

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Черкасова Деніса Олексійовича

Прізвище, ім'я, по батькові студента(ки)

на здобуття ступеня вищої освіти Бакалавра

Програмний застосунок для розрахунку робочого часу співробітників підприємства

Назва теми

Рівень вищої освіти Перший (бакалаврський)

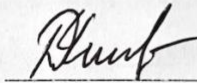
Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

КВРІПЗ. 2101094.01.20.ПЗ

Виконав студент IV курсу, група ПЗ-21-1

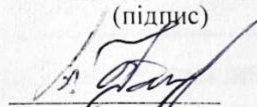


(підпис)

Деніс ЧЕРКАСОВ

(Ім'я, ПРІЗВИЩЕ)

Керівник канд. техн. наук, доцент



(підпис)

Юрій ФОРКУН

(Ім'я, ПРІЗВИЩЕ)

Нормоконтролер канд. техн. наук, доцент



(підпис)

Оксана ЯШИНА

(Ім'я, ПРІЗВИЩЕ)

Кількість балів _____

Оцінка за шкалою:

національною _____ /ЄКТС _____

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення



Підпис

Леонід БЕДРАТЮК

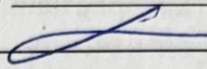
(Ім'я, ПРІЗВИЩЕ)

11 червня 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри 173
Л. П. Бедратюк 
2.01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Черкасову Денису Олексійовичу

Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Програмний застосунок для розрахунку робочого часу співробітників підприємства

Керівник кваліфікаційної роботи Форкун Юрій Вікторович, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2024 р. № 6

2. Строк подання студентом роботи на кафедру 01.06.2025 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

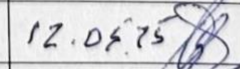
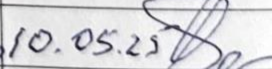
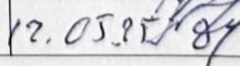
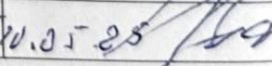
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі, проектування програмного застосунку, програмна реалізація та тестування

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Презентаційні матеріали (слайди, 17 шт.), діаграма варіантів використання, діаграма архітектури патернів, фізична діаграма бази даних, діаграма класів, взаємодії модулів, блок-схема алгоритму роботи програмного застосунку для розрахунку робочого часу співробітників підприємства, діаграма послідовностей для процесу редагування завдання

6. Консультанти розділів кваліфікаційної роботи

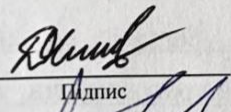
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Яшина О. М., к.т.н., доцент	<u>12.05.25</u> 	<u>10.05.25</u> 
Антиплагіат	Форкун Ю.В., к.т.н., доцент	<u>12.05.25</u> 	<u>10.05.25</u> 

7. Дата видачі завдання « » 2025р.

КАЛЕНДАРНИЙ ПЛАН

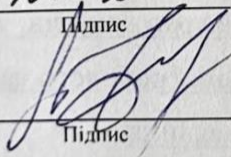
Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою дипломного проєктування, визначення та узгодження індивідуальних тем кваліфікаційних робіт (КвР)	01.12 – 31.12.2024	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.01 – 20.02.2025	
3 Проєктування програмного забезпечення	21.02 – 20.03.2025	
4 Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	21.03 – 30.04.2025	
5 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2025	
6 Попередній захист КвР	Травень 2025	Згідно графіка
7 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2025	
8 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2025	

Студент


Підпис

Черкасов Д.О.
Ініціали, прізвище

Керівник роботи


Підпис

Форкун Ю. В.
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмний застосунок для розрахунку робочого часу співробітників підприємства»

Автор роботи: Черкасов Денис Олексійович

Керівник роботи: Форкун Юрій Вікторович

Пояснювальна записка: 133 с., 19 рис., 3 табл., 3 дод., 41 джерела.

Графічна частина: 17 слайдів.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНИЙ ЗАСТОСУНОК,
СПІВРОБІТНИК, РОБОЧИЙ ЧАС, ШТАТНИЙ РОЗПИС, ПЛАНУВАЛЬНИК

Кваліфікаційна робота бакалавра присвячена розробці програмного застосунку для розрахунку робочого часу співробітників підприємства.

У першому розділі ми проаналізували предметну область, здійснили огляд існуючих підходів, а здійснили оцінку наявного на ринку програмного забезпечення. Визначено мету, завдання розробки проекту. Здійснено вибір архітектури та здійснено декомпозицію програмного застосунку для розрахунку робочого часу співробітників підприємства.

В кваліфікаційній роботі було продемонстровано розроблено базу даних, здійснено програмну розробку застосунку, здійснено тестування, послідовність виконання тестування, для перевірки відповідності програмного застосунку зазначеним функціональним вимогам, які наведено в технічному завданні до розробки програмного застосунку з розрахунку робочого часу співробітників підприємства. Наведено детально інструкцію користувача

9.06.2021 р

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.2101094.01.20. ПЗ	Пояснювальна записка	125		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4	КвРІПЗ.2101094.01.20. Е8	Логічна модель бази даних	1		
5	A4	КвРІПЗ.2101094.01.20.Е8	Структура класів	1		

КвРІПЗ, 2101094.01.20.ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Черкасов Д.О.	<i>[Signature]</i>	10.06
Керівник		Форкун Ю. В.	<i>[Signature]</i>	10.06
Н. Контр.		Яшина О.М.	<i>[Signature]</i>	10.06
Зав. Каф.		Бедратюк Л.П.	<i>[Signature]</i>	10.06

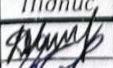
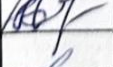


Програмний застосунок для розрахунку робочого часу співробітників підприємства
Пояснювальна записка

Лім.	Арк.	Аркушів
	1	1

ХНУ, ІПЗ-21-1

ЗМІСТ

ВСТУП	8
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	10
1.2 Аналіз останніх публікацій, досліджень та існуючих рішень.....	12
1.3 Аналіз наявного програмного забезпечення	17
1.4 Постановка задачі та перелік задач для реалізації	20
1.5 Висновки по розділу	21
2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОЇ СИСТЕМИ	22
2.1 Вибір типу архітектури та зразків проектування	22
2.2 Опис декомпозиції	24
2.3 Опис залежностей	28
2.4 Висновки по розділу	29
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ	30
3.1 Розроблення бази даних програмного застосунку	30
3.2 Розроблення програмних модулів програмного застосунку.....	32
3.3 Керівництво користувача програмного застосунку.....	35
3.4 Технічні характеристики програмного застосунку.....	42
3.5 Тестування програмного застосунку	44
3.6 Висновки розділу програмної реалізації та тестування програмного застосунку	52
ВИСНОВКИ	53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	54
ДОДАТКИ.....	58

КвРІПЗ. 2101094.01.20.ПЗ									
Змн.	Арк.	№ докум.	Підпис	Дата	Програмний застосунок для розрахунку робочого часу співробітників підприємства Пояснювальна записка	Лім.	Арк.	Аркуші	
				1.06.				1	1
				10.06					
				10.06					
				10.06					
Виконав		Черкасов Д.О.				ХНУ, ІПЗ-21-1			
Керівник		Форкун Ю. В.							
Н. Контр.		Яшина О.М.							
Зав. Каф.		Бедратюк Л.П.							

ВСТУП

Перед усіма керівниками підприємств та організацій, пов'язаних з розробкою та виробництвом, часто постає питання, чи варто братися за те чи інше замовлення? Буде це вигідно організації і співробітникам? Без урахування ситуації, при якій продукт чи проект просто не може бути реалізований, витрати на його розробку будуть досить високими, і навіть прибуток від продажу їх не відшкодує збитки. Зазвичай, якщо компанія ще досить молода, легко можна розрахувати ціну для кожного проекту загалом, оскільки цих проектів досить не багато. Але якщо уявити собі досить велику компанію з великою кількістю замовлень – то це досить складна задача. Ручний розрахунок кожного задання для кожного проекту майже неможливий. Тому краще за все розробити систему, яка автоматично розраховує час затрачений співробітниками, додаткові витрати на різні матеріали та обладнання, що виникають в процесі роботи, і буде зберігати інформацію про проекти і окремі завдання в базі даних програмного застосунку для подальшого використання в своїй роботі.

Також досить складно визначити заробітну плату для кожного співробітника підприємства, яка виплачується, як погодинна заробітна плата. Оскільки неможливо керівництву точно визначити, скільки часу кожен співробітник витрачає на роботу над певним проектом. Розумним рішенням є використання цього процесу в тій же самій системі, яка обчислює вартість проекту і його окремих завдань.

Розглядаючи ці ключові аспекти, дана розробка має на меті надати цінну інформацію, а також комплексну основу, як для приватних осіб і організацій, які прагнуть створити свій успішний бізнес, що спеціалізується на використанні часових затрат в оплаті праці своїх співробітників.

Актуальність теми кваліфікаційної роботи обумовлена досить високим зростанням попиту на такі рішення для розрахунку робочого часу співробітників підприємства, що особливо відчутно в умовах переходу організацій до дистанційної форми їх роботи.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Програмний застосунок, який розробляється в межах даної кваліфікаційної роботи, має на своїй меті не лише автоматизувати процес обліку та розрахунку робочого часу та оплати праці співробітникам, але й створити передумови для підвищення якості роботи співробітників.

Мета нашої кваліфікаційної роботи полягає в проектуванні та розробці програмного застосунку для розрахунку робочого часу співробітників підприємства, яка має мати зрозумілий і зручний інтерфейс, для різних груп користувачів: управнців, менеджерів та працівників.

У роботі охоплено, як теоретичні аспекти створення таких програмних засобів, але і практичні кроки з їх реалізації та впровадження, що дозволить розробити потужну систему для розрахунку робочого часу співробітників підприємства. Сучасні тенденції розвитку ІТ показують, що такі застосунки відіграють велику роль в бізнесі, створюючи інновації та роблять організації конкурентоспроможними на ринку.

Отже кваліфікаційна робота спрямована на розробку та впровадження додатку, який має відповідати потребам організацій. Отримані результати нашої роботи можуть бути застосовані в реальному житті для розрахунку робочого часу співробітників підприємства.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		9

1 Дослідження предметної області

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Важливим фактором успішної комерційної діяльності підприємства у виробничій сфері є аналіз його економічної ефективності не тільки на самому підприємстві в цілому, але і на окремих робочих місцях для кожного співробітника. Проект надає методологію розрахунку робочого часу співробітників підприємства по всьому ланцюгу роботи співробітників над проектами та окремими завданнями.

Заробітна плата працівників співробітників є одним з найважливіших факторів, що визначають собівартість продукції яка вироблена підприємством. Однією з найважливіших областей економіки підприємства є розрахунок заробітної плати для співробітників. З цим аспектом економіки підприємства стикається не тільки керівництво, а й весь персонал цього підприємства. Звичайно, всі хочуть проводити свої розрахунки та мати від цього прибуток. Перепоною до цього є висока складність розрахунку робочого часу співробітників підприємства в компаніях зі значною кількістю співробітників. Досить природно, що багато керівників, які здійснюють розрахунок робочого часу співробітників підприємства та заробітну плату співробітникам, хочуть полегшити свою роботу і збільшити свій дохід, перенісши свою тяжку повсякденну роботу на автоматизовані технології обробки інформації.

Наведемо на прикладі того, як може виникнути додана вартість продукції певного проекту. Припустимо, що організація отримує партію замовлень на матеріали, які необхідні для виготовлення деяких запчастин та здійснює їх продажі. Далі, після того, як матеріал був доставлений на саме підприємство, він обробляється декількома співробітниками. Для того щоб ці співробітники могли обробляти даний матеріал, вони повинні бути підключені мати своє обладнання та

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>10</i>

засоби, а за ними повинні працювати інші співробітники. Вартість затрат, яка йде на працю співробітників, і їх заробітна плата співробітників розраховується за певними правилами виходячи з можливостей кожного такого співробітника, і ще інших затрат. Крім того, адміністрація не повинна випускати з уваги певні формажори, такі як, наприклад, несправності техніки та обладнання, які також включені в цю додаткову вартість.

Існує дві основні форми нарахування заробітної плати - погодинна і за ставкою. На підприємстві, де була апробована дана робота, використовувалась погодинна оплата праці, а зарплата співробітника підприємства розраховується за встановленими нормами, та оплатою праці за фактично відпрацьований час. Без належного ведення табеля обліку фактичного робочого часу співробітника неможливо належним чином здійснити йому погодинну оплату праці. Метою даної кваліфікаційної роботи є впровадження такої системи обліку роботи, яка в залежності від співвідношення фактичного робочого часу затраченого співробітником часу, витраченого на одиницю конкретного виду продукції, виробленої на підприємстві, логічно буде вписуватись в систему обліку виникнення вартості на додаткові витрати на кожному робочому місці співробітника по всьому технологічному ланцюжку роботи на підприємстві, які спрямовані на поділ витрат на окремі завдання на д проектами. Це досить гарна ідея, щоб добре провести вільний час зі своєю родиною чи своїми. Це дозволить такому випадку вести самому бухгалтерський облік, який дозволяє використовувати створену базу трудомісткості при прогнозуванні термінів виконання і вартості виконання індивідуальних завдань та проектів.

Тому, враховуючи вищевикладені нами факти, необхідно створити програмний застосунок, який, спочатку, розраховує заробітну плату кожного співробітника в кінці кожного виконання проекту і буде розраховувати додаткові витрати, які певний продукт отримає в процесі роботи над ним співробітником та обладнанням на підприємстві.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>11</i>

1.2 Аналіз останніх публікацій, досліджень та існуючих рішень

Розглянемо проект, який має схожу до нашого завдання концепцію і складається з наступних частин:

- створення замовником кадрового порталу співробітників , названого “ForumStaff portal”;
- керівництво та вдосконалення системи управління персоналом підприємства, реалізація завдань з управління оплати праці персоналу;
- website.ua надає автоматичне перетворення отриманого резюме із завантаженням у свою базу даних;
- розвиток існуючих систем оплати праці;
- визначає інтерфейс, який цей портал використовує для отримання даних від своїх користувачів.

Кадровий портал є внутрішнім для організації і використовується для організації своєї внутрішньої комунікації та оперативної взаємодії між співробітниками. Це є сучасний і дуже ефективний інструмент, який допомагає структурувати робочий процес на підприємстві та організувати доступ до інформації, що зберігається в системі управління кадрами. Портал можна використовувати для швидкого вирішення багатьох різних адміністративних питань. Зокрема, своєчасно повідомляти співробітників, або ж негайно отримувати з ними зворотний зв'язок. На такому порталі можуть працювати співробітники підприємства різних підрозділів, а також на всіх рівнях управління підприємства.

Інші схожі застосунки були перенесені наступним чином:

- перенесена структура порталу на персональні веб-сторінки різних організацій;
- методи нарахування заробітної плати, в тому числі фонд оплати праці співробітників, облік їх робочого часу, а також штатний розклад з усіма доступними вакансіями;

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						12
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- форми для кожного працівника з даними про їх день народження, телефон та місцезнаходження та проживання;
- система здійснення пошуку, де можна знайти усю інформацію про співробітників у даній в системі;
- інформація про способи оплати, банки, правила, накази та інша необхідна інформація, яку повинні знати та розуміти усі співробітники організації;
- звіти спеціально тут створюються для того, щоб певна група користувачів (в основному менеджменту та керівників) могла працювати з усіма даними про відсутність певного співробітника (відпустки, відрядження, лікарняний, хвороба, журнал прийому/звільнення співробітника);
- інформація про зарплатну плату та її нарахування лист (в даний час у співробітників є можливість зайти на портал і переглянути зарплатний лист за будь-який місяць). При перегляді платіжної відомості користувач отримує додаткові дозволи і підвищує конфіденційність даних);
- таблиць співробітника;
- у режимі пошуку можна навіть отримати інформацію про заяви, заяви-анкети, правила прийому, шаблони правил для оформлення будь-якого корпоративного процесу;
- зміна та оновлення персональних даних самими користувачами порталу;
- графік роботи;
- створення бази резюме співробітників;
- автоматизація обробки вхідних резюме потрібних співробітників.

Доступ до таблиці співробітників був строго розділений за певними повноваженнями: керівник організації переглядав всі дані в таблицях співробітників, а начальник відділу тільки про підлеглих. Така ж схема використовувалася при всі відкритті вакансій. На основі вакансій, представлених у звіті, можна подати окрему заявку на пошук співробітників та працівників. Тобто можна їх редагувати та заповнювати. Заявка проходить певну процедуру затвердження і надходить у відділ підбору та роботи з персоналом, де її розглядають

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		13

і передають рекрутерам, які здійснюють пошук нового співробітника. Ці функції описані на схемах, показаних на рис. 1.2.1. і 1.2.2.

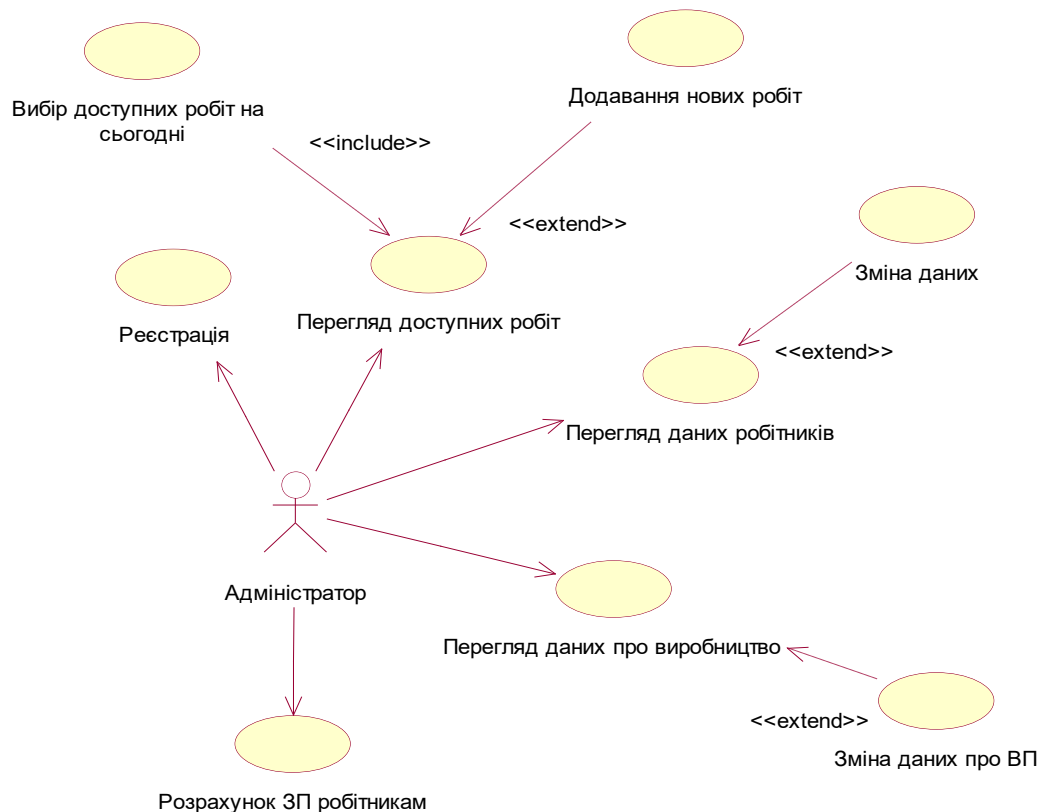


Рисунок 1.2.1 – Діаграма діяльності роботи менеджменту підприємства системи

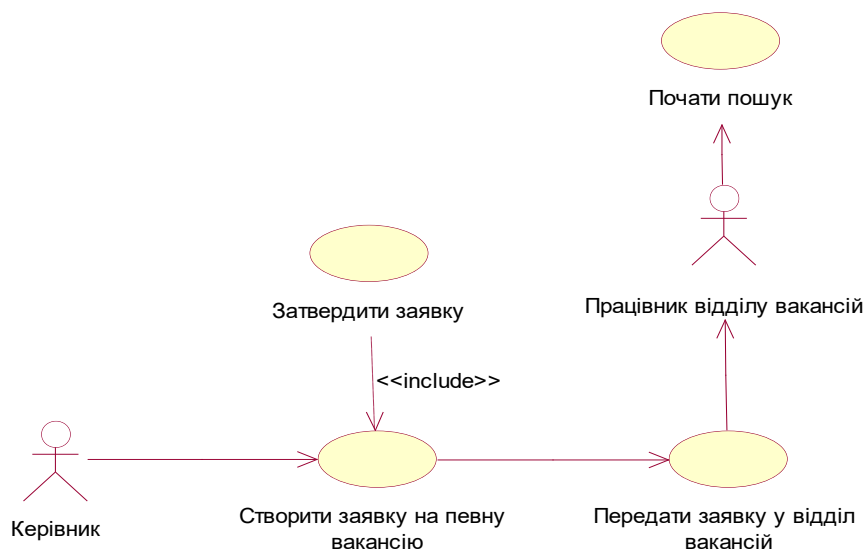


Рисунок 1.2.2 – Діаграма пошуку співробітника за вакансіями

Були реалізовані два варіанти пошуку - простий (для звичайних користувачів) і розширений (для груп людей з розширеними правами), різниця полягає в обсязі наданих даних. Простий варіант пошуку дозволив мені здійснити пошук за деякими спеціальними критеріями, а розширений варіант пошуку дозволяв здійснювати пошук за деякими спеціальними критеріями, як нарахування заробітної плати.

Необхідно також проінформувати відділ обслуговування персоналу про необхідність підготовки звіту, який буде відповідати певним правилам за певними обставинами, наприклад, у разі звільнення співробітника, потрібно в результаті чого анулювати раніше надані права доступу. Ну, а як тільки звільнення співробітника реєструється в кадровій системі, то фахівець з безпеки, перевібивши усі лог-файли, має підтвердити, що є нове звільнення, але не збирає такі дані.

Співробітники можуть переглянути табель роботи за будь-який період, щоб дізнатися, наскільки точно була його розрахована зарплата .

Зміни та оновлення деяких персональних даних самими користувачами порталу є досить інноваційною функцією, яка сильно відрізняє цей проект від усіх попередніх таких проектів. Клієнт повинен зосередитися на впровадженні такого механізму, посилити кадровий склад і бути зацікавленим в оновленні персональних даних кожного співробітника.

Крім того, співробітники можуть змінити інформацію про себе: особисті дані, інформацію про родичів, свою освіту, свою фото тощо. Недосконалість цього процесу - дуже складне і актуальне питання в багатьох організаціях, оскільки вся інформація ретельно перевіряється та вноситься в систему при прийомі співробітника на роботу. В майбутньому інформація рідко відстежується, аж до тих пір, поки співробітник не надасть нові документи, такі як свідоцтво про шлюб, про зміну прізвища тощо. Однак інша інформація, така як освіта, народження дитини і тощо, оновлюватися відповідно не буде. Тому впроваджено систему, яка дозволяє співробітникам в першу чергу переглядати особисту інформацію, а в другу – вже управляти нею та, відповідно змінювати.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						15
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Інше портальне рішення було побудовано на базі системи bos-HR[1]: формування табелів обліку робочого часу для відділів. Є певний набір відділів, в яких призначено відповідальну особу, за кожною особою з яких умовно закріплені «хронометрист» та «дублер» – відділ, який об'єднує співробітників та працює з ними, коли відповідальна особа на роботі відсутня на своєму робочому місці. Відповідальним за табель нарахування заробітної плати може бути призначений будь-який співробітник з економічною освітою, і немає ніяких обмежень на приналежність до якогось певного відділу. Це корисно з наступної точки зору: наприклад в банку «Приват» більше 3000 відділів, що з точки зору менеджменту досить багато. І тут потрібні були кошти для менеджменту та керівництва. Процедура агрегування полягає в наступному: фахівці з персоналу, що відповідають за табелі обліку робочого часу для організацій в цілому, в системі BOSHR, співробітники відділу кадрів починають формувати плановий табель для обліку робочого часу, після чого приступають до самої процедури складання табелів обліку робочого часу співробітників та табелів обліку робочого часу для табелів обліку робочого часу для табелів обліку робочого часу для розкладів тощо. Відповідальний за розрахунок заробітної плати отримує електронний лист про необхідність нарахування за певний період часу (авансовий платіж, або ж зарплату). Якщо на момент відправлення йому листа відповідальна особа відсутня на робочому місці (а неявка реєструється в табелі успішності), то цей лист отримує «дублер». Представник виконує агрегацію і закриває підготовлений табель обліку робочого часу співробітника. Начальник платіжного відділу організації підтверджує дотримання графіка роботи всіх відділів організації. Якщо який-небудь відділ організації не буде зачинений, відповідальний співробітник зможе швидко оцінити ситуацію і вжити негайних заходів. Також є можливість відправити лист-нагадування відповідальному «хронометристу». В результаті такого рішення він значно розширює усі можливості агрегації системи BOS-HR.

					КвРІПЗ.210194.01.20 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Однією з ресурсномістких щоденних завдань є обробка журналів робіт. Керівнику потрібно прочитати і обробити кожен, але вибрати зберегти найбільш цікаві для використання в організації. В автоматизованій обробці резюме до існуючих функцій співробітника вносяться систему всі раніше накопичені резюме і створили електронне сховище з функціями пошуку. В результаті впровадження в систему можна обробити тисячі резюме, для кожного з яких може бути створена картка співробітника з основними їх даними. А також може бути вбудований механізм автоматичної обробки для резюме, що надходить месенджерами чи по електронній пошті. При отриманні кожного листа в системі автоматично створюється запис про претендента, додаються основні дані і оригінал резюме, а також перевіряються дублікати в базі даних.

1.3 Аналіз наявного програмного забезпечення,

Проведемо аналіз наявного на ринку програмного забезпечення та оцінимо їх оцінку за такими критеріями оцінки по 10-бальній шкалі, яке застосовується до оцінки такого програмного забезпечення:

- Функціональність – наскільки повно покриває потреби обліку часу;
- Зручність інтерфейсу – інтуїтивність і легкість користування;
- Мобільність – наявність мобільного застосунку;
- Інтеграції – з іншими сервісами (Slack, Google Workspace, тощо);
- Аналітика – звіти, візуалізація, аналітика часу;
- Гнучкість налаштувань – адаптація до специфіки підприємства;
- Ціна/можливості – співвідношення вартості до функціоналу.

1.3.1 Toggl Track - простий та ефективний тайм-трекер для команд та фрілансерів, який вдало підходить для обліку проектного часу.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.1 - Оцінка Toggl Track

Критерій	Оцінка
Функціональність	8
Зручність інтерфейсу	9
Мобільність	10
Інтеграції	9
Аналітика	8
Гнучкість налаштувань	7
Ціна/можливості	8
Безпека даних	9

1.3.2 Clockify - безкоштовна система обліку робочого часу з можливістю платного апгрейду. Серед основних переваг - має досить зручний дашборд.

Таблиця 1.2 - Оцінка Clockify

Критерій	Оцінка
Функціональність	9
Зручність інтерфейсу	8
Мобільність	9
Інтеграції	8
Аналітика	9
Гнучкість налаштувань	8
Ціна/можливості	10
Безпека даних	8

1.3.3 Hubstaff – досить потужний трекер з моніторингом активності, GPS, скріншотами — ідеально підходить для віддалених команд.

Таблиця 1.3 - Оцінка Hubstaff

Критерій	Оцінка
Функціональність	10
Зручність інтерфейсу	7
Мобільність	8

Інтеграції	9
Аналітика	9
Гнучкість налаштувань	9
Ціна/можливості	7
Безпека даних	9

1.3.4 RescueTime – автоматичне відстеження активності користувача за допомогою ШІ — аналіз продуктивності

Таблиця 1.4 – Оцінка RescueTime

Критерій	Оцінка
Функціональність	7
Зручність інтерфейсу	8
Мобільність	8
Інтеграції	7
Аналітика	10
Гнучкість налаштувань	6
Ціна/можливості	7
Безпека даних	8

1.3.5 Time Doctor – контроль часу, скріншоти, відвідуваність, звіти. Орієнтований на великі та середні команди.

Таблиця 1.5 – Оцінка Time Doctor

Критерій	Оцінка
Функціональність	9
Зручність інтерфейсу	7
Мобільність	9
Інтеграції	8
Аналітика	8
Гнучкість налаштувань	8
Ціна/можливості	8
Безпека даних	7

Як бачимо з наведених прикладів, кожна така система має свої переваги та недоліки, тому нами було прийнято рішення про створення власного програмного застосунку

1.4 Постановка задачі та перелік задач для реалізації

Отже, резюмуючи все вищесказане, можна сказати, що метою даного проекту є створення програмного застосунку, яка буде виконувати наступні завдання:

1. зберігати інформацію про співробітників, що працюють в компанії (номер, прізвище, посаду, контактні дані, проекти):
2. зберігає інформацію про всі завершені проекти на підприємстві, тобто наступну інформацію про них:
 - назва проекту;
 - вартість самого проекту;
 - час, витрачений на цей проект та окремі задачі;
3. дозволяє співробітнику побачити всі можливі проекти і задачі та вибрати ті, які йому цікаві і потрібні;
4. менеджер може переглядати всі доступні проекти та завдання та додавати нові за потреби чи вимогою;
5. менеджер може перевірити всі коефіцієнти для зарплати для всіх співробітників і змінити відповідно їх у міру необхідності;
6. розрахувати додану вартість проектів та продуктів, що відображається в табелях та відомостях про заробітну плату;
7. відповідальний може розрахувати зарплату працівника, а також час, проведений на роботі, виходячи з його коефіцієнта ефективності.

Ці можливості системи можна представити у вигляді діаграм варіантів використання, які можете побачити їх розділі 2.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						20
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1.5 Висновки по розділу

Отже, в даному розділі ми проаналізували предметну область, здійснили огляд існуючих підходів, а здійснили оцінку наявного на ринку програмного забезпечення. Також, ми визначили всі цілі, всі завдання, які повинні бути виконані в цьому проекті. Ми розглянули існуючі рішення цієї проблеми. Це допоможе в розробці структури та дизайну нашого проекту.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

2 Розробка архітектури програмної системи

2.1 Вибір типу архітектури та зразків проектування

На початку проектування нашого програмного продукту необхідно вибрати архітектуру для нього. З огляду на особливості наших програмних застосунків, ми можемо сказати, що нам підходить такі архітектури програмування:

- архітектура незалежних компонентів;
- архітектура репозиторію.

Так, архітектура незалежних компонентів застосунку складається з таких компонентів, які працюють паралельно (теоретично) і іноді взаємодіють. Розглянемо архітектуру клієнт-сервер, яка саме є однією з підкатегорій цієї архітектури.

1. Архітектура клієнт-сервер - це один із шаблонів архітектури програмного забезпечення. Це включає наступні ключові компоненти:

- набір серверів, які надають інформацію або інші послуги додатку;
- набір клієнтів, що використовують послуги, що надаються сервером;
- мережа, яка забезпечує взаємодію між клієнтом і сервером.

Сервери незалежні один від одного. Клієнти також працюють незалежно один від одного і паралельно. Суворої прив'язки клієнта до сервера не існує. Серверу не властиво обслуговувати одночасні запити від різних клієнтів. Клієнт, з іншого боку, має доступ до певного сервера. Клієнт повинен знати про доступні сервери, але він може не знати про присутність інших клієнтів.

Ця архітектура підходить для даного випадку, оскільки вона надає системі єдиний сервер для обробки запитів клієнтів, наприклад, для перегляду всіх доступних замовлень, які можуть бути виконані, або для перегляду поточних звітів про роботу співробітників.

Наступна архітектура, яка використовується в нашому проекті, - це Архітектура зберігання. Суть його полягає в тому, що всі дані, використовувані

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		22

підсистемою, зберігаються в центральній базі даних. У той же час, оскільки сховище саме по собі є пасивним елементом, вся відповідальність за нього лежить на підсистемі. Архітектура сховища є правильним вибором, оскільки в системі обробляється і зберігається великий обсяг інформації.

Структура архітектури Application Server розробленого нашого програмного застосунку, наведено на рисунку 2.1.1, та має три основні частини: клієнтську частину, сервер додатків та сервер бази даних.

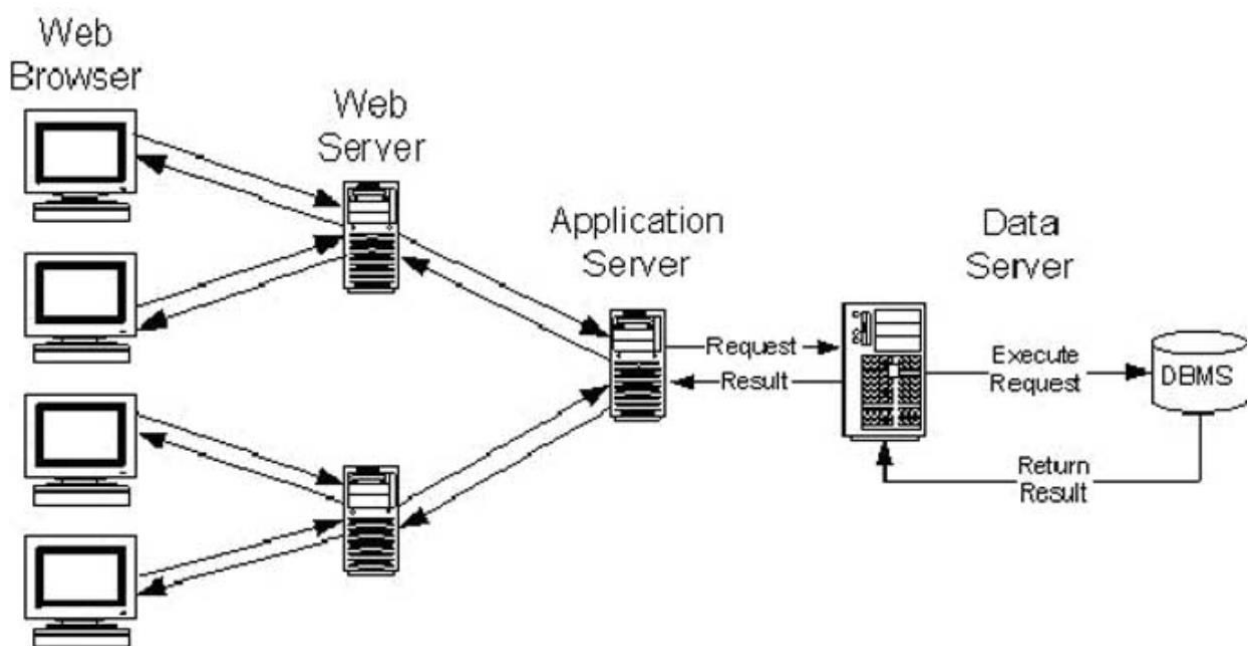


Рисунок 2.1.1. – Схема архітектури Application Server програмного застосунку

Тут клієнтська частина архітектури призначена для взаємодії з користувачем та кінцевим додатком відображення (браузер), який забезпечує відображення інтерфейсу для роботи з сервером додатків, який в свою чергу працює сервером бази даних .

Сервер додатків системи отримує запити від клієнта, обробляє їх та передає дані, а робить запит на дані в сервера бази даних. Сервер додатків відправляє запити на дані, так серверу бази даних. Такі типи архітектур мають назву

багаторівневих та складаються з трьох-чотирьох рівнів. Четвертий рівень відповідає більше за обробку з'єднання між іншими компонентами, зокрема клієнтом браузером, СКБД та сервером додатків. Вигода полягає в тому, що кілька веб-серверів можуть підключатися до одного сервера додатків, тим самим, збільшуючи обробку більшого числа одночасно підключених користувачів.

2.2 Опис декомпозиції

Давайте поглянемо на наш проект. По суті, цей проект включає в себе 2 великих процеси і виконує його декомпозицію. Тобто процес розрахунку робочого часу і заробітної плати користувача на основі даних, отриманих в ході цього процесу, і додаткової вартості, яку матеріал набуває в процесі виробництва. Далі давайте розділимо ці 2 процеси та створимо їх діаграму IDEF0.

Почнемо з процесу нарахування заробітної плати на основі часу, який працівник проводить у компанії, і цього часу. Зарплата в нашому випадку залежить від того, як довго пропрацював співробітник, і від коефіцієнта його професіоналізму. Фактор професіоналізму-це показник, який визначає, чи залежить заробітна плата працівника від його робочого часу. Тобто, якщо звичайний робітник працює 1 день по 24 години на добу, то майстер володіє високою кваліфікацією, тому таким чином ви зможете працювати краще і швидше, ніж Майстер, і будете отримувати менше грошей протягом тих же 24 годин. Через це його коефіцієнт вище, ніж у звичайного співробітника.

Отже, процес нарахування заробітної плати виглядає наступним чином: спочатку в базі даних створюється запит для пошуку робочого часу конкретного співробітника. Пошук здійснюється по всіх підрозділах Міністерства внутрішніх справ, де він працював. Після цього весь час підсумовується, і розрахунок заробітної плати починається безпосередньо з використанням коефіцієнта

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		24

професіоналізму, який також зберігається в базі даних. Потім адміністратору надсилається перша інформація про заробітну плату. Як показано на рис. 2.1 і рис. 2.2, давайте створимо діаграму IDEF0 для цього процесу.

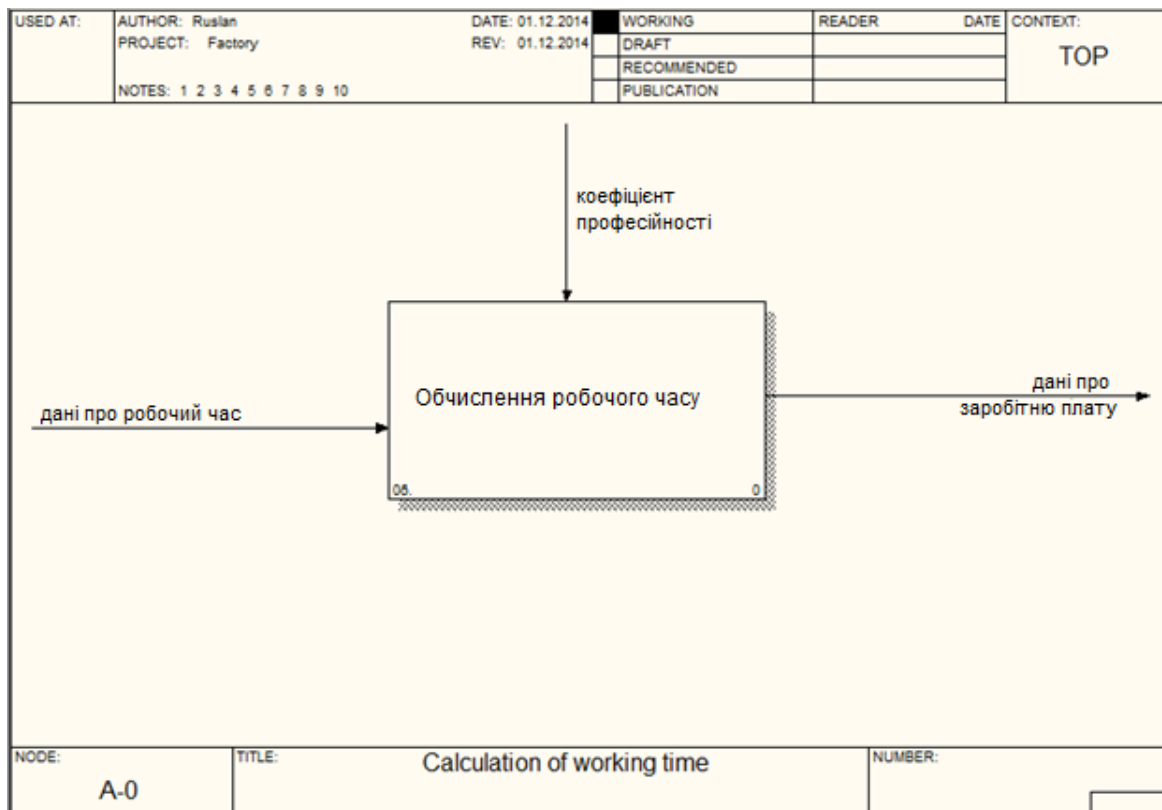


Рисунок 2.2.2 – IDEF0-діаграма обчислення робочого часу для першого рівня

Далі розглянемо сам процес розрахунку доданої вартості, яку матеріал набуває тут в процесі виробництва. Наприклад, для виготовлення металевих пластин нам потрібні кріпильні елементи, які утримують їх у купі разом, тому цей процес вступає в силу, як тільки матеріал надходить в певний цех. Але головним фактором є верстат, на якому обробляється матеріал.

Щоб обробити певну кількість матеріалу, верстат необхідно підключити до основної лінії і змастити. Також нам необхідно виплачувати заробітну плату співробітникам, які працюють на цьому верстаті. Тому нам, для того, щоб визначити кінцеву вартість продукту, система повинна зібрати інформацію від усіх підрозділів

МДВ і додати їх до первісної вартості. Для цього з бази даних береться інформація про час, що витрачається на одиницю роботи конкретної машини, з точки зору енергоспоживання та інших факторів, таких як мастило, лакофарбове покриття тощо, після чого визначається, хто саме працював на цій машині, коефіцієнт його професіоналізм береться з бази даних, і на цій основі визначається додаткова вартість, пов'язана із зарплатою співробітника. Крім того, після цього, усі ці дані підсумовуються і з урахуванням всіх додаткових витрат визначається кінцева вартість продукту.

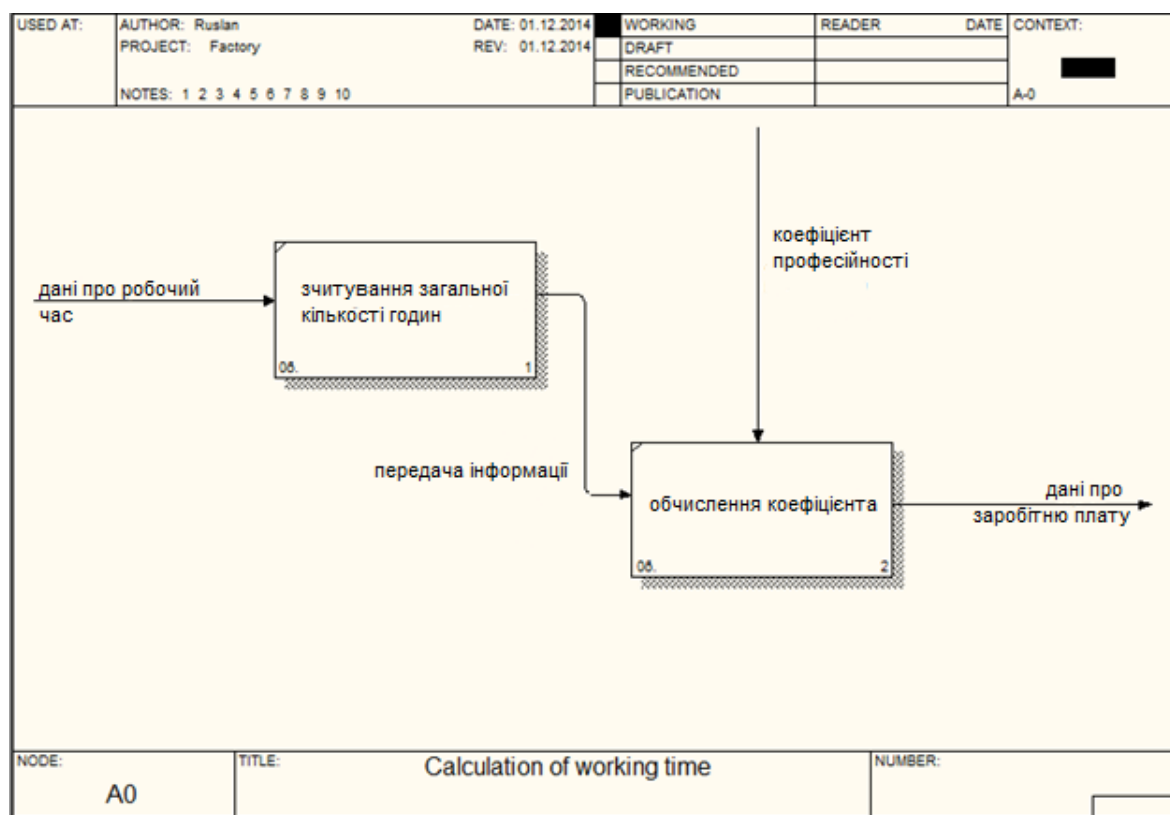


Рисунок 2.2.3 – IDEF0-діаграма обчислення робочого часу для другого рівня

Потім система перевіряє, чи є в базі даних запис про цей товар. Якщо записів немає, інформація про цей товар додається до бази даних. Це робиться для того, щоб ви могли отримати доступ до бази даних і знайти потрібну інформацію, якщо вам потрібно знову розрахувати додану вартість цього товару. Давайте

створимо діаграму IDEF0 для аналізу декомпозиції цього процесу. Дана декомпозиція відображена на рисунках 2.3 і 2.4.

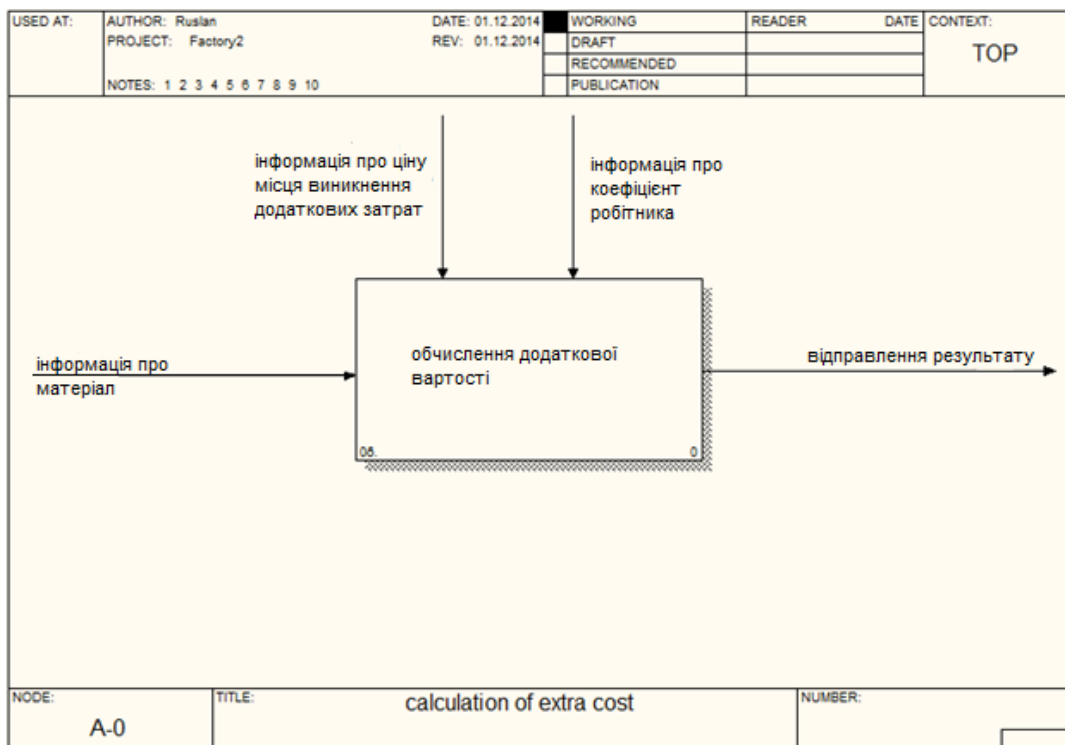


Рисунок 2.2.4 – IDEF0-діаграма нарахування додаткової вартості робіт першого рівня

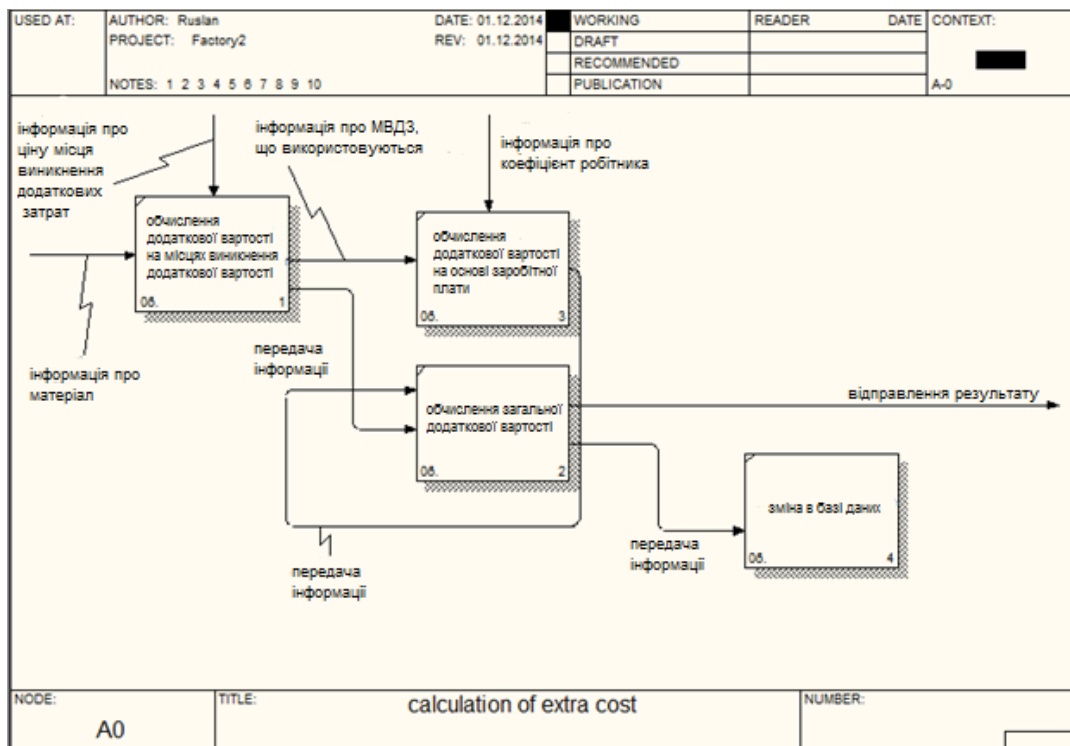


Рисунок 2.2.5 – IDEF0-діаграма нарахування для додаткової вартості робіт для першого рівня

Змн.	Арк.	№ докум.	Підпис	Дата

КвРІПЗ.210194.01.20 ПЗ

Арк.

27

2.3 Опис залежностей

Отже, після розбирання нашої системи ми розділили її на наступні такі основні програмні модулі:

- Модуль для вилучення усього робочого часу для співробітників з бази даних відповідно до заявленого ним;
- Модуль розрахунку заробітної плати для співробітників з урахуванням фактора їх професіоналізму;
- Модуль розрахунку загальної доданої вартості при передачі матеріалів до МРЧ і комунікацій;
- Модуль розрахунку загального часу вартості для заробітної плати працівників, які працюють у МРЧ;
- Модуль розрахунку загального часу вартості для матеріалів;
- Модуль, який вносить зміни в саму базу даних на основі даних, отриманих з модуля, який реалізує та обчислює загальну додану вартість матеріалу.

Цей модуль витягує з бази даних години роботи всіх співробітників і взаємодіє з самою базою даних в залежності від заявленого співробітника. Цей модуль вимагає певних даних, отриманих з цього модуля, тому він обчислює заробітну плату працівника за визначеним коефіцієнтом його професіоналізму. Неможливо розрахувати усю заробітну плату без урахування робочого часу співробітника, тому, за словами співробітника, для цього розрахунку потрібні дані з того модуля, який витягує з самої бази даних години роботи для всіх співробітників. Він не взаємодіє з іншими модулями системи.

Модуль розрахунку заробітної плати для співробітників з урахуванням фактора професіоналізму витягує з самої бази даних всі години роботи співробітників з вищевказаних причин, згідно заявленому співробітнику, і взаємодіє з користувачем, що викликав виконання цього модуля.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						28
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Модуль, який розраховує загальну додану вартість за проходження матеріалів через МРЧ, взаємодіє з введеними користувачем системи даними та інформацією, отриманою з самої бази даних. Ця інформація включає дані про додану вартість, яку приносить проходження матеріалів через МРЧ. Цей модуль взаємодіє з модулем, який обчислює усю загальну додану вартість заробітної плати для працівників, які працюють МРЧ, передає інформацію про МРЧ, через яке проходить даний продукт, тим самим відбираючи усіх працівників, заробітна плата для яких враховується у доданій вартості. Крім того, він надсилає також дані про додану вартість, отриману при передачі усіх матеріалів через МРЧ та комунікацій, та взаємодіє з самим модулем, який обчислює усю загальну додану вартість матеріалів, оскільки вона включена до цієї загальної суми.

Модуль розрахунку загальної доданої вартості для заробітної плати працівників, які працюють з МРЧ, взаємодіє з модулем для розрахунку загальної доданої вартості матеріалів, що проходять через МРЧ, та модулем для розрахунку загальної доданої вартості матеріалів. Принцип його взаємодії аналогічний до принципу взаємодії модуля, який розраховує загальну додану вартість за проходження матеріалів через МРЧ.

Модуль, який вносить зміни в саму базу даних на основі даних, отриманих з модуля, який обчислює загальну додану вартість для матеріалу, взаємодіє з самим цим модулем системи, між інформацією, яку необхідно ввести в саму базу даних, і самою базою даних, в яку ці дані були користувачем фактично введені.

Отже, ми перерахували усі модулі, що відносяться до цього проекту, усі залежності між ними. Я точно описав, як відбувається взаємодія між ними, як дані передаються з одного модуля в інший і як вони приймаються. Я визначив, які модулі починають працювати швидше, ніж інші, і які з них залежать від них.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		29

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ

3.1 Розроблення бази даних програмного застосунку

Проектування моделі бази даних у програмному застосунку розрахунку робочого часу співробітників базується на створенні такої структури, яка дасть змогу забезпечити ефективність зберігання та збереження і обробку даних, які необхідні для реалізації функціональних вимог до системи. База даних в нашому програмному застосунку необхідна для зберігання основних даних та містить такі відношення:

Employee – відношення з інформацією про співробітників, які працюють на підприємства;

Project – відношення з інформацією про проекти над яким працюють співробітники підприємства;

Client – відношення з інформацією про клієнтів, які є замовниками проектів;

Process – відношення з інформацією про стан роботи над проектом;

Report – відношення з інформацією про затрачений час роботи над проектом;

Таблиця 3.1.1 – Структура таблиці Employee бази даних програмного застосунку.

Назва поля	Тип даних	Ключ	Опис
id-employee	int	PK	Ідентифікатор співробітника
id_project	int	FK	Номер проекту
name_e	varchar		ППП співробітника
login	varchar		Логін для входу в систему
email	varchar		Електронна пошта
passw	varchar		Пароль

Таблиця 3.1.2 – Структура таблиці Project бази даних програмного застосунку

Назва поля	Тип даних	Ключ	Опис
id- project	int	PK	Номер проекту
id_client	int	FK	Ідентифікатор спіробітника
name_prj	varchar		Назва проекту
date_start	datetime		Дата початку роботи над проектом
date_end	datetime		Дата закінчення роботи над проектом

Таблиця 3.1.3– Структура таблиці Client бази даних програмного застосунку

Назва поля	Тип даних	Ключ	Опис
id- client	int	PK	Ідентифікатор клієнта
name_client	int		Ім'я клієнта
field_name_prj	varchar		Додаткові дані про клієнта
email_client	varchar		e-mail клієнта
phone_client	datetime		Телефон клієнта

Таблиця 3.1.4– Структура таблиці Report бази даних програмного застосунку

Назва поля	Тип даних	Ключ	Опис
id-report	int	PK	Ідентифікатор звіту
explosion	varchar		Опис звіту
id_employe	int		Ідентифікатор спіробітника
time_report	datetime		e-mail клієнта

Таблиця 3.1.4– Структура таблиці Process бази даних програмного застосунку

Назва поля	Тип даних	Ключ	Опис
id-process	int	РК	Ідентифікатор звіту
status	varchar		Ім'я клієнта
time_proc	int		Затрачений час на проект
day	date		День закінчення
id_report	datetime		Номер звіту

Фізична діаграма побудованої бази даних нашого проектного застосунку наведена на рисунку 3.1.1

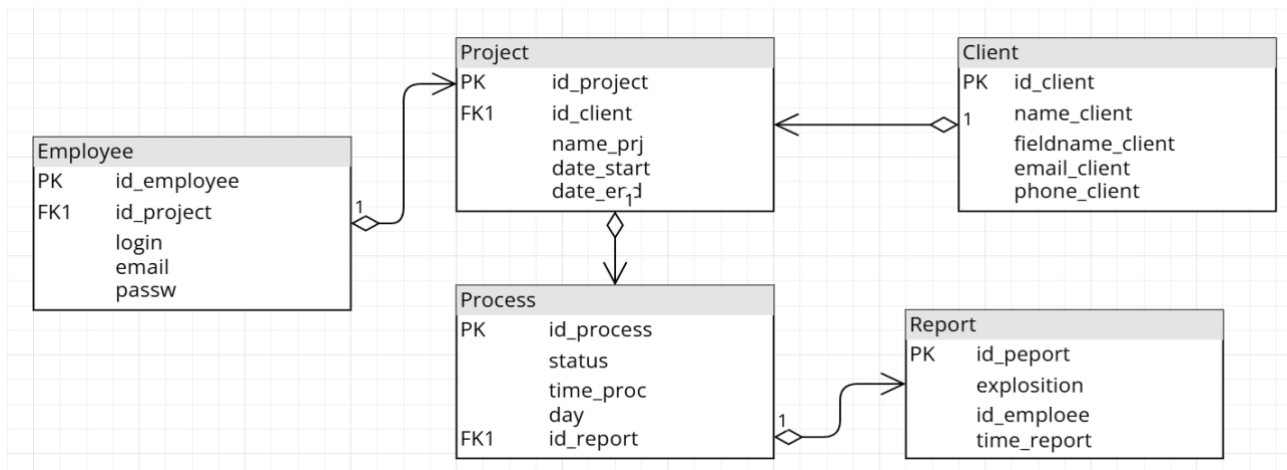


Рисунок 3.1.1 – Фізична модель бази даних

3.2 Розроблення програмних модулів програмного застосунку

Проектований програмний застосунок при проектуванні його модулів повністю розроблявся у чіткій взаємодії між клієнтом і сервером, їх взаємодію та взаємодією з базою даних.

В наведеній діаграмі діяльності нашого застосунку наведеній на Рисунку 3.2.1 наведено головний алгоритм роботи проектного програмного

забезпечення. На рисунку подано взаємодію найважливіших процесів та дій, які відбуваються в застосунку з для врахування затрат робочого часу її співробітників.

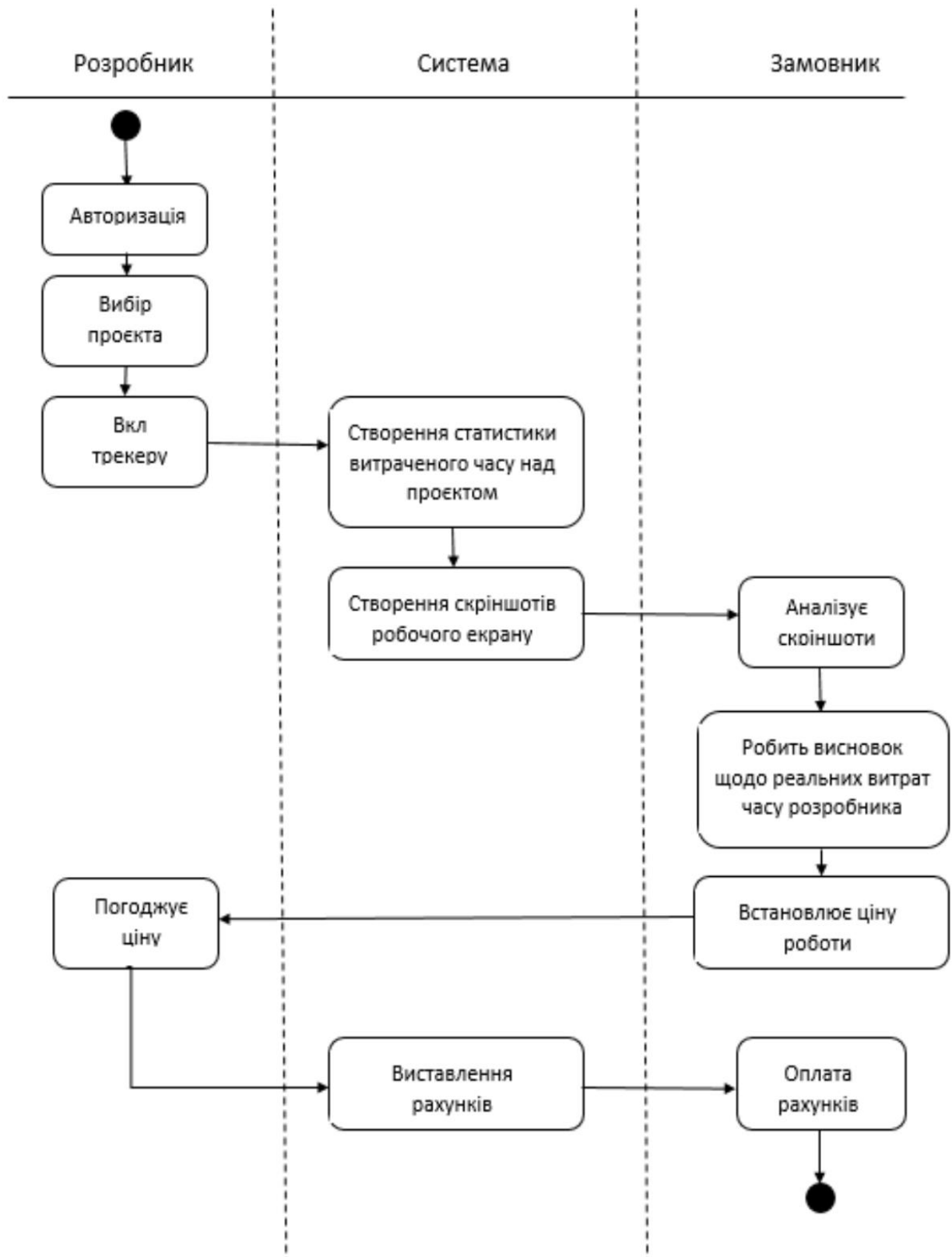


Рисунок 3.2.2 – Діаграма дільності застосунку

Змн.	Арк.	№ докум.	Підпис	Дата

Таким чином при формуванні класів системи та їх функціональних властивостей була врахована структура бази даних і відповідно було розроблено п'ять головних класів для розрахунку робочого часу співробітника організації (Рисунок 3.2.2)

Наведена діаграма класів дає наочне представлення для взаємодії з базою даних та показує, які властивості і поля класів вирішують технічне завдання для проекту з його функціональними можливостями для вирішення поставлених задач при розробці його архітектури

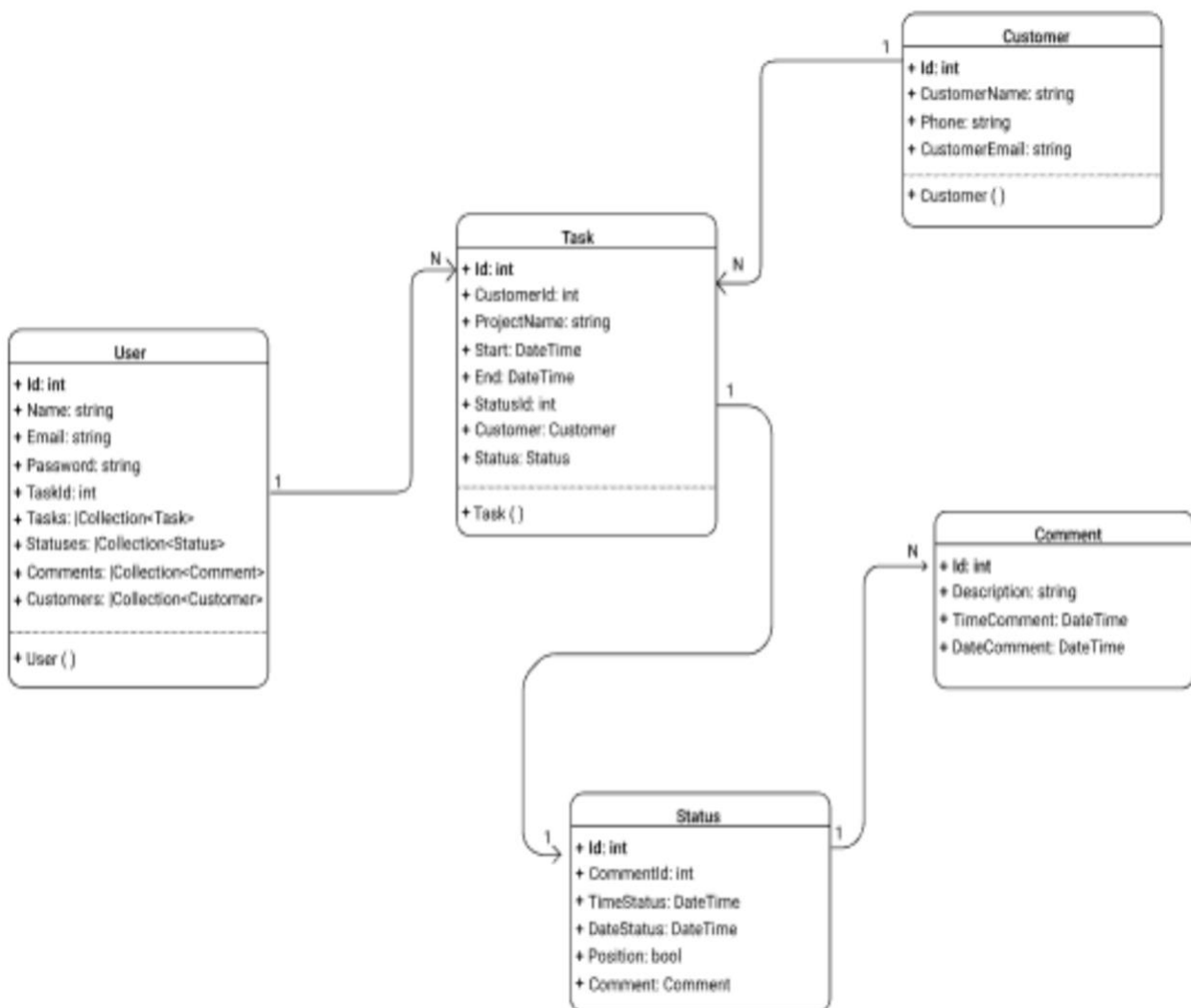


Рисунок 3.2.2 – Діаграма класів програмного застосунку розрахунку робочого часу співробітника

3.3 Керівництво користувача програмного застосунку

Для роботи з нашим програмним застосунком необхідно в браузері, на клієнтів перейти за адресою, де розміщена клієнтська частина застосунку (хостинг, або сервер) в рядку адреси. Після завантаження головного вікна застосунку у браузер користувачеві буде запропоновано зайти в застосунок, або зареєструватись у цьому застосунку (Рисунок 3.3.1).

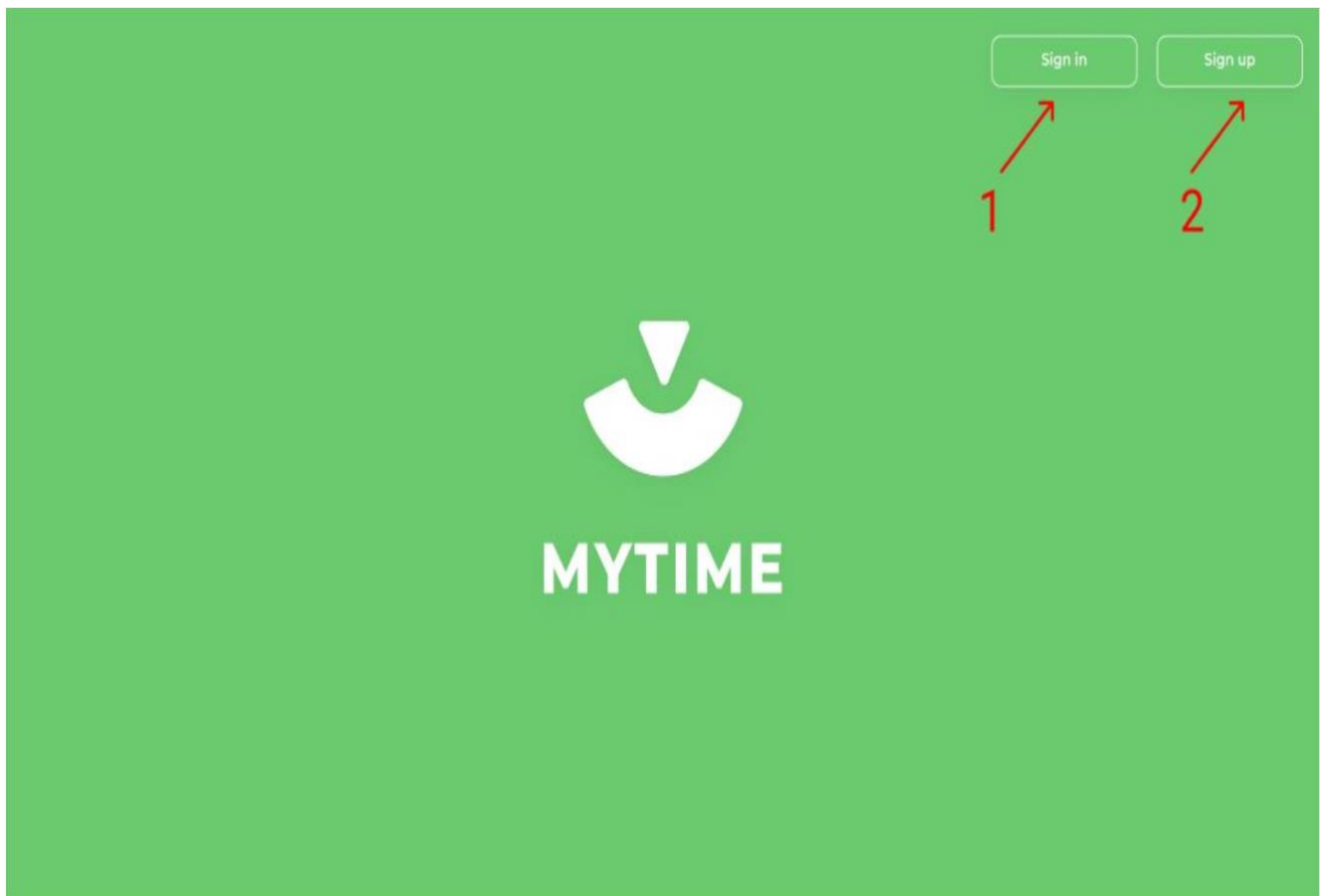


Рисунок 3.3.1 – Головне вікно програмного застосунку

Для незареєстрованих користувачів будуть доступні на формі два елементи застосунку системи, а саме:

- кнопка Увійти(1), при натисненні, на яку буде відображатись веб-сторінка з формою для входу в систему;

					КвРІПЗ.210194.01.20 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

– кнопка Зареєструватись (2) - при натисненні, на яку буде відображатись веб-сторінка з формою для реєстрації в нашій системі.

Відповідно, при натисканні кнопку 2, відбудеться перехід на форму для реєстрації в системі.

Форма вебсторінки реєстрації в систему наведена на рисунку 3.3.2

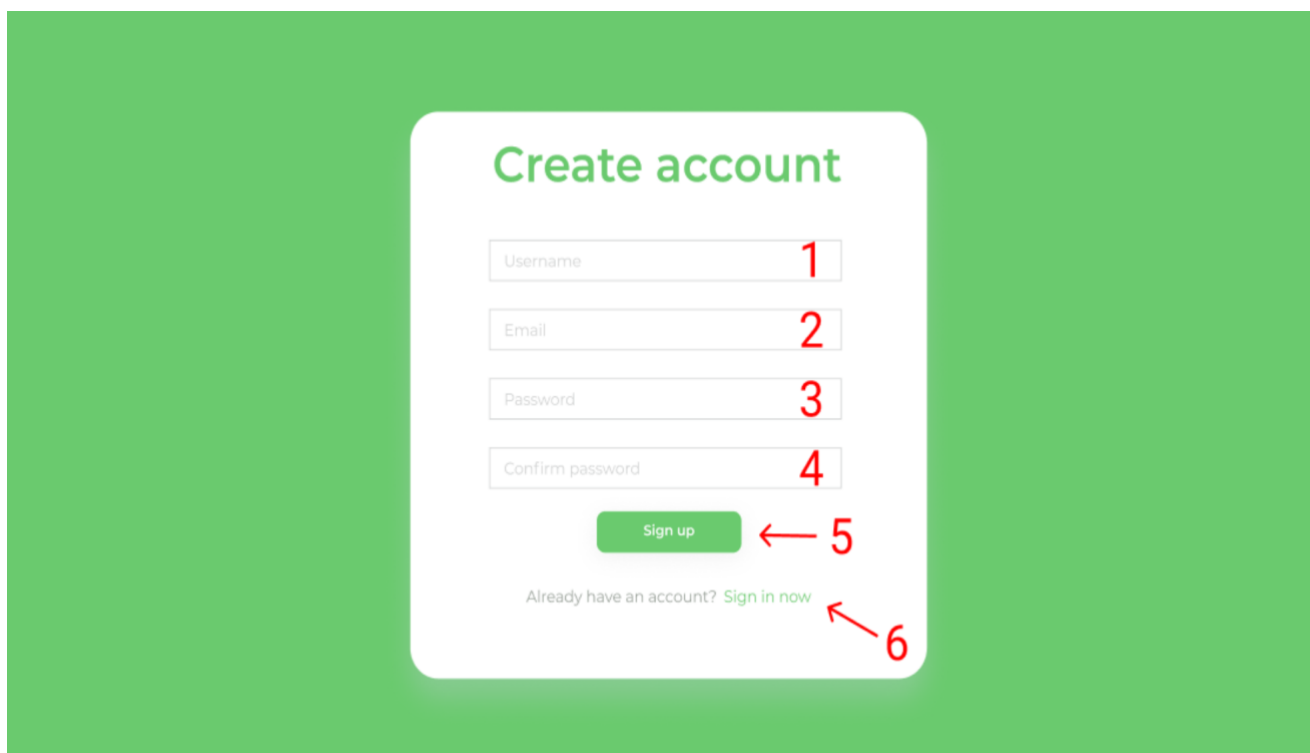
The image shows a 'Create account' form on a green background. The form is white with rounded corners and contains the following elements: a title 'Create account' in green; four input fields labeled 'Username', 'Email', 'Password', and 'Confirm password'; a green 'Sign up' button; and a link 'Already have an account? Sign in now'. Red arrows with numbers 1 through 6 point to these elements: 1 points to the Username field, 2 to the Email field, 3 to the Password field, 4 to the Confirm password field, 5 to the Sign up button, and 6 to the Sign in now link.

Рисунок 3.3.2 – Веб-форма для реєстрації в програмному застосунку

Форма реєстрації містить такі елементи:

1. Ім'я користувача – для введення в це поле імені нового користувача;
2. Пошта – для введення в це поле електронної адреси користувача;
3. Пароль – поле для введення паролю користувача;
4. Підтвердити пароль – поле для підтвердження введеного паролю користувачем;
5. Війти – кнопка для підтвердження реєстрації користувача у додатку;
6. Війти під іншим іменем – посилання для переходу на початкову сторінку авторизації.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						36
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Успішний процес реєстрації закінчується поверненням користувача на початкову сторінку програмного застосунку.

Форму веб-сторінки реєстрації користувача наведена на рисунку 3.3.3

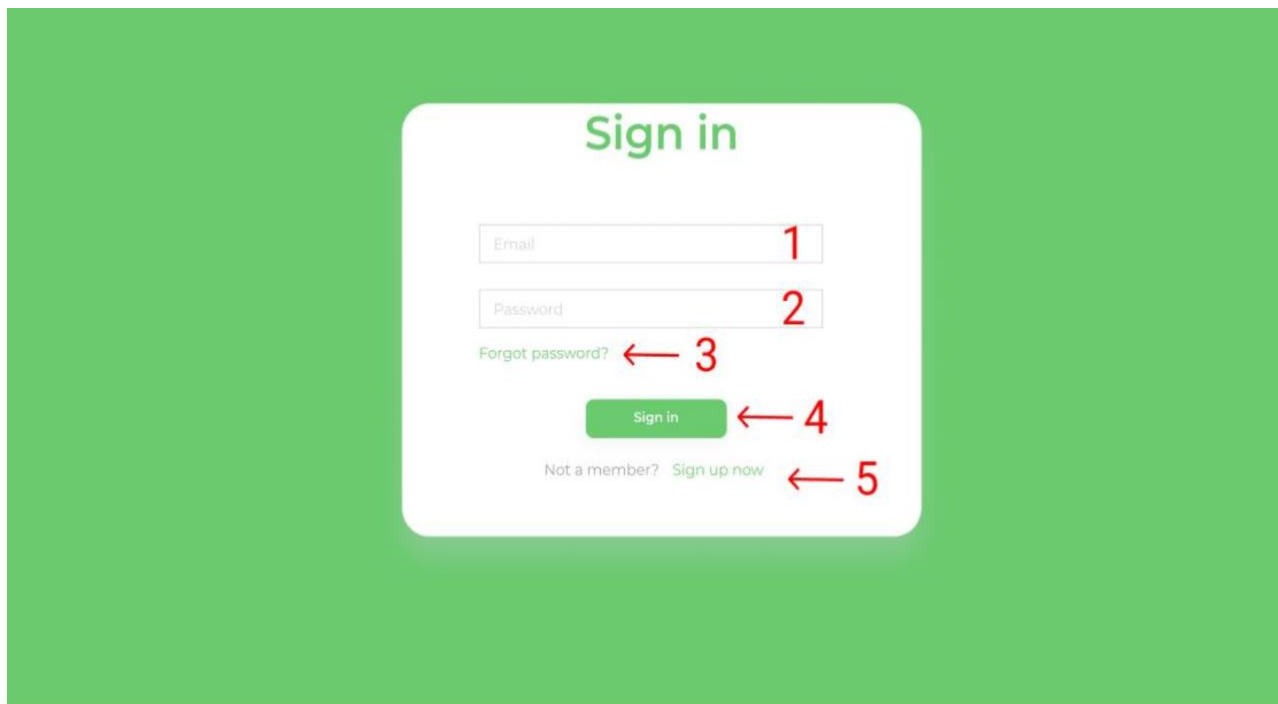


Рисунок 3.3.3 – Форма веб-сторінки реєстрації користувача

Дана форма містить та елементи на цій сторінці:

1. Пошта - поле для введення електронної адреси користувача системи;
2. Пароль – для введення пароля користувача системи;
3. Повернути пароль – посилання, при натисненні на яке відбувається підтвердження пароля для входу через електронну пошту;
4. Війти – кнопка для підтвердження реєстрації користувача у додатку;
5. Війти під іншим іменем – посилання для переходу на початкову сторінку авторизації.

Якщо система, після введених вірних даних, пропускає користувача та атоматично користувач поадає на форму перегляду проектів та завдань проектів.

Якщо ж користувач вперше заходить у систему – то його перекидає на сторінку створення нового проекту.

Дану форму наведено на рисунку 3.3.4

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		37

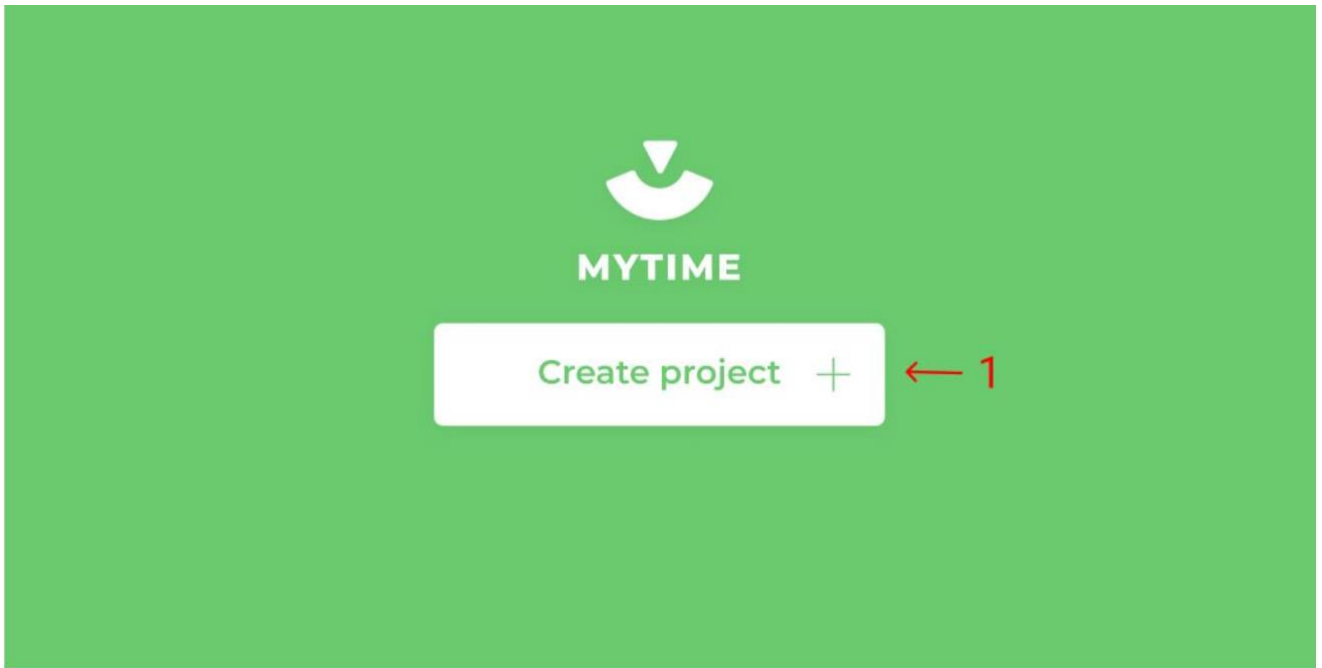


Рисунок 3.3.4 – Веб-форма створення та додавання нового проекту

На цій формі міститься тільки один елемент, при натисненні на який, здійснюється пересилання на форму для введення нових даних для нового завдання в системі (Рисунок 3.3.5)

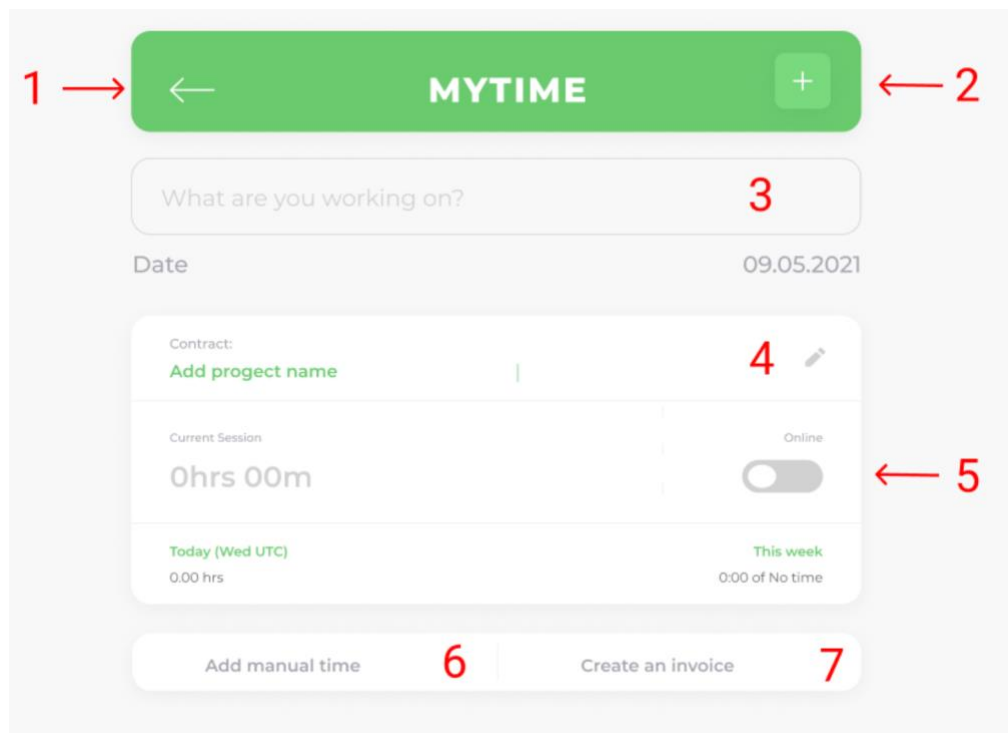


Рисунок 3.3.5 – Веб-форма застосунку для створення нового завдання

					КвРІПЗ.210194.01.20 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Після введення всіх необхідних даних відбувається перехід на форму, на якій відображаються дані про проект та роботу над ним з термінами і проведеним часом над ним співробітником (Рисунок 3.3.6).

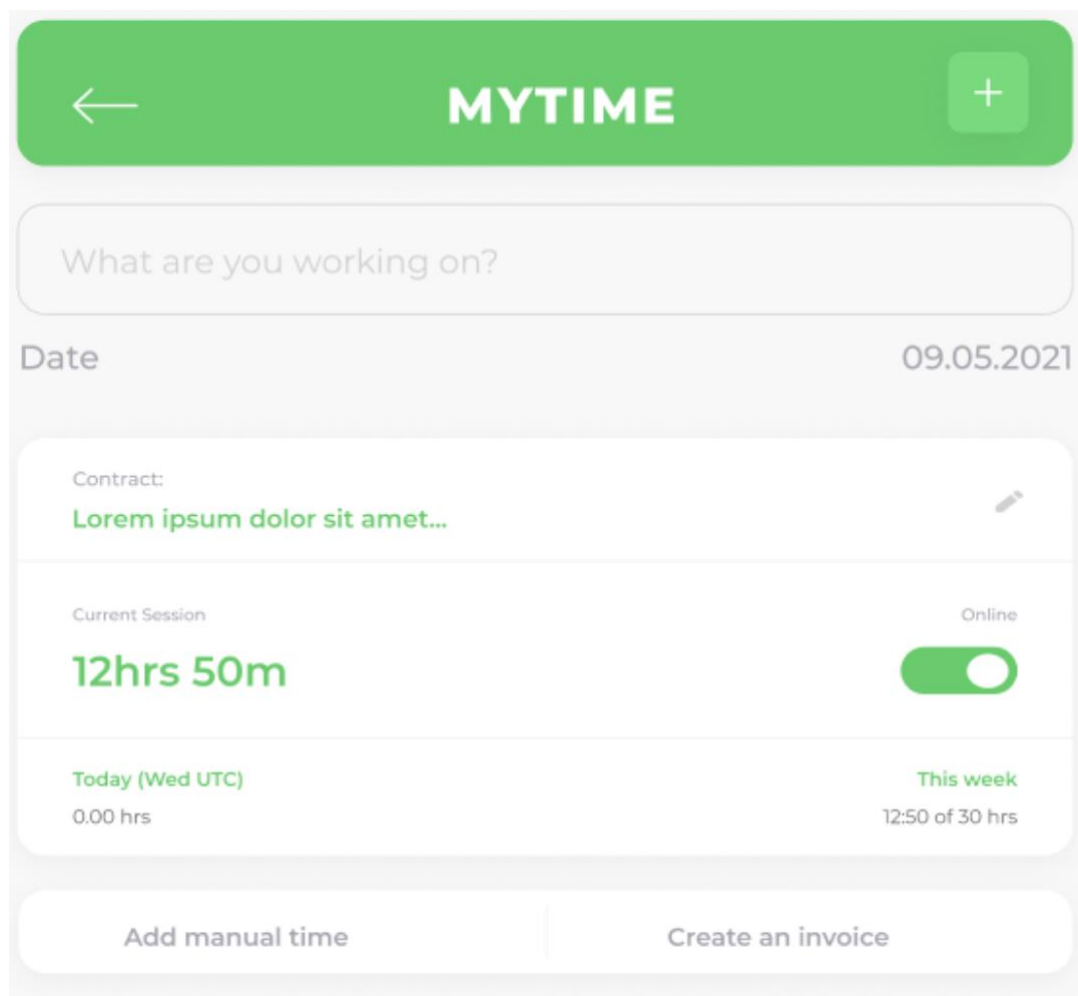


Рисунок 3.3.6 – Веб-форма створеного завдання з активним лічильником

На даній формі можна повертатись на попередню сторінку всіх проектів – за це відповідає кнопка з оберненою стрілкою.

Натиснувши на значок плюса, ми зможемо додати новий проект.

Поле введення над дати призначено для введення резюме проекту при вмиканні повзунка включення відліку часу.

Назва проекту вноситься після дати занесення.

Повзунком нижче ми зможемо включити відлік часу для проекту. Крім того нижче передбачено поле для введення та коригування часу в ручному режимі, якщо

					КвРІПЗ.210194.01.20 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

виникає така необхідність для проекту в цілому, чи для окремого завдання в силу різних визначених причин. Також на формі передбачено кнопка для відправки повідомлення замовнику.

При переході на форму, де відображаються активні проекти ми побачимо форму зображену на рисунку 3.3.7

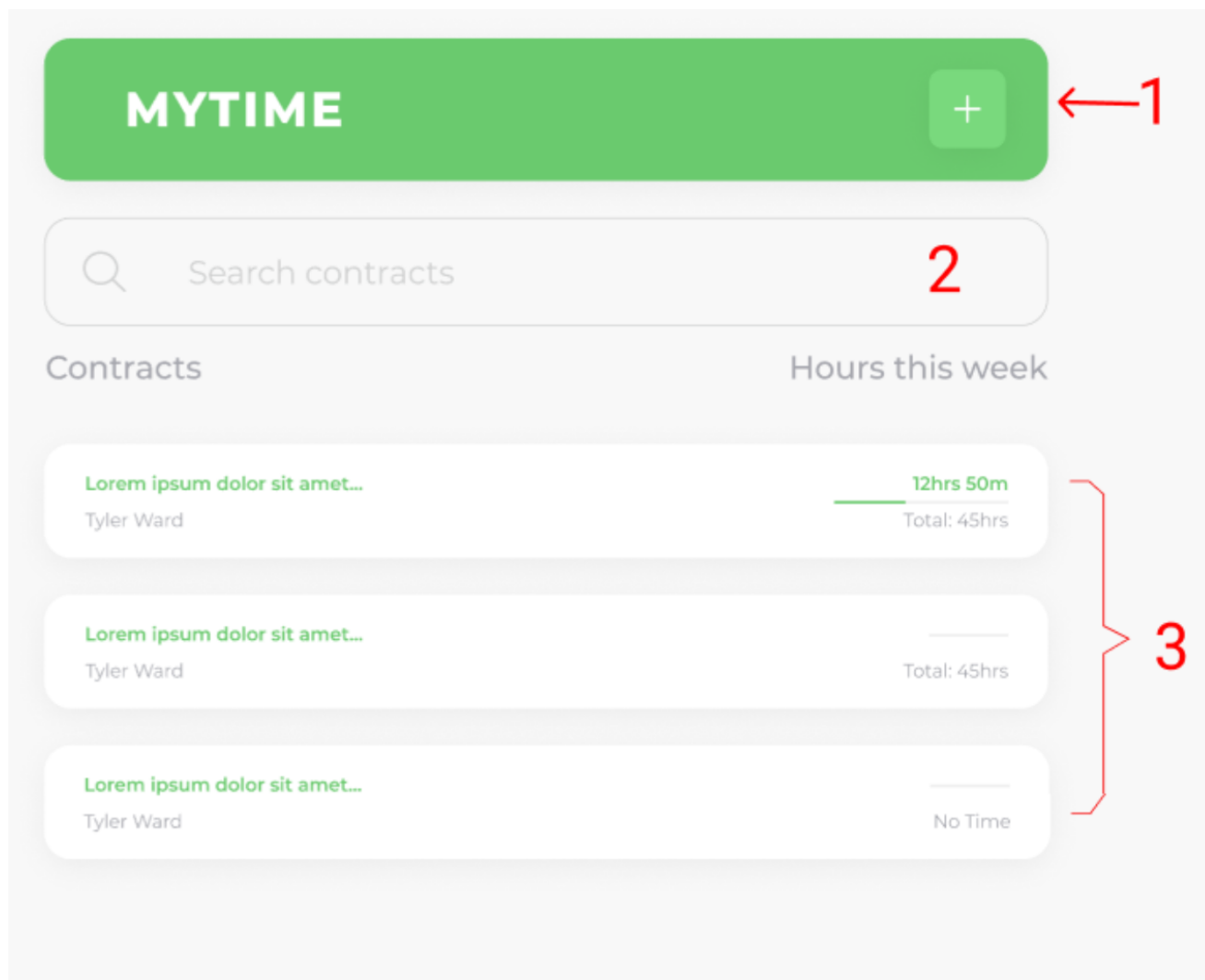


Рисунок 3.3.7 – Веб-форма активних завдань та проектів системи

Зокрема на цій формі є елемент, щоб добавляти нові завдання та проекти кнопка для пошуку проектів клієнтами та повний список усіх активних проектів, що розробляється певними співробітниками.

На рисунку 3.3.8 можемо переглядати не тільки про проекти, а також коментарі залишені під ним іншими користувача, що також дасть можливість

					КвРІПЗ.210194.01.20 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

вносити корективи в розрахунок затраченого часу на проект кожним окремим співробітником.

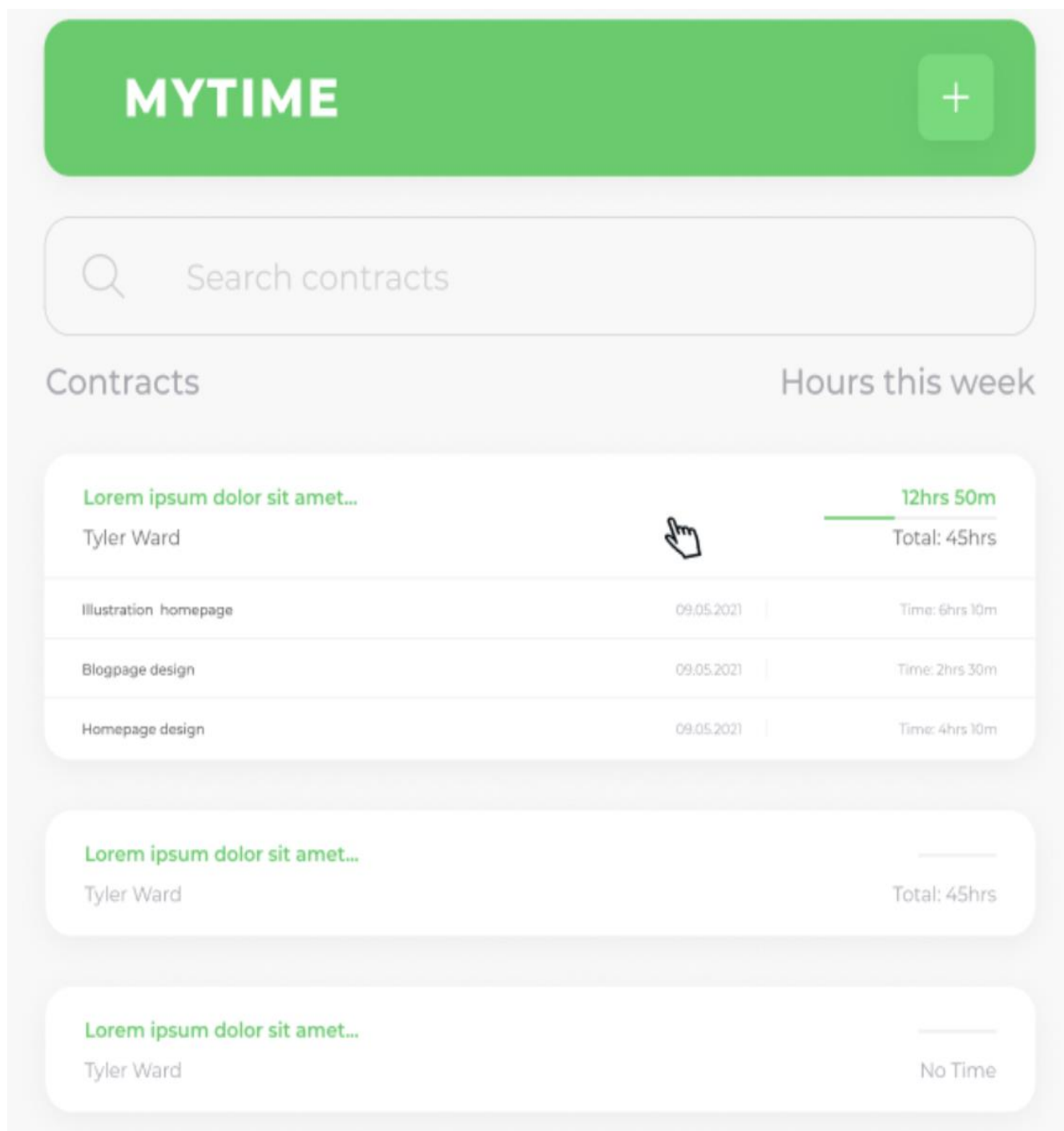


Рисунок 3.3.8 – Форма для перегляду коментарів залишених у системі

На цій формі, при виборі конкретного проекту буде відображатись відкривається список коментарів залишених різними типами користувачів, відповідно з часом та датою додавання коментаря.

Таким чином головна увага приділяється тому, що кожен раз при включенні лічильника , необхідно залишати повідомлення про причини його включення та вимкнення, а також причини, що до цього призвели.

Таким чином стане зрозумілим над яким проектом працював співробітник, що він з ним робив, і з яких причин вмикався/вимикався лічильник при роботі над завданням, чи певним проектом.

3.4 Технічні характеристики програмного застосунку

Програмний застосунок розрахунку робочого часу співробітників спроектований та розроблений за допомогою сучасних інформаційних технологій, які забезпечили чітку та стабільну його роботу. Продемонструвало гнучкість його роботи, простоту підтримки серверної і клієнтської частин. Розроблена архітектура застосунку досить ефективно розподіляє роботу функціоналу між його компонентами, забезпечуючи взаємодію їх між собою та безперебійну роботу загалом.

Серверна частина нашого застосунку розроблена на базі:

- мов програмування Python та JavaScript, які обробляють вхідні запити від клієнтів;
- візуальне середовище програмування та розробки VS Code;
- середовище виконання: Node.js;
- модуль mysql-python для роботи з базою даних;
- серверна СКБД MySQL.

Для роботи серверної частини необхідно мінімальне технічне обладнання:

- процесор з двома ядрами із частотою не менше 1,5 GHz;
- оперативної пам'яті від 4 GB та більше;

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

– постійної пам'яті не менше 5 GB для самого додатку, СКБД, даних та лог-файлів;

– інтернет-зв'язок через будь-якого провайдера, що не блокує порти для telegram каналів.

Клієнтська частина розроблена на Python та JavaScript, які обробляють вхідні запити від клієнтів, та відповідає за створення інтерфейсу, інтегрує різні сучасні інструменти розробки та проектування, та забезпечує роботу з API месенджера Telegram.

Основні характеристики:

– мов програмування Python та JavaScript, які обробляють вхідні запити від сервера та клієнтів;

– візуальне середовище програмування та розробки VS Code;

– бібліотеку Java React;

– маршрутизацію;

– HTTP-клієнт для з'єднання клієнта з сервером і навпаки;

– стилі застосунку за допомогою мов програмування CSS та HTML;

– спеціалізовану бібліотеку uuidt.

Для коректної роботи клієнтської частини потрібні:

– браузер Chrome, Safari, Opera, Firefox або MS Edge з сучасними оновленнями;

– мінімальна роздільна здатність для екрану клієнта: 1080x640Мрх;

– оперативна пам'ять від 8 GB та більше;

– стабільне інтернет-з'єднання через провайдера для взаємодії з серверною частиною.

Для повноцінного функціонування програмної застосунку необхідно також відсутність блокування портів, як локальним брандмауером так і на стороні провайдера інтернет-послуг.

Даного технічного забезпечення цілком достатньо для його повноцінної комфортної роботи.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						43
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

3.5 Тестування програмного застосунку

3.5.1 Мета, вибір та обґрунтування методів та засобів тестування програмного застосунку

Мета тестування нашого програмного застосунку розрахунку робочого часу співробітників, це перевірка коректності його роботи системи, також чи відповідають його функціональні можливості, які використовуються при розрахунку робочого часу співробітників підприємства, попередньо заявленим його функціональним та технічним вимогам.

Тестування будемо на основі стандартів проводитися на основі наступних вимог стандартів до програмного забезпечення:

- інформаційна технологія: види випробувань програмних систем;
- автоматизовані та інформаційні системні вимоги до змісту їх документів.

3.5.2 Верифікація та валідація вебсистеми

Для оцінки функціональних та технічних можливостей програмного застосунку розрахунку робочого часу співробітників було проведено, відповідно, верифікацію та валідацію, які підтверджують коректність реалізації усіх її компонентів та відповідність системним технічним вимогам. Процес перевірки та тестування включав в себе тестування основних функцій програмного застосунку, а також перевірку обробки вхідних та вихідних даних, оцінку стабільності роботи застосунку, виявлення можливих дефектів. Особлива увага при оцінці роботи застосунку приділялася валідації різних форматів даних, тестуванню прав доступу різних типів користувачів, синхронізація роботи застосунку з роботою бази даних та коректності виконання різних дій користувачів.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						44
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Тестування діяльності роботи програмного застосунку проводилося з використанням автоматизованих тестів з допомогою плагінів у середовищі VStudio Code. Це забезпечило точність, коректність та повторюваність усіх отриманих результатів.

3.5.3 Результати тестування

В ході нашого тестування перевірялося на відповідність програмного застосунку розрахунку робочого часу співробітників підприємства вимогам поставленого технічного завдання. У таблицях (3.1 – 3.13) саме наведено тестовий список основних його функціональних можливостей, які реалізуються з допомогою розробленого нами програмного застосунку.

Таблиця 3.1– Тестування головної сторінки застосунку

Назва тестування	Відображення головної сторінки програмного забезпечення
Попередні вимоги для тестування	Робота браузера
Деталі тестування	1. В адресний рядку браузера введено вірну url-адресу 2. Натиснення кнопки на сторінці для переходу за адресою посилання
Результат тестування	В браузері відображено вірно головну сторінку
Відповідність результату	Відповідає

Таблиця 3.2– Тестування сторінки реєстрації

Назва тестування	Відображення вікна реєстрації
Попередні вимоги для тестування	Завантажена головна веб-сторінка
Деталі тестування	<ol style="list-style-type: none"> 1. Заповнення поля логіну на формі 2. Заповнення поля паролю на формі 3. Підтвердження через поле електронної пошти 4. Натиснення кнопки входу в програмну систему
Результат тестування	Вдалих вхід в систему при вірному введені логіна, e-mail та паролю
Відповідність результату	Відповідає

Таблиця 3.3– Тестування сторінки авторизації

Назва тестування	Відображення вікна авторизації
Попередні вимоги для тестування	Завантажена головна веб-сторінка
Деталі тестування	<ol style="list-style-type: none"> 1. Заповнення поля логіну на формі 2. Заповнення поля паролю на формі 3. Натиснення кнопки входу в програмну систему
Результат тестування	Вдалих вхід в систему при вірному введені логіна та паролю
Відповідність результату	Відповідає

Таблиця 3.4– Тестування сторінки входу

Назва тестування	Відображення модального вікна вдалого входу
Попередні вимоги для тестування	Завантажена головна веб-сторінка програмної системи
Деталі тестування	1. Натиснення кнопки входу в програмну систему
Результат тестування	Вдалих вхід в систему при вірному при натисненні на кнопку
Відповідність результату	Відповідає

Таблиця 3.5– Тестування сторінки введення даних співробітника

Назва тестування	Відображення вікна для введення даних про користувача
Попередні вимоги для тестування	Завантажена веб-сторінка з редагування даних співробітника програмної системи
Деталі тестування	1. Натиснення кнопки редагування співробітника 2. Перевірка вірності введення даних полях форми
Результат тестування	Вдалих внесення даних в систему при при натисненні на кнопку додавання
Відповідність результату	Відповідає

Таблиця 3.6– Тестування сторінки редагування даних співробітника

Назва тестування	Відображення вікна для редагування даних про співробітника
Попередні вимоги для тестування	Завантажена веб-сторінка з редагування даних співробітника програмної системи
Деталі тестування	1. Натиснення кнопки редагування даних співробітника 2. Перевірка вірності введення даних полях форми
Результат тестування	Вдалий редагування в базі даних натисненні на кнопку редагування
Відповідність результату	Відповідає

Таблиця 3.7 – Тестування сторінки редагування проектів та робіт співробітника

Назва тестування	Відображення вікна для редагування проектів та робіт співробітника
Попередні вимоги для тестування	Завантажена веб-сторінка з редагування даних про проекти та роботи співробітника
Деталі тестування	1. Натиснення кнопки редагування даних про проекти та роботи співробітника 2. Перевірка вірності введення даних полях форми
Результат тестування	Вдалий редагування в базі даних про проекти та роботи при натисненні на кнопку редагування
Відповідність результату	Відповідає

Таблиця 3.8 – Тестування сторінки запуску системи розрахунку робочого часу співробітника

Назва тестування	Запуск системи розрахунку робочого часу співробітника
Попередні вимоги для тестування	Завантажена веб-сторінка системи розрахунку робочого часу співробітника
Деталі тестування	<ol style="list-style-type: none"> 1. Натиснення кнопки запуску розрахунку часу роботи співробітника 2. Перевірка вірності запису даних у базу даних про час роботи співробітника
Результат тестування	Вдалий запис у базі даних робочий час співробітника при натисненні на кнопку відліку часу
Відповідність результату	Відповідає

Таблиця 3.9 – Тестування сторінки роботи системи розрахунку робочого часу співробітника

Назва тестування	Запуск системи розрахунку робочого часу співробітника
Попередні вимоги для тестування	Відображається сторінка з проектом співробітника
Деталі тестування	<ol style="list-style-type: none"> 1. Натиснення співробітником кнопки запуску розрахунку часу роботи 2. Перевірка вірності запису даних у базу даних про час роботи співробітника
Результат тестування	Вдала робота системи і вдалий запис у базу даних про робочий

	час співробітника при натисненні на кнопку відліку часу роботи
Відповідність результату	Відповідає

Таблиця 3.10 – Тестування сторінки запуску системи розрахунку робочого часу співробітника

Назва тестування	Запуск зупинки системи розрахунку робочого часу співробітника
Попередні вимоги для тестування	Завантажена веб-сторінка системи розрахунку робочого часу співробітника
Деталі тестування	1. Натиснення кнопки зупинки розрахунку часу роботи співробітника 2. Перевірка вірності запису даних у базу даних про час роботи співробітника
Результат тестування	Вдалих запис у базі даних робочий час співробітника при натисненні на кнопку відліку часу
Відповідність результату	Відповідає

Таблиця 3.11 – Тестування сторінки зупинки системи розрахунку робочого часу співробітника

Назва тестування	Запуск зупинки системи розрахунку робочого часу співробітника
Попередні вимоги для тестування	Завантажена веб-сторінка системи розрахунку робочого часу співробітника

Деталі тестування	<p>1. Натиснення кнопки зупинки розрахунку часу роботи співробітника</p> <p>2. Перевірка вірності запису даних у базу даних про час роботи співробітника під час зупинки роботи системи</p>
Результат тестування	Вдалий запис у базі даних робочий час співробітника при натисненні на кнопку зупинки відліку часу роботи співробітника
Відповідність результату	Відповідає

Таблиця 3.12 – Тестування сторінки звітності роботи системи часу роботи співробітника

Назва тестування	Перегляд звітності відпрацьованих годин співробітника
Попередні вимоги для тестування	Відображення сторінки робіт проекту
Деталі тестування	<p>1. Натиснення кнопки звітності про час роботи співробітника</p> <p>2. Перевірка вірності запису даних у базу даних про час роботи співробітника під час перевірки системи звітності програмного застосунку</p>
Результат тестування	Вдале відображення з бази даних про робочий час співробітника при натисненні на кнопку звітності відліку часу роботи співробітника
Відповідність результату	Відповідає

Таблиця 3.12 – Тестування сторінки оплати праці часу роботи співробітника

Назва тестування	Перегляд сторінки оплати праці часу роботи співробітника
Попередні вимоги для тестування	Відображення сторінки оплати часу роботи співробітника
Деталі тестування	<ol style="list-style-type: none"> 1. Натиснення кнопки формування оплати за часи роботи співробітника 2. Перевірка вірності запису даних у базу даних про оплату роботи співробітника під час розрахунку оплати праці у програмного застосунку
Результат тестування	Вдале відображення з бази даних про робочий час співробітника при натисненні на кнопку звітності відліку часу роботи співробітника
Відповідність результату	Відповідає

3.6 Висновки розділу програмної реалізації та тестування програмного застосунку

В цьому розділі було продемонстровано розроблено базу даних, здійснено програмну розробку застосунку, здійснено тестування, послідовність виконання тестування, для перевірки відповідності програмного застосунку зазначеним функціональним вимогам, які наведено в технічному завданні до розробки програмного застосунку з розрахунку робочого часу співробітників підприємства. Наведено детально інструкцію користувача

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		52

ВИСНОВКИ

Для вирішення головної проблеми передбачення різних ситуацій, коли вартість самого виробничого процесу є високою та трудомісткою, і не може бути покрита доходом від продажу продукції компанії, витрати можна для цього визначити заздалегідь, перевіривши для цього в базі даних, перед прийняттям певного системного замовлення, перевірити чи не були вже виконані подібні роботи. Додаткові витрати на замовлення і матеріали, та розробку, які ще не були внесені в базу даних, обробляються програмним застосунком, а додаткові витрати таким чином вираховуються і вносяться в базу даних програмного застосунку.

В кваліфікаційній роботі було продемонстровано розроблено базу даних, здійснено програмну розробку застосунку, здійснено тестування, послідовність виконання тестування, для перевірки відповідності програмного застосунку зазначеним функціональним вимогам, які наведено в технічному завданні до розробки програмного застосунку з розрахунку робочого часу співробітників підприємства. Наведено детально інструкцію користувача

Також вирішується тут проблема розрахунку заробітної плати працівників на погодинній основі, оскільки це було б дуже складно робити вручну без автоматизації, коли є досить велика кількість працівників і частота їх появи змінюється. Програма автоматизує цей процес, розраховуючи години і заробіток до виплати для кожного співробітника, який працює з проектами.

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
						53
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. BOSSHR. Режим доступу: <https://www.bosshrm.com/>. Дата звернення: 11.03.2025
2. Бедратюк Л. П., Радельчук Г. І. Методичні настанови для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення»: кваліфікаційна робота / Л. П. Бедратюк, Г. І. Радельчук. – Хмельницький: ХНУ, 2023. – 60 с.
3. Форкун Ю., Мартинюк В., Яшина О. Метод розробки та проектування архітектурної складової програмного застосунку // Measuring and Computing Devices in Technological Processes. – 2023. – № 1. – С. 87–93. – [Електронний ресурс]. – Режим доступу: <https://evntpu.kpi.ua/article/view/281299> (дата звернення: 03.06.2025).
4. Протоковський А., Форкун Ю. Аналіз стратегій рекомендаційних систем знайомств людей та груп за інтересами у віртуальних спільнотах // Інформація, комунікація, суспільство – 2020: матеріали 9-ї Міжнародної науково-практичної конференції. – Хмельницький: ХНУ, 2020. – С. 119–122. – [Електронний ресурс]. – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2020/05/119.pdf> (дата звернення: 03.06.2025).
5. How HR Automation Can Optimize the Candidate Experience. URL: <https://www.hrtechnologist.com/articles/recruitment-onboarding/how-hr-automation-can-optimize-the-candidate-experience/> (дата звернення: 03.01.2025). Natural Language Processing (NLP) in Human Resources – Azure OpenAI Documentation. URL: <https://azure.microsoft.com/en-us/products/ai-services/openai-service/> (дата звернення: 17.02.2025).
6. Gartner Says 75% of Organizations Will Have Adopted AI for HR by 2025. URL: <https://www.gartner.com/en/newsroom/press-releases/2023-09-12-gartner-says-75-percent-of-organizations-will-have-adopted-ai-for-hr-by-2025> (дата звернення: 09.01.2025) LinkedIn. URL: <https://www.linkedin.com> (дата звернення: 12.02.2025).

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

7. Resume Worded – AI-Powered Resume Review. URL: <https://resumeworded.com> (дата звернення: 05.02.2025).
8. Speech and Language Processing (3rd ed. draft) by Daniel Jurafsky and James H. Martin. URL: <https://web.stanford.edu/~jurafsky/slp3/> (дата звернення: 14.02.2025).
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL: <https://arxiv.org/abs/1810.04805> (дата звернення: 06.01.2025).
9. GPT-4 Technical Report. URL: <https://openai.com/research/gpt-4-technical-report> (дата звернення: 20.02.2025).
10. The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. URL: <https://web.stanford.edu/~hastie/ElemStatLearn/> (дата звернення: 02.01.2025). КвРІПЗ.2101092.01.18.ПЗ Арк. 81 Змін. Арк. № докум. Підпис. Дата
11. Evaluating Machine Learning Models – Classification Metrics. URL: <https://developers.google.com/machine-learning/crash-course/classification/metrics> (дата звернення: 15.02.2025).
12. Microservices – Martin Fowler. URL: <https://martinfowler.com/articles/microservices.html> (дата звернення: 23.02.2025).
13. How to Beat the Applicant Tracking System (ATS) in 2024 – The Muse. URL: <https://www.themuse.com/advice/how-to-beat-the-applicant-tracking-system-ats-resume> (дата звернення: 28.02.2025).
13. How to design chatbots – The Ultimate Guide | UXDesign.cc. URL: <https://uxdesign.cc/how-to-design-chatbots-the-ultimate-guide-9ddc7b28669e> (дата звернення: 27.02.2025).
14. PyMuPDF Documentation. URL: <https://pymupdf.readthedocs.io/en/latest/> (дата звернення: 03.03.2025).
15. Hugging Face – Transformers Documentation. URL: <https://huggingface.co/docs/transformers/> (дата звернення: 05.03.2025).
16. Azure OpenAI Service – Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/ai-services/openai/> (дата звернення: 09.03.2025).

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		55

16. How to Beat the Applicant Tracking System (ATS) in 2024 – The Muse. URL: <https://www.themuse.com/advice/how-to-beat-the-applicant-tracking-system-ats-resume> (дата звернення: 11.03.2025).
17. Python Documentation – json Library. URL: <https://docs.python.org/3/library/json.html> (дата звернення: 17.03.2025).
18. Docker Documentation – Get Started. URL: <https://docs.docker.com/get-started/overview/> (дата звернення: 19.03.2025).
19. Web Scraping with Python – Ryan Mitchell. O’Reilly Media. (дата звернення: 22.03.2025).
20. Azure OpenAI Service – Models Overview. URL: <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models> (дата звернення: 25.03.2025).
21. Aiogram FSM – aiogram documentation. URL: <https://docs.aiogram.dev/en/latest/dispatcher/fsm.html> (дата звернення: 29.02.2025).
22. How to Format a Resume – The Balance Careers. URL: <https://www.thebalancecareers.com/how-to-format-your-resume-2062332> (дата звернення: 01.03.2025).
23. Software Testing Basics: A Guide to Types, Principles, and Methods. (дата звернення: 05.03.2025).
24. What is Load Testing? – k6 Blog. URL: <https://k6.io/blog/what-is-load-testing/> (дата звернення: 09.03.2025).
25. Effective Bug Management and Error Analysis in Software Testing. (дата звернення: 15.03.2025).
26. Types of Resumes – Indeed Career Guide. URL: <https://www.indeed.com/career-advice/resumes-cover-letters/types-of-resumes> (дата звернення: 28.03.2025).
27. Debugging Techniques in Python – Real Python. URL: <https://realpython.com/python-debugging-pdb/> (дата звернення: 23.05.2025).
28. Load Testing vs. Stress Testing – Apache JMeter Blog. URL: https://jmeter.apache.org/usermanual/glossary.html#load_test (дата звернення: 12.05.2025).

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		56

29. The Do's and Don'ts of Resume Writing – Harvard Business Review. URL: <https://hbr.org/2020/01/the-dos-and-donts-of-resume-writing> (дата звернення: 30.04.2025)
30. Best Web Development Stacks to Use in 2025. URL: <https://www.nobledesktop.com/classes-near-me/blog/best-web-development-stacks> (дата звернення 06.03.25)
31. What is MERN Stack? A Complete Guide [2025] - GUVI Blogs. URL: <https://www.guvi.in/blog/guide-for-mern-stack/> (дата звернення 06.03.25)
32. MERN stack tutorial: The Complete Guide with Examples. URL: <https://deadsimplechat.com/blog/mern-stack-the-complete-guide/> (дата звернення 07.03.25)
33. How to Add Drag and Drop in React with React Beautiful DnD. URL: <https://www.freecodecamp.org/news/how-to-add-drag-and-drop-in-react-with-react-beautiful-dnd/> (дата звернення 08.03.25)
34. Advantages and Benefits of React JS: The Essential Tool for Modern Developers. URL: <https://maybe.works/blogs/react-js-advantages> (дата звернення 12.03.25)
35. What reasons to use React for web development?. URL: <https://www.netguru.com/blog/why-use-react> (дата звернення 13.03.25)
36. JWT vs. OAuth: Comparing Authentication Protocols for Web Apps. URL: <https://frontegg.com/blog/oauth-vs-jwt> (дата звернення 13.03.25)
37. Security Best Practices for MERN Stack Applications. URL: <https://auth0.com/blog/security-best-practices-for-mern-stack-applications/> (дата звернення 14.03.25)
38. Building Accessible UI Components in React with ARIA. URL: <https://dev.to/webdevlapani/building-accessible-react-components-with-react-aria-5518> (дата звернення 18.03.25)
39. Jest testing: A complete tutorial. URL: <https://www.testim.io/blog/jest-testing-a-helpful-introductory-tutorial/> (дата звернення 29.03.25)

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		57

40. 40. Visual Testing with Jest. URL: <https://www.geeksforgeeks.org/testing-with-jest/> (дата звернення 03.04.25)

					<i>КвРІПЗ.210194.01.20 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		58

ДОДАТКИ

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Ця робота виконується в рамках дипломного проекту з розробки вебсистеми, призначеної для організації та контролю виконання проектних завдань. Програмний застосунок, який розробляється в межах даної кваліфікаційної роботи, має на своїй меті не лише автоматизувати процес обліку та розрахунку робочого часу та оплати праці співробітникам, але й створити передумови для підвищення якості роботи співробітників.

Мета нашої кваліфікаційної роботи полягає в проектуванні та розробці програмного застосунку для розрахунку робочого часу співробітників підприємства, яка має мати зрозумілий і зручний інтерфейс, для різних груп користувачів: управнців, менеджерів та працівників. Умовне позначення системи: «Програмний застосунок для розрахунку робочого часу співробітників підприємства»

1 Підстава для розробки

Розробка здійснюється на основі документа «Завдання на дипломний проект», затвердженого завідувачем кафедри інформаційних технологій.

Назва розробки: Програмний застосунок для розрахунку робочого часу співробітників підприємства

2 Призначення розробки

Вебсистема для координації та моніторингу виконання проектних завдань призначена для забезпечення зручного та ефективного інструменту, який дозволяє командам планувати, контролювати та оптимізувати робочі процеси в реальному часі. Система спрямована на підтримку різних груп

користувачів — менеджерів, виконавців і адміністраторів — у створенні структурованих робочих просторів, де можна визначати завдання, відстежувати їхній прогрес і координувати діяльність незалежно від географічного розташування.

Основний функціонал системи включає:

- створення, редагування та видалення проєктних завдань із зазначенням пріоритетів, дедлайнів і виконавців;
- організацію завдань у візуальні дошки з можливістю перетягування між статусами;
- сповіщення змін і створення завдань, нагадування про дедлайн;
- безпечне управління доступом із розподілом ролей і прав для різних користувачів.

Вебсистема розробляється для компаній і команд, які прагнуть підвищити ефективність управління проєктами, автоматизувати рутинні операції та забезпечити прозорість робочих процесів. Вона сумісна з усіма сучасними веббраузерами та адаптована для використання на різних пристроях із доступом до Інтернету, включаючи комп'ютери, планшети та смартфони.

3 Вимоги до програмного забезпечення

3.1 Функціональні вимоги

Функціональні вимоги визначають основні можливості вебсистеми для організації та контролю виконання проєктних завдань, забезпечуючи зручність і ефективність для всіх груп користувачів — менеджерів, виконавців і адміністраторів.

Реєстрація користувача: користувач може створити обліковий запис, вказавши ім'я, електронну пошту та пароль. Пароль має бути не коротшим за 10 символів і містити великі та малі літери, цифри й спеціальні символи. Після реєстрації на email надсилається лист із посиланням для підтвердження облікового запису.

Автентифікація користувача: система забезпечує вхід за допомогою електронної пошти та пароля. У разі введення некоректних даних відображається повідомлення про помилку, а після 5 невдалих спроб вхід блокується на 10 хвилин.

Створення проєктного завдання: користувач може додати нове завдання, вказавши назву, опис, дедлайн, пріоритет (низький, середній, високий), виконавця та прикріпити файли (наприклад, зображення чи документи). Дані зберігаються в базі даних із автоматичним оновленням статусу.

Редагування та видалення завдань: користувач має можливість змінювати будь-які параметри завдання (назву, опис, дедлайн, пріоритет, виконавця) або видаляти завдання. Зміни відображаються в реальному часі для всіх учасників проєкту.

Візуалізація завдань: завдання відображаються у форматі дошки Kanban із колонками, що представляють статуси (наприклад, «Заплановано», «У роботі», «Завершено»). Користувач може переміщувати завдання між колонками методом drag-and-drop.

Фільтрація та пошук завдань: система дозволяє шукати завдання за ключовими словами (назва, опис, виконавець) і фільтрувати їх за статусом, пріоритетом або дедлайном із можливістю сортування за датою створення чи дедлайном.

Сповіщення про події: користувачі отримують сповіщення про наближення дедлайнів, призначення на завдання чи зміну статусу через інтерфейс системи або на електронну пошту.

3.2 Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики вебсистеми для організації та контролю виконання проєктних завдань, які впливають на її продуктивність, безпеку, зручність і надійність.

Швидкодія: час завантаження основної сторінки не має перевищувати 2 секунди за стандартного інтернет-з'єднання, а обробка запитів (наприклад, створення завдання) — не більше 1,5 секунди для забезпечення плавної роботи.

Перехід між розділами: перемикання між різними сторінками чи модулями системи (наприклад, від дошки завдань до звітів) має відбуватися протягом 1 секунди для підтримки безперервного робочого процесу.

Безпека: система використовує протокол HTTPS для шифрування даних під час передачі між клієнтом і сервером. Паролі захищаються алгоритмом хешування (наприклад, bcrypt), а автентифікація реалізується через JSON Web Token (JWT) із захистом від підроблення токенів.

Інтерфейс і зручність: дизайн системи розроблено з урахуванням принципів UX/UI, із чіткою навігацією, легко читаємим шрифтом (наприклад, Roboto або Open Sans) і гармонійною кольоровою палітрою, що мінімізує втому очей і забезпечує комфортне використання.

Доступність: вебсистема працює цілодобово (24/7), із плановими перервами для технічних оновлень чи усунення збоїв, які не перевищують 1% загального часу роботи на місяць.

3.3 Умови експлуатації

Вебсистема може використовуватись у сучасних браузерах на різних пристроях, таких як комп'ютери, планшети чи смартфони.

3.4 Вимоги до технічного забезпечення та продуктивності

Вебсистема для організації та контролю виконання проєктних завдань розроблена для роботи через браузери на пристроях із доступом до Інтернету, не потребуючи локальної установки на мобільних пристроях чи комп'ютерах. Доступ до системи здійснюється виключно через вебінтерфейс, що забезпечує гнучкість використання на різних платформах.

Для стабільної та ефективної роботи вебсистеми рекомендуються такі технічні характеристики:

Операційні системи: сумісність із Windows, Linux, macOS, а також Android і iOS через підтримку сучасних веббраузерів (Chrome, Firefox, Safari, Edge).

Процесор: двоядерний процесор із частотою від 1,2 ГГц або вищою для оптимальної продуктивності.

Оперативна пам'ять: мінімально 2 ГБ, рекомендується 4 ГБ і більше для комфортної роботи з великою кількістю завдань.

Вільне місце: щонайменше 1 ГБ для кешування даних браузера та тимчасового зберігання файлів, завантажених у систему.

Інтернет-з'єднання: стабільна швидкість не нижче 10 Мбіт/с для швидкого завантаження сторінок і обробки запитів у реальному часі.

3.5 Вимоги до програмної сумісності та інтеграцій

Вебсистема для організації та контролю виконання проектних завдань буде розроблена з використанням сучасного технологічного стеку, що забезпечить її продуктивність, масштабованість і зручність у розробці. Для створення фронтенду буде використано бібліотеку React, яка дозволить реалізувати динамічний і адаптивний користувацький інтерфейс. Бекенд буде розроблено на основі Node.js із застосуванням фреймворку Express.js для створення ефективного RESTful API та обробки запитів. Для зберігання даних, таких як завдання, профілі користувачів і сповіщення, буде використано базу даних MongoDB, яка забезпечує гнучкість і масштабованість.

Для забезпечення додаткової функціональності та оптимізації роботи системи буде здійснено інтеграцію з такими технологіями та сервісами:

JSON Web Token (JWT): буде застосовано для реалізації безпечної автентифікації та авторизації користувачів, що гарантуватиме захист токенів і управління сесіями.

Цей технологічний стек і набір інтеграцій буде використано для створення надійної, безпечної та масштабованої вебсистеми, яка відповідатиме потребам користувачів у сфері управління проєктними завданнями.

3.6 Вимоги до інтерфейсу та дизайну

Вебсистема для організації та контролю виконання проєктних завдань матиме наступні вимоги до інтерфейсу та дизайну, які забезпечать зручність і привабливий вигляд для користувачів:

Кольорова гама: буде використано комбінацію сірих, блакитних і зелених відтінків для створення сучасного та спокійного візуального стилю, із акцентними кольорами (наприклад, зелений для підтверджень) для виділення ключових елементів.

Навігація: панель навігації буде розміщена у верхній частині екрана (хедер), де передбачатимуться кнопки для переходу до дошки завдань, аналітики, профілю користувача та пошуку завдань, із логічним розташуванням для швидкого доступу.

Адаптивність: інтерфейс буде розроблено з урахуванням респонсивного дизайну, що забезпечить коректне відображення на пристроях різної роздільної здатності, включаючи комп'ютери, планшети та смартфони.

Ці вимоги до дизайну сприятимуть створенню інтуїтивно зрозумілого та естетично привабливого інтерфейсу, що відповідатиме принципам UX/UI та потребам користувачів.

4 Вимоги до програмної документації

При передачі вебсистеми для організації та контролю виконання проєктних завдань замовнику буде надано наступний комплект документації та матеріалів:

- програмний код;

- опис функціональних можливостей;
- інструкція користування;
- технічне завдання.

5 Стадії та етапи розробки

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання	01.01.25 – 20.02.25	Формулювання вимог до системи та документація, призначення, опис цілей та завдань
Ескізний проєкт	21.02.25 – 20.03.25	Розробка дизайну інтерфейсу, обрання середовища програмування
Технічний проєкт	21.03.25 – 30.04.25	Опис технічних характеристик та архітектури
Робочий проєкт	01.05.25 – 25.05.25	Реалізація логіки та основних функціональних можливостей вебзастосунку. Інтеграція з базою даних та взаємодія із зовнішніми сервісами
Розробка програмної документації	Травень 2025	Оформлення документації, передбаченої технічним завданням
Тестування системи	26.05.25 – 30.05.25	Проведення різних тестів для перевірки правильності роботи функцій застосунку
Впровадження	3 01.06.2025	Підготовка до розгортання системи, налаштування домену та передача вебзастосунку на експлуатацію користувачам

6 Порядок перевірки та приймання

Процес перевірки та приймання вебзастосунку здійснюється відповідно до визначених раніше вимог та містить такі етапи:

- функціональне тестування: перевірка основного функціоналу розробленого вебзастосунку;
- тестування продуктивності: оцінка часу завантаження сторінок, відповіді від сервера тощо;
- дотримання норм безпеки: перевірка захисту персональних даних, шифрування паролів, захист від атак;
- кросбраузерне тестування: перевірка правильності роботи в різних браузерах та пристроях;
- тестування інтеграції: перевірка взаємодії програмного забезпечення із сторонніми сервісами та API;
- тестування інтерфейсу: оцінка того, наскільки зручний та інтуїтивний інтерфейс різних сторінок та елементів вебзастосунку;
- відповідність документації: перевірка на відповідність встановленим вимогам та дотримання всіх інструкцій і також вирішення різноманітних проблем.

ДОДАТОК Б
(обов'язковий)

КОД ПРОГРАМИ

```
% extends 'base.html'%}
{% block title %}
Страница авторизации
{% endblock %}
{% block body %}
<h1>Страница авторизации </h1>
<main class="container mt-5">
<form method="post">
<h1 class="h3 mb-3 fw-normal">Please sign in</h1>
<input type="email" name="email" class="form-control" id="email"
placeholder="name@example.com"><br>
<input type="password" name="password" class="form-control" id="password"
placeholder="Password"><br>
<button class="w-100 btn btn-lg btn-primary" type="submit"
formaction="/create">Sign
in</button>
</form>
</main>
{% endblock %}
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0,
```

```
minimum-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<link rel="stylesheet" href="{{ url_for('static', filename='css/main.css') }}">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css">
<title>{% block title %} {% endblock %}</title>
</head>
<body>
</body>
{% block body %} {% endblock %}
</html>
{% extends 'base.html'%}
{% block title %}
Создание проекта
{% endblock %}
{% block body %}
<div class="container mt-5">
<h1>Создание проекта </h1>
<header class="bd-header bg-dark py-3 d-flex align-items-stretch border-bottom
border-
blue">
<div class="container-fluid d-flex align-items-center">
<h1 class="h3 mb-3 fw-normal">My time</h1>
<button class="ms-auto link-light" type="submit" formaction="/">+</button>
</div>
</header>
<div class="container mt-5">
<input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search">
```

```
</div><br>
<div class="row gy-3">
  <div class="col-md-6">
    <label class="form-label">Name on card</label>
  </div>
  <div class="col-md-6">
    <label class="form-label">Credit card number</label>
  </div>
</div>
<div class="styles217493542__root Grid2890349840__columnRoot _1gSj_"
  style="grid-area:
  span 1 / span 8 / auto / auto;">
  <div class="_-8w2N _6fWpo">
    <div>
      <div data-hook="step" class="Box2645419772__root Box2645419772---
        direction-10-
        horizontal Box2645419772---justifyContent-6-middle" style="--
        Box2645419772-gap:0;">
        <div class="Box2645419772__root Box2645419772---direction-8-vertical
          _1QxH3
          _3Ekjt
          {% endblock %}
          {% extends 'base.html'%}
          {% block title %}
          Главная страница
          {% endblock %}
          {% block body %}
        <header class="container mt-5">
        <form>
        <a class="btn btn-sm btn-outline-secondary" href="/authorisation">Sign in</a>
```

```
<a class="btn btn-sm btn-outline-secondary" href="/registration">Sign up</a>
</form>
</header>
<h1>Главная страница </h1>
{% endblock %}
{% extends 'base.html'%}
{% block title %}
Страница регистрации
{% endblock %}
{% block body %}
<h1>Страница регистрации </h1>
<!--
<div class="container mt-5">
<form method="post">
<h1 class="h3 mb-3 fw-normal">Please sign in</h1>
<input type="email" name="email" class="form-control" id="floatin_gemail"
placeholder="name@example.com"><br>
<input type="password" name="password" class="form-control"
id="floatingPassword"
placeholder="Password"><br>
<button class="w-100 btn btn-lg btn-primary" type="submit"
formaction="/create">Sign
in</button>
</form>
</div>
-->
<main class="container mt-5">
<form method="post">
<h1 class="h3 mb-3 fw-normal">Please sign up</h1>
```

```
<input type="text" name="user" class="form-control" id="user"
placeholder="User"><br>
<input type="email" name="email" class="form-control" id="email"
placeholder="name@example.com"><br>
<input type="password" name="password" class="form-control" id="password"
placeholder="Password"><br>
<input type="password" name="confirmpassword" class="form-control"
id="confirmpassword" placeholder="Confirm Password"><br>
<button class="w-100 btn btn-lg btn-primary" type="submit"
formaction="/autorisation">Sign
up</button>
</form>
</main>
```

```
{% endblock %}
```

```
import telebot
```

```
from telebot import types
```

```
TOKEN = '1765799695:AAFyY7YtsANfI10yJD4fdFPKLKEliSOOVRY'
```

```
ДП 7122.00.000 ПЗ
```

```
Арк.
```

```
61
```

```
Змн. Арк. № докум. Підпис Дата
```

```
bot = telebot.TeleBot(TOKEN)
```

```
@bot.message_handler(commands=['start'])
```

```
def email(message):
```

```
bot.send_message(message.chat.id, " Введи, пожалуйста, email
разработчика")
```

```
@bot.message_handler(content_types=['text'])
```

```
def auto(message):
```

```
if message.text == "marisha73@gmail.com":
```

```

bot.send_message(message.chat.id, " Вы подключились, к проекту Lorem
ipsum dolor sit
amet")
bot.send_message(message.chat.id, " Каждые 10 минут вам будут приходить
скриншоты работы разработчика")
p1 = open("p1.png", 'rb')
p2 = open("p2.png", 'rb')
p3 = open("p3.png", 'rb')
p4 = open("p4.png", 'rb')
p5 = open("p5.png", 'rb')
bot.send_photo(message.chat.id, p1)
bot.send_photo(message.chat.id, p2)
bot.send_photo(message.chat.id, p3)
bot.send_photo(message.chat.id, p4)
bot.send_photo(message.chat.id, p5)
markup = types.InlineKeyboardMarkup()
item = types.InlineKeyboardButton("Оплатить проект", callback_data='hi')
markup.add(item)
bot.send_message(message.chat.id, " Разработчик закончил работу над
проектом",
reply_markup=markup)
bot = telebot.TeleBot(TOKEN)
@bot.message_handler(commands=['start'])
def email(message):
bot.send_message(message.chat.id, " Введи, пожалуйста, email
разработчика")
@bot.message_handler(content_types=['text'])
def auto(message):
if message.text == "marisha73@gmail.com":

```

```
bot.send_message(message.chat.id, " Вы подключились, к проекту Lorem  
ipsum dolor sit  
amet")  
bot.send_message(message.chat.id, " Каждые 10 минут вам будут приходить  
скриншоты работы разработчика")  
p1 = open("p1.png", 'rb')  
p2 = open("p2.png", 'rb')  
p3 = open("p3.png", 'rb')  
p4 = open("p4.png", 'rb')  
p5 = open("p5.png", 'rb')  
bot.send_photo(message.chat.id, p1)  
bot.send_photo(message.chat.id, p2)  
bot.send_photo(message.chat.id, p3)  
bot.send_photo(message.chat.id, p4)  
bot.send_photo(message.chat.id, p5)  
markup = types.InlineKeyboardMarkup()  
item = types.InlineKeyboardButton("Оплатить проект", callback_data='hi')  
markup.add(item)  
bot.send_message(message.chat.id, " Разработчик закончил работу над  
проектом",  
reply_markup=markup)  
else:  
bot.send_message(message.chat.id, " Вашего email нет в базе")  
bot.polling()  
if __name__ == "__main__":  
app.run(debug=True)# указывает на ошибки  
import random2  
from datetime import datetime  
import numpy as np
```

```

import pyautogui
import imutils
import cv2
def random():
minute_random = int(random2.uniform(1, 11))
print(minute_random)
return minute_random
def time_now(time_hour, time_minute):
if time_hour == 1 and time_minute == 0:
time_hour = datetime.now().hour
return time_hour
elif time_hour == 0 and time_minute == 1:
time_minute = datetime.now().minute
return time_minute
else:
print('veli ne pravelnie znachenia')
def only_minute(hour, minute):
only_minute = hour*60+minute
return only_minute
def randomnie_minute(minute_ten, ran):
ДП 7122.00.000 ПЗ
Арк.
63
Змн. Арк. № докум. Підпис Дата
next_minute = (minute_ten + 1) * 10 + ran
return next_minute
def future_time():
ran=random()
minute_ten = int(time_now(0,1)/10)

```

```

next_minute = randomnie_minute(minute_ten, ran)
if next_minute >= 60:
new_time_now_hour = time_now(1, 0) + 1
new_time_now_minute = next_minute - 60
alone_minute = only_minute(new_time_now_hour,new_time_now_minute)
print(alone_minute)
return alone_minute
else:
alone_minute = only_minute(time_now(1, 0), next_minute)
print(alone_minute)
return alone_minute
def main():
if only_minute(time_now(1, 0), time_now(0, 1)) == future_time():
while True:
#if only_minute(time_now(1,0),time_now(0, 1)) == future_time():
image = pyautogui.screenshot(region=(0, 0, 300, 400))
image = cv2.cvtColor(np.array(image), cv2.COLOR_RGB2BGR)
cv2.imshow('Screenshot', imutils.resize(image, width=600))
cv2.imwrite('pic.png', image)
break
main()
" style="--Box2645419772-gap:0;"><div><div class="Box2645419772__root
Box2645419772---direction-10-horizontal    Box2645419772---alignItems-13-
space-between
Box2645419772---justifyContent-6-middle"          style="--Box2645419772-
gap:0;"><div
class="Box2645419772__root          Box2645419772---direction-10-horizontal
Box2645419772---
justifyContent-6-middle"    style="min-height:    24px;    --Box2645419772-
gap:0;"><svg

```

```
viewBox="0 0 18 18" fill="currentColor" width="18" height="18"
class="_3TSQe_2YSSI"
data-hook="step-incomplete"><circle cx="9" cy="9"
r="2"></circle></svg><div
class="Box2645419772__root Box2645419772---direction-10-horizontal"
style="padding-right:
12px; padding-left: 8px; --Box2645419772-gap:0;"><span
class="Text4265422782__root
Text4265422782---size-4-tiny Text4265422782---skin-8-standard
Text4265422782---weight-4-
thin Text4265422782---list-style-9-checkmark" data-size="tiny" data-
secondary="false" data-
skin="standard" data-light="false" data-weight="thin" data-list-
style="checkmark"><span data-
hook="step-title" class="Text4265422782__root Text4265422782---size-4-tiny
Text4265422782---skin-8-standard Text4265422782---weight-6-normal
Text4265422782---list-
style-9-checkmark awxcV" data-size="tiny" data-secondary="false" data-
skin="standard" data-
light="false" data-weight="normal" data-list-style="checkmark"> </span><span
data-
hook="step-time-to-complete" class="Text4265422782__root Text4265422782-
--size-4-tiny
Text4265422782--secondary Text4265422782---skin-8-standard
Text4265422782--light
Text4265422782---weight-4-thin Text4265422782---list-style-9-checkmark
_1gbdK" data-
size="tiny" data-secondary="true" data-skin="standard" data-light="true" data-
weight="thin"
data-list-style="checkmark">(7&nbsp;мин.)</span></span></div></div><div
class="Box2645419772__root Box2645419772---direction-10-horizontal"
style="--
Box2645419772-gap:0;"><div class="Box2645419772__root Box2645419772--
-direction-10-
```

horizontal Box2645419772---justifyContent-6-middle _3m5zQ" style="margin-right: 12px; --

Box2645419772-gap:0;"><button data-size="tiny" data-skin="standard" data-underline="none"

data-weight="thin" class="buttonnext967279998__root Focusable1296630989__root

TextButton834966688__root TextButton834966688---skin-8-standard TextButton834966688---

underline-4-none TextButton834966688---weight-4-thin TextButton834966688--size-4-tiny"

data-hook="step-skip" type="button" tabindex="0" aria-disabled="false">Пропустить</button></div><div

class="Box2645419772__root Box2645419772---direction-10-horizontal" style="--

Box2645419772-gap:0;"><button data-made-for="true" data-size="small" data-priority="primary" data-skin="standard" class="buttonnext967279998__root

Focusable1296630989__root Button2487934312__root Button2487934312---skin-8-standard

Button2487934312---priority-7-primary Button2487934312---size-5-small Button2487934312--

ellipsis _3XmQO" data-hook="step-cta" type="button" tabindex="0" aria-disabled="false"><span class="buttonnext967279998__content Ellipsis3115213275__text

Ellipsis3115213275---ellipsisLines-10-singleLine">Добавить

товар</button></div></div></div></div></div></div></div></div></div><div class="_-

8w2N "><div><div data-hook="step" class="Box2645419772__root Box2645419772---

direction-10-horizontal Box2645419772---justifyContent-6-middle" style="-- Box2645419772-

gap:0;"><div class="Box2645419772__root Box2645419772---direction-8-vertical

_1QxH3

```
" style="--Box2645419772-gap:0;"><div><div class="Box2645419772__root
Box2645419772---direction-10-horizontal    Box2645419772---alignItems-13-
space-between
Box2645419772---justifyContent-6-middle"          style="--Box2645419772-
gap:0;"><div
class="Box2645419772__root          Box2645419772---direction-10-horizontal
Box2645419772---
justifyContent-6-middle"    style="min-height:    24px;    --Box2645419772-
gap:0;"><svg
viewBox="0 0 18 18"    fill="currentColor"    width="18"    height="18"
class="_3TSQe_2YSSI"
data-hook="step-
incomplete"><circle    cx="9"    cy="9"    r="2"></circle></svg><div
class="Box2645419772__root
Box2645419772---direction-10-horizontal"
style="padding-right: 12px; padding-left: 8px; --Box2645419772-gap:0;"><span
class="Text4265422782__root Text4265422782---size-4-tiny Text4265422782-
--skin-8-
standard Text4265422782---weight-4-thin Text4265422782---list-style-9-
checkmark" data-
size="tiny" data-secondary="false" data-skin="standard" data-light="false" data-
weight="thin"
data-list-style="checkmark"><span          data-hook="step-title"
class="Text4265422782__root
Text4265422782---size-4-tiny          Text4265422782---skin-8-standard
Text4265422782---weight-6-
normal Text4265422782---list-style-9-checkmark awxcV" data-size="tiny" data-
secondary="false"    data-skin="standard"    data-light="false"    data-
weight="normal" data-list-
style="checkmark">Настройте    способы    оплаты</span><span    data-
hook="step-time-to-
```

complete" class="Text4265422782__root Text4265422782---size-4-tiny
Text4265422782--

secondary Text4265422782---skin-8-standard Text4265422782--light
Text4265422782---

weight-4-thin Text4265422782---list-style-9-checkmark _1gbdK" data-
size="tiny" data-

secondary="true" data-skin="standard" data-light="true" data-weight="thin"
data-list-

style="checkmark">(4 мин.)</div></div><div

class="Box2645419772__root Box2645419772---direction-10-horizontal"
style="--

Box2645419772-gap:0;"><div class="Box2645419772__root Box2645419772--
-direction-10-

horizontal Box2645419772---justifyContent-6-middle _3m5zQ" style="margin-
right: 12px; --

Box2645419772-gap:0;"><button data-size="tiny" data-skin="standard" data-
underline="none"

data-weight="thin" class="buttonnext967279998__root
Focusable1296630989__root

TextButton834966688__root TextButton834966688---skin-8-standard
TextButton834966688---

underline-4-none TextButton834966688---weight-4-thin TextButton834966688-
--size-4-tiny"

data-hook="step-skip" type="button" tabindex="0" aria-disabled="false"><span
class="buttonnext967279998__content">Пропустить</button></div><<
div

class="Box2645419772__root Box2645419772---direction-10-horizontal"
style="--

Box2645419772-gap:0;"><button data-made-for="true" data-size="small" data-
priority="secondary" data-skin="standard" class="buttonnext967279998__root
Focusable1296630989__root Button2487934312__root Button2487934312---
skin-8-standard

Button2487934312---priority-9-secondary Button2487934312---size-5-small

thin Text4265422782---list-style-9-checkmark" data-size="tiny" data-secondary="false" data-skin="standard" data-light="false" data-weight="thin" data-list-style="checkmark">/span>(3 мин.)</div></div><div class="Box2645419772__root Box2645419772---direction-10-horizontal" style="--Box2645419772-gap:0;"><div class="Box2645419772__root Box2645419772--direction-10-horizontal Box2645419772---justifyContent-6-middle _3m5zQ" style="margin-right: 12px; --Box2645419772-gap:0;"><button data-size="tiny" data-skin="standard" data-underline="none" data-weight="thin" class="buttonnext967279998__root Focusable1296630989__root TextButton834966688__root TextButton834966688---skin-8-standard TextButton834966688---underline-4-none TextButton834966688---weight-4-thin TextButton834966688--size-4-tiny"

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ СЛАЙДИ

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

На тему:

Програмний застосунок для розрахунку робочого
часу співробітників підприємства

Підготував: Студент IV курсу, групи ІПЗ-2 І-І
Денис Черкасов

Керівник: к.т.н., доцент Форкун Юрій Вікторович

АКТУАЛЬНІСТЬ ТЕМИ

Перед усіма керівниками підприємств та організацій, пов'язаних з розробкою та виробництвом, часто постає питання, чи варто братися за те чи інше замовлення? Буде це вигідно організації і співробітникам? Без урахування ситуації, при якій продукт чи проект просто не може бути реалізований, витрати на його розробку будуть досить високими, і навіть прибуток від продажу їх не відшкодує збитки.

Актуальність теми кваліфікаційної роботи обумовлена досить високим зростанням попиту на такі рішення для розрахунку робочого часу співробітників підприємства, що особливо відчутно в умовах переходу організацій до дистанційної форми їх роботи.

МЕТА

Програмний застосунок, який розробляється в межах даної кваліфікаційної роботи, має на своїй меті не лише автоматизувати процес обліку та розрахунку робочого часу та оплати праці співробітникам, але й створити передумови для підвищення якості роботи співробітників.

Мета нашої кваліфікаційної роботи полягає в проектуванні та розробці програмного застосунку для розрахунку робочого часу співробітників підприємства, яка має мати зрозумілий і зручний інтерфейс для різних груп користувачів: управлінців, менеджерів та працівників.

Сучасні тенденції розвитку ІТ показують, що такі застосунки відіграють велику роль в бізнесі, створюючи інновації та роблять організації конкурентоспроможними на ринку

Дослідження предметної області та постановка задачі

метою даного проекту є створення програмного застосунку, яка буде виконувати наступні завдання:

-зберігати інформацію про співробітників, що працюють в компанії (номер, прізвище, посаду, контактні дані, проекти):

-зберігає інформацію про всі завершені проекти на підприємстві, тобто наступну інформацію про них:

- назва проекту;

-вартість самого проекту;

час, витрачений на цей проект та окремі задачі;

Дослідження предметної області та постановка задачі

- дозволяє співробітнику побачити всі можливі проекти і задачі та вибрати ті, які йому цікаві і потрібні;
- менеджер може переглядати всі доступні проекти та завдання та додавати нові за потреби чи вимогою;
- менеджер може перевірити всі коефіцієнти для зарплати для всіх співробітників і змінити відповідно їх у міру необхідності;
- розрахувати додану вартість проектів та продуктів, що відображається в табелях та відомостях про заробітну плату;
- відповідальний може розрахувати зарплату працівника, а також час, проведений на роботі, виходячи з його коефіцієнта ефективності.

АНАЛІЗ НАЯВНОГО ПРОГРАМНО-ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

аналіз наявного на ринку програмного забезпечення та оцінимо їх оцінку за такими критеріями оцінки по 10-бальній шкалі, яке застосовується до оцінки такого програмного забезпечення:

№	Назва	Функціональність	Інтерфейс	Мобільність	Інтеграції	Аналітика	Гнучкість	Ціна/можливості	Безпека
1	Toggl Track	8	9	10	9	8	7	8	9
2	Clockify	9	8	9	8	9	8	10	8
3	Hubstaff	10	7	8	9	9	9	7	9
4	RescueTime	7	8	8	7	10	6	7	8
5	Time Doctor	9	7	9	8	8	8	7	9

Жодна з аналізованих платформ не поєднує відкритість API, має обмежену підтримку повноцінну локалізацію під українськомовного користувача.

ВИБІР ТИПУ АРХІТЕКТУРИ ТА ШАБЛОНІВ ПРОЄКТУВАННЯ

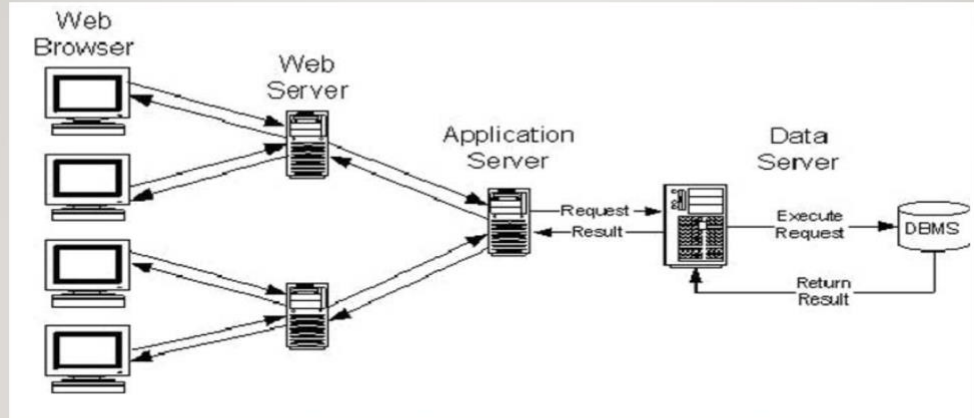
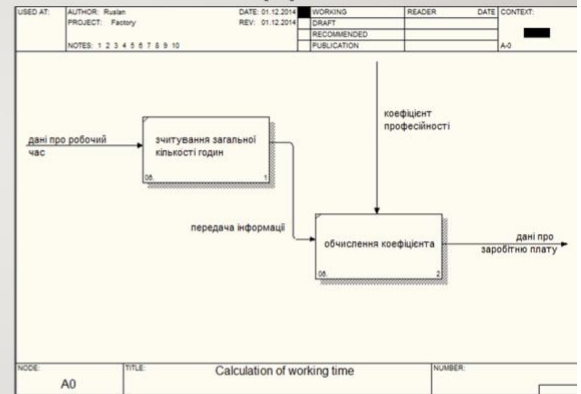
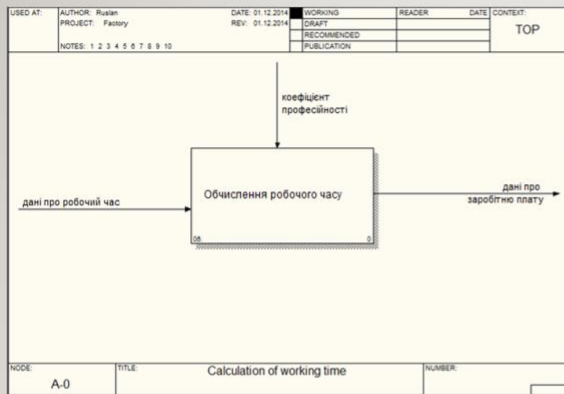


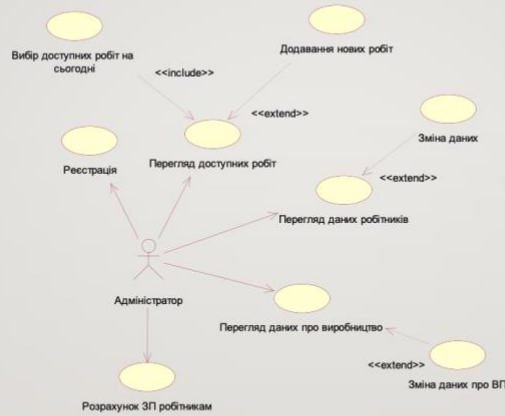
Схема архітектури Application Server програмного застосунку

ПРОЄКТУВАННЯ АРХІТЕКТУРИ МОДУЛІВ



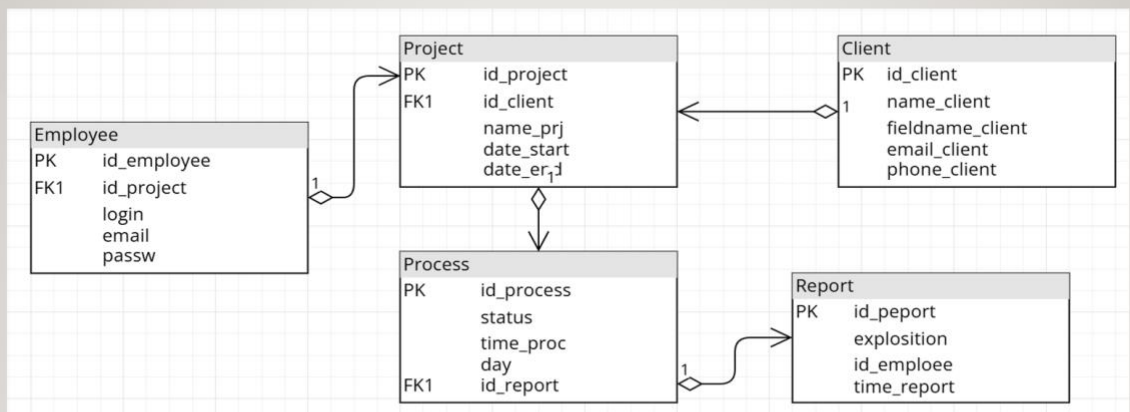
IDEF0-діаграми обчислення робочого часу для першого та другого рівнів

ОПИС ДЕКОМПОЗИЦІЇ, ЗАЛЕЖНОСТЕЙ, ІНТЕРФЕЙСІВ

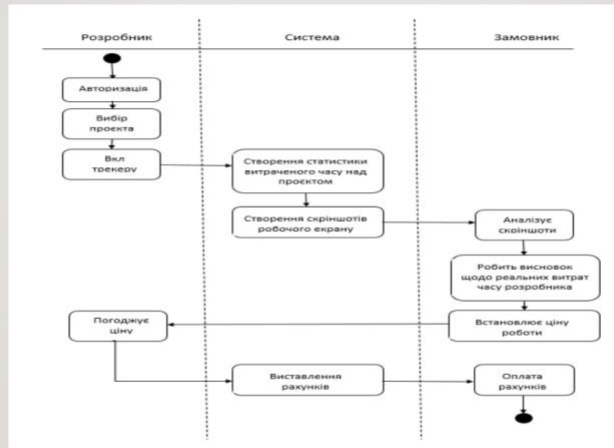


Дії адміністратора в даній системі

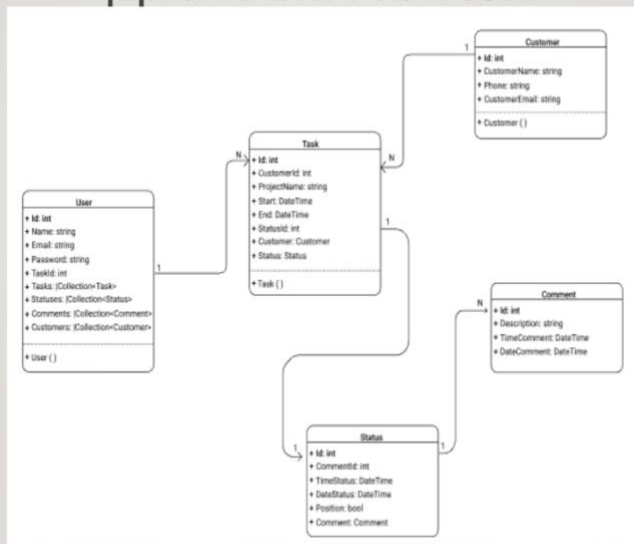
Діаграма бази даних



ПРОЄКТУВАННЯ МОДУЛІВ І ДАНИХ



ДІАГРАМА КЛАСІВ



Вимоги до технічного та програмного забезпечення

Серверна частина нашого застосунку розроблена на базі:
мов програмування Python та JavaScript, які обробляють вхідні запити від клієнтів;
візуальне середовище програмування та розробки VS Code;
середовище виконання: Node.js;
модуль mysql-python для роботи з базою даних;
серверна СКБД MySQL.

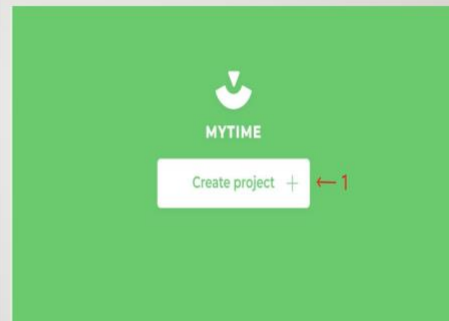
Для роботи серверної частини необхідно мінімальне технічне обладнання:
процесор з двома ядрами із частотою не менше 1,5 GHz;
оперативної пам'яті від 4 GB та більше;
постійної пам'яті не менше 5 GB для самого додатку, СКБД, даних та лог-файлів;
інтернет-зв'язок через будь-якого провайдера, що не блокує порти для telegram каналів.

Клієнтська частина розроблена на Python та JavaScript, які обробляють вхідні запити від клієнтів, та відповідає за створення інтерфейсу, інтегрує різні сучасні інструменти розробки та проектування, та забезпечує роботу з API месенджера Telegram.

РЕАЛІЗАЦІЯ МОДУЛІВ

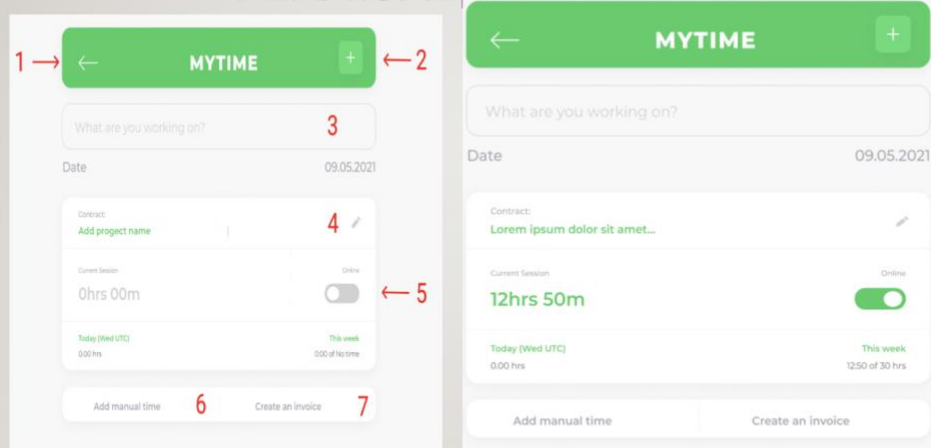


Модуль для входу в систему



Модуль створення проектів та зadanь

РЕАЛІЗАЦІЯ МОДУЛІВ



Модулі обчислення часу роботи

ВИСНОВКИ

В кваліфікаційній роботі було продемонстровано архітектуру додатку, розроблено базу даних, здійснено програмну розробку застосунку, здійснено тестування, послідовність виконання тестування, для перевірки відповідності програмного застосунку зазначеним функціональним вимогам, які наведено в технічному завданні до розробки програмного застосунку з розрахунку робочого часу співробітників підприємства.

Наведено детально інструкцію користувача

Також вирішується тут проблема розрахунку заробітної плати працівників на погодинній основі, оскільки це було б дуже складно робити вручну без автоматизації, коли є досить велика кількість працівників і частота їх появи змінюється. Програма автоматизує цей процес, розраховуючи години і заробіток до виплати для кожного співробітника, який працює з проектами.

ДЯКУЮ ЗА УВАГУ!

ГРАФІЧНА ЧАСТИНА

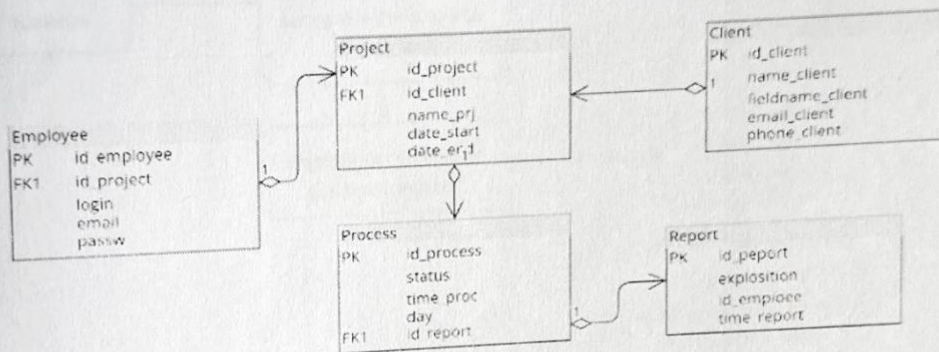
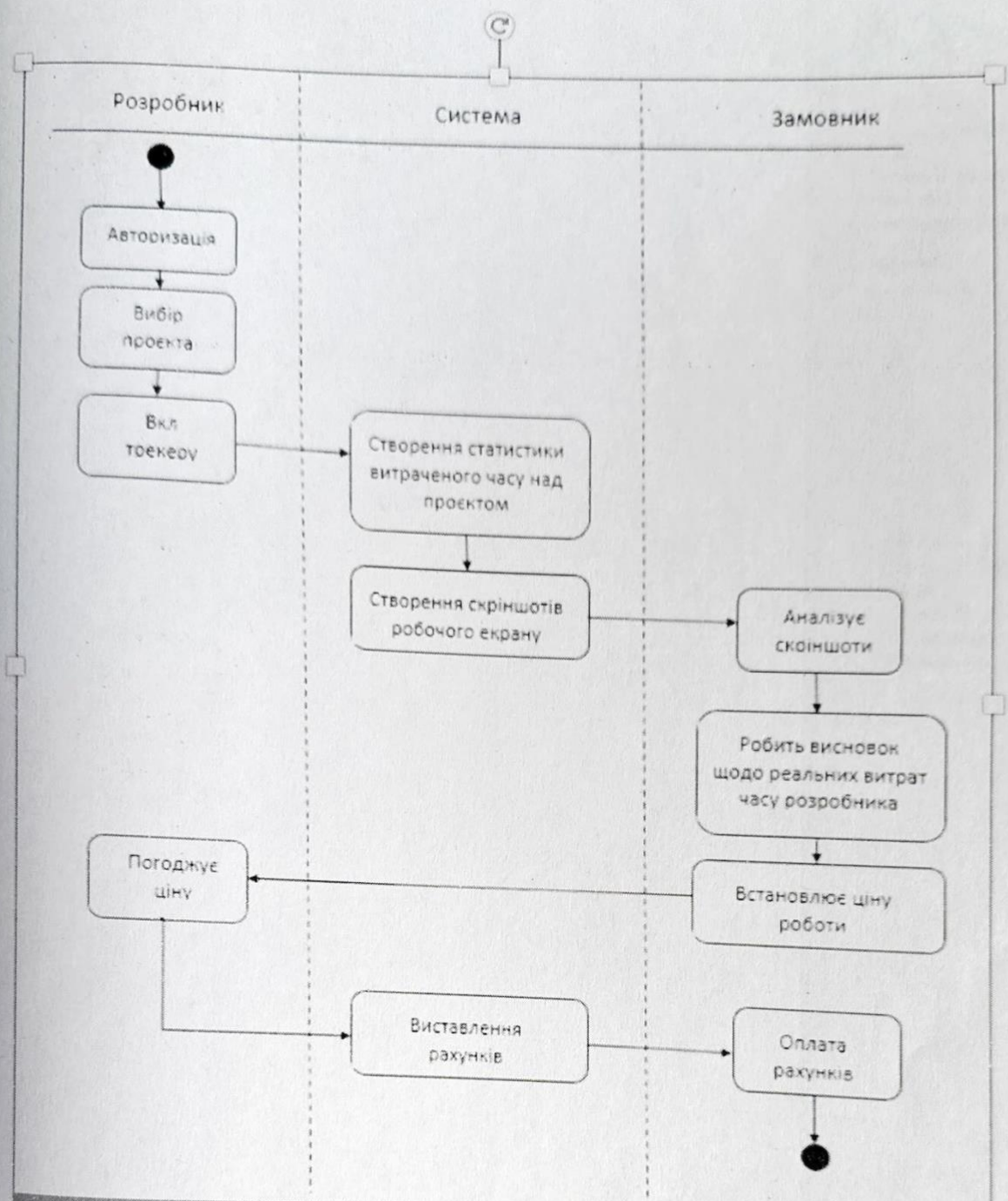


Рисунок 3.1.1 – Фізична модель бази даних

				КвРЦЗ. 2101094.01.20.ПЗ		
				Програмний застосунок для розрахунку робочого часу співробітників		
				Лист		Маса
				Аркуш X		Аркушів X
				ХНУ, ІПЗ-21-1		
Зам.	Арк.	№ докум.	Підпис	Дата		
Розробив		Черкасов Д. О.	<i>[Signature]</i>	1.06.		
Керівник		Форкун Ю. В.	<i>[Signature]</i>	01.06.		
Консульт.						
Перевірив						
Н. Коопр.		Яшина О. М.	<i>[Signature]</i>	10.06.		
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	10.06.		



Alt Text: A diagram of a company's application workflow.

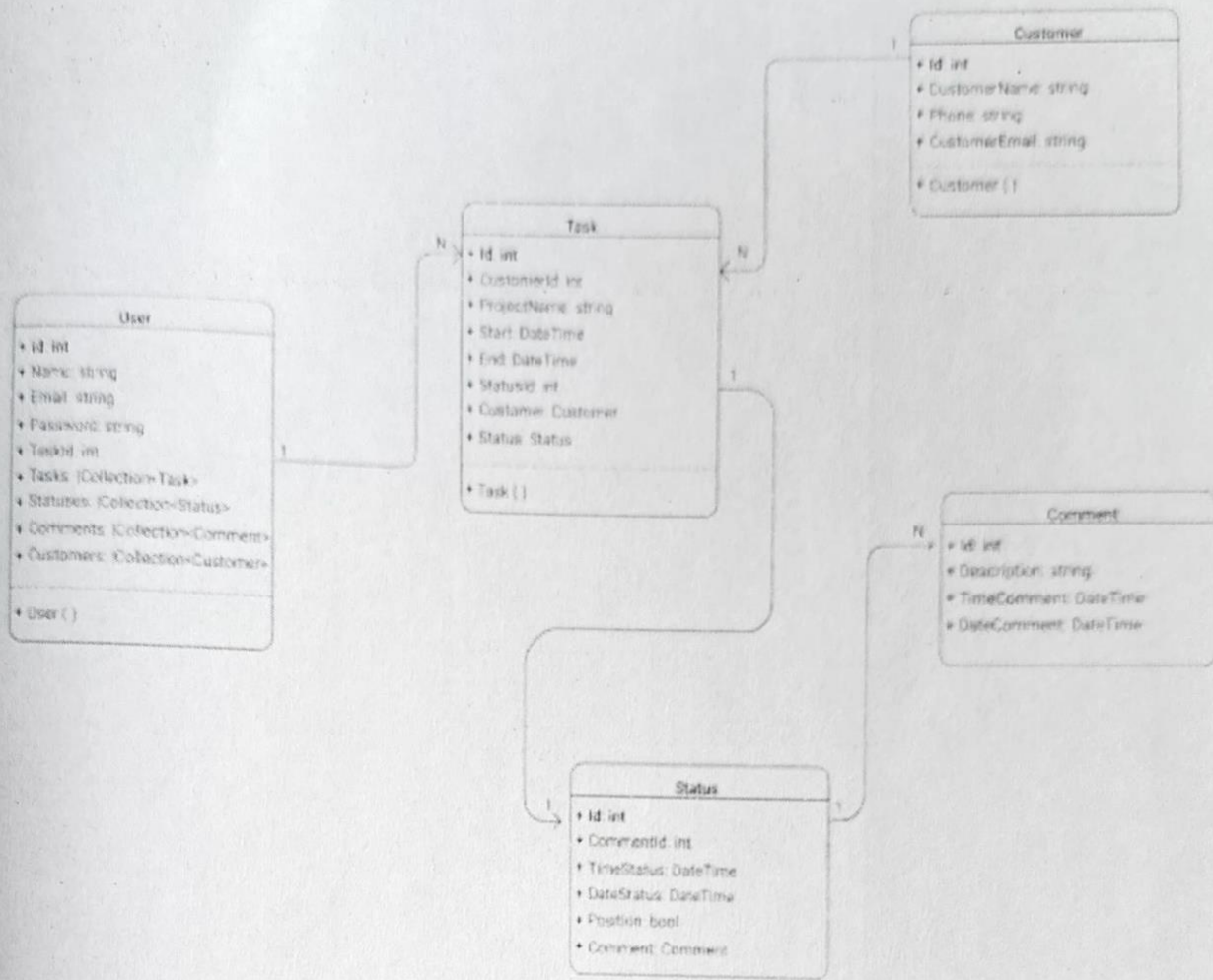
КвРІПЗ. 2101094.01.20.ПЗ

Програмний застосунок
для розрахунку робочого
часу співробітників

Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Черкасов Д. О.	<i>[Signature]</i>	1.06
Керівник		Форкун Ю. В.	<i>[Signature]</i>	10.06
Консульт.				
Перевірів				
Н. Контр.		Яшина О. М.	<i>[Signature]</i>	10.06
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	10.06

Літ.	Маса.	Масштаб
Аркуш X	Документів X	

ХНУ, ІПЗ-21-1



КвРПЗ. 2101094.01.20.ПЗ

Програмний застосунок
для розрахунку робочого
часу співробітників

Лім. Маса. Масштаб

Аркуш X Аркушів X

ХНУ, ІПЗ-21-1

Зам.	Дрк.	№ докум.	Підпис	Дата
Розробив		Черкасов Д. О.	<i>[Signature]</i>	1.06
Керівник		Фаркун Ю.В.	<i>[Signature]</i>	1.06
Консульт.				
Перевірив				
Н. Контр.		Яшина О.М.	<i>[Signature]</i>	10.06
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	10.06

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії
програмного забезпечення проф.
Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Черкасова Дениса Олексійович
факультет ІТ, ІV курс, група ІПЗ-21-1

ЗАЯВА

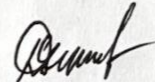
З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.02.25

дата



підпис

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Черкасов Денис Олексійович

Тема Програмний застосунок для розрахунку робочого часу співробітників підприємства

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 58.

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі розглянуто створення програмного застосунку для розрахунку робочого часу співробітників підприємства. Проведено аналіз предметної області, сформульовано вимоги, обґрунтовано архітектурні та технологічні рішення. Роботі було розроблено базу даних, здійснено програмну розробку застосунку, здійснено тестування, послідовність виконання тестування, для перевірки відповідності програмного застосунку зазначеним функціональним вимогам, які наведено в технічному завданні до розробки програмного застосунку з розрахунку робочого часу співробітників підприємства.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі обґрунтовано актуальність теми, сформульовано мету, завдання, об'єкт і предмет дослідження. У першому розділі проведено змістовний аналіз предметної області. Проаналізовано сучасні сервіси, їх функціонал, переваги та недоліки. Виділено ключові проблеми, які вирішує розроблюваний програмний застосунок, сформульовано функціональні й нефункціональні вимоги. У другому розділі спроектовано архітектуру вебсистема на основі шаблону AS та клієнт-серверної моделі. Розроблено структуру бази даних, повну ER-діаграму, структуру взаємодії між компонентами. Обґрунтовано вибір стеку технологій. Створено інтерфейс користувача з урахуванням зручності використання. У третьому розділі реалізовано серверну і клієнтську частини, створено базу даних, функціональні модулі для створення завдань, редагування, розподілення відповідальності. Проведено тестування вебсистеми та проаналізовано результати, що підтверджують відповідність функціоналу поставленим вимогам.

4. Позитивні сторони роботи Актуальність обраної теми полягає у потребі зручного цифрового інструменту для організації і контролю виконання проєктних завдань. Робота відзначається глибоким аналізом аналогів, якісним проєктуванням, застосуванням сучасних технологій (JAVA, MySQL, React, Node.js, JWT), а також орієнтацією на зручність користувача.

5. Негативні сторони роботи Обмежена реалізація фільтрації завдань – доцільно було б додати фільтрацію по дедлайнах. Також варто передбачити можливість спільного редагування проектів та завдань.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення відповідає тематиці роботи та містить діаграми архітектури, послідовностей, компонентів тощо. Пояснювальна записка оформлена згідно вимог і має логічну структуру викладення матеріалу.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота є завершеною, цілісною та технічно грамотною. У пояснювальній записці видно послідовність дій — від аналізу задачі до її практичного втілення. Сучасні технології та обрані архітектурні рішення дали змогу створити працездатний сервіс із достатнім рівнем зручності. Матеріал структурований, а наведені ілюстрації доречно доповнюють основний текст.

8. Інші зауваження _____

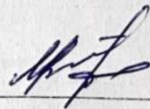
9
. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі. відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ

Вікторівна *в.т.н., доцент кафе. КІІС, Капустен Марія*

“ _____ ”
(підпис)

202 _____ р.



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Програмний застосунок для розрахунку робочого часу співробітників підприємства

Автор Черкасов Денис Олексійович

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Рівень вищої освіти Перший (бакалаврський)

Спеціальність 121 «Інженерія програмного забезпечення»

Науковий керівник Форкун Юрій Вікторович, канд. техн. наук, доцент

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

Висновок	а про відповідність
академічного плагіату	
Заявлені, виявлені в роботі, є законними і не є академічним плагіатом (далі – плагіатом). Робота відповідає вимогам до захисту.	відповідає
Запозичення не є академічним плагіатом, розміщені в розділах, які не описують суттєво авторське дослідження, але кількість цитат перевищує обсяг, виправданий цією метою роботи (далі – зазначаються детальні та аргументовані підстави для запозичень до правомірних). Робота приймається до захисту, але має бути виправана.	
Запозичення не є академічним плагіатом, але частково розміщені в розділах, які не описують суттєво авторське дослідження, а кількість цитат перевищує обсяг, виправданий цією метою роботи. Робота може бути допущена до захисту після виправання та доопрацювання і успішно пройде повторну перевірку на наявність плагіату.	
Виявляє навмисні текстові спотворення, передбачувані спроби укривтя текстових даних або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
Виявляє порушення академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, завдання, анотація, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій та у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 4,8%, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

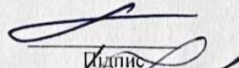
Дата

Завідувач кафедри


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Юрій ФОРКУН
Ім'я, ПРІЗВИЩЕ

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Черкасов Денис

Співавтор:

Назва: БКР_Програмний застосунок для розрахунку робочого часу співробітників підприємства Черкасов

Експерт:

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1:8.2%

Коефіцієнт подібності 2:2.6%

Мікропробіли: 0

Заміна букв: 5

Інтервали: 0

Білі знаки: 4

Дата створення звіту: 2025-06-11 05:14:28.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата

11-06-25

експерт



Anti-Plagiarism (UA) v-15.281 Educational**The maximum coincidence with one document 5.0%****Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 7%**

ID: 244368 Title: БКР_Програмний застосунок для розрахунку робочого часу співробітників підприємства_Черкасов Added in a DB: 2025-06-10 Authors: Черкасов Денис Heads: Форкун Ю.В., кад. техн.наук, доцент Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	44967	653	3331 (7%)	44 (7%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes