

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Інформаційних управляючих систем  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Дослідження моделей і методів оцінювання трудовитрат для  
ІТ-проєкту веб-застосунку  
(тема)

Виконав:

студент 2 курсу, групи УПГІТМ-21-1

Денис ПОТРЯСОВ

(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проєктами в  
галузі інформаційних технологій

(повна назва освітньої програми)

Керівник проф. каф. ІУС Максим ЄВЛАНОВ

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, прізвище)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри   
(підпис)

« 3 » квітня 2023 р.

**ЗАВДАННЯ**

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Потрясову Денису Євгеновичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження моделей і методів оцінювання трудовитрат для ІТ-проєкту веб-застосунку».

Затверджена наказом університету від 03 квітня 2023 р. № 319 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16.05.2023р.

3. Вихідні дані до роботи: науково-технічні публікації та інтернет-джерела з тематики кваліфікаційної роботи

4. Перелік питань, що потрібно опрацювати в роботі: формування проблеми щодо модифікації; огляд та загальний аналіз проблеми індивідуального завдання; вибір цілі та напрями удосконалення; постановка завдань до вирішення даної проблеми; розробка модифікованого методу оцінки трудовитрат ІТ-проєкту з розробки веб-додатку; розробка методології для модифікованої моделі оцінювання трудовитрат для ІТ-проєкту веб-застосунку; апробація оригінального та модифікованого методу об'єктних точок під час оцінювання трудовитрат ІТ-проєкту з розробки веб-застосунку.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та джерел	03.04.2023	Виконано
2	Опис постановки задачі дослідження	03.04.2023-04.04.2023	Виконано
3	Обробка матеріалів передатестаційної практики	05.04.2023-06.04.2023	Виконано
4	Аналіз існуючих методів оцінки трудовитрат з розробки веб-застосунків	06.04.2023-07.04.2023	Виконано
5	Розробка модифікованого методу об'єктних точок	08.04.2023-09.04.2023	Виконано
6	Розробка методології для модифікованого методу для розробки веб-додатків	08.04.2023-09.04.2023	Виконано
7	Апробація модифікованого методу об'єктних точок для розроблення веб-додатків	10.04.2023-14.04.2023	Виконано
8	Підготовка та оформлення пояснювальної записки та графічного матеріалу до атестаційної роботи	14.04.2023-16.04.2023	Виконано
9	Підготовка презентаційних матеріалів	17.04.2023-11.05.2023	Виконано
10	Подання студентом роботи для перевірки на плагіат	11.05.2023	Виконано
11	Надання роботи на підпис науковому керівнику	09.05.2023	Виконано
12	Попередній захист роботи	10.05.2023	Виконано
13	Надання роботи на рецензію	10.05.2023	Виконано
14	Надання роботи на підпис завідувачу кафедри	12.05.2023	Виконано
15	Захист кваліфікаційної роботи	17.05.2023	Виконано

Дата видачі завдання 03 квітня 2023 р.

Студент \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

проф. каф. ІУС Максим ЄВЛАНОВ

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 80 сторінок, 16 рисунків, 7 таблиць, 12 джерел, 1 додаток.

ВЕБ-ДОДАТОК, КАЛЕНДАРНИЙ ПЛАН, МЕТОД, МОДИФІКАЦІЯ МЕТОДУ, ПРОЄКТ, ОЦІНКА ТРУДОВИТРАТ, СПИСОК РОБІТ, ONION АРХІТЕКТУРА.

Актуальність цієї роботи проявляється в необхідності модифікування методу об'єктних точок, який надав би можливість правильно та ефективно оцінювати трудовитрати ІТ-проєкту з розробки веб-застосунку з багат шаровою архітектурою та особливо проєкти з onion архітектурою ніж існуючий метод.

Об'єктом дослідження кваліфікаційної роботи є процес оцінювання трудовитрат ІТ-проєкту з розробки веб-застосунків.

Метою досліджень кваліфікаційної роботи є дослідження моделей та методів оцінки трудовитрат ІТ-проєкту з розробки веб-застосунків, та модифікація моделі оцінки даної задачі, який буде більш ефективний, ніж існуючий метод та ефективніший у використанні.

Методи дослідження: аналіз та модифікація методу об'єктних точок для оцінювання трудовитрат ІТ-проєктів з розробки веб-застосунків на архітектурі, який буде ефективнішим існуючого.

Отримані наукові результати: розроблено модифікований метод оцінки трудовитрат ІТ-проєкту з розробки веб-застосунків; запропоновано алгоритм реалізації модифікованого методу; запропоновано опис використання модифікованого методу, запропонована апробація модифікованого методу.

Результати кваліфікаційної роботи можуть бути впроваджені в діяльність ІТ-компаній, які займаються розробкою веб-додатків й не тільки.

Проведена апробація модифікованого методу об'єктних точок на веб-додатку з onion-архітектурою показала, що даний метод дійсно дозволяє більш ефективно та правильно визначити оцінку трудовитрат ІТ-проєкту з розробки веб-застосунків ніж існуюча модель.

## **ABSTRACT**

The explanatory note to the attestation work contains: 80 pages, 16 figures., 7 tables, 12 sources, 1 attachment.

WEB APP, CALENDAR PLAN, METHOD, METHOD MODIFICATION, PROJECT, LABOR ESTIMATE, WORK LIST, ONION ARCHITECTURE.

The relevance of this work is manifested in the need to modify the method of object points, which would provide an opportunity to correctly and efficiently estimate the labor costs of an IT project for the development of a web application with a multilayer architecture, and especially projects with an onion architecture, than the existing method.

The object of the research of the qualification work is the process of estimating the labor costs of the IT project for the development of web applications.

The purpose of the research of the qualification work is to study the models and methods of estimating the labor costs of the IT project for the development of web applications, and to modify the model of the assessment of this task, which will be more effective than the existing method and more efficient to use.

Research methods: analysis and modification of the method of object points for estimating the labor costs of IT projects for the development of web applications on the onion architecture, which will be more effective than the existing one.

Obtained scientific results: a modified method of estimating the labor costs of an IT project for the development of web applications was developed; an algorithm for implementing the modified method is proposed; a description of the use of the modified method is offered, the approbation of the modified method is offered.

The results of the qualification work can be implemented in the activities of IT companies that are engaged in the development of web applications and not only.

The approbation of the modified method of object points on a web application with onion architecture showed that this method really allows for a more effective and correct estimation of the labor costs of an IT project for the development of web applications than the existing model.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	9
ВСТУП .....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....	12
1.1 Огляд та загальний аналіз проблеми індивідуального завдання .....	12
1.2 Огляд та дослідження Onion-архітектури та її шарів.....	13
1.3 Аналіз методів оцінок трудовитрат в ІТ проєктах .....	16
2 МОДИФІКАЦІЯ МЕТОДУ ОБ'ЄКТНИХ ТОЧОК ЗА ONION АРХІТЕКТУРОЮ .....	22
2.1 Метод об'єктних точок.....	22
2.2 Проблема методу об'єктних точок.....	26
2.3 Модифікація методу на базі onion-архітектури .....	27
3 РОЗРОБКА МЕТОДИКИ ДЛЯ МОДИФІКОВАНОЇ МОДЕЛІ ОЦІНЮВАННЯ ТРУДОВИТРАТ ДЛЯ ІТ-ПРОЄКТУ ВЕБ-ЗАСТОСУНКУ	31
3.1 Методика використання оригінального методу.....	31
3.2 Методика використання модифікованого методу .....	37
4 АПРОБАЦІЯ ОРИГІНАЛЬНОГО ТА МОДИФІКОВАНОГО МЕТОДУ ОБ'ЄКТНИХ ТОЧОК ПІД ЧАС ОЦІНЮВАННЯ ТРУДОВИТРАТ ІТ-ПРОЄКТУ З РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ .....	39
4.1 Статут проєкту.....	46
4.2 Приклад використання оригінального методу.....	49
4.3 Приклад використання модифікованого методу об'єктних точок .....	55
ВИСНОВОК.....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	60
ДОДАТОК А Графічний матеріал .....	<b>Ошибка! Закладка не определена.</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ПЗ – програмне забезпечення;

СУБД – система управління базами даних;

CASE – Computer-Aided Software Engineering;

COCOMO – COConstructive COst Model;

CSS – Cascading Style Sheets;

DIP – Dependency Inversion Principle;

HTML – HyperText Markup Language;

MSSQL – Microsoft SQL Server;

SLIM – Software Lifecycle Management;

UI – User Interface;

WBS – Work Breakdown Structure.

## ВСТУП

Кожне підприємство з організацією своєї діяльності має вести журнал обліку проведених робіт. Розумно організований облік виконаних робіт є необхідною умовою функціонування будь-якого виду бізнесу. Облік виконаних робіт та наданих послуг дає можливість проводити аналіз продуктивної роботи на підприємстві, контролювати стан розробки, оперативно реагувати на зміни довкілля та вирішувати внутрішні проблеми.

Розроблені на початку 2000-х років моделі та методи оцінювання витрат на виконання ІТ-проектів засновані на тих архітектурах, які були найпоширенішими у 1990-х роках. Застосування цих моделей та методів для оцінювання витрат на виконання ІТ-проектів розробки нових ІТ-продуктів, заснованих на сучасних архітектурах, може призвести до свідомо неправильних оцінок. Тому існує необхідність у вдосконаленні існуючих моделей та методів оцінювання трудовитрат на виконання ІТ-проекту, яке б враховувало особливості сучасних архітектур ІТ-продуктів.

Таким чином, головною метою цієї роботи є модифікація існуючого методу який не пристосований для коректного розрахунку сучасних багатошарових проектів, у метод який буде більш ефективніший та більш точний та легкий у використанні ніж існуючий.

В якості основної архітектури моделі веб-додатку було обрано Onion-архітектуру. Onion-архітектура є поділом веб-додатку на рівні. В цій архітектурі є один незалежний рівень, що знаходиться у центрі архітектури. Від цього рівня залежить другий рівень, від другого – третій тощо. Тобто виходить, що довкола першого незалежного рівня нашаровується другий-залежний. Навколо другого нашаровується третій, який може залежати і від першого.

В якості основного методу було обрано метод об'єктних точок. Використання методу об'єктних точок є оцінювання розміру ПЗ в умовах

використання взаємопов'язаних прототипів з використанням інтегрованого CASE-середовища прискореної розробки.

Тому метою дослідження є вдосконалення існуючої моделі які забезпечать можливість оцінювання трудовитрат на розробку веб-додатків з Onion-архітектурою.

Робота виконується відповідно до вимог методичних вказівок [1].

Для оформлення даного звіту використовувалися державні стандарти України [2];[3].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Огляд та загальний аналіз проблеми індивідуального завдання

В епоху інформаційних технологій, що швидко розвиваються, зростаючої кількості високобюджетних проєктів у галузі розробки програмного забезпечення, дуже важливим стає вміння оцінити на ранніх етапах можливі вигоди та збитки від проєкту, проаналізувати можливі сценарії розвитку подій. Помилка, недооцінка складнощів, з якими має зіткнутися в процесі розробки, переоцінка сил команди розробників, неприйняття до розрахунку тих чи інших факторів часто призводить до фінансових втрат і навіть банкрутства ІТ-компаній. За статистикою лише чверть всіх розпочатих проєктів завершується своєчасно та у повному обсязі, чверть взагалі скасовується, і половина проєктів завершується з перевищенням бюджетних витрат чи з запізненням.

Також у частини завершених проєктів з перевищенням термінів та вартості проєктів частково урізано функціонал, який був зазначений у технічному завданні. Це призводить до додаткових збитків компанії розробника, якщо такі ситуації не були передбачені у контракті з замовником.

При розробці складних програмних систем, які зазвичай входять до складу корпоративних інформаційних систем, необхідно знизити залежність якості результатів від таких суб'єктивних факторів, як кваліфікація виконавців, їх досвід, організація процесу розробки. І тому потрібні промислові технологічні методи оцінки ресурсів, необхідні розробки програмного забезпечення.

Основними причинами помилок оцінки є:

- невизначеність на ранніх стадіях проєктування;
- недосконалість організації процесу розробки;
- неповні технічні вимоги;
- оптимістична та суб'єктивна оцінка трудомісткості завдань.

Тому менеджер проєкту в компанії розробника потребує інструменту, що дозволяє якомога точніше оцінити трудовитрати та вартість проєкту, а за великої кількості проєктів цей процес має бути автоматизований.

ІТ-компанії є піонерами в галузі автоматизованого обліку та планування робіт. Існує велика кількість програм-органайзерів, планувальників та інших інструментальних засобів, що дають змогу розподіляти виконання тих чи інших видів творів за виконавцями. Проблема полягає у тому, що ефективне управління залежить від обраної архітектури сучасного web-додатку. Тема є актуальною, бо кількість ІТ-проєктів зростає з великими темпами, а це означає, що зростає і необхідність визначення підходів до планування робіт, виходячи з особливостей та архітектури кожного проєкту.

В якості основної архітектури моделі веб-додатку було обрано розглянути Onion-архітектури.

## 1.2 Огляд та дослідження Onion-архітектури та її шарів

Архітектура Onion - це паттерн архітектури програмного забезпечення, яка була продемонстрована Microsoft і використовується для створення додатків, які легко підтримуються, масштабуються і тестуються. Onion архітектура заснована на концепції шарів, при цьому кожен шар відповідає за певний аспект функціональності, що надається додатком (рисунок 1.1). [4]

# Onion Architecture

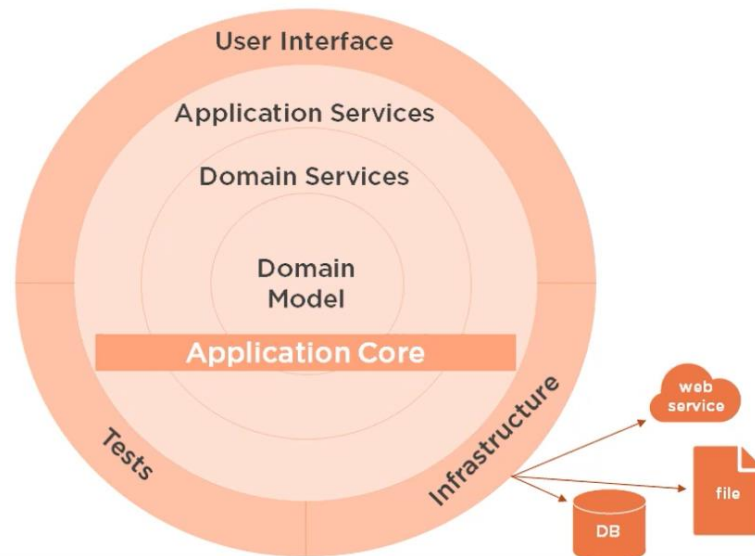


Рисунок 1.1 - Схема Onion-архітектури

Технологія розроблена для того, щоб бути вільно з'єднаною, що полегшує підтримку та модифікацію програми з часом.

Onion архітектура складається з чотирьох основних шарів:

- презентаційний рівень: це самий поверхневий рівень додатку, і той, який спілкується з користувачем. Він складається з UI компонентів, таких як веб-сайти, контролери та інші, всі з яких керують введеннями і виведеннями від користувачів.

- прикладний рівень: Це рівень де зберігається бізнес-логіка програми. Цей шар обробляє варіанти використання та регулює взаємодію між різними рівнями. Він не залежить від будь-якої конкретної технології або інфраструктури і може бути випробуваний самостійно;

- доменний рівень: Цей рівень зберігає всі конкретні для додатка об'єкти домену і бізнес-правила. Зображує ядро додатку і відповідає за те, як працює додаток. Він не залежить від будь-якої конкретної технології і може бути перероблений для використання в різних додатках;

– інфраструктурний рівень: Цей рівень містить деталі реалізації додатка, такі як бази даних, файлові системи або зовнішні служби. Він надає додатку необхідну інфраструктуру для роботи і зв'язку з іншими системами.

Основні переваги Onion архітектури:

– onion архітектура дотримується принципу DIP (Принцип інверсії залежностей), який стверджує, що модулі високого рівня не повинні залежати від модулів низького рівня, а скоріше від їх абстракцій. Це значить, що зовнішні рівні не повинні залежати від внутрішніх рівнів, замість цього залежності повинні бути інвертовані, при цьому внутрішні рівні залежать від абстракцій зовнішніх рівнів;

– технологія сприяє чіткому розподілу завдань між рівнями і зменшує зв'язок між ними. Це спрощує тестування додатка і його модифікацію, не зачіпаючи інших частин системи. Крім того, він підтримує принципи інверсії залежностей і інверсії управління, які дозволяють зробити додаток більш модульним і розширюваним;

– підтримує масштабованість, дозволяючи розділити додаток на дрібніші незалежні модулі, які можна легко масштабувати по горизонталі. Архітектура також підтримує вертикальне масштабування, що дозволяє оптимізувати рівні продуктивності;

– адаптована архітектура, що означає, що вона може бути модифікована для задоволення потреб широкого спектру застосувань і специфікацій. Оскільки шари можна модифікувати або розширювати, не впливаючи на інші компоненти системи, набагато простіше додавати нові функції або змінювати поведінку додатка;

– оскільки архітектура не прив'язана до будь-якої конкретної технології або фреймворку, а це означає, що її можна реалізувати за допомогою різних мов програмування та інструментів. Microsoft надає деякі рекомендації та приклади того, як реалізувати архітектуру Onion за допомогою .NET, але принципи можуть бути застосовані і до інших платформ.

В цілому, архітектура Onion надає потужний і гнучкий підхід до побудови складних програмних систем. Це вимагає ретельного планування та проєктування, але може призвести до створення зручного у супроводі та масштабованого додатка, яке легше модифікувати та тестувати.

### 1.3 Аналіз методів оцінок трудовитрат в ІТ проєктах

Оцінка трудовитрат у ІТ-проєктах є невід'ємною частиною планування та управління проєктами. Точна оцінка трудовитрат допомагає гарантувати, що проєкт буде завершено вчасно та в рамках бюджету.

Існує безліч різних моделей оцінки вартості. При виборі для конкретної компанії найбільш адекватної моделі, необхідно визначити критерії, якими можна оцінити пропоновані моделі.

Методи та моделі оцінки вартості ПЗ поділяються на дві групи: неалгоритмічні методи (Price-to-win, Bottom-Up, експертна оцінка, оцінка за аналогією, параметрична оцінка та ін.) та алгоритмічні моделі (SLIM та СОСОМО).

Сутність неалгоритмічних методів у тому, що з оцінці вартості ПЗ використовуються певні схеми та принципи, а не математичні формули.

До цих методів належать такі:

1. Експертна оцінка: Експертна оцінка включає консультації з експертами в предметній галузі та керівниками проєктів, які мають досвід роботи з аналогічними проєктами.[5] Ці експерти можуть дати уявлення про вимоги проєкту, задіяні завдання та ресурси, необхідні для завершення проєкту. Думки експертів можуть бути використані для складання оцінки витрат на оплату праці.

Експертна оцінка може бути цінним ресурсом для оцінки трудовитрат у ІТ-проєктах, особливо для проєктів, що включають нові технології або

методології. Важливо проконсультуватися з експертами, які мають досвід роботи в конкретних галузях проекту і здатні дати уявлення про пов'язані завдання, необхідні ресурси, а також потенційні ризики та проблеми.

2. Аналогова оцінка: Аналогова оцінка включає використання фактичних витрат аналогічного проекту для оцінки трудовитрат нового проекту. Цей метод передбачає, що витрати на робочу силу для нового проекту будуть такими ж, як і для аналогічного проекту, залежно від розміру проекту, складності та інших факторів.

Аналогова оцінка може бути корисним методом, коли для нового проекту доступні обмежені дані або коли проект аналогічний попереднім проектам. Однак важливо переконатися, що попередній проект, використаний як основа для оцінки, можна порівняти за розміром, складністю та іншими важливими факторами. Також важливо скорегувати оцінку з урахуванням будь-якої різниці між проектами.

3. Оцінка «Bottom-Up»: оцінка «Bottom Up» включає розбивку проекту на більш дрібні завдання та оцінку трудовитрат для кожного завдання. Ці оцінки об'єднуються для створення загальної оцінки трудовитрат за проектом. Цей метод точніший, ніж інші методи, але може зайняти багато часу. Схема даного методу оцінювання зображена на рисунку 1.2[6]:

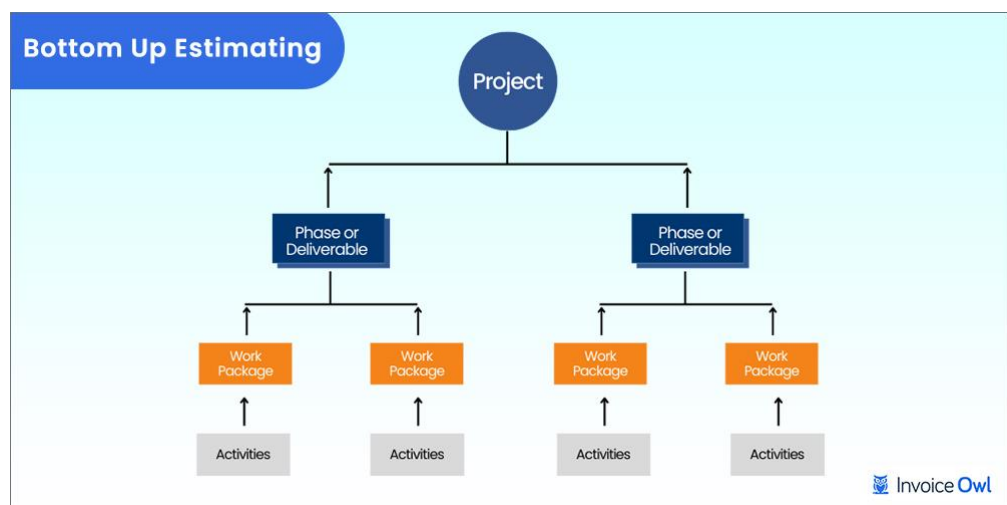


Рисунок 1.2 – Схема методу оцінювання «Bottom Up»

Оцінка «Bottom Up» може дати детальну та точну оцінку витрат на робочу силу, але вона може зайняти багато часу і може виявитися нездійсненною для великих чи складних проєктів. Щоб зробити процес більш ефективним, може бути корисно використовувати структурну декомпозицію робіт (WBS), щоб розбити проєкт на дрібніші завдання та оцінити трудовитрати для кожного завдання.

4. Оцінка за трьома точками. Оцінка за трьома точками включає створення трьох оцінок для кожного завдання: найкращий сценарій, найгірший сценарій і найбільш ймовірний сценарій. Ці оцінки використовуються для розрахунку середньозваженого значення, яке використовується для створення оцінки трудовитрат для завдання.

Оцінка за трьома точками може бути корисним методом для проєктів з високим ступенем невизначеності або ризику. Три оцінки (найкращий випадок, найгірший випадок і найімовірніший) можуть бути використані для розрахунку діапазону можливих витрат на робочу силу та виявлення потенційних ризиків та можливостей.

5. Параметрична оцінка. Параметрична оцінка включає використання статистичних даних для оцінки трудовитрат. Цей метод використовує історичні дані для створення математичної моделі, яка оцінює трудовитрати на основі розміру проєкту, складності та інших факторів. Цей метод корисний для проєктів, подібних до попередніх проєктів, для яких доступні статистичні дані. Параболічна схема оцінювання зображена на рисунку 1.3:

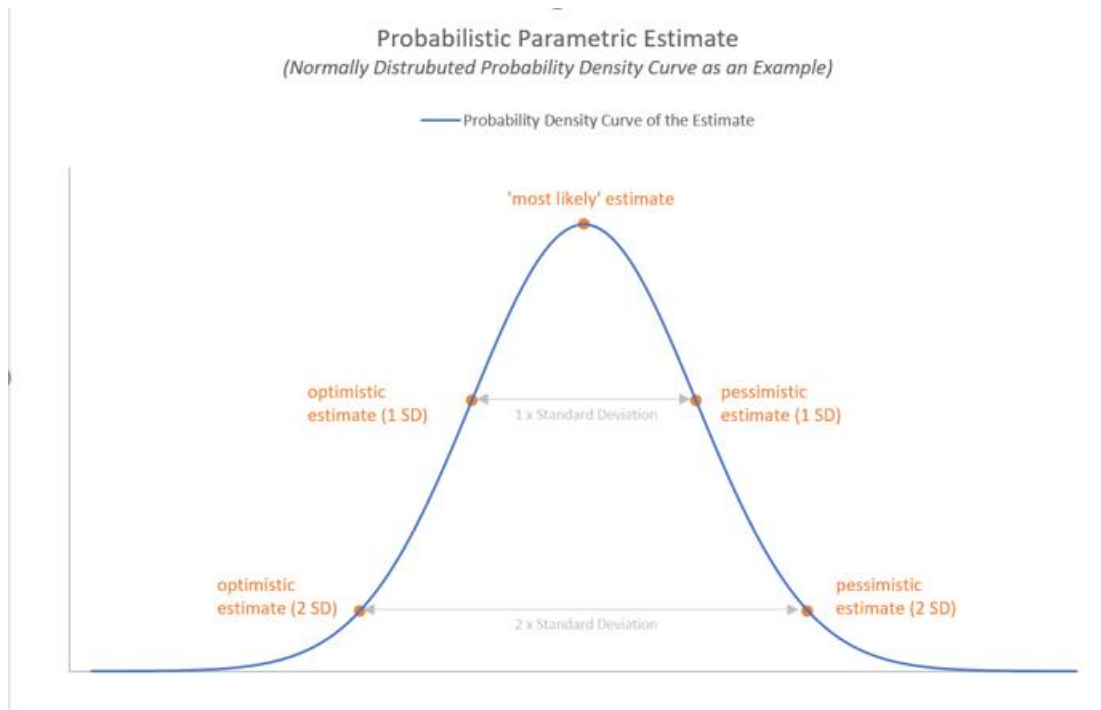


Рисунок 1.3 - Параболічна схема параметричної оцінки

Параметрична оцінка може бути потужним методом оцінки витрат на робочу силу, але вона потребує значного обсягу історичних даних та досвіду статистичного аналізу. Важливо переконатися, що дані, що використовуються для створення математичної моделі, є точними та актуальними для нового проєкту.

6. Аналіз резервів. Аналіз резервів включає виділення коштів на непередбачувані витрати на покриття непередбачених витрат на оплату праці, які можуть виникнути в ході проєкту. Резервні фонди можуть бути відсотком від загального кошторису витрат на робочу силу або фіксовану суму, засновану на ризиках і невизначеності проєкту.

7. Метод Price-to-win полягає в тому, що незалежно від передбачуваних реальних витрат на розробку проєкту, оцінка вартості ПЗ коригується відповідно до побажань замовника. Price-to-win фактично є політикою проведення переговорів із клієнтом, тому часто застосовується компаніями, які не мають коштів для якісної оцінки проєктів. Застосування

методу може призвести до нестачі ресурсів для виконання проєкту, невиконання термінів здачі проєкту і як результат – втрата контракту чи банкрутство компанії розробника.

Аналіз резервів може допомогти переконатися, що проєкт має достатньо коштів на покриття непередбачених трудовитрат чи ризиків. Важливо визначити потенційні ризики та фактори невизначеності, які можуть вплинути на проєкт, та виділити відповідну суму коштів на непередбачені витрати.

Моделі оцінки вартості ПЗ є однією або декількома функцій, що описують залежність між характеристиками проєкту та витратами на його реалізацію. На кшталт використовуваних функцій моделі поділяють на лінійні, мультиплікативні, статечні. По використанню історичних даних поділяють на емпіричні та аналітичні. Найчастіше реалізованими і добре документованими моделями є модель Путнема SLIM (статечна, аналітична) та модель COCOMO (статечна, емпірична).

Модель SLIM, створена Ларі Путнемом із компанії Quantitative Software Management, заснована на аналізі життєвого циклу програмного продукту на термінах релеївського розподілу персоналу проєкту в часу. Вона підтримує найбільш популярні методи виміру розміру коду, зокрема й функціональні точки. Це дозволяє використовувати криву для оцінки графіка проєкту та його дефектів. Для зміни форми кривої використовується коефіцієнт нарощування потужностей MBI (Manpower Buildup) та технологічний коефіцієнт фактор продуктивності PF (Productivity Factor). У програмній реалізації SLIM можна зберігати та аналізувати інформацію, отриману від інших, завершених проєктів, а потім використовувати її для калібрування моделі, або якщо така інформація відсутня, тоді користувач може відповісти на низку навідних питань, щоб розрахувати значення MBI та PF на основі існуючої бази даних.

Модель COCOMO, створена Баррі Боемом, спочатку ґрунтувалася на результатах аналізу 63 програмних проєктів різних типів. При цьому оцінювався фактичний розмір коду, трудовитрати, а також фактична тривалість розробки.

У моделі використовуються 3 режими, за допомогою яких різняться складність системи та середовища розробки – органічний режим, зблокований режим та впроваджений режим. Органічний режим застосовуємо до невеликих програмних продуктів, зблокованих до проєктів середньої величини, які вимагають невеликих інновацій. Впроваджений режим характеризується високою складністю, більшим обсягом інновацій та великою командою розробників.

Точність оцінки залежить від рівня деталізації, який може бути базовим (для виконання оцінки використовується лише значення розміру коду та відомості про поточний режим), проміжний (додаються значення 15 додаткових змінних, які отримують оцінку за 6-бальною шкалою) та деталізованим (додаються додаткові змінні).

З часом і зростанням вимог до систем, модель СОСОМО виявилася застарілою у значній своїй частині. Саме з цієї причини і ряд інших важливих проблем, було розроблено модель СОСОМО II, вперше опубліковану 1999 року.

Перевагою даної моделі в тому, що вона дозволяє досліднику використовувати фактичну інформацію та експертну оцінку. СОСОМО II використовує моделі композиції програми (об'єктні показники), раннього етапу проєктування та етапу пост-архітектури. Вони замінили базовий, проміжний та деталізований етапи моделі СОСОМО. Модель композиції програми дозволяє брати до уваги сучасні методи розробки, наприклад такі, як створення прототипу. Оскільки під час візуальної розробки оцінка розміру продукту за допомогою рядків коду не завжди є вірним. Натомість у моделі використовуються об'єктні показники. Це кількість екранних форм, звітів, модулів, кожен із яких співвідноситься з одним із трьох рівнів – простий, середній та складний відповідно до рівня складності.

Важливо відзначити, що оцінка трудовитрат - це не разова дія, а безперервний процес протягом усього проєкту.

## 2 МОДИФІКАЦІЯ МЕТОДУ ОБ'ЄКТНИХ ТОЧОК ЗА ONION АРХІТЕКТУРОЮ

### 2.1 Метод об'єктних точок

Метод об'єктних точок (метод об'єктних точок) - це метод оцінки трудовитрат, застосування в програмній інженерії для визначення розміру та складності продукту продукту. Він був розроблений у 1980-х роках Дж. Кемпанією, яка використовується в різних версіях програмної інженерії, включаючи розробку та тестування програмного забезпечення.

Декларованою галуззю використання методу об'єктних точок є оцінювання розміру ПЗ в умовах використання взаємопов'язаних прототипів з використанням інтегрованого CASE-середовища прискореної розробки.

Переваги методу об'єктних точок (в порівнянні з методом функціональних точок):

- менша дисперсія;
- значно менший час, необхідний для отримання оцінки;
- менша складність використання.

На основі методу об'єктних точок лежить концепція об'єктів та їх взаємодія у системі. Термін «об'єкт» в методі визначає як об'єкти екрани, звіти та 3GL-модулі.[7] Даний термін не пов'язаний із визначеннями терміну «об'єкт», які базуються на таких ознаках, як належність до класу, наслідування, інкапсуляція, передача повідомлень тощо. Взаємодія - це між об'єктами, які користуються потоками даних, управлінням або повідомленнями.

Для оцінки трудовитрат за методом об'єктних точок необхідне виконання наступних кроків:

Етап 1. Оцінити кількість об'єктів досліджуваного продукту як кількість екранів, звітів та 3GL-компонентів, які будуть входити в розроблюваний

додаток (при оцінюванні виходити із стандартного опису цих об'єктів у використуваному інтегрованому CASE-середовищі).

Етап 2. Класифікувати кожен екземпляр об'єкту за рівнем складності (простий, середній, високий рівні) залежно від значень характерних розмірів з використанням наступної схеми (таблиця 2.1).[8]

Таблиця 2.1 - Оцінки рівня складності екземплярів об'єктів в методі об'єктних точок

Для екранів				Для звітів			
Кількість використуваних представлень (Views)	Джерела у вигляді таблиць даних			Кількість використуваних секцій	Джерела у вигляді таблиць даних		
	Усього <4 (<2 srvr <3 clnt)	Усього <8 (2-3 srvr 3-5 clnt)	Усього 8+ (>3 srvr >5 clnt)		Усього <4 (<2 srvr <3 clnt)	Усього <8 (2-3srvr 3-5clnt)	Усього 8+ (>3 srvr >5 clnt)
<3	Простий	Простий	Середній	0-1	Простий	Простий	Середній
3-7	Простий	Середній	Складний	2-3	Простий	Середній	Складний
>8	середній	Складний	Складний	4+	Середній	Складний	Складний

Етап 3. Зважити результат класифікації за наступною схемою (таблиця 2.2). Ваги відображують витрати, які необхідні для реалізації відповідного екземпляру об'єкту вказаного рівня складності.

Таблиця 2.2 - Значення ваг рівнів складності об'єктів у методі об'єктних точок

Тип об'єкту	Рівень складності		
	Простий	Середній	Складний
Екранна форма	1	2	3
Звіт	2	5	8
3GL-КОМПОНЕНТ			10

Етап 4. Для отримання кількості об'єктних точок *ObjectPoints* просумувати значення ваг усіх зважених об'єктів.

Етап 5. Оцінити відсоток повторного використання, яке очікується в даному проєкті, та розрахувати кількість нових об'єктних точок за формулою (2.3).

$$NOP = \frac{(ObjectPoints) \times (100 - \%reuse)}{100}$$

Рисунок 2.3 - Формула розрахунку кількості нових об'єктних точок

Етап 6. Визначити коефіцієнт виробничості PROD за наступною схемою (таблиця 2.4).

Таблиця 2.4 - Значення коефіцієнту виробничості PROD у методі об'єктних точок

Досвід та спроможності розробників, зрілість та спроможності CASE	Дуже низький	Дуже низький	Нормальний	Високий	Дуже високий
PROD	4	7	13	25	50

Етап 7. Розрахувати оцінку кількості людино-місяців за формулою (2.5), які вимагаються для реалізації ІТ-проєкту з розробки досліджуваного продукту.

$$PM = \frac{NOP}{PROD}$$

Рисунок 2.5 - Формула розрахунку оцінки кількості людино-місяців, які вимагаються для реалізації ІТ-проєкту

Переваги методу об'єктних точок включають можливість швидкої оцінки трудовитрат, точність та об'єктивність оцінки, а також облік усіх основних аспектів системи. Тим не менш, метод має деякі обмеження, такі як складність визначення об'єктів та їх взаємодій, а також від досвіду використання і контексту проєкту.

## 2.2 Проблема методу об'єктних точок

В основі методу об'єктних точок лежить концепція об'єктів та їх взаємодія у системі. Термін «об'єкт» в методі визначає як об'єкти екрани, звіти та 3GL-модулі. Проблема полягає у тому що цей метод не розрахований на проекти з багатошаровою архітектурою.

У програмуванні модуль відноситься до самодостатньої одиниці коду, який можна імпортувати, повторно використовувати та скласти для створення більших програм. Модуль можна розглядати як будівельний блок, який інкапсулює певну функціональність або набір пов'язаних функцій і може використовуватися в різних контекстах.

Модулі 3GL (мови третього покоління) — це тип модулів, які використовуються в таких мовах програмування, як C, Pascal і Fortran. Ці модулі зазвичай містять набір функцій або процедур, які можна викликати з інших частин програми. Вони часто використовуються для модуляризації коду та сприяння повторному використанню коду.

Однією з ключових характеристик модулів 3GL є те, що вони зазвичай компілюються в машинний код або байт-код перед виконанням. Це може призвести до швидшої продуктивності порівняно з інтерпретованими мовами, оскільки скомпільований код можна оптимізувати для конкретного апаратного забезпечення та операційної системи, на якій він працюватиме.

Загалом модулі 3GL забезпечують структурований та ефективний спосіб організації та повторного використання коду, що може допомогти скоротити час розробки, сприяти зручності обслуговування коду та покращити загальну якість програмного забезпечення.

Коли метод був у розробці останній тип об'єктів, 3GL компоненти, припускався як програмні модулі реалізуючі бізнес логіку та взаємодію за базами даних. Допускали що для клієнт-серверної архітектури, схема якої

зображена на рисунку 2.6, на основі якої створювався даний метод мови 3го покоління забезпечують зборку бізнес логіки та логіку обміну з базами даних в один програмний модуль.

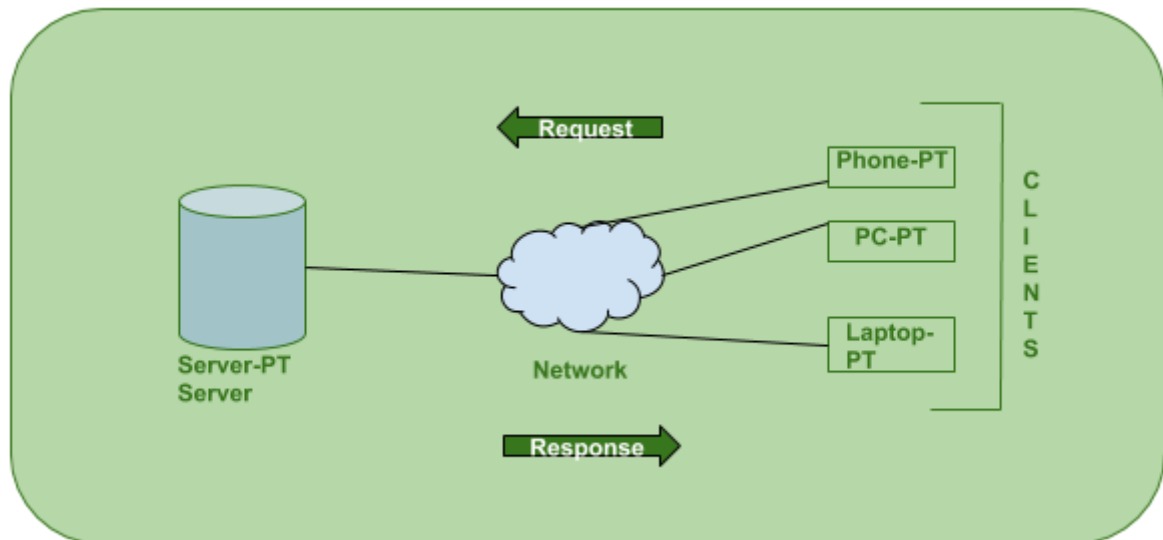


Рисунок 2.6 – Клієнт серверна архітектура

Тому вважається що в методі об'єктних точок 3GL модуль завжди буде один. Якщо подивитися на таблицю 2.2, то можна побачити, що цей модуль завжди тільки складний та його рівень складності – 10.

### 2.3 Модифікація методу на базі onion-архітектури

Якщо роздивлятися проекти з onion архітектурою, то твердження що в одному 3GL модулі мають бути всі функції не є точним тому що, ми формуємо «луковицю» як можливість доповнення, введення нових, заміни існуючих та видалення застарілих 3GL модулів. Кожен модуль реалізує свою відокремлену

функцію або свій набір операцій по обробці даних з якими ця «луковиця» взаємодіє.

Система припускає декілька рівнів відділяючі модулі роботи з базами даних від бізнес логіки роботи програмного веб-застосунку, яка може бути багаторівневою. Зрештою виходячи на шар інтерфейсу з користувачем який отримує дані не бачачи, що ховається за шаром «луковиці».

Виходячи з вище написаного, пропонується модифікувати цей метод таким чином: замість одного 3GL модуля у якому була бізнес логіка та логіка обміну даними, зробити N кількість модулів в яких були б закладений ряд функцій, та створити нову таблицю завдяки якій можливо оцінювати рівень складності 3GL модулів, також модернізувати таблиці оцінки рівня складності екземплярів об'єктів для коректної оцінки трудовитрат та значення ваг рівнів складності об'єктів у методі об'єктних точок.

В таблиці 2.7 та таблиці 2.8 оцінки рівня складності екземплярів об'єктів замість «екранів» та «звітів», пропонується ввести поняття «проміжний інтерфейс» та «зовнішній інтерфейс».

У onion архітектурі проміжний інтерфейс (Intermediate Interface) є інтерфейсом між шарами програми, який визначає, як взаємодіють внутрішні шари між собою.

Кожен шар має власний інтерфейс, який зв'язується з проміжним інтерфейсом, щоб забезпечити ізоляцію та легку заміну шару без впливу на інші шари. Це дозволяє розробникам змінювати або додавати функціональність у кожному шарі, не торкаючись інших шарів програми.

Проміжний інтерфейс також дозволяє уникнути прямого зв'язування між шарами, що забезпечує кращу підтримуваність і збільшує можливість повторного використання коду.

Зовнішній інтерфейс є частиною архітектури, яка взаємодіє із зовнішнім світом, таким як користувачі, інші додатки, системи тощо.

Зовнішній інтерфейс визначає, як зовнішні користувачі взаємодіятимуть із додатком, які функції будуть доступні і як вони використовуватимуться. Він

також визначає, як дані будуть передаватися між клієнтом та додатком, яким чином оброблятимуться помилки та як виконуватимуться запити та відповіді.

В onion архітектурі зовнішній інтерфейс зазвичай реалізується через адаптери (adapters) або фасади (facade), які забезпечують простий та зручний інтерфейс для зовнішніх клієнтів та приховують складності взаємодії з бізнес-логікою та внутрішніми шарами програми. Це дозволяє спростити взаємодію між додатком та користувачем, а також покращити його масштабованість та гнучкість.

Таблиця 2.7 – Модифікована оцінка рівня складності екземплярів об'єктів в методі об'єктних точок

Для проміжних інтерфейсів				Для зовнішніх інтерфейсів			
Кількість модулів з якими взаємодіє інтерфейс	Джерела у вигляді структур даних			Кількість використуваних секцій	Джерела у вигляді таблиць або структур даних		
	Усього <4	Усього <8	Усього 8+		Усього <4	Усього <8	Усього 8+
<3	Простий	Простий	Середній	0-1	<u>Простий</u>	Простий	Середній
3-7	Простий	Середній	Складний	2-3	Простий	Середній	Складний
>8	Середній	Складний	Складний	4+	Середній	Складний	Складний

Таблиця 2.8 - Модифіковане значення ваг рівнів складності об'єктів у методі об'єктних точок

Тип об'єкту	Рівень складності		
	Простий	Середній	Складний
Проміжний інтерфейс	1	2	3
Зовнішній інтерфейс	2	5	8
3GL-компонент	4	7	10

Також для коректної оцінки трудовитрат я пропоную ввести таблицю (таблиця 2.9) оцінки рівня складності 3GL модулів на основі кількості функцій які вкладені в модуль та кількості класів які реалізують бізнес-логіку цих функцій.

Таблиця 2.9 - Оцінка рівня складності 3GL модулів в методі об'єктних точок

Для 3GL модулів				
Кількість функцій вкладені в модуль	які в	Кількість класів які реалізують бізнес-логіку цих функцій		
		<5	5-15	>15
1		Простий	Середній	Складний
2-5		Середній	Складний	Складний
>5		-	Складний	Складний

Вище описано було, що оцінка 3GL модулів не є коректною тому, що раніше при розробці даного методу не було багатошарових проєктів та кількість функцій була не велика. Якщо брати оригінальний метод оцінювання 3GL модулів, то він не дає точної оцінки.

### **3 РОЗРОБКА МЕТОДИКИ ДЛЯ МОДИФІКОВАНОЇ МОДЕЛІ ОЦІНЮВАННЯ ТРУДОВИТРАТ ДЛЯ ІТ-ПРОЄКТУ ВЕБ- ЗАСТОСУНКУ**

#### **3.1 Методика використання оригінального методу**

Етап 1. Оцінити кількість об'єктів досліджуваного продукту як кількість екранів, звітів та 3GL-компонентів, які будуть входити в розроблюваний додаток (при оцінюванні виходити із стандартного опису цих об'єктів у використовуваному інтегрованому CASE-середовищі).

У методі об'єктних точок об'єкти є основними функціональними одиницями, на які розбивається система. Об'єкти може бути як внутрішніми компонентами системи, і зовнішніми системами, із якими система взаємодіє. Кожен об'єкт має свою функціональну роль та визначає вхідні та вихідні дані, які він обробляє, а також зв'язки з іншими об'єктами. Вони допомагають визначити обсяг роботи, пов'язаний з кожним об'єктом, а також виділити найбільш критичні об'єкти, які потребують додаткової уваги під час планування та управління проектом.

Екран – це інтерфейс забезпечуючий введення та виведення даних. На відміну від звіту, екран забезпечує введення початкових даних для роботи програмних модулів, забезпечує передачу проміжних даних в ході роботи модулю. Обмін проміжних даних модуля з базою даних або навпаки та забезпечує при необхідності видачу проміжних або остаточних даних для інших модулів або інших систем, не користувачу.

Екрани складаються з представлень та структур даних відображених на цьому екрані. Представлення можуть бути як цілісний елемент фрагменту екрану, для розглядання таким елементом є цілісне текстове меню в Adobe Acrobat зображене на рисунку 3.1.

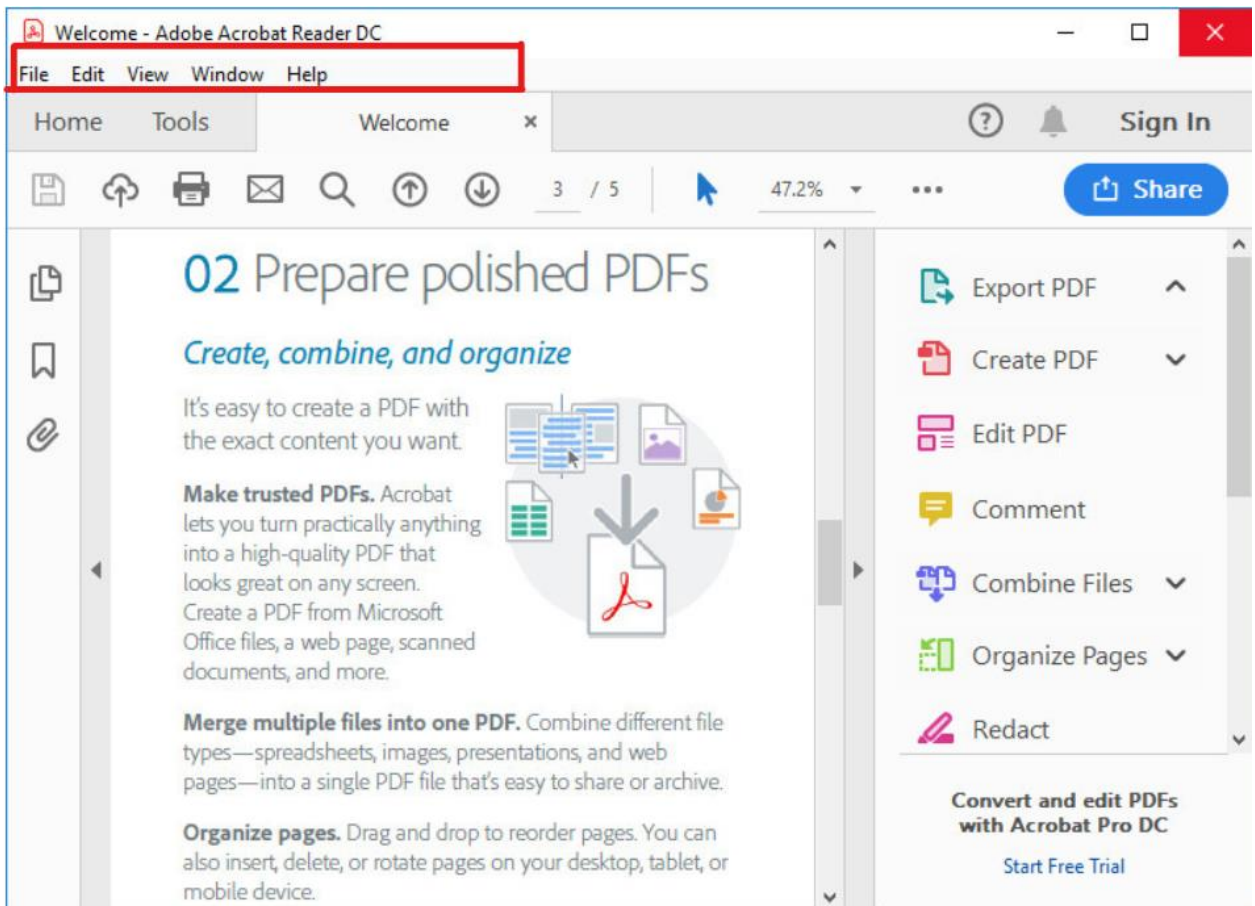


Рисунок 3.1 - Цілісне текстове меню в Adobe Acrobat

Звіти – це об’єкт видачі структур даних в файли для друку або баз даних. Дані об’єкти видають користувачу результати його роботи або вихідні дані, результат виконання функції та рішення функціональної задачі. Звіти розбиваються на секції які визначають внутрішню структуру документу.

Зазвичай як окремий елемент виділяється заголовок який зображений на рисунку 3.2.

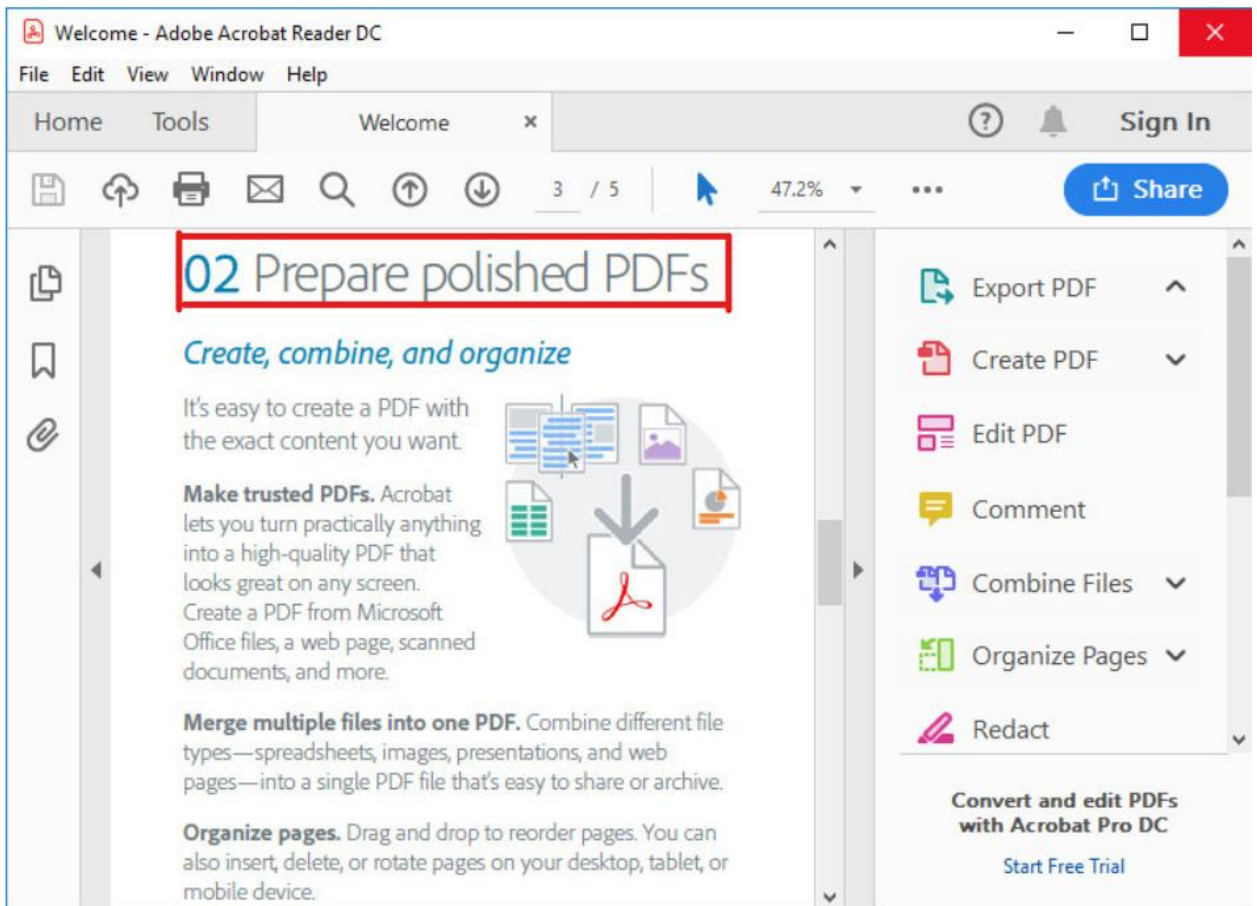


Рисунок 3.2 – Окремий елемент в Adobe Acrobat

3GL компоненти - це програмні модулі реалізуючі бізнес логіку та взаємодію з базами даних.

Етап 2. Класифікувати кожен екземпляр об'єкту за рівнем складності (простий, середній, високий рівні) залежно від значень характерних розмірів з використанням таблиці 2.1.

Для прикладу візьмемо стартовий документ Adobe Acrobat та порахуємо представлення та джерела для оцінки екранів на рисунку 3.3.

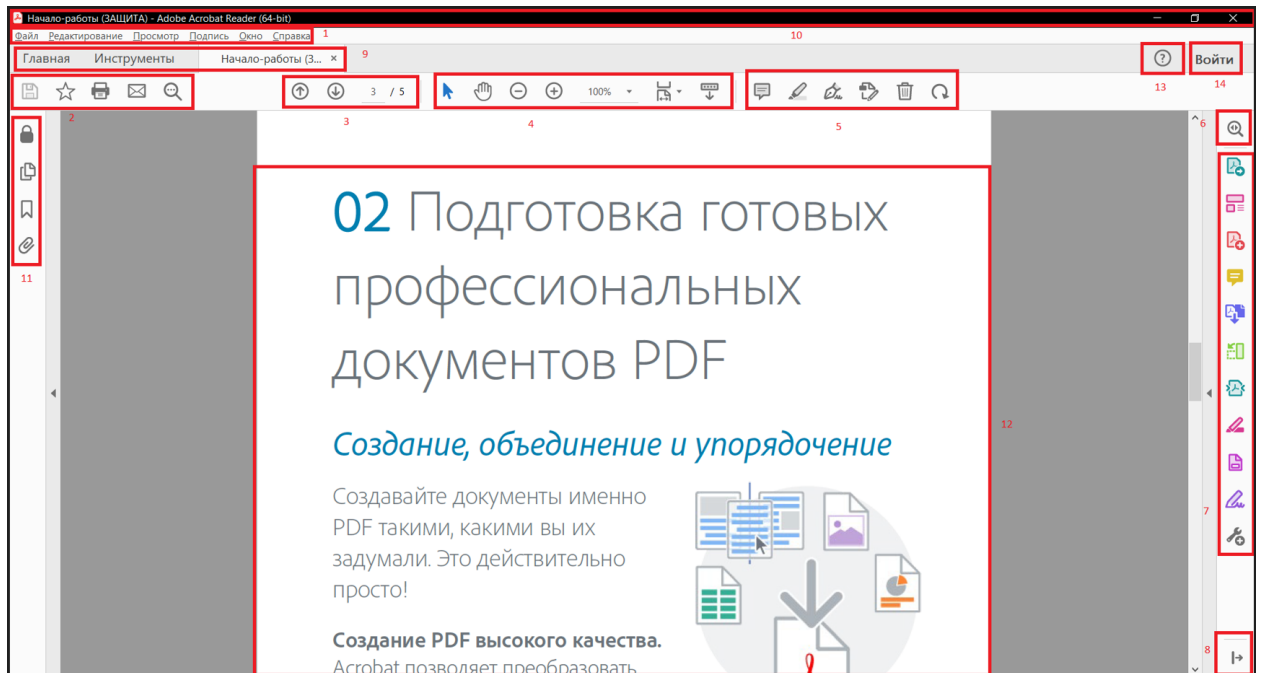


Рисунок 3.3 – Кількість використувуваних представлень

Дивлячись на рисунок ми бачимо 14 представлень та одне джерело тому, що ми працюємо з одним документом. Тому маємо екран середнього рівня складності.

Далі, візьмемо цей же документ, та порахуємо секції та джерела для оцінки звітів на рисунку 3.4.

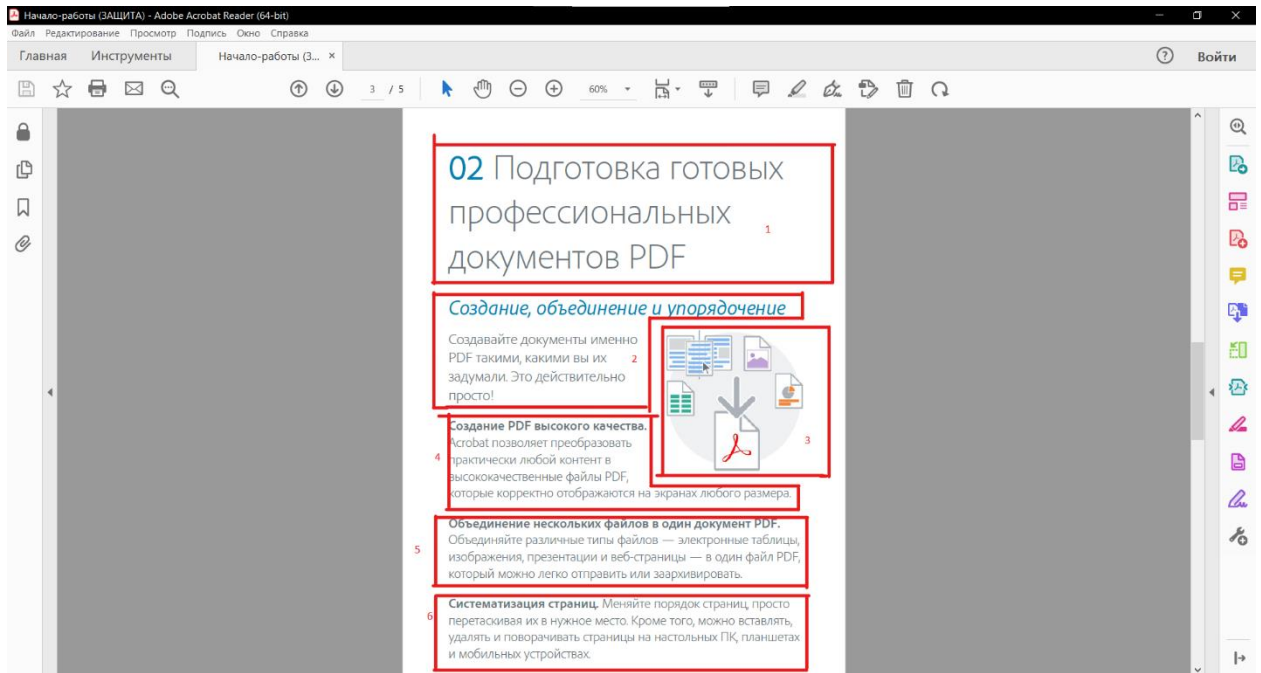


Рисунок 3.4 – Кількість використовуваних секцій

Дивлячись на рисунок ми бачимо 6 використовуваних секцій та одне джерело тому, що працюємо з одним документом. Тому маємо звіт середнього рівня складності.

Етап 3. Зважити результат класифікації згідно з таблицею 2.2. Ваги відображують витрати, які необхідні для реалізації відповідного екземпляру об'єкту вказаного рівня складності.

Згідно цієї таблиці:

- екранна форма = 2;
- звіт = 5;
- 3GL модуль = 10.

Етап 4. Для отримання кількості об'єктних точок ObjectPoints просумувати значення ваг усіх зважених об'єктів.

Етап 5. Оцінити відсоток повторного використання, яке очікується в даному проєкті за наступною формулою (3.5), та розрахувати кількість нових об'єктних точок за формулою (2.3).

Для підрахунку відсотку повторного використання використовується формула (3.5):

$$\%reuse = (\text{used components} / \text{total components}) / 100$$

Рисунок 3.5 – Формула для підрахунку відсотку повторного використання

Де *used components* – кількість використовуваних компонентів, а *total components* – загальна кількість компонентів. Під поняттям «компоненти» мається на увазі екрани, звіти та 3GL модулі.

Етап 6. Визначити коефіцієнт виробничості PROD за таблицею 2.4.

CASE (Computer-Aided Software Engineering) – це набір інструментів та методів розробки програмного забезпечення, які допомагають забезпечити високу якість програм, відсутність помилок та простоту супроводу програмних продуктів[9].

Під CASE розуміється також набір методів та засобів проектування інформаційних систем з інтегрованими автоматизованими засобами, які можна використовувати у процесі розробки програмного забезпечення.

Можливості CASE включають засоби аналізу, проектування та програмування. CASE автоматизує проектування інтерфейсів, документацію та генерацію структурованого коду бажаною мовою програмування.

Існують дві основні концепції комп'ютерного програмного забезпечення CASE:

- прості та «прозорі» методи полегшення розробки та/або супроводу програмного забезпечення;
- інженерний підхід до розробки та/або супроводу програмного забезпечення.

Типові CASE-інструменти:

- засоби управління конфігурацією;
- засоби моделювання даних;
- засоби аналізу та проєктування;
- інструменти перетворення моделі;
- засоби редагування програмного коду;
- засоби рефакторингу коду;
- генератори коду;
- інструменти для створення UML-діаграм.

Етап 7. Розрахувати оцінку кількості людино-місяців, які вимагаються для реалізації IT-проєкту з розробки досліджуваного продукту за формулою 2.5.

### 3.2 Методика використання модифікованого методу

Етап 1. Оцінити кількість об'єктів у кожному шарі досліджуваного продукту як кількість проміжних інтерфейсів, зовнішніх інтерфейсів та 3GL-компонентів, які будуть входити в розроблюваний додаток (при оцінюванні виходити із стандартного опису цих об'єктів у використовуваному інтегрованому CASE-середовищі).

Етап 2. Класифікувати кожен екземпляр об'єкту у всіх шарах окремо за рівнем складності (простий, середній, високий рівні) залежно від значень характерних розмірів з використанням таблиць 2.7 та 2.9.

Етап 3. Зважити результат класифікації за таблицею 2.8. Ваги відображують витрати, які необхідні для реалізації відповідного екземпляру об'єкту вказаного рівня складності.

Етап 4. Для отримання кількості об'єктних точок ObjectPoints просумувати значення ваг усіх зважених об'єктів.

Етап 5. Оцінити відсоток повторного використання компонентів в кожному шарі окремо, яке очікується в даному проєкті за наступною формулою (3.5), та розрахувати кількість нових об'єктних точок за формулою (2.3).

Етап 6. Визначити коефіцієнт виробничості PROD за таблицею 2.4.

Етап 7. Розрахувати оцінку кількості людино-місяців, які вимагаються для реалізації ІТ-проєкту з розробки досліджуваного продукту за формулою 2.5.

Таким чином аналітик зможе отримати максимально точний та вичерпний аналіз по кожному з шарів програми та отримати точну оцінку кількості людино-місяців, які вимагаються для реалізації ІТ-проєкту.

#### **4 АПРОБАЦІЯ ОРИГІНАЛЬНОГО ТА МОДИФІКОВАНОГО МЕТОДУ ОБ'ЄКТНИХ ТОЧОК ПІД ЧАС ОЦІНЮВАННЯ ТРУДОВИТРАТ ІТ-ПРОЄКТУ З РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ**

Продукт, який розробляється в рамках проєкту «Програмна система аналізу та підбору компанії біо-друку органів» є веб-застосунок. Система повинна мати багаторівневу, багат шарову архітектуру і легко масштабуватися.

Мета програмної системи полягає в тому, щоб замовити бажаний орган у компаній, що займаються 3D-біодруком клітин та біоречовин замовника, і зробити ці органи доступними для трансплантації. Система дозволяє зв'язатися з компанією, замовити потрібний орган, обговорити особливості пацієнта та стежити за процесами. Користувач має можливість залишати відгуки про компанії.

Основна мета проєкту — створити систему, яка є сполучною ланкою між покупцем, якому потрібен орган, та компанією, яка може його поставити.

Даний веб-додаток дозволяє подати заявки на друкування органу, взаємодіяти між пацієнтом і працівником компанії для уточнення деталей та встановлення і узгодження дат транспортування біоматеріалів та органу; відстежування процесів від подання біоматеріалу, до отримання надрукованого органу; залишати відгуки про користувачів для того щоб клієнти або друкуючі компанії могли оцінити доброчесність один одного; реєстрація компанії, що друкує органи, та їх представників у розробляємому веб-додатку; отримання інформації про процес друку органу, передача органу або біоматеріалів та окремі завдання користувачів; блокування клієнтів або представників компаній, що не виконали своєї частини угоди або не дотримувалися користувацької угоди.

Також у цьому веб додатку реципієнт матиме можливість написати конкретній компанії та дізнатися про всі цікаві моменти.

Серед переваг розроблюваного ПЗ можна виділити новизну підходу (централізованого) щодо пошуку компаній які займаються біо-друком органів, надання дійсно релевантної інформації щодо точних часових рамок для друку потрібного органу, надання релевантної інформації щодо опису тих процедур які потрібні для підготовки та друку, що дійсно знадобляться при виконанні замовлення.

У даного проєкту є ряд наступних обмежень та виключень які можуть бути описані бізнес правилами:

- для того щоб використовувати веб-додаток користувач має бути зареєстрованим та авторизованим;
- для того щоб мати можливість подати заявку на отримання органу, користувач має бути повнолітнім та мати паспорт;
- якщо біоматеріали пацієнта не можна використати для друку органу, то угоду між клієнтом і компанією буде розірвано.

Обмеження пов'язані з тим щоб компанії мали можливість повертати гроші замовникам у яких біоматеріал не придатний для друку, або якщо користувач не досяг повноліття та не має відповідних документів для підтвердження.

Веб-додаток повинен мати зручний та інтуїтивний інтерфейс та бути легким в управлінні.

Планується, що веб-додаток буде безкоштовний та не буде містити рекламу. Це робиться для того, щоб привернути увагу великої кількості цільової аудиторії.

Сферою використання проєкту є сфера медицини.

На жаль у нашому світі бувають ситуації, коли люди тяжко хворіють або отримують тяжкі поранення та потребують пересадку органів. В наш час далеко не завжди, легко знаходиться донор, який готовий віддати частину або повністю свій орган і потребуючим цього треба довгий час чекати у чергах для отримання органу, який вони так потребують, просто для продовження звичного життя. Проте в наш час, вже існують технології та обладнання, що

можуть друкувати органи на основі клітин потребуючої цього людини, проте на ринку ще не існує системи, яка б могла спростити процес пошуку та дала можливість потребуючим замовити необхідний орган у одної з компаній, яка займається біодруком органів.

Основна цільова аудиторія користувачів – це хворі люди, тяжко поранені військові, компанії біодруку органів які будуть зацікавлені в залученні нових клієнтів та популяризації їхнього бренду, представники приватного бізнесу, державні клініки, тощо.

В подальшому даний веб-застосунок може бути масштабований та представлений в інших країнах по всьому світу.

Проте, однією з головних цілей розробки даного веб-застосунку залишається створення користі потребуючих людей, що не можуть чекати в чергах для того щоб дочекатись потрібний орган.

Результатом виконання проєкту є програмний продукт у вигляді зручного, з інтуїтивним інтерфейсом веб-додатку, що являє собою комплексне рішення, яке має багат шарову архітектуру.

Архітектура буде складатися з чотирьох рівнів: Core Layer, Data Access Layer, Business Layer та Presentation Layer [10].

Core Layer – рівень ядра системи, який буде пов'язаний з усіма іншими рівнями. У цьому рівні будуть розміщуватися моделі, що використовуються у системі, а саме бізнес та моделі бази даних, та інтерфейси, що будуть реалізовуватися в інших рівнях.

Data Access Layer – рівень доступу до даних, основною задачею якого є взаємодія з базою даних MSSQL для отримання, запису оновлення чи видалення необхідних даних [10].

Business Layer – рівень бізнес логіки. Цей рівень містить в собі основну логіку системи, такі як алгоритми, трансформація даних, отриманих з бази, у бізнес моделі, валідація та інше. Для отримання даних рівень бізнес логіки взаємодіє з Data Access Layer та реалізує інтерфейси сервісів с Core Layer.

Presentation Layer – рівень представлення, з яким взаємодіє користувач системи. Цей рівень взаємодіє з рівнями Business Layer та Core Layer.

Дерево цілей - це методологія планування та управління проєктами, яка допомагає розбити основну мету на більш конкретні та вимірні підцілі, домогтися їх реалізації та нарешті досягти головної мети.[11]

Суть дерева цілей полягає у створенні деревоподібної структури, де на вершині розташовується головна мета, а на нижніх рівнях розташовуються більш конкретні підцілі, які необхідно досягти для реалізації верхніх цілей. У цьому кожен рівень цілей пов'язані з іншими рівнями, і утворюється дерево.

Дерево цілей приведено на рисунку 4.1.

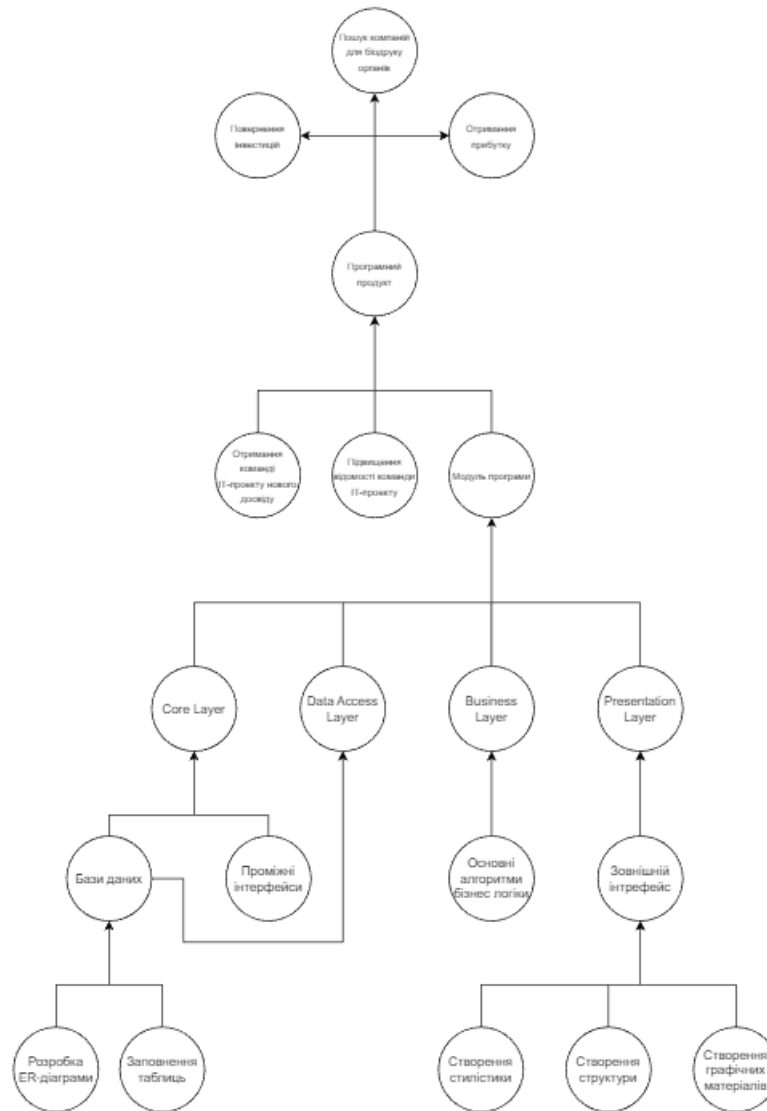


Рисунок 4.1 – Дерево цілей IT-проєкту

Загальна мета проєкту (ціль нульового рівня) – організувати пошук компаній біодруку органів для нужденних, отримати користь від реалізації програми надзвичайних ситуацій та зробити інвестиції зовнішніх інвесторів вигідними.

Розробка розуміється як проєктування протягом 250 робочих днів з вартістю проєкту 120 тис. грн та датою початку проєкту 13 липня 2023 року.

Для досягнення загальної мети необхідно досягти мети першого рівня – розробки веб-застосунку. Більш точне уявлення про досягнення цілей першого

рівня дають цілі іншого рівня. Для досягнення такої мети першого рівня команда ІТ-проєкту повинна отримати новий актуальний досвід розробки проєкту, щоб успішно здійснити розробку даного ІТ-проєкту та підвищити ступінь конвергенції спеціалістів ІТ-проєкту, щоб інвестори повірили в розвиток та інвестували в проєкт, а також необхідно розробити програмний модуль.

Цілі третього рівня дають більш точне уявлення про досягнення цілей іншого рівня. Для досягнення мети іншого рівня, наприклад реалізації програмного модуля, необхідна реалізація усіх шарів архітектури.

Четвертий рівень дерева цілей дає більш детальне уявлення про роботу, яку необхідно виконати для досягнення цілей третього рівня.

Для реалізації серверної частини необхідно створити базу даних, спроектувати архітектуру програмного забезпечення, як компоненти програмного забезпечення взаємодіють один з одним, створити інтерфейси які будуть реалізовуватися на інших рівнях, створити шари бізнес-логіки та доступу до даних, які і будуть фактично визначати наш веб додаток.

Для реалізації клієнтської частини необхідно створити рівень представлення, щоб користувач міг вільно користуватися цим веб-додатком.

Цілі п'ятого рівня дають більш точне уявлення про досягнення цілей. Щоб створити базу даних, потрібно розробити її ER діаграму (рисунок 4.2), записати таблиці та зв'язки в третій нормальній формі, а потім заповнити поля таблиці необхідними значеннями.

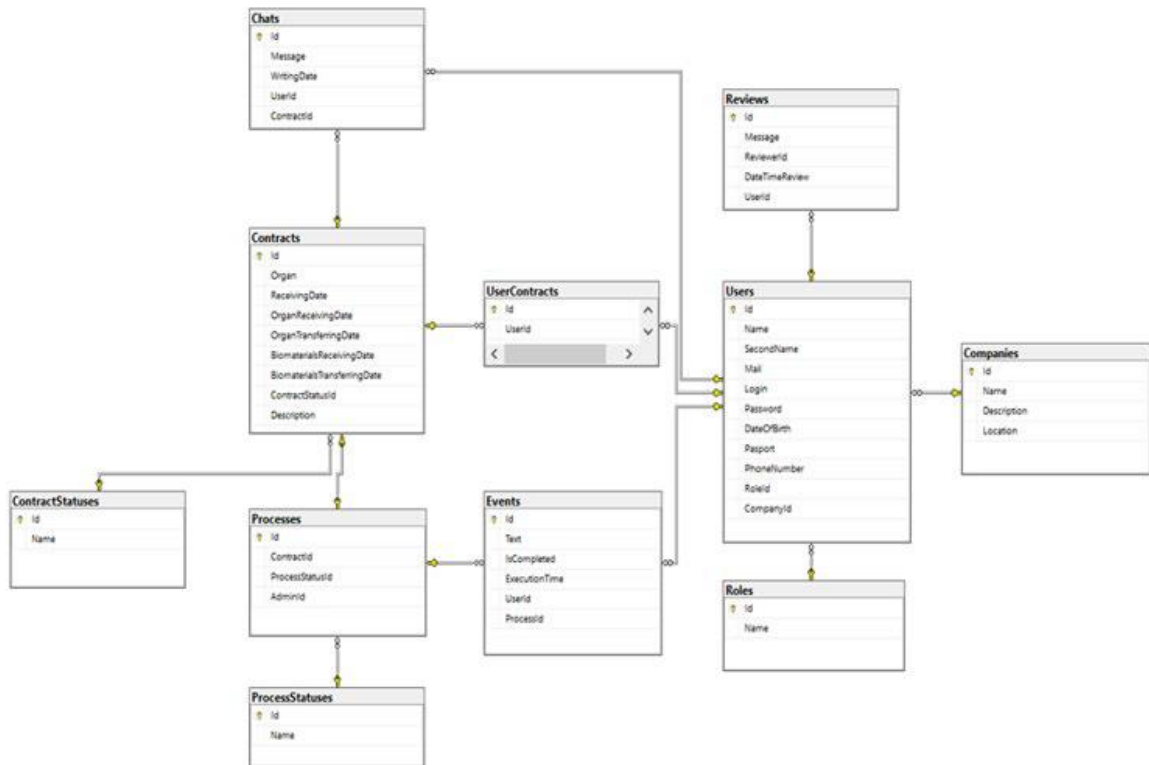


Рисунок 4.2 – ER діаграма

Для створення рівня представлення потрібно створити структури інтерфейсу користувача, прототипувати його, створити стиль інтерфейсу, а потім, на основі раніше написаного, розробити відповідний графічний матеріал.

Дата початку та завершення робіт визначаються згідно графіку виконання робіт:

- дата початку проєкту: 13 липня 2023 року;
- дата завершення робіт: 20 травня 2024 року.

До обмежень можна додати матеріальні, трудові та фінансові ресурси. Таким чином, бюджет проєкту становить 120 000 грн.

Фонд П. Порошенка є джерелом фінансування та інші зацікавлені сторони, інвестори.

Для розробки використовуються ліцензійне ПЗ: операційна система Windows; пакет програм «Office»; середовище розробки Visual Studio 2023,

набір сервісів для тестування; система управління базою даних (СУБД) MSSQL, мови програмування C#, HTML, CSS5.

#### 4.1 Статут проєкту

Статут проєкту (Project charter) – це документ, який описує цілі, очікування, рамки та завдання проєкту, а також призначає керівника проєкту та встановлює його повноваження.[12] Це початковий документ, який формується на початковому етапі проєкту, ще до того, як розпочинається робота по ньому.

Статут проєкту є важливою інструкцією для управління проєктом на всьому його шляху. Він допомагає членам команди та зацікавленим сторонам розуміти, що вони можуть очікувати від проєкту, і що необхідно зробити, щоб досягти поставленої мети.

Статут проєкту наведено в таблиці 4.3.

Таблиця 4.3 – Статут проєкту «Програмна система аналізу та підбору компанії біо-друку органів»

Назва проєкту: «Програмна система аналізу та підбору компанії біо-друку органів»	
Керівник проєкту:	Посада – Проєктний менеджер Потрясов Д.Є.
Куратори проєкту:	Фонд П. Порошенка
Власник проєкту	Команда «Medical Care»
Опис проєкту: «Програмна система аналізу та підбору компанії біо-друку органів» – веб-застосунок, що має багаторівневу, багат шарову архітектуру і легко масштабується.	

## Продовження таблиці 4.3

Програмна система призначена для замовлення потрібного органу у компаній які займаються 3D біопрінтигом з клітин клієнта і біо-речовин, та надають ці органи для трансплантації. Система дозволяє зв'язатися з компанією, замовити потрібний орган, обговорити особливості пацієнта та відстежувати процеси. Користувач має можливість залишати відгуки про компанії.

Мета: створити систему яка буде зв'язуючою ланкою для клієнта потребуючого орган та компанією яка може надати цей орган.

Очікувані результати

Якісні: спрощення процесу пошуку нових органів для потребуючих цього людей.

Кількісні: збільшення кількості медичних закладів які будуть користуватись веб-застосунком.

Допущення та обмеження:

Строки виконання проєкту дорівнюють 238 робочих дня з моменту впровадження уставу проєкту.

У разі нехватки коштів, бюджет проєкту може перевищувати 120 000 грн, для цього можуть бути залучені нові інвестиції до проєкту від нових зацікавлених сторін. У разі виникнення непередбачуваних ситуацій терміни виконання проєкту можуть бути збільшені на 15%.

Ризики:

Можливі ризики при розробці веб-додатку:

- недостатнє фінансування на початковій стадії розробці;
- велика кількість правок на фінальній стадії розробки ІТ-проєкту

Кінець таблиці 4.3

Фаза проєкту	Основний зміст	Дата виконання
Фаза ініціації	1) Дослідження предметної області; 2) Виявлення проблем, актуалізація рішень та постановка задачі; 3) Створення статуту проєкту; 4) Створення вимог користувача (технічне завдання);	13.07.2023 – 08.08.2023
Фаза планування	5) Підписання угоди про планування проєкту; 6) Розробка прототипу проєкту 7) Обговорення тонких місць прототипу проєкту; 8) Створення документів про завершення планування	08.08.2023 – 12.09-2023
Фаза розробки	9) Створення ER-діаграми та заповнення таблиць 10) Створення базових інтерфейсів 11) Створення шару доступу до даних 12) Створення алгоритмів бізнес логіки 13) Створення прототипу інтерфейсу	12.09.2023 – 09.04.2024
Фаза завершення	14) Проведення тестування; 15) Запуск проєкту; 16) Здача проєкту; 17) Підпис договору про закінчення та здачу проєкту.	09.04.2024 – 20.05.2024
Документ підготовано: Потрясов Д.Є. – здобувач вищої освіти групи УПГІТм-21-1		

## 4.2 Приклад використання оригінального методу

У прикладі будуть зроблені розрахунки на основі оригінального методу об'єктних точок. У проєкті будуть відсутні звіти тому, що веб-застосунок не дає нічого користувачу для друку.

Кількість об'єктів дорівнює восьми: 7 екранів та один 3GL модуль.

На рисунках 4.4-4.11 зображені екрани та кількість використовуваних представлень на них.

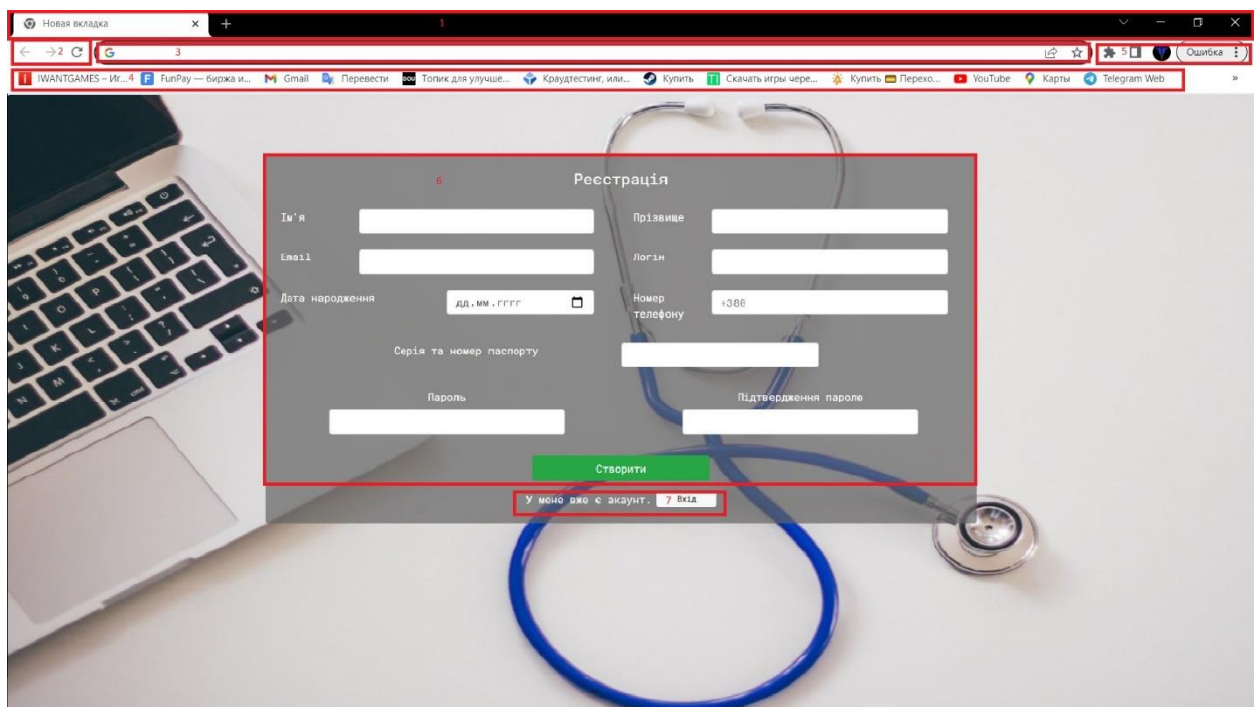


Рисунок 4.4 – Форма реєстрації

У формі реєстрації маємо 7 представлень та одне джерело даних. Форма реєстрації являється екраном та має простий рівень складності.

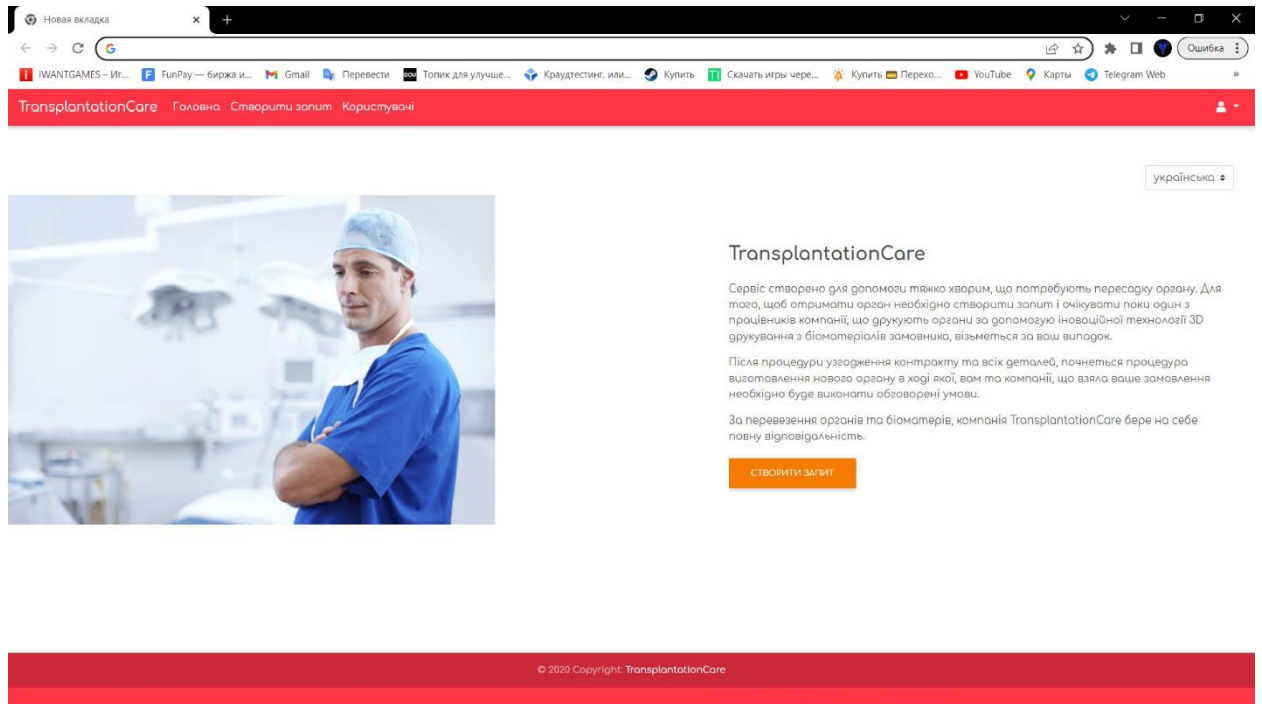


Рисунок 4.5 – Головна сторінка

На головній сторінці 7 представлень та одне джерело даних. Головна сторінка являється екраном та має простий рівень складності.

Новая вкладка x +

← → ↻

iwantgames - Иг... FunPay — биржа и... Gmail Перевести Топик для улучше... Краудестинг, ил... Купить Скачать игры чере... Купить Переко... YouTube Карты Telegram Web

TransplantationCare Головна Створити запит Користувачі

українська

### Запит на отримання органу

Заповніть поля для створення запиту на отримання органу, необхідного для пересадки

Необхідний орган

Дата отримання

Опис

[Противопоказань немає](#)

[СТВОРИТИ](#)

© 2020 Copyright: TransplantationCare

Рисунок 4.6 - Сторінка запиту для отримання органу

На сторінці запиту для отримання органу 8 представлень та одне джерело даних. Сторінка запиту для отримання органу являється екраном та має середній рівень складності.

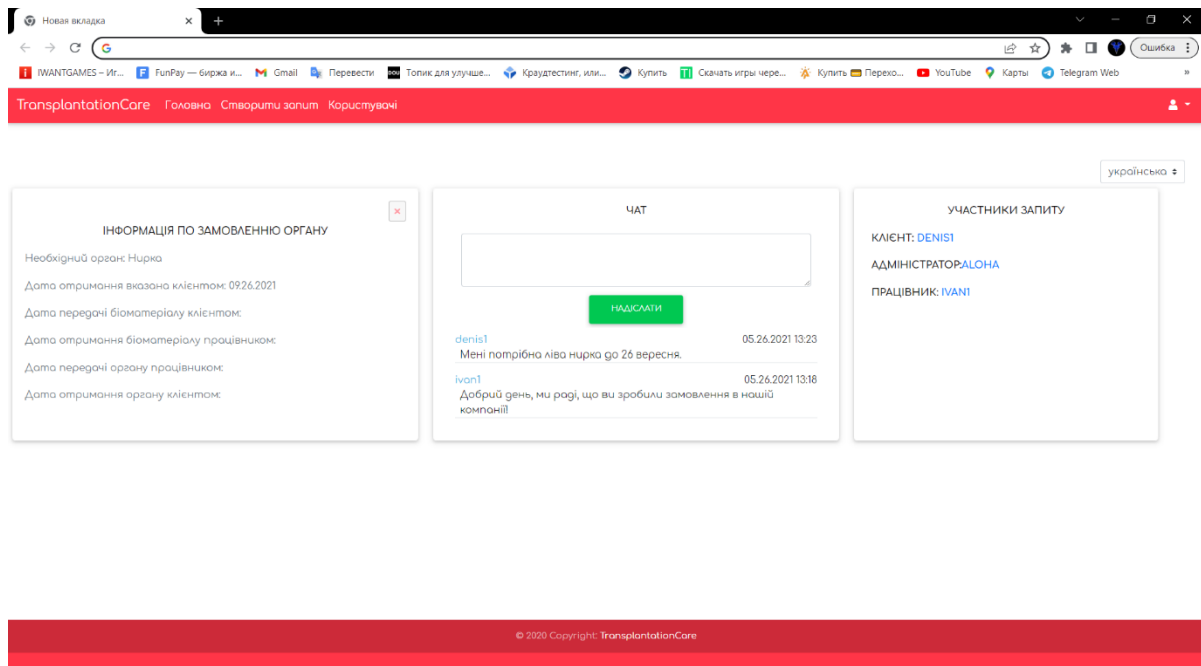


Рисунок 4.7 - Сторінка з інформацією про замовлення та чатом з представником компанії

На сторінці з інформацією про замовлення та чатом з представником компанії 10 представлень та одне джерело даних. Сторінка з інформацією про замовлення та чатом являється екраном та має середній рівень складності.

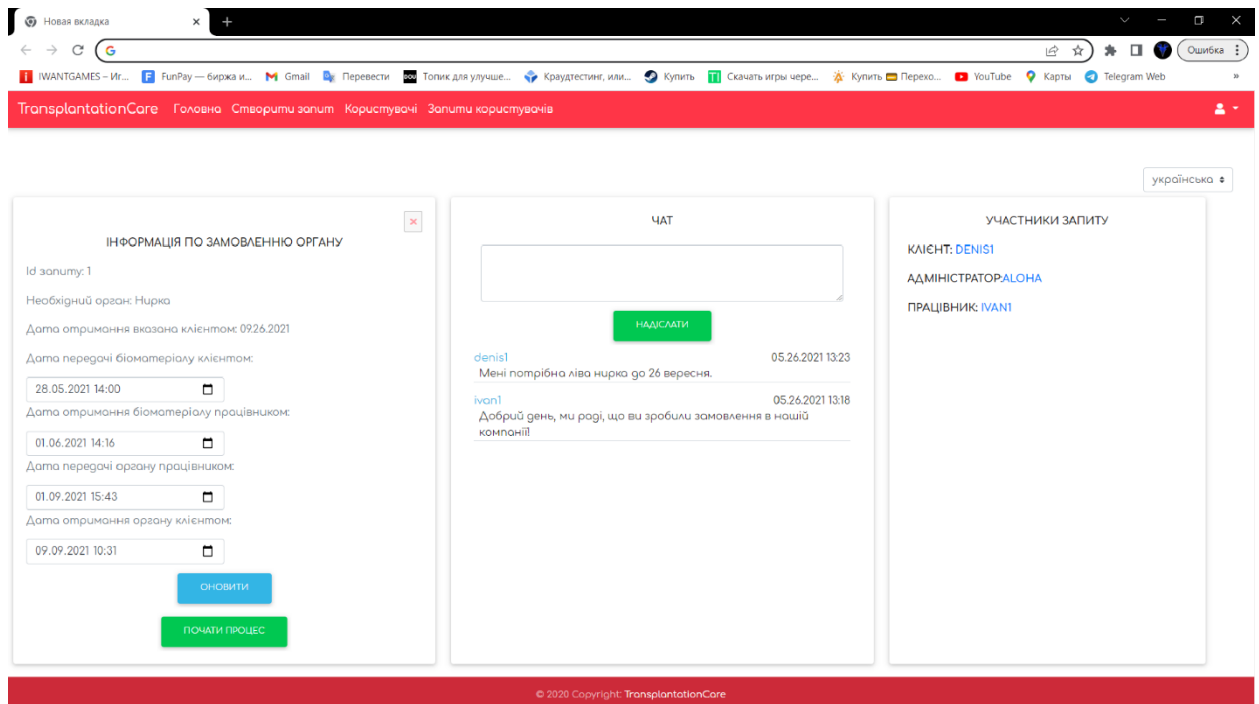


Рисунок 4.8 - Сторінка запиту на друкування органу, що використовується адміністратором

На сторінці запиту на друкування органу, що використовується адміністратором 10 представлень та одне джерело даних. Сторінці запиту на друкування органу адміністратора являється екраном та має середній рівень складності.

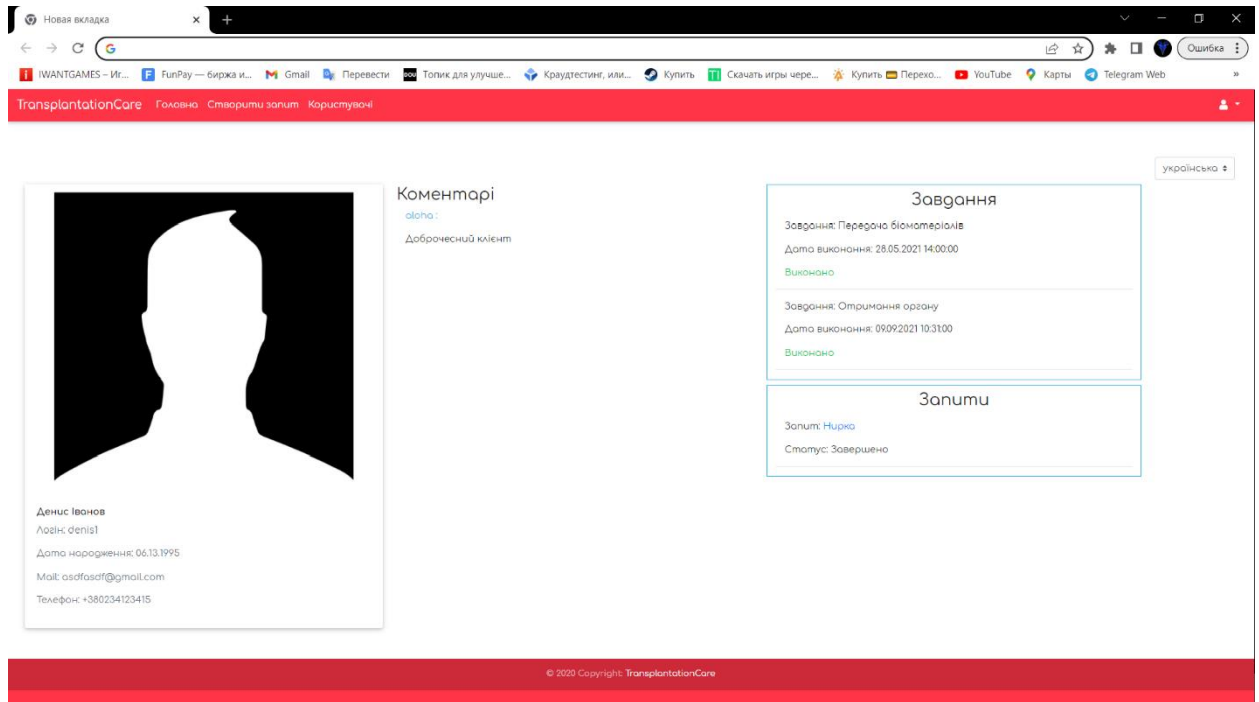


Рисунок 4.9 - Сторінка користувача

На сторінці користувача 10 представлень та одне джерело даних. Сторінка користувача являється екраном та має середній рівень складності.

Маємо 2 простих та 4 середніх екранів та один складний 3GL модуль за таблицею 1. Згідно з таблицею 2, ObjectPoints = 20. Відсоток повторного використання = 0, тому що проєкт новий та не має попередника з якого команда розробників могли би взяти частину компонентів.

$$NOP = 20 \times (100 - 0) / 100$$

Так як проєкт розробляється з масштабом використання по всій країні, то команда розробників має високий досвід та спроможність та спроможності CASE, тому PROD = 25.

За формулою 2, PM = 20/25 = 0.8 людино-місяців які вимагаються для реалізації заданого ІТ-проєкту.

### 4.3 Приклад використання модифікованого методу об'єктних точок

У прикладі будуть зроблені розрахунки на основі модифікованого методу об'єктних точок.

Для кожного класу був створений свій проміжний інтерфейс. Веб-застосунок має 5 інтерфейсів для класів валідації: валідація реєстрації, валідація реєстрації для компанії біодруку, логін валідація, валідація підпису запиту, валідація дати отримання органу. Для кожного класу бізнес логіки був створений окремий інтерфейс: інтерфейс для взаємодії з обліковим записом, інтерфейс для створення запитів, інтерфейс для взаємодії з представниками компаній біо-друку, інтерфейс для взаємодії з запитами, інтерфейс для взаємодії з відгуками та інтерфейс взаємодії з користувачами. Для зручного доступу до даних було реалізована патерн “Repository”, який імплементить інтерфейси репозиторіїв для конкретних таблиць бази даних з Core рівня. Загальна кількість проміжних інтерфейсів дорівнює 22.

Розраховуємо оцінку рівня складності кожного проміжного інтерфейса валідації:

- інтерфейс валідації реєстрації має 5 функціональних модулів з якими взаємодіє та 1 класи, тому має простий рівень складності;
- інтерфейс валідації реєстрації для компанії біодруку має 1 функціональний модуль з яким взаємодіє та 1 клас, тому має простий рівень складності;
- інтерфейс валідації логіну має 1 функціональний модуль з яким взаємодіє та 1 клас, тому має простий рівень складності;
- інтерфейс валідації підпису запиту має 1 функціональний модуль з яким взаємодіє та 1 клас, тому має простий рівень складності;
- інтерфейс валідація дати отримання органу має 1 функціональний модуль з яким взаємодіє та 1 клас, тому має простий рівень складності.

Розраховуємо оцінку рівня складності для кожного проміжного інтерфейсу бізнес-логіки:

– інтерфейс для взаємодії з обліковим записом пов'язаний з двома функціональними модулями та одним класом, тому має простий рівень складності;

– інтерфейс для створення запитів пов'язаний з вісьма функціональними модулями та одним класом, тому має середній рівень складності;

– інтерфейс для взаємодії з представниками компаній біо-друку пов'язаний з двома функціональними модулями та одним класом, тому має простий рівень складності;

– інтерфейс для взаємодії з запитами пов'язаний з сімома функціональними модулями та одним класом, тому має простий рівень складності;

– інтерфейс для взаємодії з відгуками пов'язаний з одним функціональним модулем та одним класом, тому має простий рівень складності;

– інтерфейс для взаємодії з користувачами пов'язаний з трьома функціональними модулями та одним класом, тому має простий рівень складності;

Розраховуємо оцінку рівня складності для кожного проміжного інтерфейсу репозиторіїв:

– репозиторій чату взаємодіє з одним функціональним модулем та двома класами, тому має простий рівень складності;

– репозиторій компаній взаємодіє з двома функціональними модулями та трьома класами, тому має простий рівень складності;

– репозиторій угод взаємодіє з трьома функціональними модулями та двома класами, тому має простий рівень складності;

- репозиторій подій взаємодіє з чотирма функціональними модулями та трьома класами, тому має простий рівень складності;
- репозиторій статусу замовлення взаємодіє з двома функціональними модулями та одним класом, тому має простий рівень складності;
- інтерфейс репозиторію взаємодіє з шістьма функціональними модулями та одним класом, тому має простий рівень складності;
- репозиторій відгуків взаємодіє з одним функціональним модулем та двома класами, тому має простий рівень складності;
- репозиторій ролей взаємодіє з двома функціональними модулями та трьома класами, тому має простий рівень складності;
- репозиторій угод користувачів взаємодіє з п'ятьма функціональними модулями та п'ятьма класами, тому має середній рівень складності;
- репозиторій користувачів взаємодіє з шістьма функціональними модулями та десятьма класами, тому має складний рівень складності;

Розрахуємо оцінку рівня складності для кожного зовнішнього інтерфейсу:

- на рисунку 4.3 маємо дві секції та одне джерело даних, тому форма реєстрації має простий рівень складності;
- на рисунку 4.4 маємо більше чотирьох секцій та одне джерело даних, тому головна сторінка має середній рівень складності;
- на рисунку 4.5 більше чотирьох секцій та одне джерело даних, тому головна сторінка має середній рівень складності;
- на рисунку 4.6 більше чотирьох секцій та три джерела даних, тому сторінка з інформацією про замовлення та чатом з представником компанії має середній рівень складності;
- на рисунку 4.7 більше чотирьох секцій та три джерела даних, тому сторінка запити на друкування органу, що використовується адміністратором має середній рівень складності;

– На рисунку 4.8 більше чотирьох секцій та шість джерел даних, тому сторінка користувача має складний рівень складності.

Кожний функціональний модуль прописані всередині конкретних класів, тому всі 54 функціональні модулі мають простий рівень складності.

Маємо 19 простих, 2 середніх та 1 складний проміжний інтерфейс; 1 простий, 4 середніх та 1 складний зовнішній інтерфейс; 54 простих функціональних модулі.

Згідно з таблицею 5,  $\text{ObjectPoints} = 1 \times 19 + 2 \times 2 + 3 + 2 + 5 \times 4 + 8 + 54 \times 4 = 19 + 4 + 5 + 20 + 8 + 216 = 272$ .

Відсоток повторного використання = 0, тому що проєкт новий та не має попередника з якого команда розробників могли би взяти частину компонентів.

$$\text{NOP} = 272 \times (100 - 0) / 100 = 272.$$

Так як проєкт розробляється з масштабом використання по всій країні, то команда розробників має високий досвід та спроможність та спроможності CASE, тому  $\text{PROD} = 25$ .

За формулою 2,  $\text{PM} = 272 / 25 = 10.88$  людино-місяців які вимагаються для реалізації заданого ІТ-проєкту.

Таким чином, після використання модифікованого методу об'єктних точок, аналітик отримує правильну та ефективну оцінку за доволі невеликий час.

## ВИСНОВОК

В результаті виконання кваліфікаційної роботи було модифіковано метод об'єктних точок, оцінки трудовитрат ІТ-проєкту з розробки веб-застосунку, який є більш ефективний ніж існуючий метод та легкий у використанні.

Перша частина «Аналіз предметної галузі» була присвячена встановленню актуальності досліджень, аналізу існуючих популярних методів та моделей оцінки трудовитрат реалізованих ІТ-проєктів, було визначено та сформульовано мету модифікувати існуючу методику оцінки трудовитрат ІТ-проєктів з розробки веб-додатків на основі onion-архітектури.

У другому розділі «Модифікація методу об'єктних точок за архітектурою» у результаті отриманих досліджень були коротко описані оригінальні принципи методу об'єктних точок, коротко описані принципи модифікованого методу, приведені модифіковані таблиці та поясненням модифікації.

У третьому розділі «Розробка методики для модифікованої моделі оцінювання трудовитрат для ІТ-проєкту веб-застосунку» приведені алгоритми оригінального та модифікованого методу об'єктних точок.

У четвертому розділі «Апробація оригінального та модифікованого методу об'єктних точок під час оцінювання трудовитрат ІТ-проєкту з розробки веб-застосунку» наведено опис проєкту, за допомогою якого було випробувано модифікований та оригінальні методи.

Основною сферою використання є сфера ІТ. Покращений метод надає змогу оцінити трудовитрати на багат шаровий проєкт доволі точно та просто.

Недолік цього методу є його довгий підрахунок та оцінка рівнів складності екземплярів ІТ-проєкту. Цей недолік можна вирішити створивши додаток який міг би автоматизувати процес підрахунку екземплярів та провести їх оцінку.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської кваліфікаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проєктами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 28 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31.
3. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.
4. Palermo, J. (2008). The Onion Architecture: part 1, Programming withPalermo. URL: <https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/> (Дата звернення: 23.03.2023).
5. Якісні методи оцінки ризиків. URL: [https://stud.com.ua/45518/ekonomika/yakisni\\_metodi\\_otsinki\\_rizikiv](https://stud.com.ua/45518/ekonomika/yakisni_metodi_otsinki_rizikiv) (дата звернення: 23.03.2023).
6. Bottom Up Estimating – Reasons To Adopt BottomUp Estimating Technique. URL: <https://www.invoiceowl.com/estimating-guide/bottom-up-estimating/> (дата звернення: 23.03.2023).
7. Object points. URL: <https://www.gristprojectmanagement.us/software-4/object-points.html> (дата звернення: 30.03.2023).
8. Software Engineering | Application Composition Estimation Model (COCOMO II | Stage 1). URL: <https://www.geeksforgeeks.org/software-engineering-application-composition-estimation-model-cocomo-ii-stage-1/> (дата звернення: 01.04.2023).

9. CASE. URL: <https://uk.wikipedia.org/wiki/CASE> (дата звернення: 20.04.2023).

10. Еванс Е. Предметно орієнтоване проєктування (DDD) [Текст] Е. Еванс - М.: «Вільямс», 2011. — С. 448

11. Що таке дерево цілей? URL: <https://pyrogiv.kiev.ua/shho-take-derevo-cilej/> (дата звернення: 20.04.2023).

12. Статут проєкту. URL: <https://oberemokii.com/uk/design-documentation/managing-the-content-and-timing/project-charter> (дата звернення: 27.04.2023).