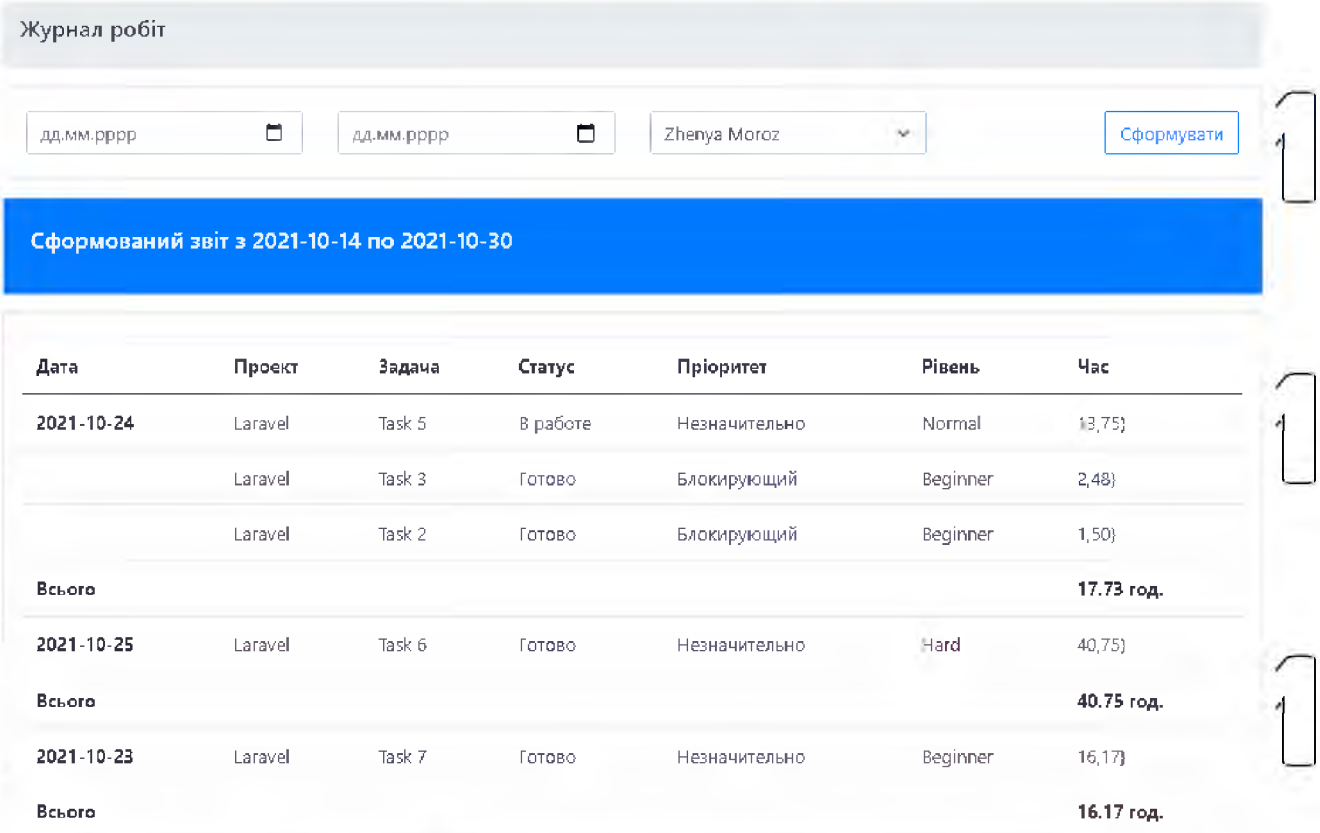

*Рис. 3.2.9 Сторінка списку задач.*

На рис 3.2.9 зображено сторінку списку задач, на даній сторінці реалізовано можливість пошуку задачі та сортування задач по таким критеріям:

НУБіП України

- Статус;
- Пріоритет;
- Проект;
- Виконавець;

На сторінці також є можливість швидкого переключення між проетами та додавання нової задачі до проєкту. Всі задачі відображаються у вигляді карточок, які містять: стандартну картинку, початок опису та можливість переходу на детальну сторінку.




*3.2.10 Сторінка журналу робіт.*

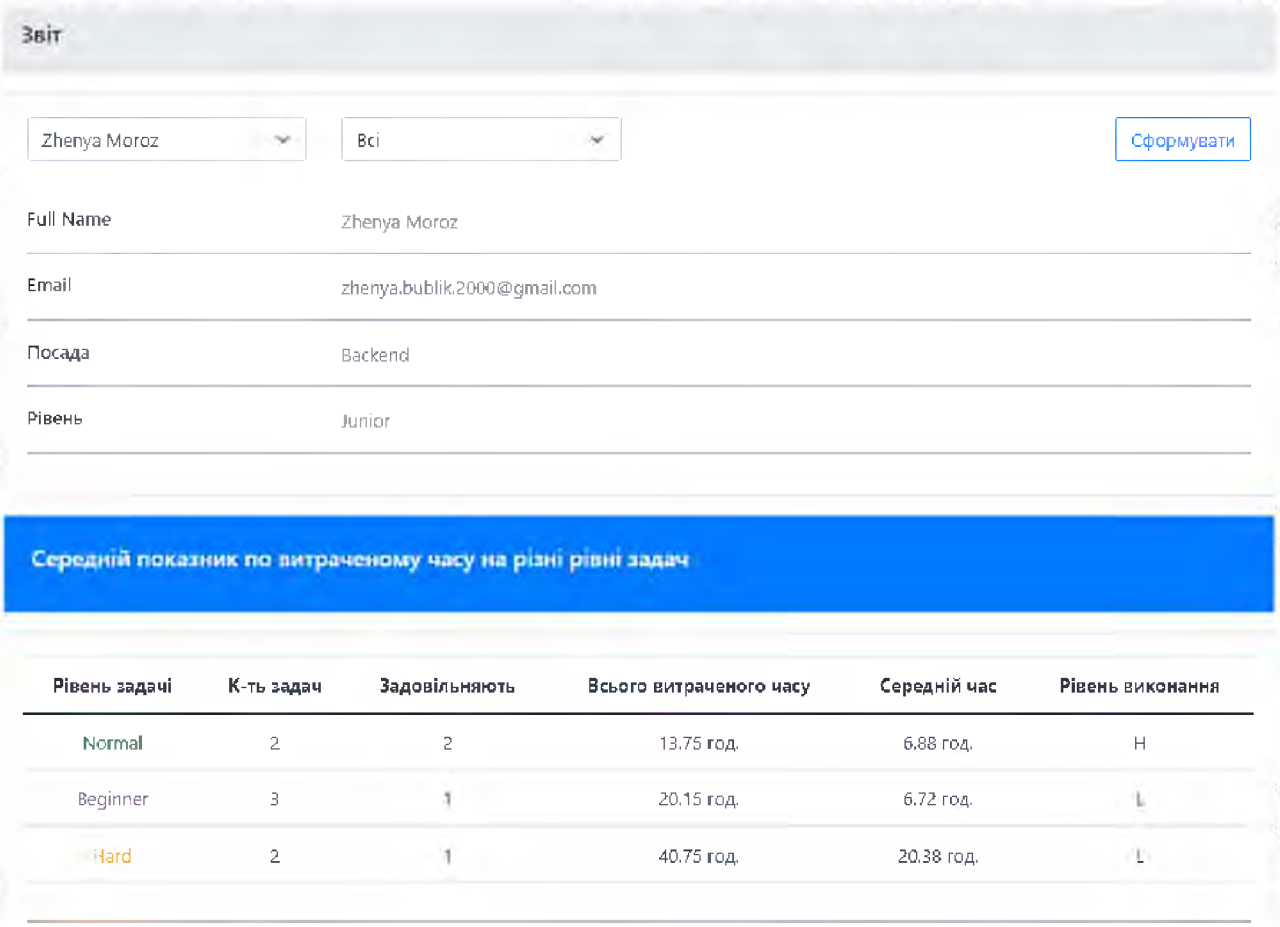
Для того щоб користувач міг відстежити кількість витраченого свого часу або іншого учасника, було створено сторінку з можливістю фільтрування по даті з якого по який період, та по учасникам рис. 3.2.10. Звіт будується у вигляді таблиці з такими даними:

- Дата;
- Проект;
- Статус задачі;

- Пріоритет задачі;
- Кількість часу витраченого на задачу;
- Кількість витраченого часу користувачем загалом за день;

Також було створено сторінку статистики виконаних задач, яка зображена на рис. 3.2.11. На цій сторінці розробник або проектний менеджер має можливість побудувати методом Наївного Байєса звіт по кожному учаснику з можливістю фільтрування по рівню задачі. Після створення звіту, ми отримуємо повну інформацію по розробнику, та таблицю з розрахованими даними, в яку входять:

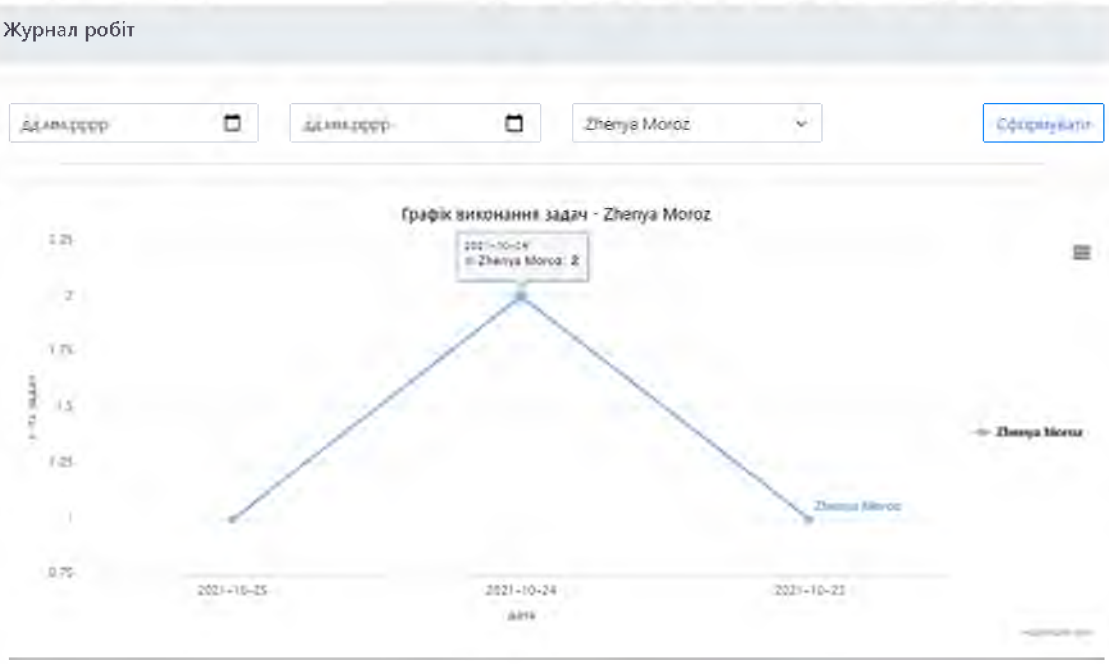
- Рівень задачі;
- Кількість задач;
- Кількість задач, які задовольняють свою оцінку;
- Кілкість витраченого часу на всі задачі певного рівня;
- Середня кількість витраченого часу на задачі певного рівня;
- Рівень виконання задач;




НУБіП України

*3.2.1 Сторінка статистики виконаних задач*

Також розробник або проектний менеджер, може побудувати звіт у вигляді графіку на якому зображено кількість виконаних задач, в пений проміжок часу певним учасником проекту. Даний звіт можна вивантажити у форматі excel або в різних форматах зображень. Сторінка зображена на рис 3.2.13.




*3.2.13 Сторінка графіку виконаних задач.*

## 3.3. Сховище даних

Основним завданням сховища даних є організація єдиного місця збору і зберігання даних.

В основі створення сховища даних лежать дві основні ідеї.

1. Інтеграція роз'єднаних деталізованих даних в одному сховищі. У процесі інтеграції повинно виконуватися узгодження деталізованих даних та, їх агрегація. Дані можуть потрапляти в сховище даних з оперативних баз даних, зовнішніх джерел.

2. Дилити набір даних та програм, які використовуються для оперативного опрацювання даних та які застосовуються для вирішення завдань аналізу.

На даний момент часу розрізняють два види СД, це віртуальне та класичне (фізичне).

Класичне сховище даних є найпоширенішим та унікальним репозиторієм інформації які містять дані з оперативних джерел. Дана середа сховища призначена тільки для режиму читання та складається з агрегованих даних та детальних даних, які повністю інтегровані та очищені, також в ньому зберігається вся детальна історія даних. Сховище даних, з точки зору архітектури реалізує свої функції перш за все через підмножину залежних вітрин даних .

Вітрина даних, це спеціальне локальне тематичне сховище, яке обслуговує окремий напрям діяльності.

До переваг архітектури класичного сховища даних входять:

- несуперечність інформації;
- один набір процесів бізнес-правил та вилучення;
- загальна семантика;
- Кероване та централізоване середовище;
- Легке створення та наповнення вітрин даними;
- Один репозиторій х метаданими.

Недоліки такого архітектурного рішення:

- Щоб реалізувати, потрібно багато витрат;
- Потребує багато ресурсів;
- Потребує ресурси в масштабі цілого підприємства.

Віртуальне сховище - це система, яка дає доступ до звичайної системи, в свою чергу емулює роботу зі сховищем даних. Є два способи для організації віртуального сховища. Використати спеціальні засоби доступу для бази даних, або створити ряд уявлень в базі даних (наприклад, продукти класу desktop OLAP). Головними перевагами такої організації сховища, це невисока вартість під час реалізації та простота, одна платформа з джерелом інформації, відсутність мережевих з'єднань між джерелом інформації і сховищем даних. Але є і недостатки, це час обробки запитів, відсутність історії інтеграція даних з іншими джерелами , залежність від доступності та структури основної бази

даних, тому що, дані з джерел оперативно витягуються, перетворюються і інтегруються в момент формування запиту.

Наповнення інформаційних сховищ відбувається в декілька етапів:

- екстракція (витяг) – імпорт даних у сховище з інформаційних підсистем;
- трансформація – консолідування, агрегування даних (тобто обчислюються сумарні або ін. показники), розбиття їх на фракції, коригування та трансформування у відповідні формати, „очищення” від індивідуальних ознак;
- завантаження у сховище у вигляді „історичних пластів”, кожен з яких належить до конкретного періоду часу.

Структура сховища даних для нашої системи представлена на рис. 3.3.1.

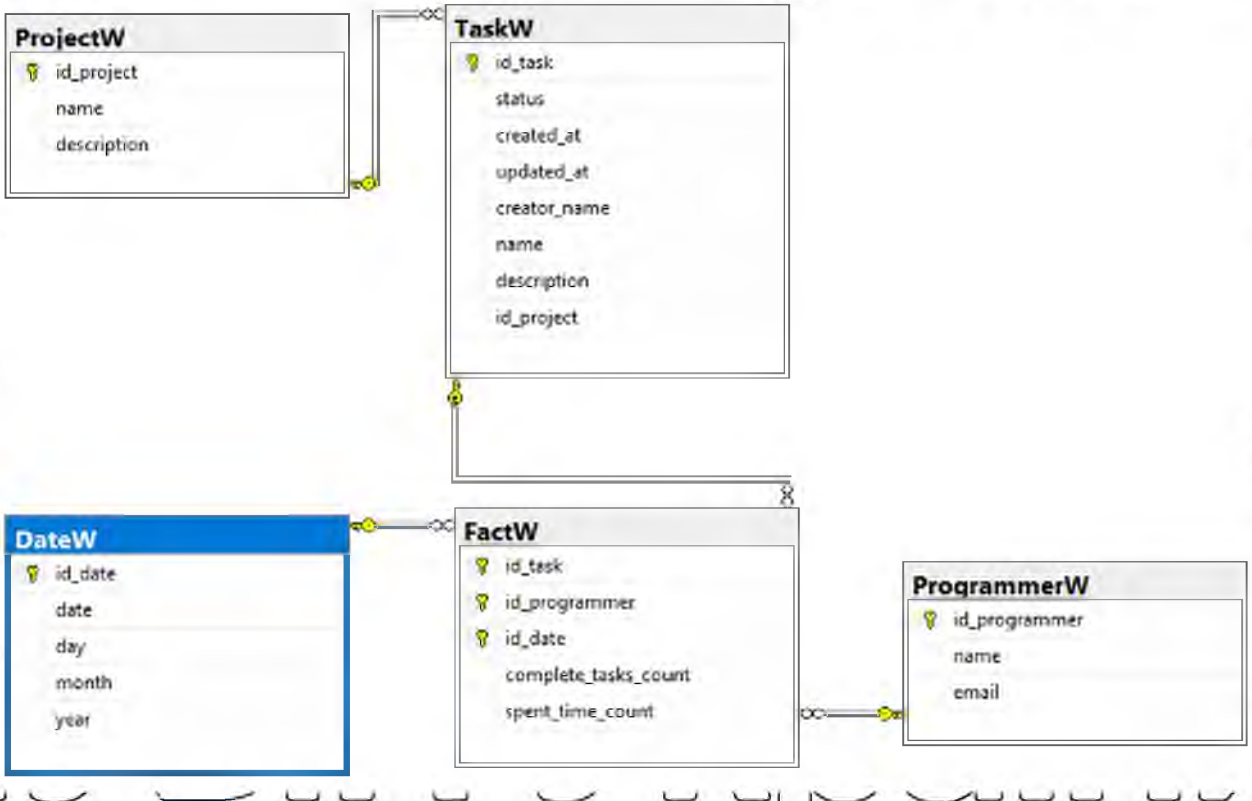



*Рис. 3.3.1 Структура сховища даних.*

Для зберігання даних ми використали структурну схему «сніжинка». Схема «сніжинка» представляється у вигляді централізованих таблиць фактів, які пов'язані з кількома вимірами.

Сніжинка, це метод який нормалізує таблиці вимірювань у схемі зірочка. Коли дана схема нормалізується за всіма таблицями вимірів, в результаті нагадує сніжинку з таблицею фактів всередині, як ми можемо спостерігати на рис. 3.3.1 . Факти ,це частини, які можна виміряти або проаналізовати. Найкращими фактами є адитивні та числові дані, які можуть бути представлені послідовною низкою значень.

Основний принцип «сніжинки», це нормалізація таблиць вимірювань за допомогою видалення атрибутів які мають низьку потужність та формують окремі таблиці. Також на даній схемі окрім таблиці фактів розміщено чотири виміри.

Вимірами виступають, якісні атрибути, які служать умовами вибирання даних під час запитів, або це заголовки рядів у звітах. До таких таблиць відносяться таблиці, які містять умовно-постійну інформацію. Тобто дані, які описують виміри, це зазвичай ,неадитивні та відносно статичні.

Список вимірів створеного сховища даних:

ProjectW – містить основну інформацію про проекти;

TaskW –містить інформацію про задачі, адже над однією задачею може працювати декілька людей;

DateW – містить часові проміжки;

FactW - до таблиці фактів мігрують всі ключі з таблиць вимірів, окрім цього ми зберігаємо інформацію про витрачений час на задачі та разуємо кількість виконаних.

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

НУБіП України

# 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

## 4.1 Механізм вилучення, обробки та передачі даних

Для того щоб провести аналіз даних, попередньо потрібно їх обробити. Обрабка даних, це комплекс технічних та програмно-математичних засобів для вирішення класу проблем автоматизованим способом. Для того щоб провести аналіз над отриманими даними, було обрано BI.

BI – це комп'ютерні методи та інструменти, які використовуються для організацій, що займаються перекладом ділової інформації у вигляді транзакцій в форму, яка придатна для проведення бізнес-аналізу, а також всі засоби які призначені для роботи з таким чином інформацією.

BI – займається в першу чергу очищенням, перетворенням даних у зручний формат для аналізу, інтерпретують велику кількість даних загострюючи увагу лише на ключових факторах, які впливають на ефективність та відстежують результати прийнятя рішень

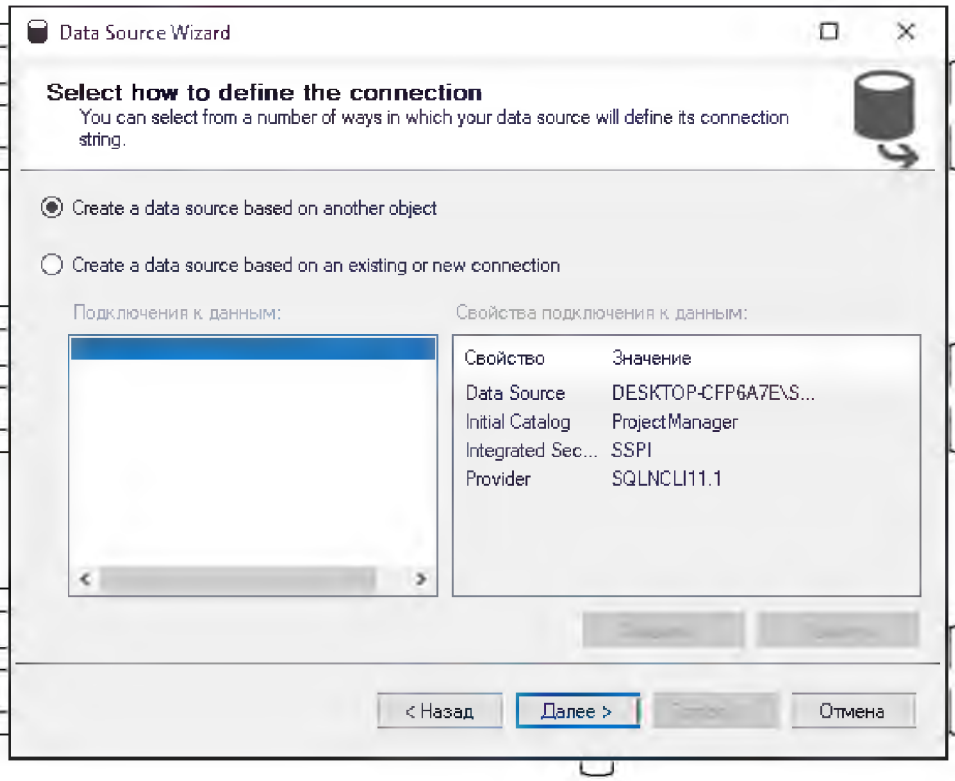



*Рис. 4.1.1 Підключення до джерела даних*

Інтерактивні звіти дозволяють аналізувати дані і виявляти закономірності, щоб приймати більш сперативні і обгрунтовані рішення. А за допомогою вхсдять до складу сервера звітів Power BI сервісів SQL Server Reporting Services можна професійно оформляти звіти відповідно до своїх бізнес-потреб.

Для побудови кубу було встановлено VisualStudio з розширенням та в ньому створено проект (створення рішення) типу Analysis Service (рис.4.1.1).

Далі створюємо Data Source View, який представлено на рис 4.1.2. Під Data Source View розуміється зріз джерела, яке буде використовуватися для заповнення сховища, при цьому в нього можуть входити як таблиці, так і уявлення (view) реляційної бази - джерела даних.

Служби аналізу можуть отримати доступ лише до таблиць або представлень у представленні джерела даних, будь-що за межами представлення джерела даних недоступне. Деякі основні властивості представлення джерела даних:

- Одне подання джерела даних може отримувати дані лише з одного джерела даних.

- Служби аналізу дозволяють нам створювати кілька представлень джерел даних з одного джерела даних.
- Служби аналізу дозволяють нам створювати представлення джерел даних із кількох джерел даних.

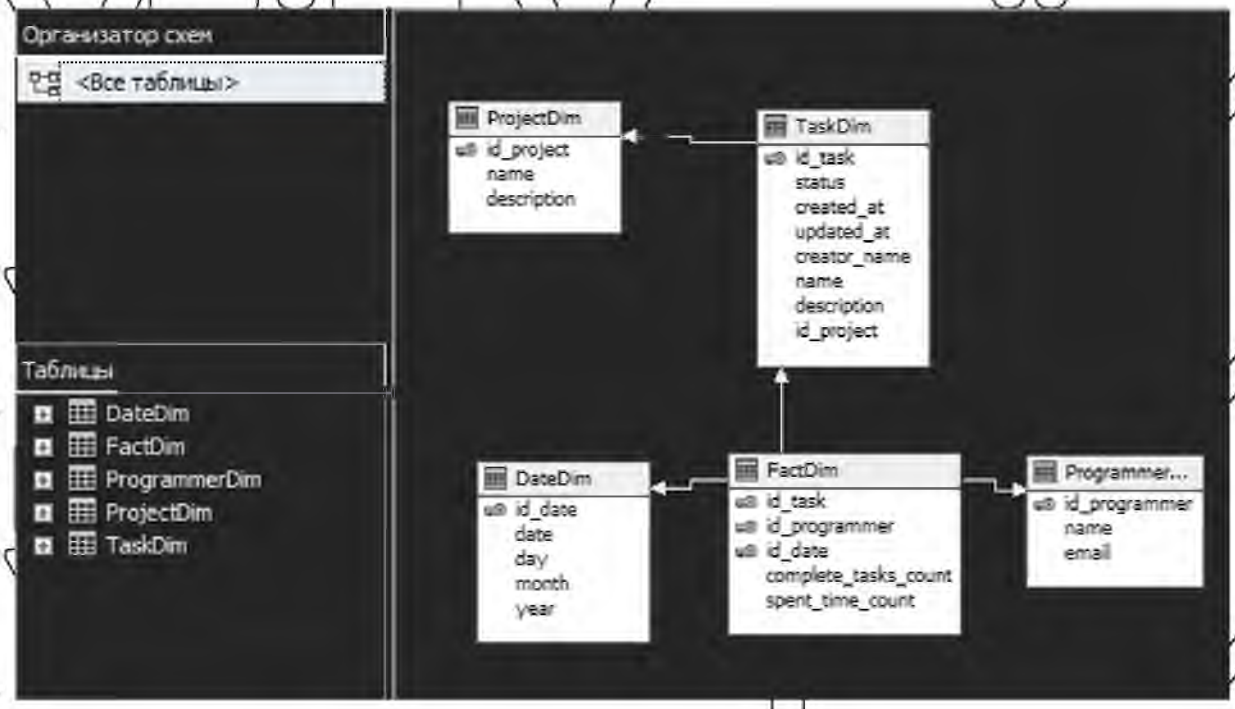



*Рис. 4.1.2 Підключення до джерела даних*

Далі ми переходимо до розгортання кубу. На рис. 4.1.3 можна побачити всі виміри які було створено після розгортання кубу. На рис. 4.1.4 зображена сама схема кубу.

НУБіП України
НУБіП України
НУБіП України

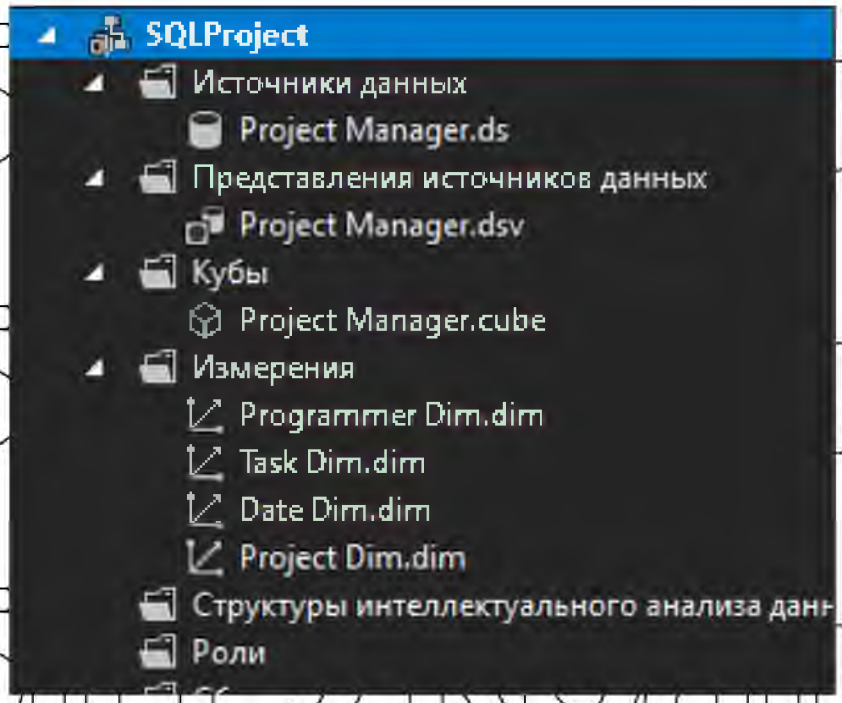



*Рис. 4.1.3 Результат розгортання куба*

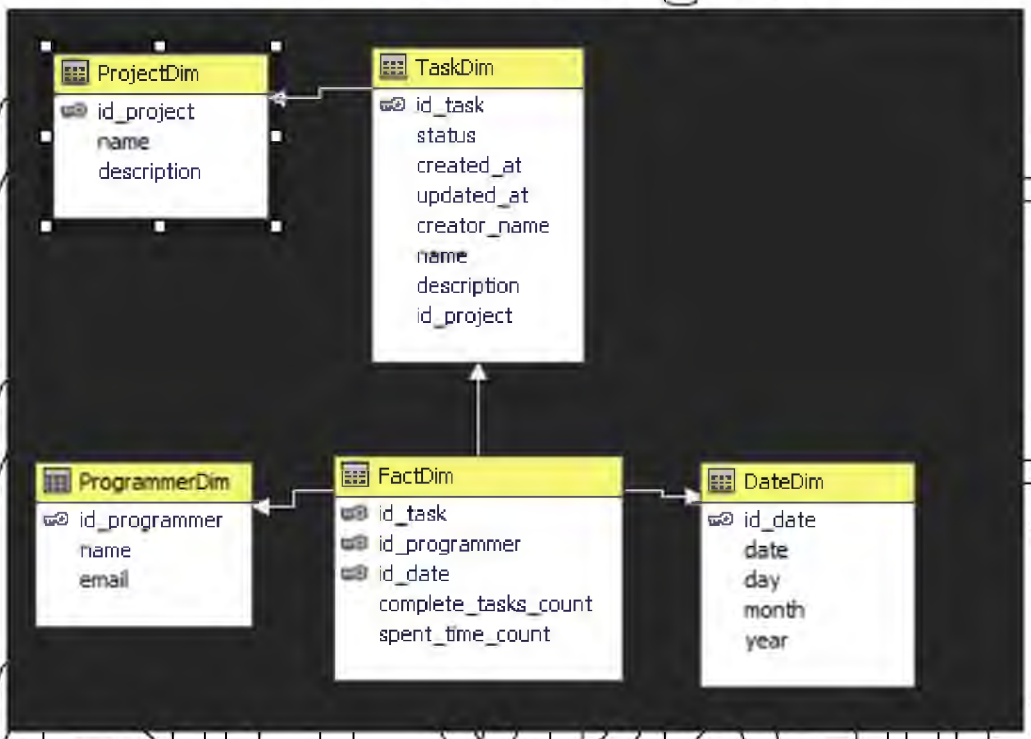



*Рис. 4.1.4 Розгорнутий куб*

Щоб отримати дані з нашої бази даних та заповнити ними куб, було використано DataFlow. Data Flow – модель проектування або представлення потоків даних у графічному вигляді в середині інформаційної системи. Дана діаграма потоків даних, також може використовуватись для візуалізації процесу обробки даних.

Для такого проектування було встановлено роширення SSIS в VisualStudio та створено проект (рис. 4.1.5).

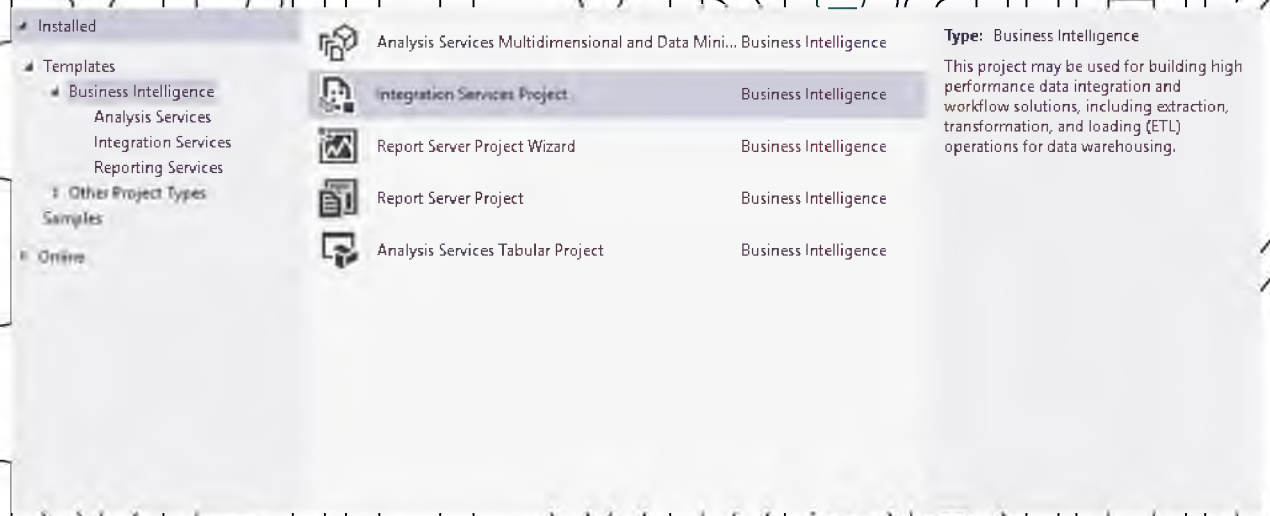



*Рис. 4.1.5 Створення проекту для заповнення кубу даними*

Потім створили задачу потоку даних для вимірів першої черги (рис. 4.1.6), а в ній створили потоки даних для кожного виміру (рис. 4.1.7) та факту (рис. 4.1.8).

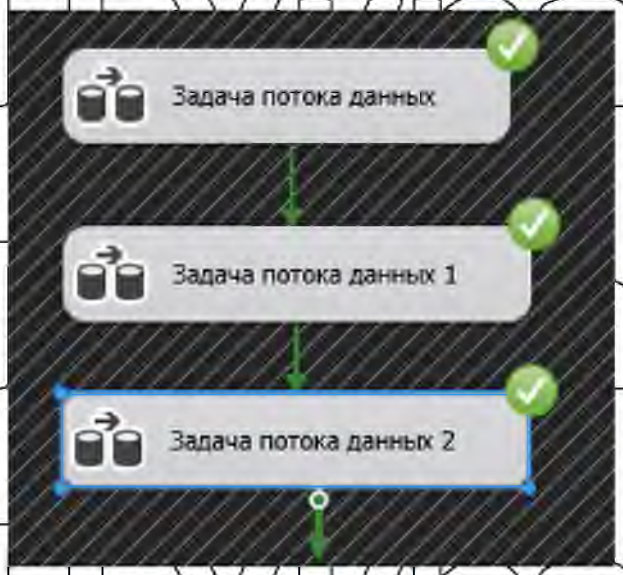



Рис. 4.1.6 Задача потоку управління даними

До складу завдання потоку даних входить підсистема обробки потоку даних, що переміщує дані між джерелом і призначенням і дозволяє користувачеві перетворювати, очищати та змінювати дані в процесі переміщення. Додавання завдання потоку даних до потоку керування пакетом дозволяє пакету вилучати, перетворювати та завантажувати дані.

*Рис. 4.1.7 Потік вимірів*

*Рис. 4.1.8 Потік фактів*

На рис. 4.1.6 та 4.1.8 представлено запущений проект, як можна спостерігати, результат виконання успішний, тобто повна обробка даних пройшла також успішно.

### 4.2 Використані методи Data Mining

Одним з методів аналізу було обрано алгоритм 1-R. Даний алгоритм потрібен для формування елементарних правил для класифікації об'єкта. Він будує правила за значенням однієї незалежної змінної, тому називається 1R(rule)-алгоритм.

Для використання даного алгоритму було визначено два класи:

Висока продуктивність (H) – коли рівень виконання задач є нижчим за вказаний показник.

Низька продуктивність (L) – коли рівень виконання задач є вищим за вказаний показник.

Незалежними змінами виступають, «програміст» та «задача». Залежна зміна «Рівень виконання задач». Результат використання 1-R алгоритму представлені на рис. 4.2.1 - 4.2.3.

Data Mining

Програміст | Вихідний набір даних | Задача

| Програміст | Посада | Рівень задачі | Рівень виконання |
|---|---|---|---|
| Ruslan | Middle | Easy | H |
| Ruslan | Middle | Beginner | H |
| Ruslan | Middle | Normal | H |
| Ruslan | Middle | Hard | H |
| Ruslan | Middle | Experienced | L |
| Maxim | Middle | Beginner | L |
| Maxim | Middle | Normal | H |
| Maxim | Middle | Easy | H |
| Maxim | Middle | Hard | H |
| Maxim | Middle | Experienced | H |
| Zhenya | Junior | Easy | H |
| Zhenya | Junior | Beginner | H |
| Zhenya | Junior | Normal | H |

Рис. 4.2.1 Вихідний набір даних

НУБіП України
НУБіП України
НУБіП України

Data Mining

Програміст | Вихідний набір даних | Задача

| Програміст | Посада | Рівень задачі | Всього значень | Задовільняють | Ймовірність |
|---|---|---|---|---|---|
| Ruslan | Middle | Easy | 10 | 9 | 0.9 |
| Ruslan | Middle | Beginner | 10 | 6 | 0.6 |
| Ruslan | Middle | Normal | 10 | 9 | 0.9 |
| Ruslan | Middle | Hard | 6 | 5 | 0.8 |
| Ruslan | Middle | Experienced | 2 | 1 | 0.5 |
| Maxim | Middle | Beginner | 12 | 6 | 0.5 |
| Maxim | Middle | Normal | 11 | 10 | 0.9 |
| Maxim | Middle | Easy | 11 | 8 | 0.7 |
| Maxim | Middle | Hard | 5 | 3 | 0.6 |
| Maxim | Middle | Expenenced | 2 | 2 | 1.0 |
| Zhenya | Junior | Easy | 13 | 11 | 0.8 |
| Zhenya | Junior | Beginner | 12 | 11 | 0.9 |
| Zhenya | Junior | Normal | 11 | 10 | 0.9 |

*Рис. 4.2.2 Програміст*

Програміст | Вихідний набір даних | Задача

| Рівень задачі | Всього значень | Задоволеність | Ймовірність | Рівень виконання |
|---|---|---|---|---|
| Easy | 48 | 24 | 0.5 | H |
| Beginner | 45 | 20 | 0.4 | L |
| Normal | 43 | 19 | 0.4 | L |
| Hard | 13 | 8 | 0.6 | H |
| Experienced | 4 | 3 | 0.8 | H |

*Рис. 4.2.3 Задача*

Головна особливість алгоритму 1R, який був використаний що для кожного значення кожної незалежної змінної будуються правила, для цього правила обчислюється помилка — кількість об'єктів з тим самим значенням незалежної змінної, але не відповідних тому значенням залежної змінної, яка

НУБіП України

найчастіше зустрічається для даного значення незалежної змінної. У підсумку обирається змінна за якою можна з найбільшою точністю класифікувати об'єкти.

Так і в нашому випадку під час визначення ймовірності, ми змогли класифікувати об'єкти за класом H та L. Ймовірність розраховували на основі даних кількості виконаних задач всього, поділених на кількість задач виконаних з кількістю витраченого часу не більше чим було визначено.

В результаті ми отримали два основних аналізи по тому, з якою ймовірностю програміст виконає задачу не витративши часу більше ніж відведено. Також статистику по задачам з отриманих даних усієї команди розробників, з якою ймовірністю їх виконають не витративши часу більше ніж відведено на неї.

Для рішення задачі класифікації з врахуванням значень декількох незалежних змінних було використано метод Наївного Байєса.

Для використання даного алгоритму також було визначено два класи:

- Висока продуктивність (H) – коли рівень виконання задач є нижчим за вказаний показник.
- Низька продуктивність (L) – коли рівень виконання задач є вищим за вказаний показник.

На рис. 4.2.4 зображено звіти у вигляді таблиці з програмістами, рівнями задачі та класифікацію цих даних за двома класами H та L. Розрахунки по кількості та ймовірності класифікації за двома класами за незалежними змінами «програміст» та «рівень задачі» представлені на рис. 4.2.5 - 4.2.6. Загальні результати проведення розрахунків даних по класам H та L представлено на рис. 4.2.7.

Використовуючі метод Наївного Байєса ми класифікували об'єкти за декількома незалежними змінними. В нашому випадку незалежними змінними виступають програміст та рівень задачі. На основі значень цих мінних ми однозначно визначили, до якого класу належать спостереження, байєсів класифікатор повідомить ймовірність приналежності до цього класу. В результаті обробки даних було проаналізовано ефективність роботи двох

програмістів рівня middle. Результат вийшов однаковий на рівні виконання задач з рівнем "Normal", а саме ефективність виконання задач такого рівня дорівнює 78%.

Data Mining

Програміст | Вихідний набір даних | Задача | Таблиця вимірів | Результати

| Програміст | Рівень задачі | Рівень виконання |
|---|---|---|
| Ruslan | Easy | H |
| Ruslan | Beginner | H |
| Ruslan | Normal | H |
| Ruslan | Hard | H |
| Ruslan | Experienced | L |
| Maxim | Beginner | L |
| Maxim | Normal | H |
| Maxim | Easy | H |
| Maxim | Hard | H |
| Maxim | Experienced | H |
| Zhenya | Easy | H |
| Zhenya | Beginner | H |
| Zhenya | Normal | H |
| Zhenya | Hard | H |
| Ilya | Beginner | H |
| Ilya | Normal | L |
| Ilya | Easy | L |
| Ilya | Hard | H |

*Рис. 4.2.4 Таблиця вимірів*

# Data Mining

Програміст | Вихідний набір даних | Задача | Таблиця вимірів | Розрахунки

**Кількість**

| Програміст | H | L |
|---|---|---|
| Ruslan | 4 | 1 |
| Maxim | 4 | 1 |
| Zhenya | 4 | |
| Illya | 2 | 2 |

**Ймовірність**

| Програміст | H | L |
|---|---|---|
| Ruslan | 0,286 | 0,250 |
| Maxim | 0,286 | 0,250 |
| Zhenya | 0,286 | 0,000 |
| Illya | 0,143 | 0,500 |

*Рис. 4.2.5 Розрахунки програміста*

**Кількість**

| Рівень | H | L |
|---|---|---|
| Beginner | 3 | 1 |
| Easy | 3 | 1 |
| Normal | 3 | 1 |
| Hard | 4 | 0 |
| Experienced | 1 | 1 |

**Ймовірність**

| Рівень | H | L |
|---|---|---|
| Beginner | 0,214 | 0,250 |
| Easy | 0,214 | 0,250 |
| Normal | 0,214 | 0,250 |
| Hard | 0,286 | 0,000 |
| Experienced | 0,071 | 0,250 |

*Рис. 4.2.6 Розрахунки рівня задач*

| Рівень | Кільк. загал. | Ймовірність |
|---|---|---|
| H | 14 | 0,778 |
| L | 4 | 0,22 |

**Результаты**

| Рівень | Програміст | Рівень задачі | - | - | Сумма |
|---|---|---|---|---|---|
| H | Ruslan | Normal | 0.047616712 | 0.7759371562876 | |
| L | Ruslan | Normal | 0.01375 | 0.2240628437124 | 1 |
| H | Maxim | Normal | 0.047616712 | 0.7759371562876 | |
| L | Maxim | Normal | 0.01375 | 0.2240628437124 | 1 |

*Рис. 4.2.7 Результати*

Також одним із способів класифікації при обробці даних було використано пошук асоціативних масивів. Даний спосіб є дуже поширеним застосуванням Data Mining. Суть цієї задачі, це визначити набір об'єктів, які найчастіше зустрічаються, в інформаційних масивах. Ця задача є окремим випадком задачі класифікації.

Під час аналізу потрібна інформація, це інформація про послідовність подій, що відбуваються. При виявленні закономірностей у таких послідовностях можна з певною часткою ймовірності передбачати появу подій у майбутньому, що дає змогу приймати правильніші рішення. Результати пошуку асоціативних масивів, визначення правил та їх важливість представлена на рис. 4.2.8 - 4.2.10.

*Рис. 4.2.8 Зв'язки між наборами елементів*

НУБіП України

| Подде... | Размер | Набор элементов |
|---|---|---|
| 58 | 1 | Spent Time Count < 4,11771842 |
| 39 | 1 | Id Level < 2 |
| 39 | 2 | Id Level < 2, Spent Time Count < 4,11771842 |
| 31 | 1 | Id Level = 2 - 3 |
| 29 | 1 | Spent Time Count = 4,11771842 - 9,3373803888 |
| 25 | 1 | Id Level = 3 - 4 |
| 19 | 2 | Id Level = 2 - 3, Spent Time Count < 4,11771842 |
| 16 | 2 | Id Level = 3 - 4, Spent Time Count = 4,11771842 ... |
| 13 | 1 | Id Level >= 4 |
| 12 | 1 | Spent Time Count = 9,3373803888 - 14,286642752 |
| 12 | 2 | Spent Time Count = 4,11771842 - 9,3373803888,... |
| 7 | 1 | Spent Time Count = 14,286642752 - 24,8141826... |
| 7 | 2 | Spent Time Count = 9,3373803888 - 14,2866427... |
| 5 | 2 | Spent Time Count = 9,3373803888 - 14,2866427... |
| 5 | 2 | Spent Time Count = 14,286642752 - 24,8141826... |
| 2 | 1 | Spent Time Count >= 24,8141826144 |
| 2 | 2 | Spent Time Count >= 24,8141826144, Id Level >... |
| 2 | 2 | Spent Time Count = 14,286642752 - 24,8141826... |
| 1 | 2 | Id Level >= 4, Spent Time Count = 4,11771842 - ... |

*Рис. 4.2.9 Згруповані елементи*

| Ве... | Важность | Правило |
|---|---|---|
| 1,000 | 0,540 | Id Level < 2 -> Spent Time Count < 4,11771842 |
| 0,640 | 0,582 | Id Level = 3 - 4 -> Spent Time Count = 4,11771842 - 9,3373803888 |
| 0,613 | 0,078 | Id Level = 2 - 3 -> Spent Time Count < 4,11771842 |

*Рис. 4.2.10 Отримані правила*

При використанні даного методу, ми побудували модель за допомогою Visual Studio, , IntegrationServices та AnalysisServices , у вигляді інтелектуального аналізу даних. Для побудови моделі, було використано попередньо оброблене сховище даних, з перевантаженням даних інформаційного сховища з транзакційної бази даних. Після створення моделі, було згруповано набори елементів тобто задач (рис. 4.18), які згруповані по кількості витрачего часу розробником на певний рівень задачі. Зв'язки між наборами представлені на рис. 4.17. Наступним етапом є отримання певних правил, які допожуть для визначення певних ріжень, тобто в нашому випадку, це визначення оцінки для задачі, та правильного розподілення для кожного розробника відповідного рівня.

### 4.3 Розрахунок KPI

Наступна технологія, яку було використано, це KPI. Для цієї системи було створено два показники KPI:

• Середня кількість витраченої години на завдання, яке має наближуватися до 21 години

• Кількість виконаних завдань, яка повинна наближатися до 11

На рис. 4.3.1 відображено опис створення міри та KPI для обрахунку середнього витраченого часу та кількість виконаних задач. На рис. 4.3.2 - 4.3.3 відображені обраховані KPI по витраченому часу на задачу та кількості виконаних задач.

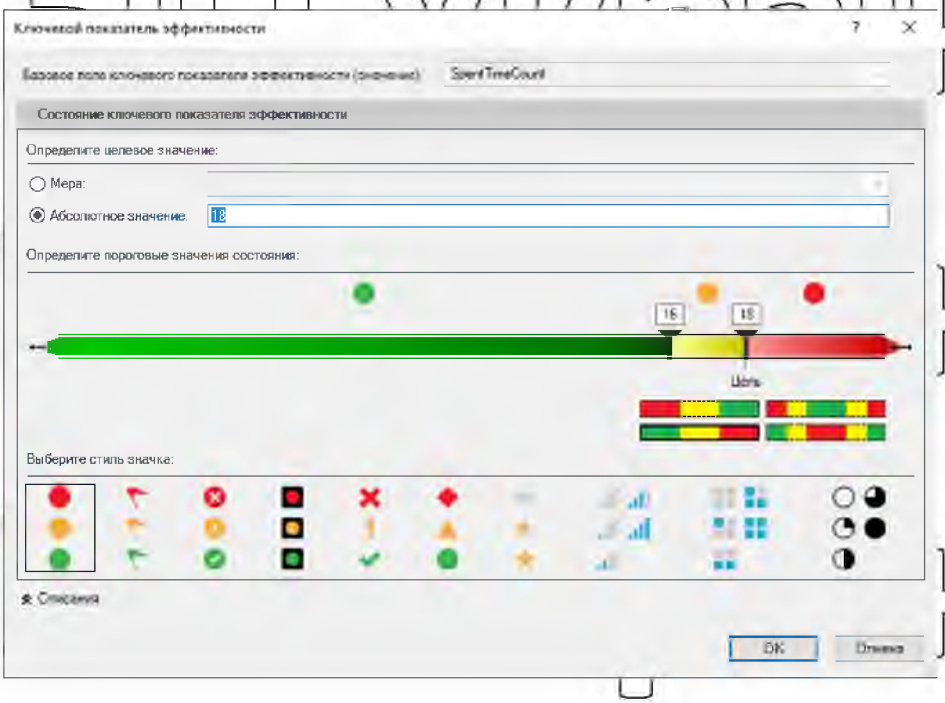



*Рис. 4.3.1 Створення KPI для обрахунку середнього часу витраченого на задачу.*

| Названия строк | Сумма використаного часу на завдання | Середній витрачений час на завдання | Ціль | Статус |
|---|---|---|---|---|
| Дмитро Павлюк | 277 | 21,31 | 18 | |
| Євгеній Мороз | 241 | 16,07 | 18 | |
| Максим Маташин | 523 | 20,92 | 18 | |
| Руслан Дейсан | 190 | 15,83 | 18 | |
| **Общий итог** | **1231** | **18,94** | **18** | |

*Рис. 4.3.2 KPI витраченого часу на задачу*

| Сумма по кількості виконаних задач | Ціль по кількості виконаних завдань | Стан |
|---|---|---|
| 12 | 11 | |
| 10 | 11 | |
| 13 | 11 | |
| 8 | 11 | |
| 43 | 11 | |

*Рис. 4.3.3 KPI кількості виконаних задач*

Також за допомогою програми Excel для роботи було проаналізовано дані, щоб в подальшому можна було прийняти виважені рішення.

За допомогою функцій цієї програми, можна виробляти з цифрами різних маніпуляцій: складування, вичитування, умноження, видалення, здійснення консолідації даних, автоматизація звітів, здійснення складних масивних даних, які будуть автоматично оновлені при грамотному складі.

На рис. 4.3.4 відображено побудову звіту який відображує кількість витраченого часу та виконаних задач за часовим діапазонам.

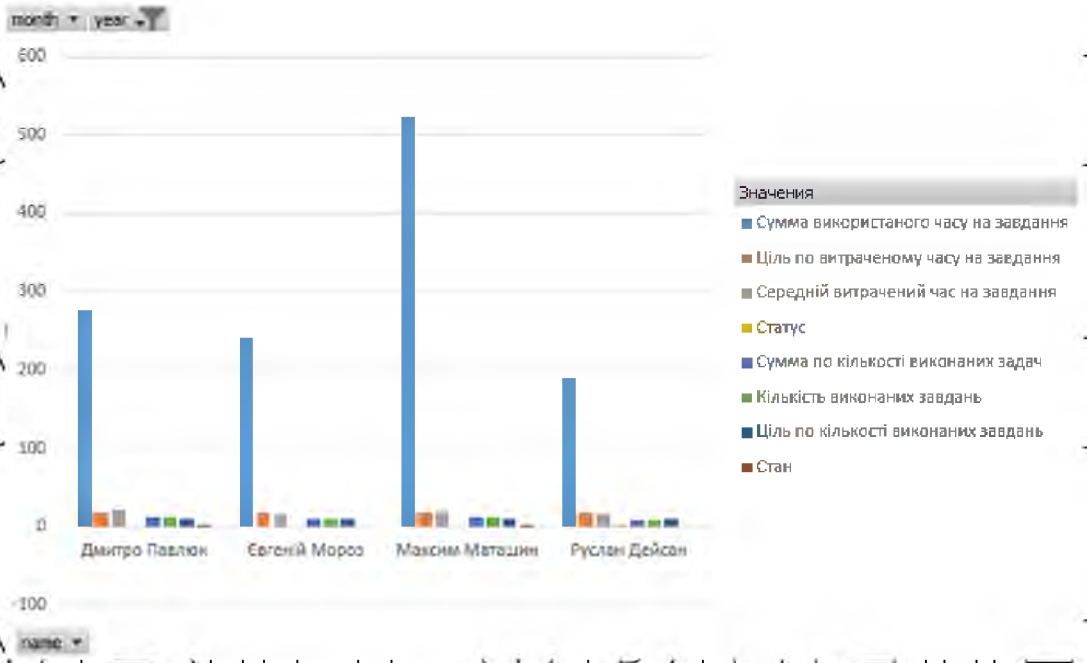



*Рис. 4.3.4 Показники кількості витраченого часу програмістом на задачу*

Задопомогою розрахунку KPI (рис. 4.3.2) було визначено, що середній витрачений час в двох працініків не зовсім критичний але можна й підтянути швидкість краще, в Євгенія гарний результат на дві години менше ніж поставлена ціль та Руслан виконує завдання на трохи більше від всіх інших, але це також залежить від кількості виконаних задач, результат яких наведено на рис.

4.3.3. Там ми бачимо, що Руслан витратив найменше часу, тому що він виконав малу кількість завдань та напевно легких, а хто найбільше витратив часу той перевиконав план на поставлену ціль виконаних задач.

За допомогою даного звіту, ми можемо визначити найефективніших розробників, та в майбутньому правильно розподілити кількість задач, яких потрібно виконати, індивідуально під кожного розробника. Також за допомогою побудованого графіку в Excel у вигляді часового діапазону, можна визначити, хто з розробників не впорується з виконанням задач за відведений час та співвідношення кількості виконаних задач з цілю виконаних задач на кожного розробника.

НУБіП України

## ВИСНОВОК

Під час виконання нацкової роботи, було розроблено експертну систему. Було детально проаналізовано предметну область, для якої розроблюється система, її актуальність та проаналізовано вже існуючі рішення . На основі вимог, створено технічне завдання, в якому детально описано, які функціональні можливості, повинна містити система.

Наступним етапом є створення архітектури системи, яка представляє взаємодіють між собою окремих компонентів та учасників системи. Для роботи системи було створено базу даних, попередньо було розроблено логічну модель даних та реалізовано за допомогою SQL Server 2014, на основі якої створено сховище даних.

Після створення сховища та визначення його структури, всі дані були перенесені в сховище даних. Наступним етапом було розгортання сховища, визначення вимірів, мір, розгортання та заповнення кубу даними, за допомогою Visual Studio SQL Server Data Tools.

В ході створення експертної системи було розглянуто дві основних технології для підготовки та обробки даних, OLAP та Data Mining. Використанні

два методи Data Mining 1Rule та Наївного Байеса. За домпомогою 1Rule, було класифіковано об'єкти за класом H та L. В результаті було визначено, з якою ймовірностю програміст виконає задачу не витративши часу більше ніж відведено на неї, та статистику по рівню виконання цілої команди. Метод наївного Байеса надав можливість проаналізуваим ефективність роботи програмістів різних рівнів та ефективність виконання задач різного рівня.

Також одним з методів, який було використано, це KPI. Даний метод, дав можливість отримати відвовіді на такі важливі питання:

- Скільки в середньому розробник витрачає часу на виконання задачі ?
- Яку кількість задач виконує розробник за певний проміжок часу ?

Всі отриманні дані в ході аналізу за допомогою різних методів, дають можливість визначити найефективніших розробників та дані по кожному розробнику, які допомагають в прийнятті подальших рішень.

Розроблювана експертна система покращить організацію роботи в команді розробників, тому що вона надає можливість аналізувати роботу кожного з учасників команди в будь-який момент часу, для виявлення сильних та слабких сторін команди.

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України

НУБіП України

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Психологія тимбілдингу: навчальний посібник / Романовський О.Г., Шаполова В.В., Квасник О.В., Гура Т.В. ; за заг. ред. Романовського О.Г., Калашникової С.В. – Харків : «Друкарня Мадрид», 2017. – 92 с. ISBN 978-617-7470-63-1.
2. Методи управління людськими ресурсами при формуванні команд мультипроектів та програм: монографія/Н. В. Доценко,Л.Ю. Сабадош, І.В. Чумаченко; за заг ред. І.В.Чумаченко; Харків. Нац. ун-т міськ. госп-ва ім. О.М. Бекетова, 2015.-201с.
3. Революційний метод управління проектами / Джефф Сазерленд. – М.: Манн, Іванов і Фербер, 2015. -288 с.
4. Швабер К. Авторитетний посібник зі Скраму: Правила Гри [Електронний ресурс] / К.Швабер, Дж. Сазерленд. – Режим доступу: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-UA.pdf.

5. Интуитивное и научное в управлении проектами / В.А. Трач // Управління проектами та розвиток виробництва: Зб.наук.пр. – Луганськ: вид-во СНУ ім. В.Даля, 2015. – №2(54). – С. 33-46.
6. Формула та напрями наукових досліджень зі спеціальності «Управління проектами та програмами» / С.Д. Бушуєв., В.Д. Гогунський., І.В. Кононенко., // Управління проектами: стан та перспективи : VIII Міжнар. наук.-практ. конф. – Миколаїв : НУК, 2015. – С. 28 – 31.
7. М. Петрик, О. Петрик «Моделювання програмного забезпечення», 2015 рік, 200стр.
8. Управление проектами : учебное пособие / [И. И. Мазур, В. Д. Шапиро, Н. Г. Ольдерогге и др.]. – М. : Издательство «Омега-Л», 2010. — 960 с.
9. Пат. WO2020191317 (A1) США, PROJECT MANAGEMENT METHOD AND SYSTEM FOR COMPUTER CODE MAPPING AND VISUALIZATION; Заявлено 20.03.2020; Опубл. 24.09.2020.
10. Пат. US2019286439 (A1) США, GAMIFIED PROJECT MANAGEMENT SYSTEM AND METHOD; Заявлено 15.07.2019; Опубл. 19.09.2019.
11. Пат. JPH08305663 (A) США, TEAMWORK SUPPORT SYSTEM; Заявлено 19.10.196; Опубл. 22.11.1996.
12. Пат. KR20200074656 (A) Корея, API Open source API based software project task management system; Заявлено 23.05.2020; Опубл. 25.06.2020
13. Методы и модели анализа данных OLAP и Data Mining. /Барсегян Ф., Куприянов М., Степаненко В., Холод И. – СПб БХВ-Петербург, 2008. – 384 с
14. Бахрушин В.Е. Методы анализа данных: учебное пособие для студентов / В.Е. Бахрушин. – Запорожье: КПУ, 2011. – 268 с.

НУБіП України

15. Бергер, А. Б. Microsoft SQL Server 2005 Analysis Services. OLAP та багатовимірний аналіз даних / А.Б. Бергер. – К.: БХВ-Петербург, 2018. – 147 с.

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України