

**Національний університет кораблебудування  
імені адмірала Макарова**

Інститут комп'ютерних наук та управління проектами  
(повне найменування інституту, назва факультету )

Кафедра інформаційних управляючих систем та технологій  
(повна назва кафедри)

**Пояснювальна записка**

до дипломного проекту (роботи)

бакалавр

(освітньо-кваліфікаційний рівень)

на тему Розробка автоматизованої інформаційної системи управління  
проектами ІТ-компанії

Виконав: студент 4 курсу, групи 4142  
напряму підготовки (спеціальності)

122 «Комп'ютерні науки»

(шифр і назва напряму підготовки, спеціальності)

Михалко М.І.

(прізвище та ініціали)

Керівник Гайда А.Ю.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

м. Миколаїв – 2020 року

# Національний університет кораблебудування імені адмірала Макарова

Інститут комп'ютерних наук та управління проектами  
Кафедра інформаційних управляючих систем та технологій  
Освітньо-кваліфікаційний рівень бакалавр  
Спеціальність 122 «Комп'ютерні науки»  
(шифр і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІУСТ

К.т.н., доц. Михелєв І.Л. \_\_\_\_\_

“ ” \_\_\_\_\_ 2020 року

## **ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

Михалку Миколі Івановичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка автоматизованої інформаційної системи управління проектами ІТ-компанії.

керівник проекту (роботи) Гайда А.Ю.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “19” травня 2020 р. № 288-уч

2. Строк подання студентом проекту (роботи) 20.06.2020.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація системи, матеріали практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ. 1. Аналіз предметної області, розробка концепції автоматизованої інформаційної системи управління проектами ІТ-компанії та постановка задачі. 2. Розробка проектних рішень автоматизованої інформаційної системи управління проектами ІТ-компанії. 3. Розгортання автоматизованої інформаційної системи управління проектами ІТ-компанії. 4. Охорона праці та техніка безпеки. Висновки. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Схема організаційної структури компанії. Схема моделі кооперації. Концепція автоматизованої інформаційної системи управління проектами ІТ-компанії. Діаграма варіантів використання автоматизованої інформаційної системи. Діаграма декомпозиції кожного із варіантів використання автоматизованої інформаційної системи. Діаграма діяльності. Концептуальна модель бази даних. Логічна модель бази даних. Діаграма основних класів автоматизованої інформаційної системи. Фізична модель бази даних. Схематичне представлення інтерфейсів системи. Діаграма розгортання автоматизованої інформаційної системи. Реалізовані графічні інтерфейси.

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Гайда А.Ю.		
2	Гайда А.Ю.		
3	Гайда А.Ю.		
4	Гайда А.Ю.		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області, розробка концепції автоматизованої інформаційної системи та постановка задачі	25.03.2020	
1.1	Аналіз предметної області	20.01.2020	
1.2	Формування вимог	10.02.2020	
1.3	Аналіз існуючих рішень	15.02.2020	
1.4	Розробка концепції	20.03.2020	
1.5	Постановка задачі	25.03.2020	
2	Розробка проектних рішень автоматизованої інформаційної системи	17.05.2020	
2.1	Загальносистемні рішення	30.03.2020	
2.2	Рішення з інформаційного забезпечення	15.04.2020	
2.3	Рішення з програмного забезпечення	20.04.2020	
2.4	Розробка інтерфейсів	30.04.2020	
2.5	Рішення з технічного забезпечення	17.05.2020	
3	Розгортання автоматизованої інформаційної системи	01.06.2020	
3.1	Розробка документації системи	25.05.2020	
3.2	Розгортання системи	01.06.2020	
4	Охорона праці та техніка безпеки	09.06.2020	
4.1	Вступ	02.06.2020	
4.2	Напруженість праці користувача ЕОМ	08.06.2020	
4.3	Ергономіка робочого місця	09.06.2020	

Студент

\_\_\_\_\_ ( підпис )

Керівник проекту (роботи)

\_\_\_\_\_ ( підпис )

Михалко М.І.  
(прізвище та ініціали)

Гайда А.Ю.  
(прізвище та ініціали)

## Анотація

Темою роботи є розробка автоматизованої інформаційної системи управління проектами ІТ-компанії. Система спрямована на підвищення ефективності діяльності компанії в частині управління проектами.

В роботі виконано аналіз діяльності об'єкта та його організаційної структури, визначенні недоліки системи управління проектами компанії, сформовані вимоги до автоматизованої інформаційної системи управління проектами та проведено аналіз існуючих рішень. На основі вимог до інформаційної системи розроблено концепцію та проектні рішення автоматизованої інформаційної системи. Розробка проектних рішень складається з загальносистемних рішень, рішень з інформаційного, програмного та технічного забезпечення.

В результаті розробки було створено автоматизовану інформаційну систему управління проектами ІТ-компанії, що має веб-інтерфейс. Програмне забезпечення системи виконане мовами програмування PHP та JS з використанням бібліотек: Laravel Framework 6.2, Laravel jwt-auth, Axios, Vue.js, Vuex, Vue-resources, Vue-router.

Робота викладена на 111 сторінках друкованого тексту, містить 21 рисунок, 3 таблиці та список використаної літератури з 27 найменувань.

					122.4142.08.ПЗ.Анотація	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		4

## Annotation

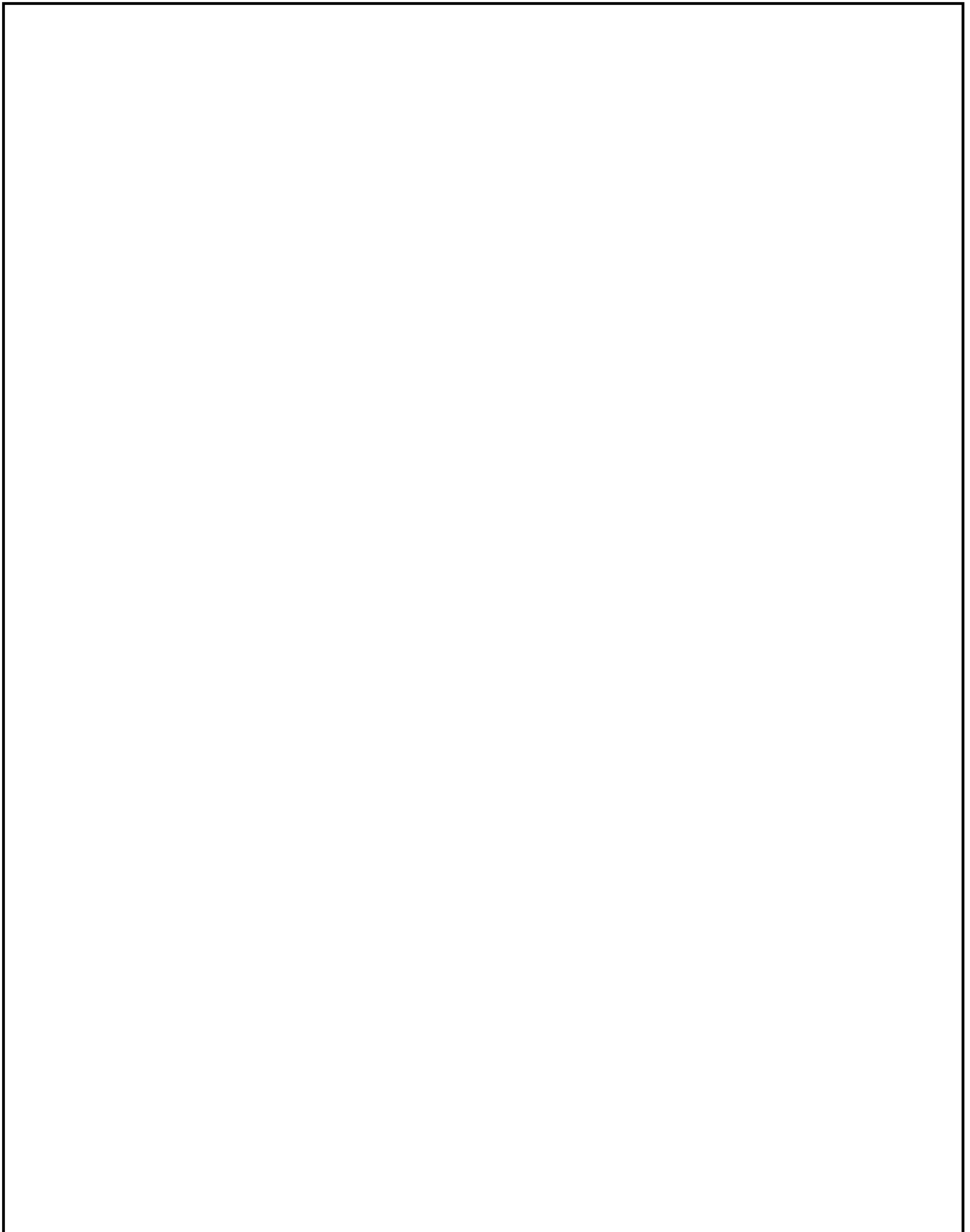
The theme of the work is the development of an automated information system for project management of an IT company. The system is aimed at improving the efficiency of the company in terms of project management.

The analysis of the object and its organizational structure, identification of the shortcomings of the project management system of the company, the requirements for the automated information system of project management and analysis of existing solutions. Based on the requirements for the information system, the concept and design solutions of the automated information system have been developed. Development of design solutions consists of system-wide solutions, solutions for information, software and hardware.

As a result of the development, an automated project management information system of an IT company with a web interface was created. The system software is made in PHP and JS programming languages using libraries: Laravel Framework 6.2, Laravel jwt-auth, Axios, Vue.js, Vuex, Vue-resources, Vue-router.

The work is spread over 111 pages of printed text, contains 21 pictures, 3 tables and a list of used literature from 27 items.

					122.4142.08.ПЗ.Анотація	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>5</b>



					<b>122.4142.08.ПЗ</b>			
Зм.	Аркаш	№ документа	Підпис	Дата				
					<b>Розробка автоматизованої інформаційної системи управління проектами ІТ- компанії</b>	Літ.	Аркуш	Аркушів
Студент	Михалко М.І.						6	106
Керівник	Гайда А.Ю.					<b>НУК</b>		
Зав. каф.	Михелєв І.Л.							

## Зміст

Вступ.....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ ТА ПОСТАНОВКА ЗАДАЧІ.....	14
1.1. Аналіз предметної області.....	14
1.1.1. Опис об'єкта автоматизації та його організаційної структури .....	14
1.1.2. Визначення недоліків системи управління проектами ІТ-компанії..	19
1.2. Формування вимог до автоматизованої інформаційної системи .....	21
1.2.1. Вимоги до переліку задач.....	21
1.2.2. Вимоги до вхідної та вихідної інформації системи .....	24
1.3. Аналіз існуючих рішень .....	26
1.3.1. Автоматизовані інформаційні системи.....	26
1.3.2. Додаток KPlato .....	26
1.3.3. Додаток OpenProj.....	27
1.3.4. Додаток GanttProject .....	27
1.3.5. Система управління проектами Basecamp.....	28
1.3.6. Веб-додаток ActiveCollab .....	28
1.3.7. Порівняльний аналіз .....	29
1.4. Розробка концепції автоматизованої інформаційної системи .....	31
1.5. Постановка задачі .....	35
1.6. Висновки.....	42
2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ	44
2.1. Загальносистемні рішення автоматизованої інформаційної системи .....	44
2.1.1. Діаграми варіантів використання.....	44
2.1.2. Діаграма діяльності.....	51
2.2. Рішення з інформаційного забезпечення автоматизованої інформаційної системи.....	54
2.2.1. Концептуальна та логічна моделі бази даних .....	54
2.2.2. Аналіз нормалізації сутностей бази даних .....	57

2.3. Рішення з програмного забезпечення автоматизованої інформаційної системи.....	58
2.3.1. Вибір інструментальних засобів .....	58
2.3.2. Структура програмного забезпечення .....	63
2.3.3. Фізична модель бази даних .....	64
2.3.4. Методи і засоби розробки програмного забезпечення .....	65
2.4. Розробка інтерфейсів автоматизованої інформаційної системи .....	67
2.5. Рішення з технічного забезпечення автоматизованої інформаційної системи.....	70
2.6. Висновки.....	71
<b>3. РОЗГОРТАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ.....</b>	<b>73</b>
3.1. Розробка документації системи .....	73
3.2. Розгортання системи.....	73
<b>4. ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ.....</b>	<b>76</b>
4.1. Вступ.....	76
4.2. Напруженість праці користувача ЕОМ.....	77
4.2.1. Рівень штучного освітлення .....	77
4.2.2. Мікроклімат робочого місця .....	78
4.2.3. Шум та вібрації у робочому приміщенні.....	78
4.2.4. Оцінка електробезпеки .....	79
4.3. Ергономіка робочого місця .....	80
Висновки.....	84
Список використаної літератури .....	85
Додаток А - Технічне завдання .....	89
Додаток Б – Загальний опис системи.....	93
Додаток В – Інструкція з використання системи .....	94
Додаток Г – Структура програмного забезпечення .....	103
Додаток Д – Тексти програмних модулів системи.....	104



## Вступ

Широкомасштабне використання в усіх сферах діяльності інформаційних технологій, що являють собою складний комплекс інформаційно-телекомунікаційних систем, потребує впорядкування та оптимізації процесів їх розбудови та підтримки.

ІТ-проект – сукупність процесів створення нового або модифікації наявного ІТ-продукту, що може передбачати такі види робіт: аналіз вимог замовника, формування технічного завдання, проектування та реалізація роботи.

Для забезпечення ефективного виконання ІТ-проектів визначено такі категорії ІТ-проектів:

I категорія – проекти, скеровані на створення нових ІТ-продуктів згідно з вимогами замовника.

II категорія – проекти, спрямовані на модифікацію ІТ-продуктів згідно з вимогами замовника.

III категорія – проекти, спрямовані на модифікацію наявних ІТ-продуктів для виправлення виявлених дефектів реалізації або оптимізацію характеристик функціонування.

Управління проектом – це процес керівництва та координації людських, матеріальних та фінансових ресурсів протягом життєвого циклу проекту шляхом застосування сучасних методів та техніки управління для досягнення визначених у проекті результатів за складом та обсягом робіт; вартістю, часом, якістю та задоволенню інтересів учасників проекту [1, 2].

Ефективність діяльності компанії, а саме реалізація ІТ-проектів, багато в чому залежить від системи, яка використовується для управління розробкою ІТ-проектів. Мати сучасну та багатофункціональну систему управління проектами є необхідним критерієм для проведення ефективної діяльності ІТ-компанії.

					122.4142.08.ПЗ.Вступ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		9

До основних процесів управління здійснення ІТ-проектів входять:

- Планування обсягу робіт, необхідних для виконання проекту і досягнення його цілей.
- Організацію ресурсів для їх виконання в межах встановленого бюджету і строків.
- Впровадження розробленої програми дій.
- Контроль за виконанням плану або його коригування у разі необхідності.

Управління проектом на основі його життєвого циклу проходить в 4 основних фази [3]:

- Вибір проекту.
- Планування.
- Реалізація і контроль.
- Завершення.

Проекти виникають як наслідок потреб. Увесь процес управління проектом починається тоді, коли існує потреба, яку потрібно задовольнити. Рішення вибору проекту приймаються виходячи з наявності доступних ресурсів і кількості потреб, що їх треба задовольнити, а також беручи до уваги величину витрат на задоволення цих потреб і порівняльну важливість задоволення одних потреб та ігнорування інших.

Планування здійснюється протягом усієї тривалості проекту. Як тільки проект було підтримано, починається формальне і детальне планування. Визначаються віхи проекту, формулюються завдання та їхня взаємозалежність. По ходу здійснення проекту його план може піддаватися постійному коригуванню з урахуванням виникнення непередбачених обставин і реагування на них.

Коли складено формальний план, можна розпочинати виконання проекту. При визначеному змісті здійснення це серцевина проекту, оскільки, відповідно до сформульованого плану, спрямовуються зусилля на виконання робіт для задоволення потреб кінцевих користувачів.

					122.4142.08.ПЗ.Вступ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		10

Під час здійснення проекту його керівники постійно контролюють хід виконання. Вони зважають на те, що вже зроблено за проектом, оцінюють ступінь реалізації плану і визначають, чи немає великих розбіжностей між запланованим і фактичним виконанням. В управлінні проектами такі розбіжності називаються відхиленнями.

В основі процесу контролю лежить збір і розгляд даних про просування проекту. За наявності такої інформації керівники проектів мають приймати рішення про подальші дії і заходи. Наприклад, якщо відставання від графіка виходить за допустимі межі, менеджер може вирішити прискорити виконання певної кількості критичних робіт, виділивши для цього більший обсяг ресурсів. Або якщо виявляється, що на певну групу робіт їх виконавці витратили на 40 відсотків менше ресурсів, аніж було заплановано, це відхилення, без сумніву, буде проаналізоване, оскільки нерідко економія коштів означає, що окремі роботи взагалі не виконуються або ж у них «зрізуються кути».

Важливим моментом в реалізації проектів є інформаційне забезпечення операційної діяльності. Тому необхідно мати можливість збереження даних необхідних для реалізації в системі управління для того, щоб завжди був доступ до цієї інформації.

Як правило інформація по проекту є важлива і її втрата або заволодіння третіми особами несе за собою негативні наслідки. Тому система повинна забезпечувати захист інформації та здійснювати створення резервних копій для відновлення даних у разі їх пошкодження або повної втрати.

Розробка плану, контроль за ходом виконання проекту, аналіз завершених робіт є трудомісткими процесами на які витрачається багато часу. Тому система управління проектами повинна мати автоматизований збір та формування даних для того, щоб полегшити та пришвидшити процес управління. Також зручний доступ до інформаційного забезпечення операційної діяльності та надійність є обов'язковими моментами в системі управління проектами. Приймаючи до уваги всі перераховані потреби, можна зробити висновок, що

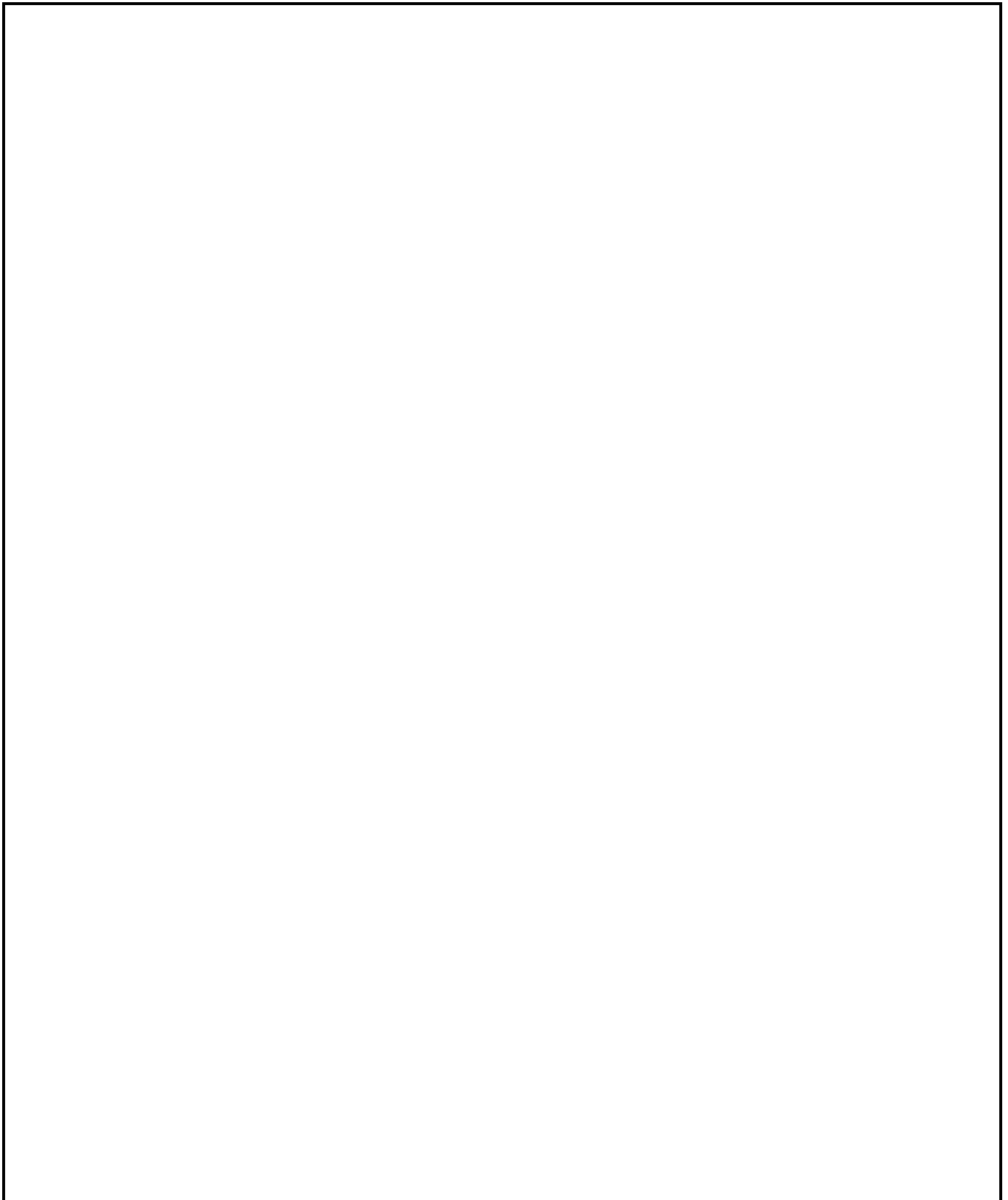
створення автоматизованої інформаційної системи управління проектами ІТ-компанії є актуальною темою.

Метою роботи є підвищення ефективності процесу управління проектами ІТ-компанія, яку можна досягти завдяки автоматизованій системі управління проектами.

Головною задачею роботи є розробка інформаційної системи, яка буде відповідати всім сучасним вимогам багатofункціональної системи управління проектами ІТ-компанії. Для того, щоб її виконати необхідно провести роботи, які включають в себе:

- Аналіз діяльності об'єкта та його організаційної структури.
- Визначення недоліків системи управління проектами ІТ-компанії.
- Аналіз можливих шляхів усунення знайдених проблем в системі управління проектами, приймаючи до уваги організаційну структуру компанії.
- Формулювання вимог, яким повинна відповідати автоматизована інформаційна система.
- Аналіз існуючих рішень.
- Вибір підходящих засобів, за допомогою яких буде досягнуто поставлених вимог до інформаційної системи.
- Розробка проектних рішень автоматизованої інформаційної системи управління проектами.
- Реалізація автоматизованої інформаційної системи.
- Впровадження автоматизованої інформаційної системи управління проектами в ІТ-компанії.

					122.4142.08.ПЗ.Вступ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		12



					122.4142.08.ПЗ.Р1			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Аналіз предметної області, розробка концепції автоматизованої інформаційної системи управління проектами ІТ-компанії та постановка задачі	Літ.	Аркуш	Аркушів
Студент	Михалко М.І.						13	30
Керівник	Гайда А.Ю.					НУК		
Зав. каф.	Михелєв І.Л.							

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1. Аналіз предметної області

### 1.1.1. Опис об'єкта автоматизації та його організаційної структури

На сьогоднішній день існує велика кількість ІТ-компаній, які надають різноманітні послуги, починаючи з підтримки користувачів та закінчуючи розробками високоефективних та масштабних систем. Всі ці компанії, в процесі розробки проектів, використовують інформаційні системи для управління розробки проектами. Для того, щоб забезпечити конкурентноспроможність компанії ця система повинна бути сучасною та багатофункціональною.

Було виявлено, що компанія “FutureDev” не може конкурувати з багатьма іншими ІТ-компаніями через застарілу та малофункціональну систему управління проектами. Для того, щоб підвищити шанси цієї компанії та піднятися вище на ринку ІТ-постуг необхідно виявити існуючі проблеми інформаційної системи та усунути їх.

ІТ-компанія “FutureDev” - це невелика компанія, яка займається розробкою різноманітних продуктів в інформаційній сфері, таких як: веб-сайти різного призначення(багатокористувацькі веб-додатки, шаблони сайтів та ін.) та додатки для смартфонів на платформах – Android та IOS. Компанія складається з 5 структурних підрозділів в яких працює 30 працівників. Компанія має наступні структурні підрозділи:

- 1) Директор.
- 2) Бухгалтерський підрозділ у складі 2 осіб.
- 3) Підрозділ веб-програмування в якому працює менеджер проектів, 9 програмістів, 2 тестувальника та 1 дизайнера.
- 4) Підрозділ мобільних розробок в якому працює менеджер проектів, 7 програмістів, 2 тестувальника та 1 дизайнера.

5) Обслуговуючий персонал, який включає системного адміністратора та осіб обслуговуючих офіс компанії.

Основна частина співробітників працює в офісі, але декілька осіб працюють віддалено.

“FutureDev” має лінійну організаційну структуру, в якій кожний підлеглий підпорядкований тільки одному керівнику і в кожному підрозділі виконується весь комплекс робіт. Системний адміністратор підпорядковується менеджерам проектів обох підрозділів: веб-програмування та мобільних розробок.

На основі опису структурних підрозділів з метою графічного представлення організаційної структури компанії та опису діяльності кожного з підрозділів розробимо схему організаційної структури компанії “FutureDev”.  
Схема представлена на рисунку 1.1.

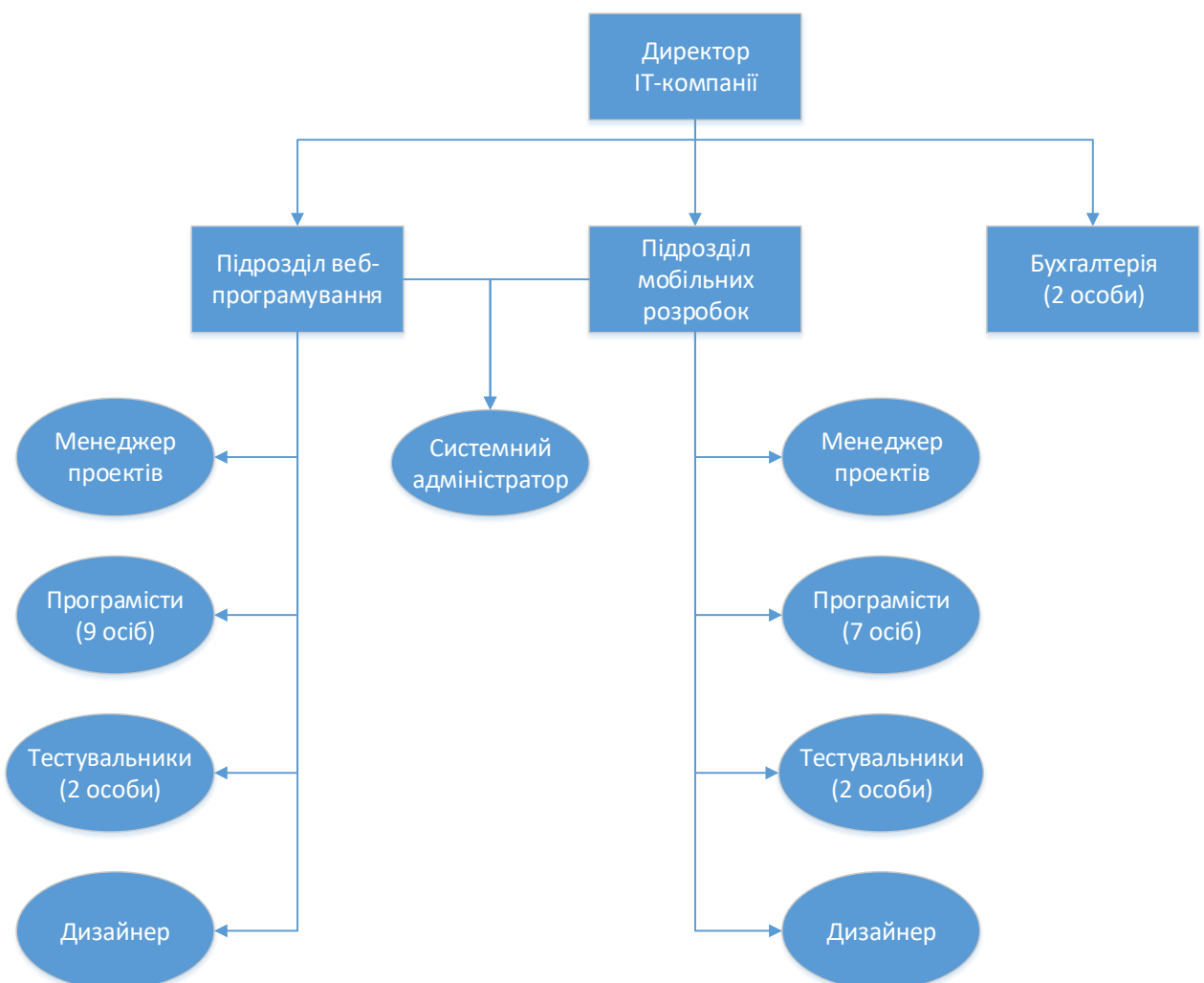


Рисунок 1.1 – Схема організаційної структури компанії “FutureDev”

На основі схеми організаційної структури з метою виявлення проблем в системі управління проектами детально розглянемо роботу кожного підрозділу. Функція директора полягає в загальному керуванні компанією, він несе повну відповідальність за всі прийняті рішення, організовує роботу з цілю здійснення ефективної взаємодії всіх підрозділів компанії.

Другим важливим підрозділом у компанії “FutureDev” є бухгалтерський, що складається з 2 осіб, які займаються розрахунками витрат по проектам, персоналу та складають звіти. Всі ці роботи бухгалтери виконують в ручному режимі, адже для управління проектами компанія використовує малофункціональну систему, яка не володіє функціоналом автоматичного розрахунку витрат та формуванням звітів. Через це один бухгалтер не зміг би справитися з такою кількістю робіт.

Оскільки компанія займається двома різними видами розробок: веб та мобільні додатки, то існує два підрозділи в яких і проводяться ці розробки. В кожному із підрозділів присутній менеджер проектів. Вони проводять свою діяльність незалежну один від одного, адже при роботі менеджери повинні бути компетентні в питанні продукту, який розробляють та встигати реалізовувати проекти вчасно.

Перед початком реалізації проекту менеджер створює попередній план – перше уявлення, що включає в себе проект та обговорює всі деталі з замовником. Далі менеджер визначає приблизні терміни виконання та виконує приблизний розрахунок вартості проекту. До цього процесу залучаються розробники та бухгалтери, які допоможуть з меншим відхиленням визначити терміни та вартість, адже не завжди легко оцінити на скільки складною є система, яку необхідно реалізувати. Після того, як було проаналізовано необхідні деталі, визначено головну ціль проекту та узгоджено з директором компанії, то з замовником зіставляється договір в якому описуються важливі деталі, які забезпечують надійність та ясність обом сторонам договору. Після того, як було підписано всі необхідні документи менеджер складає генеральний

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		16



план виконання робіт, за яким буде вестися розробка. Цей план затверджується директором.

Для управління проектами “FutureDev” використовує додаток в якому створюється проект. Тому на цьому етапі менеджер реєструє проект в додатку: вказує його назву, описання для того, щоб програмісти, тестувальники та дизайнери зрозуміли над чим вони будуть працювати та підключає необхідних співробітників до роботи над проектом. Далі складається номенклатурний план та проводиться розподіл робіт між програмістами та дизайнером. Всі роботи задаються в додатку у вигляді завдань, які містять назву, опис та призначаються окремому співробітнику. Після того, як менеджер створює завдання на робочій сторінці співробітника воно з’являються. Але виконавцю необхідно постійно стежити за завданнями, які були призначені для нього, адже функціонал додатку не має автоматичного сповіщення, яке б звільнило співробітників постійно моніторити нові завдання. Такий недолік додатку призводить до затримки виконання робіт, що в свою чергу негативно відображається на швидкості роботи та призводить до більших витрат.

Також завданням, які повинні бути виконані в певний проміжок часу призначаються терміни виконання. Оскільки веб-сайт не вміє автоматично відстежувати назначені завданню строки здачі, то менеджеру проекту доводиться постійно переглядати завдання всіх розробників та тестувальників, щоб у випадку перевищення часу виділеного на вирішення завдання прийняти рішення зі зміни термінів або ж підключення до виконання завдання інших співробітників, щоб воно було виконано якомога швидше.

Після проведення всіх вище описаних робіт співробітники починають реалізовувати поставлені перед ними задачі. Коли виконавець закінчує виконання конкретного завдання він документує у вигляді коментаря підтвердження, що завдання було виконано, описує складнощі, які виникли в процесі роботи та вказує скільки часу було витрачено для того, щоб закінчити роботу. Час вказується в ручному режимі, що є не зручно. Адже необхідно

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		17

постійно пам'ятати коли було розпочату роботу над завданням. Як наслідок зазначений час може бути не коректним.

Коли завдання було виконано та збережено витрачений час, воно переадресовується тестувальнику. І знову ж таки, тестувальник не отримує повідомлення про те, що йому необхідно перевірити завдання. Тому він має постійно перевіряти наявність призначених йому завдань. Тестувальник перевіряє зроблену роботу та назначає статус: виконано або не виконано. У першому випадку завдання помічається, як виконане та автоматично знімається зі співробітника, а в другому знову відправляється на розробника для доопрацювання.

Коли всі завдання проекту були вирішені, що означає завершення проекту, то відбувається закриття проекту. Для цього менеджер переглядає в додатку чи не залишилось жодного завдання та помічає проект, як завершений. Бухгалтер у свою чергу на основі звітів, які були сформовані в ручному режимі розраховує витрати по проекту. Після чого менеджер переглядає звіти бухгалтера, аналізує та приймає рішення, якщо необхідно, по складу команди для наступного проекту.

Також в компанії працює системний адміністратор, який підпорядковується обом менеджерам проектів. Він відповідає за роботу всіх комп'ютерних систем. У випадку якихось збоїв йому необхідно якомога швидше вирішувати проблему для того, щоб всі співробітники могли виконувати свою роботу.

Згідно з структурою компанії та процесом розробки проектів для візуального представлення взаємин між особами, які відбуваються в ІТ-компанії "FutureDev" під час розробки проектів використовуємо модель кооперації, яка представлена у вигляді схеми на рисунку 1.2.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		18

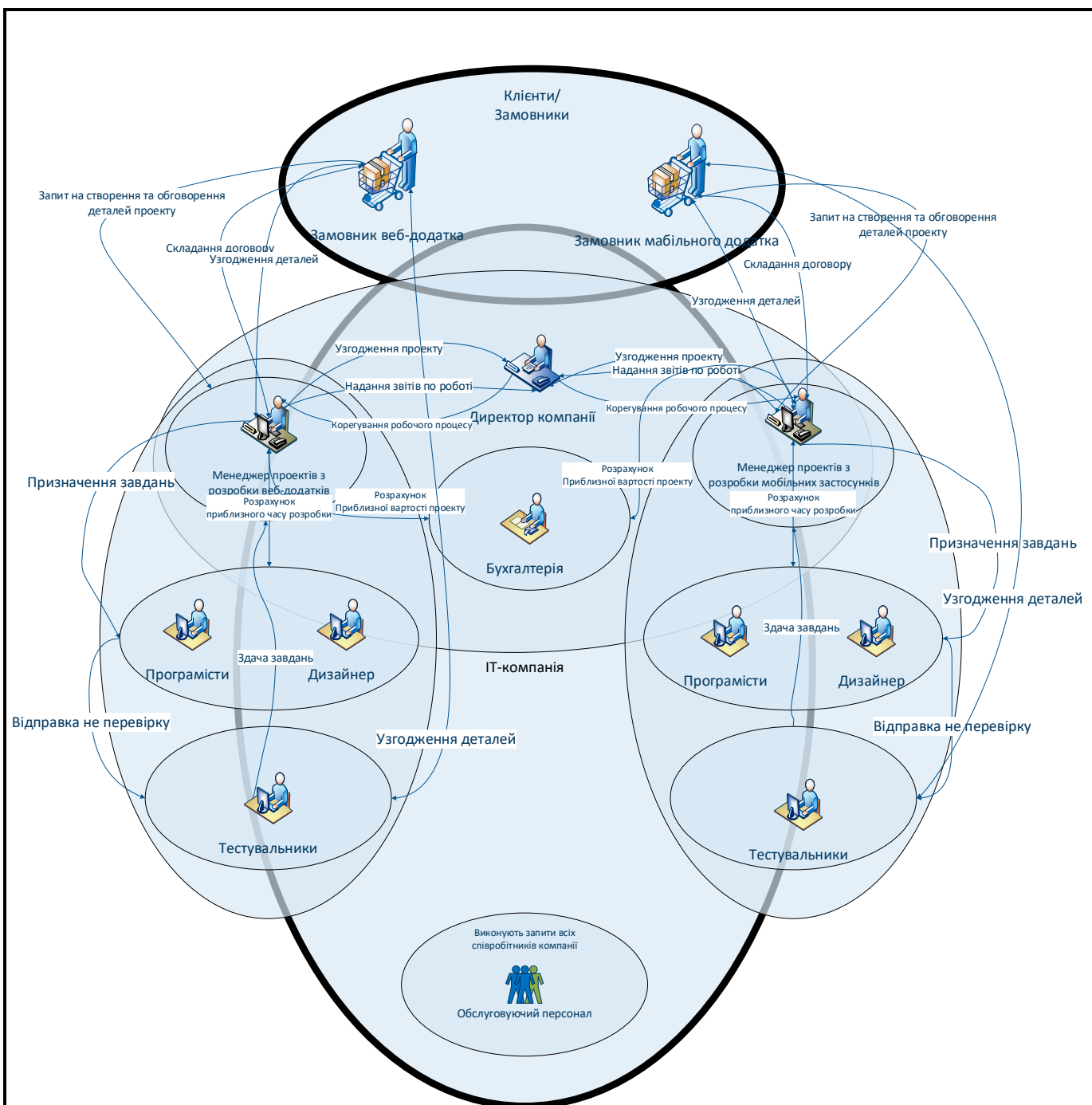


Рисунок 1.2 – Схема моделі кооперації

### 1.1.2. Визначення недоліків системи управління проєктами ІТ-компанії

На основі опису об'єктам автоматизації та його організаційної структури з метою формування вимог до автоматизованої інформаційної системи управління проєктами ІТ-компанії визначимо недоліки в існуючій системі управління проєктами компанії "FutureDev".

Зм.	Аркуш	№ документа	Підпис	Дата

Виявлено, що система управління проектами компанії “FutureDev” є малофункціональною та сповільнює розробку проектів. Вона має велику кількість недоліків, через що, всім учасникам проекту доводиться витратити багато часу на рутинну роботу. Ця система відіграє важливу роль, адже завдяки їй відбувається весь процес управління проектами у компанії. З представленої на рисунку 1.2 схеми моделі кооперацій видно, що створення проектів це складний процес, який потребують уваги та витрат часу кожного співробітника. Тому, використання малофункціональної системи управління проектами є недопустимим, якщо компанія бажає бути конкурентоспроможною.

В системі управління проектами компанії “FutureDev” виявлено наступні недоліки:

- 1) Відсутня можливість переглядати завантаженість співробітників, тому менеджер витрачає багато часу на розподілення навантаження серед всіх учасників проекту.
- 2) Менеджеру проектів постійно необхідно переглядати всі невиконані завдання та в ручному режимі повідомляти співробітників про виявлені порушення термінів виконання робіт.
- 3) Відсутня можливість зазначення приблизного часу виконання завдання, тому стає важчим оцінити продуктивність співробітників.
- 4) Відсутня можливість автоматичного формування звітів необхідних для розрахунку витрат на розробку проектів та показників продуктивності команди розробників.
- 5) Відсутній автоматичний облік часу, тому співробітникам потрібно контролювати час початку роботи над завдання та час завершення.
- 6) Відсутній функціонал сповіщення про нові завдання та замітки в проектах, тому співробітники не завжди вчасно дізнаються про важливу інформацію.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		20

7) Відсутнє автоматичне створення резервних копій, тому в разі пошкодження або втрати даних буде втрачено інформацію по розробкам проектів.

8) Учасники проекту, які працюють над завданнями не мають можливості переглядати кількість відроблених годин.

Через перераховані вище недоліки робота в компанії не є ефективною та економічно вигідною і для вирішення цієї проблеми необхідно замінити систему управління проектів на нову, більш автоматизовану та багатофункціональну, яка б допомогла мінімізувати час, який витрачається при роботі над проектами. Це зробить роботу співробітників зручною, допоможе вчасно приймати необхідні рішення по проекту та скоротить час реалізації проекту. Як наслідок це позитивно відобразиться на бюджеті компанії.

## **1.2. Формування вимог до автоматизованої інформаційної системи**

### **1.2.1. Вимоги до переліку задач**

На основі виявлених проблем в системі управління проектами компанії “FutureDev” з метою подальшої розробки концепції інформаційної системи сформулюємо вимоги до переліку задач, які повинна вирішувати інформаційна система.

Оскільки деякі працівники проводять свою діяльність віддалено від офісу, то система повинна бути доступною з будь-якого місця за умови наявності доступу до інтернету. Також повинна бути можливість доступу до системи з будь-якого пристрою, чи то смартфона, чи то комп’ютера.

Система повинна надавати можливість створення проектів та завдань, які необхідно вирішити. Для кожного завдання повинна бути можливість вказати терміни здачі, приблизний час необхідних для виконання та призначити виконавця.

Менеджеру постійно необхідно переглядати всі невиконані завдання та в ручному режимі повідомляти співробітників про відхилення від термінів здачі

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		21

завдань. Тому система повинен автоматично відстежувати терміни здачі всіх завдань для того, щоб в автоматичному режимі повідомляти виконавця та менеджера про відхилення. Це допоможе швидко приймати рішення по всіх завданням, які містять терміни здачі та мінімізує проблеми, які через це можуть виникнути. Обов'язково повинна бути можливість редагування завдання та призначення, за необхідності, нових термінів.

Під час роботи над проектом менеджер створює завдання та призначає їх співробітнику. Оскільки учасник може дізнатися про це лише зайшовши в систему, то не рідко трапляються випадки, коли деякі завдання пропускаються та не виконуються у встановлений термін. Тому система повинна сповіщати кожного учасника проекту про те, що йому було призначено нове завдання. Це дозволить швидко дізнаватися про нові задачі, які стоять перед розробниками. Також при створенні заміток система повинна автоматично сповіщати про це всіх учасників проекту.

Система повинна надавати можливість перегляду завантаженості роботою співробітників. Потрібно відображати, яка кількість завдань знаходиться у співробітника. Так, як деякі завдання мають зазначений приблизний час виконання, то потрібно відобразити підсумовану кількість годин на виконання цих завдань. Але не всі завдання містять цей час, тому потрібно відображати в окремих колонках кількість завдань, які мають оцінку часу виконання та їхня загальна сума і кількість завдань, які не мають вказаного часу виконання та загальну кількість всіх завдань. Ця інформація допоможе правильно розподілити роботу між усіма розробникам.

Система повинна автоматично формування звіти, які є необхідні для розрахунку витрат на розробку проектів та показників продуктивності команди розробників:

- 1) Витрати на співробітників, які займаються реалізацією проекту. Звіт містить інформацію по всім учасникам проекту та відображає кількість відроблених годин та витрати.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		22

- 2) Час розробки. Звіт містить інформацію по всім учасникам проекту та відображає кількість витрачених годин по кожному дню.
- 3) Робота окремого співробітника. Звіт містить інформацію по окремому учаснику проекту та відображає, яку кількість годин було витрачено, скільки вирішених завдань та витрати на співробітника.
- 4) Продуктивність команди розробників. Звіт містить інформацію по всім учасникам проекту та відображає продуктивність кожного розробника, який приймав участь у реалізації проекту. На основі виконаних завдань та витраченого часу повинна формуватися таблиця, яка відобразить яку кількість завдань було зроблено та загальну кількість витраченого часу при роботі з проектом кожним співробітником.

Автоматизоване формування звітів на основі збережених в системі даних та розрахунків витрат на співробітників скоротить час роботи бухгалтерів і, можливо, допоможе скоротити витрати компанії на додатковий персонал. Також звіти допоможуть менеджерам проектів якнайкраще сформувати команду для роботи над проектами. Тому що, на основі звітів можна буде зробити аналіз з яким типом проектів розробник справляється краще, а який погано виконує свою роботу.

Кожному учаснику проекту під час роботи необхідно самостійно відстежувати свій робочий час, тому що система не має автоматичного обліку часу під час виконання завдань. Це не зручно та після завершення завдання доводиться згадувати, коли було розпочато роботу. Отже, ще однією вимогою до нової системи є автоматичний запис часу під час виконання завдання. Це повинно відбуватися наступним чином: коли розробник починає виконувати поставлену перед ним задачу, то на сторінці цієї задачі повинна бути кнопка для запису часу. При натисканні на цю кнопку запускається лічильник, який записує час роботи співробітника. Після того, як завдання є виконаним, то необхідно натиснути на кнопку зупинки запису часу та час автоматично збережеться в системі. Також повинна бути можливість призупинення часу.

Тобто, коли співробітник закінчує роботу, але завдання ще не є вирішеним, то він натискає кнопку паузи і коли він продовжить роботу над завданням(та натисне на кнопку продовження запису часу), то час буде рахуватися не з початку, а з моменту натиснення паузи. Це значно полегшить співробітникам облік свого робочого часу та звільнить від потреби запам'ятовувати коли було розпочато роботу.

Для того, щоб забезпечити відновлення даних по проектам збережених в системі, повинно бути автоматичне створення резервних копій бази даних. Копія повинна створюватися кожного дня, а застарілі видаляти кожного тижня.

Система повинна мати автоматичне сповіщення всіх учасників проекту про те, що було створено нову замітку, яка містить необхідну інформацію для створення продукту. Це допоможе розробникам, тестувальника та дизайнерам розуміти, які є дані в їхньому доступі та не турбувати зайвий раз менеджера.

Додатковою вимогою до системи є наявність сторінки на якій кожен співробітник зможе переглянути свої робочі години за обраний ним період, який не перевищує один місяць. Ця сторінка повинна містити час та вказано завдання на яке його було витрачено.

Всі ці вимоги допоможуть краще налаштувати процес управління реалізації проекту, полегшити роботу всіх співробітників та підвищити прибитки компанії.

### **1.2.2. Вимоги до вхідної та вихідної інформації системи**

Дані, що не відповідають певним обмеженням, можуть викликати збій в роботі програми. До основних негативних наслідків відсутності валідації можна віднести:

- 1) Неможливість відновитися після збою. Не завжди система здатна «повернути все назад». Можливо, в процесі роботи система виконала якісь незворотні дії - видалила файл, відправила дані по мережі. Але навіть якщо

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		24



відновлення в принципі можливо, алгоритм відновлення може теж містити помилки, і це іноді призводить до негативних наслідків.

- 2) Додаткове навантаження на систему. Відновлення після збою - це зайва робота. Вся робота, яка була виконана до моменту збою - теж зайва. А це означає додаткове навантаження на систему, яку можна уникнути, якщо заздалегідь перевірити дані.
- 3) Ін'єкції не викликають збоїв. Один з основних способів експлуатації вразливостей в системах полягає в тому, щоб «обдурити» валідатори, тобто передати дані, які валідатор визнає коректними, але при цьому вони інтерпретуються неналежним чином, тому зловмисник може отримати несанкціонований доступ до даних або певних розділів системи. Якщо валідації немає взагалі, завдання зловмисника максимально спрощується.

Отже, система повинна мати валідацію даних для того щоб уникнути проблем, які були описані вище. Інформаційна система повинна містити наступні види валідації даних:

- 1) Посимвольна перевірка. Перевірки виконуються в призначеному для користувача інтерфейсі, по мірі введення даних.
- 2) Перевірка окремих значень. Перевірки виконуються в призначеному для користувача інтерфейсі. Перевіряються значення в окремих полях. Також програмного інтерфейсу (API) повинен здійснювати перевірку параметрів, які були отримані з інтерфейсу користувача.
- 3) Сукупність вхідних значень. Перевірка повинна відбуватися не лише для окремих значень, а й взаємозв'язків між ними, взаємних обмежень.
- 4) Перевірка стану системи після обробки даних. Якщо валідацію безпосередньо вхідних даних виконати не вдається – потрібно їх обробити, але залишити можливість повернути все до початкового стану. Такий механізм називається транзакції.

Транзакція – група послідовних операцій з базою даних, яка є логічною одиницею роботи з даними. Транзакція може бути виконана або цілком і

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		25

успішно, дотримуючись цілісності даних і незалежно від інших транзакцій, що йдуть паралельно, або не виконана зовсім, і тоді вона не може справити ніякого ефекту. Транзакції оброблюються транзакційними системами, в процесі роботи яких створюється історія транзакцій. Транзакції з базою даних використовуються для досягнення наступних цілей:

- 1) Забезпечення надійних робочих елементів, які дозволяють правильно відновити роботу у випадку збоїв та зберігати цілісність бази даних у випадку системних відмов, коли виконання операцій зупиняється (повністю або частково) і більшість операцій над базою даних залишаються незавершеними з нез'ясованим статусом.
- 2) Для забезпечення роздільного доступу для процесів, що одночасно звертаються до бази даних. При відсутності ізоляції операцій, результати отримані процесами можуть бути помилковими.

### **1.3. Аналіз існуючих рішень**

#### **1.3.1. Автоматизовані інформаційні системи**

Зважаючи на важливість системи управління проектами на сьогоднішній день існує велика кількість програм, які створені для управління ІТ-проектами. Тому важливо провести аналіз існуючих рішень для виявлення системи, яка б задовільнила всім поставленим вимогам до автоматизованої інформаційної системи управління проектами. Для аналізу було обрано 5 існуючих рішень, які описані в наступних підрозділах.

#### **1.3.2. Додаток KPlato**

KPlato – додаток для управління проектами під ОС GNU Linux. Цей додаток є безкоштовним, але малофункціональним і не відповідає всім поставленим вимогам. Так, як він є стаціонарним програмним забезпеченням ОС GNU Linux, то робота з додатком вимагає встановлення його в ОС GNU

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		26

Linux, що робить неможливим взаємодію з системою при використанні іншої ОС [24].

### 1.3.3. Додаток OpenProj

OpenProj – десктопне кросплатформне програмне забезпечення для управління проектами. Доступна для операційних систем Microsoft Windows, GNU Linux, Mac OS X. Поширюється на умовах ліцензії Common Public Attribution License Version 1.0 (CPAL). Написаний на мові програмування Java. Позиціонується, як відкрита заміна комерційного продукту Microsoft Project. Продукт розроблений у 2007 році компанією волонтерів Marc O'Brien, Howard Katz and Laurent Chretienneau. Так, як додаток був створений ще в 2007 році, то він має застарілий вигляд та не відповідає всім поставленим вимогам. Для використання необхідно встановлювати додаткове ПО в ОС кожного працівника та система не має обліку часу працівників [25].

### 1.3.4. Додаток GanttProject

GanttProject – програма, призначена для планування проектів на основі побудови діаграм Ганта і діаграм типу PERT. Програма є ліцензованим GPL (безкоштовне програмне забезпечення) програмним забезпеченням для управління проектами на базі Java, яке працює в операційних системах Microsoft Windows, GNU Linux та Mac OS X. Цей проект був ініційований у січні 2003 р. в університеті Марн-ла-Валле (Франція). Можна сказати, що GanttProject розроблений з урахуванням принципу KISS. Він містить найбільш основні функції управління проектами, такі як діаграма Ганта для планування проектів завдань та управління ресурсами за допомогою графіків завантаження ресурсів. Додаток може обробляти лише дні, а не години. Він не має таких функцій, контроль над повідомленнями та документами. Він має ряд варіантів звітування (MS Project, HTML, PDF, електронні таблиці). Додаток надає можливість збереження файлів для швидкого доступу до них [26].

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		27

### 1.3.5. Система управління проектами Basecamp

Basecamp – це платна клієнт-серверна система управління проектами. Розробляється однойменною компанією 37signals. Дана система проста у використанні, але разом з простотою їй гостро бракує життєво важливого функціонала для того, щоб її можна було використовувати в серйозних, більш-менш великих проектах. Розробники з 37signals спростили все, що тільки можна було спростити: неможливо простежити за станом проекту, тільки за загальним прогресом, все спілкування відбувається у вигляді блогу, завдання ставляться як to-do списки, що зручно тільки в разі загальних описів для розвитку проекту. Також немає можливості створювати звіти [27].

### 1.3.6. Веб-додаток ActiveCollab

ActiveCollab – це онлайн-інструмент для управління проектами та створення платформи для спільної роботи. Він зберігає всі дані по проекту в одному місці, централізовано керуючи ними. Його можна встановити як на сервер, так і користуватися в хмарі, вимірюючи успіхи в реалізації проекту і управляючи часом. ActiveCollab дозволяє оцінювати продуктивність праці співробітників, аналізувати їх профілі. Щоб допомогти вам відстежувати час і, по суті, забезпечувати продуктивність роботи всіх членів команди. ActiveCollab дозволяє перевіряти час, витрачений на виконання завдань, стежити за часом у кількох завданнях та надсилати свої записи часу в ActiveCollab.

Основні характеристики ActiveCollab включають в себе:

- Центр спільної роботи над проектом.
- Настроюється домашній екран.
- Оцінка і відстеження часу.
- Інтеграція з Email.
- Командну взаємодію.
- Управління проектом.
- Управління завданнями.

- Календар.
- Файловий менеджер.
- Перегляд профілів.

Можна зробити висновок, що ActiveCollab може стати системою, яку можна використовувати для управління проектами. Але все таки, він не має всіх функцій, які описані в вимогах до системи. Цей додаток не вміє відстежувати витрати на розробку проектів, створювати звіти та відображати завантаженість співробітників.

### 1.3.7. Порівняльний аналіз

На основі представлених описів існуючих рішень для візуального представлення функцій систем управління проектами розробимо порівняльну таблицю (таблиця 1.1) існуючих рішень. В таблиці представлені головні функції, які повинна містити сучасна система управління проектами.

Таблиця 1.1 – Функції існуючих систем управління проектами

Функція \ Додаток	KPlato	OpenProj	GanttProject	Basecamp	ActiveCollab
Відстеження часу роботи	±	-	-	+	+
Призначення термінів виконання	+	+	+	+	+
Автоматичний контроль за дедлайнами	-	-	-	-	+
Створення звітів	-	-	-	+	+
Відстеження витрат	-	-	+	+	-
Обчислення показників завантаженості	-	-	-	-	±
Збереження файлів	-	-	+	+	+

На основі представлених описів існуючих рішень для візуального представлення експлуатаційних властивостей систем управління проектами розробимо порівняльну таблицю (таблиця 1.2) існуючих рішень.

Таблиця 1.2 – Експлуатаційні властивості існуючих систем управління проектами

Додаток	KPlato	OpenProj	GanttProject	Basecamp	ActiveCollab
Експлуатаційна властивість					
Кросплатформеність	-	±	±	±	+
Інтеграція з Email	-	-	-	+	+
Інтеграція з Slack	-	-	-	-	+
Локальне розгортання	+	+	+	+	-
Низькі системні вимоги	+	-	+	+	+
Надійсність	+	+	+	+	+

З таблиці 1.1 та 1.2 видно, що жоден із додатків не містить весь набір необхідних функцій та експлуатаційних властивостей. Тому створення нової інформаційної системи управління проектами ІТ-компанії є необхідним.

В представлених існуючих рішеннях для управління проектами є рішення, які можна запозичити для нової інформаційної системи. До таких рішень відноситься: можливість збереження файлів, для швидкого доступу до них, інтеграція з Slack, яка дозволить відправляти повідомлення в месенджер Slack, локальне розгортання, яке надасть можливість компаніям встановлювати систему на власні сервера, низькі системні вимоги, що дозволить використовувати систему з більшості пристроїв та надійність.

#### **1.4. Розробка концепції автоматизованої інформаційної системи**

Розробка концепції є рекомендованою фазою створення автоматизованої інформаційної системи, що стоїть між фазами "постановка вимог до інформаційної системи" і "розробкою програмних рішень". Концепція покликана досягти певного рівня структурування ідеї автоматизованої інформаційної системи управління проектами та визначити процеси і їх учасників.

Головно метою автоматизованої інформаційної системи управління є підвищення ефективності процесу управління проектами ІТ-компанія. Автоматизована інформаційна система дозволить більш швидко та економічно вигідно створювати нові та удосконалювати існуючі ІТ-продукти.

На основі опису діяльності, організаційної структури компанії та поставлених вимог до інформаційної системи визначимо основні процеси та їх учасників.

В системі повинні відбуватися наступні процеси:

- 1) Вести облікові записи. Процес включає в себе наступні підпроцеси:
  - 1.1) Створити облікові записи.
  - 1.2) Активувати облікові записи.
  - 1.3) Редагувати облікові записи.
  - 1.4) Формувати та переглядати активність облікових записів.
  - 1.5) Видалити облікові записи.
- 2) Вести проекти. Процес включає в себе наступні підпроцеси:
  - 2.1) Створити проекти.
  - 2.2) Редагувати проекти.
  - 2.3) Здійснювати пошук проектів.
  - 2.4) Підключати та видаляти учасників проектів.
  - 2.5) Видаляти проекти.
- 3) Вести завдання. Процес включає в себе наступні підпроцеси:
  - 3.1) Створити завдання.

- 3.2) Редагувати завдання.
- 3.3) Видалити завдання.
- 3.4) Здійснювати пошук завдань.
- 3.5) Вести облік часу виконання завдань.
- 3.6) Створити коментарі до завдань.
- 4) Вести замітки. Цей процес включає в себе наступні підпроцеси:
  - 4.1) Створити замітки.
  - 4.2) Редагувати замітки.
  - 4.3) Здійснювати пошук заміток.
  - 4.4) Видалити замітки.
- 5) Створити звіти. Цей процес включає в себе створення наступних звітів:
  - 5.1) Витрати на учасників проекту. Звіт містить інформацію по всім учасникам проекту та відображає кількість відроблених годин та витрати.
  - 5.2) Час розробки. Звіт містить інформацію по всім учасникам проекту та відображає кількість витрачених годин по кожному дню.
  - 5.3) Робота учасника проекту. Звіт містить інформацію по окремому учаснику проекту та відображає, яку кількість годин було витрачено, скільки вирішених завдань та витрати на співробітника.
  - 5.4) Продуктивність учасників проекту. Звіт містить інформацію по всім учасникам проекту та відображає продуктивність кожного розробника, який приймав участь у реалізації проекту. На основі виконаних завдань та витраченого часу повинна формуватися таблиця, яка відобразить яку кількість завдань було зроблено та загальну кількість витраченого часу при роботі з проектом кожним співробітником.
- 6) Сформувані показники продуктивності та зайнятості учасників проектів. Цей процес включає в себе наступні підпроцеси:
  - 6.1) Сформувані показники продуктивності учасників проектів.



6.2) Сформуванати показники зайнятості учасників проектів.

На етапі опису організаційної структури компанії було виявлення, що в компанії присутні наступні посади співробітників:

- 1) Директор.
- 2) Менеджер проектів.
- 3) Програміст.
- 4) Дизайнер.
- 5) Тестувальник.
- 6) Бухгалтер.

Система повинна містити в собі комплекс технічних засобів, які призначені для обробки даних в інформаційній системі. До складу цього комплексу входять наступні засоби:

- 1) Збору та реєстрації інформації.
- 2) Передачі даних по каналам зв'язку.
- 3) Накопичення та збереження даних.
- 4) Контролю збереженої інформації.
- 5) Розрахунку на основі зареєстрованої інформації та видачі результатів.

Система не повинна обмежувати кількість користувачі, які одночасно взаємодіють із системою. Для цього потрібно обрати правильну структуру, яка буде містити в собі всі вищеописані засоби.

Найбільш підходящою архітектурою інформаційної системи, яка зможе задовільнити всім описаним потребам є клієнт-серверна архітектура. Ця архітектура є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти: набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них; набір клієнтів, які використовують сервіси, що надаються серверами; мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Клієнт-серверна архітектура не потребує додаткового програмного забезпечення, достатньо використання вбудованих можливостей операційних систем – браузерів, які відіграють роль клієнта. Отже, така архітектура має всі необхідні засоби. Тому саме її було обрано для реалізації системи автоматизованого управління проектами.

Як вже було сказано, клієнтом в системі є браузер, а серверну частину буде представлено у вигляді RESTfull API додатку. REST - (Representational State Transfer – «передача репрезентативного стану») архітектурний стиль взаємодії компонентів розподіленого додатку в мережі за моделлю клієнт-сервер.

Особливості архітектурного стилю:

- Кожна сутність повинна мати унікальний ідентифікатор - URI.
- Сутності повинні бути пов'язані між собою.
- Для читання і зміни даних повинні використовуватися стандартні методи HTTP протоколу.
- Повинна бути підтримка декількох типів ресурсів.

Стандартні методи архітектури:

- GET - отримання даних без їх зміни. Це найбільш популярний і легкий метод. Він тільки повертає дані, а не змінює їх.
- POST - метод, що припускає вставку нових записів.
- PUT - метод, що припускає зміну існуючих записів.
- PATCH - метод, що припускає зміну ідентифікатора існуючих записів.
- DELETE - метод, що припускає видалення записів.

RESTfull API – це повний набір віддалених викликів стандартних методів, які повертають дані у певному форматі. До головних переваг архітектури RESTfull можна включити наступні:

- Відсутність додаткових внутрішніх прошарків, що означає передачу даних в тому вигляді, що і самі дані. Тобто дані не загортаються в XML.
- Кожна одиниця інформації однозначно визначена URL – це значить, що URL являється по суті первинним ключем для одиниці даних.
- Управління інформацією ресурсу базується на протоколі передачі даних, найбільш популярним протоколом є HTTP.

Для того, щоб представити загальну архітектуру, процеси та їх учасників, на основі представлених вище процесів та посад співробітників компанії розробимо концептуальну модель інформаційної системи. Для створення моделі використаємо уніфіковану мову моделювання UML.

Концепція автоматизованої інформаційної системи управління проектами ІТ-компанії представлена на рисунку 1.3.

### **1.5. Постановка задачі**

На основі концепції інформаційної системи з метою розробки проектних рішень автоматизованої інформаційної системи управління проектами ІТ-компанії здійснимо постановку задачі.

Необхідно створити систему управління проектами ІТ-компанії, яка складається з двох частин: серверної та користувацького веб-інтерфейсу. Всі необхідні таблиці повинні створюватися міграціями для того, щоб в подальшому можна буде легко і швидко розширювати функціонал системи та здійснювати розгортання. Для даних, які заповнюються в базу даних при інсталяції повинні бути створені сіди. Ієрархія проекту має складатися з пакетів, кожен з яких містить свій функціонал. Вся робота з сайтом повинна відбуватися без перезавантаження сторінок. Але у випадках, коли в цьому є необхідність перезавантаження дозволяється.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>35</b>

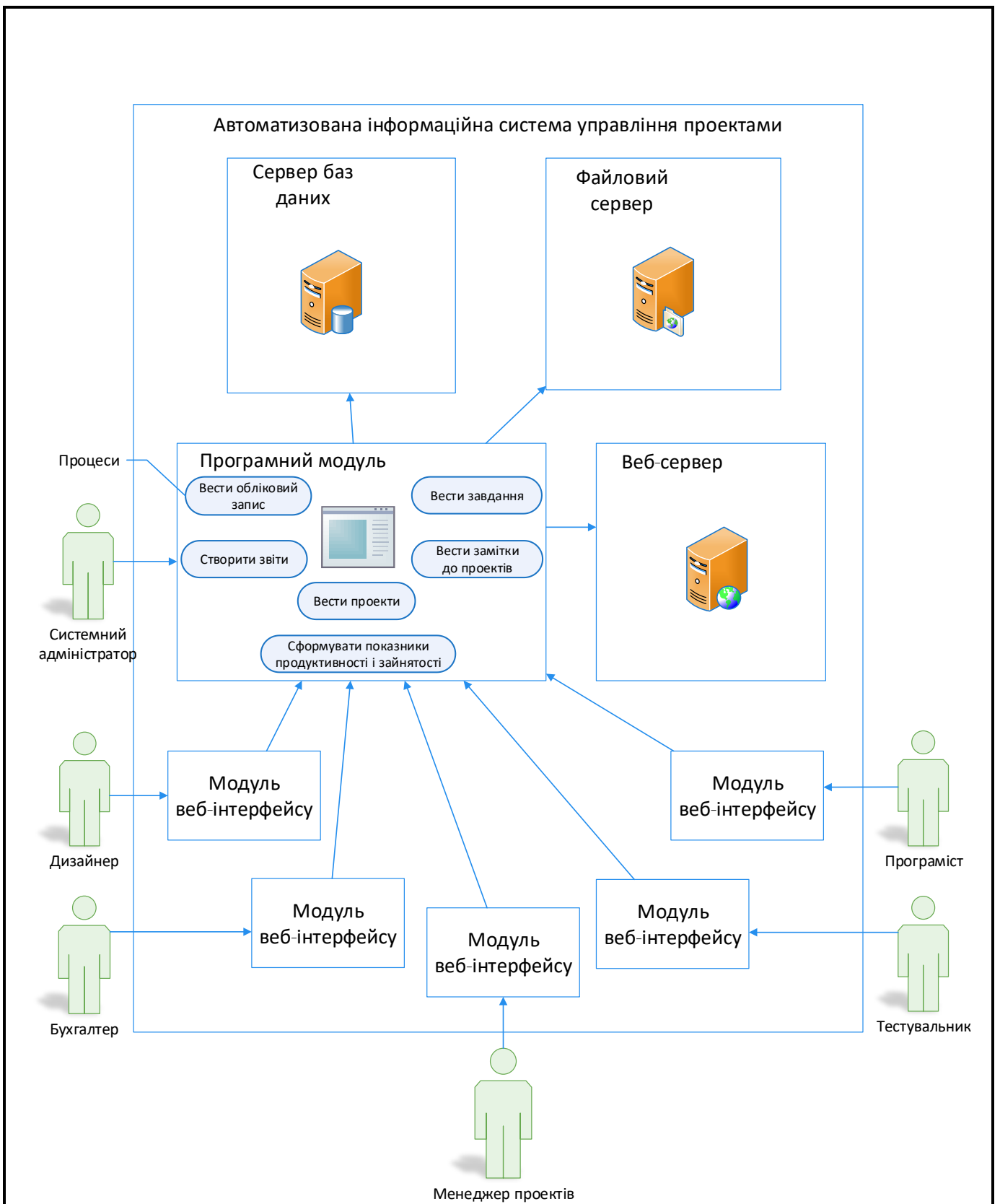


Рисунок 1.3 – Концепція автоматизованої інформаційної системи управління проектами ІТ-компанії

Зм.	Аркуш	№ документа	Підпис	Дата

Користувач повинен мати можливість зареєструватися в системі. Для цього необхідно ввести наступні данні:

- 1) Ім'я - ім'я користувача.
- 2) Ел. адреса - електронна адреса, яка буде використовуватися для подальшої авторизації, відправки повідомлень та підключення до проектів. Після реєстрації на електронну адресу потрібно відправляти повідомлення для підтвердження коректності email. До цього користувач не буде активований та не зможе створювати або приймати участь у розробці інших проектів.
- 3) Пароль - пароль, який буде використовуватися для авторизації.
- 4) Підтвердження паролю - підтвердження паролю для уникнення подальших проблем з авторизацією.

Кожен зареєстрований користувач повинен мати персональну сторінку на якій будуть відображатися персональні дані. Так, як користувач тільки здійснив реєстрацію і не заповнив всіх даних (лише ім'я та електронну адресу), то у верхній частині сторінки необхідно відобразити повідомлення про те, що необхідно перейти в налаштування профілю та закінчити заповнення даних акаунту. Також, якщо електронну адресу ще не було підтверджено, то відповідне повідомлення. Поки ці два попередження не будуть виконані потрібно обмежити доступ до сторінок, які стосуються проектної частини, тобто не можна буде створити проект або підключитися до розробки інших.

Необхідно створити можливість редагування даних користувача, підключати акаунти інших сервісів: Google та Facebook для подальшої авторизації за допомогою цих даних. На сторінці налаштувань профілю повинна бути можливість зберегти наступні дані: фото користувача, статус, ім'я та прізвище (\*), дата народження(\*), номер телефону, адреса для slack вебхуку, ім'я в скайпі. Поля повинні проходити валідацію: номер телефону – містити лише цифри та мати довжину 10 символів, електронна адреса – містить символ '@' та '.' з інтервалом в 2 символи, адреса для slack вебхуку – має формат

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		37

https://\*\*\*. Якщо якийсь поле не є валідним, то біля нього вивести повідомлення про помилку. При натисканні на кнопку збереження у випадку валідних даних вся інформація повинна зберігатися та з'являтися впливаючий попап зеленого кольору з повідомленням, що операція пройшла успішно або червоного, якщо сталася помилка.

Для зручного пошуку користувачів, необхідно створити url профіля в форматі `task-manager.com/profile/{nickname}`. Nickname повинен генеруватися автоматично при реєстрації на основі електронної адреси користувача, але в налаштуваннях акаунту повинна бути можливість редагування цього значення.

Головна сторінка профіля користувача повинна бути поділена на дві секції: 1) також розділена на дві секції: а) фото користувача, ім'я та прізвище та електронна адреса; б) інформації про користувача у формі картки: статус, дата народження, номер телефону, скайп; 2) вкладки з інформацією: а) швидкий огляд - календар активності, останні 5 дій активності та 5 останніх персональних проектів користувача; б) активність - повний список активності користувача: виконані та нові завдання, коментарі, створення та підключення проектів, створення приміток; в) співробітництво з проектами - список всіх проектів, в яких учасник приймає участь; г) персональні проекти - власні проекти користувача.

Після того, як користувач активував свій обліковий запис у системі (підтверджено електронну адресу та заповнено обов'язкові дані), то функція створення нових проектів та підключення до інших повинна активізуватися. Відповідно необхідно створити можливість створення нових проектів. Для цього потрібно виділити окрему кнопку при натисканні на яку з'являється форма з наступними полями: назва (\*), описання (\*), стадія, категорія, назва клієнтської компанії, відстеження часу розробки та витрат (ці опції будуть завжди активними, виводяться для того, щоб показати, що в проекті буде відстежуватися час розробки та витрати), валюта, бюджет.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		38

Наступним кроком створення нового проекту повинно бути підключення розробника(-ів) та клієнта. Для їх підключення необхідно ввести електронну адресу та обрати посаду: адміністратор, розробник, тестувальник. Тобто користувач вже повинен бути зареєстрований в системі, а також якщо клієнт зареєстрований в системі, то його буде підключено до проекту, в іншому випадку просто збережеться електронна адреса. Також для зручного підключення повинна бути можливість перегляду всіх користувачів, які підключені до компанії (якщо раніше вже створювалися проекти). Всім підключеним користувачам на електронну адресу та в slack, за умови, що його було підключено, повинно прийти повідомлення про те, що їх було підключено до розробки проекту. Компанія повинна реєструватися при створенні користувачем першого проекту. Також необхідно створити можливість редагувати дані компанії: назву, веб-сайт, номер телефону та список співробітників.

Потрібно надати можливість перегляду всіх підключених та власних проектів користувача. Також необхідно створити головну сторінку кожного проекту, яка буде містити наступні вкладки: Завдання, Нотатки, Час, Адміністрування. На головній сторінці повинен відображатися список всіх завдань с позначкою виконавця та згруповані по створеному списку завдань. На цій же сторінці повинна бути можливість створити список завдань та самі завдання.

Необхідно надати можливість створювати в системі завдання. Вони повинні містити наступні дані: назва(\*), опис(\*), вкладення, учасники, виконавець, термін виконання, помітки статусу виконання, час необхідний для виконання. Після створення завдання повинне прийти повідомлення на електронну адресу та в slack користувачів, які були вказані, як учасники. Також, якщо виконавця було обрано, то у цього користувача на сторінці “My Work” з’явиться нове завдання.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		39

Необхідно створити можливість переглядати зміст завдань. Сторінка повинна відображати назву завдання та проекту, ім'я створювача, дату створення, опис, закріпленні файли, поле для нових та вже створенні коментарі, ім'я виконавця, строки виконання, помітки статусу виконання, скільки часу було виділено на виконання, список підключених розробників до завдання. Якщо зміст завдання переглядає користувач, який призначений, як виконавець, то повинна відображатися кнопка для запису робочого часу, який витрачається на виконання. При натисненні на цю кнопку повинен початися запис часу, а при натисненні на стоп час автоматично збережеться та при продовженні роботи над завданням новий час повинен підсумовуватися до вже збереженого.

В завдань є поле у якому вказується дата завершення, тому потрібно реалізувати автоматичний контроль за цим показником і необхідно відправляти повідомлення адміністратору проекту та виконавцю завдання, що спливає дата задачі. Час коли буде відправлено це повідомлення повинен приймати до уваги назначену кількість часу на виконання.

Повинна бути можливість зміни виконавця завдання та при його зміні новий повинен отримати повідомлення про те, що йому потрібно продовжити розробку. Після завершення робіт над завданням, повинна бути можливість помітити його, як виконане і воно зникне зі сторінки "Моя робота" у всіх користувачів, які працюють над проектом.

Наступною вкладкою сторінки проекту є Нотатки. Необхідно реалізувати можливість створення нотатків, які повинні містити заголовок, контент та можливість прикріплення додаткових файлів будь-якого формату. Після створення нового нотатку всі учаснику проекту в якому його було створено повинні отримати повідомлення.

Вкладка Час необхідна для того, щоб учасники проекту змогли переглянути список виконаних ними завдань та скільки часу вони на це витратили. Тому, необхідно реалізувати алгоритм, який би збирав та

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		40



згрупував дані по виконаним завданням учасника проекту і відповідно відобразити ці дані на сторінці.

Ще однією вкладкою є Адміністрування і вона відображається лише для власника проекту та адміністратора. Сторінка адміністрування проекту повинна містити можливості створення звітів та формування показників роботи співробітників. Для звітів повинна бути можливість переглядати їх через браузер, а також скачувати у вигляді pdf файлу. Повинні формуватися наступні звіти:

- 1) Витрати на співробітників, які займаються реалізацією проекту. Звіт містить інформацію по всім учасникам проекту та відображає кількість відроблених годин та витрати.
- 2) Час розробки. Звіт містить інформацію по всім учасникам проекту та відображає кількість витрачених годин по кожному дню.
- 3) Робота окремого співробітника. Звіт містить інформацію по окремому учаснику проекту та відображає, яку кількість годин було витрачено, скільки вирішених завдань та витрати на співробітника.
- 4) Продуктивність команди розробників. Звіт містить інформацію по всім учасникам проекту та відображає продуктивність кожного розробника, який приймав участь у реалізації проекту. На основі виконаних завдань та витраченого часу повинна формуватися таблиця, яка відобразить яку кількість завдань було зроблено та загальну кількість витраченого часу при роботі з проектом кожним співробітником.

Так, як у компанії може бути декілька проектів і кожен із працівників може відразу приймати участь у розробці декількох, то обов'язково необхідно створити загальну сторінку для формування витрат на співробітників фірми. Тобто розрахунок повинен вестися не по окремому проекту, а по всім в яких співробітник приймає участь. Також повинна бути сторінка, яка містить деталі компанії: назва, веб-сайт, номер телефону та список співробітників, який можна редагувати.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		41

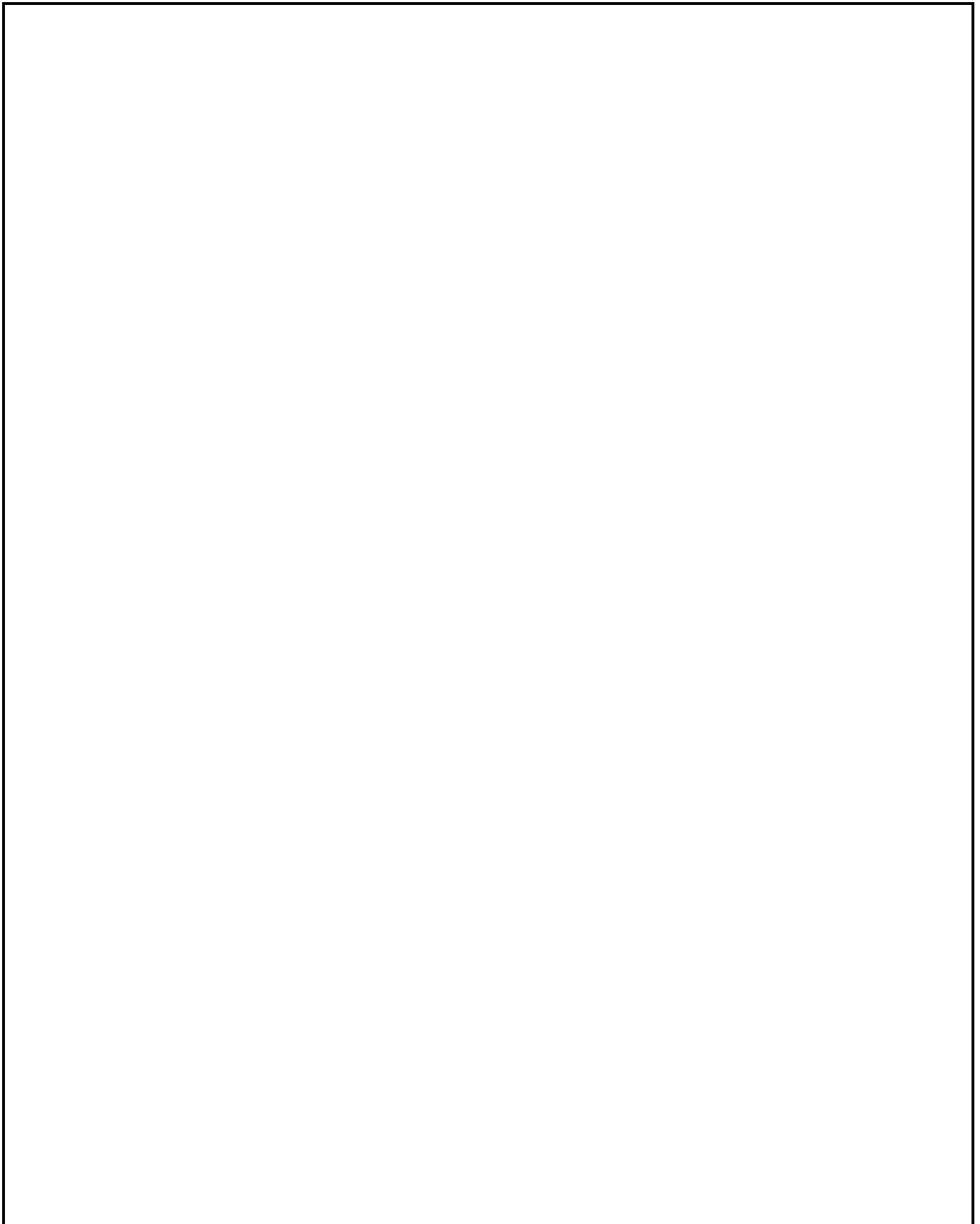
Технічне завдання для розробки інформаційної системи управління проектами ІТ-компанії представлено в додатку А.

### **1.6. Висновки**

В першому розділі було проведено аналіз діяльності об'єкта та його організаційної структури, визначенні недоліки системи управління проектами компанії, сформовані вимоги до нової автоматизованої інформаційної системи управління проектами та проведено аналіз існуючих рішень.

В результаті було розроблено концепцію інформаційної системи та представлено постановку задачі. На основі концепції та постановки задачі будуть розроблені програмні рішення автоматизованої інформаційної системи управління проектами ІТ-компанії.

					122.4142.08.ПЗ.Р1	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		42



					122.4142.08.ПЗ.Р2			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Розробка проектних рішень автоматизованої інформаційної системи управління проектами ІТ-компанії	Літ.	Аркуш	Аркушів
Студент	Михалко М.І.						43	28
Керівник	Гайда А.Ю.					НУК		
Зав. каф.	Михелєв І.Л.							

## 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ

### 2.1. Загальносистемні рішення автоматизованої інформаційної системи

#### 2.1.1. Діаграми варіантів використання

Діаграма прецедентів - UML діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання. Актори на діаграмі варіантів використання позначаються символом людини, а варіанти використання – еліпсом.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою, так званих, варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації;
- включення;
- розширення;
- узагальнення.

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення [4, 6, 9].

На основі представлених процесів, підпроцесів та їх учасників які представлені в підрозділі розробки концепції автоматизованої інформаційної

системи з метою виявлення сутностей, їхніх властивостей та взаємодії розробимо діаграму використання.

Діаграма варіантів використання автоматизованої інформаційної системи управління проектами представлена на рисунку 2.1.

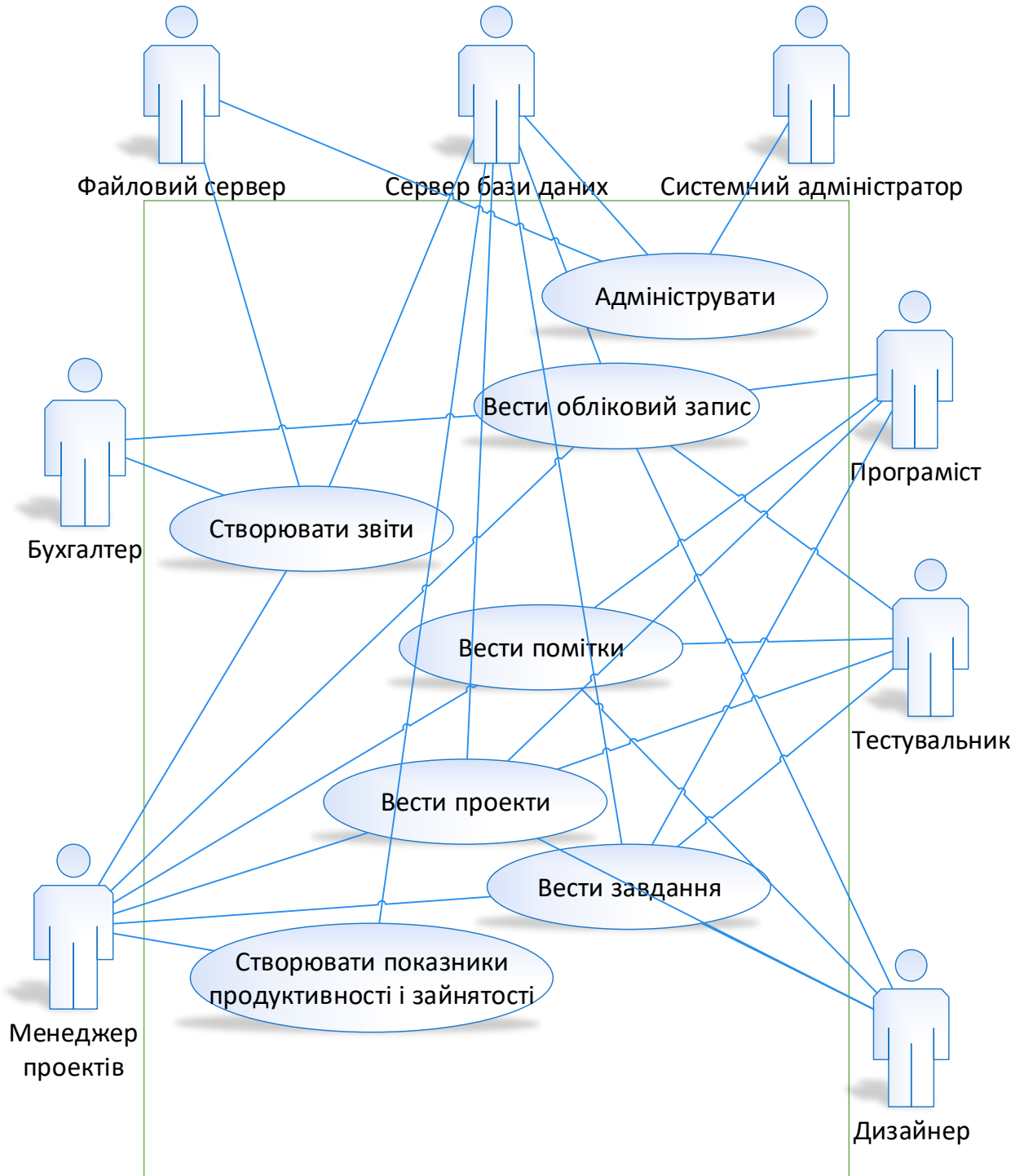


Рисунок 2.1 – Діаграма варіантів використання автоматизованої інформаційної системи управління проектами

Зм.	Аркуш	№ документа	Підпис	Дата

На основі представленої на рисунку 2.1 загальної діаграми використання автоматизованої інформаційної системи проведемо декомпозицію кожного із варіантів використання.

Декомпозиція варіанта використання “Вести обліковий запис” представлена на рисунку 2.2.

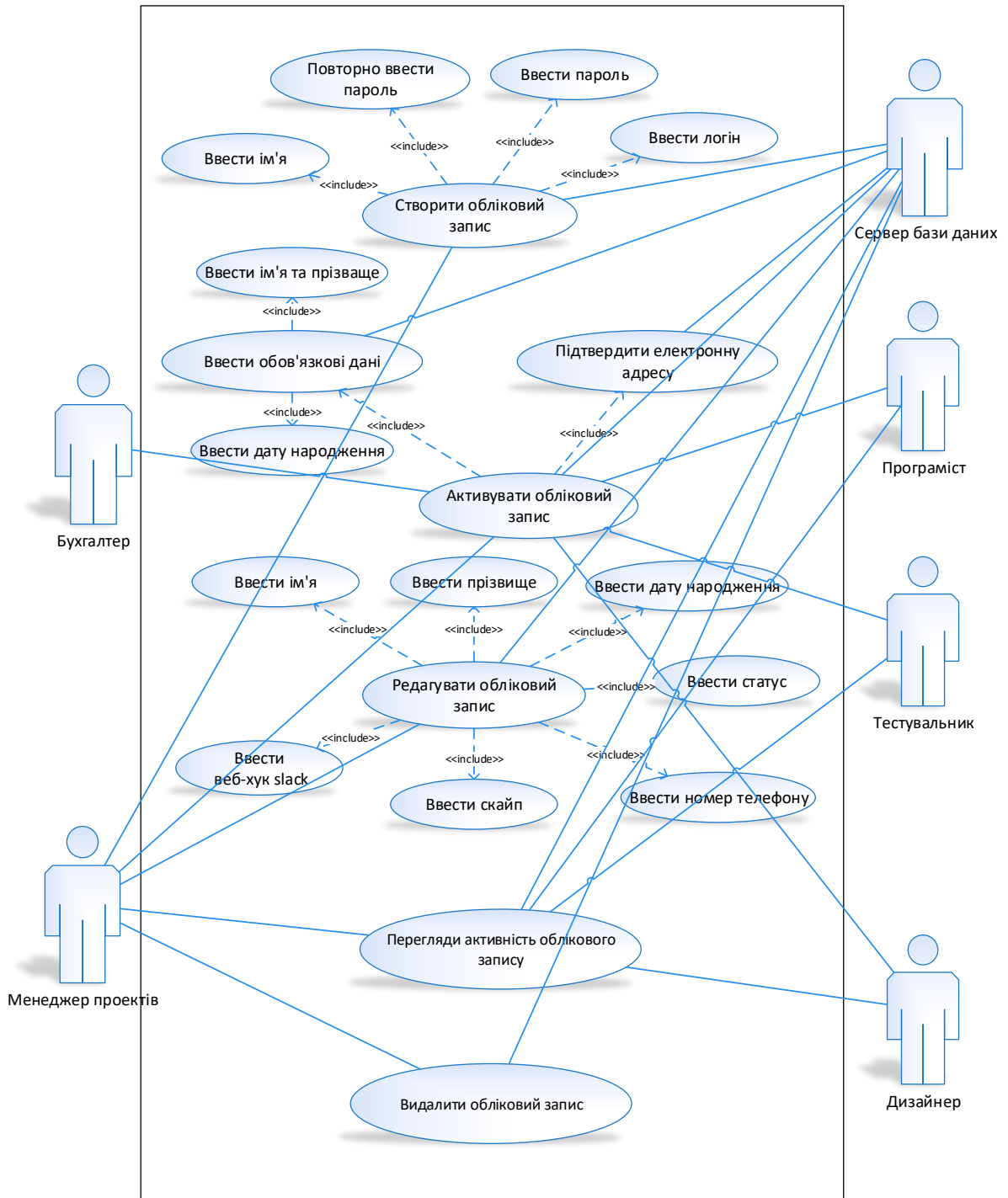


Рисунок 2.2 – Діаграма декомпозиції варіанта використання “Вести обліковий запис”

Зм.	Аркуш	№ документа	Підпис	Дата

Декомпозиція варіанта використання “Вести проекти” представлена на рисунку 2.3.

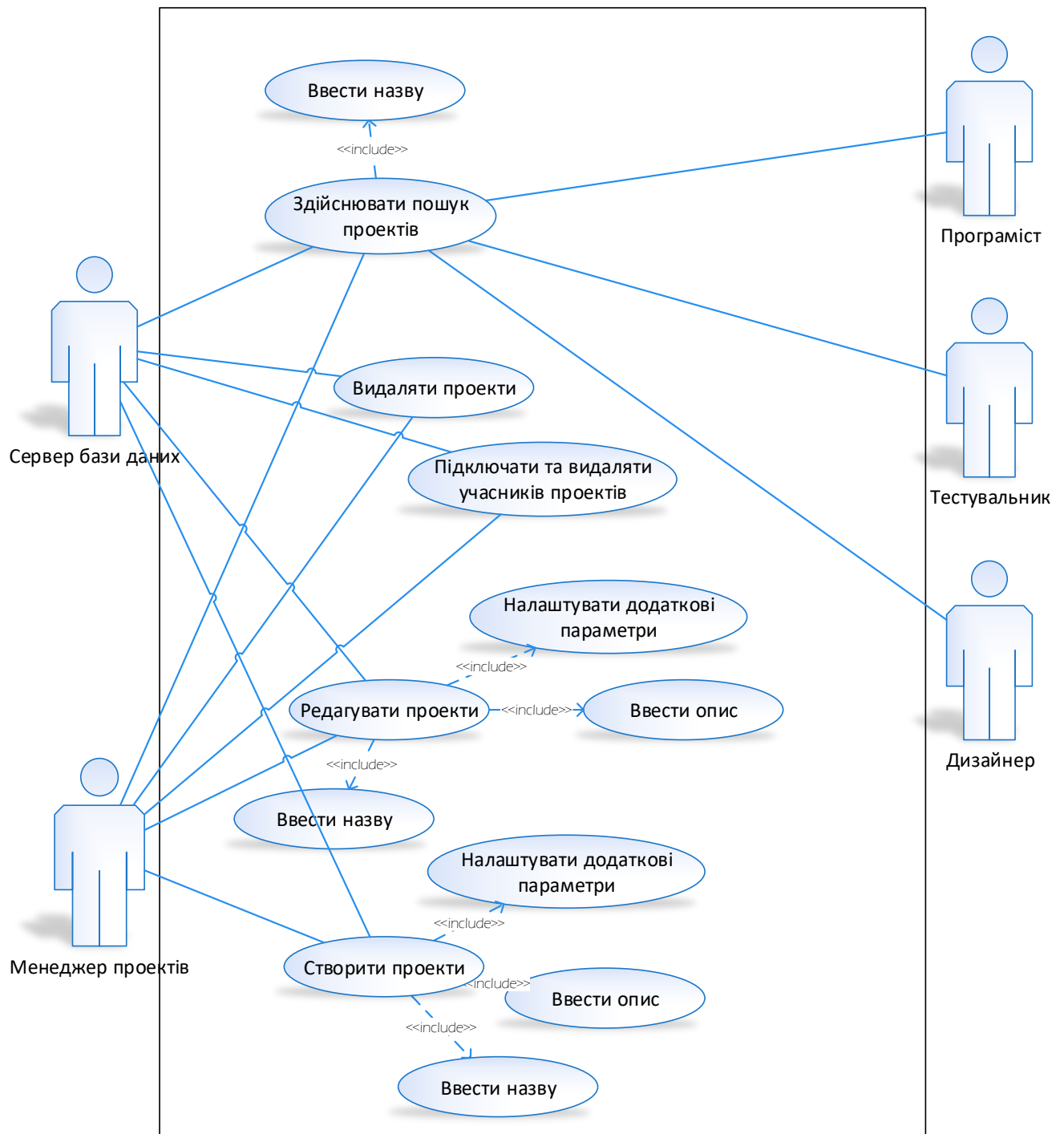


Рисунок 2.3 – Діаграма декомпозиції варіанта використання “Вести проекти”

Зм.	Аркуш	№ документа	Підпис	Дата

Декомпозиція варіанта використання “Вести завдання” представлена на рисунку 2.4.

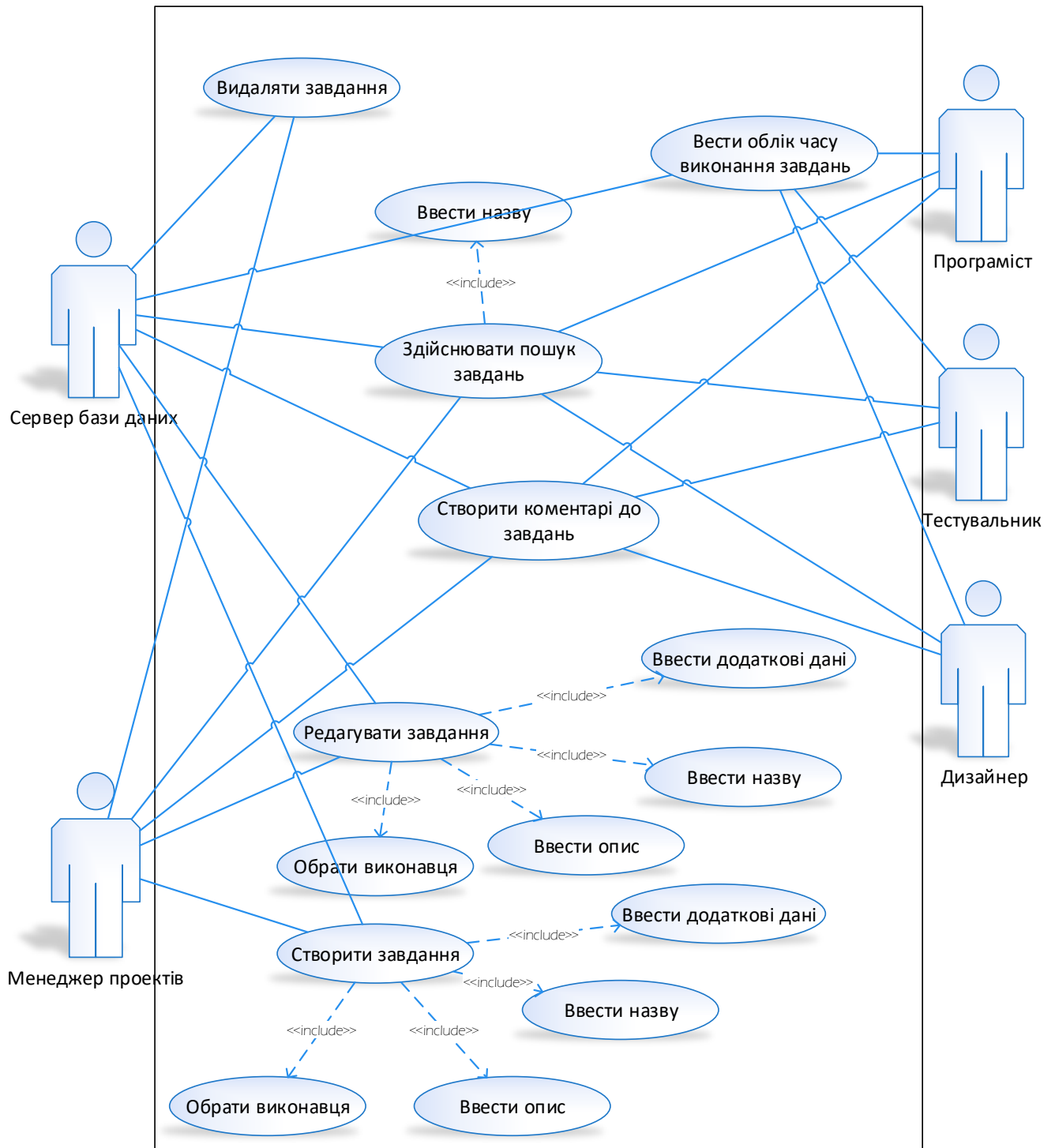


Рисунок 2.4 – Діаграма декомпозиції варіанта використання “Вести завдання”

Зм.	Аркуш	№ документа	Підпис	Дата



Декомпозиція варіанта використання “Вести замітки” представлена на рисунку 2.5.

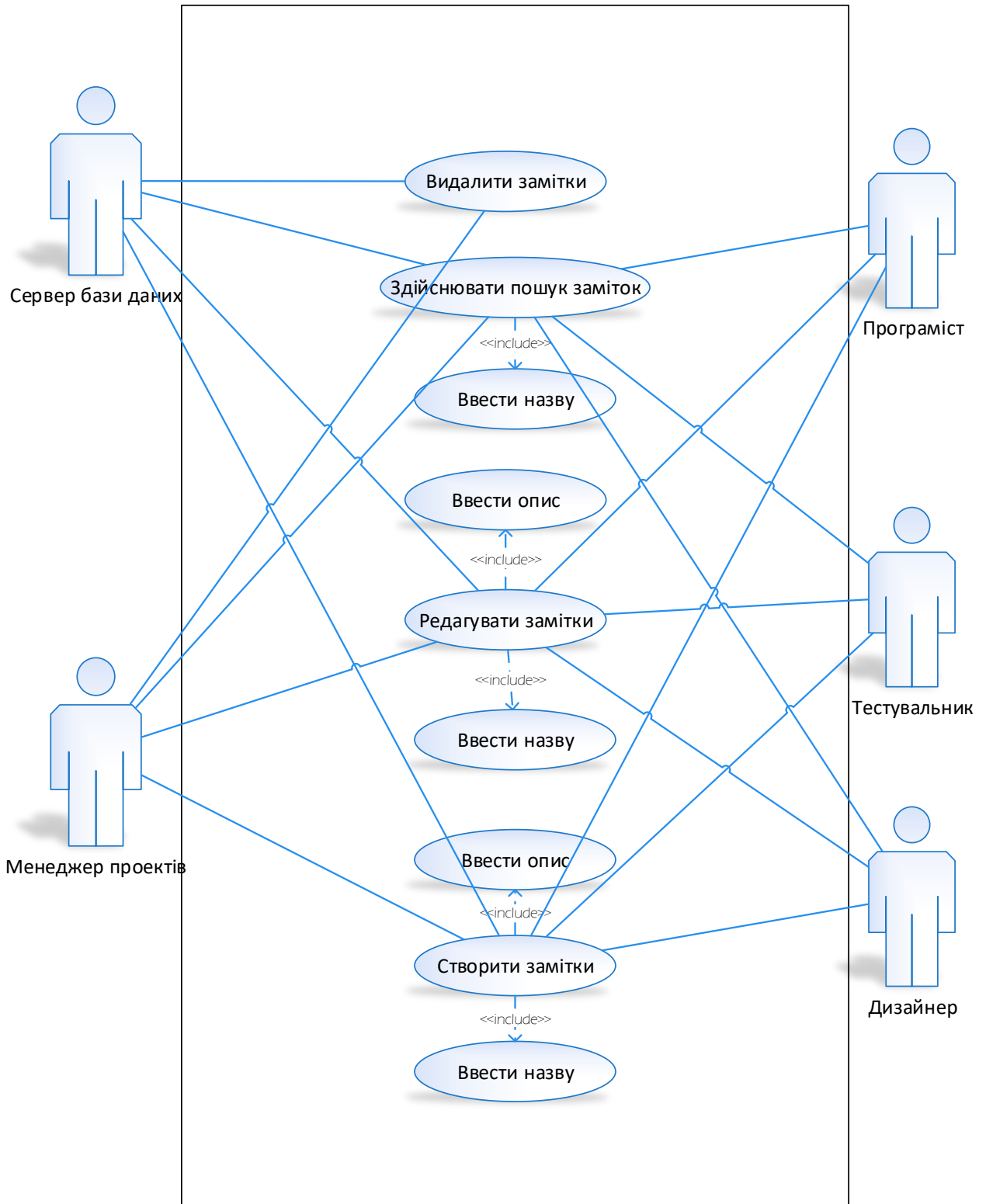


Рисунок 2.5 - Діаграма декомпозиції варіанта використання “Вести замітки”

Зм.	Аркуш	№ документа	Підпис	Дата

Декомпозиція варіанта використання “Створювати звіти” представлена на рисунку 2.6.

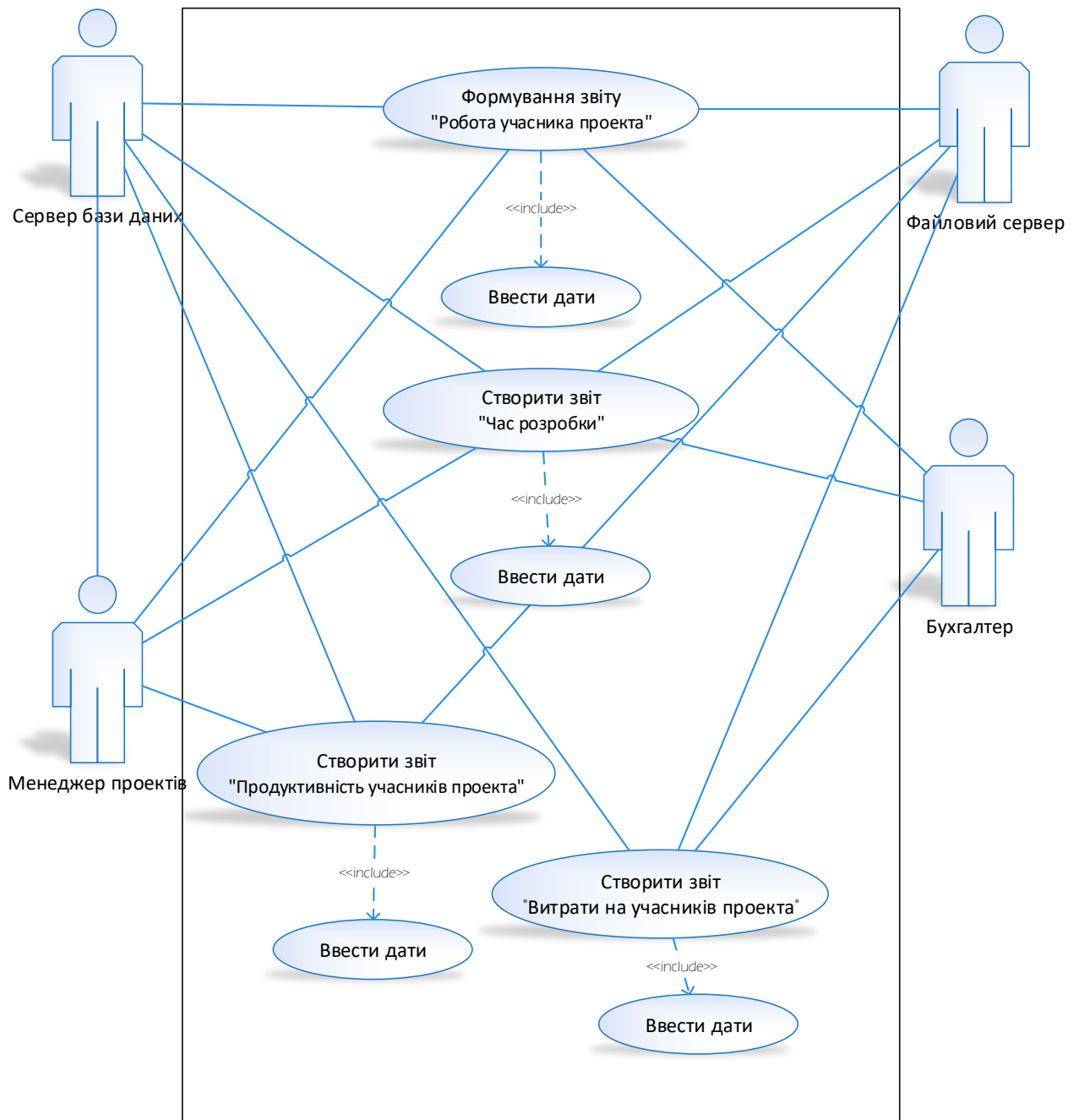


Рисунок 2.6 - Діаграма декомпозиції варіанта використання “Створювати звіти”

Зм.	Аркуш	№ документа	Підпис	Дата

Декомпозиція варіанта використання “Створювати показники продуктивності і зайнятості” представлена на рисунку 2.7.



Рисунок 2.7 - Діаграма декомпозиції варіанта використання “Створювати показники продуктивності і зайнятості”

### 2.1.2. Діаграма діяльності

На основі представлених в попередньому підрозділі діаграм використання розробимо діаграму діяльності.

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація

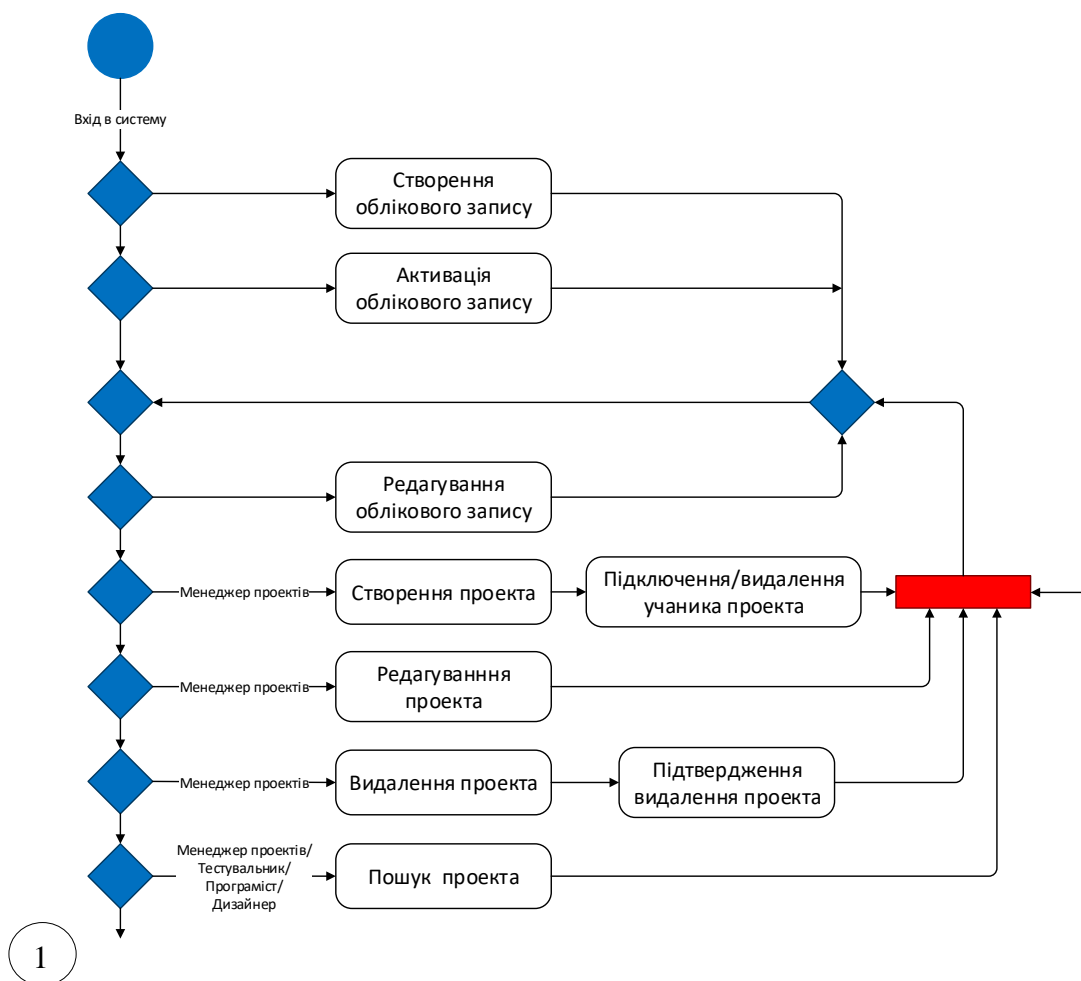
діяльності може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграми активностей будуються з обмеженої кількості фігур, з'єднаних стрілочками [23]. Найважливіші типи фігур:

- скруглені прямокутники позначають дії;
- ромби позначають рішення;
- синій кружок позначає старт (початковий стан) процесу;
- зелений кружок в колі позначає кінець (кінцевий стан).

Стрілки ведуть від старту до кінця і позначають порядок в якому відбуваються активності.

На основі представленої постановки задачі розробимо діаграму діяльності. Діаграма діяльності представлена на рисунку 2.8.



Зм.	Аркуш	№ документа	Підпис	Дата
-----	-------	-------------	--------	------

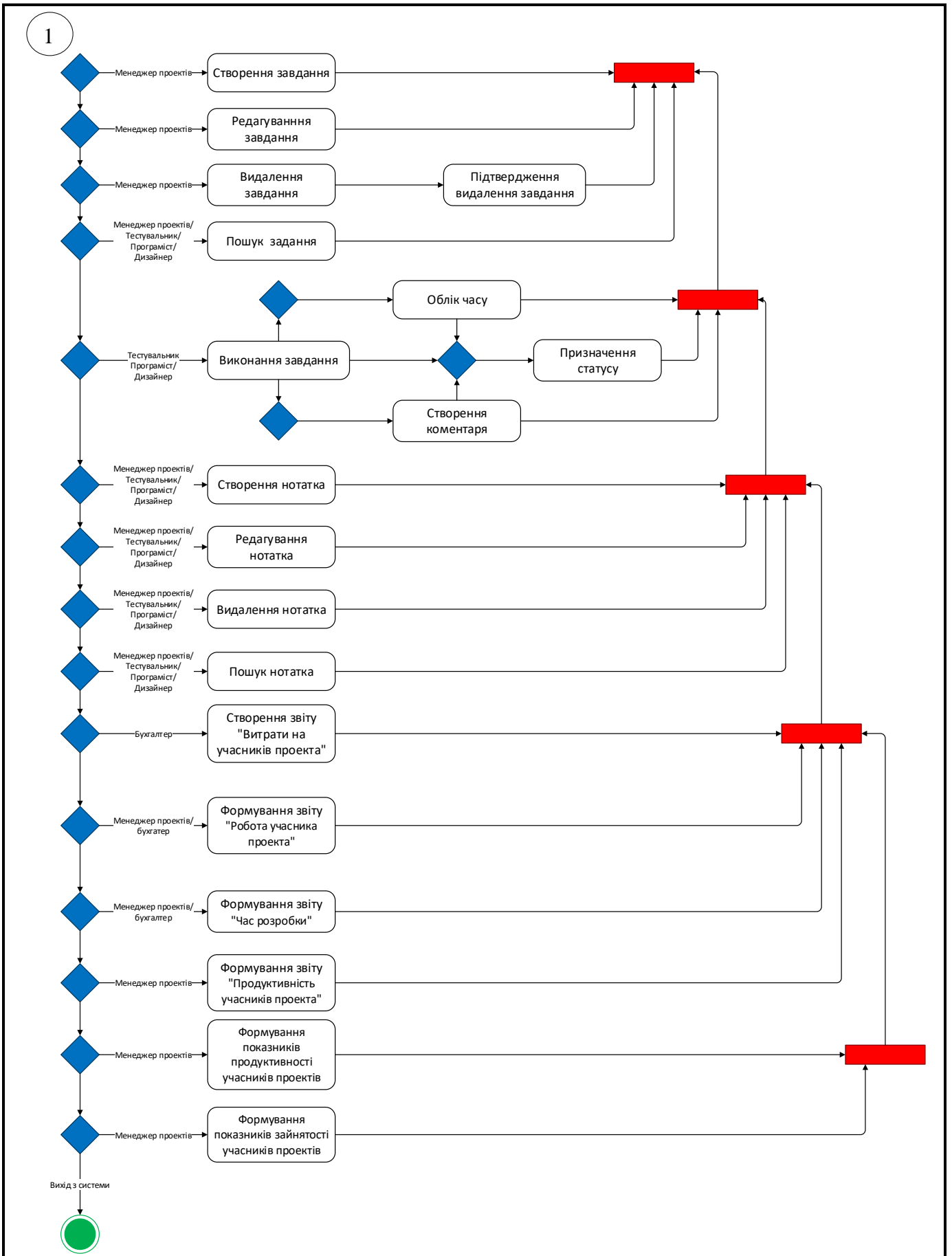


Рисунок 2.8 - Діаграма діяльності

Зм.	Аркуш	№ документа	Підпис	Дата

122.4142.08.ПЗ.Р2

Аркуш

53

## 2.2. Рішення з інформаційного забезпечення автоматизованої інформаційної системи

### 2.2.1. Концептуальна та логічна моделі бази даних

На основі причинно-наслідкових зв'язків, які властиві інформаційній системі було розроблено концептуальну модель бази даних, яка представлена на рисунку 2.9. Ця модель створюється для подальшого проектування бази даних. На концептуальній моделі у візуально зручному виді описуються зв'язки між об'єктами даних та їх характеристики.

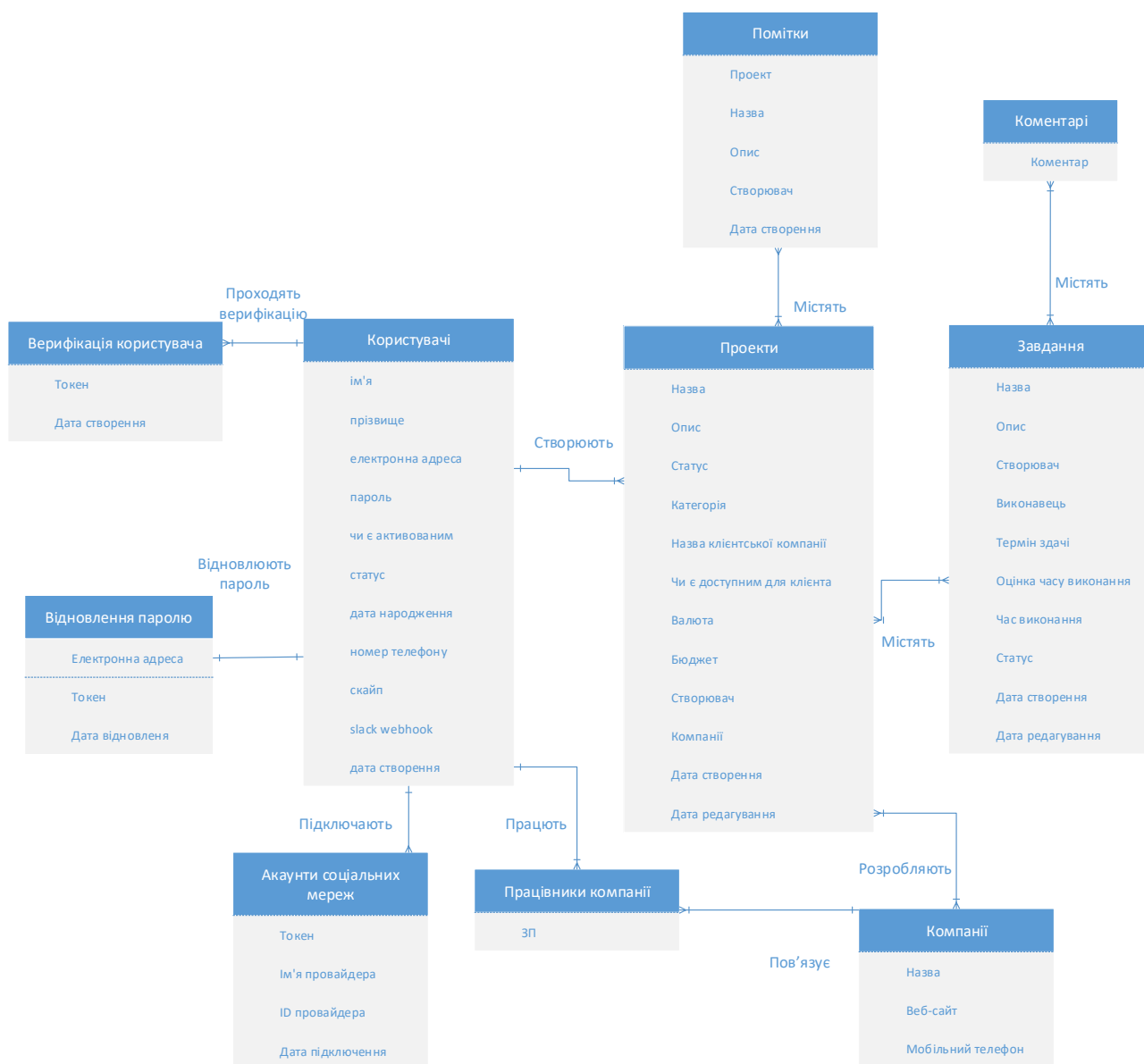


Рисунок 2.9 - Концептуальна модель бази даних

На основі представленої концептуальної моделі було спроектовано логічну модель даних досліджуваної предметної області. Ця модель виражена незалежно від конкретного продукту керування базами даних або технології збереження, але в термінах структур даних, таких як реляційні таблиці та колонки. Модель представлена на рисунку 2.10.



Рисунок 2.10 - Логічна модель бази даних

Представлена логічна моделі даних подає абстрактну структуру області інформації. Вона має схематичний характер і захоплює речі, що мають важливе значення, та опису, як вони відносяться одна до одної. Логічна модель даних є основою майбутньої фізичної моделі даних.

Згідно моделі, яка була представлена на рисунку 2.10, база даних містить наступні сутності:

- 1) Users – сутність, яка описує користувача, який був зареєстрований в системі. Описує персональні дані користувача.
- 2) User Verify – сутність, яка описує верифікацію користувача після його реєстрації в системі. Верифікація електронної адреси є обов'язковою умовою для повноцінного використання системи.
- 3) Password Resets – сутність, яка описує відновлення паролю зареєстрованого користувача у випадку його втрати або дізнання третіми особами.
- 4) User Connect At Account – сутність, яка описує підключені соціальні мережі користувача для того, щоб за допомогою цих даних можна було авторизуватися в системі.
- 5) Companies – сутність, яка описує компанії. Компанія створюються при створенні першого власного проекту користувача.
- 6) Users Company – сутність, яка описує, які користувачі є співробітниками компаній зареєстрованих в системі.
- 7) Projects – сутність, яка описує проекти, які були створені в системі для управління.
- 8) Project Members – сутність, яка описує, які зареєстровані в системі користувачі приймають участь в проекті та в якому саме і їхню посаду.
- 9) Project Members Roles – сутність описує посади користувачів, які працюють над реалізацією проекту. Сутність містить дані відразу при інсталяції системи.
- 10) Currencies – сутність, яка описує валюти, яку можна обрати при створенні проекту.



- 11) Stages – сутність, яка описує стани, які може приймати проект, який знаходиться на обліку системи.
- 12) Categories – сутність, яка описує категорію до якої може відноситися проект, який знаходиться на обліку системи.
- 13) Tasks – сутність описує виконувани завдання користувачами, які працюють над проектом.
- 14) Task Members – сутність, яка описує, які підключені до проекту користувачі були підключені до завдання.
- 15) Comments – сутність, яка описує коментарі, які створюються в завданні під час його виконання.
- 16) Notes - сутність, яка описує нотатки, які були створені користувачем у зареєстрованому в системі проекті.

### **2.2.2. Аналіз нормалізації сутностей бази даних**

Нормалізація схеми бази даних – покроковий процес розбиття одного відношення (сутності) відповідно до алгоритму нормалізації на декілька відношень на базі функціональних залежностей.

Нормальна форма – властивість відношення в реляційній моделі даних, що характеризує його з точки зору надмірності, яка потенційно може призвести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення.

Таким чином, схема реляційної бази даних переходить у першу, другу, третю і так далі нормальні форми. Якщо відношення відповідає критеріям нормальної форми  $n$  та всіх попередніх нормальних форм, тоді вважається, що це відношення знаходиться у нормальній формі рівня  $n$ .

На основі представлених концептуальної та логічної моделей бази даних, можна сказати, що структура сутностей відповідає наступним нормальним формам:

1) Перша нормальна форма. Схема бази даних відповідає наступним вимогам:

- Кожна сутність має основний ключ.
- Відсутнє повторення груп, що правильно визначає неключові атрибути.
- Атомарність: кожен атрибут має лише одне значення, а не множину значення.

2) Друга нормальна форма. Схема бази даних відповідає наступним вимогам:

- Всі відношення у БД знаходяться у першій нормальній формі.
- Дані, що повторно з'являються в декількох рядках, винесені в окремі сутності.

3) Третя нормальна форма. Схема бази даних відповідає наступним вимогам:

- Всі відношення у БД знаходяться у другій нормальній формі.
- Дані в сутностях залежать винятково від основного ключа.

Отже, структура бази даних відповідає третій нормальній формі, якої достатньо для нормального функціонування бази даних.

### **2.3. Рішення з програмного забезпечення автоматизованої інформаційної системи**

#### **2.3.1. Вибір інструментальних засобів**

Операційна система – це базовий комплекс програм, що виконує керування апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем.

Вибір операційної системи для реалізації системи є важливим етапом розробки, яким не можна нехтувати, адже від цього залежить які інструменти будуть доступні, які мінімальні системні вимоги, якого рівня безпеки можна досягти, яка буде стабільність роботи, який рівень підтримки, доступність самої ОС та звичайно зручність використання.

					122.4142.08.ПЗ.Р2	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>58</b>

Вибір здійснювався серед трьох основних операційних систем: MS Windows, GNU Linux, Mac OS. Коротко розглянемо кожен з трьох операційних систем, їхні основні переваги та недоліки.

MS Windows – сімейство комерційних операційних систем корпорації Microsoft орієнтовані на застосування графічного інтерфейсу при управлінні.

Переваги MS Windows:

- Сумісність. Практично будь-яка програма або працює, або має аналоги під Windows. Драйвери є для будь-яких пристроїв.
- Підтримка. Компанія Microsoft пропонує відмінну підтримку, тому програми легко встановлювати. А ось, наприклад, з Linux подібної підтримки не дістати.
- Доступність. Багато корисних функцій. Інтерфейс досить дружній, зрозумілий на інтуїтивному рівні, в ньому зможе розібратися навіть новачок.

Недоліки MS Windows:

- Віруси. Більшість вірусів націлені на Windows.
- Швидкість. Windows потребує багато ресурсів. У ньому маса зайвого для звичайного користувача і не професіонал не зможе розібратися, що зайве, а що ні.
- Ціна. Після кожного великого релізу доведеться купувати ОС заново. Винятком стала тільки 10 версія з тимчасовим безкоштовним оновленням.

Mac OS – сімейство операційних систем виробництва корпорації Apple.

Переваги Mac OS:

- Віруси. У цій ОС їх практично немає.
- Надійність. Система була написана під конкретні «залізні» набори (конфігурації комп'ютерів). Виходячи з цього, функціонує система так, як повинна і рідко ламається.

- Оптимізація ПО. Додатків існує не багато, але вони пишуться для ідеальної сумісності з системою і в підсумку Mac OS працює стабільно набагато довше, ніж інші, при цьому ресурси витрачаються раціонально.
- Дизайн. Зовні багато пристроїв Apple виграно виділяються серед конкурентів.

Недоліки Mac OS:

- Ціна. Mac є дорогим в співвідношенні ціна-залізо. Однак це компенсується відмінною якістю.
- Внутрішня начинка. Apple використовує власні стандарти і нав'язує їх користувачам. Як результат - змінюючи пристрій, користувач повинен звикати до особливостей використання.
- Фізичний комп'ютер. Mac OS розрахована на конкретні конфігурації, і тому вкрай не рекомендується встановлювати її на комп'ютер не з сімейства Apple.
- Сумісність. Кількість програм, написаних під Mac, сильно поступається кількості програм під Windows.

GNU Linux – це сімейство ОС на основі однойменного ядра Linux. Для цієї операційної системи досить складно дати однозначну відповідь, оскільки існує багато дистрибутивів GNU Linux, і вони, часом, сильно відрізняються. Якщо в загальних рисах описувати характеристики, тоді можна виділити наступне:

Переваги GNU Linux:

- Малі технічні вимоги. Цій системі досить одного ядра процесора і від 256 до 512 МБ оперативної пам'яті для того, щоб працювати.
- Ціна. Багато дистрибутивів безкоштовні. Їх можна встановити і внести свою лепту в розробку. Повністю модифікувати систему під себе.
- Різноманітність. Велика кількість версій. Linux - тільки ядро і щоб його використовувати, необхідно багато додаткового софту, тому існує близько сотні різних дистрибутивів. Найпопулярнішими є Ubuntu, Fedora, CentOS і Mint.

										122.4142.08.ПЗ.Р2	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата							60

- Просте застосування. Деякі дистрибутиви досить прості у використанні, є спеціальні з графічним інтерфейсом, наприклад Ubuntu чи Mint.
- Віруси. Їх також, як і в Mac, практично немає.

Недолілки GNU Linux:

- Сумісність. Багато ПЗ з відкритим кодом, де можна вносити свої зміни. Це вимагає багато часу і зусиль. Бувають моменти, коли драйверів для ваших пристроїв не існує, і їх потрібно або писати самому з нуля, або використовувати щось підходяще під всі типи даного пристрою.

Функціональність пристроїв від цього страждає.

GNU Linux підтримує практично всі основні мови програмування (Python, C / C ++, Java, Perl, Ruby і т.д.). Крім того, він пропонує широкий спектр програм, що використовуються для програмування. Термінал GNU Linux перевершує в можливостях командний рядок. Можна легко знайти багато бібліотек розроблених спочатку для Linux. Менеджер пакетів в GNU Linux допомагає встановити більшість програм легко і просто. GNU Linux має вбудовану підтримку SSH, який допоможе легко управляти серверами.

З вищеприписаного можна зробити висновок, що операційна система Linux має велику кількість переваг, порівняно з іншими ОС, особливо, як система для програмування. Тому для реалізації інформаційної системи була обрана ОС GNU Linux, а саме дистрибутив Linux Mint.

В ролі ППЗ було обрано набір вільного програмного забезпечення з відкритим кодом (LAMP), в який входять веб-сервер Apache, СУБД MySQL, та інтерпретатор PHP. Такий набір програм було обрано через свою гнучкість, продуктивність та низьку вартості (всі її складові є відкритими і можуть бути безкоштовно завантажені з Інтернету).

Apache HTTP-сервер – відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation.

					122.4142.08.ПЗ.Р2	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		61

До основних переваг Apache відносяться:

- Безкоштовний навіть для використання в комерційних цілях.
- Надійний, стабільне програмне забезпечення.
- Часто оновлюваний, регулярні патчі безпеки.
- Гнучкий завдяки своїй модульній структурі.
- Легко налаштовується.
- Багатоплатформовий (працює однаково добре на Unix і на Windows серверах).
- Велике співтовариство і легко доступна підтримка в разі будь-якої проблеми.

PHP - це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом. До головних переваг PHP входять:

- Висока швидкість роботи і, відповідно, загальна продуктивність ресурсів.
- Бюджетність, економічність.
- Простота освоєння, простий синтаксис.
- Відмінна сумісність і переносимість - php-коди працюють однаково добре з різними платформами.
- Висока гнучкість, ємність і функціональність.
- Багатозадачність і широкі можливості - створення будь-яких веб-додатків, блогів, гостьових книг, інтернет-магазинів, сайтів, робота з редирект, заголовками, pdf-документами, базами даних, електронною поштою тощо.

MySQL – компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання. Використанню СУБД MySQL дозволило: швидко встановити та почати реалізацією системи; підтримувати необмежену кількість користувачів, що одночасно працюють із БД; досягти високої швидкості виконання команд; забезпечити необхідний рівень безпеки.

Для реалізації веб-інтерфейсу було обрано JavaScript-овий фреймворк Vue.js, що використовує шаблон MVVM для створення інтерфейсів

									Аркуш
									62
Зм.	Аркуш	№ документа	Підпис	Дата					

користувача на основі моделей даних, через реактивне зв'язування даних. Головними перевагами даного засобу розробки є:

- Детальна документація. Vue.js має дуже хорошу документацію, яка може збільшити швидкість навчання і заощадити багато часу на розробку програми з використанням базових знань HTML і JavaScript.
- Адаптивність. Vue.js забезпечує швидкий період переходу від інших фреймворків.
- Чудова інтеграція. Vue.js можна використовувати як для створення односторінкових додатків, так і для більш складних веб-інтерфейсів додатків. Найважливіше, що невеликі інтерактивні частини можна легко інтегрувати в існуючу інфраструктуру, не надаючи при цьому негативного впливу на всю систему.
- Велике масштабування. Vue.js допомагає розробляти досить великі шаблони для багаторазового використання.
- Малий розмір. Vue.js може важити близько 20 КБ і при цьому зберігати свою швидкість і гнучкість, що дозволяє досягти набагато більш високої продуктивності, в порівнянні з іншими фреймворками.

В ролі програмного інструмента для реалізації системи було обрано PHPStorm. Цей інструмент є зручним та володіє розширеним функціоналом: відмінна інтеграція git, наявність Scratch файлів, які не потрібно зберігати в окремому місці, зручний пошук, як по всьому проекту так і по його частинам, зручні підказки, через що розробка ведеться швидше, можливість встановлювати різноманітні пакети, які полегшать розробку та інше.

### 2.3.2. Структура програмного забезпечення

На основі функціональної моделі, розроблених діаграм діяльності та представлених сутностей розробимо діаграму класів. Діаграма класів – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних та відношення.

					122.4142.08.ПЗ.Р2	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		63

Діаграма основних класів системи представлена на рисунку 2.11.

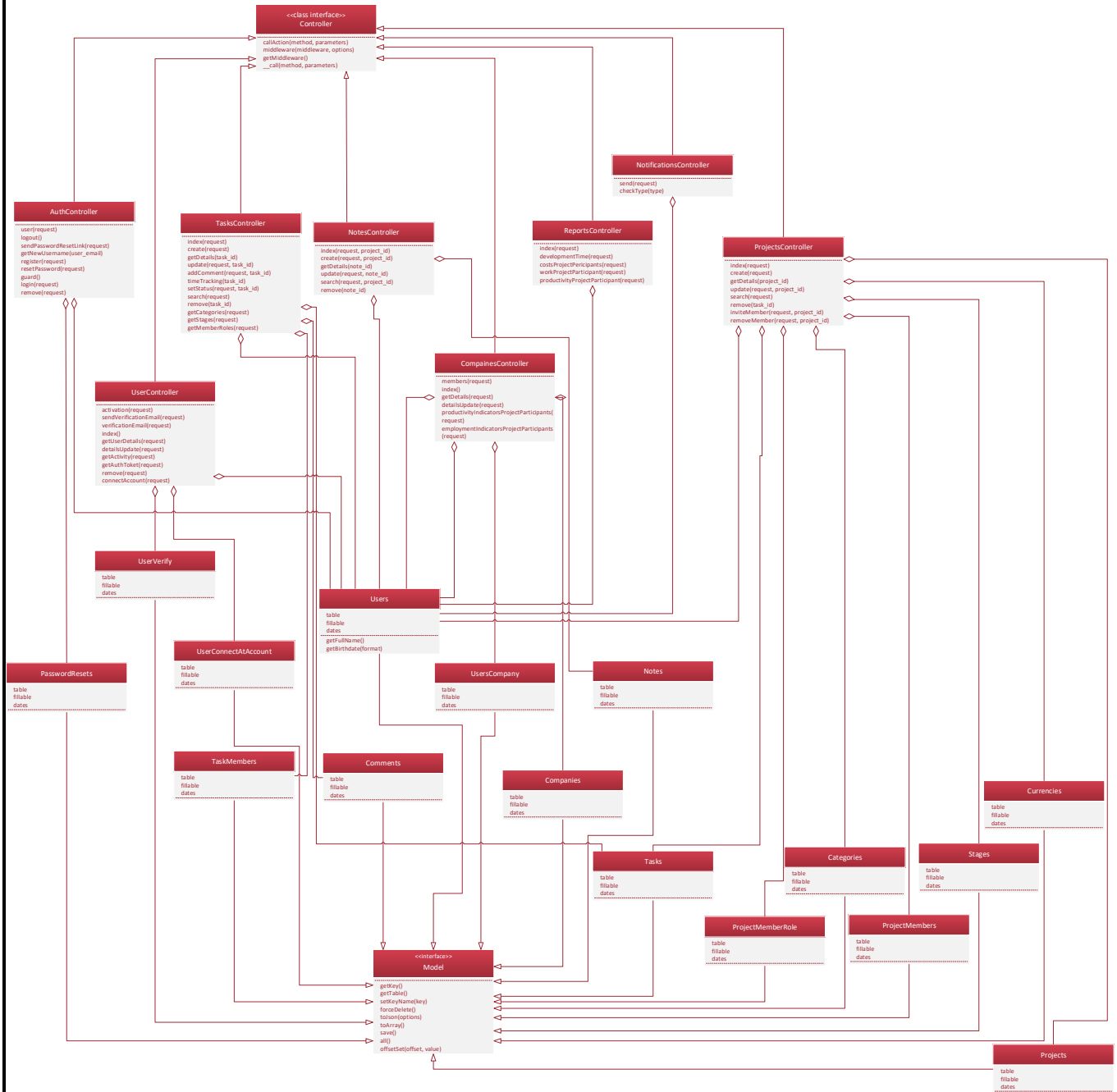


Рисунок 2.11 - Діаграма класів

### 2.3.3. Фізична модель бази даних

Фізична модель даних – подання дизайну даних, як реалізованого чи призначеного для реалізації у системі керування базами даних. Завершена фізична модель даних включає всі артефакти бази даних, необхідні для



створення відношень між таблицями чи для досягнення мети продуктивності, як-от індексів, визначень обмежень і зв'язаних таблиць або кластерів. Для реалізації моделі бази даних була обрана СКБД MySQL [5].

Робота с базою даних відбувається за допомогою PHP Data Object (PDO). Цей клас надає методи для зручної роботи з об'єктами та prepared statements, який значно підвищить продуктивність. PDO підтримує 12 різних драйверів для баз даних, включаючи MySQL, яка і була використана.

Фізична модель бази даних представлена на рисунку 2.12.

#### **2.3.4. Методи і засоби розробки програмного забезпечення**

При постановці вимог, яким повинна відповідати система та проектуванні було застосовано об'єктно-орієнтований підхід, тому реалізація системи повинна дотримуватися саме цього підходу. Такий підхід має низку переваг і головні можна виділити наступні:

- Класи дозволяють проводити конструювання з корисних компонент, що володіють простими інструментами, що дає можливість абстрагуватися від деталей реалізації.
- Дані та операції разом утворюють певну сутність і вони не «розмазуються» по всій програмі, як це нерідко буває в разі процедурного програмування.
- Локалізація коду і даних покращує наочність і зручність супроводу програмного забезпечення.
- Інкапсуляція інформації захищає найбільш критичні дані від несанкціонованого доступу.
- Уникнення дублювання коду. Це дозволяє уникати помилок в кодї, інтерфейс стає більш уніфіковані.

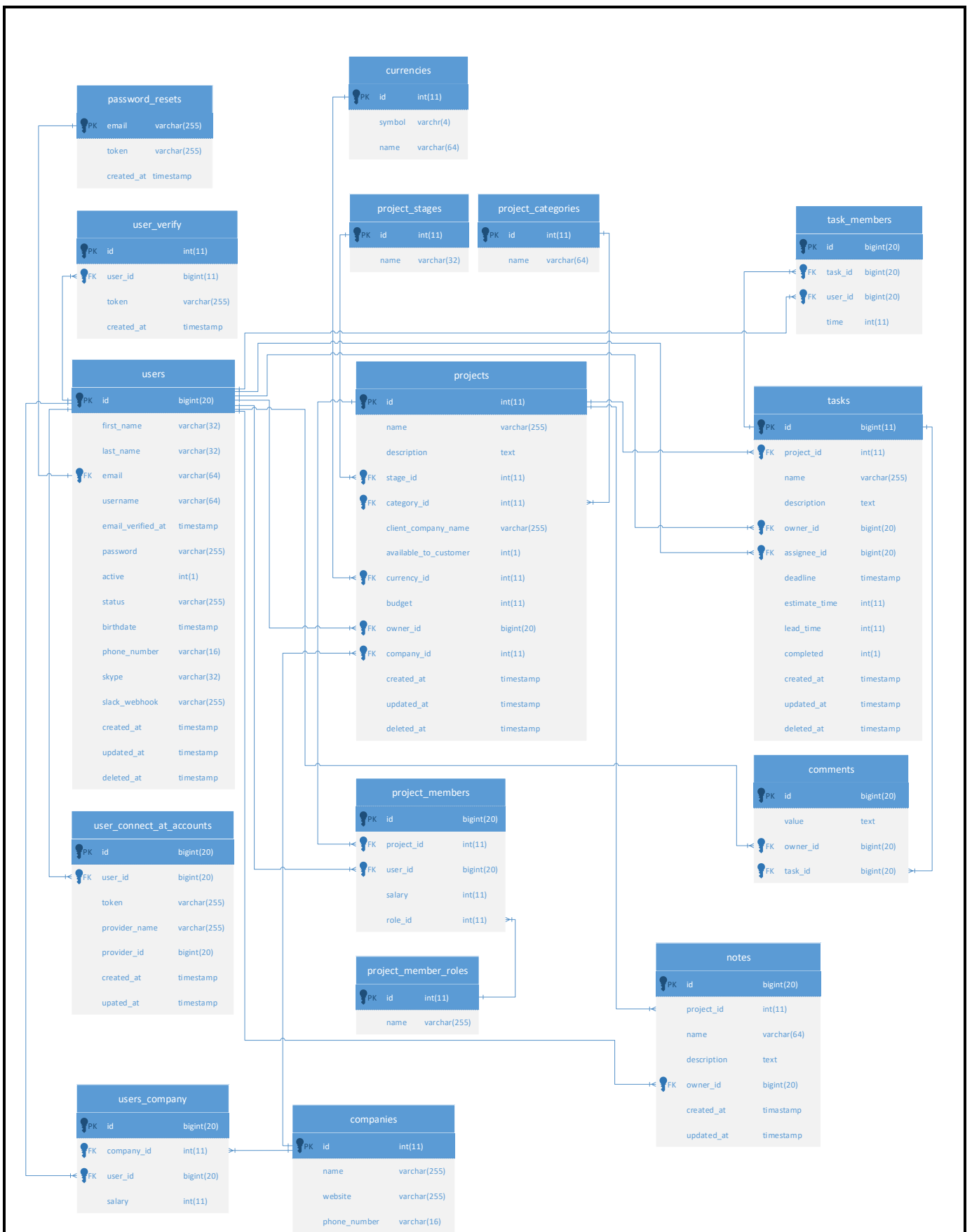


Рисунок 2.12 – Фізична модель бази даних

Засобом розробки програмного забезпечення серверної частини системи є платформи незалежна мова програмування PHP версії 7.2. Крім цього використовуються додаткові засоби, які полегшать та пришвидшать реалізацію функціонала: Laravel Framework 6.2, fideloper proxy, guzzlehttp guzzle, laravel ui, laravel jwt-auth, fzaninotto faker, Carbon. Структура програми серверної частини повинна представлятися у вигляді окремих пакетів, ка з легкістю можна буде переносити на системи з подібним функціоналом, тому їх можна виділити, як окремий засіб для розробки.

В клієнтській частині засобом розробки програмного забезпечення є JavaScript - динамічна, об'єктно-орієнтована прототипна мова програмування, використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Крім чистого JavaScript використовуються наступні засоби: Axios, jQuery, Popper.js, Vue.js, Vuex, Vue-resources, Vue-router, Vue-templae-compiler, Vuetify, Vue-material-design-icons, Vue-auth, Bootstrap 4. Система представлена у вигляді односторінкового застосунку, тобто вся робота відбувається без перезавантаження сторінки.

#### **2.4. Розробка інтерфейсів автоматизованої інформаційної системи**

На основі моделі прецедентів, яка визначає перелік задач та на основі інформаційних моделей, які визначають перелік сутностей розробимо моделі інтерфейсів.

У кожного зареєстрованого користувача є персональна сторінка на якій відображаються його персональні дані та додаткові вкладки, які були описані при постановці задачі. Схематично персональна сторінка користувача представлена на рисунку 2.13.

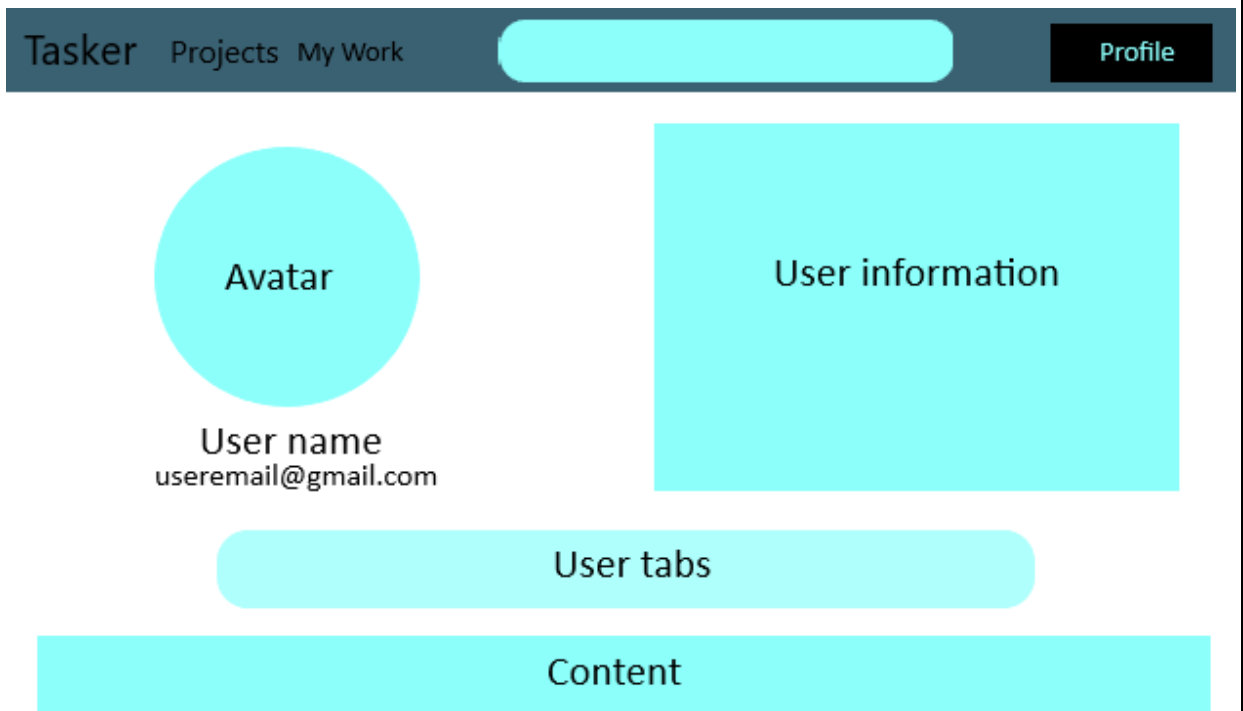


Рисунок 2.13 – Персональна сторінка користувача

Користувач має можливість переглядати власні проекти та ті в яких він приймає участь. Схематично така сторінка представлена на рисунку 2.14.



Рисунок 2.14 – Схематичне зображення сторінки з проектами

Кожен проект має головну сторінку, яка містить наступні вкладки: Завдання, Нотатки, Час, Адміністрування. На головній сторінці відображається список всіх завдань с позначкою виконавця та згруповані по створеному списку завдань. На цій же сторінці є можливість створити список завдань та самі завдання. Схематично головна сторінка проекту представлена на рисунку 2.15.

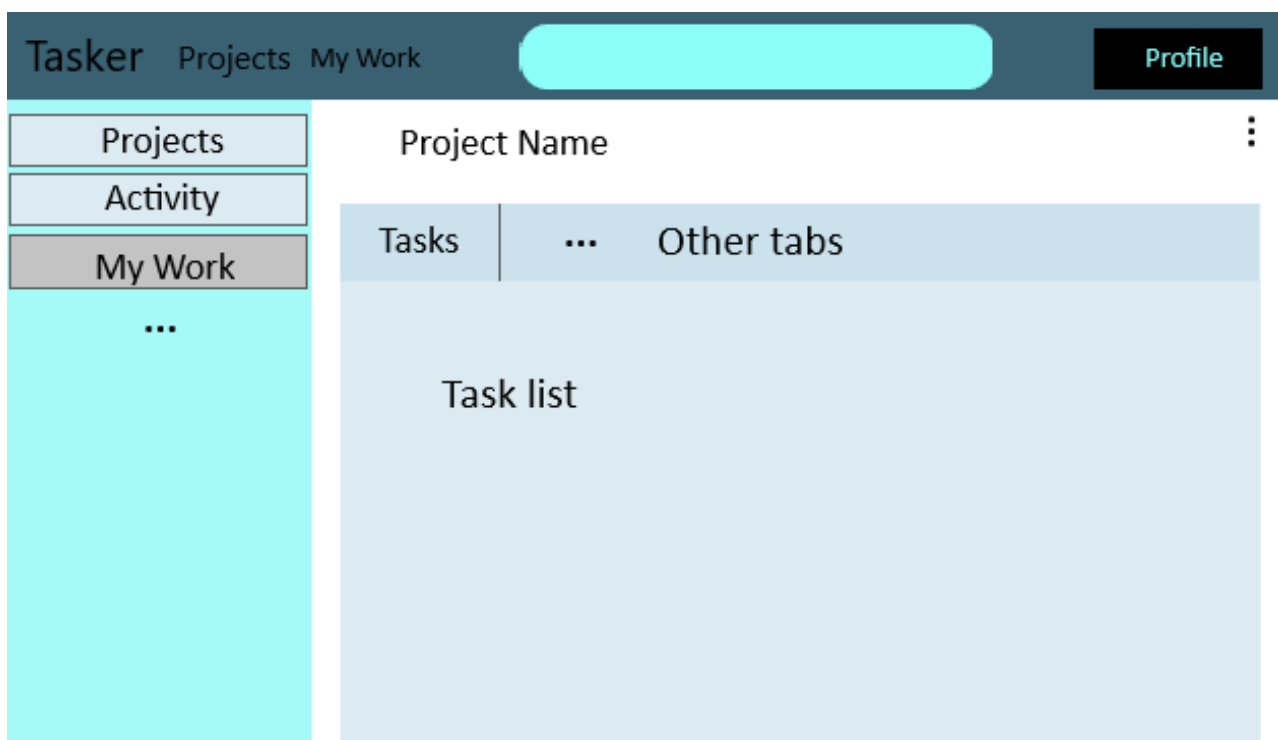


Рисунок 2.15 – Схематичне зображення головної сторінки проекту

Сторінка підключення та видалення учасників проекту і клієнтів представлена на рисунку 2.16.

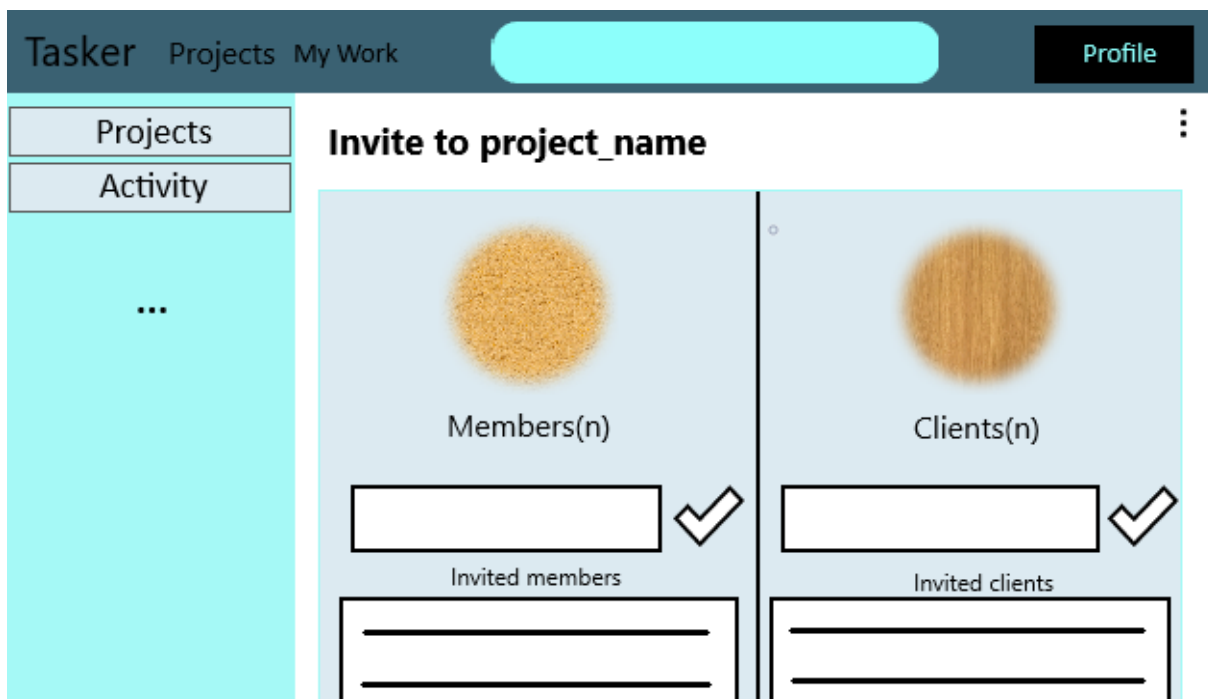


Рисунок 2.16 – Схематичне зображення сторінки підключення та видалення учасників проекту та клієнтів

## 2.5. Рішення з технічного забезпечення автоматизованої інформаційної системи

На основі рішень з програмного забезпечення представимо вимоги до технічного забезпечення.

Оскільки в ролі ОС для реалізації системи було обрано OCLinux, а саме дистрибутив Linux mint, то мінімальні системні вимоги є наступними:

- x86-процесор (для Linux Mint 64-bit потрібно 64-бітний процесор. Linux Mint 32-bit працює і на 32-бітному, і на 64-бітному процесорах);
- ОЗУ: 512 МБ (Cinnamon, XFCE, MATE);
- 9 ГБ вільного місця на жорсткому диску;
- відеокарта: будь-яка, для якої є драйвер;
- привід CD-ROM або порт USB.

Рекомендованими системними вимогами для функціонування Linux mint є:

- x86-процесор (для Linux Mint 64-bit потрібно 64-bit процесор. Linux Mint 32-bit працює і на 32-bit, і на 64-bit процесорах);

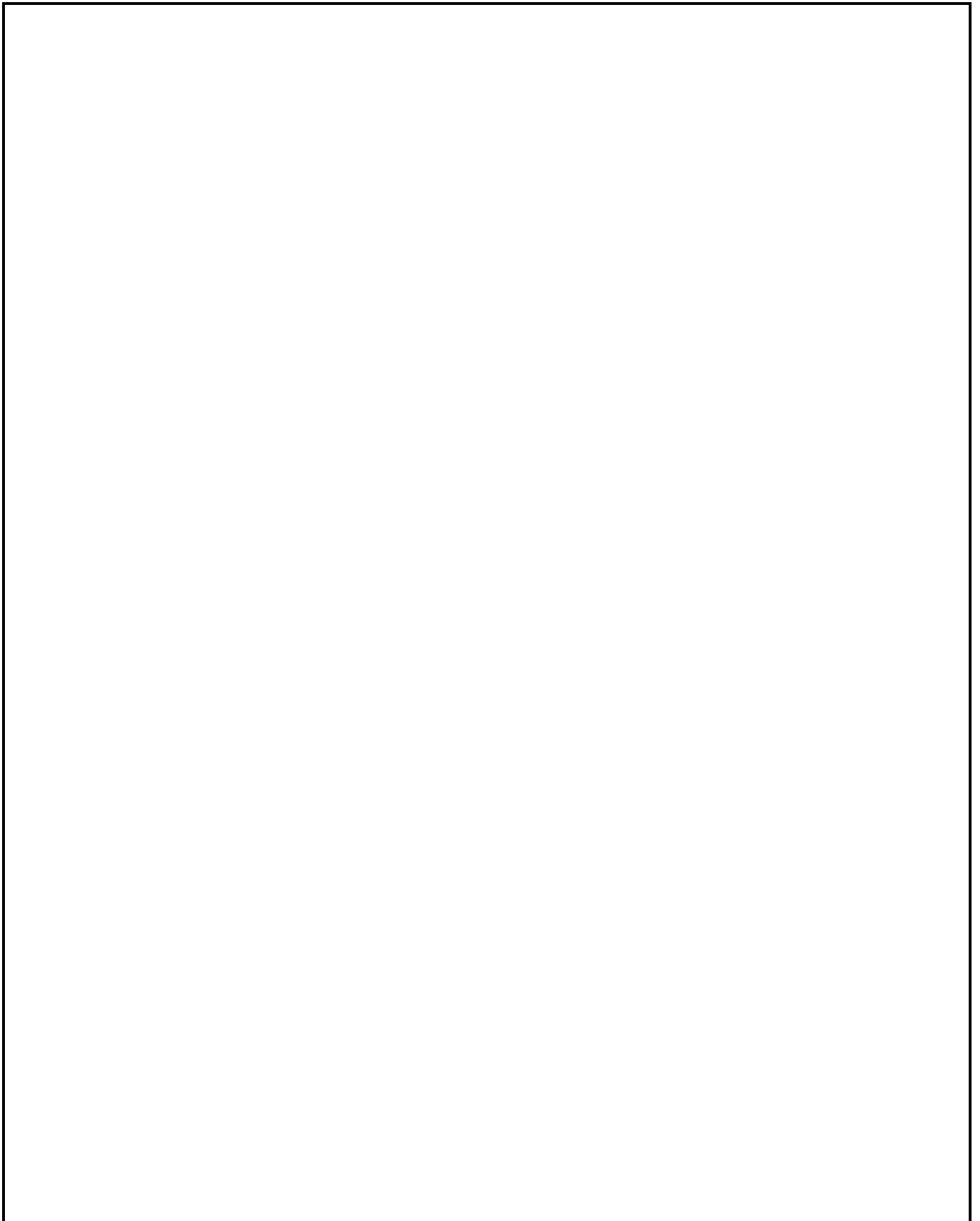
- ОЗУ: 1 ГБ (XFCE, MATE), 2 ГБ (Cinnamon);
- 20 ГБ вільного місця на жорсткому диску;
- відеокарта: 1024 × 768;
- привід DVD-ROM або порт USB.

Оскільки при реалізації системи будуть використовуватися бібліотеки, які необхідно завантажити в програму з віддалених репозиторіїв, а також необхідний доступ до документації по програмних інструментах, то обов'язковою вимогою до технічного забезпечення є наявність інтернет з'єднання.

## 2.6. Висновки

В результаті розробки технічного проекту було спроектовано діаграми використання та діяльності, які наглядно відображають, які функції необхідно реалізувати в системі та хто буде взаємодіяти з ними. Розроблено концептуальну модель бази даних, яка необхідна для опису зв'язків між об'єктами даних та їх характеристиками та логічну модель бази даних, яка стала основою для моделювання фізичної моделі бази даних, яка є остаточним варіантом представлення структури бази даних. Також були описані загальносистемні рішення, рішення з програмного та технічного забезпечення, проведено розробку інтерфейсів та діаграми розгортання системи. На основі даного розділу буде реалізовано автоматизовану інформаційну систему управління проектами ІТ-компанії.

					122.4142.08.ПЗ.Р2	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		71



					122.4142.08.ПЗ.РЗ			
Зм.	Аркаш	№ документа	Підпис	Дата				
Студент	Михалко М.І.				Розгортання автоматизованої інформаційної системи управління проектами ІТ- компанії	Літ.	Аркуш	Аркушів
Керівник	Гайда А.Ю.						72	3
Зав. каф.	Михелєв І.Л.					НУК		



### **3. РОЗГОРТАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ ІТ-КОМПАНІЇ**

#### **3.1. Розробка документації системи**

В результаті розробки системи було реалізовано інформаційну систему управління проектами ІТ-компанії. Для загального розуміння призначення системи було розроблено загальний опис системи, який представлений в додатку Б.

Для ознайомлення з функціоналом автоматизованої інформаційної системи управління проектами ІТ-компанії було розроблено інструкція з використання системи, яка представлена в додатку В. Інструкція надає детальний опис по функціоналу системи.

Ознайомитися зі структурою програмного забезпечення реалізованої інформаційної системи можна в додатку Г. В додатку описані всі програмні інструменти, які було використано в процесі реалізації інформаційної системи.

#### **3.2. Розгортання системи**

На основі розроблених проектних рішень розробимо діаграму розгортання. Діаграма розгортання – це модель, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Діаграма розгортання системи зображена на рисунку 3.1.

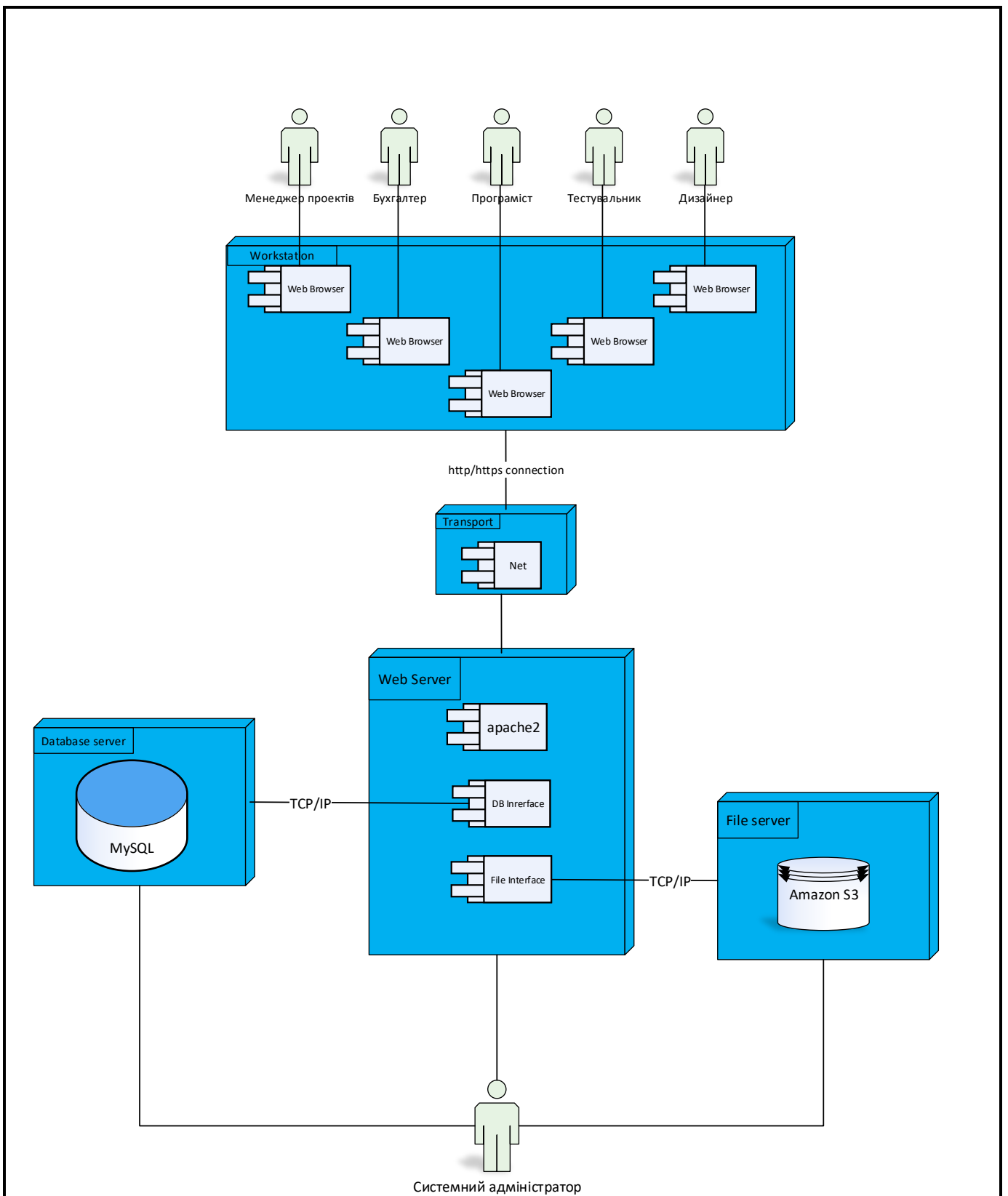
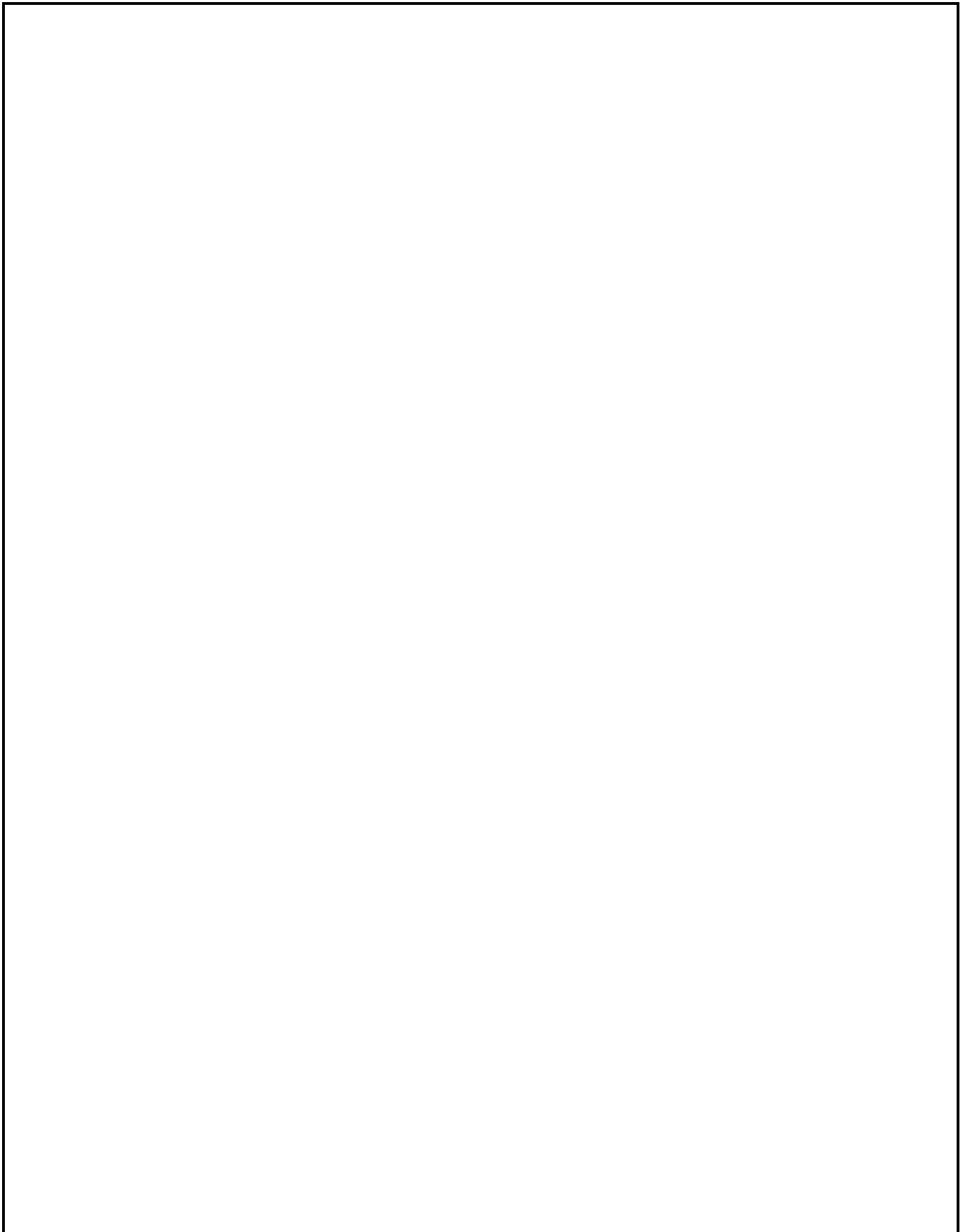


Рисунок 3.1 - Діаграма розгортання системи

Зм.	Аркуш	№ документа	Підпис	Дата



					122.4142.08.ПЗ.Р4			
Зм.	Аркаш	№ документа	Підпис	Дата				
Студент	Михалко М.І.				Охорона праці та техніка безпеки	Літ.	Аркуш	Аркушів
Керівник	Гайда А.Ю.						75	9
Зав. каф.	Михелєв І.Л.					НУК		

## 4. ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ

### 4.1. Вступ

Однією із характерних особливостей сучасного розвитку суспільства є зростання сфер діяльності людини, в яких використовуються інформаційні технології. Використання персональних комп'ютерів загострило проблеми збереження здоров'я, вимагає удосконалення існуючих та розробки нових підходів до організації робочих місць. Саме тому питання охорони праці при організації робіт посідають важливе місце і залишаються актуальними.

Якщо праця людини протікає у сприятливих умовах, вона сприяє розвитку всіх його здібностей, забезпечує широкі можливості для високопродуктивної і творчої роботи, сприяє зниженню аварійності та випадків виробничого травматизму. Саме тому охорона праці розглядається як одна з найважливіших економічних і соціальних задач не тільки окремого підприємства, але й держави в цілому.

Робоче приміщення та місце має відповідати вимогам щодо охорони праці при організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (ВДТ).

Критична ситуація в Україні у сфері безпеки праці проявляється високим рівнем виробничого травматизму і професійної захворюваності, незадовільними умовами праці та санітарним станом підприємств, внаслідок чого держава втрачає кваліфікованих працівників, а натомість отримує десятки тисяч осіб, які потребують повноцінного соціального захисту. За останні 5 років на виробництві загинуло майже 5 тисяч працівників і понад 85 тисяч – травмовано.

Домінуючими причинами формування несприятливих умов праці залишаються недосконалі технології, машини і механізми, їхня несправність, невикористання засобів захисту, порушення правил техніки безпеки, режимів праці і відпочинку.

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		76

## 4.2. Напруженість праці користувача ЕОМ

Робота програміста пов'язана з значним зоровим навантаженням, що вимагає забезпечення належного освітлення. В приміщенні рівень природного освітлення і штучного світла повинен відповідати нормам. Інженер-програміст працює з ЕОМ та іншим офісним обладнанням, що є джерелом небезпеки ураження електричним струмом. Трудова діяльність програміста пов'язана з постійним перебуванням в приміщенні, тому для комфортних умов праці необхідно створити належний мікроклімат.

Згідно нормативним документам [19] та [20] можна виділити такі шкідливі виробничі чинники, що зазвичай простежуються в робочих приміщеннях:

1. Недостатній рівень штучного освітлення.
2. Мікроклімат робочої зони: температура, відносна вологості, швидкість руху повітря.
3. Підвищений рівень шуму на робочому місці.
4. Небезпечна напруга в електричному ланцюзі.
5. Підвищений рівень вібрації.

### 4.2.1. Рівень штучного освітлення

Основним документом, який регламентує норми освітленості є [21]. Категорія виконуваних робіт програміста відноситься до робіт високої точності. Тому нормативне значення загального освітлення робочого приміщення повинно бути  $E = 300-500$  лк. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного числа світильників, їхнього типу і розміщення.

Встановлення нормального рівня штучного освітлення згідно всім нормам є важливим етапом в організації робочого приміщення для роботи за

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		77

комп'ютером, тому ні в якому разі не можна нехтувати або невідповідально ставитися до цього етапу.

#### 4.2.2. Мікроклімат робочого місця

Відповідно до [20] праця програміста за важкістю відноситься до легкої фізичної роботи. Основним документом, який регламентує норми мікроклімату робочої зони є [22]. Комп'ютери і офісна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні. В приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. В санітарних нормах встановлені величини параметрів мікроклімату, що створюють комфортні умови. Значення параметрів оптимальних та допустимих параметрів мікроклімату згідно з [22] для приміщень, та фактичних параметрів представленні в таблиці 4.1.

Таблиця 4.1 – Порівняння фактичних і нормативних характеристик температури, швидкості руху та відносної вологості повітря.

№	Мікроклімат у приміщенні	Фактичне значення	Нормативне значення
1	Температура повітря у теплий період року	23-25 °С	23-25 °С
2	Температура повітря у холодний період року	22-24 °С	22-24 °С
3	Швидкість руху повітря	0,1 м/сек	0,2-0,5м/сек
4	Відносна вологість повітря	60-40 %	60-40 %

#### 4.2.3. Шум та вібрації у робочому приміщенні

У приміщенні завжди присутні внутрішні джерела постійного шуму:

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		78

- вентилятори блоків ЕОМ;
- принтери;
- дисководи.

Зовнішніми джерелами шуму і вібрації в приміщенні є проїжджаючі транспортні засоби. Шум, створений усіма перерахованими джерелами, можна кваліфікувати як постійний. Наявність постійного шуму в робочій зоні приводить до розладу центральної нервової системи і до таких захворювань як неврози, тому фактичний рівень шуму в робочій зоні не повинний перевищувати 50 дБА.

У разі перевищення рівня шуму необхідна додаткова звукоізоляція. У якості звукоізолюючих матеріалів, які застосовують у конструкціях для зниження передачі структурного (ударного) звуку переважно використовують мати та плити із скляного та мінерального волокна, м'які плити з деревних стружок, картон, гуму, утеплений лінолеум, а також заміна вікон на звукоізолюючі.

#### 4.2.4. Оцінка електробезпеки

Головні норми, яким повинен відповідати стан електробезпеки в робочому приміщенні:

- Електропроводка повинна бути захована і ізольована від працівників спеціальним коробом.
- Кожне робоче місце з ЕОМ повинно бути обладнане окремими розетками по 220 В.
- Відносна вологість повітря – 60%, температура повітря 22-24 °С, струмопровідний пил і хімічно активні речовини в повітрі відсутні.
- Підлога: ізолююча – наприклад, лінолеум.
- Комп'ютери повинні мати захисне занулення (за допомогою окремого захисного нульового провідника).

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		79

При виконанні робіт по ремонту і обслуговуванню ЕОМ обслуговуючий персонал зобов'язаний керуватися "Правилами техніки безпеки при експлуатації електроустановок споживачами". До роботи не допускаються особи, які не пройшли навчання з техніки безпеки.

### **4.3. Ергономіка робочого місця**

Проектування робочих місць, забезпечених відео терміналами, відноситься до числа важливих проблем ергономічного проектування в області обчислювальної техніки.

Робоче місце і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця програміста повинні бути дотримані наступні основні умови: оптимальне розміщення устаткування, що входить до складу робочого місця і достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення. Ергономічними аспектами проектування відео термінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, характеристики робочого крісла, вимоги до поверхні робочого столу, можливість регулювання елементів робочого місця. Головними елементами робочого місця програміста є стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення програміста. Рациональне планування робочого місця передбачає чіткий порядок і постійність розміщення предметів, засобів праці і документації. Те, що потрібне для виконання робіт частіше, розташоване в зоні легкої досяжності робочого простору.

Моторне поле – простір робочого місця, в якому можуть здійснюватися рухові дії людини.

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		80



Максимальна зона досяжності рук - це частина моторного поля робочого місця, обмеженого дугами, описуваними максимально витягнутими руками при 80 русі їх в плечовому суглобі.

Оптимальна зона - частина моторного поля робочого місця, обмеженого дугами, описуваними передпліччями при русі в ліктьових суглобах з опорою в точці ліктя і з відносно нерухомим плечем. Схематичне зображення зони досягнення рук представлено на рисунку 4.1.

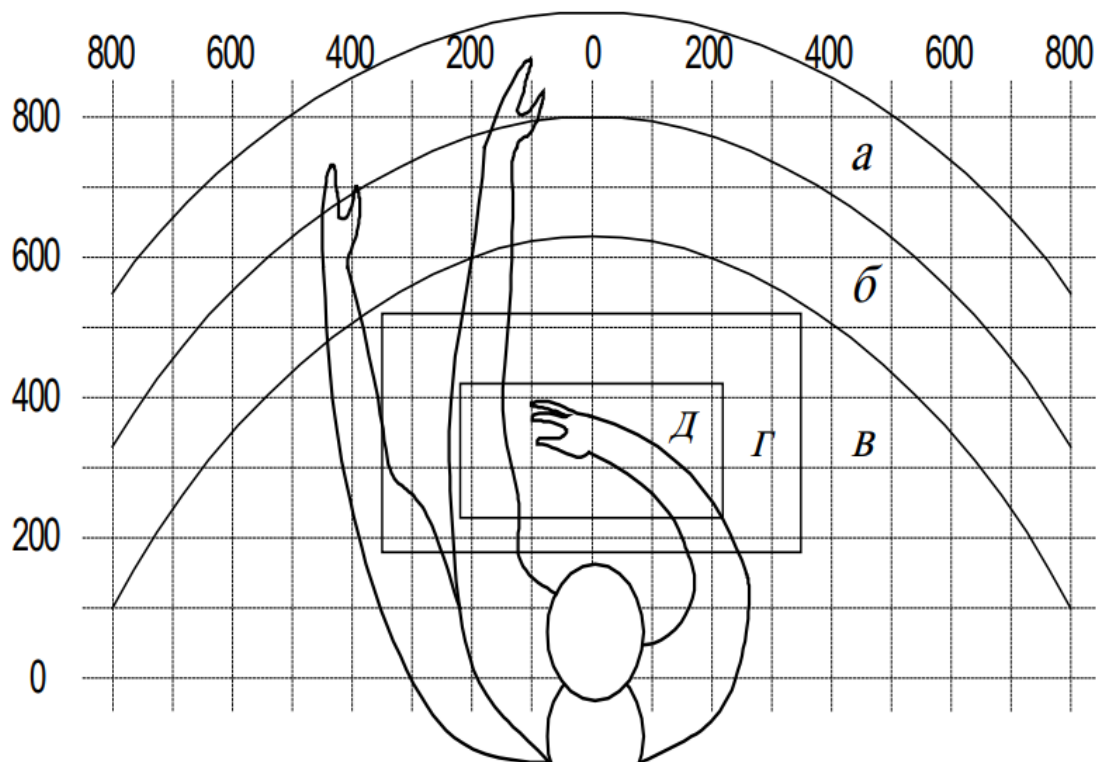


Рисунок 4.1 – Зони досягнення рук в горизонтальній площині

- а) зона максимальної досяжності;
- б) зона досяжності пальців при витягнутій руці;
- в) зона легкої досяжності долоні;
- г) оптимальний простір для грубої ручної роботи;
- д) оптимальний простір для тонкої ручної роботи.

Для комфортної роботи стіл повинен задовольняти наступним умовам :

Зм.	Аркуш	№ документа	Підпис	Дата

- 1) Висота столу повинна бути вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники.
- 2) Нижня частина столу повинна бути сконструйована так, щоб програміст міг зручно сидіти, не був вимушений підтискати ноги.
- 3) Поверхня столу повинна володіти властивостями, що виключають появу відблисків в полі зору програміста.
- 4) Конструкція столу повинна передбачати наявність висувних ящиків (не менше 3 для зберігання документації, лістингів, канцелярських обладнань).
- 5) Висота робочої поверхні рекомендується в межах 680-760 мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути біля 650 мм.

Велике значення надається характеристикам робочого крісла. Так, висота сидіння над рівнем підлоги, що рекомендується, знаходиться в межах 420-550 мм. Поверхня сидіння м'яка, передній край закруглює, а кут нахилу спинки - регульований.

Необхідно передбачати при проектуванні можливість різного розміщення документів: збоку від відео-терміналу, між монітором і клавіатурою і т.д. Крім того, у випадках, коли відео-термінал має низьку якість зображення, наприклад помітні мигтіння, відстань від очей до екрану роблять більше (біля 700мм), ніж відстань від ока до документа (300-450мм). Взагалі при високій якості зображення на відео-терміналі відстань від очей користувача до екрану, документа і клавіатури може бути рівним.

Положення екрану визначається:

- Відстанню прочитування (0,6 - 0,7 м).
- Кутом прочитування, напрямом погляду на 20° нижче горизонталі до центру екрану, причому екран перпендикулярний цьому напрямку.

Повинна також передбачатися можливість регулювання екрану:

- По висоті +3 см.
- По нахилу від -10 до +20 щодо вертикалі.
- В лівому і правому напрямках.

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		82

Під час користування комп'ютером медики радять встановлювати монітор на відстані 50-60 см від очей. Фахівці також вважають, що верхня частина відеодисплея повинна бути на рівні очей або трохи нижче.

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення її привабливості, що позитивно впливає на продуктивність праці.

					122.4142.08.ПЗ.Р4	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>83</b>

## Висновки

В результаті розробки автоматизованої інформаційної системи управління проектів ІТ-компанії було проведено аналіз існуючої системи управління у компанії та виявлено низку проблем: співробітникам доводиться постійно контролювати надходження нових проектів та створювані в них завдання; менеджеру проектів постійно необхідно переглядати всі невиконані завдання та в ручному режимі повідомляти про недотримання термінів здачі робіт; необхідність мати два бухгалтера, тому що використовується компанією система не вміє автоматично проводити розрахунки та формувати звіти та інші. Приймаючи до уваги всі виявлені проблеми були сформовані вимоги до нової системи управління проектами.

Розроблена концепція системи спрямована на покращення робочого процесу співробітників компанії, що призведе до підвищення ефективності роботи.

По розробленій концепції було розроблено технічний проект системи в якому детально описано вимоги до системи та засоби, якими цих вимог можна досягти. Описано структуру програмного забезпечення, процес роботи з системою та спроектовано моделі бази даних.

На основі даних проектування було розроблено програмний продукт, який відповідає всім поставленим вимогам. Цю систему можна впровадити в будь-яку компанію, яка має схожу з “FutureDev” організаційну структуру. Це підвищить продуктивність команди розробників проекту та дозволить компанії отримати більші прибутки.

Отже, можна зробити висновок, що було досягнуто поставленої цілі роботи. Після того, як система буде впроваджена в компанії, то вона зможе отримувати більше прибутків.

					122.4142.08.ПЗ.Висновки	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		84

### Список використаної літератури

1. Огірко О., Крап-Спісак Н. “ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ УПРАВЛІННЯ ПРОЕКТАМИ”. Огірко О., Крап-Спісак Н., 2016 - Львівський державний університет внутрішніх справ.
2. “Управління ІТ-проектами”. Тернопільський національний економічний університет. Тернопіль – 2015, Кафедра інформаційно-обчислювальних систем та управління.
3. Міжнародний стандарт ISO/IEC 12201-2010 Інформаційна технологія. Системна та програмна інженерія. Процеси життєвого циклу програмних засобів.
4. Алпатов Ю.Н. “Моделювання процесів та систем управління”. Навчальний посібник. – СПб.: Видавництво “Лань”, 2018. – 140с. (Посібники для вузів. Спеціальна література).
5. Хоменко А.Д., Циганков В.М., Мальцев М.Г. “Бази даних: Навчальний посібник для вищих навчальних закладів”. Під редакцією професора Хоменка А.Д. – 6-е видання. доп. – СПб.: КОРОНА-Вік. 2009 – 736с.
6. ГОСТ 19.003-80. “Схеми алгоритмів та програм. Позначення умовні графічні.”
7. Методичні вказівки до виконання дипломного проектування зі спеціальності 6.050101 “Комп'ютерні науки” кваліфікації “Бакалавр інформаційних управляючих систем і технологій” / Ажищев В. Ф., Кошкін К. В., Морозова Г. С. – Миколаїв : НУК, 2014. – 57 с.
8. Маслов А.В. “Проектування інформаційних систем в економіці”. Навчальний посібник/ Маслов А.В. – Томськ. Видавництво Томського політехнічного університету, 2008. – 216 с.
9. Вендров А.В. “CASE-технології. Сучасні методи і засоби проектування інформаційних систем”. с. 123.

					122.4142.08.ПЗ.СВЛ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		85

10. Alan Forbes. "The Joy of PHP Programming: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL" 2015. ст. 180.
11. Alan Beaulieu. "Learning SQL: Master SQL Fundamentals (2nd Edition)" / Alan Beaulieu – "O'Reilly Media". 2009 – 309с.
12. "PHP Language Reference" php.net [Електронний ресурс]: php.net. Режим доступу: <https://www.php.net/manual/en/langref.php> . - Дата доступу: 10.10.19.
13. Vue.js Docs [Електронний ресурс]: <https://vuejs.org/>. Режим доступу: <https://vuejs.org/v2/guide/>. – Дата доступу: 01.09.2019-01.06.2020.
14. Material Design Component Framework [Електронний ресурс]: <https://vuetifyjs.com/>. Режим доступу: <https://vuetifyjs.com/en/>. Дата доступу: 20.01.2020-01.06.2020.
15. Material Design Icons [Електронний ресурс]: <https://materialdesignicons.com/> Режим доступу: <https://materialdesignicons.com/>. Дата доступу: 27.01.2020-01.06.2020.
16. Vuex Docs [Електронний ресурс]: <https://vuex.vuejs.org/>. Режим доступу: <https://vuex.vuejs.org/guide/>. Дата доступу: 27.01.2020-01.06.2020.
17. Vue Router Docs [Електронний ресурс]: <https://router.vuejs.org/>. Режим доступу: <https://router.vuejs.org/guide/>. Дата доступу: 27.01.2020-01.06.2020.
18. Laravel Docs [Електронний ресурс]: <https://laravel.com/docs/6.x>. Режим доступу: <https://laravel.com/docs/6.x>. Дата доступу: 05.06.2019-01.06.2020.
19. ССБТ «Небезпечні і шкідливі виробничі фактори. Класифікація». ДСТУ 12.0.003-74\*. [Електронний ресурс]: ДСТУ 12.0.003-74\*. – М., 1980 – Режим доступу: <http://www.budinfo.org.ua/doc/1810987.jsp>. – Дата доступу: 05.05.2020.
20. «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та

напруженості трудового процесу». Наказ Міністерства охорони здоров'я України 08 квітня 2014 року N 248–. [Електронний ресурс] : – Режим доступу: <http://golovbukh.ua/regulations/2340/2592/2593/332659/> – Дата доступу : 05.05.2020.

21.«Державні будівельні норми. Природне та штучне освітлення». ДБН В 2.5.28- 2006. [Електронний ресурс]: ДБН В 2.5.28-2006. – Режим доступу: <http://www.info-build.com.ua/normativ/detail.php?ID=45079>. – Дата доступу: 25.04.2015.

22.Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98

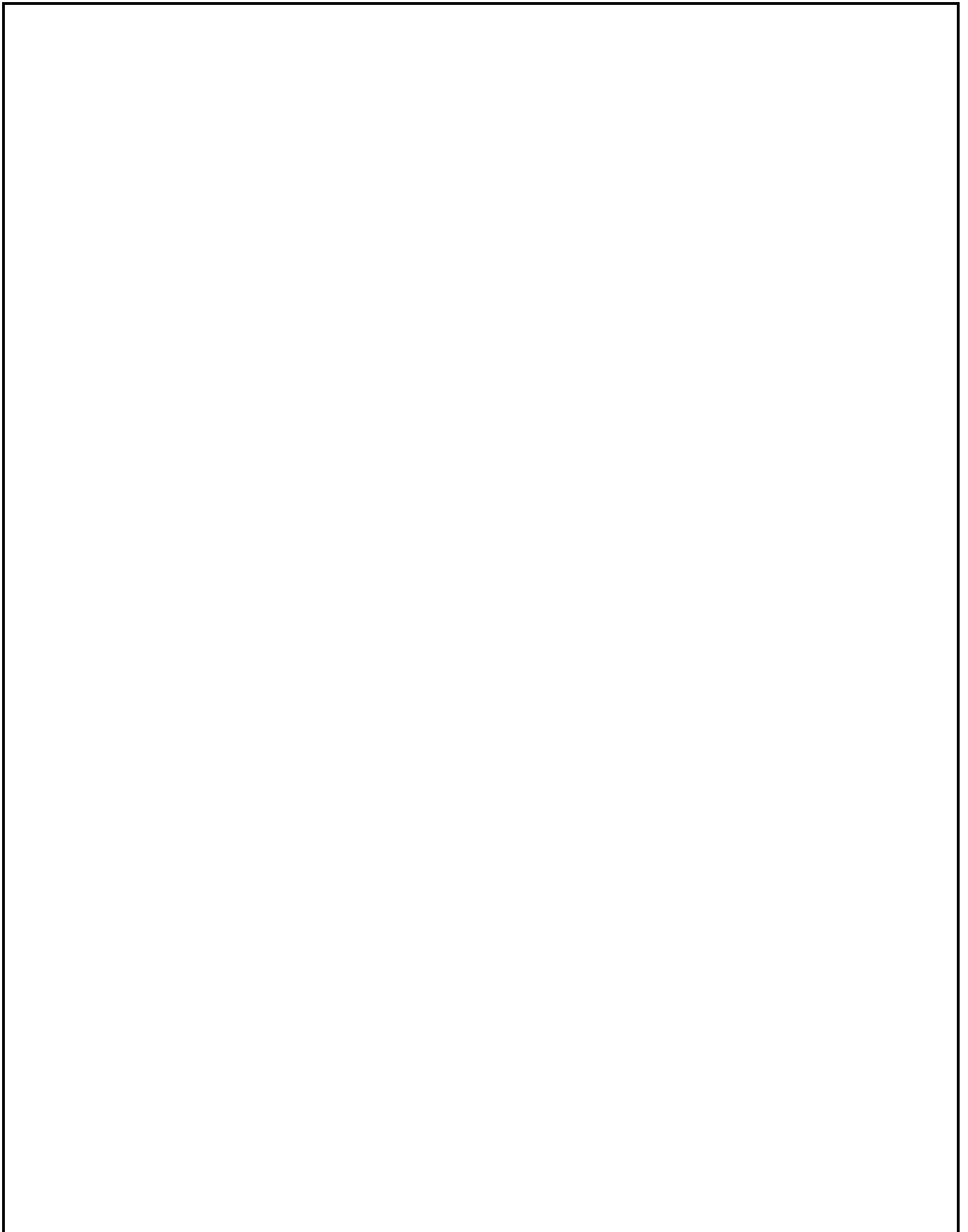
23.Вікіпедія [Електронний ресурс]: <https://uk.wikipedia.org/>. Режим доступу: [https://uk.wikipedia.org/wiki/Діаграма\\_діяльності](https://uk.wikipedia.org/wiki/Діаграма_діяльності). Дата доступу: 27.03.2020.

24.Wikipedia [Електронний ресурс]: <https://en.wikipedia.org/>. Режим доступу: [https://en.wikipedia.org/wiki/Calligra\\_Plan](https://en.wikipedia.org/wiki/Calligra_Plan). Дата доступу: 25.04.2020.

25.Wikipedia [Електронний ресурс]: <https://en.wikipedia.org/>. Режим доступу: <https://en.wikipedia.org/wiki/OpenProj>. Дата доступу: 25.04.2020.

26.Wikipedia [Електронний ресурс]: <https://en.wikipedia.org/>. Режим доступу: <https://en.wikipedia.org/wiki/GanttProject>. Дата доступу: 25.04.2020.

27.Wikipedia [Електронний ресурс]: <https://en.wikipedia.org/>. Режим доступу: [https://en.wikipedia.org/wiki/Basecamp\\_\(company\)](https://en.wikipedia.org/wiki/Basecamp_(company)). Дата доступу: 25.04.2020.



					122.4142.08.ПЗ			
Зм.	Аркаш	№ документа	Підпис	Дата				
Студент		Михалко М.І.			Додатки	Літ.	Аркуш	Аркушів
Керівник		Гайда А.Ю.					88	23
Зав. каф.		Михелєв І.Л.				НУК		



## Додаток А - Технічне завдання

### 1. Вступ

- 1.1. Назва – Автоматизована інформаційна система управління проектами ІТ-компанії.
- 1.2. Призначення програми – автоматизувати процеси управління, які відбуваються в ІТ-компаніях під час реалізації проектів шляхом автоматичного контролю виділеного часу на виконання завдання, створення звітів на основі збережених в системі даних про робочий час, виконані завдання та бюджету.
- 1.3. Область застосування - ІТ-компанія або невеликі групи розробників, які хочуть зручно контролювати робочий час, декларувати роботу над завданнями та швидко формувати звіти по виконаній роботі.

### 2. Призначення розробки

Система повинна надавати можливість створення проектів та завдань в яких вказуються терміни здачі та оцінка часу виконання. В процесі роботи над створеними завданнями користувач повинен мати можливість вести облік часу по кожному завданню. На основі цих даних, система повинна автоматично контролювати терміни здачі та у разі прострочення повідомляти про це користувачів, згідно їхньої ролі в проекті. Також на основі збережених даних повинні автоматично формуватися звіти по виконаним роботам, які можна переглядати в онлайн або скачувати у вигляді файлу та таблиці, які відображають рівень зайнятості та продуктивності розробників.

Всі перераховані функції спрямовані на мінімізацію додаткових витрат часу розробників та підвищення продуктивності – це і є головним призначенням системи.

Додатковим призначенням системи є зручне створення поміток, коментарів до завдань та перегляд активності користувачів.

					122.4142.08.ПЗ.Додаток А	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		89

### 3. Вимоги до системи

#### 3.1. Вимоги до функціональних характеристик

Система повинна містити наступні функціональні характеристики:

- Створення та редагування облікового запису користувача; додаткові засоби для здійснення швидкої авторизації в системі.
- Створення проектів шляхом декларування даних: назва, опис, категорія, валюта, бюджет, учасники.
- Автоматичне створення компанії на основі даних користувача, який декларує свій перший проект та редагування інформації про компанію: назва, мобільний телефон, електронна адреса, веб-сайт.
- Підключення та видалення користувачів до компанії та проектів. В разі підключення до проекту повинне відбуватися автоматичне підключення до компанії, якщо користувач відсутній.
- Створення завдань для реалізації проекту. При цьому вказується: назва, опис, термін задачі, оцінка часу виконання та виконавець.
- Облік часу виконання завдань, який відбувається після натиснення на кнопку старту в завданні.
- Можливість створення коментарів до завдань.
- Створення поміток в проекті. При цьому вказується: назва та опис.
- Автоматизовані сповіщення про нові проекти, завдання, помітки та прострочення термінів виконання робіт.
- Можливість перегляду рівня зайнятості учасників проекту та їхньої продуктивності на основі виконаних завдань.
- Створення звітів, які автоматично формуються на основі даних по виконанню завдань в проекті.
- Розрахунок витрат на учасників проекту на основі даних, які вказуються при підключенні користувача та часу витраченому на виконання завдань.

#### 3.2. Вимоги до надійності

3.2.1. Вимоги до забезпечення надійності функціонування програми

					122.4142.08.ПЗ.Додаток А	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		90

Стійке функціонування системи має бути забезпечено надійним функціонуванням бездротової мережі і мережі електропостачання в місці, де буде розміщений сервер.

3.2.2. Відмова виконання програмного продукту через некоректні дії користувача.

- Перевірка прав користувача на виконання дії.
- Перевірка введених даних перед виконанням будь-яких дій.
- Автоматичне створення копії бази даних кожного дня та видалення застарілих (більше одного тижня).

### **3.3. Умови експлуатації**

Системою можуть користуватися будь-які користувачі. Для експлуатації системи необхідна наявність інтернет з'єднання на серверній частині та клієнтській. Інтернет з'єднання є не обов'язковим у разі розгортання системи на локальному сервері, але необхідна наявність засобів для функціонування програми. Доступ відбувається через браузер, отже ще однією умовою є його наявність у користувача.

### **3.4. Вимоги до інформаційної та програмної сумісності**

3.4.1. Вимоги до інформаційних структур і методів розв'язання  
Вимоги до інформаційних структур і методів розв'язання відсутні.

3.4.2. Вимоги до програмних засобів , які використовуються  
програмою

Програмні засоби що використовуються мають бути вільно поширювані.

### **3.5. Вимоги до маркування та упаковки**

3.5.1. Вимоги до маркування

- Читабельність тексту, чіткість ілюстрацій.
- Достовірність інформації про систему.

3.5.2. Вимоги до упаковки

Вимоги до упаковки відсутні.

### **3.6. Вимоги до транспортування та зберігання**

Вимоги до транспортування програми відсутня.

### **3.7. Спеціальні вимоги**

Спеціальні вимоги відсутні.

## **4. Вимоги до програмної документації**

### **4.1. Попередній склад програмної документації**

Попередній склад програмної документації повинен включати в себе:

- технічне завдання;
- устав проекту.

### **4.2. Склад програмної документації**

Програмна документація складається з наступних програмних документів:

- Устав проекту.
- Технічне завдання.
- Опис системи.
- Код програми.
- Керівництво користувача.

## **5. Техніко-економічні показники**

### **5.1. Економічні переваги розробки**

Орієнтовна економічна ефективність не розраховується.

## **6. Стадії розробки програмного забезпечення**

Розробка повинна включати 5 стадій:

1. Технічне завдання.
2. Ескізний проект.
3. Технічний проект.
4. Робочий проект.
5. Впровадження.

					122.4142.08.ПЗ.Додаток А	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		92

## Додаток Б – Загальний опис системи

Автоматизована інформаційна система управління проектами ІТ-компанії є інструментом, який позбавить користувача виконувати великі кількості ручних робіт, які включають в себе: облік часу, контроль термінів виконання, формування звітів та розрахунку витрат. Основними функціональними характеристиками системи є:

- Створення та редагування проектів.
- Створення, редагування та контроль за завданнями, які необхідно вирішити для реалізації проекту.
- Формування показників продуктивності та рівня зайнятості користувачів, які займаються реалізацією проекту.
- Формування звітів, які необхідні для розрахунків витрат на реалізацію проекту та аналізу роботи розробників проектів.

Система реалізована за допомогою мови програмування PHP та js з використанням додаткових бібліотек для SPA додатку та має зручний та інтуїтивно зрозумілий інтерфейс.

Структура всіх сутностей повністю відповідає фізичній моделі бази даних, яка була розроблена в роботі. Для зручного та швидкого розгортання системи всі сутності в таблиці створювалися міграціями.

Деякі сутності в базі даних є статичними і повинні бути заповненими при розгортанні системи. Для їх заповнення було розроблено сіді, які автоматично заповнять базу даних необхідною інформацією при розгортанні системи.

					122.4142.08.ПЗ.Додаток Б	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		93

## Додаток В – Інструкція з використання системи

Веб-додаток містить головну сторінку на яку може перейти будь-який користувач. Вона має ознайомлювальний характер та в тезисах відображає деякі головні функції системи. Головна сторінка додатку представлена на рисунку В.1.

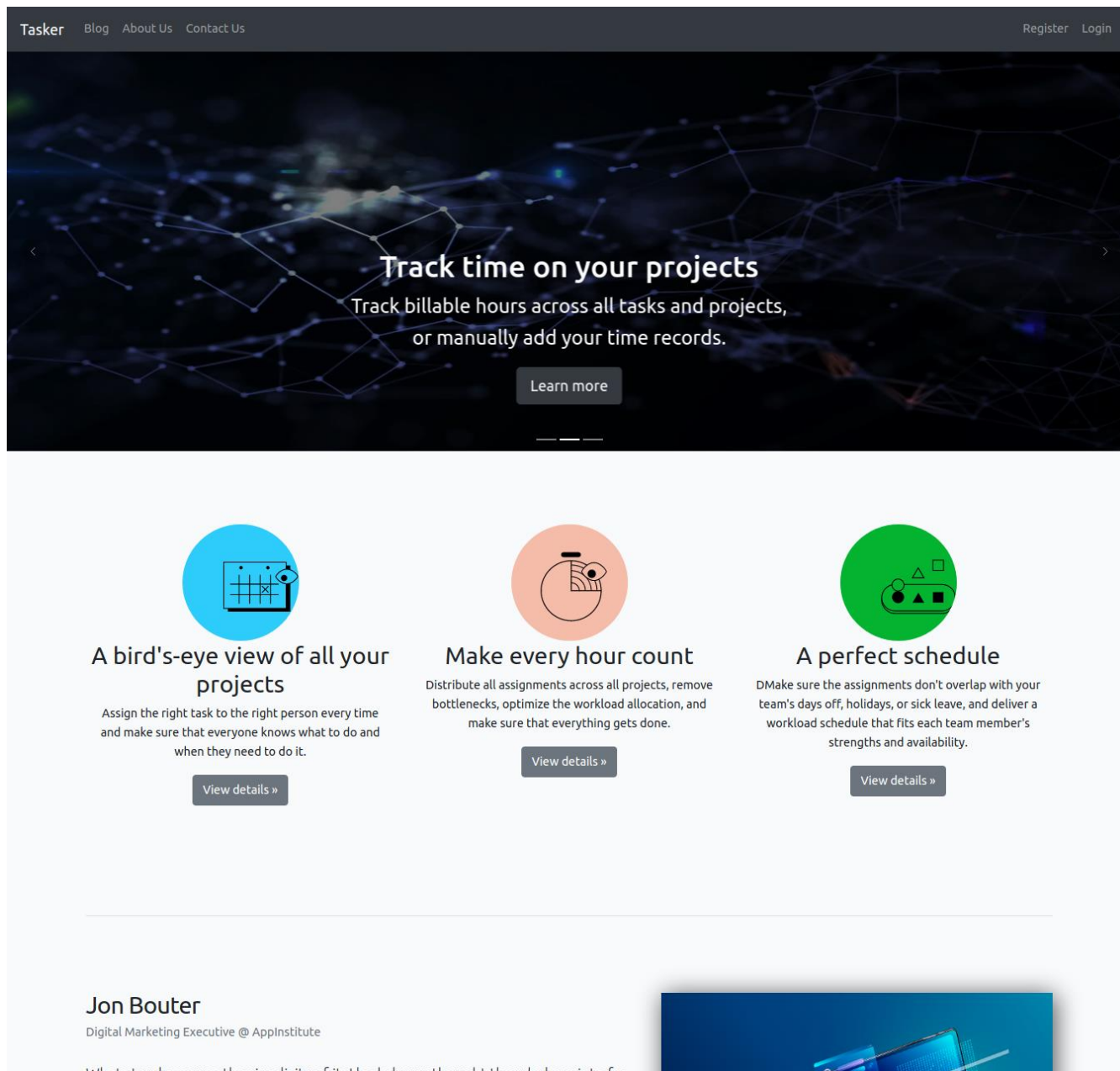


Рисунок В.1 – Головна сторінка

					122.4142.08.ПЗ.Додаток В	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		94

Для того, щоб почати використовувати автоматизовану інформаційну систему управління проектами необхідно створити обліковий запис. Сторінка реєстрації представлена на рисунку В.2.

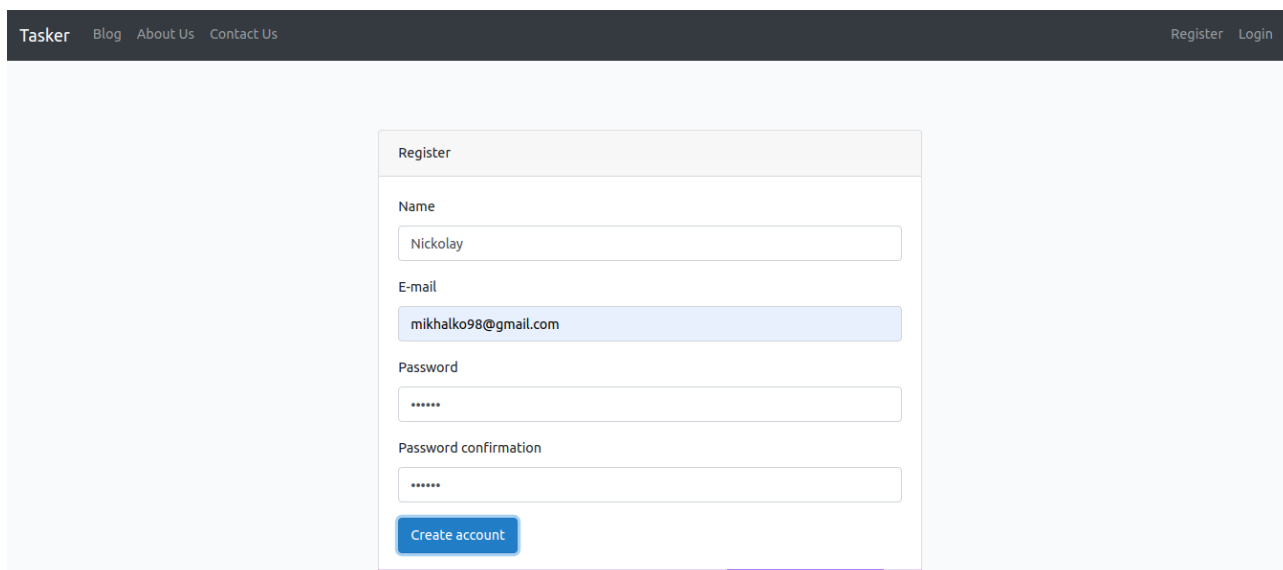


Рисунок В.2 – Сторінка реєстрації

Після того, як було введено всі необхідно дані та дані пройшли валідацію в системі буде створено новий обліковий запис. Для авторизації необхідно повторно ввести електронну адресу та пароль. Якщо користувач ввів коректні дані, то відбувається переадресація на головну сторінку профіля. Сторінка авторизації та головна сторінка профіля представлені на рисунках В.3, В.4.

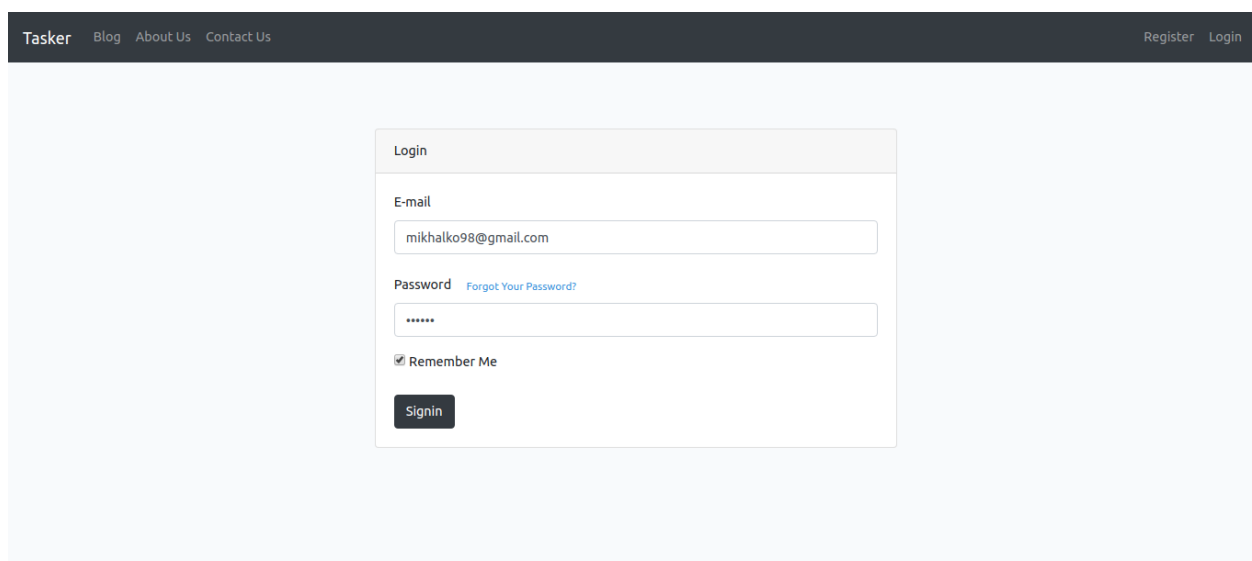


Рисунок В.3 – Сторінка авторизації

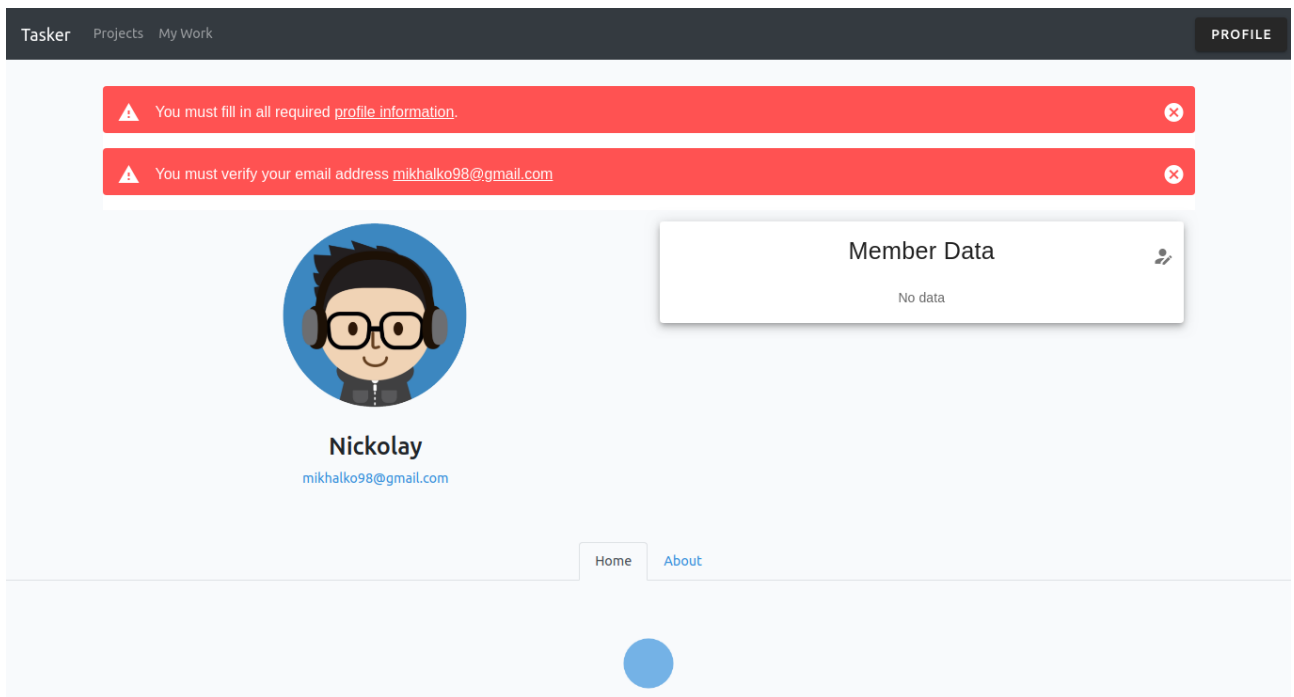


Рисунок В.4 – Головна сторінка профіля

Оскільки обліковий запис було тільки створено і користувач ще не підтвердив свою електронну адресу і не заповнив обов’язкові дані профіля, то відображаються два попередження червоного кольору. Після успішної реєстрації на вказану електронну адресу відправляється повідомлення з посиланням, перейшовши за яким відбувається підтвердження email. Лист відправлений на поштовий сервіс представлений на рисунку В.5.

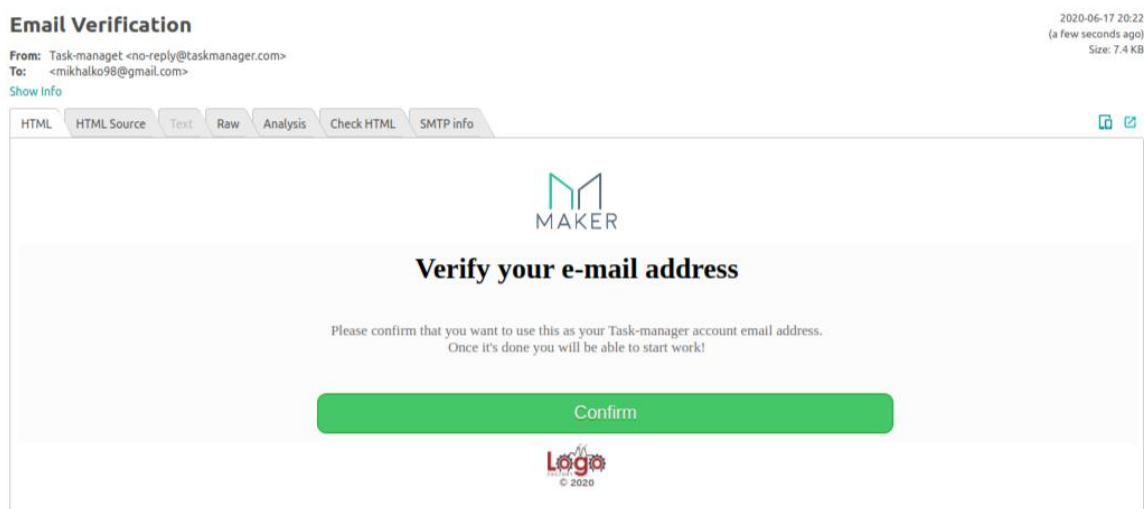


Рисунок В.5 – Електронний лист для підтвердження електронної

					122.4142.08.ПЗ.Додаток В	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		96



Коли користувач натискає на кнопку в електронному листі, то відбувається переадресація на сторінку активації і якщо вона пройшла успішно, то користувач побачить повідомлення, яке представлено на рисунку В.6.

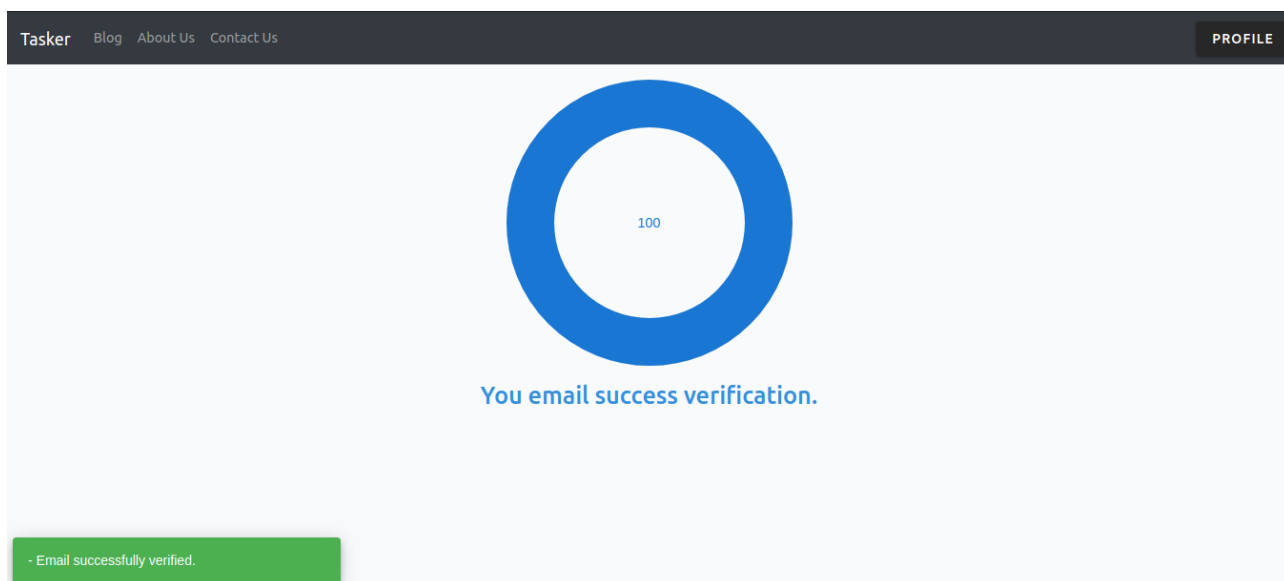


Рисунок В.6 – Успішна верифікація електронної адреси

Для повної активації облікового запису необхідно заповнити всі обов'язкові дані користувача. Сторінка для заповнення цих даних представлена на рисунку В.7. Головна сторінка активованого облікового запису представлена на рисунку В.8.

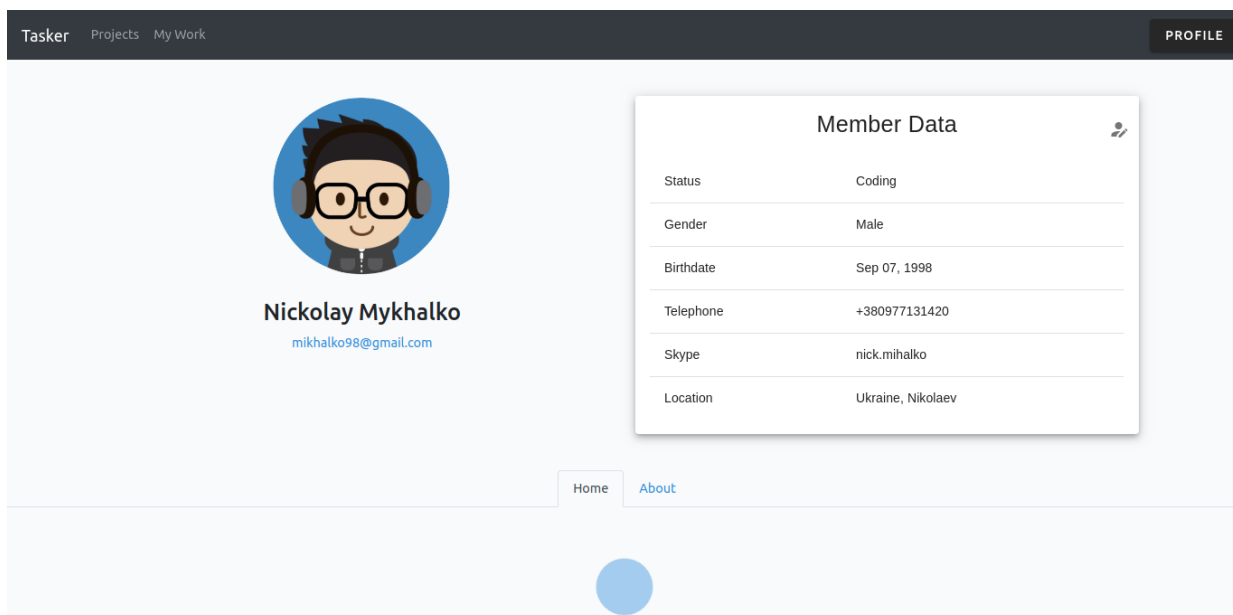


Рисунок В.8 – Активований обліковий запис

Tasker Projects My Work
PROFILE

**User Settings**

- Profile
- Account
- Billing
- Notification

### Public Avatar

You can change your avatar here

Upload new avatar

Choose file... No file chosen

---

### Main settings

This information will appear on your profile

Status

Coding ✕

Enter your status

First name

Nickolay ✕

Enter your first name

Last name

Mykhalko ✕

Enter your last name

Select

Male ▼

Select your gender

Birthdate date

1998-09-07

1998

<
September 1998
>

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Phone number

+380977131420 ✕

Enter your phone number

Contact e-mail

mikhalko98@gmail.com ✕

Enter your contact e-mail

Skype

nick.mihalko ✕

Enter your skype

Website

Enter your web-site

Location

Ukraine, Nikolaev ✕

Enter your location

Bio

Enter your bio

UPDATE PROFILE SETTINGS

Рисунок В.7 – Сторінка редагування облікового запису

Додаткові опції облікового запису: підключення соціальних мереж, зміна nickname та видалення облікового запису можна обрати на сторінці редагування акаунту, яка представлена на рисунку В.8.

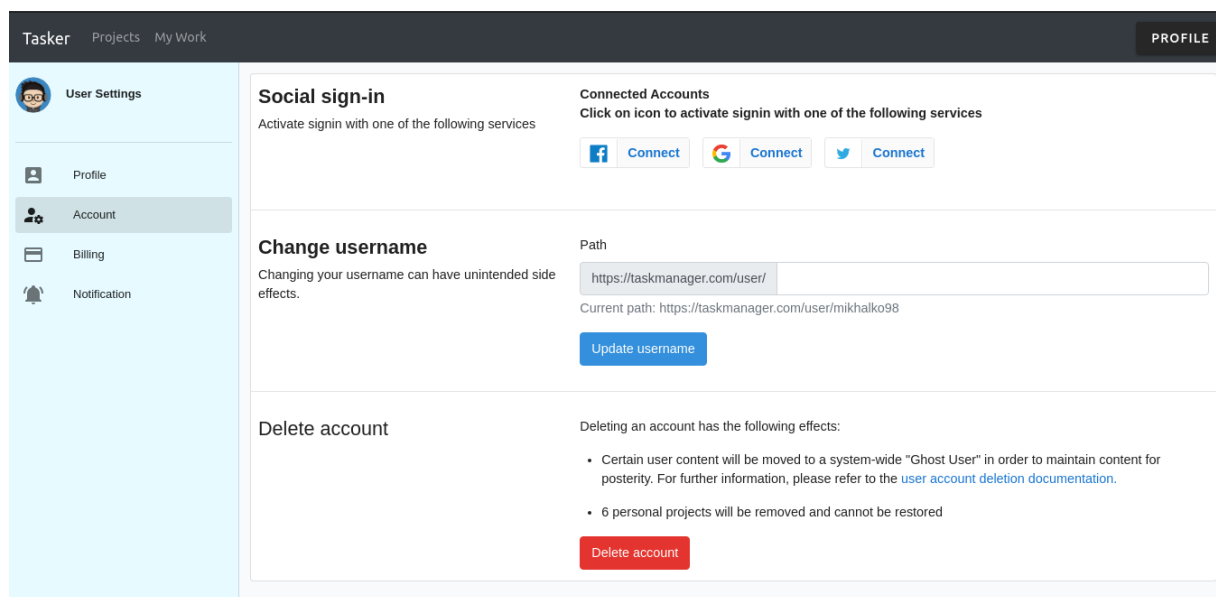


Рисунок В.8 – Додаткові опції облікового запису

Після того, як було активовано обліковий запис, можна створити проект перейшовши на сторінку “Projects”, яка представлена на рисунку В.9. При натисканні на кнопку “Create a new project” відобразиться форма, яка представлена на рисунку В.10.

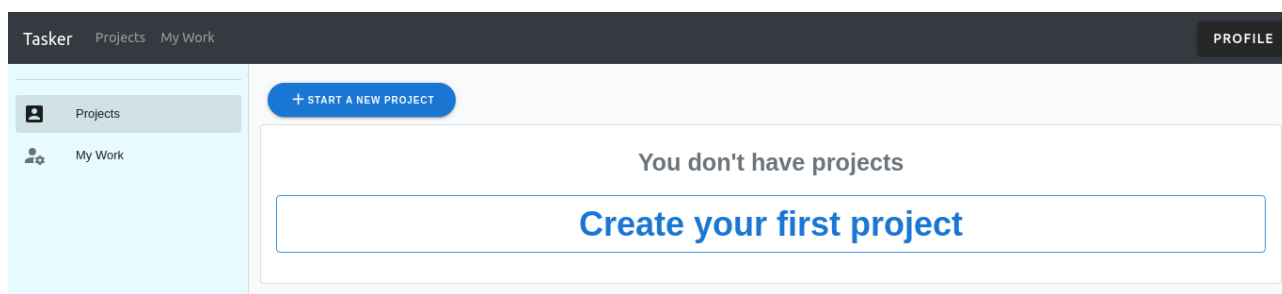


Рисунок В.9 – Сторінка “Projects”

The image shows a 'New Project' form in a web application. The form is titled 'New Project' and is located in the 'Tasker' application. The form includes the following fields and options:

- Project Name:** A text input field containing 'Tasker'. A tooltip explains that the project name helps navigate through many projects.
- Project description:** A text area containing a description of the 'Tasker' web-application. A tooltip explains that a brief description helps developers familiarize themselves with the project.
- Label:** A dropdown menu set to 'New'. A tooltip explains that the label shows the project's stage.
- Category:** A dropdown menu set to 'Web-site'. A tooltip explains that the category is used for filtering and searching.
- Client company:** A text input field containing 'Company name'. A tooltip explains that the client company is used for filtering and searching.
- Enable time and expense tracking:** A toggle switch that is currently turned on.
- Allow clients to see time and expenses:** A toggle switch that is currently turned off.
- Currency:** A dropdown menu set to 'USD - US Dollar'.
- Budget:** A text input field containing '50000'. A tooltip explains that this amount is considered as both fixed budget and income.

At the bottom of the form, there is a green button labeled 'CREATE PROJECT'.

Рисунок В.10 – Форма створення проекту

Наступним кроком, після заповнення форми, представленої на рисунку 10 є підключення розробників та клієнтів. Сторінка для їх підключення представлена на рисунку В.11.

Разом зі створення першого проекту відбувається автоматична реєстрація компанії, налаштувати дані якої можна на сторінці “Edit company”. Всі користувачі, які підключені до компанії не потребуватимуть підтвердження при підключенні до проекту. Якщо ж відбувається підключення учасника проекту, який відсутній в компанії, то з’являється модальне вікно в якому необхідно обрати посаду та рівень заробітної плати і лише після цього користувача буде

					122.4142.08.ПЗ.Додаток В	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		100

підключена до проекту та компанії. Видалення учасника або клієнта проекту відбувається на цій же сторінці (рисунок В.11).

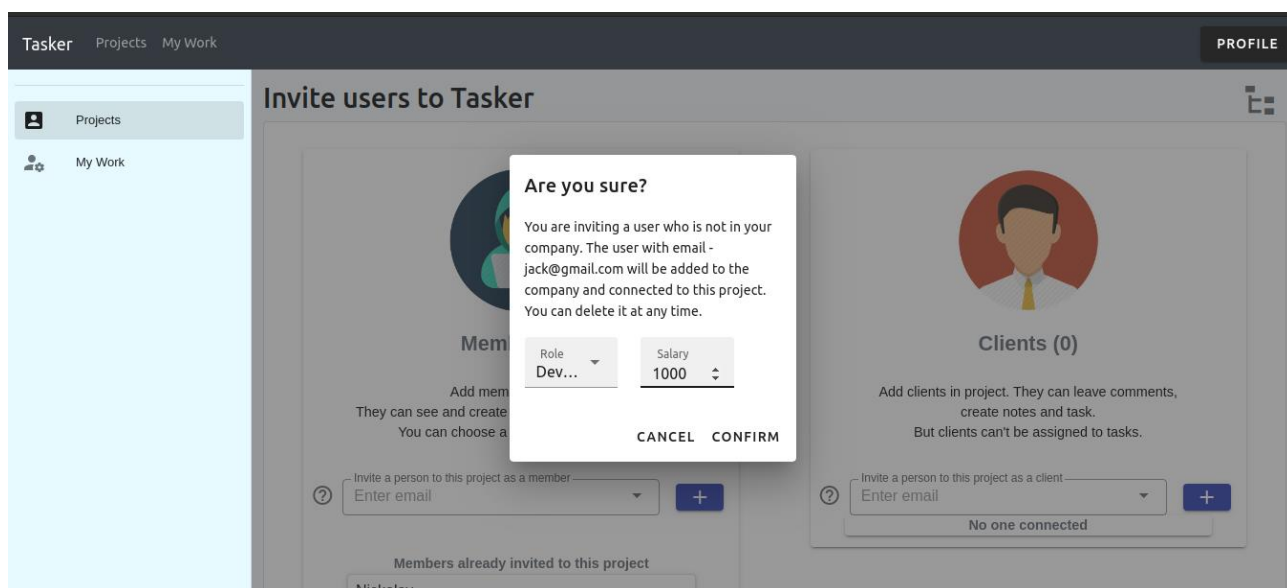


Рисунок В.11 – Підключення розробників та клієнтів

Для налаштування підключених до компанії користувачів необхідно перейти на сторінку “Company members”. На ній можна підключити нових або видалити існуючих користувачів. Також можна налаштувати їхні посади та рівні заробітних плат.

Створення нового завдання відбувається на головній сторінці проекту. Для його створення необхідно натиснути на кнопку “Add a task” і заповнити всі дані у формі. Після цього нове завдання з’явиться у списку всіх завдань проекту та на сторінці “My Work” у назначеного виконавця. Також призначений виконавець отримає повідомлення на електронну адресу та месенджер “Slack”(якщо підключений).

Для перегляду завдання та обліку його часу виконання необхідно натиснути на назву завдання (сторінка “My Work” або список всіх завдань проекту) після чого відобразяться дані завдання та часомір при натисканні на який почнеться автоматичний запис часу виконання завдання. При повторному натисканні на часомір облік часу зупиниться та час збережеться в системі.

					122.4142.08.ПЗ.Додаток В	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		101

Звіти та показники завантаженості і продуктивності учасників проектів можна переглянути перейшовши на сторінку адміністрування проекту. На ній окремо перераховані всі звіти. Для їх створення необхідно обрати, який звіт потрібно створити та вказати період. Після цього відбудеться автоматичне створення звіту у форматі pdf, який можна вивантажити з системи. Показники завантаженості і продуктивності створюються аналогічно.

					122.4142.08.ПЗ.Додаток В	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		102

## Додаток Г – Структура програмного забезпечення

В результаті реалізації автоматизованої інформаційної системи управління проектами ІТ-компанії було створено додаток з використанням мови програмування PHP для серверної частини та JS для користувацького інтерфейсу. Крім цього використовувались існуючі інструменти для створення сучасних та багатофункціональних систем: Laravel Framework 6.2, fideloper proху, guzzlehttp guzzle, laravel ui, laravel jwt-auth, fzaninotto faker, Carbon для серверної частини та Axios, jQuery, Poper.js, Vue.js, Vuex, Vue-resources, Vue-router, Vue-templae-compiler, Vuetify, Vue-material-design-icons, Vue-auth, Bootstrap 4 для користувацького веб-інтерфейсу.

Структура програмного забезпечення розроблена у вигляді окремих пакетів. Це дозволило згрупувати фрагменти коду в один логічний модуль, який можна використовувати повторно.

Для встановлення залежностей власнорозроблених та існуючих пакетів було використано Composer. Composer - менеджер пакетів прикладного рівня для мови програмування PHP що забезпечує стандартний формат для управління залежностями у програмному забезпеченні та необхідними бібліотеками. Для компіляції JavaScript модулів в єдиний JS-файл, збору всіх ресурсів та слідкуванням за змінами в процесі реалізації системи було використано webpack. Webpack - це постачальник модулів JavaScript з відкритим кодом. Він створений в основному для JavaScript, але може трансформувати активні компоненти, такі як HTML, CSS та зображення, якщо включені відповідні навантажувачі. Webpack приймає модулі із залежностями та генерує статичні активи, що представляють ці модулі.

Важливою системою, яка використовувалась при реалізації системи є git - розподілена система керування версіями файлів та спільної роботи. За допомогою цього інструмента було зручно контролювати закінчені фрагменти коду та повертатися до попередніх версій програми. Це скоротило час реалізації системи.

					122.4142.08.ПЗ.Додаток Г	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		103

## Додаток Д – Тексти програмних модулів системи

Система складається з великої кількості модулів та підсистем, тому нижче представлено тексти лише деяких програмних модулів. Повну структуру та тексти програмних модулів можна переглянути в системі контролю версії [gitlab.com](https://gitlab.com/mykhalko07/work_manager) за посиланням - [https://gitlab.com/mykhalko07/work\\_manager](https://gitlab.com/mykhalko07/work_manager).

### Лістинг 1 – ProjectController.php

```
<?php

namespace Niko\Projects\Controllers\Api;

use Carbon\Carbon;
use Illuminate\Http\JsonResponse;
use Illuminate\Routing\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

//Models
use Niko\Projects\Models\Companies;
use Niko\Projects\Models\CompanyMembers;
use Niko\Projects\Models\ProjectCategories;
use Niko\Projects\Models\ProjectMembers;
use Niko\Projects\Models\ProjectMembersRoles;
use Niko\Projects\Models\Projects;
use Niko\Projects\Models\ProjectStages;
use Niko\Users\Models\Users;

class ProjectsController extends Controller
{
    /**
     * Get list viewer project
     *
     * @param Request $request
     * @return JsonResponse
     */
    public function projects(Request $request)
    {
        $viewer = Auth::user();
        if(!$viewer->active || !$viewer->email_verified_at)
            return response()->json(['status' => false, 'redirect' => ['name' =>
            'user.profile', 'params' => ['username' => $viewer->username]]]);

        $viewer_projects = Projects::where('owner_id', '=', $viewer->getKey())-
        >orderBy('created_at', 'DESC')->get();
        foreach ($viewer_projects as $viewer_project){
            //set Stage
            $viewer_project->stage = ProjectStages::where('id', '=',
            $viewer_project->stage_id)->pluck('name')->first();

            //set category
            $viewer_project->category = ProjectCategories::where('id', '=',
            $viewer_project->stage_id)->pluck('name')->first();

            //set date start
            $viewer_project->date_start = Carbon::parse($viewer_project-
            >created_at)->format('M d, Y');
```

					122.4142.08.ПЗ.Додаток Д	Аркуш
						104
Зм.	Аркуш	№ документа	Підпис	Дата		



```

    }
    $projects = [
        'viewer_projects' => !empty($viewer_projects)? $viewer_projects-
>toArray(): [],
    ];

    return response()->json(['status' => true, 'projects' => $projects]);
}

/**
 * Get invited members
 *
 * @param Request $request
 * @param null $project_id
 * @return JsonResponse
 */
public function invitedMembers(Request $request, $project_id = null)
{
    if(empty($project_id))
        return response()->json(['status' => false, 'errors' => ['Project id
can\'t be empty'], 'redirect' => ['name' => 'work.projects']]);

    $project = Projects::find($project_id);
    if(!$project)
        return response()->json(['status' => false, 'errors' => ['Project
not found.'], 'redirect' => ['name' => 'work.projects']]);

    $project_members = ProjectMembers::select('project_members.*',
'users.name', 'users.email', 'project_members_roles.name as role_name')
->leftJoin('users', 'project_members.user_id', '=', 'users.id')
->leftJoin('project_members_roles', 'project_members_roles.id',
'=', 'project_members.role_id')
->where('project_id', '=', $project_id)
->get();

    $invited_members = $project_members->toArray();
    return response()->json(['status' => true, 'invited_members' =>
$invited_members]);
}

/**
 * @param Request $request
 * @param null $project_id
 * @return JsonResponse
 */
public function notInvitedCompanyMembers(Request $request, $project_id =
null)
{
    if(empty($project_id))
        return response()->json(['status' => false, 'errors' => ['Project id
can\'t be empty'], 'redirect' => ['name' => 'work.projects']]);

    $viewer = Auth::user();
    $project = Projects::find($project_id);
    if(!$project)
        return response()->json(['status' => false, 'errors' => ['Project
not found.'], 'redirect' => ['name' => 'work.projects']]);

    $company = Companies::find($project->company_id);

    if(!$company)
        return response()->json(['status' => false, 'errors' => ['Company
not found.']]));
}

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>105</b>

```

        $project_members = ProjectMembers::select('project_members.*',
'users.name', 'users.email', 'project_members_roles.name as role_name')
        ->leftJoin('users', 'project_members.user_id', '=', 'users.id')
        ->leftJoin('project_members_roles', 'project_members_roles.id',
'=', 'project_members.role_id')
        ->where('project_id', '=', $project_id)
        ->where('user_id', '!=', $viewer->getKey())
        ->get()
        ->keyBy('user_id')
        ->toArray();

        $company_members = CompanyMembers::select('company_members.*',
'users.name', 'users.email', 'company_members_roles.name as role_name')
        ->leftJoin('users', 'company_members.user_id', '=', 'users.id')
        ->leftJoin('company_members_roles', 'company_members_roles.id',
'=', 'company_members.role_id')
        ->where('company_id', '=', $company->getKey())
        ->where('user_id', '!=', $viewer->getKey())
        ->get()
        ->keyBy('user_id')
        ->toArray();

        $not_invited_company_members = [];
        foreach ($company_members as $user_id => $company_member){
            if(!isset($project_members[$user_id])){
                $not_invited_company_members[] = $company_member;
            }
        }
        return response()->json(['status' => true, 'not_invited_company_members'
=> $not_invited_company_members]);
    }

    public function removeMember(Request $request, $project_id = null)
    {
        if(empty($project_id))
            return response()->json(['status' => false, 'errors' => ['Project id
can\'t be empty'], 'redirect' => ['name' => 'work.projects']]);

        $remove_member_id = $request->get('member_id');
        if(!$remove_member_id)
            return response()->json(['status' => false, 'errors' => ['Member id
can\'t be empty']]);

        $project = Projects::find($project_id);
        if(!$project)
            return response()->json(['status' => false, 'errors' => ['Project
not found.'], 'redirect' => ['name' => 'work.projects']]);

        $project_member = ProjectMembers::where('user_id', '=',
$remove_member_id)->first();
        if(!$project_member)
            return response()->json(['status' => false, 'errors' => ['User not
found.']] );

        $project_member->delete();

        return response()->json(['status' => true, 'message' => 'Member
successfully removed.']);
    }

    public function details(Request $request, $project_id = null)

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
						<b>106</b>
Зм.	Аркуш	№ документа	Підпис	Дата		

```

    {
        if(!$project_id)
            return response()->json(['status' => false, 'errors' => ['Project id
can\'t be empty.'], 'redirect' => ['name' => 'work.projects']]);

        $project = Projects::find($project_id);

        if(!$project)
            return response()->json(['status' => false, 'errors' => ['Project
not found.'], 'redirect' => ['name' => 'work.projects']]);

        //set Stage
        $project->stage = ProjectStages::where('id', '=', $project->stage_id)-
>pluck('name')->first();

        //set category
        $project->category = ProjectCategories::where('id', '=', $project-
>stage_id)->pluck('name')->first();

        //set date start
        $project->date_start = Carbon::parse($project->created_at)->format('M d,
Y');

        return response()->json(['status' => true, 'project_data' => $project]);
    }

    public function getNotCompanyMembers(Request $request, $project_id = null)
    {
        $search = $request->get('search');

        if(!$project_id)
            return response()->json(['status' => false, 'errors' => ['Project id
can\'t be empty.'], 'redirect' => ['name' => 'work.projects']]);

        $project = Projects::find($project_id);
        if(!$project)
            return response()->json(['status' => false, 'errors' => ['Project
not found.'], 'redirect' => ['name' => 'work.projects']]);

        $company = Companies::find($project->company_id);
        $company_members = CompanyMembers::where('company_id', '=', $company-
>getKey())->pluck('user_id')->toArray();

        $select_users = Users::select()->whereNotIn('id', $company_members);
        if($search){
            $select_users->where('email', 'LIKE', '%'.$search.'%');
        }
        $users = $select_users->get();

        return response()->json(['status' => true, 'users' => $users-
>toArray()]);
    }
}

```

## Лістинг 2 – Projects.php

```

<?php

namespace Niko\Projects\Models;

use Illuminate\Database\Eloquent\Model;

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		107

```

class Projects extends Model
{
  /**
   * The table associated with the model.
   *
   * @var string
   */
  protected $table = 'projects';

  /**
   * The attributes that are mass assignable.
   *
   * @var array
   */
  protected $fillable = [
    'name', 'description', 'stage_id', 'category_id', 'client_company_name',
    'available_to_customer', 'currency_id', 'budget', 'owner_id', 'company_id'
  ];

  /**
   * The attributes that should be mutated to dates.
   *
   * @var array
   */
  protected $dates = ['created_at', 'updated_at'];

  /**
   * The attributes that should be hidden for arrays.
   *
   * @var array
   */
  protected $hidden = [];
}

```

### ЛІСТИНГ 3 – Projects.vue

```

<template>
  <div class="container" id="projects">
    <v-app id="inspire">
      <v-btn
        rounded
        class="ma-2 start-project"
        :loading="loading"
        :disabled="loading"
        color="primary"
        max-width="200"
        @click="loader = 'loading'"
        :to="{name: 'work.projects.create'}"
      >
        <v-icon dark size="20">mdi-plus</v-icon> Start a New Project
      </v-btn>
    </v-app>

    <div class="card card-default p-3 d-inline-block w-100">

      <v-app id="inspire">
        <v-list>
          <v-list-group
            v-if="Object.keys(projects.viewer_projects).length != 0"
            prepend-icon="mdi mdi-alpha-p-box-outline"
          >

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		108

```

        value="true"
      >
        <template v-slot:activator>
          <v-list-item-content>
            <v-list-item-title>Personal Projects</v-list-
item-title>
            </v-list-item-content>
          </template>
          <v-list-item>
            <v-card v-for="(project, i) in
projects.viewer_projects" :key="i" class="project-card m-3 float-left">
              <v-img
src="https://cdn.vuetifyjs.com/images/cards/sunshine.jpg"
                height="200px"
                class="mask flex-center"
              ></v-img>
              <v-chip v-if="project.stage" class="label-
project-card" color="primary" label text-color="white">
                <v-icon left>mdi-label</v-icon>
                {{ project.stage }}
              </v-chip>
              <v-card-title>
                {{ project.name }}
              </v-card-title>
              <v-card-subtitle class="pb-0">
                {{ project.category }}
              </v-card-subtitle>
              <v-card-subtitle v-if="project.date_start"
class="pt-0 text--disabled">
                <small>Start: {{ project.date_start
}}</small>
              </v-card-subtitle>
            </v-card>
          </v-list-item>
        </v-list-group>
        <div v-else>
          <h3 class="text-center w-100 text-muted font-weight-
bold">You don't have projects</h3>
          <h1 class="text-center w-100 text-muted font-weight-bold
mt-5 border border-primary rounded">
            <router-link :to="{ name: 'work.projects.create' }"
class="nav-link">Create your first project</router-link>
          </h1>
        </div>
      </v-list>
    </v-app>
  </div>
</div>
</template>

<script>
  import axios from "axios";

  export default {
    data() {
      return {
        loader: null,
        loading: false,
        projects: {
          viewer_projects: {},
          include_projects: {},

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		109

```

        }
    },
    mounted() {
        this.getProjects();
    },
    methods: {
        getProjects(){
            var url = `/projects/projects`;
            axios
                .get(url)
                .then( response => {
                    if(!response.data.status){
                        if(response.data.redirect){
                            this.$router.push(response.data.redirect);
                        }
                        this.$store.dispatch('setSnackbarOptions', {
                            'text': ['Server Errors.'],
                            'color': 'error'});
                        this.$store.dispatch('showSnackbar');
                    }else{
                        if(response.data.projects){
                            this.projects = response.data.projects;
                        }
                    }
                })
                .catch((error) => {
                    console.log(error);
                });
        }
    },
    watch: {
        loader () {
            const l = this.loader;
            this[l] = !this[l];

            setTimeout(() => (this[l] = false), 3000);

            this.loader = null
        },
    },
}
</script>

<style scoped>
.v-application {
    background-color: var(--v-background-base) !important;
}
.start-project{
    font-size: 10px;
    font-weight: 600;
    text-decoration:none;
}
.project-card{
    width: 30%;
}
@media (max-width: 1280px) {
    .project-card{
        width: 45%;
    }
}
@media (max-width: 770px) {
    .project-card{
        width: 90%;
    }
}

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>110</b>

```

    }
}
.label-project-card{
    position: absolute;
    top: 10px;
    left: 0;
}
.project-card {
    -webkit-transition: all 0.6s ease;;
    -moz-transition: all 0.6s ease;;
    -o-transition: all 0.6s ease;;
    transition: all 0.6s ease;
}
.project-card:hover{
    -webkit-transition: all 0.6s ease;;
    -moz-transition: all 0.6s ease;;
    -o-transition: all 0.6s ease;;
    transition: all 0.6s ease;
    -webkit-transform: scale(1.02);
    transform: scale(1.02);
    -webkit-filter: drop-shadow( 3px 3px 5px rgba(0, 0, 0, .5));
    filter: drop-shadow( 3px 3px 5px rgba(0, 0, 0, .5));
}
</style>

```

					122.4142.08.ПЗ.Додаток Д	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>111</b>