НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.9:005:631.1

«ПОГОДЖЕНО»

Декан факультету інформаційних технологій

Главунова О.Г., д.пед.н., професор

_____

«___» _____ 20__р

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

Голуб Б.Л., к.т.ц., доцент

_____

«___» _____ 20___ р

МАГІСТЕРСЬКА РОБОТА

на тему: Програмне забезпечення керування діяльності керівника агрофірми

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма Програмне забезпечення інформаційних систем

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми _____ / Голуб Б.Л., доц., к.т.н. /

(підпис) (ПІБ) (вчене звання і ступінь)

Керівник магістерської роботи _____ / Бородкін Г.О., с.в. /

(підпис) (ПІБ) (вчене звання і ступінь)

Виконав _____ / Войт В.В. /

(підпис) (ПІБ студента)

КИЇВ – 2021

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ

НУБіП України

ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних наук

доц., к.т.н. _____ Б. Л. Голуб
(вчене звання і ступінь) (підпис) (ініціали і прізвище)

«___» _____ 20__ р.

ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Войту Владиславу Віталійовичу
(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія Програмного Забезпечення»
Освітня програма «Програмне забезпечення інформаційних систем»
Орієнтація освітньої програми освітньо-професійна

1. Тема магістерської роботи: Програмне забезпечення керування діяльності керівника агрофірми затверджена наказом ректора НУБіП від «29» жовтня 2020 р. №1636 «С»
2. Термін подання завершеної роботи на кафедру 2021 _____ _____
рік місяць число
3. Вихідні дані до магістерської роботи: списки справ та доручень
4. Перелік питань, що підлягають дослідженню:
   1) Дослідження предметної області, постановка задач.
   2) Огляд існуючих рішень та літературних джерел.
   3) Вибір методів дослідження.

Дата видачі завдання «29» жовтня 2020 р.

Керівник магістерської роботи _____ Бородкін Г. О.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Войт В. В.
(підпис) (прізвище та ініціали студента)

НУБіП України

АНОТАЦІЯ

Об’єкт дослідження – агрофірма, яка використовує електронний список для планування діяльності. Предмет дослідження – програмне забезпечення системи планування діяльності керівника агрофірми.

Мета проекту полягає у створенні системи планування трудового дня в агропромисловому комплексі для більш ефективного використання часу.

Результати проекту – створена система планування діяльності готова до експлуатації програми в агрофірмах. Під час тестування додатку не виникало жодних помилок, а можливості, котрі надає система цілком задовольняють поставленні задачі.

НУБіП України

ABSTRACT

The object of research is an agricultural firm that uses an electronic list to plan activities. The subject of research - the software of the planning system of the head of the agricultural firm.

The aim of the project is to create a system of planning a working day in the agro-industrial complex for more efficient use of time.

Project results - the activity planning system is ready for operation of the program in agricultural firms. There were no errors during the testing of the application, and the capabilities provided by the system fully satisfy the task.

НУБіП України
НУБіП України
НУБіП України

НУБіП України

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД - база даних.

СУБД – система управління базами даних.

UML - уніфікована мова моделювання, (англ. Unified Modeling Language).

SQL - декларативна мова програмування, (англ. Structured query language).

ІТ – інформаційні технології.

ПЗ - програмне забезпечення.

HTML - мова розмітки гіпертексту, (англ. HyperText Markup Language).

CSS – каскадні таблиці стилів, (англ. Cascading Style Sheets).

PHP - скриптова мова програмування, (англ. Personal Home Page).

JS - динамічна, об'єктно-орієнтована прототипна мова програмування.

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

НУБіП України

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 5

ВСТУП 8

1 10

1.1 11

1.2 13

1.3 16

1.3.1 16

1.3.2 17

1.4 17

2 18

2.1 21

2.1.1 21

2.1.2 23

2.1.3 25

2.2 28

2.2.1 28

2.2.2 29

2.2.3 31

2.2.4 32

3 33

3.1 34

НУБіП України

3.2 36

3.3 38

3.4 39

3.4.1 39

3.4.2 40

3.4.3 41

3.4.4 44

3.4.5 45

3.4.6 45

3.4.7 46

3.4.8 46

3.4.9 47

4 47

4.1 48

4.2 61

4.3 61

ВИСНОВОК **60**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **62**

НУБіП України

ВСТУП

Майбутній результат – це можливості сучасного рішення. Майбутнє залежить від високого рівня невизначеності та випадковості, але наперед планування діяльності може підвищити шанси вашого успіху. Це пов'язано з багатьма сторонами нашого життя – особистого та професійного характеру. Планування стає все більш важливим для людей, які керують своїм підприємством, оскільки стає все важче розкрити особисті та професійні аспекти. Для агробізнесу планування може стати ще більш важливим через невизначеність, притаманну сільськогосподарському виробництву. Деякі ключові джерела невизначеності включають виробничий ризик, ціновий ризик, фінансовий (або процентний) ризик та зміни в державних програмах.

Ця публікація пояснює важливість планування сільськогосподарських компаній, від постачальників сировини до виробників і переробників, а також пояснює кроки, необхідні для розробки детального бізнес-плану. Загальний процес планування однаковий для кожного типу компанії. Однак у кожного можуть бути різні індивідуальні аспекти, які впливають на зміст його плану. Проте при розробці письмового розпорядку дня ми пропонуємо рекомендовані формати, які повинні бути корисними для всіх видів агробізнесу. Використовуючи різні приклади агробізнесу, він надає широкий контекст загальної теми вашої діяльності.

Таким чином, найкращий варіант планування діяльності є цифровий список, доданий до публічних доступу. Цей спосіб не дозволяє ігнорувати вміст переліку справ та розвантажує пам'ять користувача, роблячи його роботу ефективнішою та більш сконцентрованою.

Щоб забезпечити мультиплатформенність системи, у дипломній роботі було ухвалено рішення створити систему у вигляді веб-сайту. Цей метод не вимагає завантаження та встановлення сторонніх додатків. Все, що вам потрібно для того, щоб запустити систему, - це, будь-який, браузер і відповідний доступ до мережі Інтернет.

НУБіП України

Об'єктом дослідження є агрофірма, яка використовує цифровий перелік справ для планування діяльності.

Предмет дослідження - розробка програмного забезпечення системи планування роботи керівника аграрного підприємства.

Мета дипломної роботи полягає у створенні системи планування трудового дня в агропромисловому комплексі для більш ефективного використання часу.

Завдання проекту:

- Розробка системи, яка може забезпечити всі функції, необхідні для планування.
- Створення зручного і головне - доступного інтерфейсу для користувачів, будь якої, вікової групи.
- Здатність ПЗ працювати на більшій частині апаратних та операційних платформ.

Структура роботи. Пояснювальна записка складається з переліку наступних розділів:

- Анотація
- Вступ
- Чотирьох розділів контенту
- Висновку
- Списку використаних джерел

Обсяг дипломної роботи – 64 сторінки, вміст яких наличує 4 таблиці та 22 зображення.

НУБіП України

1 АНАЛІЗ ВИМОГ ДО ПЗ

1.1 Опис предметної області

Враховуючи динамічний розвиток навколишнього середовища, підприємствам слід приділяти особливу увагу стратегічному управлінню, оскільки вони можуть врахувати всю систему зовнішніх і внутрішніх факторів середовища та визначити найбільш ефективний маршрут. для досягнення цих довгострокових цілей. Автори розглядають визначення понять «стратегія», «розвиток бізнесу» та «стратегічне планування розвитку бізнесу». Основні принципи управління, що використовуються для підвищення ефективності інноваційної діяльності. В роботі розглянуто можливості інноваційної стратегії та переваги стратегічного планування розвитку бізнесу.

Метою даного дослідження є узагальнення теоретичних підходів до визначень «стратегія», «розвиток бізнесу» та «стратегічний план розвитку бізнесу» та аналіз їх ролі в реалізації стратегічного управління компанією. Аналіз основних принципів управління, які необхідно використовувати для підвищення ефективності інноваційної діяльності визначають характеристики інноваційної стратегії та переваги стратегічного планування для розвитку підприємства.

Наукова інновація спрямована на покращення визначення стратегічних планів розвитку бізнесу. При цьому повністю враховується необхідність моніторингу компонентів розвитку бізнесу для забезпечення бажаних змін та компонентів інноваційного процесу, всупереч відомому. Також, на відміну від звичного, можна виділити окремі етапи стратегічного плану розвитку бізнесу, враховуючи етапи життєвого циклу.

Практичне значення цих результатів полягає в тому, що завдяки визначенню стратегічного планування розвитку бізнесу можна спланувати кожен етап та детально проаналізувати інновації кожного етапу відповідно до життєвого циклу підприємства. Також можна оцінити ризики компанії та їх розвиток.

НУБіП України

Щоб завершити свою роботу за графіком та досягти раніше поставлених цілей, потрібно зменшити кількість незначних дій або зовсім ігнорувати їх. Таким чином користувач зможе сконцентруватися на виконані більш важливих справ, а також, дасть змогу цінувати свій запас часу та дотримуватись дисциплін, слідувавши правилам планування, наприклад:

- Формування переліку справ. Процес поєднання майбутніх дій до списку, задавши їм відповідних параметрів.

- Послідовність. Кожна дія з переліку має власний номер, яка відповідає порядку виконання, після завершення якої (рекомендовано), можна перейти до наступного пункту.

- Загальний обсяг роботи підприємства слід розподілити рівномірно між усіма учасниками, таким чином, щоб кожен міг успішно справитись зі своїми дорученнями. Це правило називають реалістичним плануванням.

- Термін виконання. Процес визначення реальних часових меж для виконання певної задачі.

- Термін. Після визначення «терміну виконання» до налаштувань задач встановлюються початкові та кінцева терміни виконання.

- Гнучкість. Система завжди має надавати можливість редагувати список.

Вище перелічені правила дозволяють співробітникам думати про свої цілі та знаходити більш ефективні шляхи їх досягнення, виконувати завдання вчасно та формулювати «План Б», необхідний для вирішення непередбачених проблем, таких як:

- Недостатня ясність планування та визначення пріоритетів
- Неналежне планування або неналежна організація роббчого часу

Формування розпорядку дня являється одним із головних завдань керівництва. При плануванні всі мають сфокусуватися на найважливішому, пам’ятаючи про терміни виконання задач. Правильно складений план покращує ефективність та результативність компанії.

НУБіП України

Сенс планування полягає у послідовному вирішенні поставлених задач, наприклад:

1. Першими в черзі завжди стоять справи з фіксованими термінами, до виконання котрих потрібно приступити в негайному порядку.
2. Вслід виконуються щоденні, звичні справи.
3. В кінці черги розміщена вторинна робота, якою зовсім можна знехтувати, надавши пріоритет раніше переліченим.

Етапи планування розпорядку дня:

1. Постановка задач. На даному етапі встановлюються цілі та ресурси для їх вирішення.
2. Складання плану. Наповнення списку діями, результатами яких є досягнення цілей.
3. Редагування плану. Неможливо спланувати все заздалегідь. Інколи приходиться відмовлятися від менш важливих справ, задля завчасного виконання встановлених цілей.

Грамотно організовані заходи дають можливість менеджерам та працівникам ретельно обдумати свої власні цілі та знайти найбільш ефективний шлях, який не тільки виконуватиметься в найкоротший термін, а й надасть необхідні резерви для вирішення непередбачених перешкод.

Отже, можна зробити висновок, що на сьогоднішній день своєчасне та якісне виконання робіт стає більш важливим для підприємств питанням для обговорення.

1.2 Огляд інформаційних джерел та існуючих рішень

На сьогоднішній день існує безліч різних програм та сервісів, що роблять процес планування доступним для кожного. Їх відмінність полягає в тому, що деякі системи мають швидкий інтерфейс, але оснащені скутим функціоналом. Інші програми, навпаки, мають багатий функціонал, але вони повільно взаємодіють з користувачем і їх використання може дорого коштувати.

Як приклад, розглянемо сайт, який називається Wrike(Рис. 1.2.1) Цей ресурс можна вважати фаворитом своєї сфери, оскільки він включає зручну інтерфейсну систему та численні функції, що належать до основних:

- Ведення проектів – структурування на папки/проекти, завдання та підзавдання, спільне редагування текстів, керування файлами та доступами, дашборди, шаблони.
- Планування – облік та оцінка витраченого часу виконання завдань, діаграма Ганта, нагадування, загальні календарі, персональна панель задач.
- Комунікації – стрічка новин, що відображає статус виконання завдань у реальному часі, додавання коментарів, обмін повідомленнями, робота з електронною поштою.
- Звітність – інтерактивна статистика, що оновлюється, що відображає хід роботи команди над проектом.
- Гнучкі налаштування системи під особисті вимоги - створення груп користувачів, додавання власних полів у проект та управління панеллю завдань особистого облікового запису.
- Мобільні програми для Android та iOS, десктоп-програми для Mac і indows.
- Інтеграції з електронною поштою, CRM та CMS системами, чатами/месенджерами.

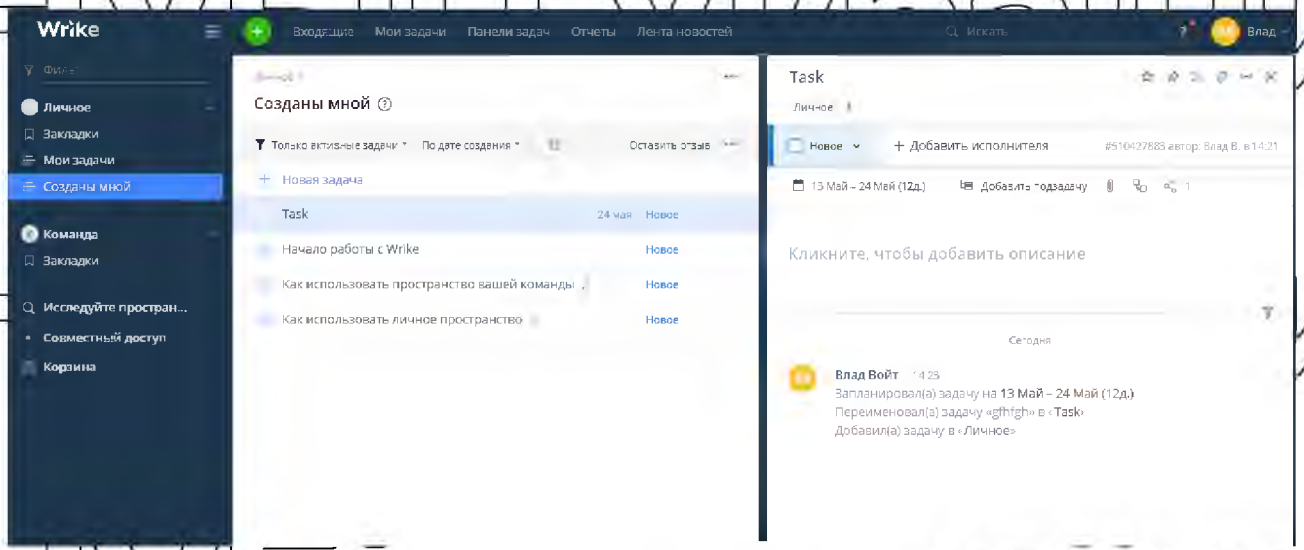


Рис 1.2.1 Wrike

Серед важливих недоліків даної системи є її платне використання та відсутність експорту та імпорту даних.

Також до існуючих рішень було винесено такий веб-сервіс та мобільний додаток, todoist.

Можливості сервісу todoist

- Швидке додавання задач
- Наявність розділів та підзадач
- Повторювання термінів виконання
- Встановлення пріоритетності
- Збережені
- Нагадування
- Доручення завдань
- Сповіщення
- Налаштування під свій стиль
- Історія активності
- Архів виконаних задач

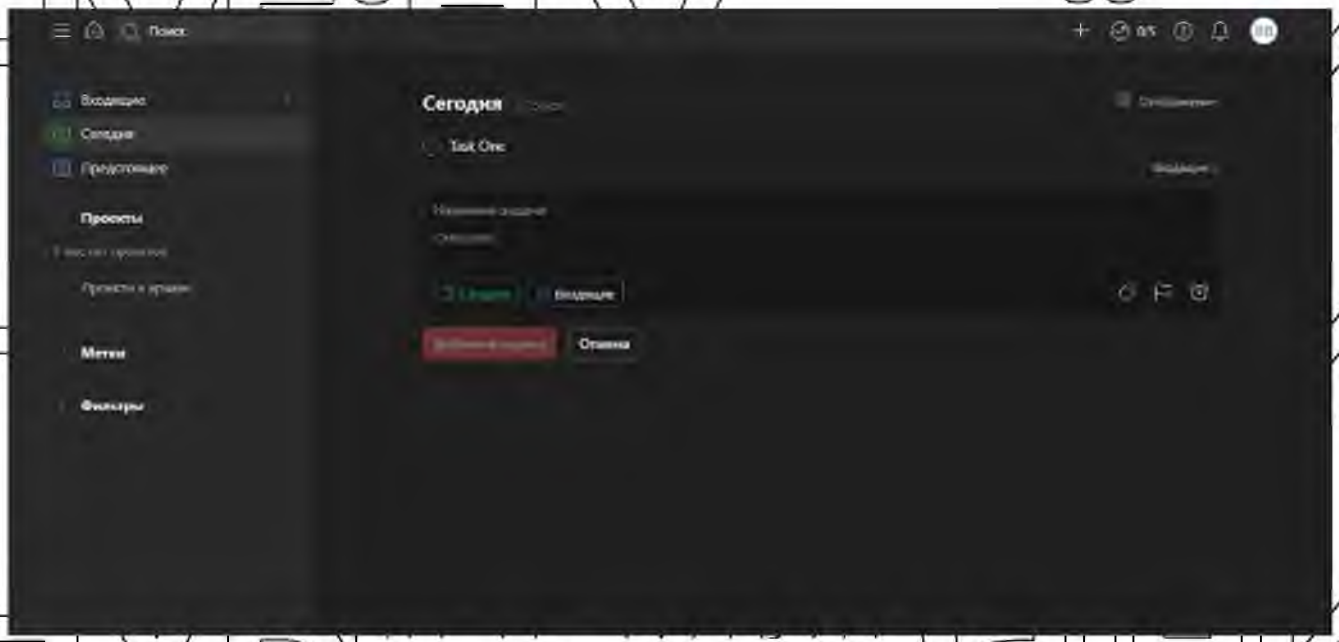


Рис 1.2.2 ToDoist

Ще одним яскравим прикладом для планування робочого час є веб-додаток – Any.do. Any.do - це додаток, що завоював безліч нагород, яким користуються мільйони людей, щоб планувати свій час і встигати більше.

Переваги використовування:

- Простота

- Гнучкість
- Надійність
- Функціональність

Недоліки:

- Постійне нагадування придбати ПРО версію сервісу
- Складний інтерфейс
- Недостатній функціонал для роботи в команді
- Примітивні статистичні дані

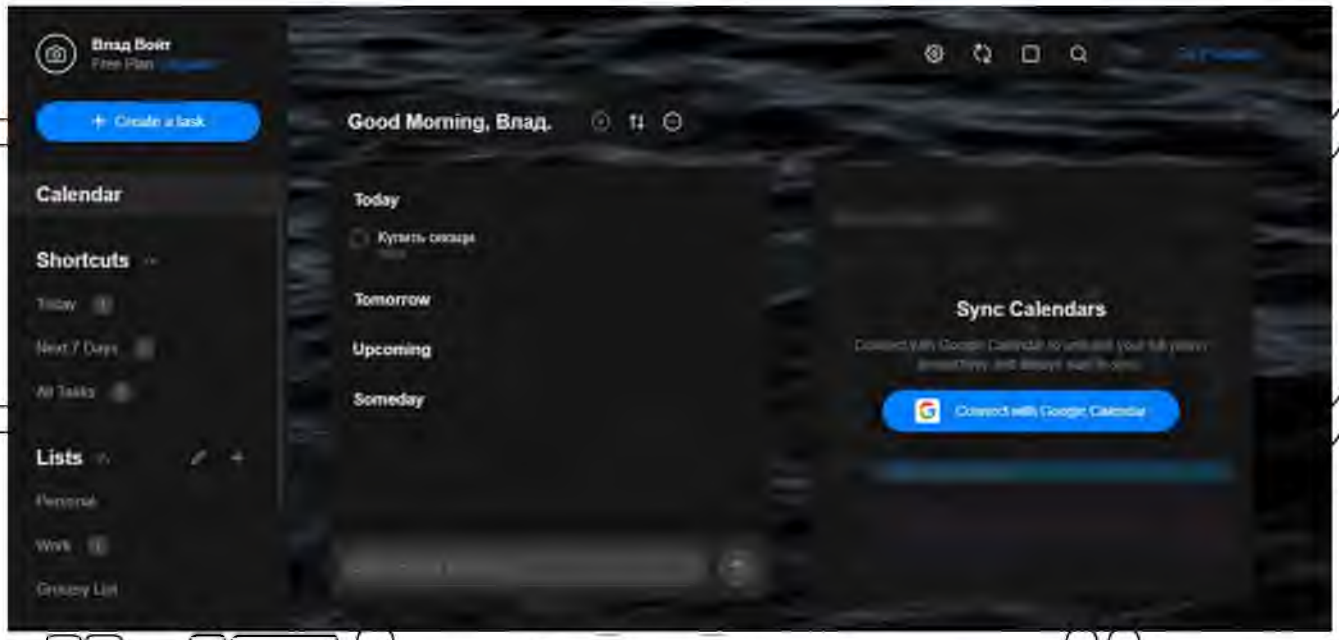


Рис 1.2.3 Any.Do

Таким чином, було прийнято рішення про створення системи, яка зможе надати подібний функціонал для професійного та особистого користування без матеріальних стягнень.

1.3 Аналіз вимог до програмної системи

1.3.1 Функціональні вимоги

Перелік функціональних характеристик системи планування:

1. Долучення нових користувачів до системи;
2. Керування обліковими записами;

НУБіП України

3. Реєстрація компаній до системи;
4. Формування статистики;
5. Додавання, перегляд, видалення та редагування списку завдань;
6. Додавання, перегляд, видалення та редагування списку доручень;

1.3.2 Вихідні дані

Перелік інформації, яку демонструє система:

1. Списки завдань;
2. Списки доручень;
3. Інформація про користувача;
4. Список працівників фірми та інформація про них;
5. Статистичні дані;
6. Календар справ;

1.4 Постановка завдання

Мета роботи полягає у розробці системи планування робочого графіку звичайного користувача та працівника агропромислового комплексу, яка сприятиме більш раціональному використанню часу.

Перелік завдань, котрі вирішує система:

- Планування
- Організація робочого процесу
- Адміністрування бази даних
- Постійний контроль
- Координування завданнями
- Функція редагування

Після успішного вирішення перерахованих вище завдань, керівник матиме змогу сконцентруватися на важливих питаннях та сформувати найбільш оптимальну структуру робочого дня.

НУБіП України

У разі не виконання важливого пункту плану, система завжди надає можливість перенести подію на майбутній період, змінивши кінцевий термін виконання.

Всі дії регулярно перевіряються, адже кожен крок записаний до бази даних, і він може проглядатися будь-якої миті. Це дає менеджеру можливість сфокусуватися на наступних питаннях:

- Чи були виконанні найбільш важливі справи?
- Чи коректно був встановлений термін виконання?
- Рівень ефективності кожного з працівників?
- Чи можна було спланувати день більш результативно?
- Чи правильно розподілені завдання між учасникам компанії?

Практичне значення роботи - це розробка системи, що забезпечує максимальну ефективність більшості підприємств. Мінімізація часу виконання завдань дозволить забезпечити високий рівень ефективності та професіоналізму кожного фахівця.

НУБіП України

2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

UML — це візуальна мова загального призначення, що використовується для моделювання програмних систем. Індустрія програмного забезпечення поділяється на використання діаграм UML.

Переваги UML:

Найбільш поширена і гнучка UML - це широко прийнята і зрозуміла платформа для розробки програмного забезпечення. Це стандартне позначення серед розробників програмного забезпечення. Більшість професіоналів програмного забезпечення можуть з упевненістю припустити, що вони принаймні знайомі з діаграмами UML, навіть якщо це не так. Це альтернатива для пояснення моделі проектування програмного забезпечення. Що робить UML ідеальним і дуже корисним для розробки програмного забезпечення, так це його гнучкість. Ви можете налаштувати елементи моделювання та взаємодії вашої діаграми UML залежно від домену або технології, яку ви використовуєте.

Архітектура програмного забезпечення повинна бути ефективною. Архітектура програмного забезпечення - це план системи. Це каркас, від якого залежить ефективність системи та її дій. Однак ця структура ефективна лише за умови належного сповіщення всіх, хто з нею працює. Тут вступає в гру уніфікована мова моделювання (UML).

UML – це багата й обширна мова, яку можна використовувати для моделювання структури та поведінки додатків і бізнес-процесів, а також об’єктно-орієнтованого програмного забезпечення. Програмний програват погоджується, що не може відмовитися від архітектурної документації. важливий. Він допомагає оцінити продуктивність, безпеку та відстеження, а також надає важливі вказівки для виконання завдання.

Завдяки широкому діапазону UML є ідеальною візуальною мовою для передачі детальних архітектурних деталей найбільшій кількості користувачів.

Щоб користуватися даної технологією, потрібно знати незначну частину мови.

НУБіП України

Існує група різних діаграм UML для моделювання додатків, але розробники використовують лише 3 або 4 для документування своїх програмних систем.

Більш модними залишаються діаграми класів, діаграми послідовності та схеми використання. Отже, щоб пояснити 80% ваших потреб у моделюванні, достатньо знати 20% UML. Вам не потрібно знати або розуміти всі позначення, щоб ефективно спілкуватися за допомогою діаграм UML. Корисно знати підмножину нотації.

Численні інструменти UML Інструменти UML є однією з найважливіших причин, чому UML так широко використовується. Інструменти UML варіюються від безкоштовного програмного забезпечення з відкритим кодом до мільйонів доларів.

Ці інструменти не тільки малюють діаграми, але й охоплюють широкий спектр. Ви можете генерувати код зі свого дизайну, застосовувати шаблони проектування, міняти вимоги, зворотний інженерний код, а також аналізувати вплив і складність.

Ці переваги та багато інструментів UML зробили UML популярною мовою моделювання та розробки в програмній інженерії. Незважаючи на багато використання та переваги, UML віддають перевагу не всім. Насправді більшість розробників програмного забезпечення не використовують UML і не критикують UML. Давайте подивимося на аргументи проти використання UML.

Недоліки UML:

Офіційне позначення не потрібно. Найсильніший аргумент для UML полягає в тому, що вам насправді не потрібна UML-діаграма для передачі вашого проекту. Ви можете отримати той самий ефект і ефект за допомогою неформальної прямокутної діаграми, створеної в PowerPoint, Visio або на дошці. Багатьом розробникам не подобається складність і формат на рівні архітектури, оскільки саме кодування є формальною мовою. Це перешкоджає використанню UML і є одним з його недоліків.

НУБіП України

Труднощі під час використання. Від початку і до теперішнього часу UML зріс у складності та розмірі. Розмір UML настільки великий, що багато людей нервують з самого початку і не можуть вивчити UML, і він не працює без UML.

В основному, архітектурно-незалежний дизайн відноситься до простої базової архітектури програмного забезпечення, яка не вимагає складних діаграм для представлення або опису дизайну. UML вважається непотрібним, якщо компанія приділяє більше уваги формальному кодуванню і широко поширена культура мінімальної конструкторської документації.

2.1 Функціональне моделювання

2.1.1 Діаграма прецедентів

В UML діаграма прецедентів моделює поведінку системи та допомагає охопити вимоги системи. Також, діаграми прецедентів описують функції високого рівня та область застосування системи. Ці діаграми також визначають взаємодії між системою та її акторами. Варіанти використання та дійові особи на діаграмах варіантів використання описують, що робить система і як учасники її використовують, але не те, як система працює всередині.

Діаграми варіантів використання ілюструють і визначають контекст і вимоги або всієї системи, або важливих частин системи. Ви можете змоделювати складну систему за допомогою однієї діаграми варіантів використання або створити багато діаграм варіантів використання для моделювання компонентів системи. Зазвичай ви розробляєте діаграми варіантів використання на ранніх етапах проекту і звертаєтеся до них протягом усього процесу розробки.

Діаграми варіантів використання корисні в таких ситуаціях:

- Перед початком проекту ви можете створити діаграми варіантів використання для моделювання бізнесу, щоб усі учасники проекту мали розуміння працівників, клієнтів та діяльності підприємства.
- Збираючи вимоги, ви можете створювати діаграми варіантів використання, щоб охопити системні вимоги та представити іншим, що має робити система.

- На етапах аналізу та проектування ви можете використовувати варіанти використання та акторів із ваших діаграм варіантів використання, щоб визначити класи, які потрібні системі.

- На етапі тестування ви можете використовувати діаграми прецедентів для визначення тестів системи.

Наступні теми описують елементи моделі на діаграмах варіантів використання:

- Прецеденти. Прецедент слугує випадком використання, котрий описує функцію, яку виконує система для досягнення мети користувача. Випадок використаиня повинен давати спостережуваний результат, який є цінним для користувача системи.

- Актори. Актор представляє роль користувача, який взаємодіє з системою, яку ви моделюєте. Користувачем може бути людина, організація, машина або інша зовнішня система.

- Підсистеми. У моделях UML підсистеми є типом стереотипних компонентів, які представляють незалежні, поведінкові одиниці в системі. Підсистеми використовуються в діаграмах класів, компонентів і варіантів використання для представлення великомасштабних компонентів у системі, яку ви моделюєте.

- Зв'язки в діаграмах варіантів використання. В UML зв'язок — це зв'язок між елементами моделі. Відношення UML — це тип елемента моделі, який додає семантику до моделі, визначаючи структуру та поведінку між елементами моделі.

На створеній діаграмі прецедентів (Рис. 2.1.1) зображено 2 дійових особи, котрі взаємодіють з системою:

1. Рівень доступу «Користувач» - це юзер, який має змогу створювати, редагувати, власні списки справ, а також переглядати статистичні дані. Однією з головних вимог є проходження користувача в системі.
2. Рівень доступу «Керівник» - це юзер, можливості якого схожі до дієвої особи описаною вище в тексті. Його відмінність полягають, в

наявності додаткового функціоналу, завдяки якому він може формувати списки доручень та переглядати список складу команди.

3. «Система» - демонструє процес взаємодії даних з роботою системи та її функціоналу.

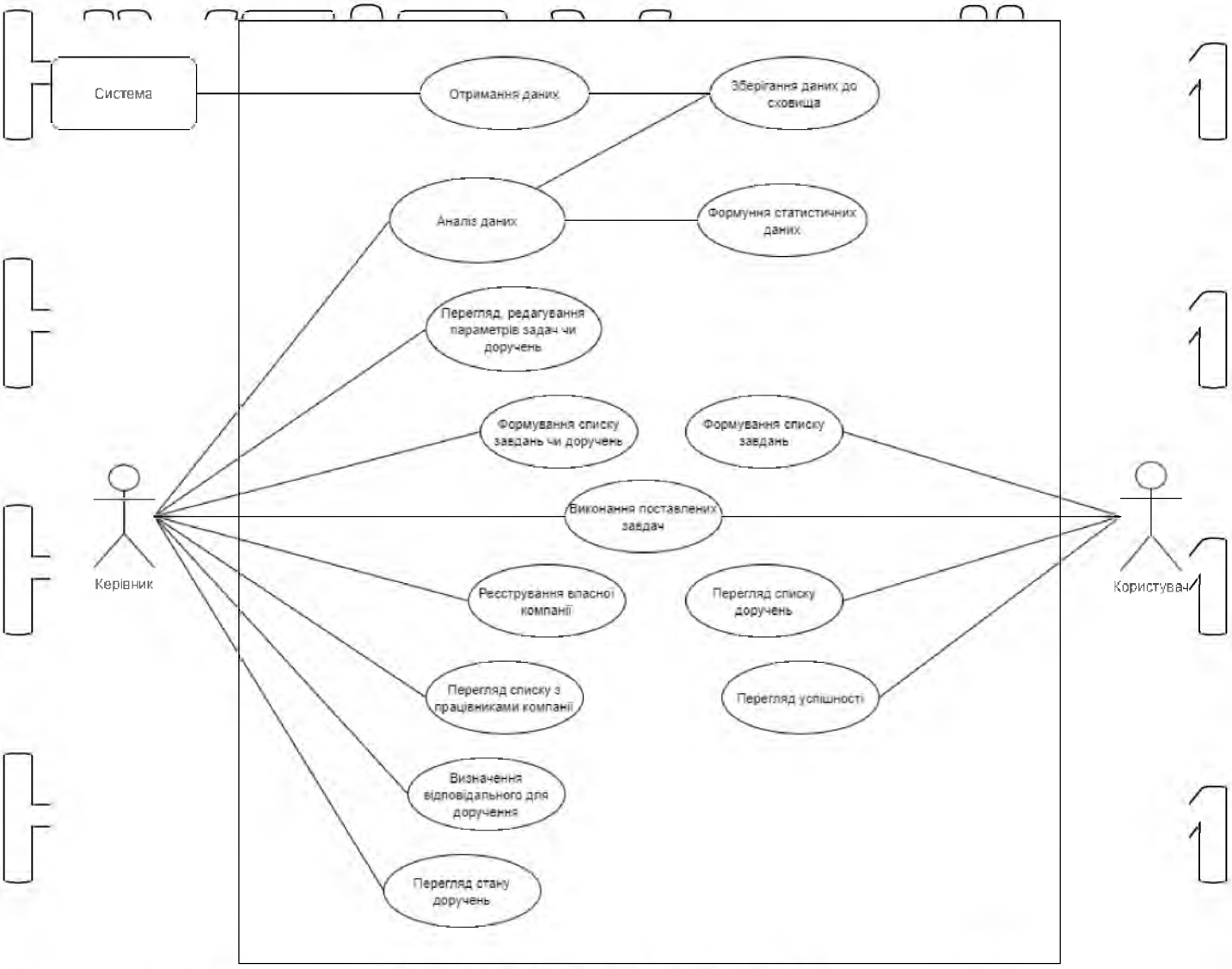


Рис 2.1.1 Діаграма прецедентів

2.1.2 Діаграма послідовностей

Діаграма послідовності використовується в першу чергу для того, щоб показати взаємодію між об'єктами в послідовному порядку, в якому ці взаємодії відбуваються. Подібно до діаграми класів, розробники зазвичай вважають, що діаграми послідовності призначені виключно для них. Однак бізнес-персонал організації може знайти діаграми послідовності, корисні для передачі інформації

про те, як зараз працює бізнес, показуючи взаємодію різних бізнес-об'єктів. Окрім документування поточних справ організації, діаграма послідовності бізнес-рівня може використовуватися як документ вимог для передачі вимог щодо майбутньої реалізації системи. Під час фази вимог проекту аналітики можуть вивести варіанти використання на наступний рівень, забезпечивши більш формальний рівень уточнення. Коли це відбувається, варіанти використання часто перетворюються на одну або кілька діаграм послідовності.

На рис. 2. 1. 2 показаний процес коригування системи у правильній послідовності. Для того, щоб мати доступ до всіх функцій програми, перш за все,

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

потрібно бути авторизованим, після чого всі можливості сайту набувають чинності. Все введене інформаційне забезпечення буде доступне на сервері.

2.1.3 Діаграма діяльності

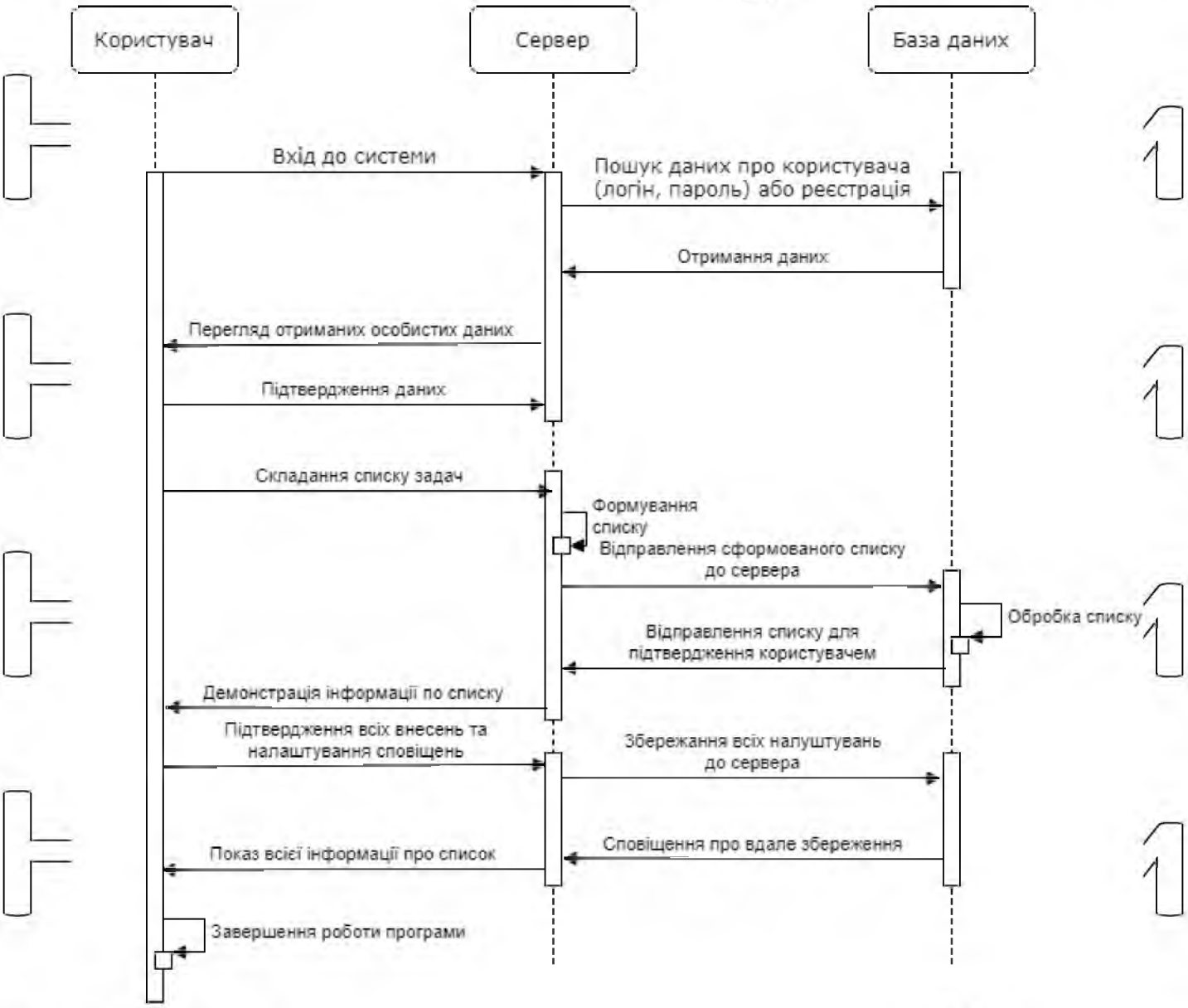


Рис 2.1.2 Діаграма послідовності

Діаграма діяльності або активності – це в основному блок-схема, яка представляє перехід від однієї діяльності до іншої. Діяльність можна описати як операцію системи.

Основні цілі діаграм активності подібні до інших діаграм. Вона фіксує динамічну поведінку системи. Інші діаграми використовуються для показу потоку повідомлень від одного об'єкта до іншого, але діаграма діяльності використовується для показу потоку повідомлень від однієї дії до іншої.

Діяльність — це конкретна операція системи. Діаграми діяльності використовуються не тільки для візуалізації динамічної природи системи, але вони також використовуються для побудови виконуваної системи за допомогою методів прямого та зворотного проектування.

Діаграми діяльності в основному використовуються як блок-схема, яка складається з дій, які виконує система. Діаграми активності не є блок-схемами, оскільки вони мають деякі додаткові можливості. Ці додаткові можливості включають розгалуження, паралельний потік, плавання тощо.

Перш ніж намалювати діаграму діяльності, ми повинні визначити наступні елементи:

- Діяльність
- Асоціація
- Умови
- Обмеження

НУБіП України

Стрілки йдуть від початку до кінця і вказують на порядок, в якому виконуються дії. На створеній діаграмі (Рис. 2.1.3) відображається взаємодія між системою та її авторизованим користувачем.

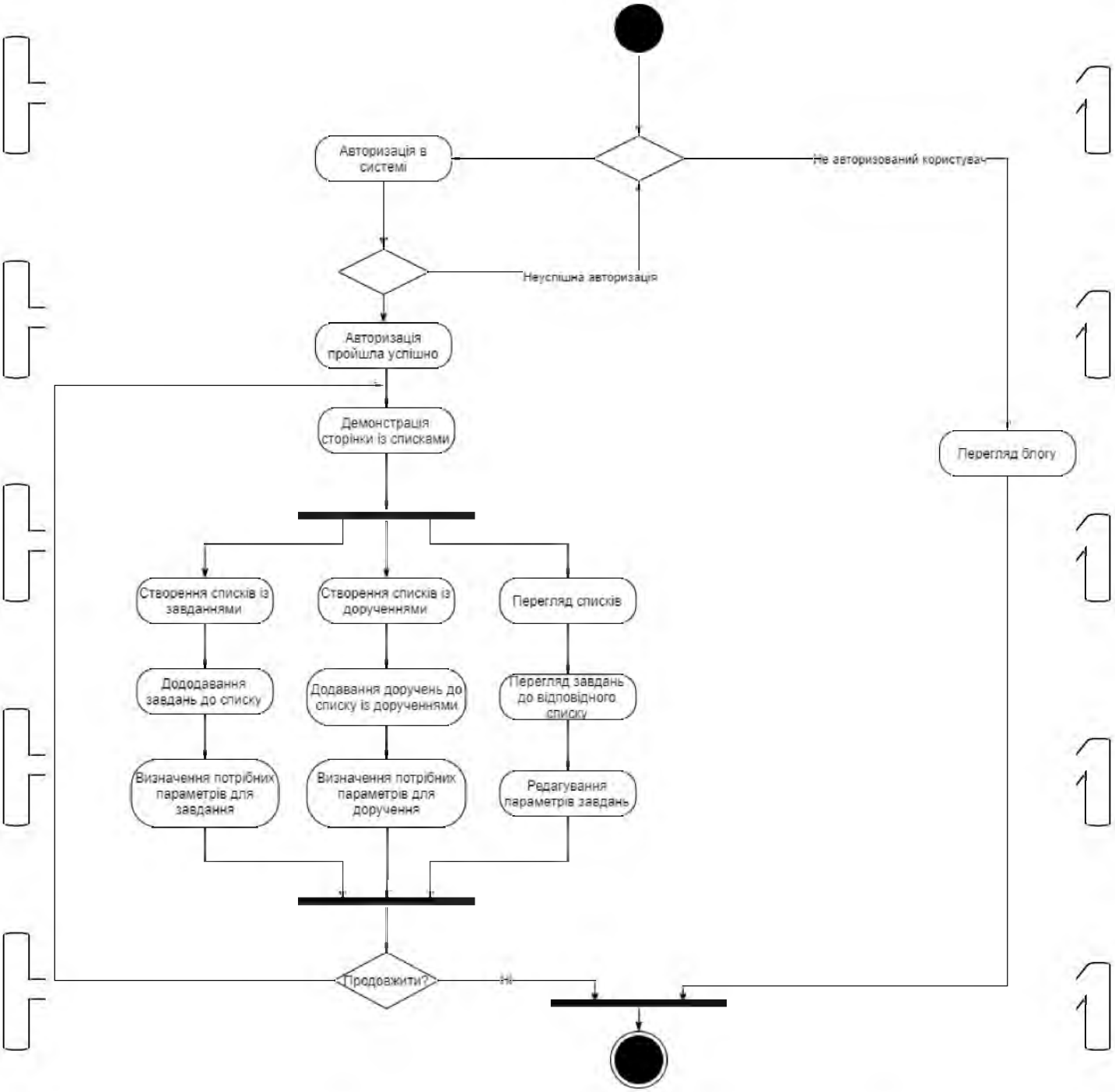


Рис 2.1.3 Діаграма активності

НУБіП України

2.2 Об’єктне моделювання

2.2.1 Діаграма компонентів

Діаграма компонентів — це особливий тип діаграми в UML. Ціль також відрізняється від усіх інших діаграм, які обговорювалися досі. Він не описує функціональність системи, але описує компоненти, які використовуються для створення цих функцій.

Таким чином, з цієї точки зору, діаграми компонентів використовуються для візуалізації фізичних компонентів у системі. Такими компонентами є бібліотеки, пакети, файли тощо.

Діаграми компонентів також можна описати як статичний вигляд системи. Статична реалізація являє собою організацію компонентів у певний момент.

Однокомпонентна діаграма не може представляти всю систему, але набір діаграм використовується для представлення цілого.

Призначення діаграми компонентів можна підсумувати як:

- Демонстрація компонентів системи.
- Створення виконуваних файлів за допомогою прямого та зворотного проектування.
- Роз’яснення взаємозв’язків компонентів.

На розробленому сайті структура будується таким чином, щоб сторінки були пов'язані з файлами, кодом, меню та БД. Файли CSS також включені до опису зображення документа. Кожен елемента відповідає за один компонент або сторінку, яка може бути окремою сторінкою або декількома функціями чи сценаріями управління зв'язаним елементом, продемонстровано на рисунку нижче.

НУБіП України

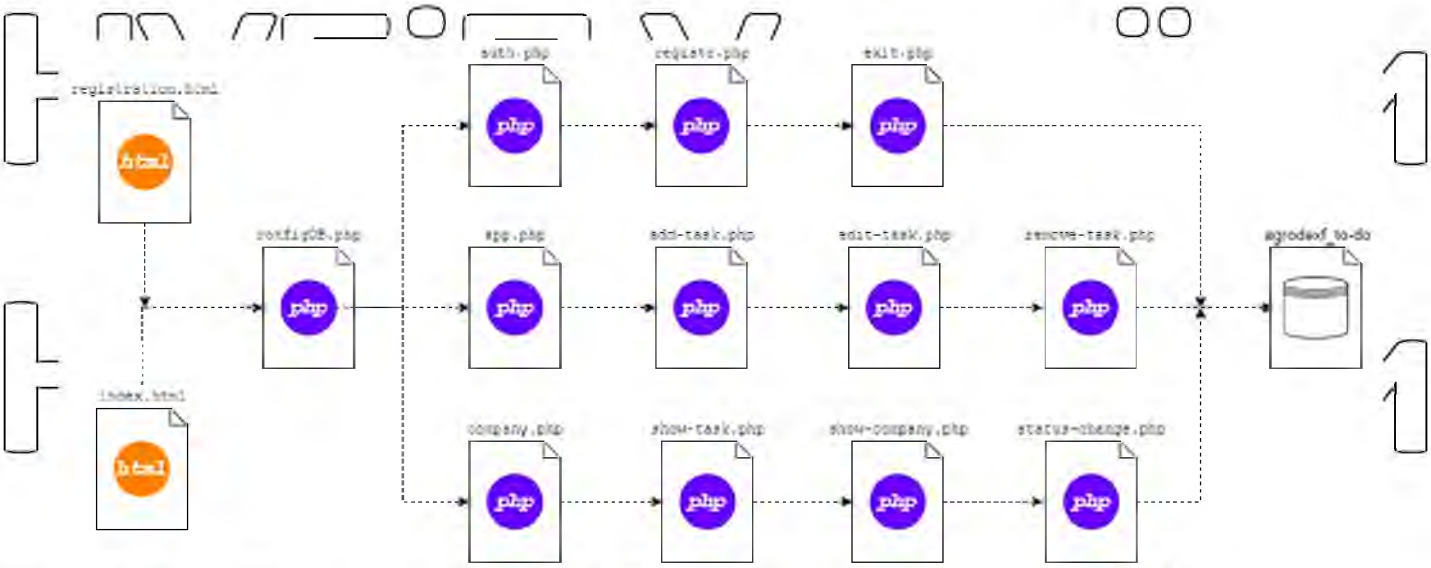


Рис 2.2.1 Діаграма компонентів

«index.html» - головна сторінка сайту

«registration.php» - сторінка з формами реєстрації та авторизацією

«configDB.php» - PHP скрипт підключення до БД

«exit.php» - PHP скрипт виходу користувача з профілю

«registr.php» - форма створення нового профілю

«auth.php» - форма входу до системи

«app.php» - сторінка «Додаток»

«company.php» - сторінка «Компанія»

«remove-task.php» - PHP скрипт видалення задач

«show-task.php» - PHP скрипт відалення поля

«show-company.php» - PHP скрипт показу

«edit-task.php» - PHP скрипт редагування задач та внесення змін до БД

«add-task.php» - PHP скрипт додавання задач до БД

«status-chenge.php» - PHP скрипт для перевірки статусу виконання задач

«agrodexf_to-do.mdb» - база даних

2.2.2 Діаграма пакетів

Діаграми пакетів — це структурні діаграми, які використовуються для показу організації та розташування різних елементів моделі у вигляді пакетів. Пакет – це групування пов'язаних елементів UML, таких як діаграми, документи, класи або

навіть інші пакети. Кожен елемент вкладений в пакунок, який зображений у вигляді папки файлів на діаграмі, а потім впорядкований ієрархічно в межах діаграми. Діаграми пакетів найчастіше використовуються для наочної організації багатошарової архітектури в будь-якому класифікаторі UML, наприклад, у програмній системі.

Переваги пакетної схеми:

- Добре розроблена схема пакетів надає численні переваги тим, хто хоче створити візуалізацію своєї системи або проекту UML.
- Вони забезпечують чітке уявлення про ієрархічну структуру різних елементів UML в межах даної системи.
- Ці діаграми можуть спростити складні діаграми класів у добре впорядковані візуальні елементи.
- Вони пропонують цінний високорівневий огляд великомасштабних проектів і систем.
- Пакетні схеми можна використовувати для візуального уточнення різноманітних проектів і систем.
- Ці візуальні елементи можна легко оновлювати в міру розвитку системи та проектів.

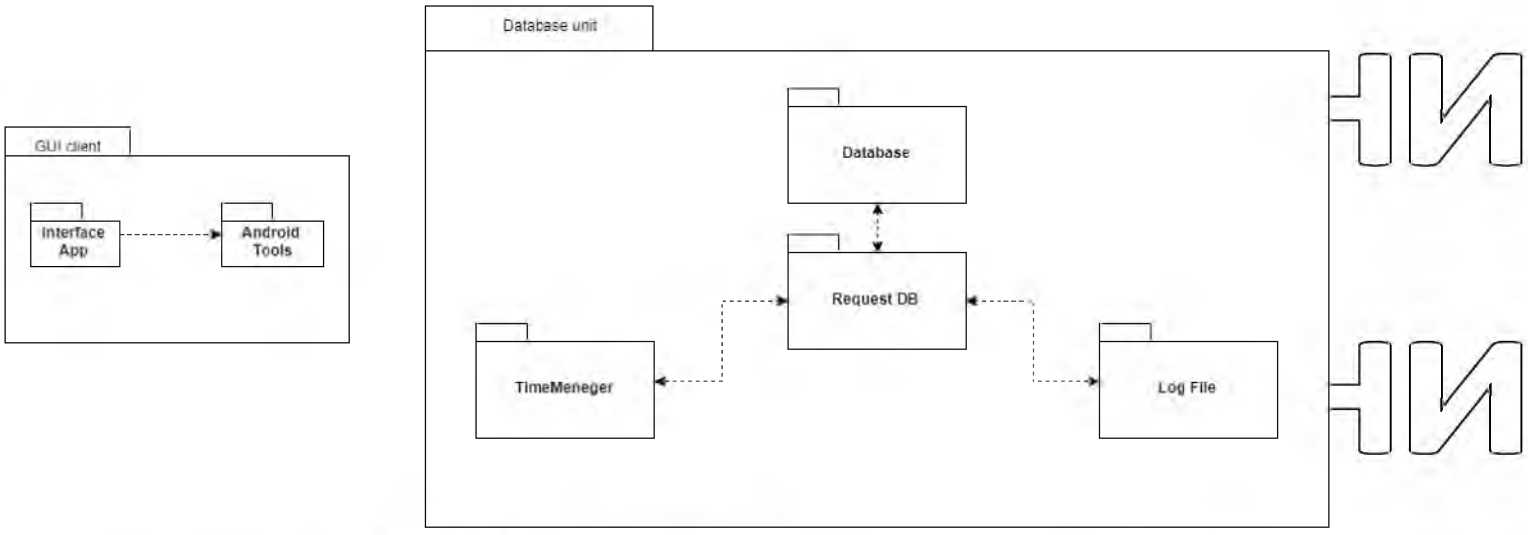


Рис 2.2.2 Діаграма пакетів

НУБіП України

2.2.3 Діаграма класів

Діаграма класів — це статична діаграма, яка використовується для моделювання статичного вигляду системи. Статичний вигляд описує словниковий запас системи.

Діаграма класів також вважається основою для діаграм компонентів і розгортання. Діаграми класів використовуються не тільки для візуалізації статичного вигляду системи, але вони також використовуються для побудови виконуваного коду для прямого і зворотного проектування будь-якої системи.

Як правило, діаграми UML не відображаються безпосередньо з жодними об'єктно-орієнтованими мовами програмування, але діаграма класів є винятком.

У двох словах можна сказати, що діаграми класів використовуються для:

- Опис статичного вигляду системи.
- Показ взаємодії між елементами статичного вигляду.
- Опис функцій, які виконує система.
- Побудова програмних додатків з використанням об'єктно-орієнтованих мов.

Діаграма класів зображена на рисунку 2.2.3

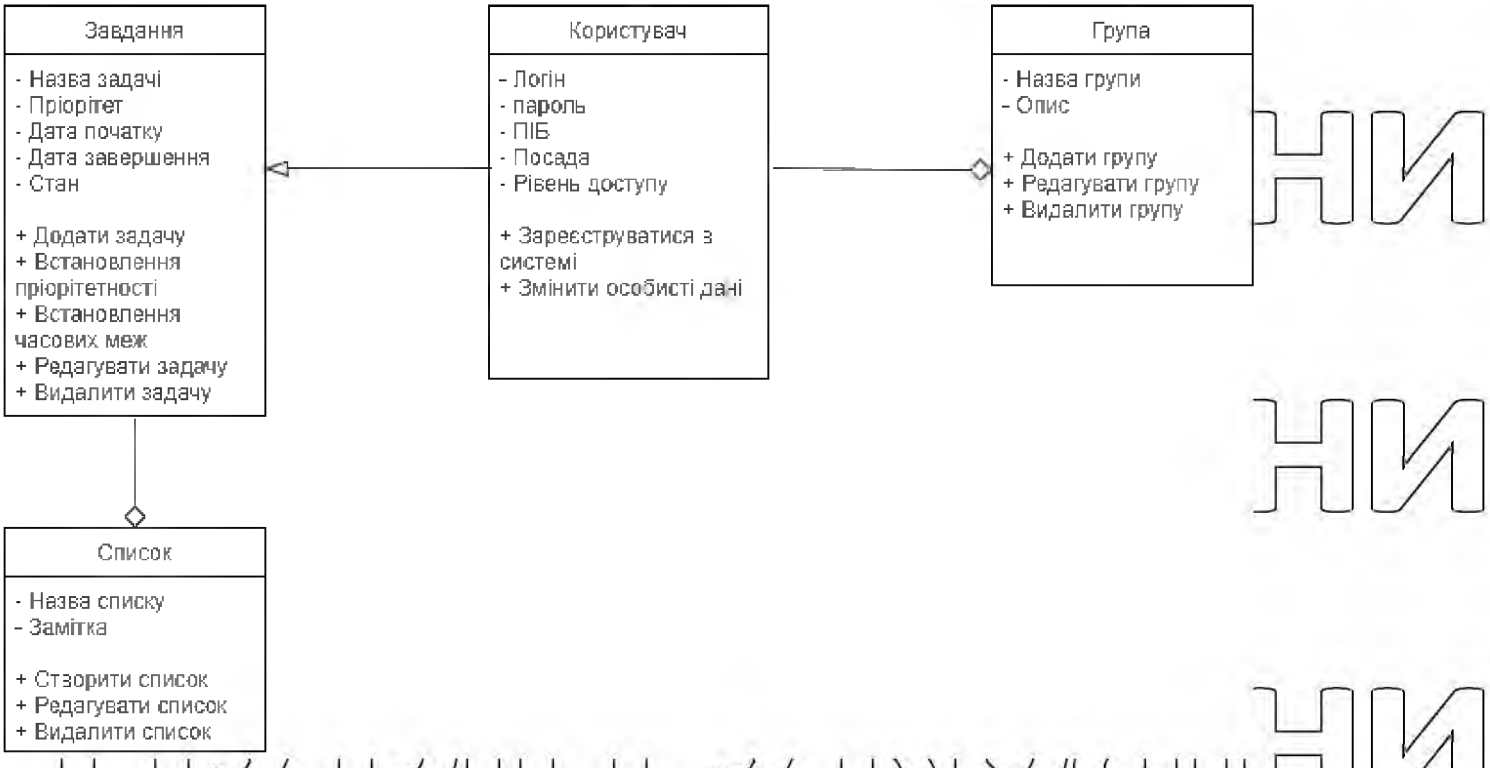


Рис 2.2.3 Діаграма класів

2.2.4 Діаграма розміщення

В UML діаграми розгортання моделюють фізичну архітектуру системи. Діаграми розгортання показують взаємозв'язки між програмними та апаратними компонентами в системі та фізичний розподіл обробки.

Діаграми розгортання, які ви зазвичай готуєте на етапі розробки впровадження, показують фізичне розташування вузлів у розподіленій системі, артефакти, які зберігаються на кожному вузлі, а також компоненти й інші елементи, які реалізують артефакти. Вузли представляють апаратні пристрої, такі як комп'ютери, датчики та принтери, а також інші пристрої, які підтримують середовище виконання системи. Шляхи зв'язку та відносини розгортання моделюють з'єднання в системі.

- Діаграми розгортання ефективні для візуалізації, визначення та документування наступних типів систем:

- Вбудовані системи, які використовують апаратне забезпечення, яке контролюється зовнішніми подразниками; наприклад, дисплей, який керується зміною температури

- Системи клієнт/сервер, які зазвичай розрізняють інтерфейс користувача та постійні дані системи

- Розподілені системи, які мають кілька серверів і можуть розміщувати кілька версій програмних артефактів, деякі з яких можуть навіть мігрувати з вузла на вузол.

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

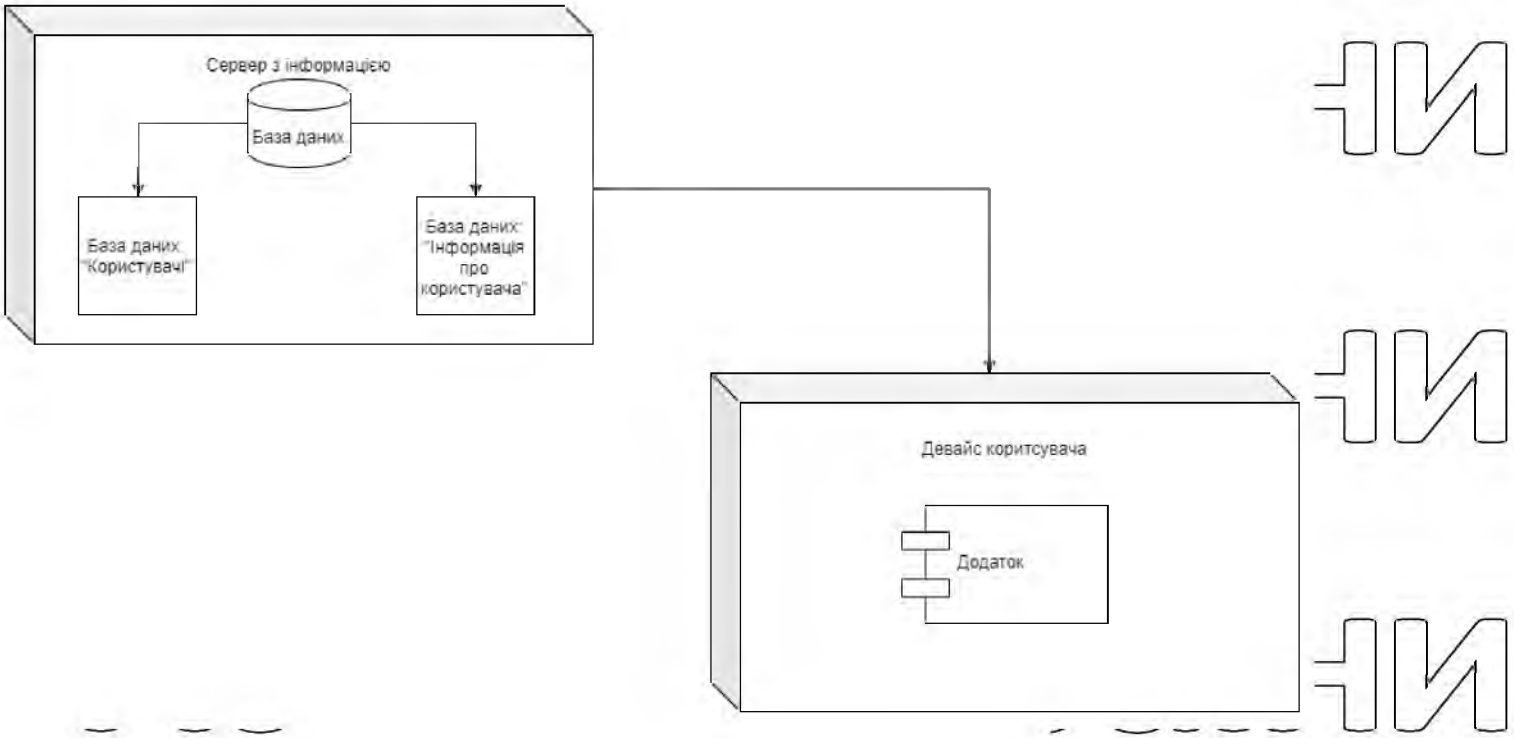


Рис 2.2.4 Діаграма розміщення

НУБіП України
НУБіП України
НУБіП України
НУБіП України

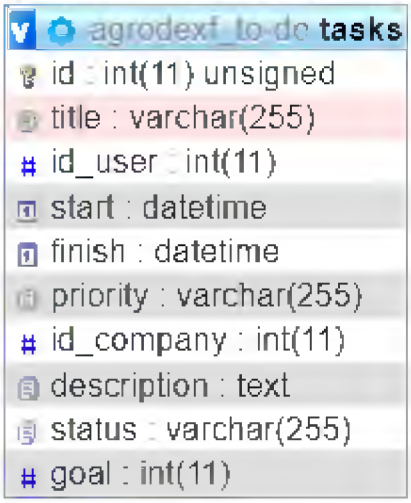
3 ПРОЕКТУВАННЯ СИСТЕМИ

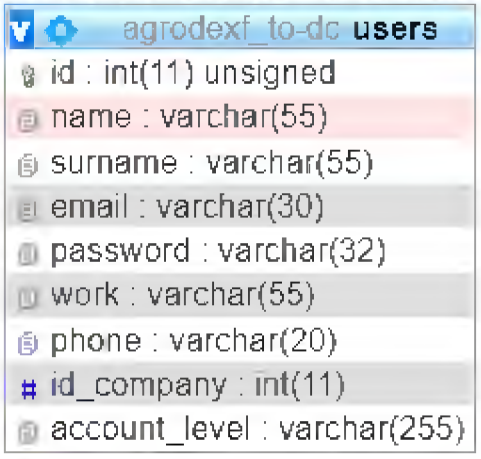
3.1 Створення інформаційної бази

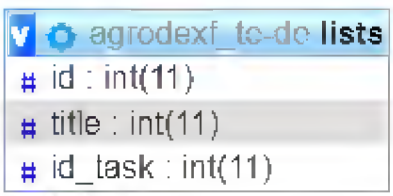
База даних була створена в середовищі «phpMyAdmin» (код продемонстровано нижче), після цього було створено таблиці за допомогою інтсрументу «дизайнер» та SQL-запиту (рис. 3.1):

```
-- База данных: `agrodexf_to-do`
--

-- Структура таблицы `company`
--


CREATE TABLE `company` (
  `id` int(11) NOT NULL,
  `id_user_director` int(11) NOT NULL,
  `company_name` varchar(255) NOT NULL,
  `company_description` text NOT NULL,
  `company_logo` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
-- Структура таблицы `lists`
--


CREATE TABLE `lists` (
  `id` int(11) NOT NULL,
  `title` int(11) NOT NULL,
  `id_task` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
-- Структура таблицы `tasks`
--


CREATE TABLE `tasks` (
  `id` int(11) UNSIGNED NOT NULL,
  `title` varchar(255) NOT NULL,
  `id_user` int(11) NOT NULL,
  `start` datetime NOT NULL,
  `finish` datetime NOT NULL,
  `priority` varchar(255) NOT NULL DEFAULT 'Низький',
```

```
  `id_company` int(11) NOT NULL DEFAULT -1,
  `description` text NOT NULL,
  `status` varchar(255) NOT NULL DEFAULT 'Не виконано',
  `goal` int(11) NOT NULL DEFAULT 5
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
-- Структура таблицы `users`
--

CREATE TABLE `users` (
  `id` int(11) UNSIGNED NOT NULL,
  `name` varchar(55) NOT NULL,
  `surname` varchar(55) NOT NULL,
  `email` varchar(30) NOT NULL,
  `password` varchar(32) NOT NULL,
  `work` varchar(55) NOT NULL,
  `phone` varchar(20) NOT NULL,
  `id_company` int(11) NOT NULL DEFAULT -1,
  `account_level` varchar(255) NOT NULL DEFAULT 'Користувач'
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```







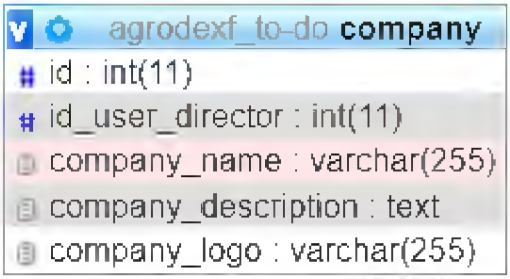


Рис 3.1 Інформаційна база

3.2 Логічна модель даних

Елементи схеми, такі як таблиці та подання, вважаються логічною моделлю бази даних . Ці об'єкти надають інформацію про доступні елементи даних.

Кожна логічна таблиця потребує фізичної моделі. Цю модель можна описати проекцією для таблиці, яка включає всі стовпці таблиці. Проекція з усіма стовпцями таблиці називається проекцією .

- Кожна таблиця, що містить дані, повинна мати принаймні одну проекцію.
- Спеціальну проекцію можна створити вручну перед вставкою даних
- Якщо проекція не визначена для таблиці, вона створюється автоматично, коли перший рядок вставляється в таблицю.

При створенні бази даних сховища даних ви можете почати з однієї проекції для кожної таблиці. Ви можете розглянути можливість додавання додаткових прогнозів від Vertica, щоб покращити продуктивність повільних запитів.

Структура логічної моделі даних було створена за допомогою інструменту Erwin, вміст якої зображено на рисунку 3.2.

Таблиця 3.2

СУТНІСТЬ	*ПРИЗНАЧЕННЯ*
FACT	*Зберігання інформації про доручення*
TIME	*Зберігання інформації про терміни виконання*
WORK	*Зберігання інформації про завдання*
EMPLOYEE	*Зберігання інформації про користувачів*

НУБіП України

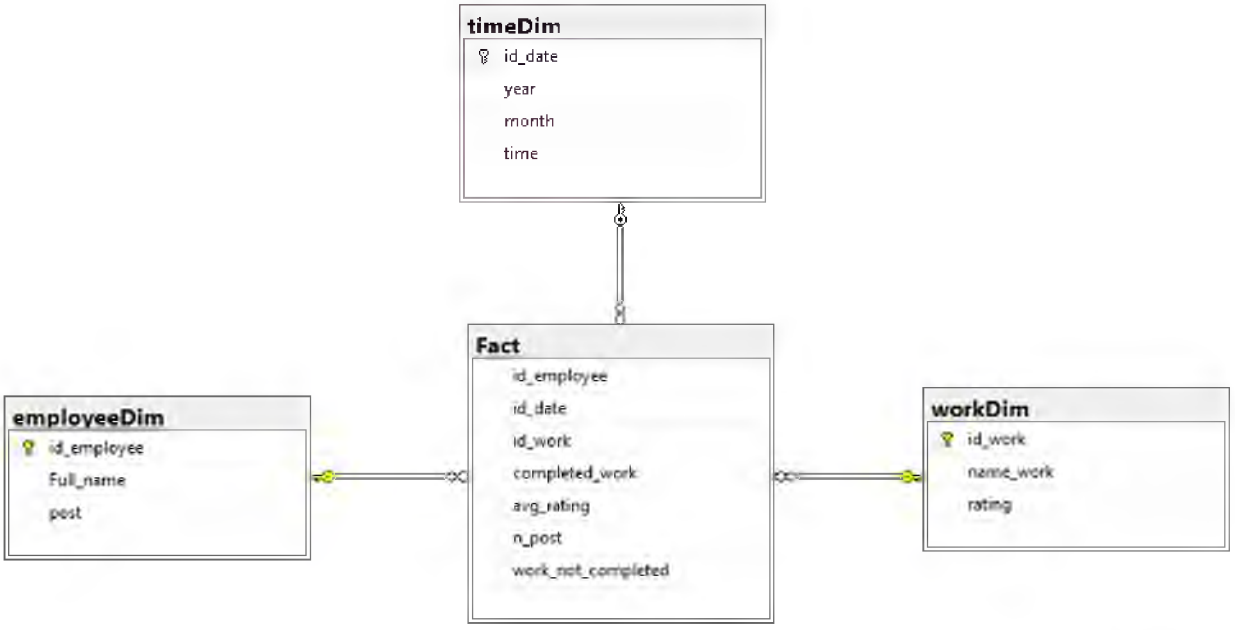


Рис 3.2 Логічна модель даних

3.3 Структура програмної системи

Структура програмної частини веб-додатку зображена у таблиці 3.3

Таблиця 3.3

Компонент	Назва	Опис
Index.html	Головна сторінка	Головна сторінка сайту, яка слугує візиткою та переадресовує користувачів до повного функціоналу додатку
Registration.php	Сторінка авторизації та реєстрації	Сторінка для входу у систему та створення нових профілів
App.php	Додаток	Сторінка, що реалізує процес планування робочого часу
Statistics.php	Статистика	Відображає статистичні дані та календар справ відносно рівня доступу користувача
Company.php	Компанії	Сторінка долучення компанії до системи та її налаштування
Users.php	Працівники	Сторінка, яка відображення членів компанії та інформацію про них
Settings.php	Налаштування	Сторінка редагування особистих даних профілю
Show-task.php		Форма показу списку справ
Add-task.php		Форма створення та додавання задач до списку
Edit-task.php		Форма редагування налаштувань завдань

НУБіП України

Company-task.php	Форма з дорученнями компанії
Show-users.php	Форма, що демонструє всіх працівників, які належать компанії
Registr.php	Форма реєстрування користувачів в системі
Auth.php	Форма авторизації користувачів в системі та створення активної сесії
remove.php	Форма видалення задач із списку

3.4 Розробка системи

Однією з головних задач проекту є надати можливість користувачам користуватися системою з будь-якого пристрою, тому для вирішення даної проблемо було прийнято рішення розробити додаток у вигляді веб-застосунку.

Для розробки системи було використано перелік технологій, середовищ та сервісів, котрі розміщені нижче в межах розділу 3.4.

3.4.1 HTML

HTML означає HyperText Markup Language. Він використовується для оформлення веб-сторінок за допомогою мови розмітки. HTML - це комбінація гіпертексту та мови розмітки. Гіпертекст визначає зв"язок між веб-сторінками. Мова розмітки використовується для визначення текстового документа в тегу, який визначає структуру веб-сторінок. Ця мова використовується для анотування (зроблення нотаток для комп'ютера) тексту, щоб машина могла його зрозуміти та відповідно маніпулювати текстом. Більшість мов розмітки (наприклад, HTML)

НУБіП України

читаються людиною. Мова використовує теги, щоб визначити, які маніпуляції потрібно виконати з текстом.

HTML – це мова розмітки, що використовується браузером для маніпулювання текстом, зображеннями та іншим вмістом, щоб відобразити його в необхідному форматі.

Особливості HTML:

- Його легко навчитися і легко використовувати.
- Він не залежить від платформи.
- На веб-сторінку можна додавати зображення, відео та аудіо.
- До тексту можна додати гіпертекст.
- Це мова розмітки.

Переваги:

- HTML використовується для створення веб-сайтів.
- Він підтримується всіма браузерами.
- Його можна інтегрувати з іншими мовами, такими як CSS, JavaScript тощо.

Недоліки:

- HTML може створювати лише статичні веб-сторінки. Для динамічних веб-сторінок необхідно використовувати інші мови.
- Щоб створити просту веб-сторінку, потрібно написати велику кількість коду.

3.4.2 CSS

CSS (каскадні таблиці стилів) — це мова таблиць стилів, яка використовується для оформлення веб-сторінки, щоб зробити її привабливою. Причина використання CSS полягає в тому, щоб спростити процес створення презентаційних веб-сторінок. CSS дозволяє застосовувати стилі до веб-сторінок. Що ще важливіше, CSS дозволяє вам робити це незалежно від HTML, який складає кожну веб-сторінку.

Існує три типи CSS, які наведені нижче:

- Вбудований CSS
- Внутрішній або вбудований CSS

- Зовнішній CSS

Чому обрано технологію CSS?

- CSS економить час: ви можете написати CSS один раз і повторно використовувати один і той же аркуш на кількох сторінках HTML.
- Легке обслуговування: щоб внести глобальні зміни, просто змініть стиль, і всі елементи на всіх веб-сторінках оновлюватимуться автоматично.
- Пошукові системи: CSS вважається технікою чистого кодування, що означає, що пошуковим системам не доведеться стикатися з «читанням» його вмісту.
- Кращі стилі перед HTML: CSS має набагато ширший набір атрибутів, ніж HTML, тому ви можете надати набагато кращий вигляд своїй HTML-сторінці в порівнянні з атрибутами HTML.
- Офлайн-перегляд: CSS може зберігати веб-програми локально за допомогою автономного кешу. Використовуючи це, ми можемо переглядати офлайн-сайти.

3.4.3 JavaScript

JavaScript є найпопулярнішою в світі легкою, інтерпретованою компільованою мовою програмування. Він також відомий як мова сценаріїв для веб-сторінок. Він добре відомий розробкою веб-сторінок, його також використовують багато небраузерних середовищ. JavaScript можна використовувати для розробок на стороні клієнта , а також для розробок на стороні сервера .

JavaScript можна просто додати до вашого HTML-файлу двома способами :

- Внутрішній JS: ми можемо додати JavaScript безпосередньо до нашого HTML-файлу, написавши код всередині тегу <script> . Тег <script> може бути розміщений всередині тегу < head> або < body> відповідно до вимоги.
- Зовнішній JS: ми можемо записати код JavaScript в інший файл із розширенням .js, а потім зв'язати цей файл із тегом < head> HTML-файлу, до якого ми хочемо додати цей код.

НУБіП України

JavaScript є широко поширеною мовою програмування. Нижче наведено деякі домени/продукти, які можна створити за допомогою JavaScript:

- Веб-сайти: JavaScript допомагає нам додати поведінку нашого веб-сайту. Це допомагає користувачам взаємодіяти з веб-сайтом. Для напр. натискання кнопок, збереження деталей, завантаження деталей на веб-сайт тощо.

- Веб-сервери: ми можемо створювати надійні серверні програми за допомогою JavaScript. Якщо бути точним, ми використовуємо фреймворки JavaScript, такі як Node.js і Express.js для створення цих серверів.

- Розробка ігор: У галузі розробки ігор широко використовується JavaScript. З додаванням HTML5 Canvas тепер можна дуже ефективно створювати 2D та 3D ігри на JavaScript.

- 3D-малюнки: JavaScript на додаток до HTML Canvas використовується для створення тривимірної графіки.

- Мобільні програми: Мобільні програми є найпопулярніщими способами спілкування в наші дні. JavaScript також використовується для розробки мобільних додатків. Існує багато фреймворків JavaScript, за допомогою яких ми можемо створювати Android, IOS та гібридні програми.

Як і всі мови програмування, JavaScript має деякі переваги та недоліки, які слід враховувати. Багато з них пов'язані з тим, як JavaScript виконується безпосередньо в браузері. Але є й інші способи використання JavaScript, які тепер дають нам ті ж переваги, що й мова програмування на стороні сервера.

Переваги JavaScript:

- Швидкість JavaScript, як правило, дуже висока, оскільки він часто запускається безпосередньо в браузері. Таким чином, JavaScript не може бути відкладений викликами до сервера, доки не потрібні зовнішні ресурси.

- Простота: синтаксис JavaScript натхненний Java і його відносно легко вивчити в порівнянні з іншими популярними мовами програмування, такими як C++.

- Популярність: JavaScript поширений у всьому Інтернеті, і з появою Node.js його використання на стороні сервера зросло. Існує багато ресурсів для вивчення

JavaScript. Як StackOverflow, так і GitHub демонструють зростаючу кількість проектів, що використовують JavaScript, і очікується, що популярність, яку він досяг за останні роки, продовжуватиме зростати.

- Сумісність. На відміну від PHP або інших мов програмування, JavaScript можна використовувати на будь-якому веб-сайті. JavaScript можна використовувати в різних типах програм завдяки підтримці інших мов, таких як Pearl і PHP.

- Завантаження сервера — це JavaScript на стороні клієнта, тому загальний попит на сервер зменшується, і простим додаткам може взагалі не знадобитися сервер.

- Прості інтерфейси: JavaScript можна використовувати для створення функцій і компонентів перетягування, таких як слайди, які значно покращують інтерфейс користувача та роботу сайту.

- Розширена функціональність: розробники можуть розширити функціональність веб-сторінок за допомогою фрагментів JavaScript для третьої групи розширень, таких як Greasemonkey.

- Універсальність. Існує багато способів використання JavaScript із серверами Node.js. Коли ви завантажуєте Node.js за допомогою Express, ви можете використовувати документ бази даних, наприклад MongoDB, і використовувати JavaScript в інтерфейсі клієнта. Ви можете розробити повну програму JavaScript від початку до кінця, просто використовуючи JavaScript.

- Регулярне оновлення. З моменту впровадження ECMAScript 5 (письмової специфікації, на якій базується JavaScript), ECMA International щорічно оновлює JavaScript. Ми отримали підтримку для браузера ES6 у 2017 році, і ми очікуємо, що ES7 буде підтримуватися в майбутньому.

Недоліки JavaScript:

- Безпека на стороні клієнта. Оскільки код JavaScript виконується на стороні клієнта, іноді можна використовувати помилки та збої. У результаті деякі люди вирішують повністю вимкнути JavaScript.

НУБіП України

- Підтримка браузера. Хоча сценарії на стороні сервера завжди дають однаковий результат, різні браузери іноді по-різному інтерпретують JavaScript. Сьогодні відмінності мінімальні, і вам не доведеться турбуватися про перевірку коду в більшості браузерів.

3.4.4 PHP

PHP (PHP: Hypertext Preprocessor - PHP: hypertext preprocessor) — це загальна мова сценаріїв програмування, широко використовується для веб-додатків Development De Web. Використовується більшістю підтримуваних на даний момент постачальників послуг і є однією з основних мов програмування для створення потужних веб-сайтів .

Ця мова розробляється та перекладається групою ентузіастів у рамках проекту з відкритим кодом. Проект поширюється за його ліцензією, яка не сумісна з GNU GPL.

З точки зору веб-програмування, PHP є однією з найпопулярніших мов сценаріїв (включаючи мови, що використовуються в JSP, Perl і ASP.NET) через свою простоту, швидке впровадження, ефективну роботу, кросплатформенність. І розподіл. Вихідний код заснований на ліцензії PHP.

Популярність веб-дизайну визначається великою кількістю інструментів, створених для оптимізації веб-додатків. Найголовніше:

- Автоматичне видалення налаштувань POST і GET, а також змінних веб-сервера у попередньо визначених конфігураціях;
- Працює з різними системами управління даними;
- Відправляти перенаправлення HTTP напряму;
- Робота з дозволами HTTP;
- Робота з файлами cookie та сеансами;
- Робота з локальними та віддаленими файлами, пілінг.
- Керувати файлами, завантаженими на сервер;

В даний час PHP використовується сотнями тисяч розробників. Згідно з рейтингом пошукових систем TIOBE, у квітні 2011 року PHP посів 5 місце з мов

програмування. Основними веб-сайтами, які використовують PHP, є Facebook, ВКонтакте, Вікіпедія та інші.

Входить до складу LAMP - найпопулярнішого програмного забезпечення для створення веб-сайтів (Linux, Apache, MySQL, PHP).Синтаксис мови походить від C, Java та Perl. PHP досить легко вивчити. Перевага PHP полягає в тому, що веб-розробники можуть швидко створювати динамічні веб-сайти.

3.4.5 jQuery

jQuery — бібліотека JavaScript, що містить готові функції мови JavaScript, всі операції jQuery виконуються з коду JavaScript. Методи jQuery дають змогу маніпулювати вмістом веб-сторінки. Методи надають задані дії відібраним в jQuery - об'єкт елементам, внаслідок чого відбувається динамічна зміна елементів та їх вмісту.

Маніпуляції HTML-елементами дозволяють уточнювати, розширювати обгорнутий набір елементів шляхом додавання нового вмісту або видаляти елементи набору. Щоб маніпулювати елементами, їх потрібно відібрати за допомогою селекторів чи методів відбору.

Кожен метод або сам щось повертає, або отримує параметр і виконує, що вказано у параметрі.

3.4.6 Bootstrap

Після випуску з відкритим кодом у 2011 році Bootstrap став популярним дуже швидко і недарма. Веб-дизайнерам і веб-розробникам подобається Bootstrap, оскільки він гнучкий і простий у роботі. Його основні переваги полягають у тому, що він адаптивний за дизайном, підтримує широку сумісність із браузером, пропонує послідовний дизайн із використанням компонентів, які можна повторно використовувати, а також дуже простий у використанні та швидкий у навчанні. Він пропонує багаті можливості розширення за допомогою JavaScript, а також вбудовану підтримку плагінів jQuery і програмний JavaScript API.

Bootstrap може бути використаний з будь-яким IDE або редактором, а також з будь-якою технологією та мовою на сервері від ASP. NET та PHP та Ruby On Rails. За допомогою Bootstrap веб-розробники можуть сфокусуватися на розробці роботи

без занепокоєння дизайном і швидко створити хороший сайт. Загалом це дає веб-дизайнерам солідну базу для створення цікавих шаблонів для Bootstrap.

3.4.7 phpMyAdmin

PHPMyAdmin є програмою, написаною на PHP, яка забезпечує повну, включаючи віддалену роботу з MySQL базами в браузері. Оскільки PHPMyAdmin у більшості випадків дозволяє безпосередньо вводити команди SQL, то робота з базою даних виявляється цілком успішним завданням, навіть у людини, яка досить поверхово знайома з SQL.

Існує ряд додатків для наповнення бази даних її структури, зокрема PHPMyAdmin, MySqlManager, MySQL Administrator, інструменти MySQL тощо. PHPMyAdmin був обраний з усіх програм баз даних, оскільки він пропонує наступні переваги:

- Безкоштовне розповсюдження. PHPMyAdmin - безкоштовна програма.
- Актуальність. Більшість розробників використовують PHPMyAdmin як додаток СУБД.
- Інтуїтивно зрозумілий інтерфейс. Ви можете відмовитися від прямого введення операторів SQL побачити результат.
- Наявність документів. Існує багато навчальних посібників для роботи з PHPMyAdmin, включених до стандартного веб-сервера XAMPP.

Завдяки підтримці ініціативи GoPHP5, для роботи з phpMyAdmin 3.x.x. необхідні MySQL 5 і PHP 5.2 .... Щоб використовувати старіші версії MySQL і PHP, розробники все ще підтримують гілку phpMyAdmin 2.x.x.

Завдяки даному середовищу було створено сховище даних та його логічну модель.

3.4.8 Visual Studio Code

Для роботи з програмним кодом було використано середовище Visual Studio Code, яке дарує максимальну зручність при написанні коду на будь-яких мовах програмування.

Visual Studio Code — це редактор вихідного коду, створений Microsoft для Windows, Linux і macOS. Функції включають підтримку налагодження, виділення синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git.

3.4.9 Hostiq.ua

Для розміщення веб-додатку в мережі інтернет було обрано хостинг – HOSTiQ.ua.

HOSTiQ – українська компанія, яка працює на ринку понад 10 років. Компанія надає все необхідне для розміщення веб-сайтів в Інтернеті - від хостингу проектів будь-якого розміру до доменів і сертифікатів SSL. HOSTiQ допомагає клієнтам створювати веб-сайти та надає чудову підтримку.

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

4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ

4.1 Інтерфейс та програмування функціональних модулів системи

На рисунку 4.1.1 зображено головну сторінку сайту, яка містить в собі меню, три секції контенту та два посилання у вигляді кнопок, котрі введуть до повного функціоналу системи.

Елемент «header» виступає в ролі головного меню сторінки та об'єднує в собі якорі до секцій та посилання до сторінки registration.php.

Секції контенту:

- Блок «Головна» - складається з вступного тексту та посилання натиснувши на яке, користувач перенаправляється до додатку app.php.
- Блок «Можливості» - демонструє функціональні та нефункціональні вимоги проекту.
- Блок «Для команди» - відображає перелік функцій для користувачів з рівнем доступу «Працівник» та «Керівник».

Рис. 4.1.1 Головна сторінка сайту

НУБіП України

Натиснувши на кнопку «Авторизація» або «Розпочати» користувач переадресовується до сторінки registration.php (Рис. 4.1.2), де він отримує можливість зареєструватися чи увійти в систему. Для зменшення об'єму структури сайту, було прийняти рішення об'єднати обидві форми на одній сторінці. Для вирішення цієї задачі було використано технології JS та CSS, які надали даній сторінці динаміку та сучасний і водночас простий дизайн. Також на зображеннях нижче продемонстровано візуальну складову та процес запису користувача до сховища даних.

```php
$name = filter_var(trim($_POST['name']),
    FILTER_SANITIZE_STRING);
    $surname = filter_var(trim($_POST['surname']),
    FILTER_SANITIZE_STRING);
    $email = filter_var(trim($_POST['email']),
    FILTER_SANITIZE_STRING);
    $pass = filter_var(trim($_POST['password']),
    FILTER_SANITIZE_STRING);
    $work = filter_var(trim($_POST['work']),
    FILTER_SANITIZE_STRING);
    $phone = filter_var(trim($_POST['phone']),
    FILTER_SANITIZE_STRING);
    $level = filter_var(trim($_POST['acc-level']),
    FILTER_SANITIZE_STRING);

require '../configDB.php';

    $query = "INSERT INTO `users` (`name`, `surname`, `email`, `password`, `work`, `phone`, `account_level`)
    VALUES('$name', '$surname', '$email', '$pass', '$work', '$phone', '$level')";
    if(isset($_POST['company'])){
        $query = "INSERT INTO `users` (`name`, `surname`, `email`, `password`, `work`, `phone`, `account_level`, `id_company`)
        VALUES('$name', '$surname', '$email', '$pass', '$work', '$phone', '$level', '{$_POST['company']}')";
    }

    $conn->query($query);

    $conn->close();
```

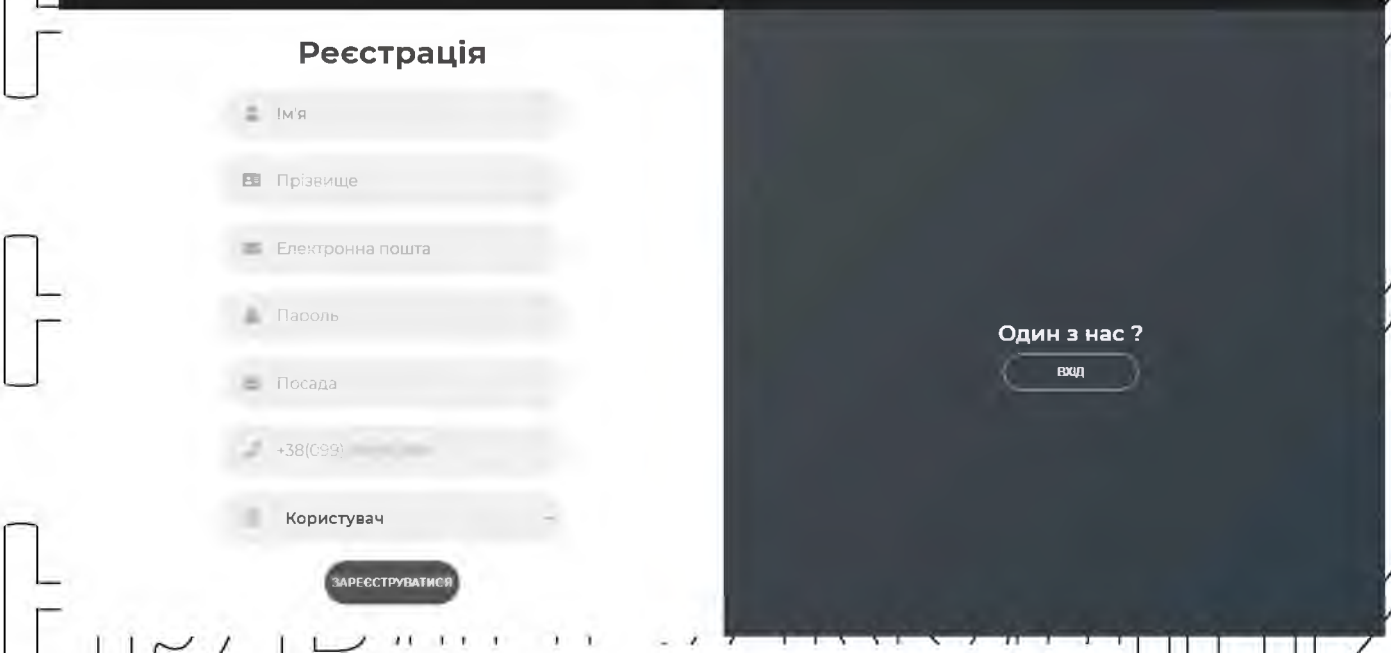


Рис. 4.1.2 Форма реєстрації

Для авторизації користувача в системі було використано метод setcookie, в успішному результаті якого, юзер отримує статус «активний» на протязі 24 годин, після його закінчення – повторити авторизацію.

```php
$email = filter_var(trim($_POST['email']),
    FILTER_SANITIZE_STRING);
    $pass = filter_var(trim($_POST['password']),
    FILTER_SANITIZE_STRING);

    require '../configDB.php';

    $result = $conn->query("SELECT * FROM `users` WHERE `email` = '$email'
    AND `password` = '$pass'");
    $user = $result->fetch_assoc();
    if(count($user) == 0) {
        echo "<script>alert('You are logged out');
        location.href='../registration.php';
        </script>";
        exit();
    };

    $result = $conn->query("SELECT * FROM `company` WHERE `id` = '{$user['id_company']}'");
    $company = $result->fetch_assoc();

    setcookie('user', $user['name'], time() + 3600 * 24, "/");
    setcookie('surname', $user['surname'], time() + 3600 * 24, "/");
    setcookie('work', $user['work'], time() + 3600 * 24, "/");
    setcookie('id', $user['id'], time() + 3600 * 24, "/");
    setcookie('company_id', $user['id_company'], time() + 3600 * 24, "/");
    setcookie('company_name', $company['company_name'], time() + 3600 * 24, "/");

    $conn->close();
```

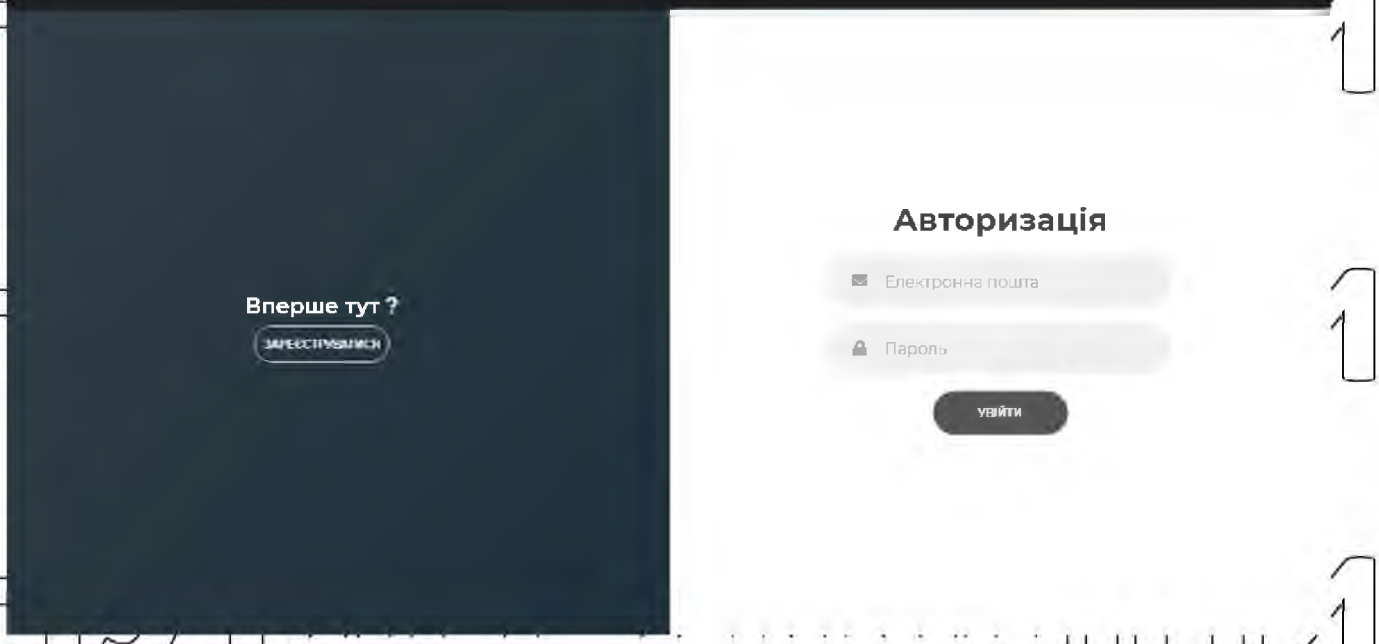


Рис. 4.1.3 Форма авторизації

На рисунку 4.1.4 зображено меню додатку з різним рівнем доступу. Для кожного з користувачів характерний власний перелік функціоналу додатку.

Користувач з рівнем доступу «Користувач»:

- Можливість додавати, редагувати список задач
- Перегляд власної успішності
- Редагування власного профілю

Користувач з рівнем доступу «Працівник»:

- Додатково до функціоналу «Користувача» з'являться сторінка з дорученнями за які відповідає власне «Працівник» та його головним обов'язком є виконати задачу вчасно
- На сторінці статистика з'являється новий блок з власною успішністю в компанії

Користувач з рівнем доступу «Керівник»:

- Можливість створення власної компанії та додавання учасників
- Створення доручень та визначення відповідальних
- Перегляд статистики компанії та членів компанії

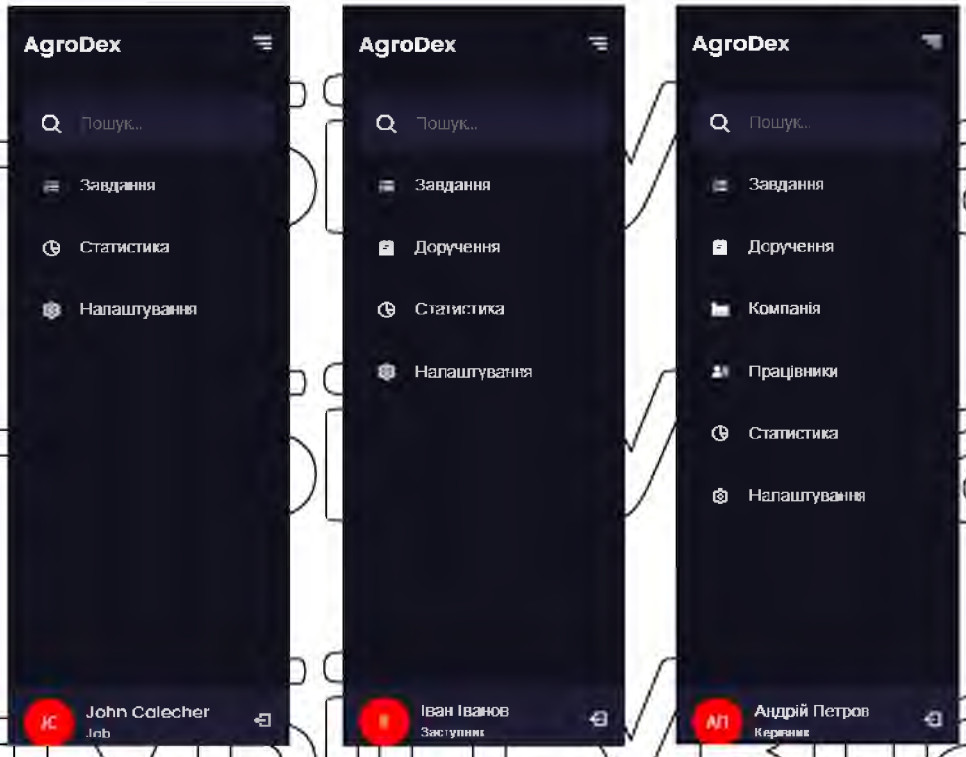


Рис. 4.1.4 Меню додатку користувачів

НУБіП України

Рисунок розташований нижче демонструє візуальну та програмну складову розділу із власними завданнями. На зображені представлено процес долучення задач до сховища даних та їх виведення у вигляді списку, як зі сторони користувача, так і додатку у вигляді програмного коду.

```php
require '../configDB.php';

$task   = $_POST['task'];
$id     = $_POST['id'];


if (isset($_POST['company_id'])){
    $sql = "INSERT INTO tasks (title, id_user, id_company) VALUES ('$task', '$id', '{$_POST['company_id']}')";
}
else{
    $sql = "INSERT INTO tasks (title, id_user, id_company) VALUES ('$task', '$id', '-1')";
}

$result = mysqli_query($conn, $sql);
```

```php
require '../configDB.php';

else{
    $sql = "SELECT * FROM `tasks` WHERE `id_user` = {$id} AND `status` = 'Не виконано' AND `id_company` = -1";
    $sqlDeleted = "SELECT * FROM `tasks` WHERE `id_user` = {$id} AND `status` = 'Виконано' AND `id_company` = -1";
}

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {

    if ($not_director){
        ?>
        <li class="task-li li-title">
            <input type="checkbox" class="checkbox" name="horns" data-id="<?php echo $row['id']; ?>">
            <span class="text"><?php echo $row['title']; ?></span>
        </li>
```

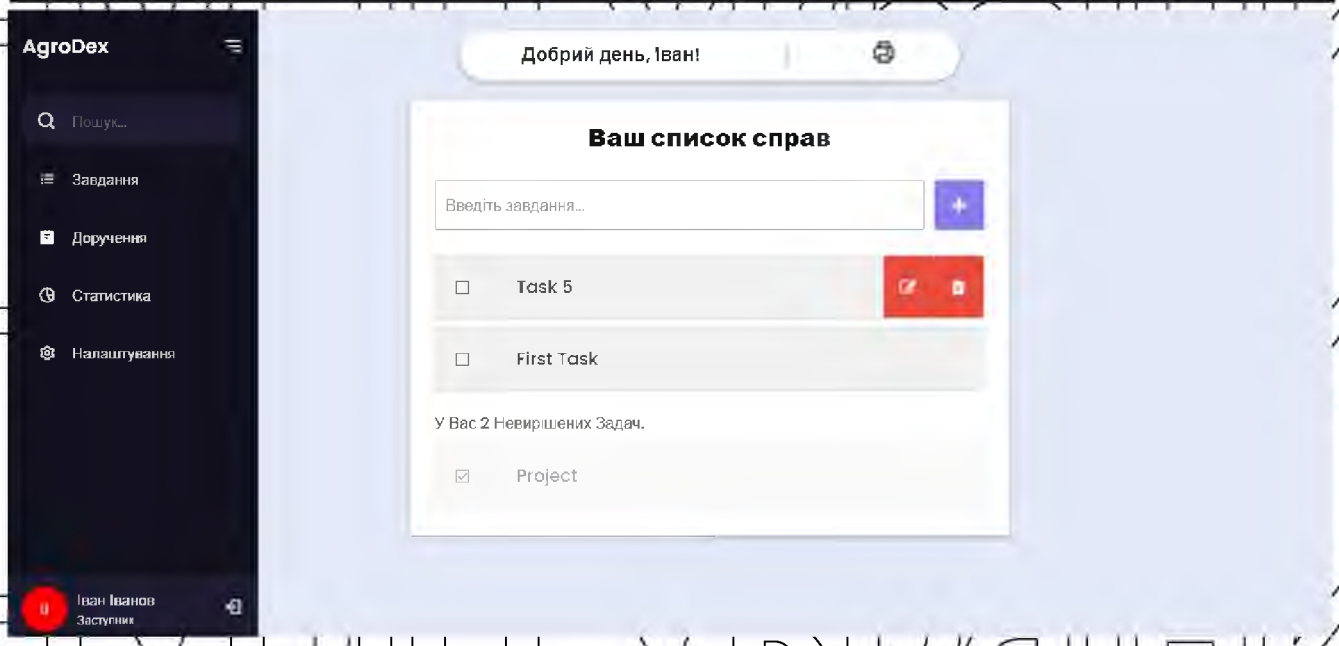


Рис. 4.1.5 Додавання задач

Для більше детального налаштування, перегляду чи видалення справ потрібно скористатися формою «edit-task» або «show-task», котра стає активною при натисканні відповідної кнопки. Зовнішній вигляд та програмна частина елементів зображенні на рисунках 4.1.6 та 4.1.7.

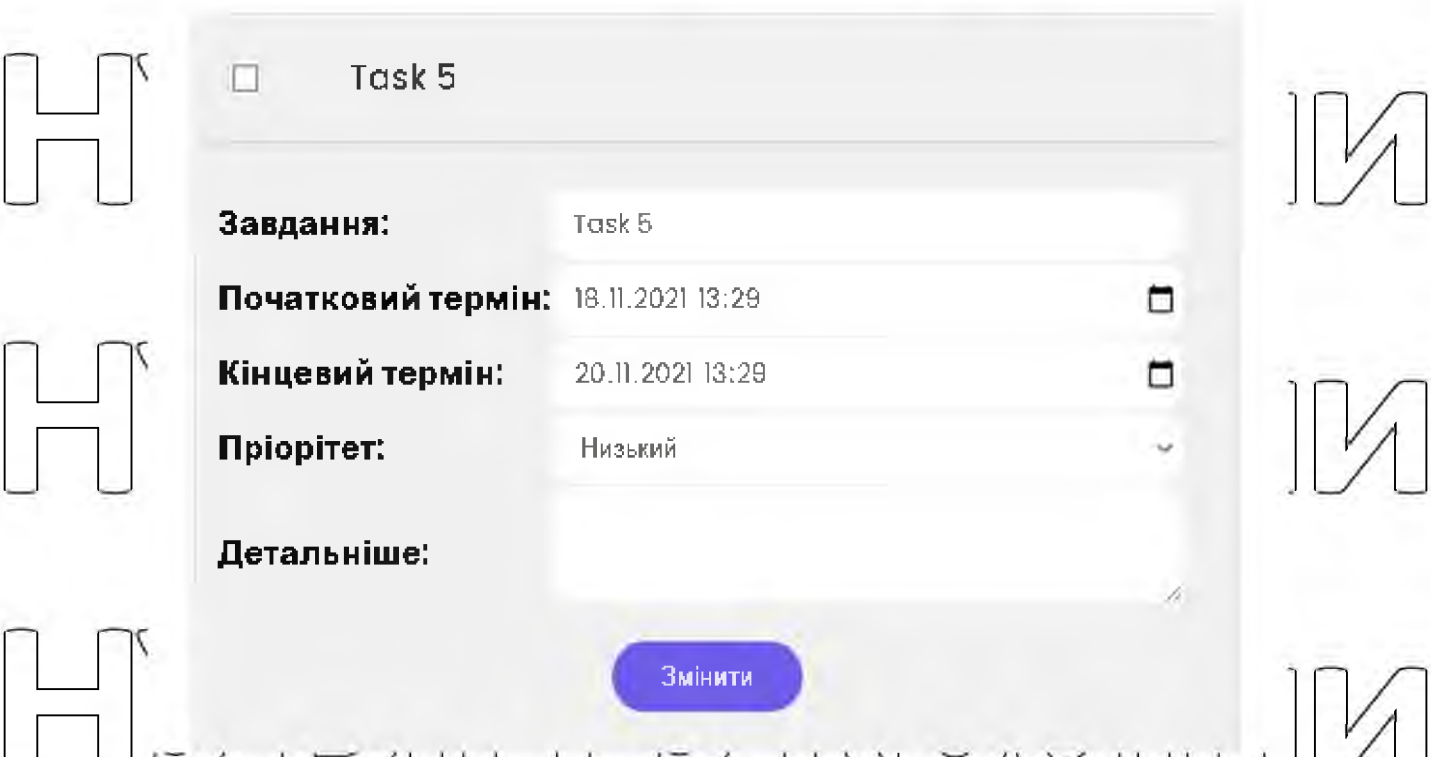


Рис. 4.1.6 Редагування задач

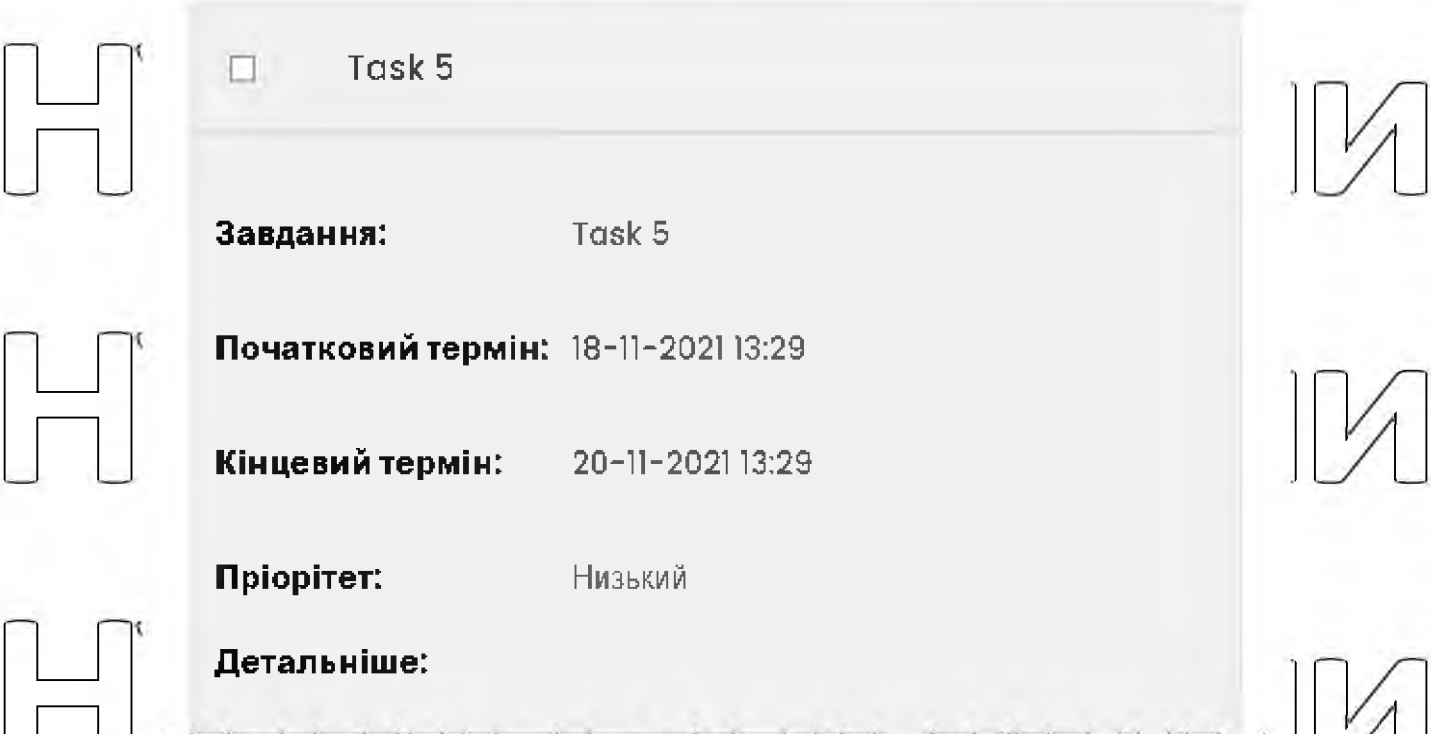


Рис. 4.1.7 Перегляд задач

Одним з елементів функціоналу є можливість друкувати або імпортувати списки в форматі PDF. Для викликання даної фуикції потрібно натиснути на кнопку виділену чорним квадратом на зображені 4.1.8. Після чого користувач зможе побачити звичну для себе форму друку (Рис. 4.1.9), в якій він може вже особисто обрати для себе - друкувати список відразу чи лише зберегти його. Дана функція має право на існування, адже в компанії, як правило, працюють люди різної вікової категорії і не у всіх можливо є з собою сучасний гаджет. Тому для працівників у яких виникають якісь труднощі з використовуванням системи – «Керівник» у любий період часу може представити їм власний список доручень вже у більш простому вигляді для них.

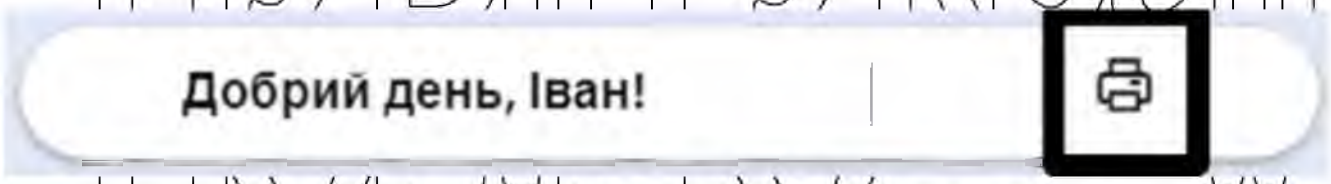


Рис. 4.1.8 Друк списку

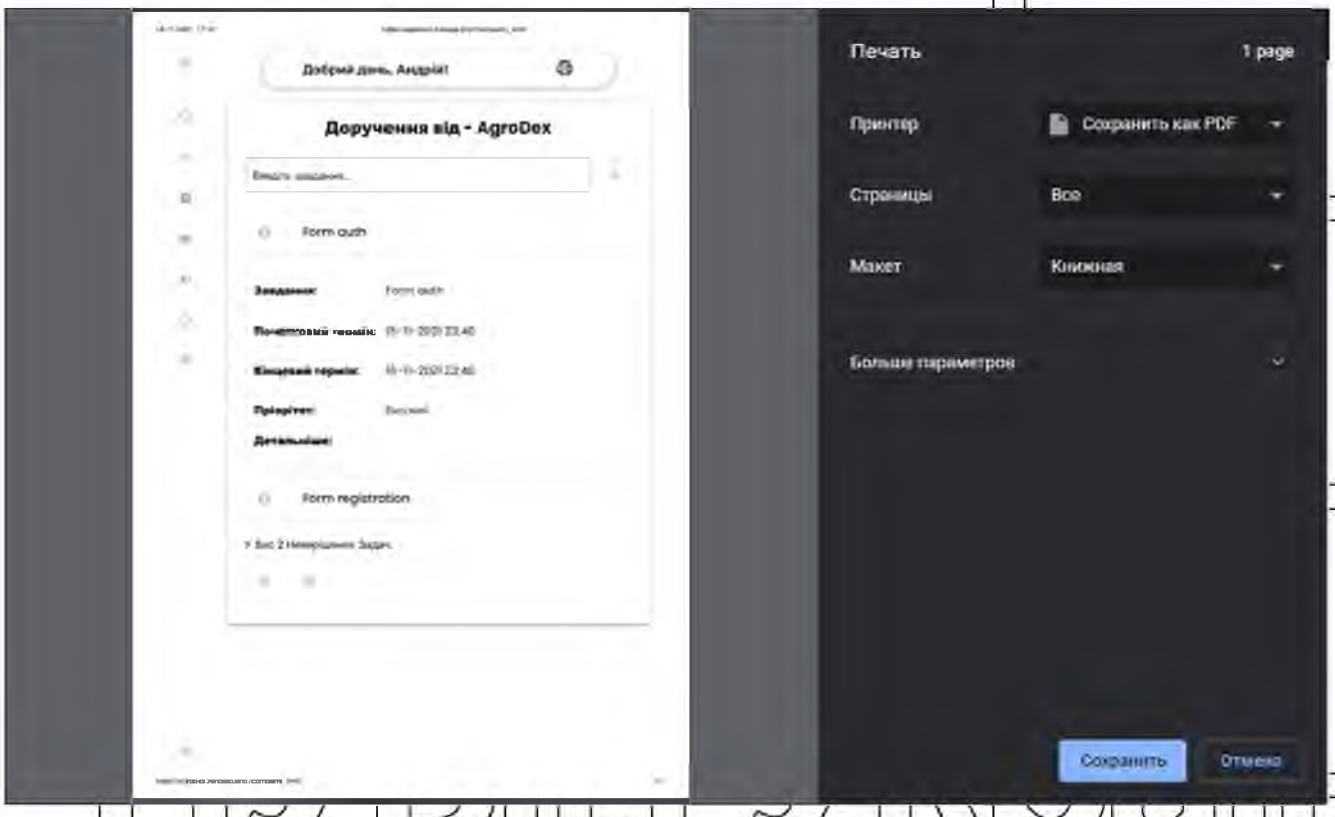


Рис. 4.1.9 Форма друку

Ваш профіль у розділі «Налаштування» - це спосіб змінити особисті дані у разі некоректного заповнення при проходженні реєстрації. Процес редагування профілю схожий до функціоналу створення нового акаунту, приклад зображено на рисунку 4.1.10.

Перелік налаштувань, котрі можна змінити:

- Ім'я
- Прізвище
- Пошта
- Пароль
- Посада
- Мобільний номер

```
require '../configDB.php';


$sql = "UPDATE `users` SET";
$sql .= "`name` = '$name', `surname` = '$surname', `email` = '$email',
`password` = '$pass', `work` = '$work', `phone` = '$phone' WHERE id = {$_POST['id']}";
$result = mysqli_query($conn, $sql);

if ($result) {
    echo 1;
} else {
    echo 0;
}
exit();
```

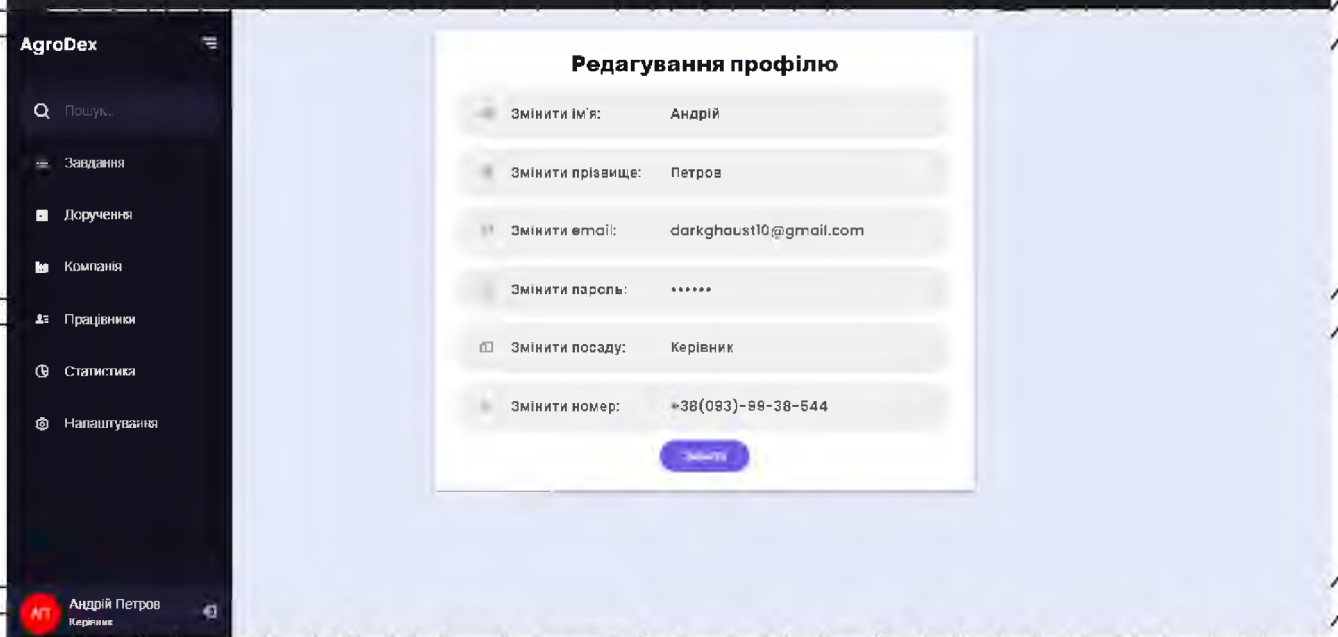


Рис. 4.1.10 Сторінка налаштування

Відповідно до того, як працює користувач – формуються результати успішності у відсотковому відношенні між виконаними та не виконаними завданнями, продемонстровано на 4.1.11.

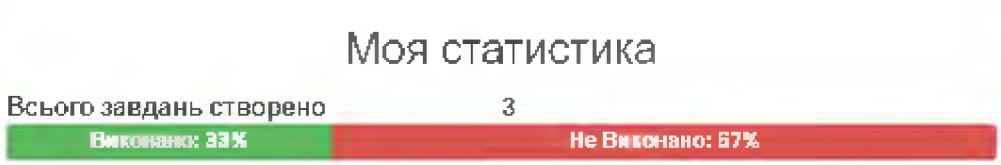


Рис. 4.1.11 Власна статистика

Доручення компанії – це список із задачами візуально дуже схожий до розділу «Завдання», відмінність полягає в тому, що він є видимим лише для користувачів з рівнем доступу «Працівник» та «Користувач». На сторінці проілюстрованій нижче можна побачити, що з'являється елемент налаштування – «Відповідальний» - це процес присвоєння керівником до створеної задачі відповідального, після чого дана дія відображається у відповідної персони на сторінці «Доручення». При цьому «Працівник» не має змоги редагувати чи видаляти присвоєне завдання.

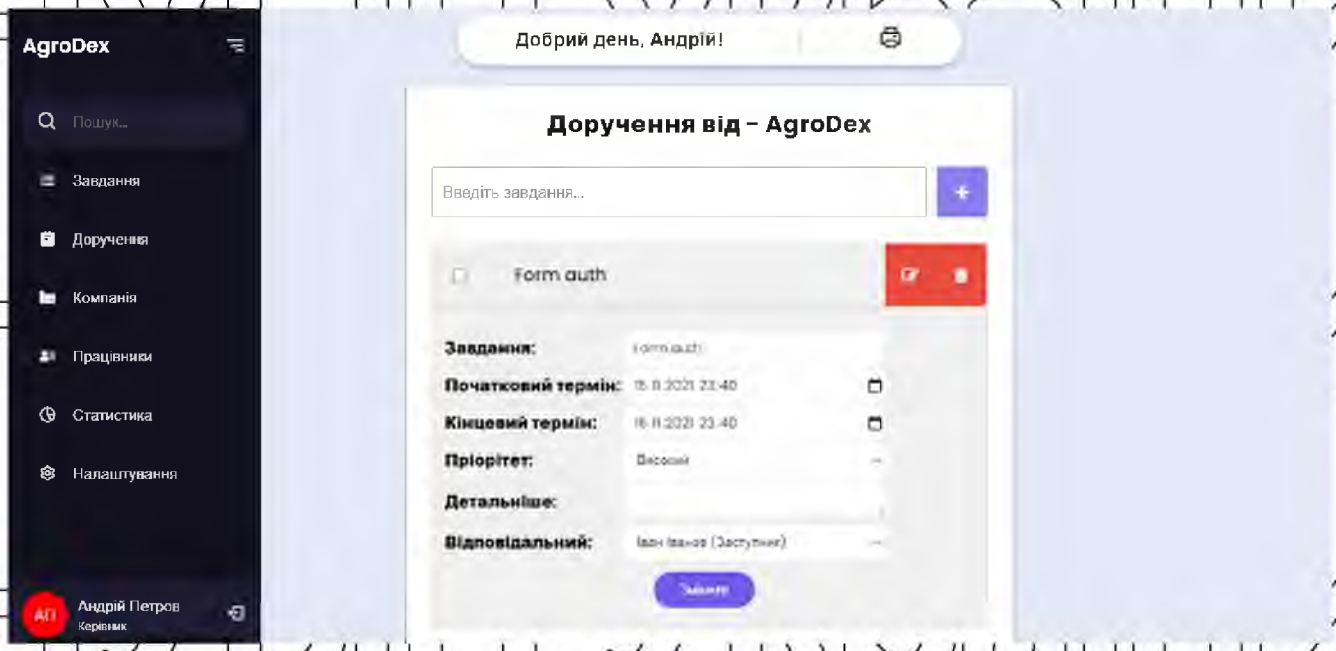


Рис. 4.1.12 Сторінка із дорученнями

Для реєстрації компанії в системі користувач має мати рівень доступу – «Керівник». При виконані попередньо озвученої вимоги у контенті головного меню з'являється розділ «Компанія» (рисунок 4.1.13), перейшовши до якого юзер отримує змогу додати свою власну організацію до системи, задавши їй такі параметри як:

- Назва організації
- Опис

Після успішної долучення підприємства до додатку керівник має змогу сформувати власну команду, запросивши працівників за допомогою листа. Електронний лист (рисунок 4.1.14) несе в собі посилання, перейшовши по якому користувачу присвоюється ідентифікатор компанії згенерований при реєстрації компанії. Дані про організацію завжди можна змінити натиснувши на відповідну кнопку.

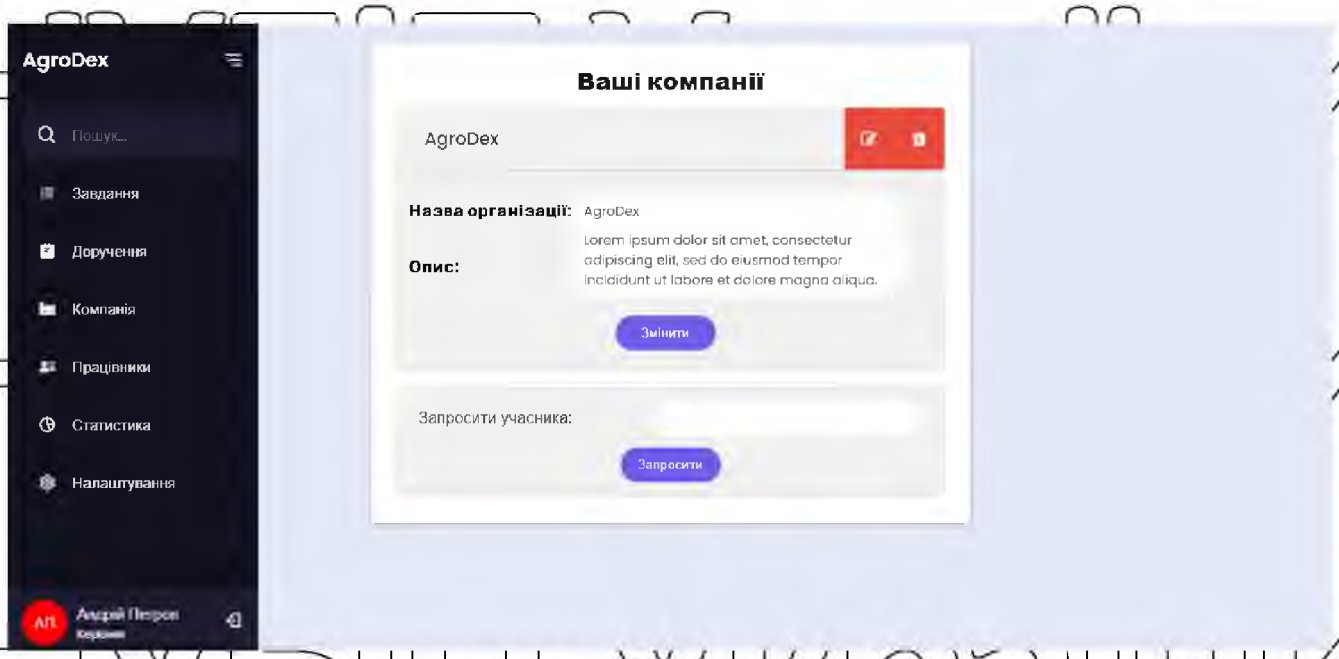


Рис. 4.1.13 Сторінка Компанія

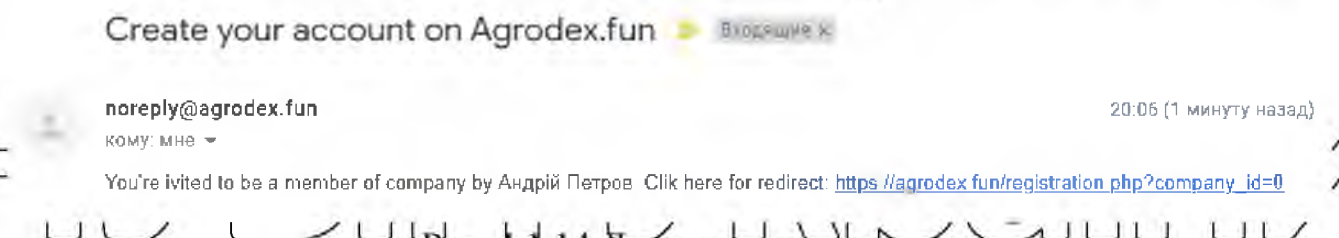


Рис. 4.1.14 Лист запрошення

Йдемо далі. Для зручності перегляду складу команди, було прийнято рішення винести інформацію про користувачів на окрему сторінку у вигляді таблиці. Розділ «Працівники» містять в собі перелік наступної інформації:

- Ім’я
- Прізвище
- Посада
- Пошта
- Мобільний телефон

Зовнішній вигляд сторінки зображено на рисунку 4.1.15.

```php
require '../configDB.php';

$id     = $_POST['id'];
$not_director = false;


if (mysqli_num_rows($result) > 0) {
    ?><h3 class="cal-stat">Члени компанії</h3><?php
    while ($row = mysqli_fetch_assoc($result)) {
        ?>
        <table class="users-table">
            <tbody>
            </tbody>
        </table>
        <?php
    }
}
else {
    echo "Членів не знайдено";
```

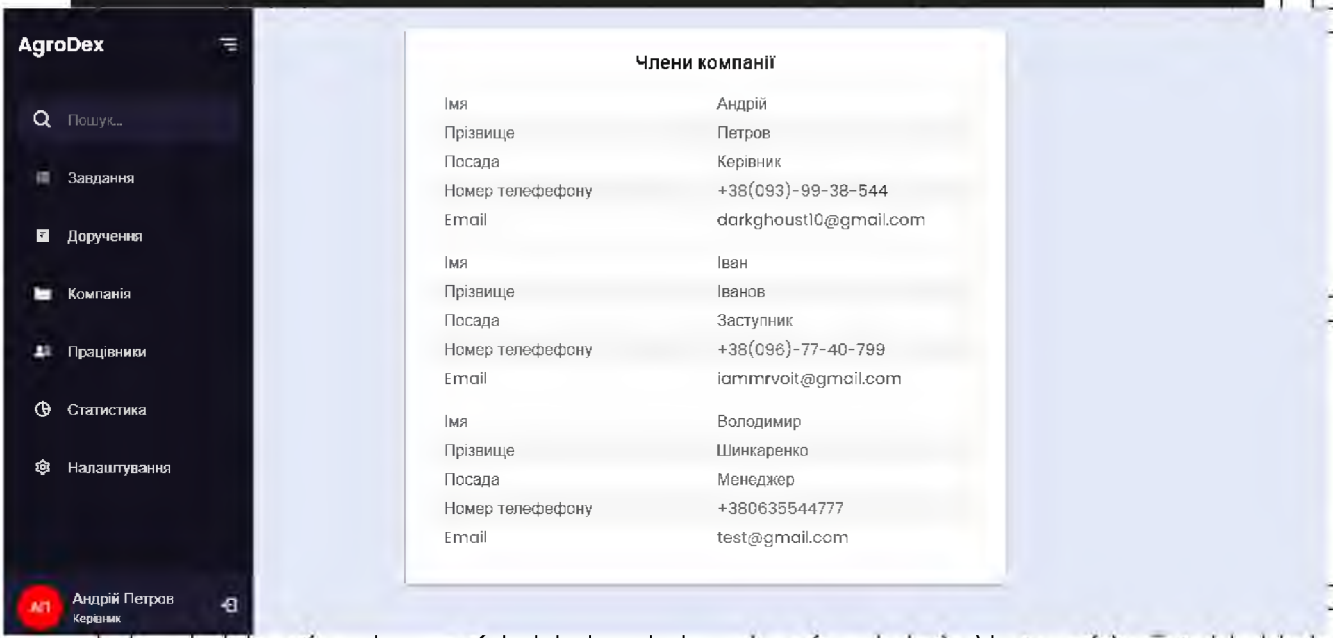


Рис. 4.1.15 Сторінка «Працівники»

Календар справ є важливим доповненням системи, адже він дозволяє візуально представляти свій план дій в часовому просторі. Для керівника дана функція відіграє важливу роль, тому що завдяки календарю він може бачити навантаження на кожного з працівників. При нерівномірному навантажені, директор може розподілити дії на інших учасників команди, що дозволить йому використовувати час ще більш ефективно. Календар справ представлено на рисунку нижче, на якому зображені задачі, яким характерні свої налаштування та відповідальні. Для пріоритетів та працівників присвоєнні власні кольори для зручності перегляду.

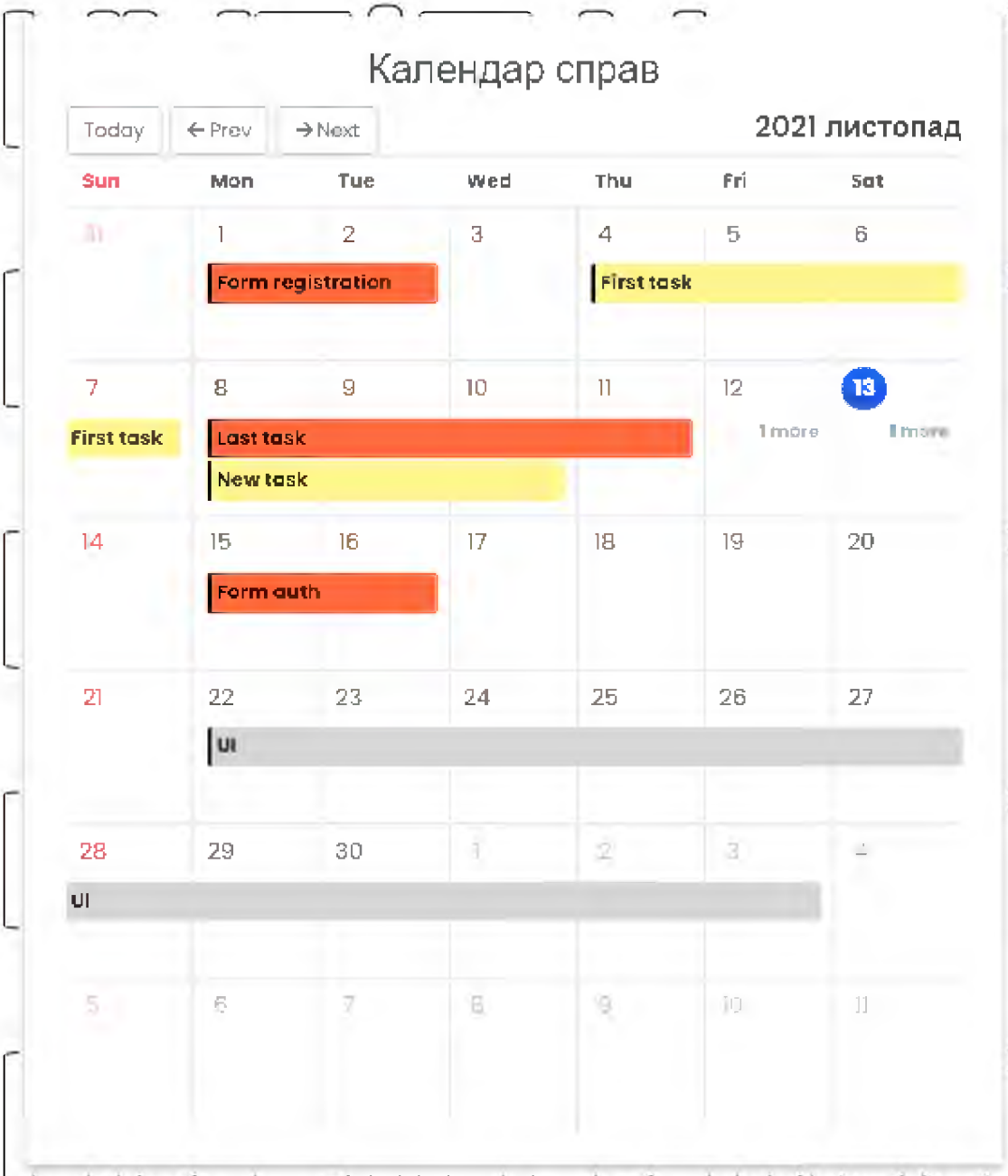


Рис. 4.1.16 Календар справ

Статистичні дані на даний момент виглядають доволі примітивно, але дозволяють переглянути успішність задач та доручень на протязі всього часу використання системи. Також нижче зображено приклад програмного коду

```php
require '../configDB.php';

$id      = $_POST['id'];
$not_director = false;

if(isset($_POST['company_id'])){
}

if (isset($_POST['company_id']) && $not_director == true){
}
elseif(isset($_POST['company_id']) && $not_director == false){
}

$sqlMy = "SELECT * FROM `tasks` WHERE `id_user` = {$id} AND `id_company` = -1";
$sqlMyDeleted = "SELECT * FROM `tasks` WHERE `id_user` = {$id} AND `status` = 'Виконано' AND `id_company` = -1";


$result = mysqli_query($conn, $sqlMy);
$resultDeleted = mysqli_query($conn, $sqlMyDeleted);

$resultArrayMy = mysqli_fetch_all($result);
$resultDeletedArrayMy = mysqli_fetch_all($resultDeleted);

if (mysqli_num_rows($result) > 0) {
    ?>
    <h3 class="cal-stat">Моя статистика</h3>
    <?php statistics($resultArrayMy, $resultDeletedArrayMy);
}
else {
    echo "Завдань не знайдено.";
}
$result = mysqli_query($conn, $sql);
$resultDeleted = mysqli_query($conn, $sqlDeleted);

$resultArrayCompany = mysqli_fetch_all($result);
$resultDeletedArrayCompany = mysqli_fetch_all($resultDeleted);

if (mysqli_num_rows($result) > 0) {
    ?>
    <h3 class="cal-stat">Статистика компанії</h3>
    <?php statistics($resultArrayCompany, $resultDeletedArrayCompany);
}
else {
    echo "Завдань компанії не знайдено.";
}
```

Рис. 4.1.17 Статистичні дані

4.2 Впровадження та експлуатація системи

Створена система планування діяльності готова до експлуатації програми в агрофірмах. Для коректної роботи додатку надано репозиторій, в якому містяться всі необхідні компоненти. Перелік форматів файлів, що належать системі:

- Файли формату CSS – формують візуальну складову веб-сторінки
- Mwb файл, котрий налічує в собі код для створення та адміністрування БД
- HTML, PHP – формують розмітку сторінки та її функціонал
- Js – відповідає за функціонал та динаміку сайту

Встановлення системи не вимагає високих вимог до апаратного забезпечення та легке в інтеграції. Для повноцінної роботи додатку достатньо лише разово розмістити увесь перелік файлів на хостингу.

При користуванні системою не виникає жодних проблем тому, що при її створенні важливим аспектом було розробити максимально зручний та простий інтерфейс користувача.

4.3 Вимоги до апаратного та програмного забезпечення

Таблиця 4.3

Вимоги до апаратних характеристик	
Параметр	**Значення**
CPU, Процесор	*AMD, Intel*
CPU clock speed, *Тактова частота процесора*	*1.0 Ghz +*
RAM, Об'єм оперативної пам'яті	*512 Mb*
Minimum channel bandwidth, *Пропускна здатність мережі*	*128 Kbit/s*
Програмні вимоги	
Operating System, **Операційна система**	**Other software,** **Інші програми**
WinXP/7/8/10/Linux	*Microsoft Internet Explorer 8.0 +, Opera 9.0 +, Google Chrome 15.0 +, Mozila Firefox 7.0+.*

НУБіП України

Висновок

Планування діяльності – це важливе завдання кожного керівництва, яка грає найважливішу роль підвищення ефективності роботи компанії. Готовий проект повинен прискорити планування і робити його доступнішим. Цей засіб надасть можливість користувачам більш ефективно використовувати трудовий час та концентруватися на важливих справах.

Під час розробки дипломного проекту було створено програмний продукт, який може цілком або частково замінити вже існуючі рішення, що використовуються на підприємстві. Система надає весь потрібний функціонал для того, щоб планування було корисним не лише керівникам, а й звичайним користувачам, оскільки функціонал передбачає використання сайту для особистих цілей.

Веб-додаток був розроблений на принципах вже існуючих рішень, в результаті розбору яких було створено цілком готову до роботи зі схожим функціоналом систему, яка складається з:

- Backend
- Frontend

Після завершення тестування програми, можна зробити висновок, що система відповідає всім поставленим задачам, котрі вона виконує, наприклад:

- Долучення нових користувачів до системи
- Реалізація входу до системи
- Формування переліку справ
- Додавання задач до списку
- Формування переліку доручень
- Надання параметрів для завдань
- Долучення компаній до системи
- Перегляд членів компанії
- Перегляд статистичних даних

НУБіП України

- Редагування облікового запису
- Коректність вводу та виводу інформації
- Безкоштовне користування, що є головною особливістю системи

До недоліків можна віднести відсутність такого функціоналу:

- Фільтр
- Примітивні статистичні дані

Роль цифрових сервісів у сільському господарстві, на жаль, недооцінюється. Але ми сподіваємось, що фахівці зможуть позитивно змінити свою думку, що підвищить роботу багатьох компаній на більш сучасний рівень.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Голуб Б.Л. Методичний посібник до вивчення дисципліни "Програмування та алгоритмічні мови". / Б.Л. Голуб, Є.М. Шукайло. – Видавничий центр НАУ, 2003. – 64 с.
2. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://pidruchniki.com/12810419646 82/ekonomika/planuvannya_divalnosti_pid priyemstva.
3. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://uk.wikiaro.ru/wiki/Open-source_software
4. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://uk.wikithe.ru/wiki/.NET_Framework
5. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://www.umj.com.ua/article/122202/avtonomizatsiya-zakladiv-ohoroni-zdorov-ya-detalnij-plan-pokrokovih-dij
6. Гафи Р.Л. Менеджмент. - СПб.: Питер, 2000. - 832с.
7. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://otherreferats.allbest.ru/bank/00733717_0.html
8. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: http://elar.nung.edu.ua/bitstream/123456789/6590/1/5749o.pdf
9. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://mozok.click/2158-stvorennya-personalnogo-navchalnogo-seredovischa-ofsn-vch-programi-dlya-stvorennya-splnih-dokumentv-opituvannya-z-vikoristannyam-cnlaynovih-form-stvorennya-saytv-konstruvuvannya-saytv-z.html
10. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://diplomba.ru/work/57389
11. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://shag.com.ua/osobististe-i-divalenistedilmi-z-vadami-movlennya-uvaga.htmluvaga.html&key=f24671caf7e807d9270ee4fc934db75f

НУБіП України

12. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://osvita.ua/vnz/reports/psychology/28404/
13. Карлоф Б. Деловая стратегия. - М. Экономика, 1995.
14. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://shpora.me/martik/bd5
15. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: http://5fan.ru/wievjob.php?id=36099
16. . Основи підприємницької діяльності та агробізнесу: [навч. посібник] / [М. М. Ільчук, Т. Д. Іщенко, В. К. Збарський та ін.]; за ред. М. М. Ільчука. – К.: Вища школа, 2002. – 398 с.
17. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: http://bukvar.su/informatika_programmirovanie/page,2,176059-Yazyki-programmirovaniya-ih-klassifikaciya-i-razvitie.html
18. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://ua-referat.com/Мови_програмування_їх_класифікація_та_розвиток
19. Роберт Дж. Мюллер. Издательство: Лорп. «Проектирование баз данных и UML». -2013. – 432 с.
20. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: http://samzan.ru/126629
21. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://dytu.ykp/blog/fotografiya/145-css-urok-1-znajomstvo.html
22. Владимир Гайдаржи, Игорь Изварин. Издательство Университет "Украина". Бази даних в інформаційних системах. – 2018. – 418с.
23. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://uk.education-wiki.com/1132418-introduction-to-css
24. Економіка підприємств. [Електронный ресурс]. – Режим доступа: http://osvita.ua/vnz/reports/econom_pidpr/19531/.
25. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://hackit-ukraine.com/627-the-advantages-and-disadvantages-of-javascript
26. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу:

НУБіП України

https://ua.referat.com/Забезпечення_безпеки_системи_і_захист_даних_в_Windows_203

27. Энтони Молинаро. Издательство: Символ-Плюс. «SQL. Сборник рецептов». – 2009. – 672с

28. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/34868/1/Babich_bakalavr.pdf

29. Г.М. Востров, Е.Е. Малахов, В.В. Мороз. Проблемы создания баз данных и информационных хранилищ.

30. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://stud.wiki/programming/2c0b65637b3ad69b5c53a894212c6c26_0.html

31. Навчальні матеріали онлайн. [Електронный ресурс]. – Режим доступу: https://uk.wikiaro.ru/wiki/Visual_Studio_Code

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