


ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«МЕТІНВЕСТ ПОЛІТЕХНІКА»

**СТАНДАРТИ ТА МЕТОДОЛОГІЇ  
БІЗНЕС-АНАЛІЗУ:**  
методичні рекомендації  
до виконання індивідуальних завдань

Запоріжжя 2025



УДК 004.4:338.3(072)  
С77

Рекомендовано Науково-методичною радою ТОВ «ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»  
(протокол № 6 від 28.02.2025 р.)

Укладач:  
Москаленко В.В., д.т.н.

**Стандарти та методології бізнес-аналізу** : методичні  
С77 рекомендації до виконання індивідуальних завдань / уклад.  
В. В. Москаленко. Запоріжжя : ТОВ «ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА», 2025. 62 с.

У методичних рекомендаціях наведено умови, поради і методичні  
підходи до виконання індивідуальних завдань з дисципліни «Стандарти  
та методології бізнес-аналізу», вимоги до оформлення, подання та  
оцінювання результатів виконання індивідуальних завдань.

Рекомендовано для студентів спеціальності 122 - Комп'ютерні  
науки першого (бакалаврського) рівня освіти.

УДК 004.4:338.3(072)

©ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ МЕТІНВЕСТ ПОЛІТЕХНІКА»,  
2025



## Зміст

ВСТУП.....	4
1. ПОРЯДОК ВИКОНАННЯ, ПОДАННЯ НА ОЦІНЮВАННЯ ТА ЗАХИСТ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ .....	7
1.1 Перелік індивідуальних завдань з дисципліни «Стандарти та методології бізнес-аналізу» .....	7
1.2 Методичні рекомендації до виконання ІДЗ № 1 .....	7
1.2.1 Завдання на виконання ІДЗ № 1 та формування бізнес-проблеми .....	7
1.2.2 Розробка інтелектуальної карти .....	9
1.2.3 Розробка Use Case diagram .....	16
1.2.4 Розробка Activity diagram .....	23
1.2.5 Діаграма послідовності .....	29
1.2.6 Діаграма станів .....	39
1.2.7 Зміст звіту з виконання ІДЗ № 1 .....	42
1.3 Методичні рекомендації до виконання ІДЗ № 2 .....	43
1.3.1 Завдання до ІДЗ № 2 та етапи виконання.....	43
1.3.2 Формування User Story як високорівневі вимоги до ПЗ... ..	44
1.3.3 Критерії прийняття User Story .....	46
1.3.4 Story Mapping як представлення загальної карти вимог до ПЗ.....	47
1.3.5 Зміст звіту з виконання ІДЗ №2 .....	54
1.4 Оформлення, подання і захист індивідуального завдання.....	54
1.5 Критерії оцінювання результатів виконання ІДЗ .....	55
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ .....	57
Додаток А. Приклад оформлення титульного аркушу.....	60



## ВСТУП

Індивідуальне завдання – це форма організації навчання, яка має на меті закріплення, узагальнення та поглиблення знань, які студенти отримують у процесі вивчення дисципліни «Стандарти та методології бізнес-аналізу», а також застосування цих знань для розв'язання практичних задач бізнес-аналізу у сфері ІТ.


Методичні рекомендації до виконання індивідуальних завдань укладено на підставі Стандарту вищої освіти за спеціальністю 122 «Комп'ютерні науки» у галузі знань 12 «Інформаційні технології» для першого (бакалаврського) рівня вищої освіти (затверджено Наказом Міністерства освіти і науки України від 28.04.2022 р. № 393).

Індивідуальне домашнє завдання (ІДЗ) – це змістовно завершена робота теоретичного-практичного характеру, що виконується на основі знань, умінь і навичок, набутих у процесі лекційних і практичних занять з дисципліни «Стандарти та методології бізнес-аналізу», яка охоплює декілька тем одного змістового модулю відповідно до робочої програми дисципліни.

Виконання ІДЗ передбачає позааудиторну самостійну роботу студента. Метою такої роботи є поглиблення теоретичних знань й отримання практичних навичок, які відповідають програмним результатам навчання з освітньої компоненти «Стандарти та методології бізнес-аналізу» освітньої програми «Комп'ютерні науки» бакалаврського рівня освіти.

Дисципліна «Стандарти та методології бізнес-аналізу» орієнтована на підготовку студентів (першого рівня вищої освіти) з бізнес-аналізу у сфері ІТ. Метою дисципліни є забезпечення студентів ґрунтовними знаннями щодо стандартів та методологій бізнес-аналізу, а також сприяння отримання практичних навичок у студентів щодо використання технологій бізнес-аналізу для виявлення та управління вимогами до програмного забезпечення (ПЗ) для систем цифрового інтелекту (СЦІ), які використовуються у сферах професійної діяльності різних фахівців підприємств [1].


Важливим аспектом дисципліни є розгляд проведення бізнес-аналізу за гнучкими методологіями (Agile). Студенти отримують знань щодо управління вимогами у процесі проєктування та розробки ПЗ для СЦІ в умовах постійної адаптації ПЗ до змінних вимог замовника. Основні завдання дисципліни включають формування знань у студентів щодо процесів обстеження і співробітництва з зацікавленими сторонами проєкту з розробки СЦІ, знань щодо управління життєвим циклом вимог, визначення дизайну майбутнього програмного рішення, оцінювання його бізнес-цінності. Дисципліна інтегрує теоретичні знання та практичний досвід для розв'язання конкретних задач бізнес-аналітика. Знання стандартів, методів бізнес-аналізу й отримання практичних



навичок їх використання для розробки ПЗ відповідно тем дипломних робіт дозволить студентам використовувати це надалі у своїй професійній діяльності у сфері ІТ [2, 3, 4]. Викладання дисципліни побудовано на основі новітньої інформації щодо стандартів та методологій бізнес-аналізу у проєктах з диджиталізації кращих українських і світових ІТ-компаній.

Освітня компонента (ОК) «Стандарти та методології бізнес-аналізу» спрямована на отримання здобувачами наступних загальних та спеціальних (фахових) компетентностей (згідно ОПП «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти) [1]:

- ЗК1 – здатність до абстрактного мислення, аналізу та синтезу;
- ЗК2 – здатність застосовувати знання у практичних ситуаціях;
- ЗК3 – знання та розуміння предметної області та розуміння професійної діяльності;
- ЗК6 – здатність вчитися й оволодівати сучасними знаннями;
- ЗК7 – здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- ЗК8 – здатність генерувати нові ідеї (креативність);
- ЗК10 – здатність бути критичним і самокритичним;
- ЗК11 – здатність приймати обґрунтовані рішення;
- ЗК12 – здатність оцінювати та забезпечувати якість виконуваних робіт;
- ЗК16 – здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь-яких інших проявів недобросовісності;
- СК6 – здатність до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризики;
- СК7 – здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів;
- СК11 – здатність до інтелектуального аналізу даних на основі методів обчислювального інтелекту включно з великими та погано структурованими даними, їхньої оперативної обробки та візуалізації результатів аналізу в процесі розв'язування прикладних задач;
- СК15 – здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування



функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування.

Програмні результати ОК за ОПП «Комп'ютерні науки» [1]:

- ПР11 – володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт);

- ПР14 – вміти застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

Інші програмні результати:

- вміти використовувати техніки бізнес-аналізу для виявлення вимог щодо проектування та розробки програмних систем у певних галузях економіки;

- застосовувати сучасні знання стандартів бізнес-аналізу для управління вимогами до програмного забезпечення згідно Agile;

- мати знання з методологій бізнес-аналізу для управління вимогами до систем цифрового інтелекту;

- мати спеціалізовані уміння/навички щодо документування результатів бізнес-аналізу, враховуючи методології розробки програмного забезпечення.

Виконання ІДЗ передбачає інтеграцію навчальної, практичної, комунікативної та інших видів діяльності здобувачів освіти з використанням матеріалів реальних проектів у сфері ІТ підприємств, зокрема активів Групи МЕТІНВЕСТ [2, 3, 4].



## 1. ПОРЯДОК ВИКОНАННЯ, ПОДАННЯ НА ОЦІНЮВАННЯ ТА ЗАХИСТ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

### 1.1 Перелік індивідуальних завдань з дисципліни «Стандарти та методології бізнес-аналізу»

Згідно робочої програми дисципліни передбачено виконання двох індивідуальних завдань для студентів. Опис ІДЗ надано у табл.1.

Таблиця 1 – Опис індивідуальних завдань

№ з/п	Опис індивідуального завдання
1	Методи бізнес-аналізу Mind Map, Use Case diagram, Activity diagram, Sequence diagram як інструменти формування вимог до ПЗ
2	Формування та аналіз вимог за Agile методологією SCRUM: розробка User Stories та створення Story Mapping

### 1.2 Методичні рекомендації до виконання ІДЗ № 1


#### 1.2.1 Завдання на виконання ІДЗ № 1 та формування бізнес-проблеми

Мета індивідуального завдання – це засвоєння технік бізнес-аналізу для виявлення вимог до програмного забезпечення, яке розроблюється в межах дипломної роботи.

Завдання для ІДЗ №1:

- 1) визначити проблему, яку буде вирішено у межах виконання дипломної роботи за допомогою ПЗ, яке необхідно розробити;
- 2) здійснити етап виявлення вимог до ПЗ, використовуючи методи бізнес-аналізу та інструменти моделювання, а саме:
  - сформулювати бізнес-вимоги до ПЗ, описати їх та представити у вигляді Mind Map;
  - сформулювати функціональні вимоги та представити їх у вигляді діаграми варіантів використання нотації UML;
  - побудувати діаграми діяльності, послідовності та кінцевих автоматів нотації UML, які описують сценарії, зміни станів, дії для реалізації відповідних варіантів використання ПЗ;
- 3) надати пояснення побудованих діаграм та Mind Map;
- 4) зробити висновок щодо отриманих результатів.

Починається виконання ІДЗ №1 з наступного. Здобувач обирає разом з керівником дипломної роботи (ДР) тему, за якою буде розроблено ПЗ, для якого буде виконуватися ІДЗ.



Допускається розпочати виконання завдання без затвердження теми ДР, але повинна бути попередньо узгоджена мета та загальні задачі ДР з її керівником.

Перед виконанням ІДЗ здобувач освіти надає коротку характеристику проблеми, на вирішення якої націлена розробка ПЗ. Наприклад:

*«У підрозділі Метінвест-СМЦ hr-процеси налагоджені з точки зору менеджменту та проводиться їх постійна оптимізація. Керівництво підрозділу постійно аналізує ефективність кожного бізнес-процесу. На основі такого аналізу було визначено необхідність підвищення ефективності бізнес-процесу «Управління персоналом та соціальних питань Метінвест-СМЦ» за рахунок автоматизації певних процедур. Для вирішення цієї проблеми були сформована задача створення дашборду для систематизації великих масивів кадрових даних та їх візуалізації для аналітики та для презентацій.*

Дашборд – це візуальний інструмент, який включає та відображає певні дані в одному просторі. Зібрана інформація представляється у вигляді діаграм, графіків, таблиць та інших візуальних елементів. Основна мета таких інформаційних панелей – це допомогти користувачу ефективно працювати з великими обсягами даних, створювати документи, звіти та ін. для досягнення бізнес-цілей та виконання різних завдань.


На основі аналізу проблеми було визначено такі завдання, які потребують використання дашборду:

- працівники компанії зможуть швидко розробити електронні візитки для різних цілей, використовуючи різноманітні шаблони;

- працівники компанії зможуть проводити пошук потрібної інформації за ключовими словами у новинах компанії, які накопичилися за кілька років, та мати можливість переглядати їх у зручному вигляді;

- для підвищення кваліфікації працівників компанії пропонується широкий спектр тренінгів, вебінарів, майстер-класів тощо, тому у відповідальних за такі заходи є потреба у фіксуванні такої інформації: які тренінги та які співробітники пройшли їх, на які зареєструвалися та ін. Також ці відповідальні співробітники повинні постійно оновлювати навчальні матеріали у тренінгах, призначати новим працівникам та працівникам, які потребують підвищення кваліфікації, проходження навчання за певними заходами; проводити реєстрацію їх на тренінги тощо;

- співробітники компанії, які відповідають за процес онбордингу, зможуть перевіряти, чи засвоюють нові співробітники той матеріал, який надається їм, оцінювати процес адаптації та навчання. Онбординг – це процес формальних і неформальних процедур щодо прийняття нового співробітника, процес починається



*з моменту підписання договору або контракту з ним, може тривати до 1 року в компанії, у залежності від корпоративної культури та чинного законодавства.»*

Після формування загальної проблеми та визначення основного призначення ПЗ, яке буде розроблено у межах виконання дипломної роботи, проводиться процес виявлення, аналізу та управління вимогами до ПЗ. Для цього використовуються різні методи та інструменти бізнес-аналізу.

### 1.2.2 Розробка інтелектуальної карти

Інтелектуальна карта (Mind Map) – це основна техніка бізнес-аналізу щодо виявлення вимог користувача до ПЗ [5]. Mind Map передбачає візуальне представлення думок, ідей і проблем, пов'язаних із бізнесом, а також надає розуміння того, що потрібно зробити .

Інтелектуальні карти зазвичай використовуються бізнес-аналітиками для визначення ідей, під час аналізу першопричин, а також із зацікавленими сторонами для визначення вимог, є корисними інструментами для мозкового штурму. Проста візуалізація зв'язків між ідеями та темами. Вони також можуть допомогти бізнес-аналітикам визначити проблеми та рішення. Інтелектуальні карти є життєво важливим інструментом для будь-якого бізнес-аналітика [6].

*Наприклад: для системи безперервного удосконалення hr-сервісів згідно наведеного опису проблеми у п.1.2.1 створено Mind Map, яка наведена на рис.1.*

Створення Mind Map розглянемо по її структурі, яка включає три основні елементи [7].

1. Центральна ідея, яка визначається першою, вона вказує користувачам на основну проблему чи проект, що обговорюється. *Наприклад, підрозділ Метінвест-СМЦ стикається з проблемою оптимізації окремих робочих функцій основного бізнес-процесу. Цю проблему будемо вирішувати на основі автоматизації цих функцій, тобто будемо розроблювати програми-асистенти для підтримки цих функцій. Отже, центральною ідеєю під час створення інтелектуальної карти буде «Розробка програм-асистентів для hr-підрозділу».*

2. Фактори, що впливають на центральну ідею. Це можуть бути нові ідеї, які генеруються шляхом мозкового штурму за основними питаннями вирішення центральної проблеми, або основні сфери, фактори, які впливають на проблему. Оскільки у прикладі розглядаємо центральну ідею розробки ПЗ, тому запропоновано обрати різновид інтелектуальної карти – Mind Map 5W (рис.2.), де у якості факторів використовуються п'ять запитань: Who (Хто), What (Що), Where (Де), When (Коли), Why (Чому), а також додаткове запитання - How (Як).

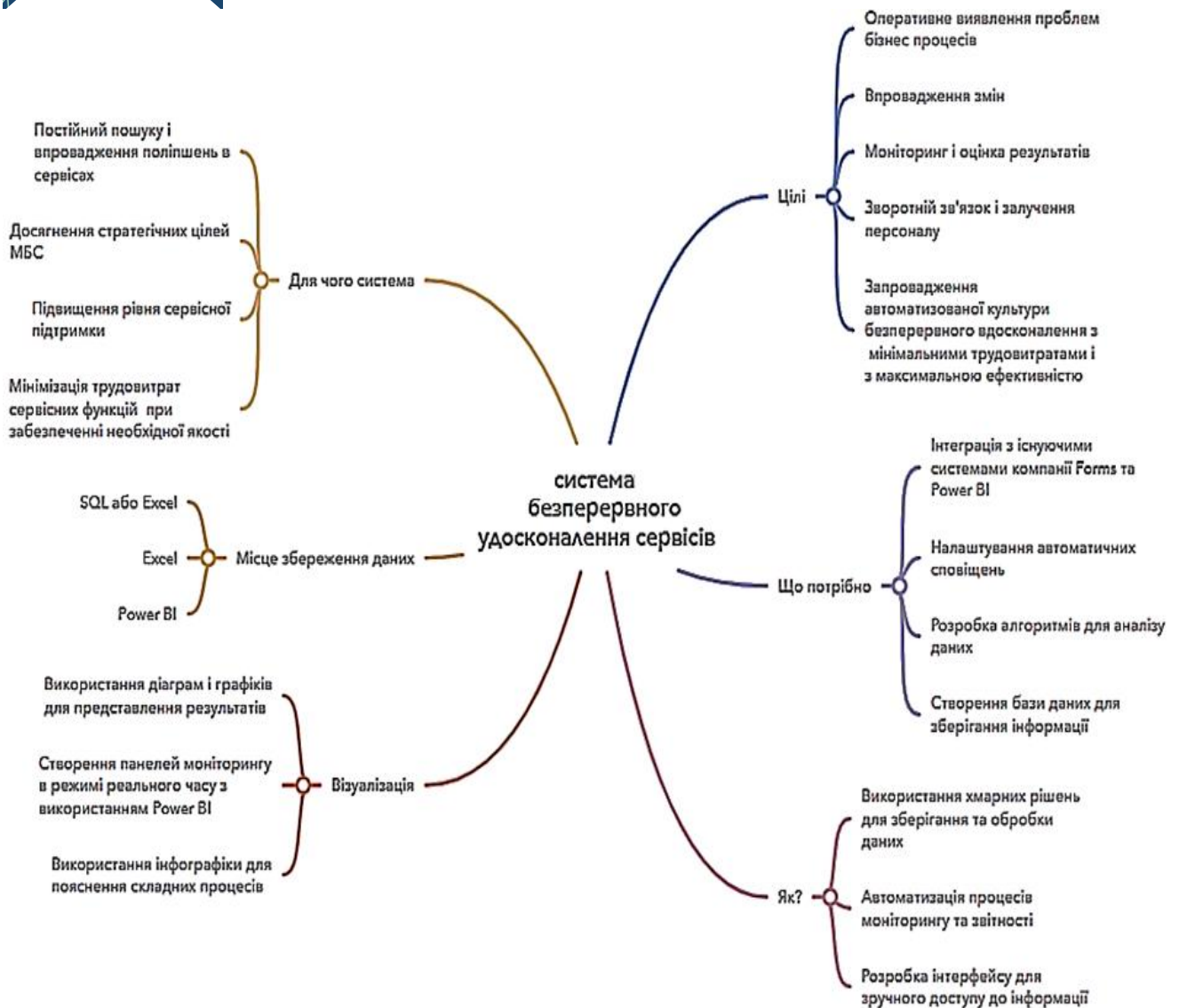


Рисунок 1 – Приклад Mind Map

На рис.2 представлено фрагмент Mind Map з визначеною центральною ідеєю для проекту розробки програм-асистентів в межах системи удосконалення hr-сервісів та питання, відповіді на які розкривають основні аспекти проекту. Mind Map створено за допомогою інструмента Coggle (<https://coggle.it/>).

3. Вивчення зон впливу факторів. Коли наведені фактори, які впливають (або визначають) основну ідею, то треба більш детально описати складові цих факторів, які зазвичай формуються узагальнено, описують різні аспекти реалізації центральної ідеї.

Наприклад, якщо брати формат інтелектуальної карти 5W, то у цьому блоці треба надати відповіді на запитання:

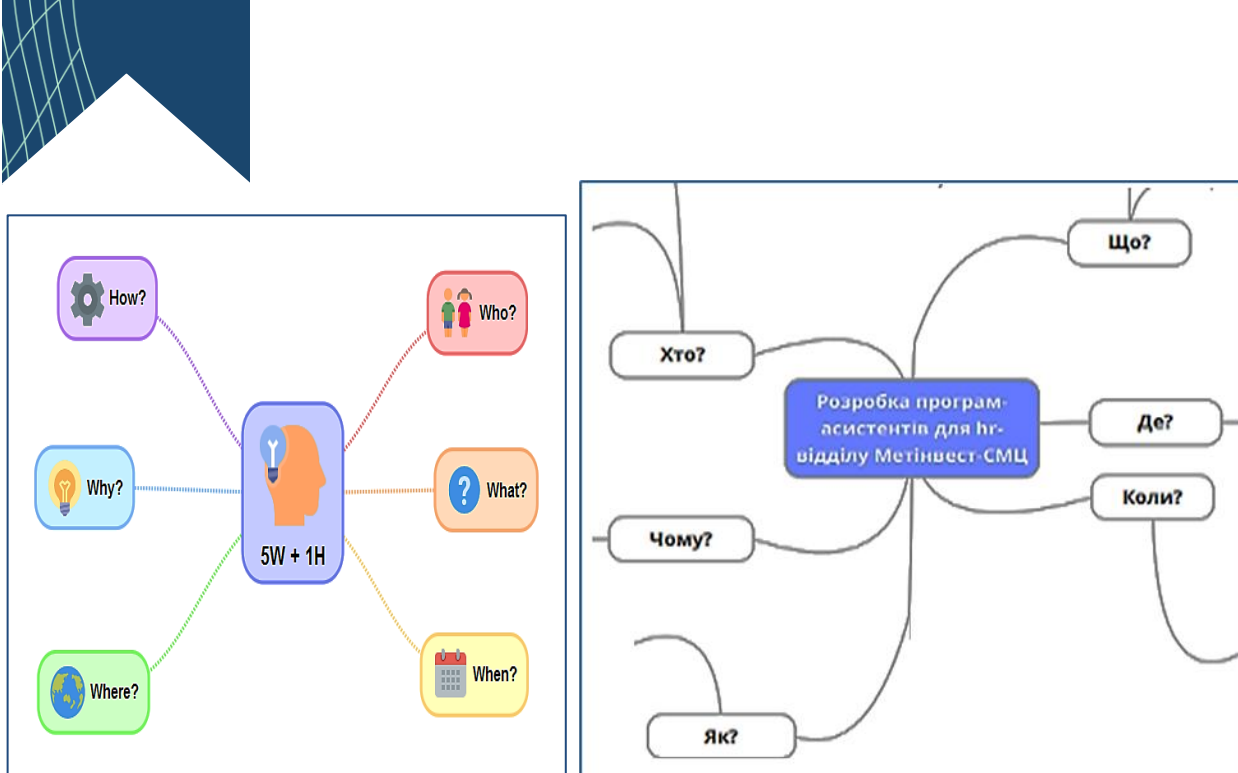


Рисунок 2 – Приклад центральної ідеї та факторів Mind map

- 1) «Хто буде приймати участь у цьому проєкті?». У якості відповіді - це основні стейкхолдери проєкту: замовники, виконавці, спонсори, партнери та ін.);
- 2) «Що буде реалізовано в межах проєкту?». Зазвичай розглядаємо проєкт щодо розробки ПЗ, тому у цій гільці надають опис основних бізнес-вимог до ПЗ, описують бізнес-правила тощо.
- 3) «Де буде реалізована проєкт?». Великі проєкти можуть буди розбиті на якісь частки, тому реалізація проєкту може здійснюватися у різних офісах компанії або різними компаніями, тому у цій гільці вказують, де расташовані виконавці проєкту.
- 4) «Чому реалізується цей проєкт?». Тут описують основні проблеми, які повинен вирішити цей проєкт. Якщо розглядається проєкт щодо розробки ПЗ, то тут описують основні потреби замовника, бізнес-кейс проєкту;
- 5) «Коли буде реалізовано цей проєкт? У цій гільці надаються основні етапи або фази проєкту як перше наближення структури та плану проєкту;
- 6) Як буде реалізовано проєкт? Зазвичай вказують, яку методологію управління проєктом та/або методологію розробки ПЗ запропоновано обрати, які платформи або інструменти будуть використані при розробці ПЗ тощо.

*Приклад формування відповідей на запитання Mind Map 5W для проєкту щодо розробки програм-асистентів як складових системи безперервного удосконалення hr-сервісів надано на рис.3 та рис.4.*

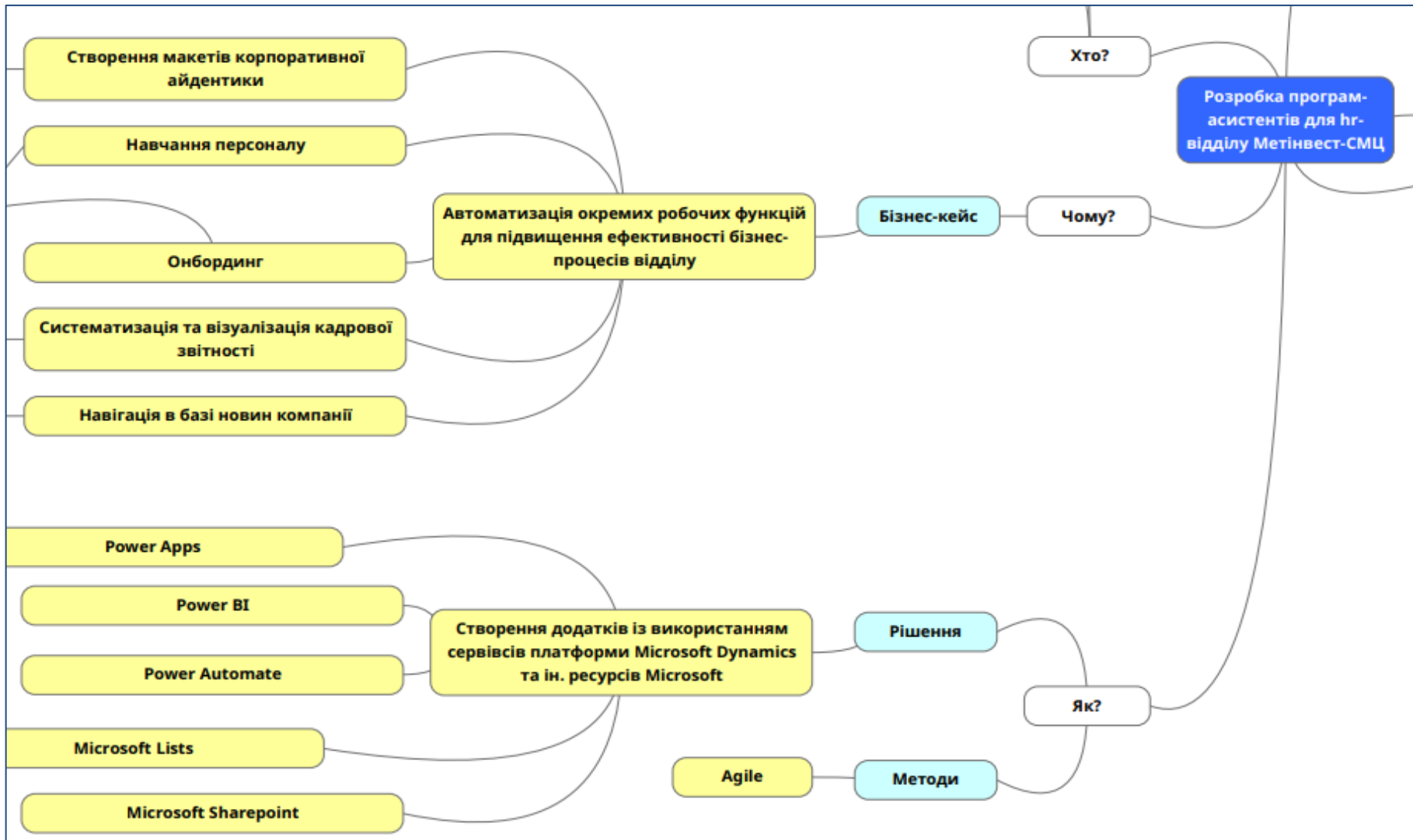


Рисунок 3 – Приклад формування відповідей на питання «Чому» та «Як» у Mind Map 5W для проєкту щодо розробки програм-асистентів (у межах системи безперервного удосконалення hr-сервісів)

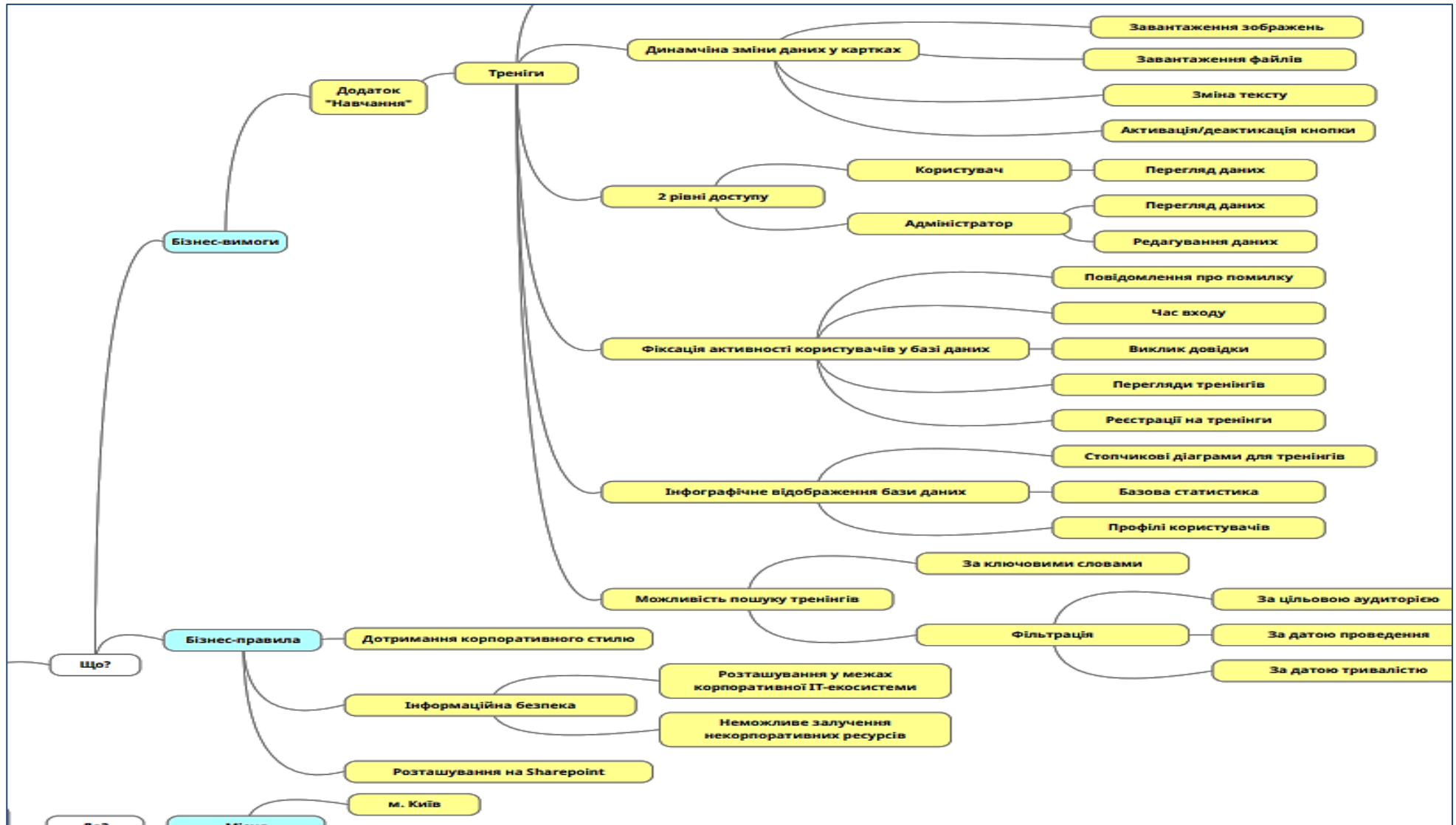


Рисунок 4 – Приклад формування відповідей на питання «Що» у Mind Map 5W

Формування такої карти може бути здійснено під час серій мозкових штурмів зі стейкхолдерами проєкту, у результаті яких буде згенеровані нові ідеї щодо скоупу проєкту, його призначення та засобів реалізації тощо [8]. Це заохочує творче мислення усіх зацікавлених осіб у реалізації проєкту, що допомагає знаходити рішення бізнес-проблем та відповідей на різні питання.

Інтелектуальна карта буде зростати та змінюватися протягом проєкту, і її можна використовувати у кожній із областей знань бізнес-аналізу.

Існує безкоштовне програмне забезпечення, яке можна використовувати як інструмент для створення інтелектуальних карт. До таких інструментів можна віднести Visual Paradigm [9, 10], Lucidchart [https://www.lucidchart.com/pages/templates/mind-map], Canva [https://www.canva.com/graphs/mind-maps/], Coggle [https://coggle.it/], draw.io [https://drawio-app.com/mind-maps/] та ін.

Розглянемо основні завдання бізнес-аналізу, де можна використати Mind Map для їх ефективного виконання [11].

1. Планування та моніторинг бізнес-аналізу: на початку проєкту створюється інтелектуальна карта для організації та координації дій стейкхолдерів, надалі її можна використати для координування та зміни планів бізнес-аналізу в залежності від накопичених знань щодо вимог до ПЗ.

На рис.5 наведено приклад використання Mind Map для опису зустрічі зі стейкхолдерами проєкту.

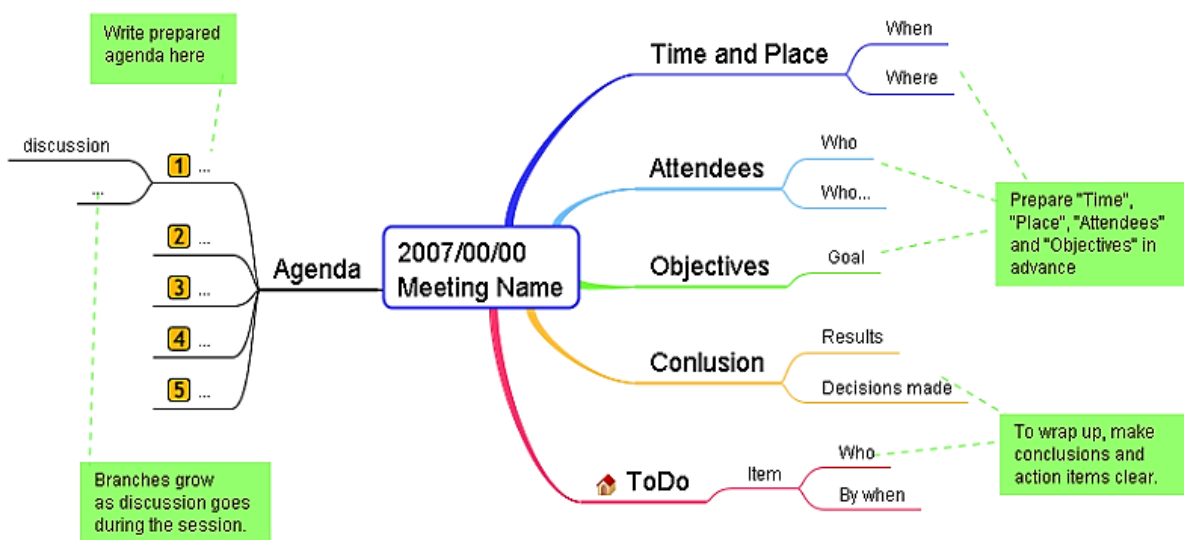


Рисунок 5 – Шаблон інтелектуальної карти організації зустрічі [12]

2. Аналіз стратегії бізнес-аналізу: при використанні Mind Map 5W для визначення стратегії бізнес-аналізу треба визначити основних зацікавлених сторін і партнерів проєкту у гільці Хто (рис. 5), а потім записати бізнес-кейс високого рівня у розділі Чому.

3. Виявлення та співпраця: коли бізнес-вимоги та бізнес-правила високого рівня зібрані, то їх додають до гілки Що у Mind Map 5W (рис.2).

4. Аналіз вимог і визначення дизайну: на етапі перевірки вимог треба визначити ключові аспекти рішень і методи доставки продукту у гільці «Як» та Коли. Інтелектуальну карту можна використовувати для візуалізації вимог.

На рис. 7 приведено приклад візуального представлення вимоги щодо формування облікового запису користувача ПЗ. Інтелектуальні карти можуть бути потужним інструментом для ієрархічного представлення бізнес-вимог.

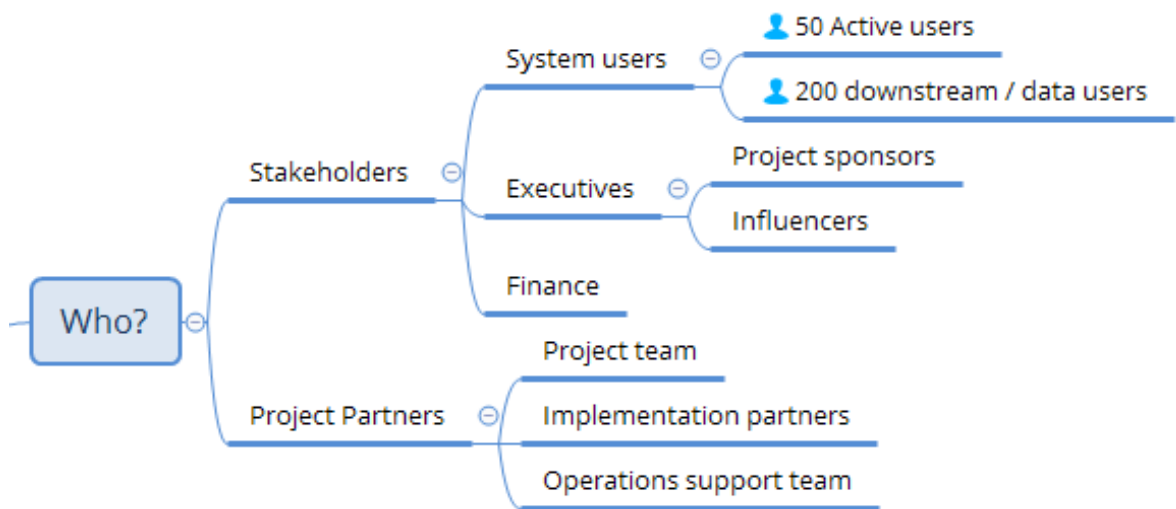


Рисунок 6 – Приклад визначення основних зацікавлених сторін і партнерів проекту [13]

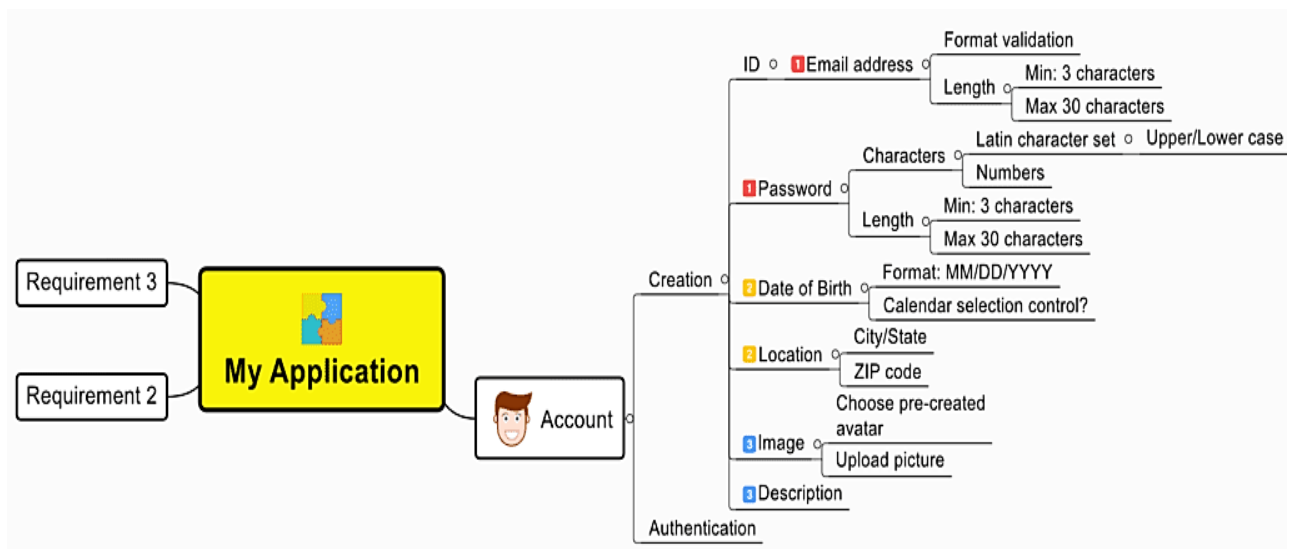



Рисунок 7 – Приклад використання Mind Map для візуального представлення вимоги до ПЗ [6]



5. Управління життєвим циклом вимог: бізнес-аналітик разом з учасниками проекту розробки ПЗ використовують Mind Map під час виконання проекту, щоб зосередитися на вимогах високого рівня та визначити зв'язки між вимогами, обґрунтувати їх реалізацію, наприклад у вигляді певного функціоналу ПЗ, проаналізувати запропоновані зміни у вимогах під час командних обговорень.

Рекомендовано при оновленні інтелект-карти протягом проекту зберігати її версій.

### 1.2.3 Розробка Use Case diagram

Діаграми варіантів використання (діаграма прецедентів, Use Case diagram) нотації UML використовується для [14]:

- подання цілей взаємодії ПЗ з користувачами;
- визначення контексту ПЗ;
- визначення функціональних вимог до ПЗ та їх організація;
- моделювання основного потоку подій у варіанті використання.

Use Case діаграма використовує два основних елементи:

1) варіант використання, або прецеден (Use Case), який характеризує конкретний спосіб взаємодії користувача з ПЗ. Варіанти використання, які представляються графічно як еліпс (рис. 8), відображають функціональність системи. Варіант використання має ім'я, яке може бути розміщене всередині еліпсу або під ним. Отже, Use case (прецедент) – це опис окремого аспекту поведінки системи з точки зору користувача. Прецедент не показує, як досягається певний результат, а тільки яка дія виконується у ПЗ.

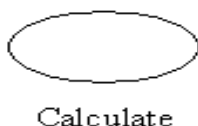


Рисунок 8 – Представлення варіанту використання у Use Case діаграмі

2) користувач - actor (учасник) – це множина логічно пов'язаних ролей, які виконуються при взаємодії з прецедентами або сутностями. Актором може бути людина, роль користувача ПЗ чи інша система, підсистема або клас, які представляють щось поза ПЗ. Один актор може представляти багатьох фізичних осіб, які здійснюють однакові дії або використовують однакові функції ПЗ (рис.9). І навпаки, одна фізична особа може грати декілька ролей, отже її функції описуються багатьма акторами.

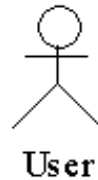


Рисунок 9 – Представлення актора у Use Case діаграмі

У Use Case діаграмі визначаються різні типи зв'язків між акторами та варіантами використання.

1. Асоціація – зв'язок між актором та варіантом використання, зазвичай пре) представлена на рис. 10. Цей зв'язок не визначає напрямок взаємодії, але вказує на загальну асоціацію. Наприклад, якщо два випадки використання часто зустрічаються разом або мають спільні елементи, це можна представити за допомогою зв'язку асоціації [15].



Рисунок 10 – Представлення зв'язку «асоціація» у Use Case діаграмі

У нотації UML можна зображати асоціацію лінією зі стрілкою. Стрілка починається з елемента, до якого належить ініціатива. Іноді це буває не актор, а варіант використання. Наприклад, програмна система може автоматично повідомляти користувача про системні помилки.

2. Зв'язок "включає" ("include") показує, що додатковий варіант використання є обов'язковим (рис. 11). Стрілка починається з основного варіанту використання та вказує на додатковий.

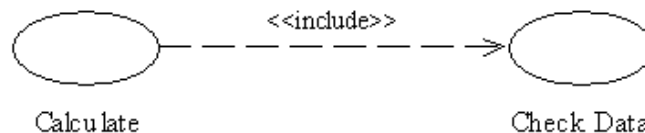


Рисунок 11 - Представлення зв'язку "включає" у Use Case діаграмі

Відношення включення часто використовується для представлення спільно використовуваних функцій. Наприклад, варіант використання «Здійснити оплату» може включати варіант використання «Автентифікація користувача» [15].

3. Зв'язок "розширює" ("extend") показує, що допоміжний варіант використання (дія) не є обов'язковим, але він розширює функціональність базового випадку використання ПЗ. Стрілка

починається з додаткового варіанту використання та вказує на основний (рис. 12). Наприклад, «Завантаження даних з файлу» розширює (уточнює) загальну функцію «Завантаження даних».

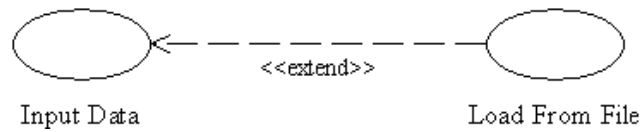


Рисунок 12 - Представлення зв'язку "розширює" у Use Case діаграмі

Зв'язок "extend" вказує на те, що розширений варіант використання може додати додаткову поведінку до базового варіанту використання, якщо виконуються певні умови. Наприклад, варіант використання «Обробка замовлення» може бути розширений варіантом використання «Застосувати знижку», якщо користувач має право на знижку [15].

На рис. 13 продемонстровано використання зв'язків асоціації, включення та розширення для представлення вимог до програми, яка розв'язує квадратне рівняння. Єдиним актором є користувач. Він може вводити дані, розв'язувати рівняння, переглядати результати на екрані. Базові варіанти можна розширити додатковими - введення даних з файлу та зберігання результатів у іншому файлі. Крім того, розв'язання рівняння передбачає обов'язкову перевірку дискримінанту.

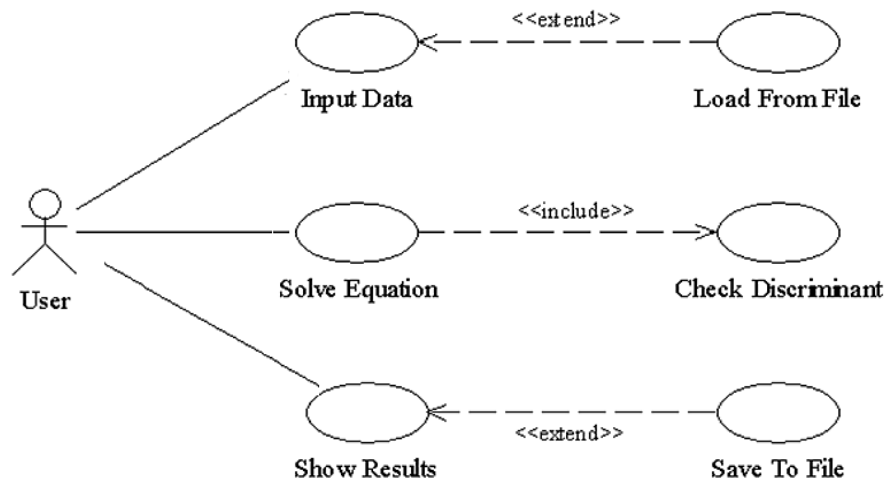


Рисунок 13 - Діаграма варіантів використання програми розв'язання квадратного рівняння

4. Зв'язок "узагальнення" показує взаємодію між конкретним елементом з узагальненим, це зв'язок «батько-нащадок» Здебільшого цей зв'язок використовується між конкретним користувачем та більш



узагальненим актором. Узагальнення показано у вигляді спрямованої стрілки з трикутником. Дочірній варіант використання з'єднаний біля основи стрілки, а кінчик стрілки пов'язаний з батьківським варіантом використання.

Наприклад, як показано на рис. 14 у Адміністратора узагальнені функції, а User може робити все те, що робить Адміністратор .

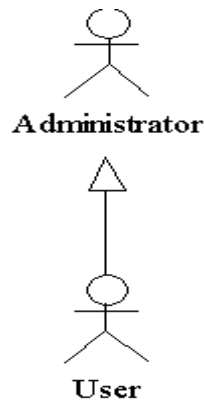


Рисунок 14 - Зв'язок "узагальнення" у Use Case діаграмі

Узагальнення також можна застосувати до варіантів використання. Коли один варіант використання узагальнює інший, це означає, що узагальнений варіант використання служить суперкласом, а узагальнюючий варіант використання є підкласом, який успадковує його поведінку. Це зазвичай використовується, щоб показати, як конкретизований варіант використання успадковує характеристики більш загального (рис.15).

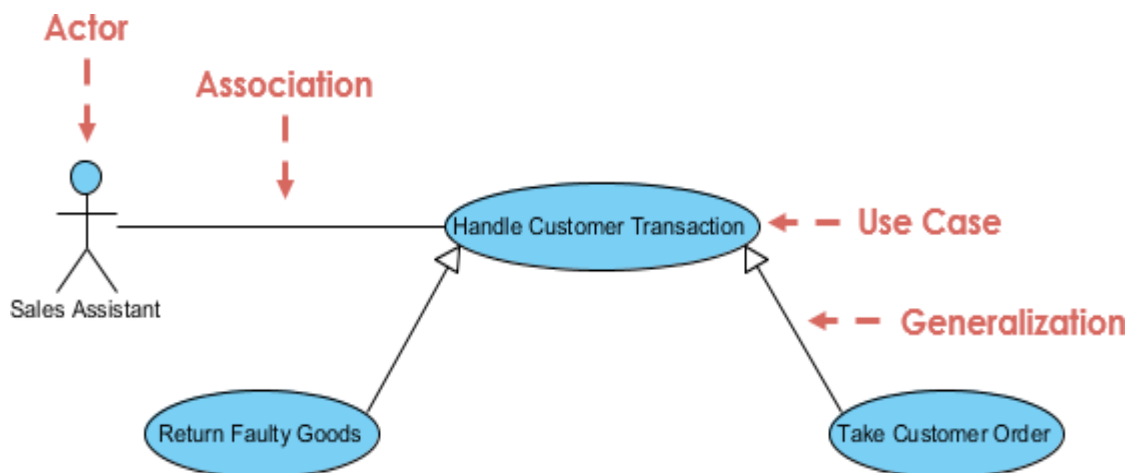


Рисунок 15 - Приклад зв'язку "узагальнення" для варіантів використання

Отже, зв'язки у діаграмі допомагають забезпечити більше розуміння того, як пов'язані різні варіанти використання у програмному додатку та як вони взаємодіють

Ілюстрація основних типів зв'язків Use Case діаграми наведено на рис.16.

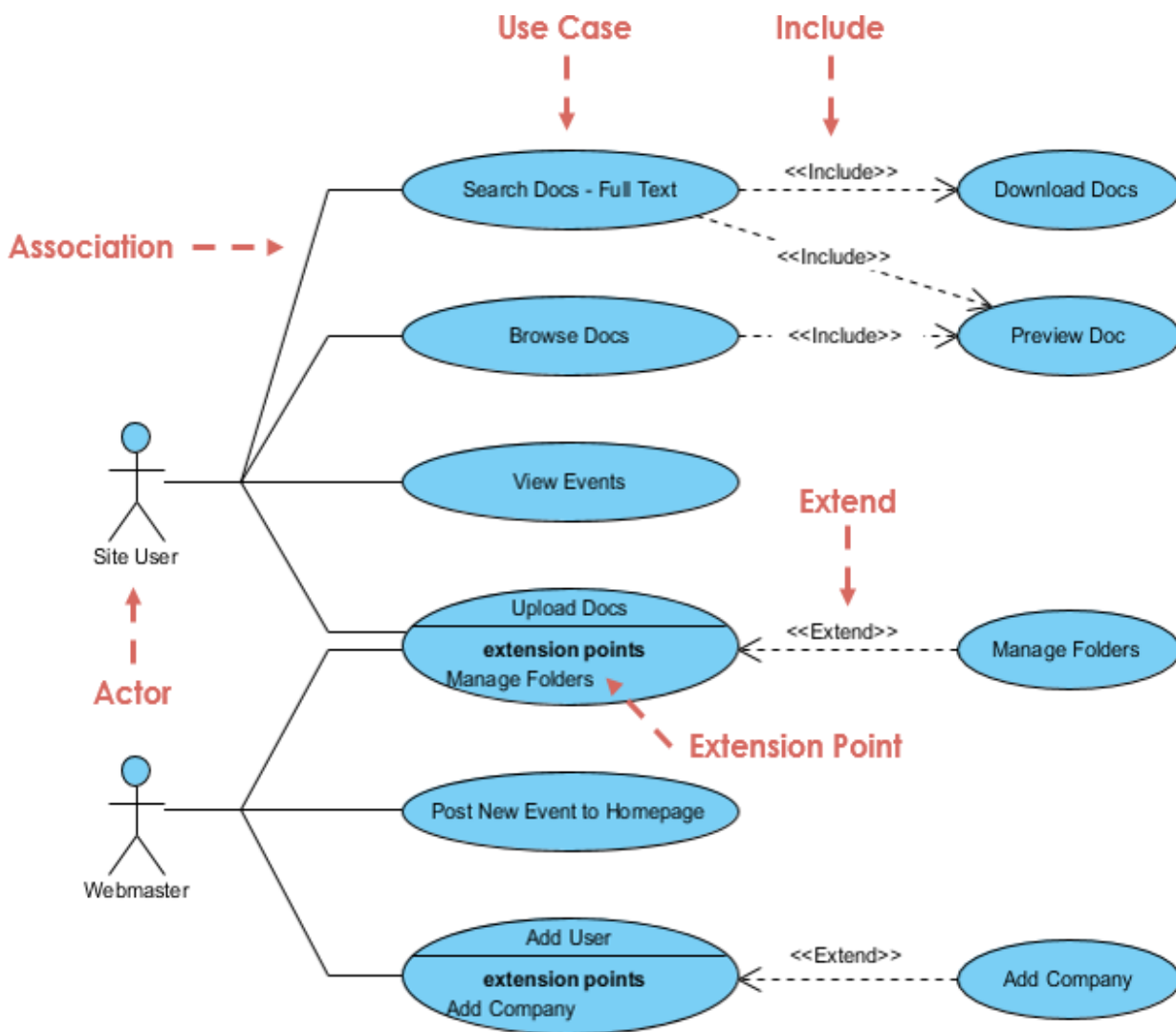
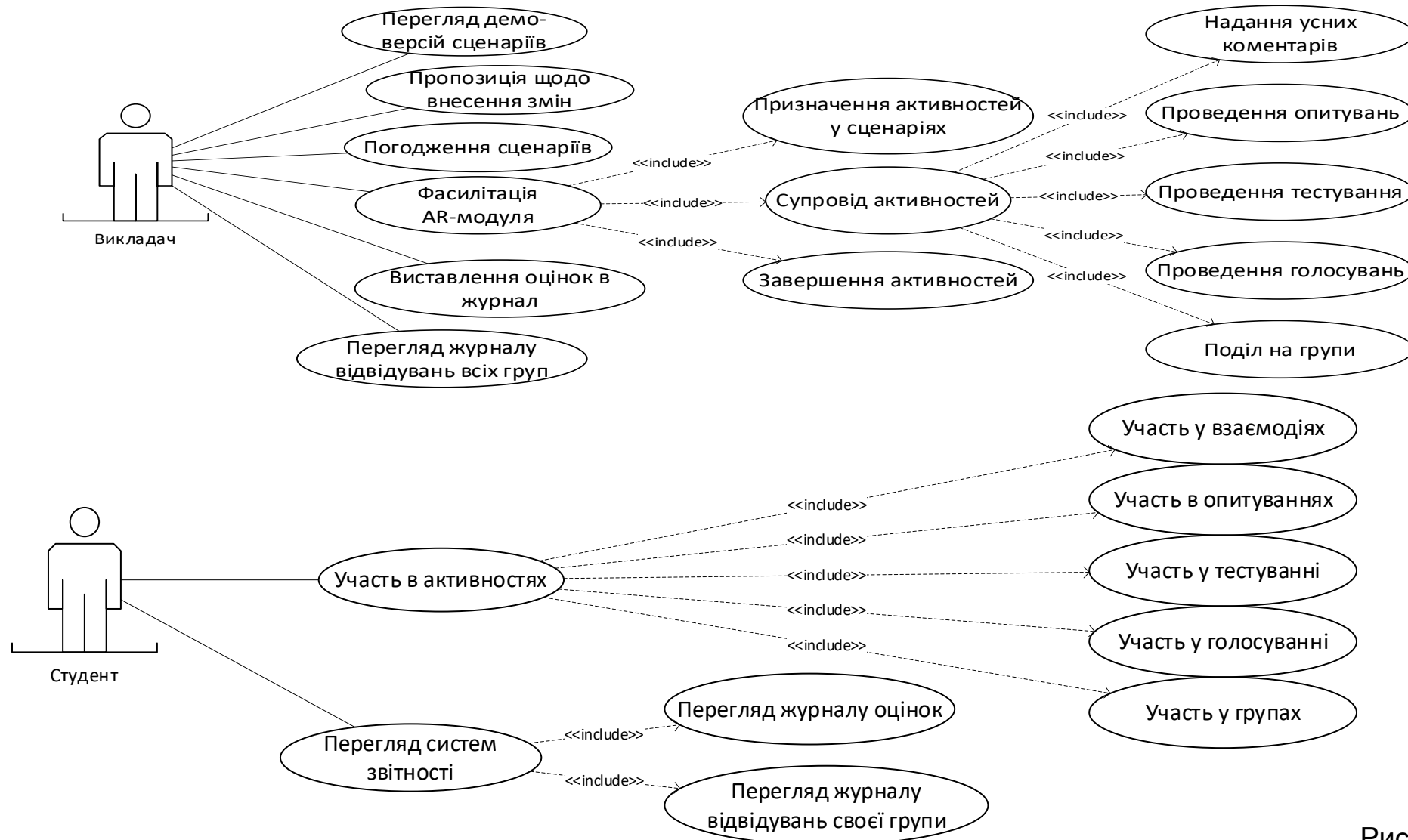


Рисунок 16 – Ілюстрація типів зв'язків Use Case діаграми [15]

Якщо діаграма варіантів використання складна, то її можна представити для кожного актора окремо.

Для наведеного у п.1.2.1 прикладу проблеми, вирішенню якому присвячена дипломна робота, представлено діаграми варіантів використання для акторів Адміністратор, Викладач, Студент на рисунках 17 та 18. Отже, для основних користувачів програми-асистента «Навчання» (модуля навчання системи безперервного удосконалення hr-сервісів) сформовані варіанти використання ПЗ, яке планується розробити в межах дипломної роботи.





Рисунок

18 - Діаграма варіантів використання для Викладача та Студента



## 1.2.4 Розробка Activity diagram

Розробка діаграми діяльності - Activity diagram – це окремий етап моделювання процесу вирішення проблеми, розв'язання задачі, який реалізується в межах ПЗ або в окремих його модулях. Діаграма діяльності допомагає візуалізувати робочі процеси, процеси або дії в програмній системі, показує, як різні елементи взаємодіють у системі [16]. Такі діаграми відображають порядок, у якому відбуваються дії, чи відбуваються вони одна за одною (послідовні) чи одночасно (одночасні дії). Ці діаграми допомагають пояснити, що викликає певні дії чи події у програмній системі

Діаграму діяльності у межах виконання ІДЗ студент може представити у таких ситуаціях [17]:

- 1) моделювання робочих процесів або бізнес-процесів, або робочого циклу (етапів), пов'язаних із варіантом використання. У цьому випадку студент створює діаграму діяльності для ілюстрації здійснення варіанту використання або декількох варіантів використання, які були представлені на Use-case діаграмі;

- 2) візуалізація переходу між різними станами системи, яка досліджується в межах дипломної роботи, наприклад, вивчається поведінка технічної або організаційної системи і треба показати динамічну її поведінку на основі подій або за різних умов функціонування, показати, як система змінюється з часом;

- 3) роз'яснення складної логіки вирішення якоїсь проблеми або представлення алгоритму розв'язання задачі, демонстрація процесу прийняття рішень із розгалуженими тощо;

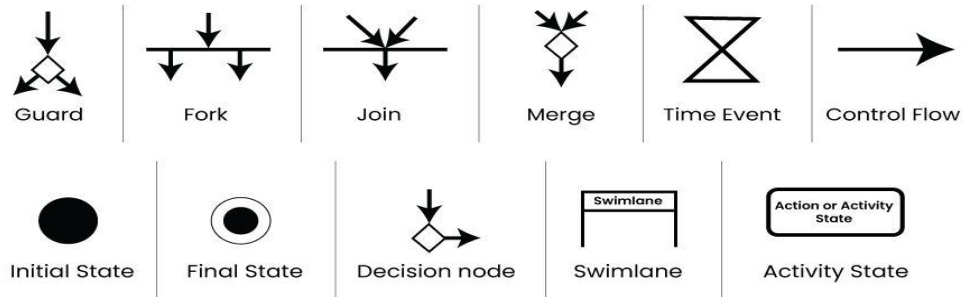
- 4) на етапі проектування й аналізу програмної системи є необхідність продемонструвати розробникам та іншим зацікавленим сторонам динамічну взаємодію різних частин цієї системи за різними умовами. Наприклад це важливо, коли розроблюється система, яка буде інтегруватися у більш складну систему або потребується при функціонуванні цієї програмної системи оброблювати послідовність певних подій (експорт, імпорт даних та ін.), тоді це все можна представити у вигляді діаграми активності.

Отже, при аналізі діаграми варіантів використання кожен прецедент реалізується за допомогою якихось дій. Якщо ці дії формують складний процес (або певні сценарії), то це доцільно представити у вигляді діаграми активності. Зазвичай діаграми діяльності включають такі елементи (рис. 19) [17, 18]:

- початковий стан;
- діяльність;
- перехід;
- розгалуження;
- поєднання потоків, дій, синхронізація;



- розподіл потоків (вилка);
- кінцевий стан.



Activity Diagram Notations



Рисунок 19 – Позначення у діаграмі діяльності [16]

Кожна діаграма діяльності повинна мати один і тільки один початковий стан (рис. 20).



Рисунок 20 - Початковий стан

Усі види діяльності (введення, виведення даних, розрахунки) зображають у вигляді прямокутника із закругленими кутами (рис. 21). Дія представляється у вигляді короткого та зрозумілого опису у середині форми та зазвичай характеризує атомарну операцію.

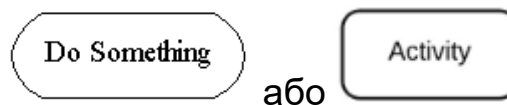


Рисунок 21 - Діяльність

Перехід між окремими діями завжди зображують у вигляді стрілки (рис.22). Стан активності може мати кілька вхідних і вихідних дій.

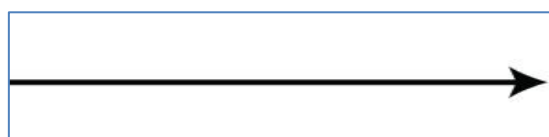


Рисунок 22 – Перехід, потік керування



Розгалуження або вузол прийняття рішення, як і у блок-схемі, зображується ромбом. Але, на відміну від блок-схем, умови переходу записують не всередині ромбу, а біля стрілок у квадратних дужках (рис.23). Він завжди містить дві або більше вихідних стрілок. Але треба представити чітко умову, щоб переконатися, що потік керування або перехід йде лише одним шляхом.

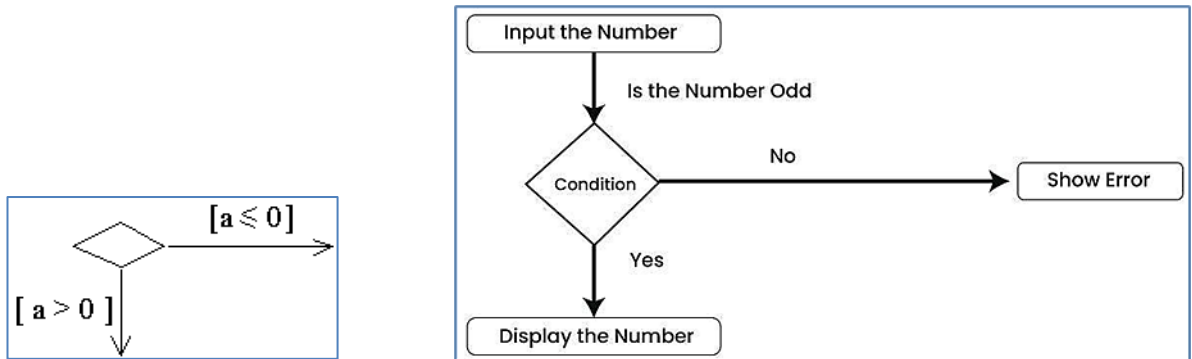


Рисунок 23 - Розгалуження

Вузол злиття об'єднує разом різні шляхи прийняття рішень, які були створені за допомогою вузла прийняття рішень. Вузол злиття використовується, коли дії, які не виконуються одночасно, потрібно об'єднати. Об'єднуються дві або більше дій в одну, якщо керування переходить до наступної діяльності, незалежно від обраного шляху попередньо. На рис.24 показано, що обидві сторони дій після розгалуження не можуть виконуватися одночасно (дії їх не сумісні), але вони зрештою зливаються в одну дію. Число не може бути водночас парним і непарним.

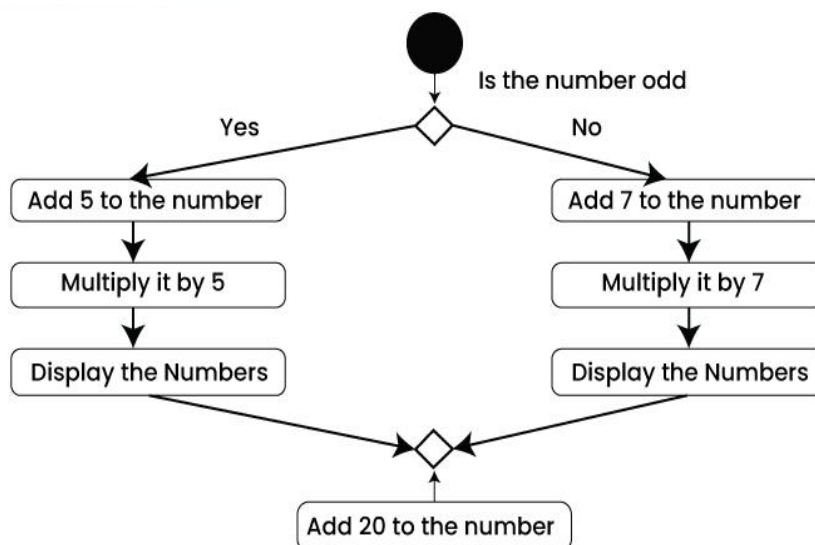


Рисунок 24 - Вузол злиття для прикладу визначення парного або непарного числа



Вузол вилки розділяє дію на набір паралельних або одночасних потоків (дій) (рис.25).

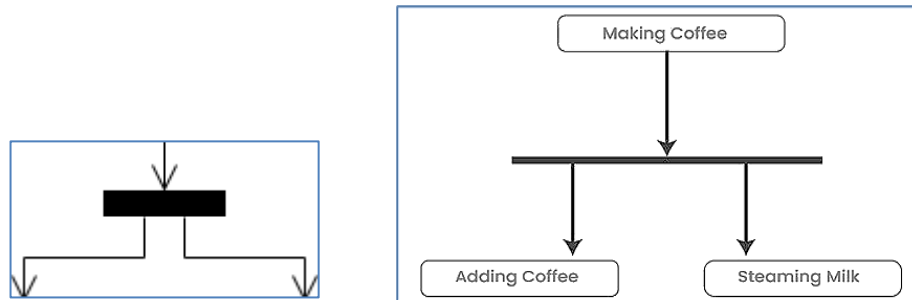


Рисунок 25 - Вузол вилки

Об'єднати набір паралельних або одночасних потоків (або дій) можна за допомогою об'єднуючого вузла (вузол об'єднання). На рис.26 показано об'єднання двох паралельних дій.

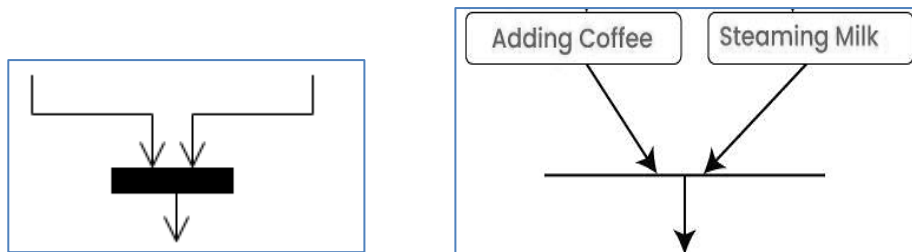


Рисунок 26 - Розгалуження

Отже, на відміну від традиційних блок-схем, на діаграмі діяльності можна зобразити дії, які виконуються паралельно. Розгалуження таких дій починається та закінчується лінійкою синхронізації (рис.27).

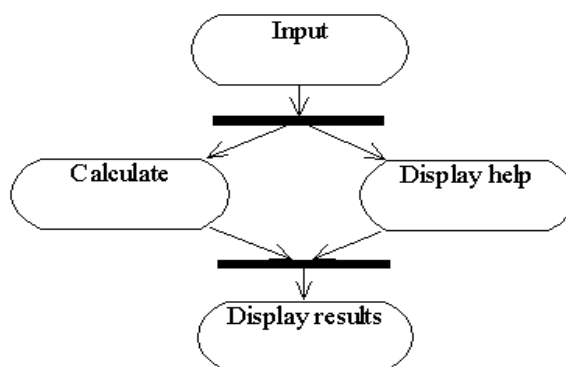


Рисунок 27 – Паралельність дій

Стан, якого система досягає, коли певний процес або діяльність завершується називається кінцевий стан. Кожана діаграма повинна

мати один або декілька кінцевих станів (рис.28). Використовується не повністю зафарбоване коло (подібно представлення кінцевого стану на діаграмі кінцевого автомата).

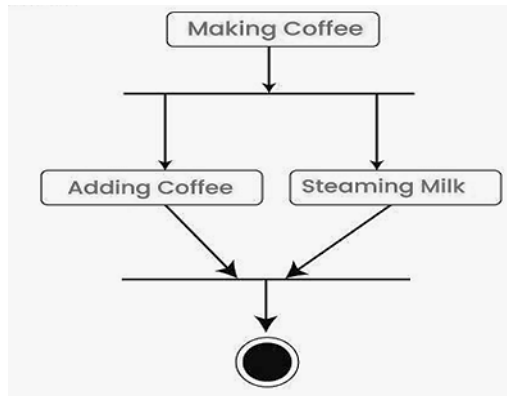


Рисунок 28 - Кінцевий стан

Загальне представлення найчастіше використовуваних елементів діаграми активності наведено на рис.29.

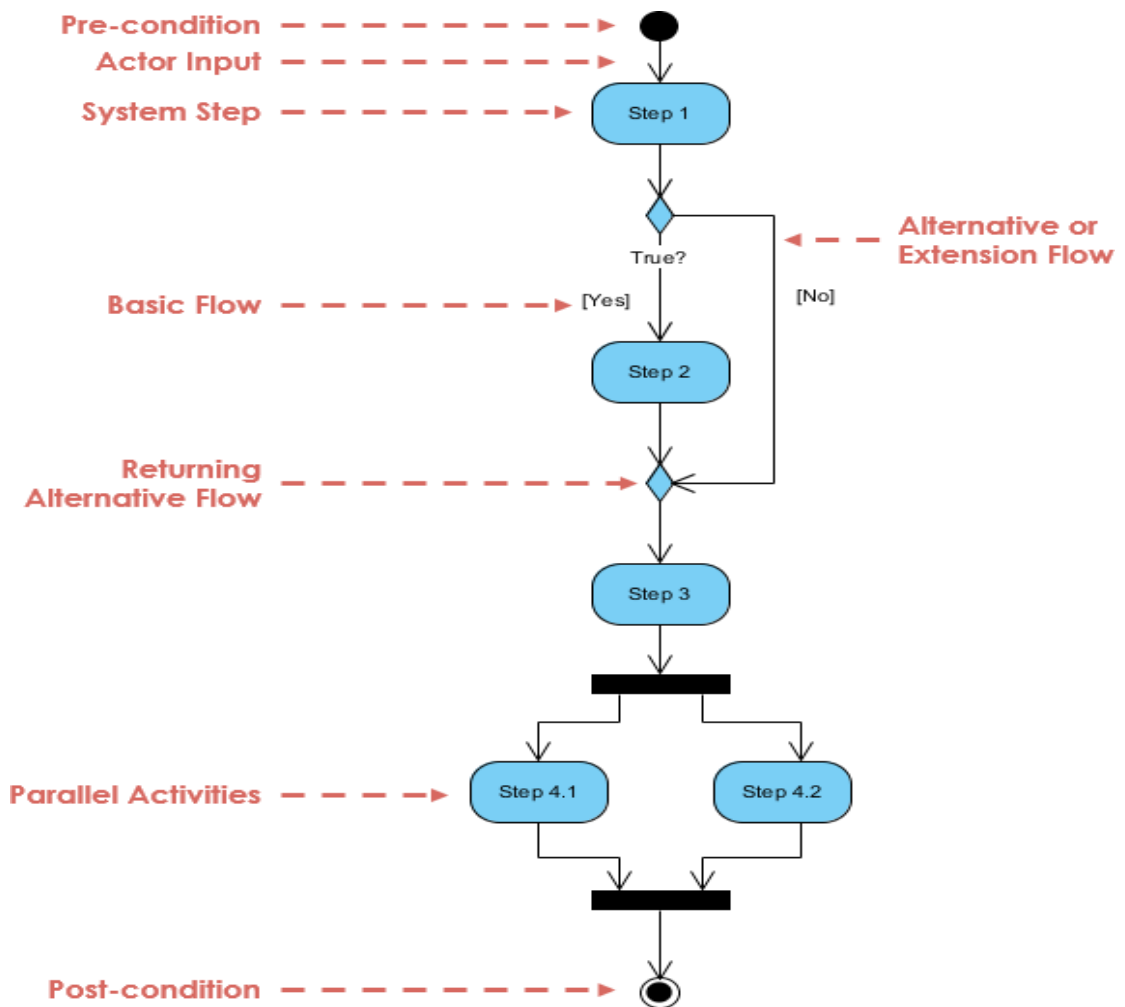


Рисунок 29 – Діаграма діяльності [17]

На рис. 30 представлено алгоритм методу дихотомії для розв'язання рівняння на інтервалі  $[a, b]$  у вигляді Activity Diagram.

На рис. 31 наведено діаграму активності, яка побудована для відображення процесу фіксування відсутніх на заняттях тренінгу. Цей процес включено до системи процесів, які забезпечують виконання варіанту використання «Керування системами звітності» для модуля навчання системи безперервного удосконалення hr-сервісів (приклад опису проблеми надано у п.1.2.1, а діаграми варіантів використання на рис.17 та 18).

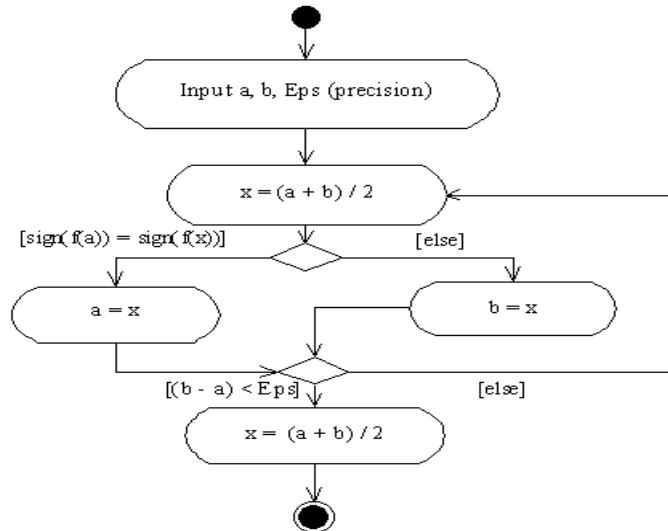


Рисунок 30 – Метод дихотомії

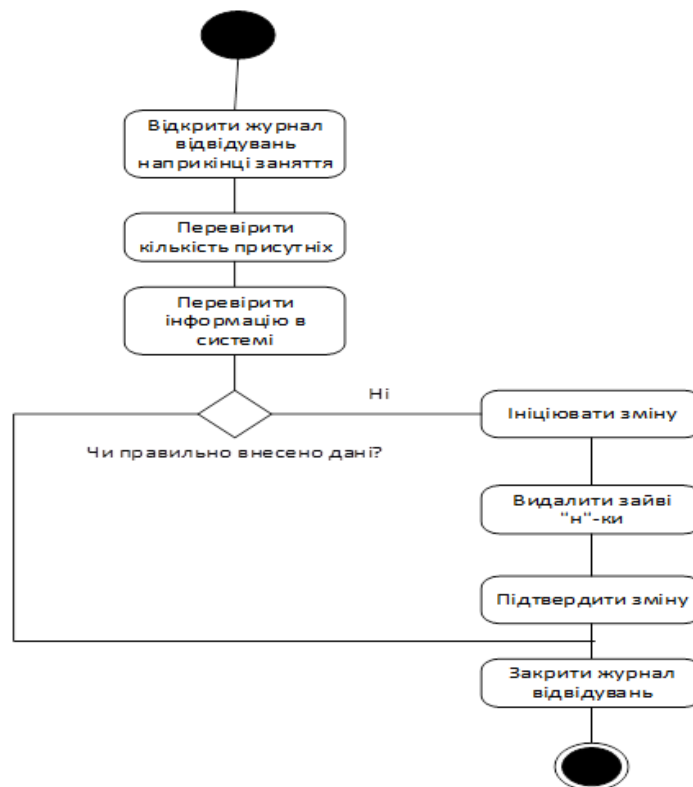


Рисунок 31 – Діаграма активності для фіксування відсутніх на заняттях



### 1.2.5 Діаграма послідовності

Діаграма послідовності — це тип діаграми взаємодії, яка описує, як і в якому порядку об'єкти працює разом. Ці діаграми використовуються розробниками ПЗ та бізнес-аналітиками, щоб зрозуміти вимоги до нової системи або задокументувати існуючий процес. Діаграми послідовності іноді називають діаграмами подій або сценаріями подій [19].

Діаграми послідовності показують порядок повідомлень, які передаються між елементами системи (ПЗ) для виконання конкретного завдання або варіанту використання. Події, які перетинають межі системи, використовуються об'єктами та людьми (акторами) для завершення своїх процесів [20].

Основні позначення діаграми послідовності.

1. Актор – це тип ролі, яку відіграє суб'єкт, який взаємодіє з системою (ПЗ, модулем системи), наприклад, шляхом обміну сигналами та даними. Актор є зовнішнім суб'єктом по відношенню до ПЗ (модуля, системи тощо). Актори представляють ролі, які виконують користувачі, зовнішнє обладнання чи інші зовнішні системи [21]. Актор не обов'язково представляє конкретну фізичну сутність, а лише певну роль якоїсь сутності, наприклад, клієнт банку. Актор використовується, коли конкретна діаграма послідовності належить варіанту використання.

2. Лінія життя представляє окремого учасника взаємодії, вона починається з об'єкта-прямокутника (голова) та зображується вертикальною пунктирною лінією. Вона служить для позначення періоду часу, протягом якого об'єкт існує в системі. Якщо об'єкт існує у системі постійно, то його лінія життя повинна продовжуватися по всій площині діаграми зверху донизу. Зазвичай об'єкти перераховуються зліва направо. Кожен екземпляр системи/об'єкта та актор (рис. 32) розміщено на лінії життя (вертикальній пунктирній лінії), яка проходить у верхній частині діаграми послідовності.

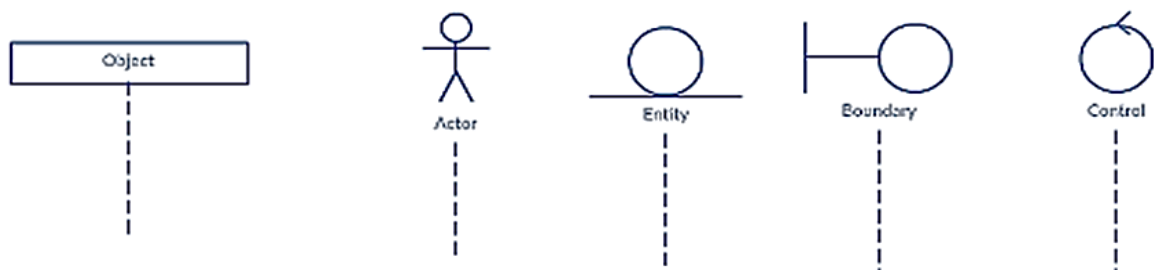



Рисунок 32 – Позначення на лінії життя [14]



Окрім об'єкта та актора на лінії життя використовують такі позначення [14]:

- сутність (entity) — представляє системні дані. Наприклад, у програмі обслуговування клієнтів суб'єкт — клієнт керує даними (сутність), пов'язаними з клієнтом.

- межа/кордон (boundary) — вказує на межу системи/ граничний елемент у системі; наприклад, екрани інтерфейсу користувача, шлюзи баз даних або меню, з якими взаємодіють користувачі

- управління (control) вказує на керівну сутність або менеджера. Він організовує та планує взаємодії між кордонами та сутностями та служить посередником між ними (рис.32).

3. Смуга активації (фокус управління)— тонкий прямокутник на лінії життя, протягом якого елемент виконує операцію. Довжина прямокутника вказує на тривалість перебування об'єктів у активному режимі. Верхня та нижня частини прямокутника вирівняні відповідно до часу початку та завершення

4. Повідомлення (виклики) з'являються у послідовному порядку на лінії життя. Повідомлення зображується за допомогою стрілок. Початок стрілки завжди торкається лінії життя відправника та лінії життя об'єкта (ПЗ, модуль системи тощо), що приймає повідомлення. Підпис може знаходитись над або всередині стрілки повідомлення. Для зручності перед повідомленням можна проставляти нумерацію дій. Повідомлення діляться на такі категорії (рис.33):

- 1) синхронне повідомлення очікує на відповідь, перш ніж взаємодія відправника з об'єктом буде продовжено. Відправник (абонент) чекає, поки одержувач (об'єкт) завершить обробку повідомлення. Абонент продовжує лише тоді, коли він знає, що одержувач обробив попереднє повідомлення, тобто він отримує повідомлення у відповідь. Велика кількість викликів в об'єктно-орієнтованому програмуванні є синхронними. Синхронного повідомлення представляється повідомлення як суцільна стрілка для представлення (зазвичай зафарбований вказівник);

- 2) асинхронне повідомлення не передбачає відповіді від отримувача. Взаємодія продовжується незалежно від того, обробив одержувач попереднє повідомлення чи ні. Використовується суцільна лінія зі звичайним вказівником стрілки;

- 3) створення учасника (Create Message) — це повідомлення, що створює нову лінію життя. Позначається пунктирною лінією, направленою до прямокутника учасника (рис.34). Якщо створений учасник робить щось після його створення, слід додати вікно активації під його полем.

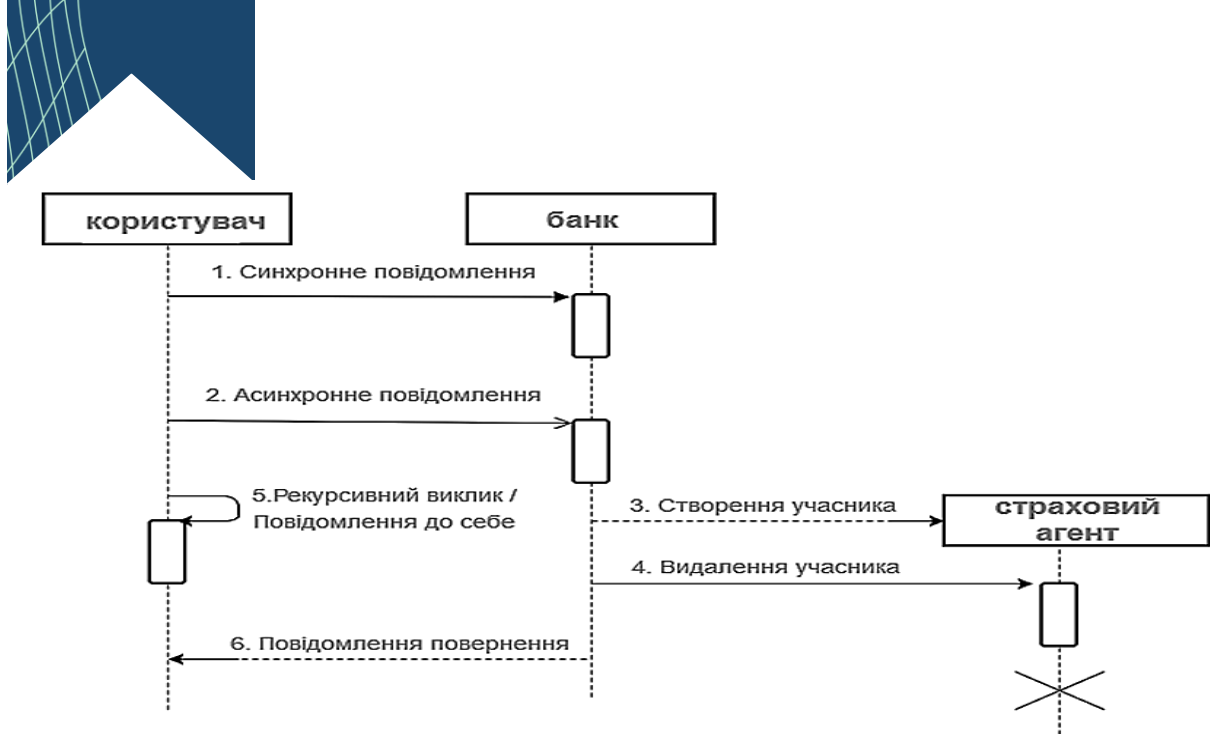


Рисунок 33 – Типи повідомлень у діаграмі послідовності [14]

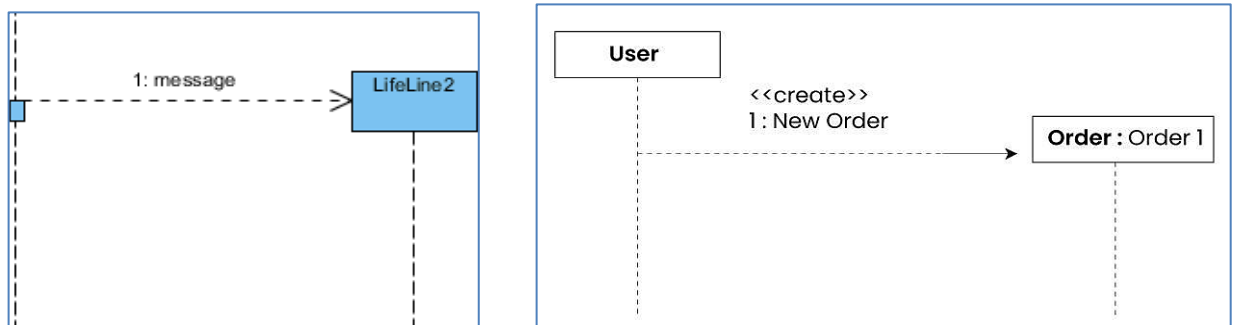


Рисунок 34 – Представлення повідомлення щодо створення нової лінії життя [21, 22]

4) видалення повідомлення або об'єкту (Destroy Message). Видалення об'єкту позначається хрестом в кінці лінії його життя. При цьому, якщо стрілка повідомлення направлена від одного учасника до іншого, значить, один учасник видаляє іншого. Якщо стрілка відсутня, учасник самознищується. Повідомлення Destroy — це свого роду повідомлення, яке представляє запит на знищення життєвого циклу цільової лінії життя (рис. 35).

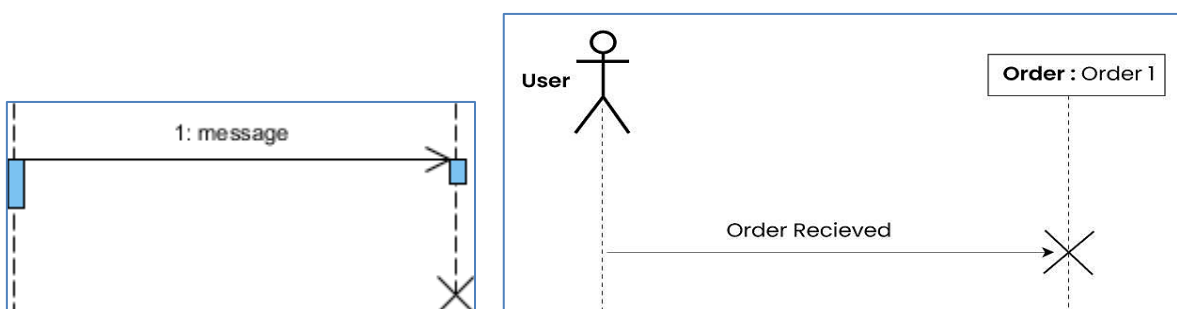


Рисунок 35 – Представлення видалення повідомлення [21, 22]



5) рекурсивне повідомлення ( Recursive Message), яке починається та закінчується на одній лінії життя. Прикладом може служити отримання доступу до камери смартфоном (рис. 36). Рекурсивне повідомлення — це різновид повідомлення, яке представляє виклик повідомлення тієї ж лінії життя (самоповідомлення, Self Message). Його ціль вказує на активацію поверх активації, з якої було викликано повідомлення.

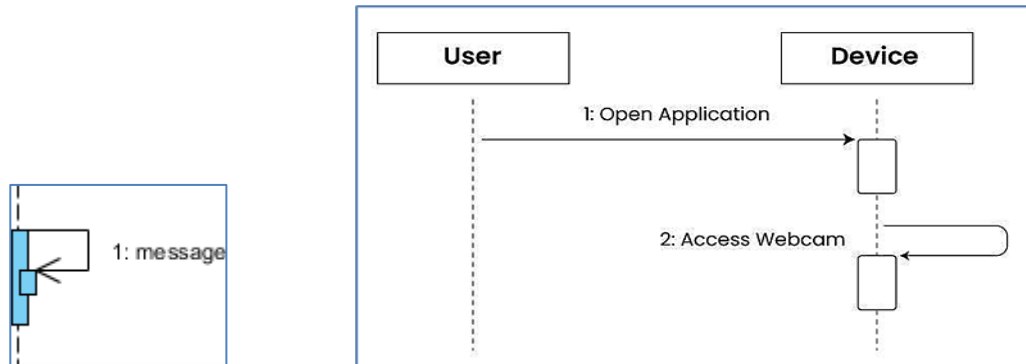


Рисунок 36 – Позначення рекурсивного повідомлення та приклад Self Message отримання користувачем доступу до камери від смартфона [21, 22]

6) повідомлення повернення (пунктирна лінія із зафарбованим вказівником) або зворотне повідомлення викликаючій стороні (рис. 37). Можна уникнути захаращення діаграм, вказуючи значення у самій стрілці початкового повідомлення.

Для зрозумілого представлення логіки діаграми послідовності включаються примітки. Примітка (або коментар) у діаграмі послідовності дає можливість додавати різні зауваження до елементів. Коментар не має семантичної сили, але може містити інформацію, корисну для розробника цієї діаграми та стейкхолдерам проекту з розробки ПЗ (рис.38).

Фокус контролю - це випадок виконання, який зображають як високий тонкий прямокутник на лінії життя. Він представляє період, протягом якого елемент виконує операцію. Верхня та нижня частини прямокутника вирівнюються відповідно до часу початку та завершення (рис. 38).

Фрагменти послідовності (або взаємодії) спрощують створення та підтримку точних діаграм послідовностей.

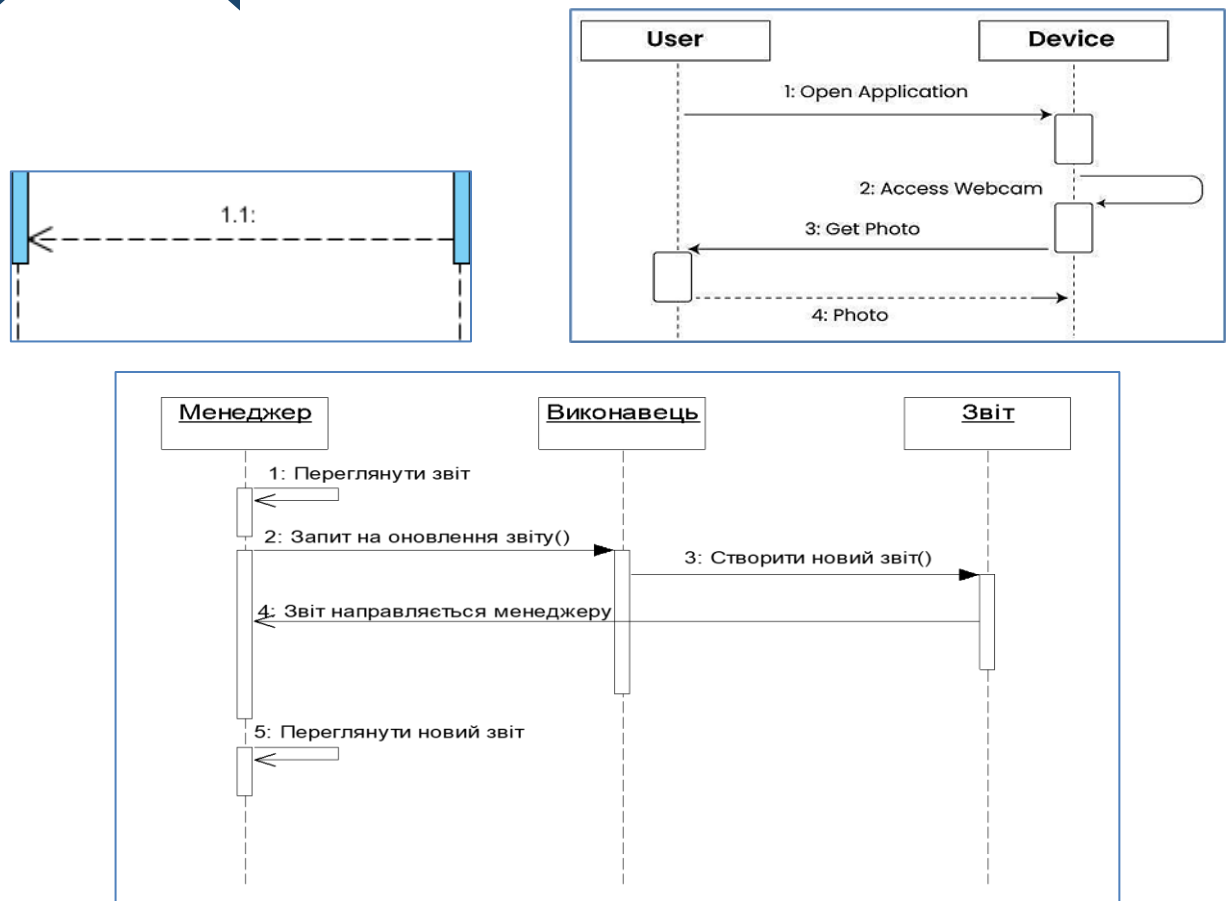


Рисунок 37 – Приклади позначення зворотнього повідомлення та використання з іншими повідомленнями [21, 22]

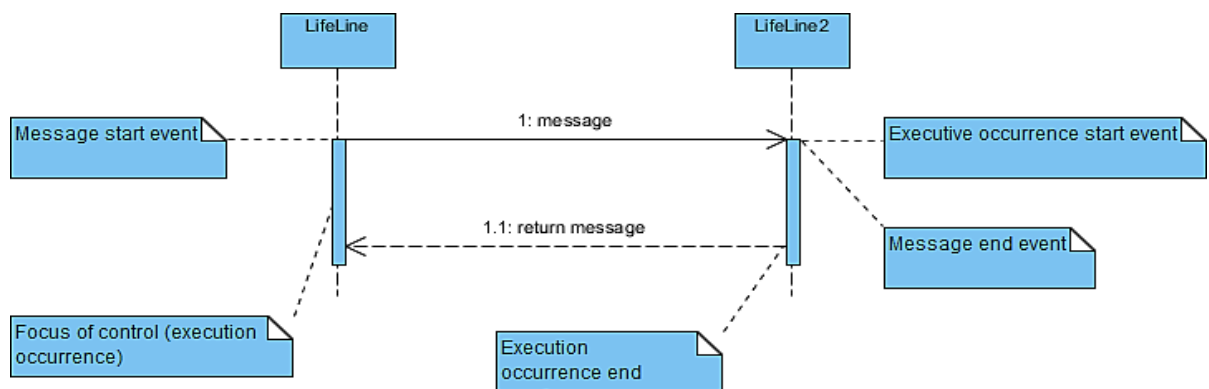


Рисунок 38 – Використання приміток та фокусу контролю [21]

Фрагмент послідовності представлено у вигляді прямокутника, який називається комбінованим фрагментом, який охоплює частину взаємодій у діаграмі послідовності. Оператор фрагмента (у верхньому лівому куті) вказує на тип фрагмента. Типи фрагментів *ref*, *assert*, *loop*, *break*, *alt*, *opt*, *neg* наведені у табл.2 [21].

Таблиця 2 – Типи фрагментів послідовності

Оператор	Тип фрагмента
alt	Альтернативні фрагменти: виконуватиметься лише той фрагмент, умова якого є істинною.
opt	Необов'язковий фрагмент: фрагмент виконується, якщо задана умова є істинною.
par	Паралельний фрагмент: кожен фрагмент виконується паралельно.
loop	Циклічний фрагмент може виконуватися кілька разів, а умова (guard) вказує на основу ітерації.
region	Критичний фрагмент (критична область): фрагмент може мати лише один потік, який виконує його один раз.
neg	Негативний фрагмент показує недійсну послідовність.
ref	Фрагмент посилання: для повторного використання частини послідовності в іншому місці діаграми
sd	Діаграма послідовності: використовується для оточування всієї діаграми послідовності.

Приклад діаграми послідовності на рис. 39 показує взаємодію актора – покупця з системою. Покупцем надається запит через форму пошуку інформації до бази даних, він отримує результат в окремій формі [20]. Попередньо проведено валідацію цього покупця і т.д.

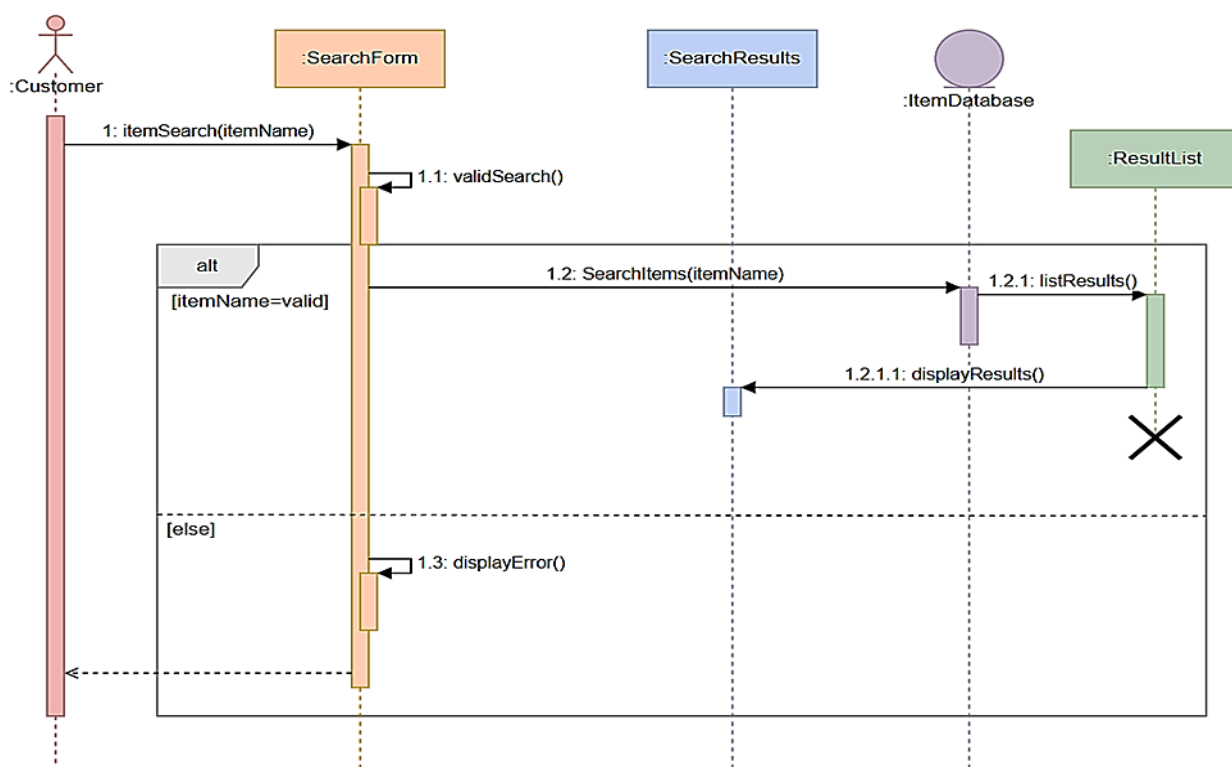


Рисунок 39 – Приклад діаграми послідовності [20]

Можна комбінувати кадри, щоб захопити, наприклад, петлі або гілки. Також для більшого розуміння процесу та уточнення умов

надсилання повідомлень та ін. використовуються часові та інші обмеження у повідомленнях. Наприклад, вони можуть застосовуватися до часу надсилання одного повідомлення або інтервалів між повідомленнями (рис.40).

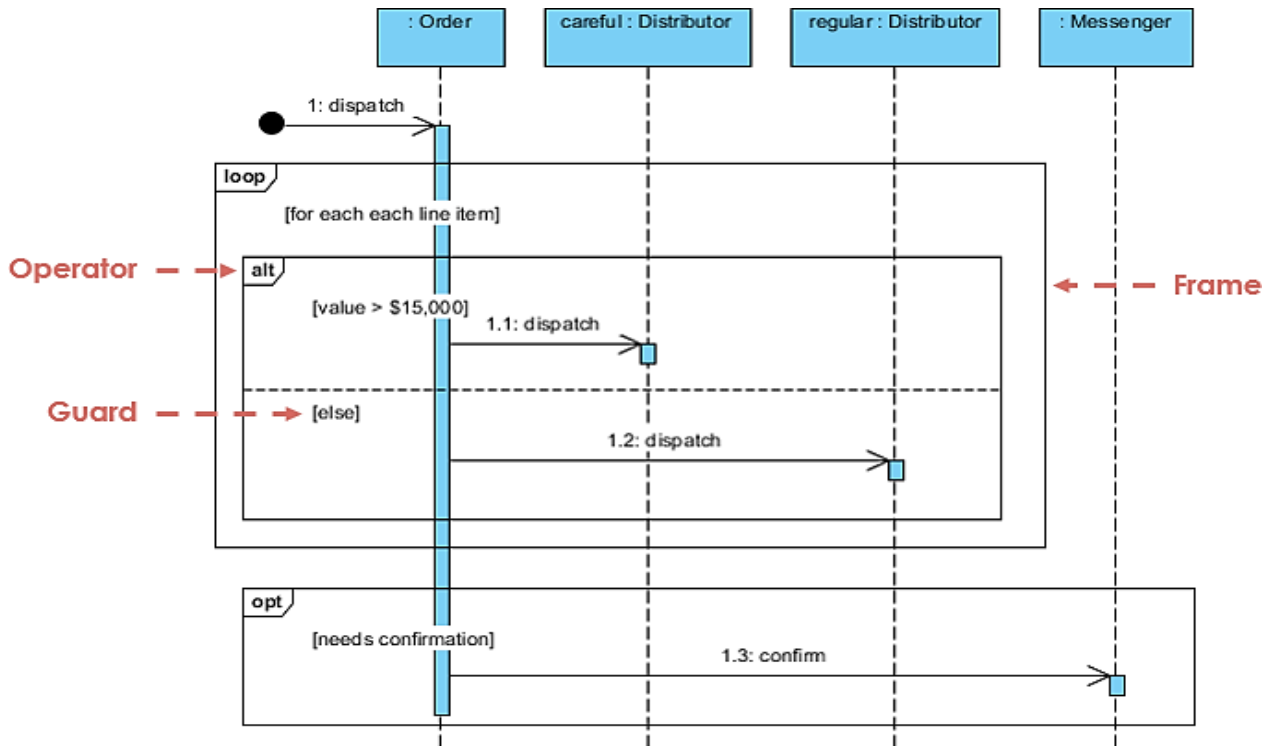


Рисунок 40 – Приклад комбінації фрагментів та використання обмежень у повідомленнях [21]

Для виконання ІДЗ діаграму послідовності треба розробити для моделювання сценаріїв використання. Вимоги користувача, які було зафіксовано як варіанти використання у Use Case діаграмі, треба уточнити у вигляді сценаріїв.

Сценарій — це один шлях або потік, який описує послідовність подій, що відбуваються під час одного конкретного виконання системи (варіанту використання), яке представлено діаграмою послідовності.

На рис. 41 представлено приклад представлення діаграми послідовності для основних сценаріїв та альтернативного шляху виконання Use Case «Пошук книги» [21]. Надано чотири сценарії для варіанту використання «Пошук книги»:

- 1) замовник вказує автора на Сторінці пошуку, а потім натискає кнопку Пошук.
- 2) система перевіряє критерії пошуку Замовника;
- 3) якщо введено автора книги, то система шукає в каталозі книги, які пов'язані з вказаним автором;
- 4) після завершення пошуку система відображає результати пошуку на сторінці результатів пошуку.

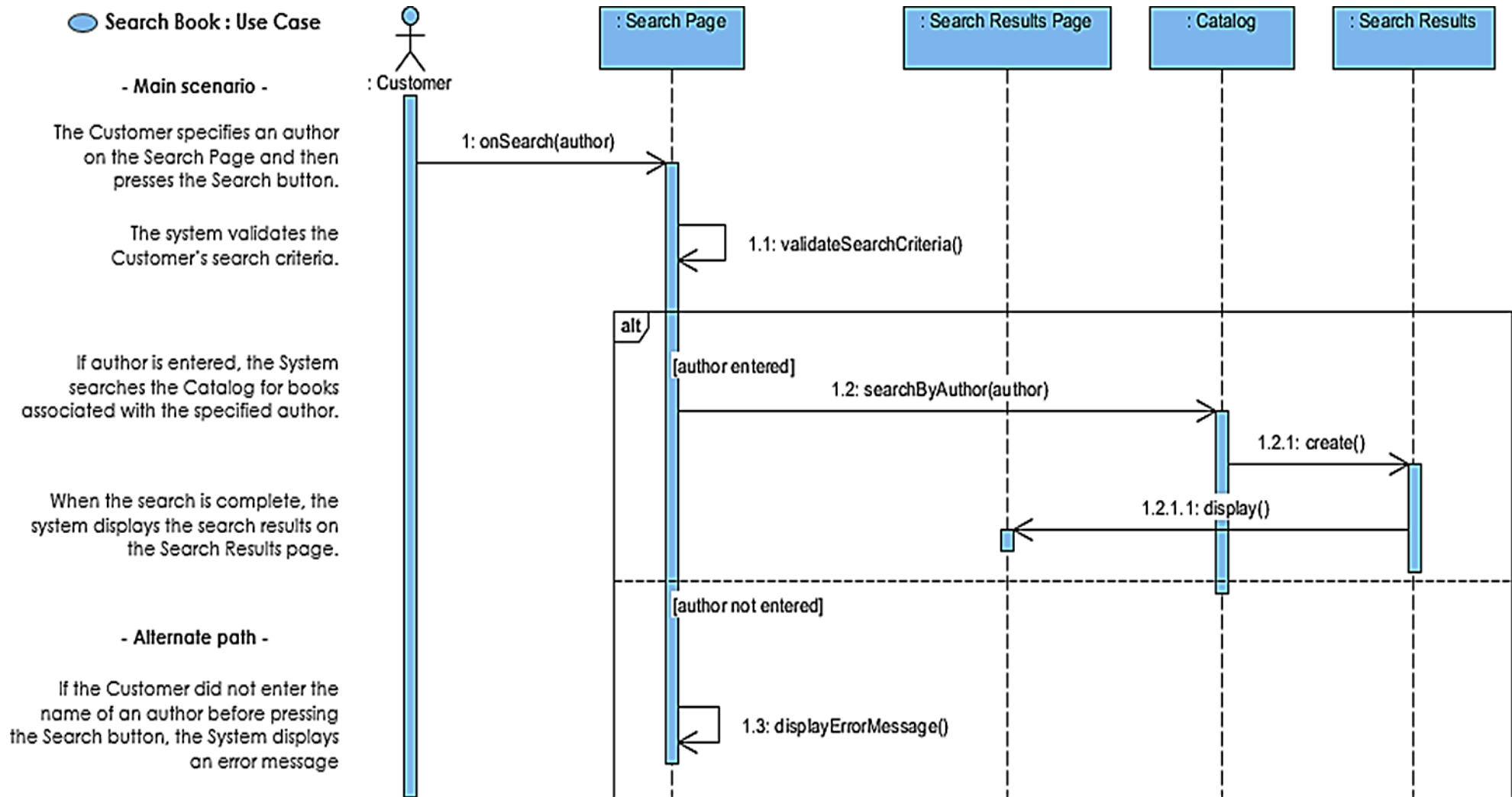


Рисунок 41 – Приклад діаграми послідовності для реалізації сценаріїв Use Case

Представлено реалізацію альтернативного шляху - якщо Замовник не ввів ім'я автора до натискання кнопки «Пошук», то Система видає повідомлення про помилку.

Отже, таке представлення реалізації вимоги щодо пошуку книги була надана у вигляді послідовності дій між користувачем та модулями програмної системи.

На рис. 42 представлено приклад діаграми послідовності для варіанту використання «Розподілу студентів по групах» у модулі навчання (програми-асистента «Навчання») системи безперервного удосконалення hr-сервісів (приклад опису проблеми надано у п.1.2.1, а діаграми варіантів використання – на рис.17 та 18)

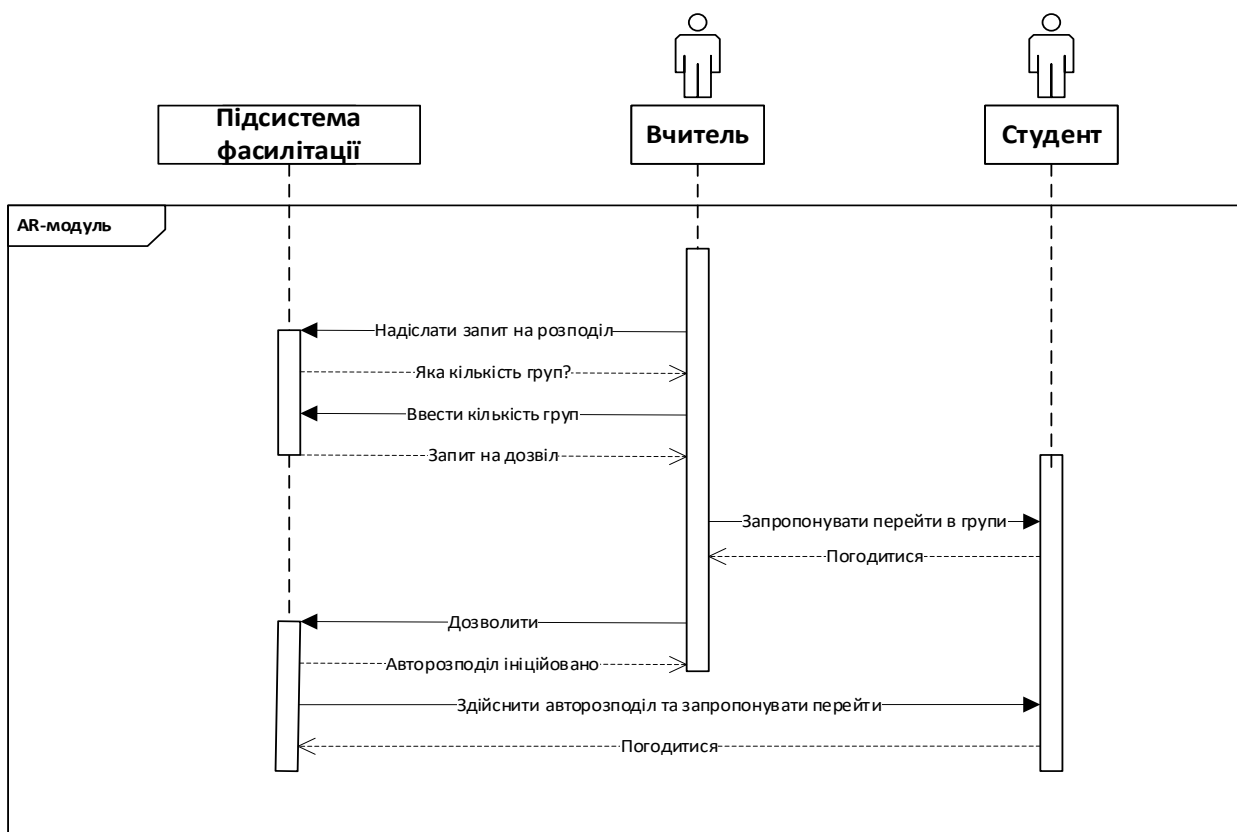



Рисунок 42 - Процес розподілу студентів по групах у модулі навчання системи безперервного удосконалення hr-сервісів

Отже, створення діаграми робиться на етапі проектування розробки програмного забезпечення, щоб проілюструвати, як різні компоненти або об'єкти взаємодіють у часу. Рекомендується розробляти її послідовно [ 22]. Розглянемо основні кроки створення діаграм послідовності.

Крок 1. Визначити сценарій. Студент згідно побудованій Use Case діаграмі обирає варіанти використання, для яких треба розробити процеси – послідовності дій або сценарії взаємодії між об'єктами системи для повного розуміння реалізації вимог, які описані як обрані



варіанти використання. Оскільки на Use Case діаграмі не вказують усі взаємозв'язки між варіантами використання, то для повного розуміння функціоналу ПЗ треба їх проілюструвати на діаграмі послідовності (або за допомогою діаграми послідовності та станів, які розглянуті у п. 1.2.4 та 1.2.6).

Крок 2: Скласти перелік об'єктів, акторів (користувачі, системи або зовнішні сутності), які беруть участь у сценарії. Учасниками можуть бути.

Крок 3. Визначити лінії життя. Лінія життя для кожного об'єкта представляє існування об'єкта під час взаємодії.

Крок 4. Розташувати лінії життя горизонтально у порядку їх участі у взаємодії. Це допомагає візуалізувати потік повідомлень між ними.

Крок 5: Додати панелі активації для кожного повідомлення. Панель активації відображає тривалість часу, протягом якого учасник активно обробляє повідомлення.

Крок 6. Додати повідомлення, використовуючи відповідні стрілки для позначення повідомлень між учасниками. Повідомлення протікають горизонтально між лініями життя, вказуючи на зв'язок між об'єктами.

Крок 7: Вказати час і порядок, використовуючи цифри, щоб вказати послідовність надсилання повідомлень, а також вертикальні пунктирні лінії для представлення подій або плину часу.

Крок 8: Додати умови та цикли, використовуючи комбіновані фрагменти для представлення умов (наприклад, операторів if) і циклів у взаємодії. Це допомагає деталізувати потік керування.

Крок 9: Розглянути та додати паралельне виконання певних дій, накресливши паралельні вертикальні пунктирні лінії та розмістивши повідомлення відповідно.

Крок 10: Проаналізувати та вдосконалити діаграму, щоб переконатися, що вона точно представляє заплановану взаємодію.

Крок 11: Додати анотації та коментарі, які доповняють контекст або роз'яснення для елементів на діаграмі.

Крок 12: Задokumentувати припущення та обмеження, які, пов'язані з взаємодією, щоб потім при розробці ПЗ це врахувати.

Для побудови такої діаграми використовують інструменти моделювання UML [19-22].

**Увага: студент обирає на свій розсуд (або після консультації з керівником дипломної роботи, викладачем цього курсу та іншими фахівцями у проєктуванні ПЗ), які варіанти використання «роз'яснити» за допомогою діаграми діяльності, а які за допомогою діаграми послідовності!**

У чому полягає різниця між діаграмою діяльності та діаграмою послідовності? Головна мета діаграми послідовності- це показати порядок виконання, або послідовність дій між користувачем (зовнішнім

об'єктом) та модулями (компонентами, класами) ПЗ, який розроблюється. Водночас діаграма діяльності потрібна для опису роботи всієї системи, вона ілюструє послідовність основних дій користувача з системою та/ або послідовність вирішення проблеми, розв'язання задачі тощо. Потік керування (порядок виконання) на діаграмі діяльності переходить від однієї операції до іншої. Цей потік може бути послідовним, розгалуженим або одночасним. На рис.43 наведено ілюстрацію роботи музичного плеєра на основі емоцій у вигляді діаграми послідовності та діаграми активності [14].

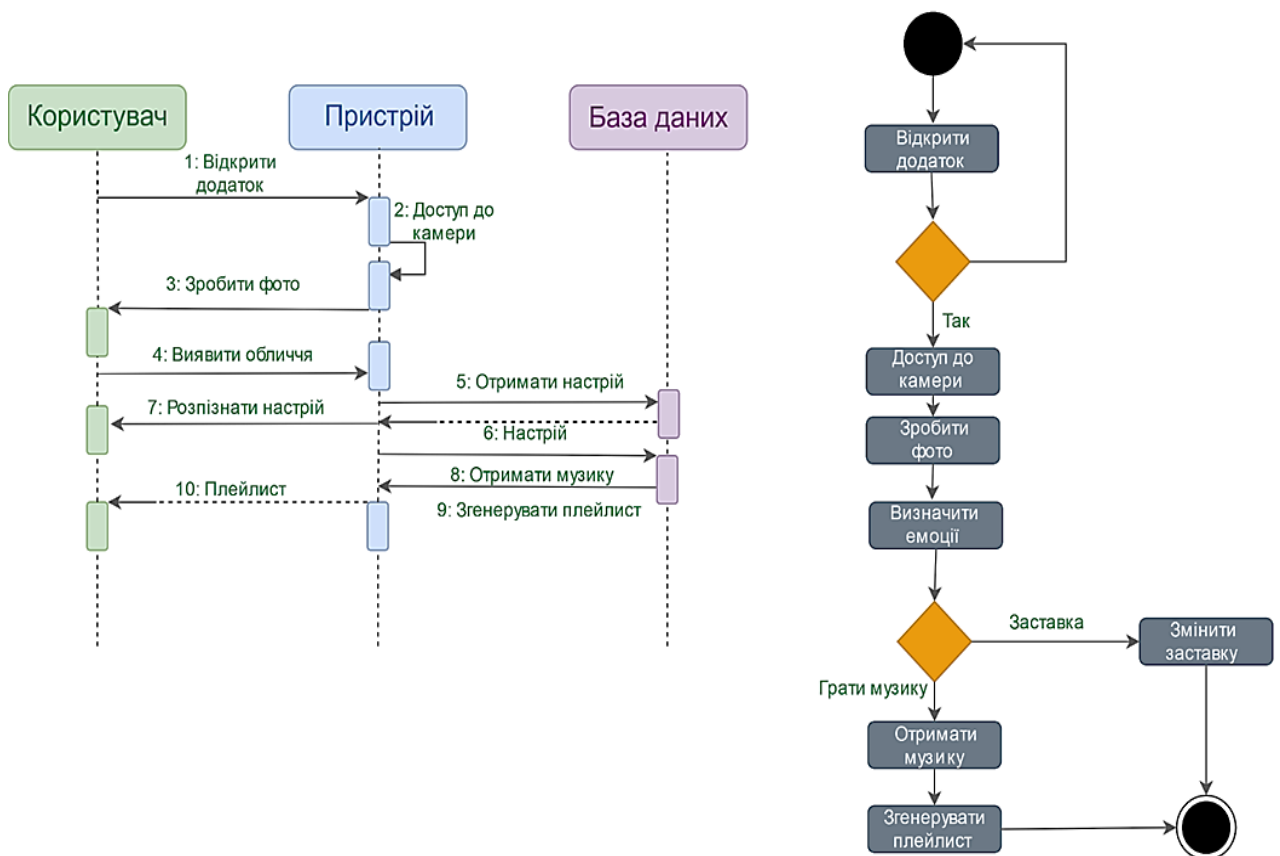
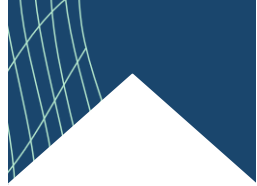


Рисунок 43 – Різниця між діаграмою послідовності та діаграмою діяльності [14]

### 1.2.6 Діаграма станів

Діаграма станів або діаграма кінцевих автоматів (State Diagrams, State Machine Diagrams) – це UML-діаграма, яка використовується для представлення стану програмної системи або компоненти, модуля ПЗ, будь-якої системи у кінцеві моменти часу [23]. Вона є корисним способом візуалізації різних станів об'єктів (сутностей), які існують у процесі реалізації ПЗ.



Діаграми станів можуть допомогти розробникам ПЗ визначити непотрібні кроки у процесі роботи з ПЗ та оптимізувати ці процеси для покращення взаємодії з клієнтами. Це поведінкова діаграма, яка представляє поведінку за допомогою кінцевих переходів між станами [24]. На діаграмах станів зображують різні стани об'єкта під час його існування і стимули (тригери), які призводять до переходу об'єкта від одного стану до іншого. Об'єкт по-різному реагує на ту саму подію залежно від того, в якому стані він перебуває [25].

Наприклад, об'єкт «Мережевий сервер» може перебувати у одному з таких станів протягом робочого процесу:

- готовий;
- очікує завдання;
- робочий стан;
- зупинка,

а подіями, які можуть спричинити зміну стану сервера можуть бути:

- створення завдання клієнтом;
- сервер отримує повідомлення «очікувати завдання»;
- клієнт надсилає запит на з'єднання з мережею;
- клієнт перериває запит;
- запит виконано / перервано;
- сервер отримує повідомлення про зупинку роботи з клієнтом /з завданням тощо.

Діаграми станів будуються з трьох типів фігур - станів, псевдостанів і переходів [26].

Стан належить лише одному класу (об'єкту) і відповідає переліку значень атрибутів, які може приймати клас.

Існує два особливих типи станів: початок і кінець. Їх особливість полягає у тому, що не існує жодної події, яка може спричинити повернення об'єкта до його початкового стану, так само, не існує жодної події, яка б могла повернути об'єкт зі стану кінця, тільки-но він його досягне.

Основні елементи діаграми станів надано на рис. 44. На рис. 45 діаграма показує різні стани, в яких існує підсистема перевірки користувача ПЗ [23].

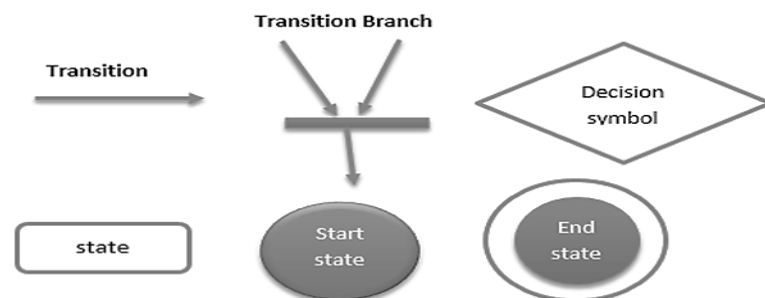


Рисунок 44 – Елементи діаграми станів [27]

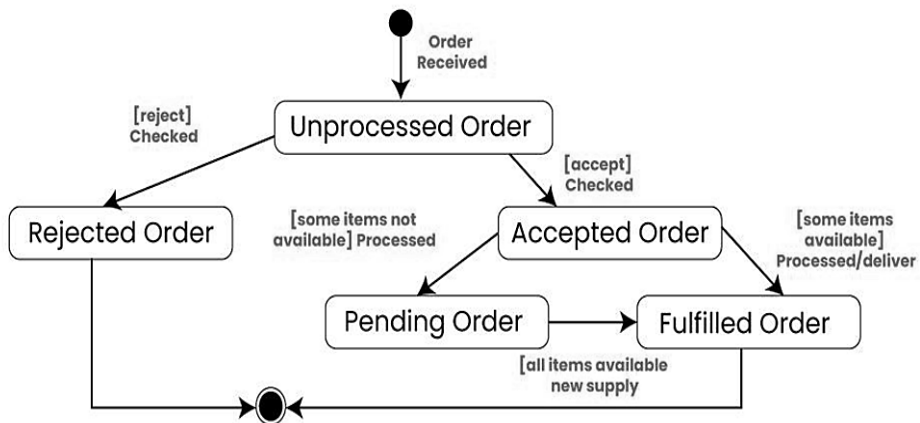
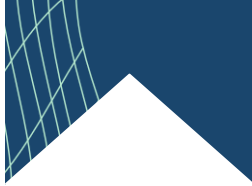


Рисунок 45 – Діаграма кінцевого автомата для підсистеми перевірки користувача ПЗ [23].

На рисунках 46 та 47 надано приклади використання діаграми станів для рахунку фактури та для об'єкту myBkCopy з класу BookCopy [25]. На рис. 48 показана діаграма станів документа – спеціальної форми, яка заповнюється при прийомі нового співробітника на роботу.

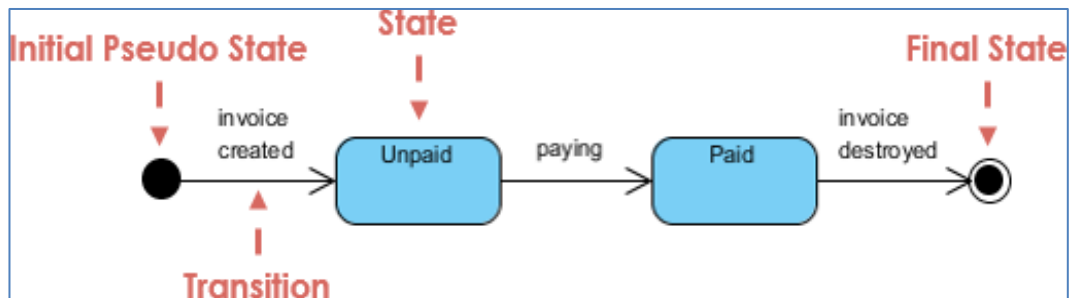


Рисунок 46 – Діаграма стану для рахунку-фактури

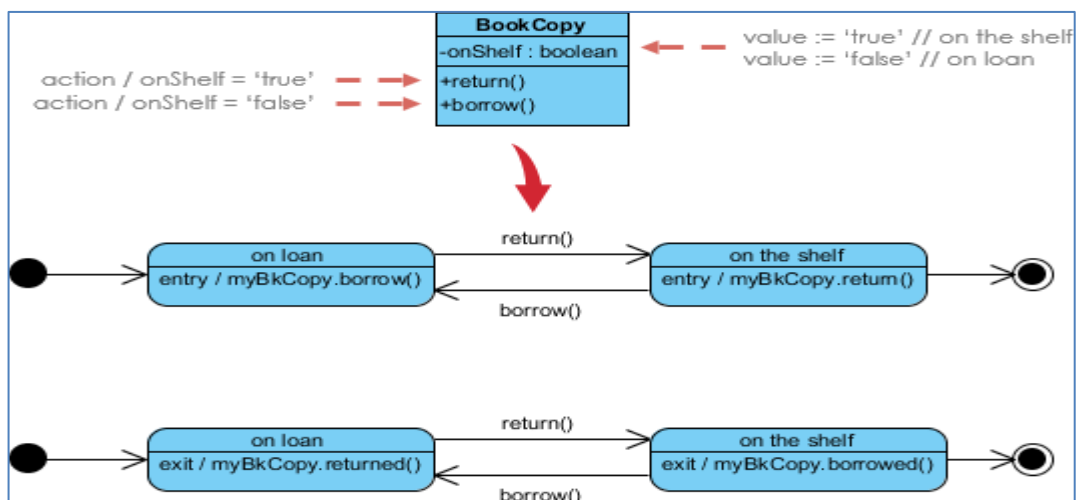


Рисунок 47 – Діаграма стану об'єкта myBkCopy з класу BookCopy

Статуси документа («не заповнена», «погоджена з HR-менеджером», «погоджена з інспектором ВК» тощо) відбивають дії, що їх генерують певні дійові особи: HR-менеджер, інспектор ВК тощо. Таким чином, діаграма станів прямо пов'язана з діаграмою прецедентів та з діаграмою діяльності, на яких показані «дії» виконавців. Тобто стан документа відображається за допомогою відповідних статусів.

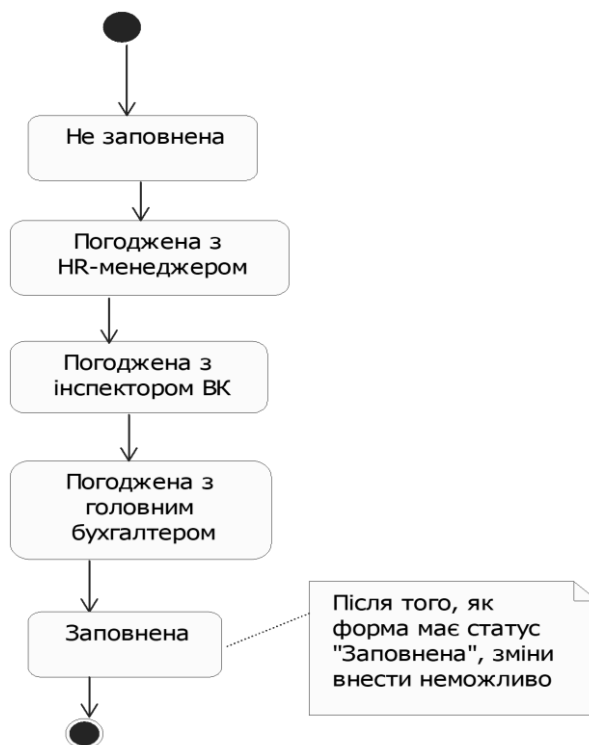


Рисунок 48 – Діаграма станів документу для нового співробітника.


Висновок: діаграми варіантів використання, діяльності, послідовності та кінцевих автоматів вважаються діаграмами поведінки, оскільки вони описують, що має статися у програмній системі (ПЗ, у компоненті, модулі ПЗ), що моделюється.

### 1.2.7 Зміст звіту з виконання ІДЗ № 1

Після виконання завдань ІДЗ №1 студентом складається звіт за результатами.

Звіт повинен містити:

- 1) титульний аркуш (приклад у додатку А);
- 2) мета та завдання індивідуального завдання №1;
- 3) опис проблеми, яка буде вирішена за допомогою ПЗ;
- 4) представити Mind Map 5W для ПЗ, що розроблюється (або Mind Map для проблеми, яка буде вирішена в межах дипломної роботи),



вказати, який інструмент було використано для її побудови, надати опис карти з обов'язковим формулюванням бізнес-вимог до ПЗ;

5) описати користувачів ПЗ та основні функціональні вимоги, представити їх у вигляді діаграми варіантів використання;

6) представити діаграми діяльності, послідовності та кінцевих автоматів, які описують сценарії, зміни станів, дії для реалізації відповідних варіантів використання користувача ПЗ (такі діаграми розроблюються для складних варіантів використання за вибором студента), надати пояснення побудованих діаграм;

7) зробити висновки за результатами виконання ІДЗ.

### 1.3 Методичні рекомендації до виконання ІДЗ № 2

#### 1.3.1 Завдання до ІДЗ № 2 та етапи виконання

Мета: вивчення процесу створення Story Mapping та отримання практичних навичок щодо розробки Story Mapping у системах моделювання та проектування, наприклад,

Завдання до виконання ІДЗ № 2: розробити User Story та User Story Mapping згідно розроблених вимог до програмного забезпечення, яке проектується в межах дипломної роботи, використати сервіс для планування процесу розробки ПЗ, наприклад:

- Miro (<https://miro.com/app/>),
- Figma (<https://www.figma.com/templates/user-story-mapping/>),
- Jira (<https://marketplace.atlassian.com/apps/1224357/story-maps-for-jira-free?tab=overview&hosting=cloud>),
- Confluence (<https://marketplace.atlassian.com/apps/1224383/agile-user-story-mapping-for-confluence?tab=overview&hosting=cloud>).

Завдання виконується для ПЗ, яке необхідно розробити в межах дипломної роботи (моделювання виявлення вимог було надано у ІДЗ № 1).

Етапи виконання індивідуального завдання №2.

1. Вивчити теоретичний матеріал щодо створення User Story та User Story Mapping для ілюстрації вимог до ПЗ за гнучкими методологіями.

2. Розробити User Story та User Story Mapping згідно розроблених вимог до програмного забезпечення, яке проектується в межах дипломної роботи, за допомогою сервісів для моделювання, наприклад Miro, Jira, Confluence та ін. Історій користувачів не повино бути менше п'яти, вони формуються з огляду на варіанти використання, які було сформовано у ІДЗ № 1.

3. Скласти звіт, включаючи скріншоти екранних форм з результатами роботи у відповідному сервісі.



### 1.3.2 Формування User Story як високорівневі вимоги до ПЗ

Історія користувача – це гнучка техніка, яка використовується для отримання опису функції програмного забезпечення з точки зору кінцевого користувача. Це не написання довгих складних технічних характеристик, а фактичне вираження програмного забезпечення як проблем, які треба вирішити для користувачів ПЗ або замовника ПЗ [28].

Історія користувача – це неформальне загальне пояснення функції програмного забезпечення, написане з точки зору кінцевого користувача. Його мета полягає в тому, щоб сформулювати, як функція програмного забезпечення надасть цінність клієнту [29].

Історії користувачів є будівельними блоками більших гнучких фреймворків, таких як епопеї та ініціативи.

Епопеї – це великі робочі елементи, розбиті на набір історій, а кілька епосів складають ініціативу (проект, частину проекту тощо) (рис. 49).

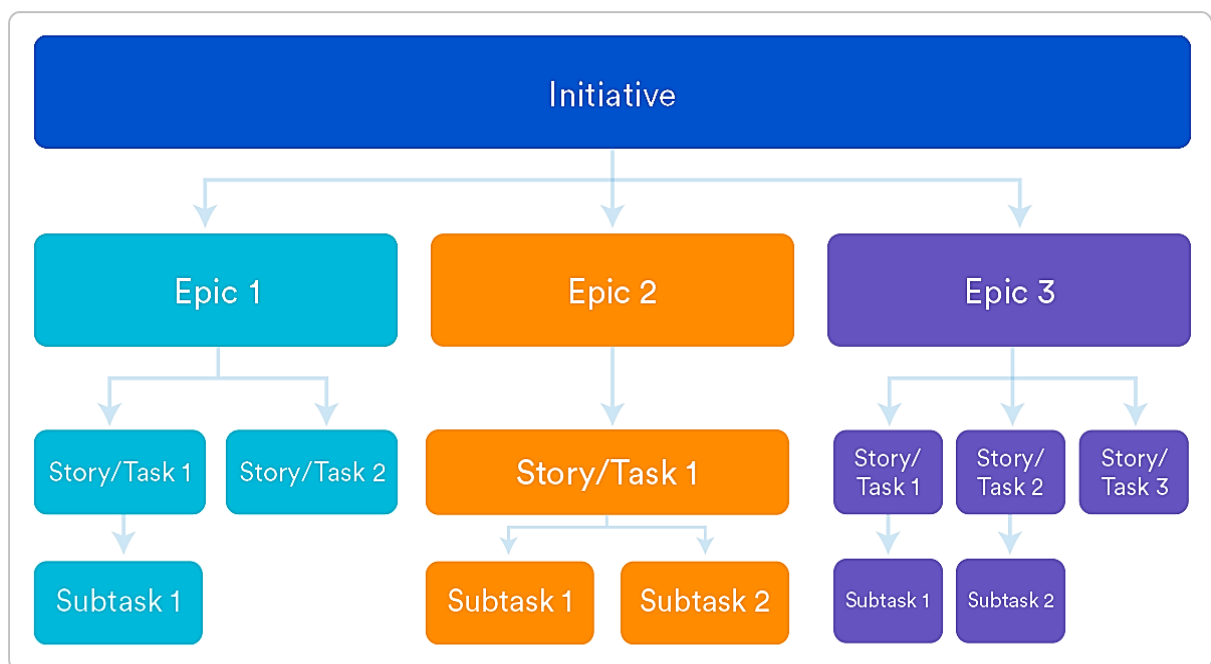


Рисунок 49 – Ієрархія Initiative, User Story, Epics, Tasks, Subtasks [29]

Історії користувачів зазвичай мають простий шаблон: «Як <тип користувача>, я хочу <якусь мету, намір - щось виконати, зробити?>, щоб <якась причина – чому?, для чого? > (рис.50).

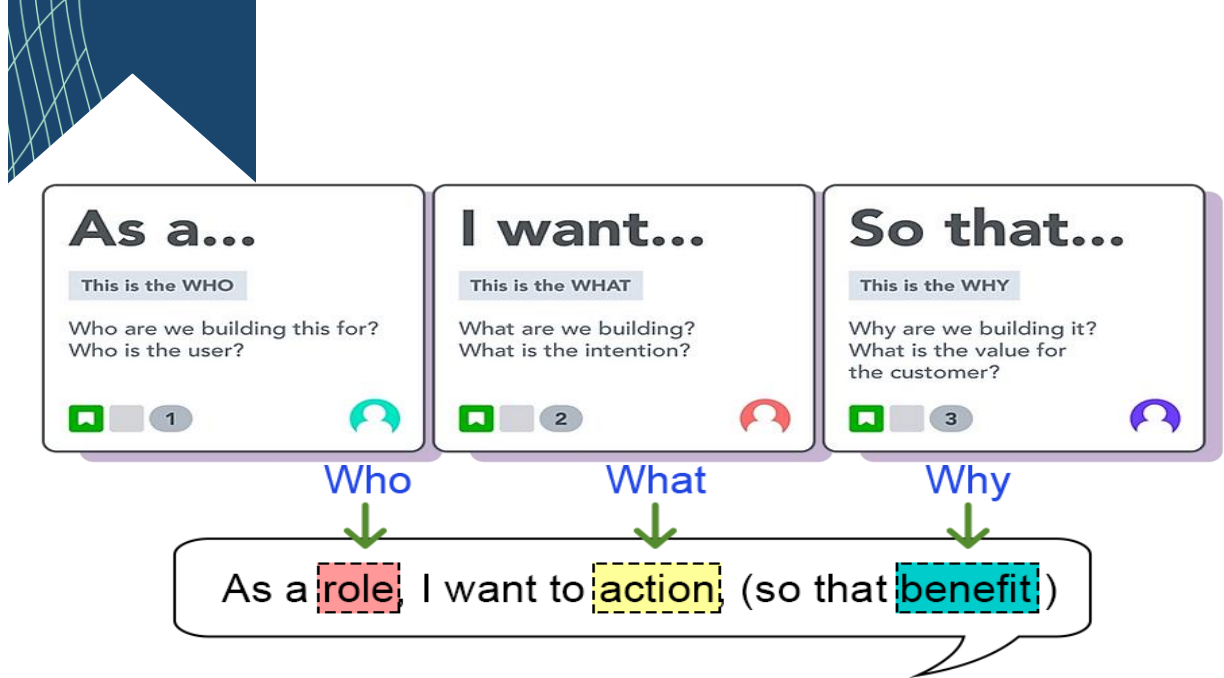


Рисунок 50 – Шаблон User Story [30, 31]

Приклади User Stories для проекту розробки модуля навчання (програми-асистента «Навчання») в межах системи безперервного удосконалення hr-сервісів (приклад опису проблеми надано у п.1.2.1, а діаграми варіантів використання на рис.17 та 18).

*«User-Stories для користувача «Співробітник компанії»:*

*Як співробітник компанії я хочу переглянути доступні тренінги та здійсню вибір, щоб взяти участь у тренінгу;*

*Як співробітник компанії я хочу створити заявку на участь у тренінгу, щоб зареєструватися ;*

*Як співробітник компанії я хочу отримувати інформаційні повідомлення про статус погодження моєї заявки, щоб відстежувати її статус та отримати повідомлення про затвердження;*

*Як співробітник компанії я хочу приєднатися до тренінгу щоб прийняти у ньому участь;*

*Як співробітник компанії я хочу отримати форму для надання зворотного зв'язку на відвіданий тренінг.*

*User-Stories для користувача «Працівник hr-відділу»:*

*Як працівник hr-відділу я хочу додати новий тренінг (видалити тренінг), щоб підтримувати актуальну інформації для навчання співробітників;*

*Як працівник hr-відділу я хочу надіслати учасникам тренінгу деталі щодо навчання, щоб підтримувати план проведення тренінгу;*

*Як працівник hr-відділу я хочу переглянути дані про проведені тренінги, щоб зробити звіт щодо виконання плану навчання співробітників;*

*Як працівник hr-відділу я хочу переглянути дані про загальну кількість реєстрацій, заявок та відгуків учасників, щоб сформувану рейтинг тренінгів.*

*User-Stories для користувача «Керівник»:*



*Як керівник я хочу розглянути заявку підлеглого, щоб прийняти рішення щодо прийняття її або відхилення».*

Деякі з цих історій користувачів, будуть епопеями. Пізніше епопеї будуть розкладені на менші історії, які легше вписуються в одну ітерацію. Крім того, будь-хто зі стейкхолдерів, залучених у процес розробки ПЗ, може писати нові історії та додавати їх до резервних копій продукту.

### 1.3.3 Критерії прийняття User Story

Історія користувача має відповідати принципам INVEST [31]:

**Independent** (незалежна): має бути самодостатньою у спосіб, який дозволяє її незалежно від інших історій випускати (тобто реалізувати у вигляді коду та потім продемонструвати результат);

**Negotiable** (підлягає обговоренню): охоплює лише суть потреб користувача, залишаючи місце для обговорення та аналізу;

**Valuable** (має цінність): забезпечує цінність для кінцевого користувача;

**Estimable** (має бути оцінена): історію користувача потрібно оцінити, щоб її можна було правильно розставити за пріоритетом у беклогу продукту та вписати до спринта;

**Small** (маленька): історія користувача — це невелика частка роботи з розробки ПЗ, яку можна виконати приблизно за 3-4 дні;

**Testable** (має бути перевірена): історія користувача має бути підтверджена за допомогою попередньо написаних критеріїв прийняття.

Критерії прийняття – це умови, які має виконати та прийняти користувач. Вони супроводжують кожну історію користувача, щоб підтвердити, що історія була правильно реалізована [28].


Критерії прийняття потрібні для того, щоб з'ясувати, що історії користувачів дозволять команді збалансувати потреби, бажання та цінності своїх клієнтів із реалізацією функціоналу ПЗ, використання якого забезпечить цю цінність.

Розглянемо приклад завершеної історії користувача з критеріями прийняття [30]:

*«Як потенційний учасник тренінгу, я хочу мати можливість зареєструватися на тренінг онлайн, щоб реєстрація була простою та без паперу».*

*Критерії прийняття:*

- *форма участі у тренінгу;*
- *користувач не може подати форму, не заповнивши усі обов'язкові поля (ім'я, прізвище, назва компанії, електронна адреса, посада, платіжна інформація);*
- *інформація з форми зберігається в реєстраційній базі;*
- *працює захист від спаму;*

- 
- оплату можна здійснити через PayPal, дебетову або кредитну картку;
  - після надсилання форми учаснику надсилається електронний лист із підтвердженням;

Після формування критеріїв команда повинна переконатися, що ці критерії прийняття враховують усі аспекти юзер стори, а саме:

- негативні сценарії функціональності;
- функціональні та нефункціональні варіанти використання;
- проблеми та рекомендації щодо продуктивності;
- які дії будуть реалізовані для функції;
- «End-to-user flow» - кінцевий потік дій до користувача;
- вплив історії користувача на інші функції;
- проблеми UX (User Experience)».

Отже, написання ефективних історій користувачів є цінною практикою, яка допоможе команді постачати програмне забезпечення, яке залишається актуальним для клієнтів (замовників).

Такі критерії прийняття треба студенту створити для сформованих User Story в межах виконання цього ІДЗ.

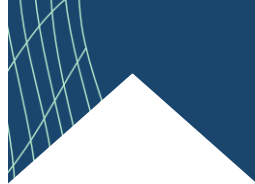
**Завдання:** створити критерії прийняття для усіх сформованих User Stories (див. завдання у п. 1.3.1). Вони формуються з огляду на діаграми активності, послідовності та станів, які було розроблено у ІДЗ № 1.

#### 1.3.4 Story Mapping як представлення загальної карти вимог до ПЗ

Історії користувачів — це корисний спосіб створити кращу версію ПЗ, яке орієнтовано на користувача та описує вимоги до ПЗ практичним і дієвим способом. Але історії користувачів самі по собі не розкривають повної картини призначення ПЗ [31].

Карта історій користувачів (Story Mapping або User Story Mapping) допомагає упорядкувати історії користувачів у керовану модель для планування, розуміння та систематизації функціональності ПЗ. Маніпулюючи структурою карти, можна виявити «прогалини та пропуски» функціоналу та інших аспектів у розробці ПЗ та взаємопов'язати історії користувачів у систему. Карта допомагає в ефективному плануванні випусків ПЗ (релізів), які приносять певну цінність користувачу з кожним релізом [31].

Картування історій користувачів змінює підхід команди до розуміння продукту, який вона розроблюватиме. Карта історій користувача візуалізує «подорож» клієнта (користувача) з продуктом, від початку (вхід до продукту) до кінця (вийти з продукту). Цей фокус на клієнтоорієнтованості гарантує, що будуть створені функції, які справді важливі для користувача (замовника) [32].



Для гнучких методологій розробки ПЗ відображення історій користувачів є альтернативою створенню переліку робіт або роботи з великими за обсягами документами вимог (технічні завдання на розробку, специфікації вимог тощо) [33].

На рис.51 надано приклад карти історій користувача системи керування документами.

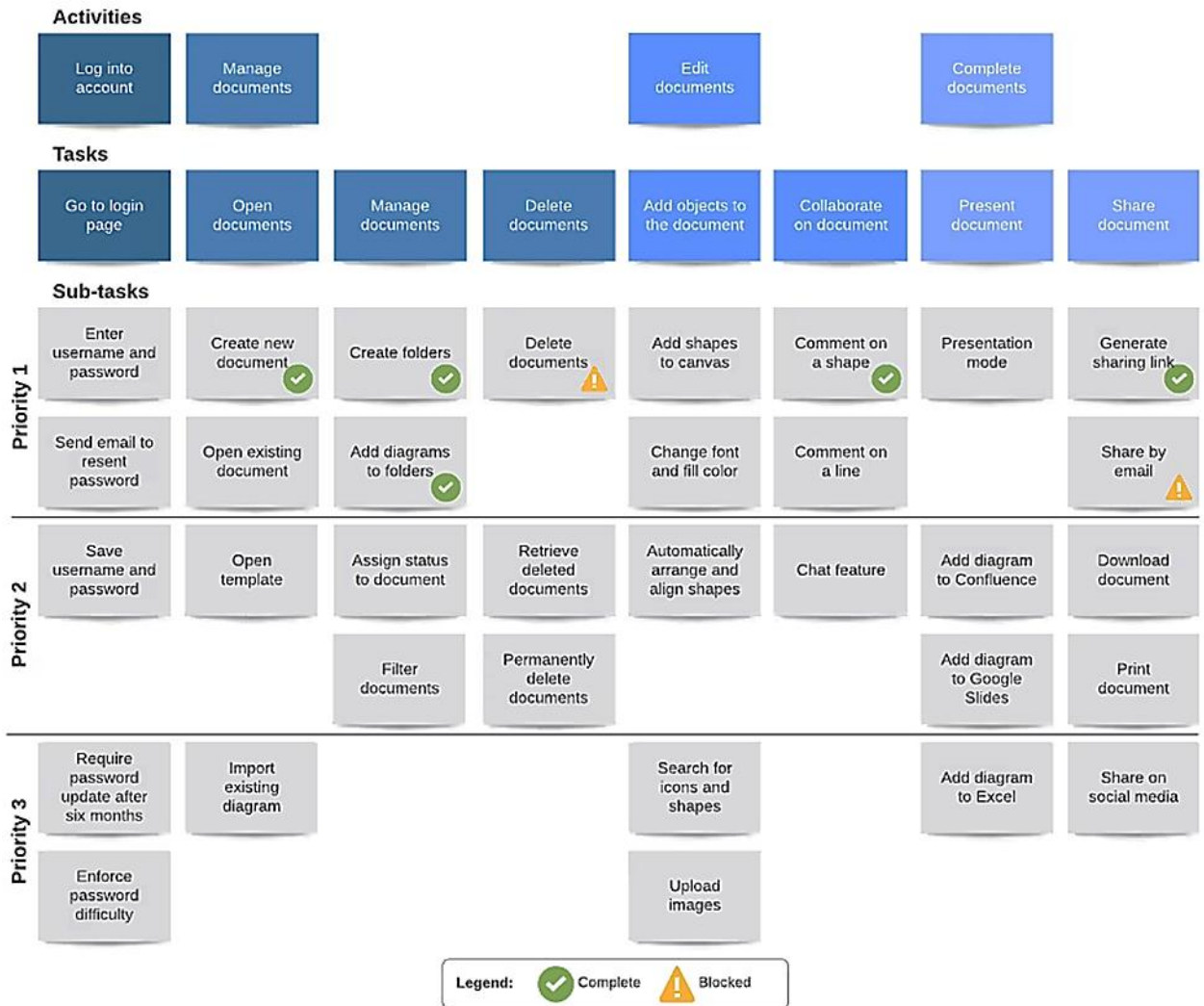


Рисунок 51 – Приклад User Story Mapping для системи керування документами [34]

Треба відмітити, що при створенні карти User Story записують у скороченому вигляді. Вважається, що попередньо було сформовано команду учасників створення карти (менеджери проєкту, власник продукту, дизайнери UX/UI, розробники, спеціалісти з маркетингу тощо). Встановлено чіткі цілі, тобто зрозуміло усім учасникам, яких ключових результатів має досягти продукт для користувача, зрозумілі бізнес-цілі замовника. Можливо, що учасники процесу сформували свої варіанти історій користувачів.



Розглянемо процес створення User Story Mapping:

1) визначити «широкі» дії користувача /користувачів, які здійснюватиме користувач при роботі з ПЗ. Дії (Activities) показують високорівневі цілі або функції ПЗ., наприклад «Увійти до облікового запису (Log into account)», «Керувати документами (Manage documents)», «Редагувати документи (Edit documents)» та «Завершити документи (Complete documents)» (рис.51). Тут часовий вимір може бути відносним, але можливо показати логічну послідовність дій користувача;

2) розбити дії користувача на кроки/завдання. Виконання цих завдань забезпечать реалізацію відповідних дій користувача. Зазвичай ці завдання представляються як епіки. Наприклад, у розділі «Керування документами (Manage documents)» є завдання «Відкрити документи (Open documents)», «Керувати документами (Manage documents)» та «Видалити документи (Delete documents)» (рис.51);

3) деталізувати кожне завдання (розписати епіки) – сформувані User Stories. Наприклад, для виконання завданням «Відкрити документи (Open documents)» реалізуються підзавдання, як як «Створити новий документ (Create new document)», «Відкрити наявний документ (Open existing document)», «Відкрити шаблон (Open template)» та «Імпортувати існуючу схему (Import existing diagram)»;

4) здійснити пріоритезацію завдань (User Stories) – розташувати їх по вертикалі у порядку зменшення їх пріоритету (зверху вниз цінність для користувача User Story (завдання) стає меншою). Цінність визначається через розмови з користувачами, завдяки аналітиці моделей (у т.ч. Use Case діаграми, діаграми процесів та ін.) або за іншими формами розуміння цінності завдань в залежності від особливостей ПЗ, що розроблюється;

5) визначення прогалін, залежностей, технічних вимог та альтернатив для кожної User Story. Це може сприяти виявленню відсутньої інформації, вимог, завдань тощо, підвищенню зручності використання ПЗ та ін;

6) визначення спринтів та релізів (випусків) ПЗ. Завдяки пріоритетам історій зверху вниз можна уявити обсяг завдань (User Stories), виконання яких принесе найбільшу цінність за найкоротший час. Карта історій користувача дозволяє команді спланувати випуск мінімально життєздатного продукту (a minimum viable product, MVP). Власник продукту та команда визначають, що слід включити до початкових і майбутніх випусків. Команда малює лінії на карті, щоб відокремити те, що входить до MVP, від того, що може чекати пізніше. Команда розміщує елементи над лінією для MVP і під лінією для майбутніх версій ПЗ. Коли вони вивчають і змінюють пріоритети, вони додають нові дії, кроки та деталі на карту для майбутніх спринтів і випусків [34] Отже, групуються User Stories в спринти розробки та випуски, як підсумок – створюється релізна стратегія ПЗ.



У процесі розробки ПЗ ця карта аналізується та змінюється.  
На рис. 52 показано структуру User Story Mapping [34], а на рис.53 надано приклад карти для модуля управління контактами.

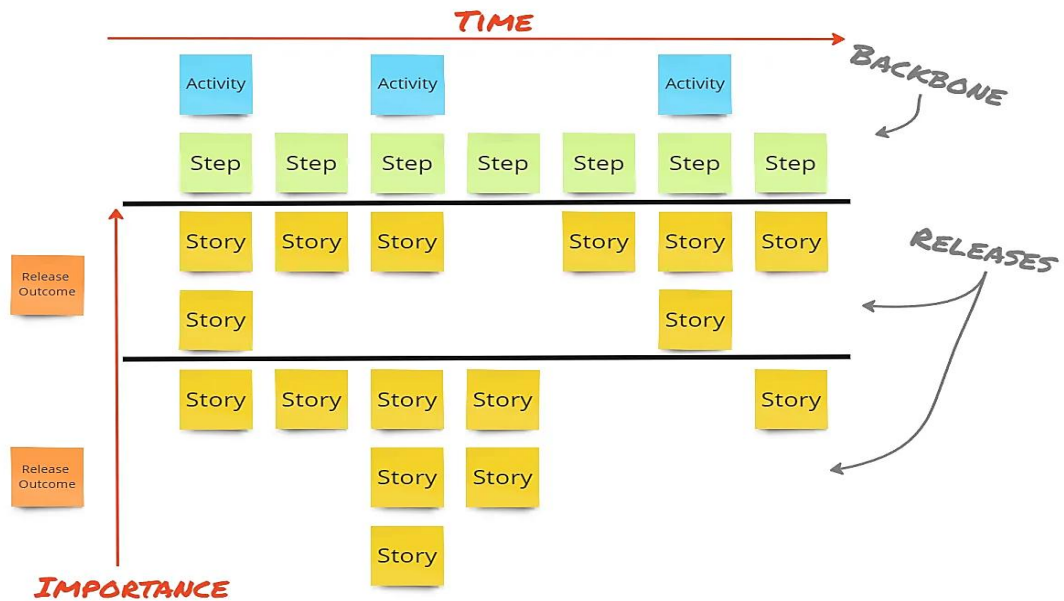


Рисунок 52 – Структура User Story Mapping [34]

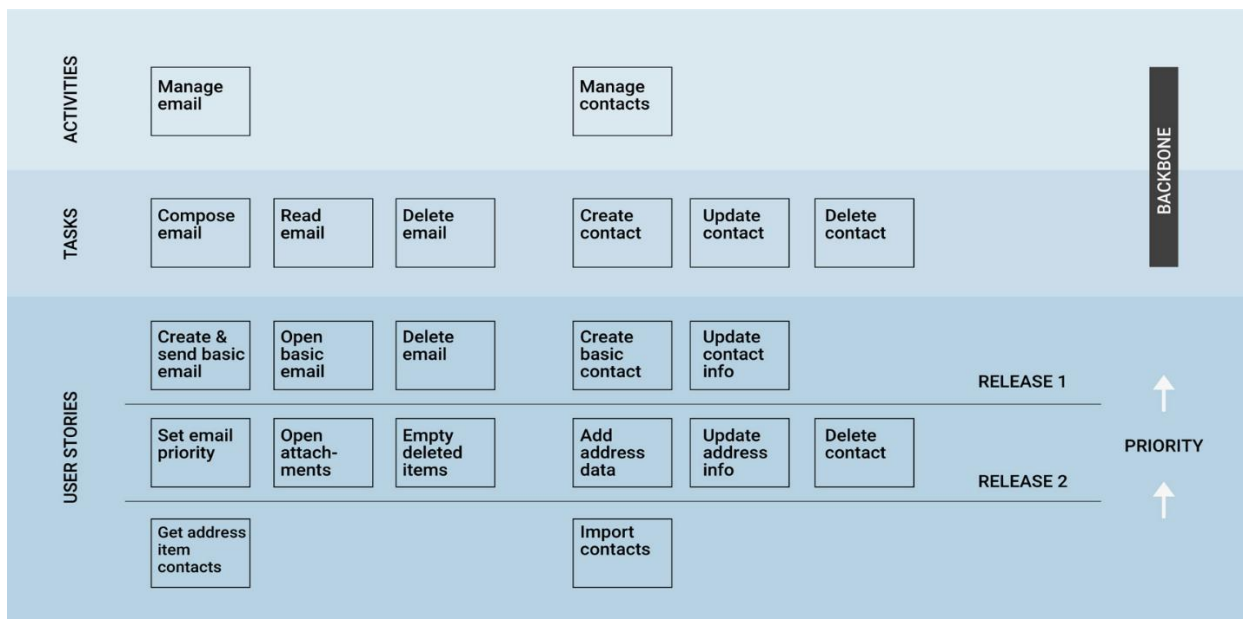


Рисунок 53 – Приклад User Story Mapping

Для розробки програми-асистента «Навчання» в межах системи безперервного удосконалення hr-сервісів (приклад опису проблеми надано у п.1.2.1, а діаграми варіантів використання на рис.17 та 18) створено User Story Mapping на основі попередньо сформованих історій користувача, приклади яких представлено у п.1.3.1. На рис.54,



рис.55 та рис.56 наведені фрагменти User Story Mapping для таких користувачів відповідно: працівника компанії як потенційного учасника тренінгу, працівника hr-відділу та керівника підрозділу, в якому працює потенційний учасник тренінгу.

На рис. 56 надано User Story Mapping з розбивкою на релізи для проекту розробки програми-асистента «Навчання»

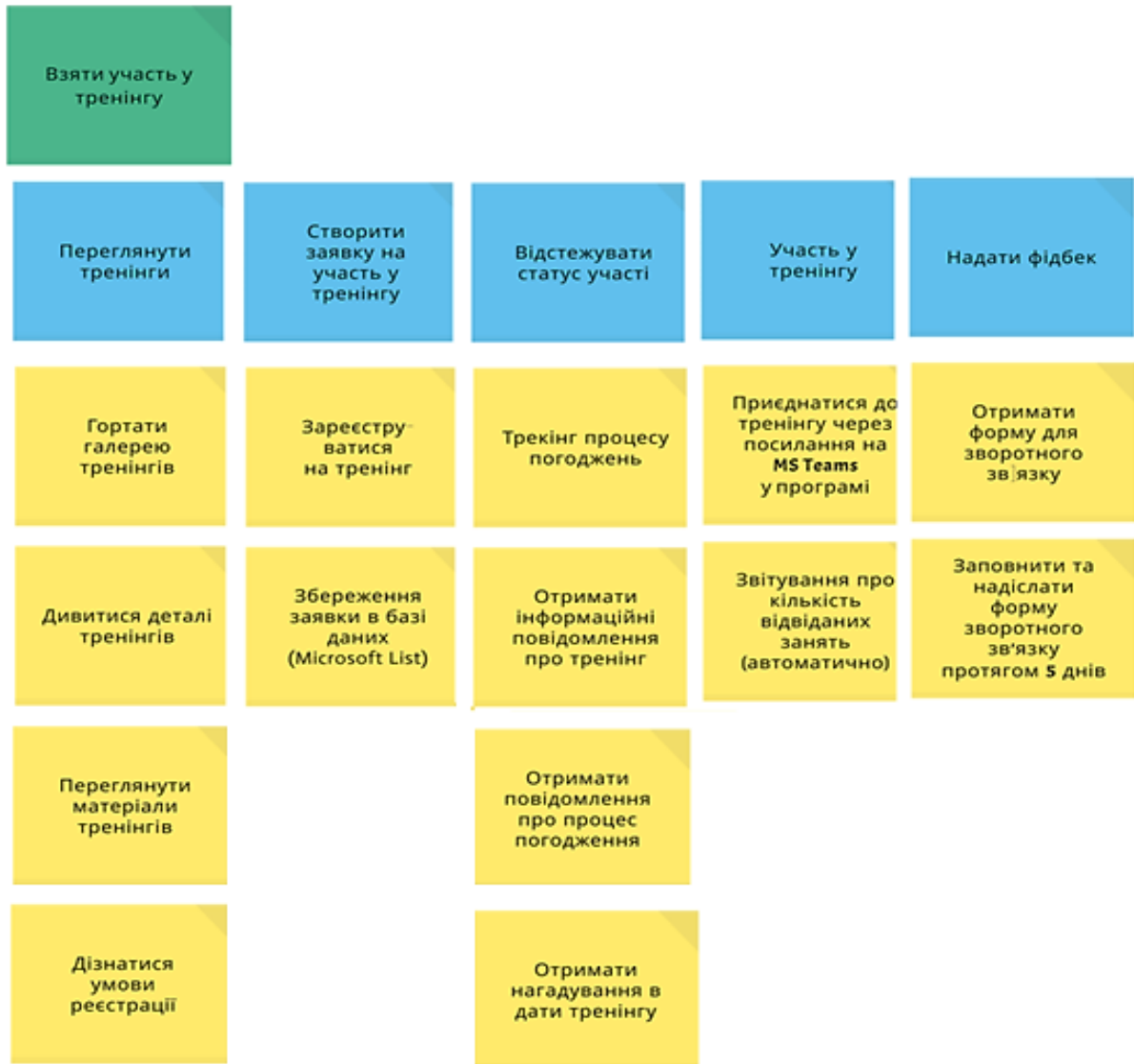


Рисунок 54 – Приклад User Story Mapping для потенційного учасника тренінгу як користувача програми-асистента «Навчання» системи удосконалення hr-сервісів

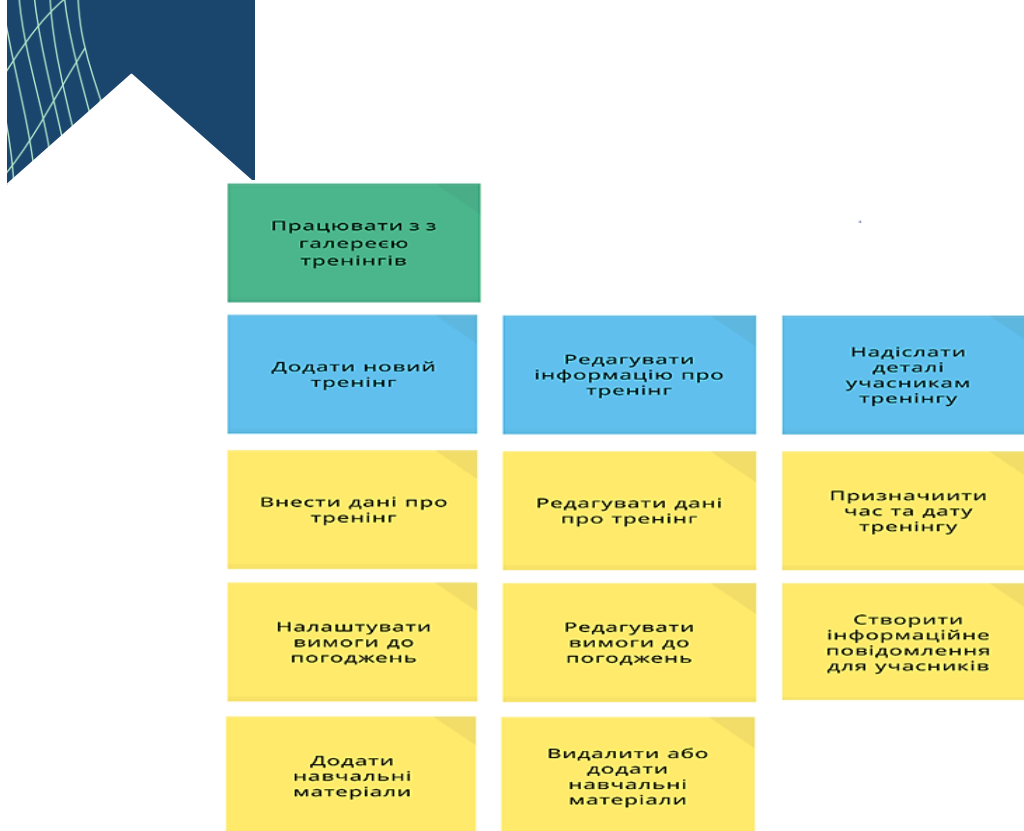


Рисунок 54 – Приклад User Story Mapping для працівника hr-відділу як користувача програми-асистента «Навчання» системи удосконалення hr-сервісів



Рисунок 55 – Приклад User Story Mapping для керівника підрозділу як користувача програми-асистента «Навчання» системи удосконалення hr-сервісів

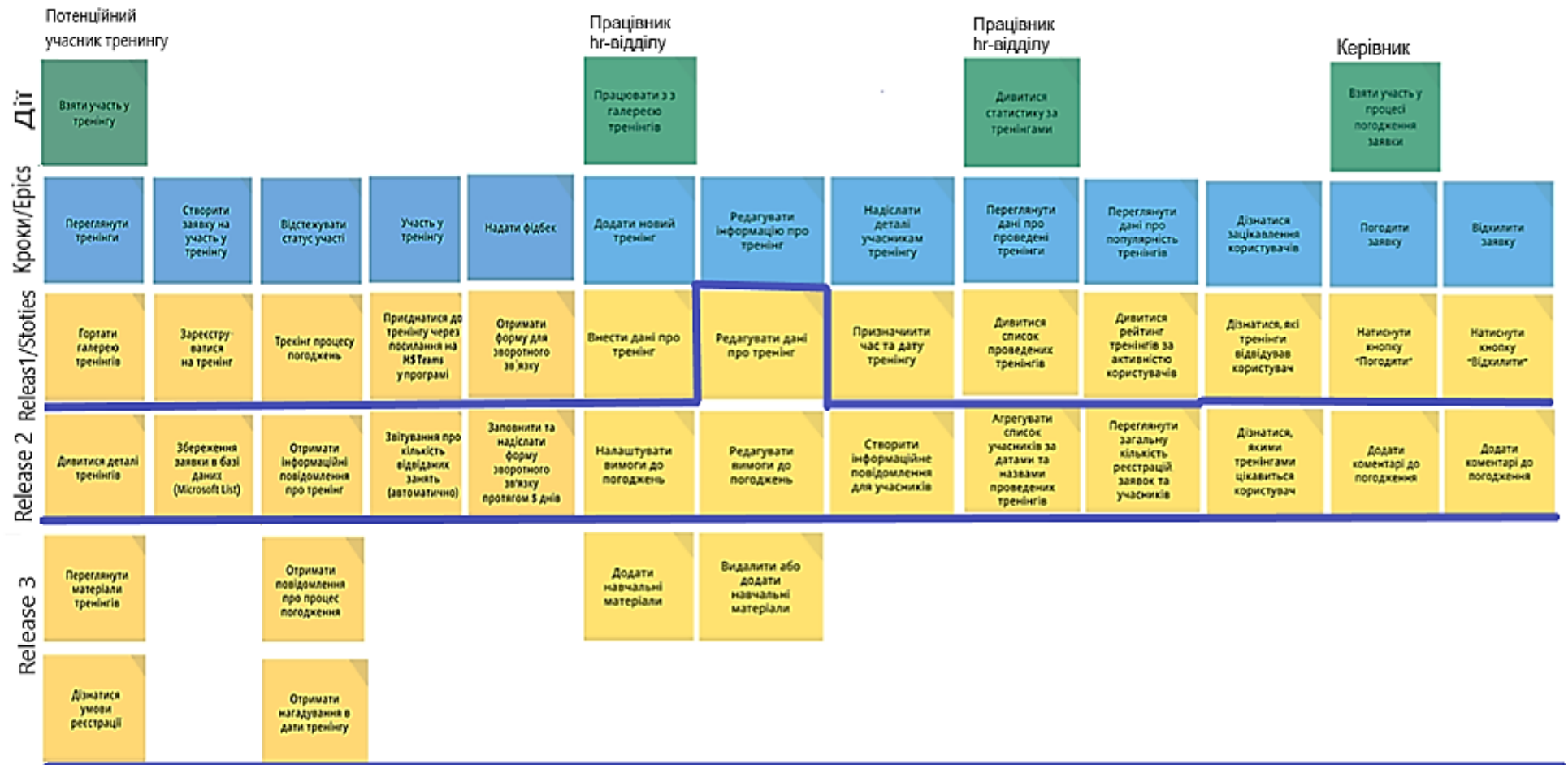


Рисунок 56 – Приклад User Story Mapping з розбивкою на релізи для проекту розробки програми-асистента «Навчання»

Використання програмних інструментів може значно покращити процес картографування історій користувача. На рис.57 наведені п'ять інструментів, які виділяються своєю здатністю спрощувати та оптимізувати відображення історій користувачів.



Рисунок 57 – Інструменти для побудови User Story Mapping [34]

### 1.3.5 Зміст звіту з виконання ІДЗ №2

Звіт з виконання ІДЗ №2 повинен містити:

- 8) титульний аркуш (приклад у додатку А);
- 9) мета та завдання ІДЗ № 2;
- 10) опис проблеми, яка буде вирішена за допомогою ПЗ;
- 11) опис User Story для кожного користувача ПЗ, що розроблюється (у загальній кількості не менше п'яти) та критерії їх прийняття (можна надати скріни з сформованими історіями у сервісу, де здійснювалась створення карти історій користувачів);
- 12) опис кроків створення User Story Mapping, надати скріни карти у обраному сервісі;
- 13) представити обґрунтування історій користувачів, які увійдуть до першого релізу – розробка MVP;
- 14) зробити висновки щодо отриманих результатів.


### 1.4 Оформлення, подання і захист індивідуального завдання

Звіт щодо виконання ІДЗ може бути подано на розгляд викладачу у вигляді текстового файлу у форматі *.docx* і прикріплено до відповідної активності у системі Moodle у форматі *.pdf*.

Звіт виконують державною мовою, науковим стилем.

Вимоги до тексту звіту: формату А4 (210x297мм), шрифт Arial, 14 кегль, інтервал – 1,5; береги: верхній, нижній – 2 см, правий – 1,5 см; лівий – 3 см, абзац – 1,27.

Посилання у тексті на джерела зазначають порядковим номером за переліком посилань у квадратних дужках, наприклад, «... у роботі [1] ...». Оформлення списку використаних джерел здійснюється за



Національним стандартом України «Інформація та документація. Бібліографічна посилання. Загальні положення та правила складання. ДСТУ 8302:2015» .

Захист ІДЗ відбувається на практичному занятті згідно з графіком контрольних точок, передбаченим робочою програмою дисципліни. Оцінка за виконання ІДЗ виставляється викладачем у відповідній активності у системі Moodle, враховуючи поточну успішність студента.

### 1.5 Критерії оцінювання результатів виконання ІДЗ

Максимальна кількість балів, яку здобувач може отримати за кожне виконане ІДЗ – 20 балів. Враховується виконання ІДЗ (максимальний бал 15) та захист його на практичному занятті (максимальний бал 5). Оскарження оцінки може бути здійснене на останньому практичному занятті модуля.

Критерії оцінювання:

- студент підготував роботу відповідно до поставленого завдання, в якій: правильно визначив проблеми, комплекс факторів, які могли вплинути на їх виникнення, обґрунтував своє бачення теоретичними концепціями або моделями, виконав необхідне проектування у разі потреби, представив висновок або власне бачення вирішення проблеми і окреслив можливі перспективи, обмеженість такого рішення, навів за необхідності методичку і результати; робота структурована, викладена діловим, науковим або публіцистичним стилем (5 балів);
- робота містить комплексну, логічну і оригінальну пропозицію проблематики індивідуального завдання аж до міждисциплінарного підходу; використання штучного інтелекту (ШІ) не забороняється, оскільки пропозиції відомих застосунків ШІ суттєво залежать від обміркованої постановки питання і уточнюючих питань; однак в разі, якщо відповідь, отримана з використанням ШІ, не є комплексною або не відповідає за стилем і викладеними позиціями іншим частинам роботи або завдання, містить очевидно неправдиву інформацію, то оцінка за цим критерієм знижується (5 балів);
- студент під час презентації / захисту свого індивідуального завдання та контрольної роботи у вигляді есе демонструє володіння термінологічним апаратом, методами бізнес-аналізу, відповідає на запитання, здатний швидко адаптувати моделі, дизайни, документи під зміни у завданнях (3 бали);
- студент дав пряму і релевантну відповідь на поставлене питання з використанням обґрунтованого посилання на теоретичний матеріал демонструє володіння термінологічним апаратом, методами бізнес-аналізу, здатний адаптувати моделі під зміни завдання, у т.ч. у вигляді додаткових запитань / зміг стисло формалізувати вербально



сутність задачі бізнес-аналізу, визначити ключові складові виконання практичної роботи, критерії якості застосованої моделі, методу бізнес-аналізу (2 бали).

Додаткові зауваження:

- студент може оскаржити отримані оцінки в порядку, передбаченому Положенням про організацію освітнього процесу (Нормативні документи : Polytechnic (metinvest.university)) та Положенням про політику та процедури врегулювання конфліктних ситуацій (Академічні політики : Polytechnic (metinvest.university));

- викладач не має права знижувати оцінку за індивідуальне завдання, якщо воно не було складено вчасно, однак в разі, якщо така робота була оцінена пізніше, ніж момент завершення теоретичного навчання у семестрі, то відповідна оцінка не враховується у рейтингу здобувачів освіти.




## СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Робоча програма навчальної дисципліни «Стандарти та методології бізнес-аналізу» / уклад. В. В. Москаленко. Запоріжжя : ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА», 2024. 13 с.
2. Положення про організацію освітнього процесу у ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА». URL: <https://metinvest.university/data/file/14/86/148623526b844f999bad7c51257baa51.pdf>.
3. Положення про визнання у ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» результатів навчання, набутих у неформальній/інформальній освіті. URL: <https://metinvest.university/data/file/eb/ef/ebef5aa7e673444797bf73d46e33615b.pdf>.
4. Положення про академічну доброчесність здобувачів вищої освіти та науково-педагогічних працівників ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА». URL: <https://metinvest.university/data/file/c1/c9/c1c998364cec4bdbb42478109c72e17c.pdf>.
5. A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide). International Institute of Business Analysis, Toronto, Ontario, Canada. Version 3.0. URL: <https://www.iiba.org/career-resources/a-business-analysis-professionalsfoundationforsuccess/babok/>.
6. Using Mind Maps to Create Visual Requirements Documents : веб-сайт. URL: <https://aimconsulting.com/insights/using-mind-maps-to-create-visual-requirements-document/> (дата звернення: 27.02.2025).
7. Mind Map Guide: How to Make a Mind Map : веб-сайт. URL: <https://www.lucidchart.com/pages/tutorial/how-to-make-a-mind-map> (дата звернення: 27.02.2025).
8. Business Mind Map Complete Guide With 30+ Examples : веб-сайт. URL: <https://www.edrawmind.com/article/business-mind-map-examples.html> (дата звернення: 27.02.2025).
9. Mind Map Example: System Requirements : веб-сайт. URL: <https://online.visual-paradigm.com/diagrams/templates/mind-map-diagram/mind-map-example-system-requirements/> (дата звернення: 27.02.2025).
10. Mind Mapping Tools : веб-сайт. URL: <https://www.visual-paradigm.com/features/mind-mapping-diagram-and-tools/> (дата звернення: 27.02.2025).
11. Top 6 Practical Mind Map for Business Plan : веб-сайт. URL: <https://www.edrawsoft.com/mindmaster/6-business-uses-mindmap.html> (дата звернення: 27.02.2025).



12. Agile Modeling with Mind Map and UML : веб-сайт. URL: <https://www.stickyminds.com/article/agile-modeling-mind-map-and-uml> (дата звернення: 27.02.2025).
13. Quillian J. Mind Maps for Business Analysis : веб-сайт. URL: <https://www.batimes.com/articles/mind-maps-for-business-analysis/> (дата звернення: 27.02.2025).
14. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти : веб-сайт. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 27.02.2025).
15. Use Case Relationship Types: A Comprehensive Guide for Effective System Design : веб-сайт. URL: <https://guides.visual-paradigm.com/use-case-relationship-types-a-comprehensive-guide-for-effective-system-design/> (дата звернення: 27.02.2025).
16. Activity Diagrams – Unified Modeling Language (UML) : веб-сайт. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/> (дата звернення: 27.02.2025).
17. What is Activity Diagram? : веб-сайт. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> (дата звернення: 27.02.2025).
18. UML Activity Diagram Tutorial : веб-сайт. URL: <https://www.lucidchart.com/pages/uml-activity-diagram> (дата звернення: 27.02.2025).
19. UML Sequence Diagram Tutorial : веб-сайт. URL: <https://www.lucidchart.com/pages/uml-sequence-diagram> (дата звернення: 27.02.2025).
20. Create a sequence diagram : веб-сайт. URL: <https://www.drawio.com/blog/sequence-diagrams> (дата звернення: 27.02.2025).
21. What is Sequence Diagram? : веб-сайт. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> (дата звернення: 27.02.2025).
22. Sequence Diagrams – Unified Modeling Language (UML) : веб-сайт. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/> (дата звернення: 27.02.2025).
23. State Machine Diagrams | Unified Modeling Language. (UML) : веб-сайт. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/> (дата звернення: 27.02.2025).
24. How to Draw a State Machine Diagram in UML : веб-сайт. URL: <https://www.lucidchart.com/pages/how-to-draw-a-state-machine-diagram-in-uml> (дата звернення: 27.02.2025).
25. What is State Machine Diagram? : веб-сайт. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/> (дата звернення: 27.02.2025).

- 
26. Draw a UML state machine diagram : веб-сайт. URL: <https://www.drawio.com/blog/uml-state-diagrams> (дата звернення: 27.02.2025).
  27. State Transition Diagram with example in software engineering : веб-сайт. URL: <https://t4tutorials.com/what-is-state-transition-diagram-software-engineering/> (дата звернення: 27.02.2025).
  28. Історії користувачів і критерії прийняття — хтось досі ними користується? : веб-сайт. URL: <https://afiya-chohollo.medium.com/user-stories-acceptance-criteria-is-anyone-still-using-them-1d947a003e7> (дата звернення: 27.02.2025).
  29. What are agile user stories? : веб-сайт. URL: <https://www.atlassian.com/agile/project-management/user-stories#:~:text=What%20are%20agile%20user%20stories,the%20end%20user%20or%20customer> (дата звернення: 27.02.2025).
  30. How to Write User Stories in Agile Software Development : веб-сайт. URL: <https://www.easyagile.com/blog/how-to-write-good-user-stories-in-agile-software-development> (дата звернення: 27.02.2025).
  31. What is User Story? : веб-сайт. URL: <https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/> (дата звернення: 27.02.2025).
  32. Mastering User Story Mapping for Customer-Centric Product Development : веб-сайт. URL: <https://www.easyagile.com/blog/user-story-mapping-customer-centric-product-development> (дата звернення: 27.02.2025).
  33. User story mapping: Practical templates and examples : веб-сайт. URL: <https://www.aha.io/roadmapping/guide/release-management/what-is-user-story-mapping> (дата звернення: 27.02.2025).
  34. User Story Mapping in Design : веб-сайт. URL: <https://www.interaction-design.org/literature/article/user-story-mapping-in-design> (дата звернення: 27.02.2025).
  35. Enhancing product clarity: the why, how, and what of User Story Mapping : веб-сайт. URL: <https://starkephillip.com/user-story-mapping/> (дата звернення: 27.02.2025).

Додаток А. Приклад оформлення титульного аркушу

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«МЕТІНВЕСТ ПОЛІТЕХНІКА»  
Факультет автоматизації виробництва  
та цифрових технологій  
Кафедра цифрових технологій  
та проєктно-аналітичних рішень

**ЗВІТ  
З ІНДИВІДУАЛЬНОГО ЗАВДАННЯ №1**

з дисципліни: «СТАНДАРТИ ТА МЕТОДОЛОГІЇ  
БІЗНЕС-АНАЛІЗУ»

на тему: «Методи бізнес-аналізу Mind Map, Use Case  
diagram, Activity diagram, Sequence diagram як  
інструменти формування вимог до ПЗ»

Роботу виконав

Студент групи 122-22-1п  
Микола КОВАЛЬ

Роботу перевірів

Валентина МОСКАЛЕНКО

ЗАПОРІЖЖЯ 2025

**mp** metinvest  
polytechnic



Навчально-методичне видання

Валентина Володимирівна Москаленко

СТАНДАРТИ ТА МЕТОДОЛОГІЇ БІЗНЕС-АНАЛІЗУ

методичні рекомендації  
до виконання індивідуальних завдань

Самостійне електронне мережеве видання

Публікується в авторській редакції