

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»

**ЗАВДАННЯ**

до лабораторних робіт  
з дисципліни

**“Аналіз вимог до програмного забезпечення.  
Робота в Jira Software ”**

для студентів  
спеціальності 121  
“Інженерія програмного забезпечення”  
всіх форм навчання

2023

Завдання до лабораторних робіт з дисципліни “Аналіз вимог до програмного забезпечення. Робота в Jira Software ” для студентів спеціальності 121 “Інженерія програмного забезпечення” всіх форм навчання / Т. І. Каплієнко, Ж. К. Камінська. – Запоріжжя: Національний університет «Запорізька політехніка», 2023. – 41 с.

Автори: Т. І. Каплієнко, к.т.н., доцент  
Ж. К. Камінська, асистент

Рецензент: Г. В. Табунщик, к.т.н., професор

Відповідальний  
за випуск: Т. І. Каплієнко, к.т.н., доцент

Затверджено  
на засіданні кафедри  
програмних засобів

Протокол №12  
від “09” червня 2023 р.

**ЗМІСТ**

Вступ.....	4
Лабораторна робота №1 Робота з вимогами та документами.....	5
Лабораторна робота №2 Діаграми бізнес процесів.....	21
Лабораторна робота №3 Створення прототипів.....	29
Лабораторна робота №4 Візуалізація вимог користувача.....	29
Література .....	39
Додаток А Перелік тем для лабораторних робіт .....	40
Додаток Б Нотація для моделювання бізнес-процесів.....	41

## ВСТУП

Дисципліна "Аналіз вимог до програмного забезпечення" спрямована на отримання студентом базових знань та практичних навичок з виявлення, аналізу та керування вимогами.

Успішний досвід розробки показує, що ефективне керування вимогами є ключовим фактором всього процесу розробки програмного забезпечення. Вимоги визначають те, що повинна робити система. Тому протягом всього життєвого циклу проєкту необхідно організувати ефективну роботу з ними. Першим кроком у цьому напрямку є організація зберігання всіх виявлених вимог, що обумовлює актуальність вивчення даної дисципліни.

Особливу увагу треба приділяти правильному оформленню звітів з лабораторних робіт. Звіт повинен містити: титульний аркуш (на ньому вказують назву міністерства, назву університету, назву кафедри, номер, вид і тему роботи, виконавця та особу, що приймає звіт, рік); мету, варіант і завдання роботи; лаконічний опис теоретичних відомостей; повний текст всіх версій розроблених документів.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word: шрифт Times New Roman, 14 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм і роздруків даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах.

Для виконання лабораторних робіт рекомендується використовувати Jira Software.

# **ЛАБОРАТОРНА РОБОТА №1 CASE–ЗАСОБИ ДЛЯ ЗБЕРІГАННЯ ТА КЕРУВАННЯ ВИМОГАМИ. РОБОТА З ВИМОГАМИ.**

## **1.1 Мета роботи**

Вивчити основні можливості Jira Software, ознайомитись з її інтерфейсом та навчитися створювати відповідні проекти, навчитись розробляти вимоги та документувати їх за допомогою плагіну «R4J - Requirements Management for Jira».

## **1.2 Короткі теоретичні відомості**

Jira Software (далі Jira) - система відстеження помилок, призначена для організації спілкування з користувачами, і для управління проектами. Дозволяє команді розробників розробляти, структурувати, встановлювати пріоритети, відстежувати, контролювати зміни вимог, які з'являються на будь-якому етапі розробки компонентів додатку.

«R4J - Requirements Management for Jira» (далі R4J) – плагін для Jira, що дозволяє створювати, редагувати вимоги до проекту та встановлювати зв'язки між ними, а також візуалізувати ці відношення у вигляді графів, діаграм і т.д.

Вимоги до програмного забезпечення — набір вимог щодо властивостей, якості та функцій програмного забезпечення, що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів тощо.

Аналіз вимог полягає в визначенні потреб та умов, які висуваються щодо нового, чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників, таких як користувачі чи бенефіціари (вигодоотримувачі).

Аналіз вимог є критичним для успішної розробки проекту. Вимоги мають бути задокументованими, вимірними, тестовними, пов'язаними з бізнес-потребами, і описаними з рівнем деталізації достатнім для конструювання системи. Вимоги можуть бути

архітектурними, структурними, поведінковими, функціональними, та не функціональними.

Функціональні вимоги — це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система.

Нефункціональні вимоги — це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути.

### 1.2.1 Встановлення Jira

Першим кроком, який необхідно виконати, це пройти реєстрацію на офіційному вебсайті Atlasssian (рис. 1.1) для встановлення Jira. Для того, щоб створити особистий кабінет, необхідно натиснути кнопку “Get it free” на сайті, або перейти за посиланням - <https://www.atlassian.com/software/jira/free>.

Після цього обрати домен для своєї хмарної платформи.

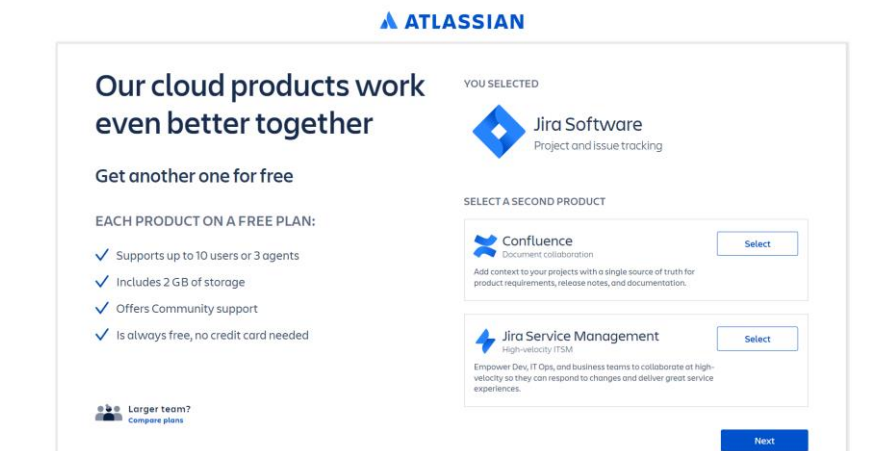


Рисунок 1.1 – Створення акаунту Jira Software

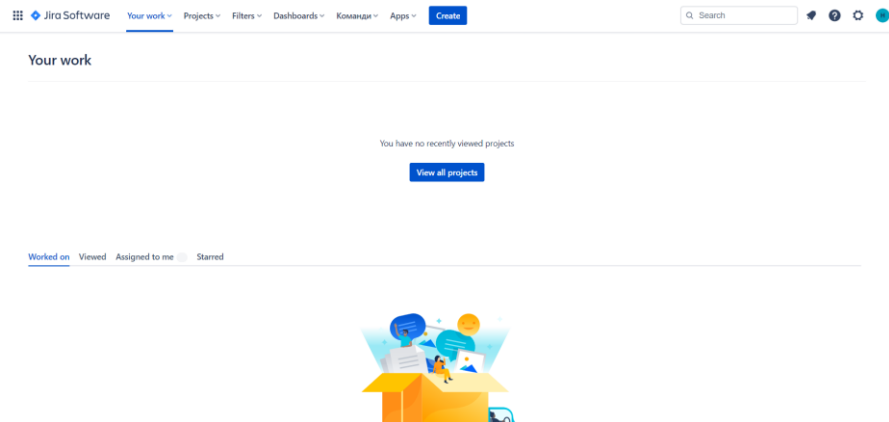


Рисунок 1.2 – Головна сторінка новоствореної хмари

### 1.2.2 Встановлення плагінів

Для встановлення плагінів для Jira необхідно натиснути на кнопку «Apps» і випадальному меню обрати пункт «Explore more apps» (рис. 1.3).

Після переходу до магазину, буде відкрито вікно пошуку застосунків. В пошуковий рядок зліва необхідно ввести назву необхідного плагіну (наприклад, «R4J») або обрати плагін зі списку запропонованих нижче (рис. 1.4). Для активації плагіну необхідно активувати тимчасову безкоштовну версію («Free Trial») або придбати його.

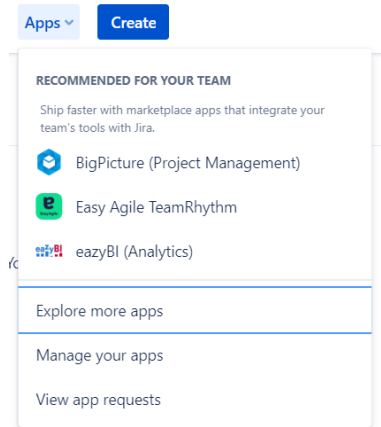


Рисунок 1.3 – Перехід до «Atlassian Marketplace»

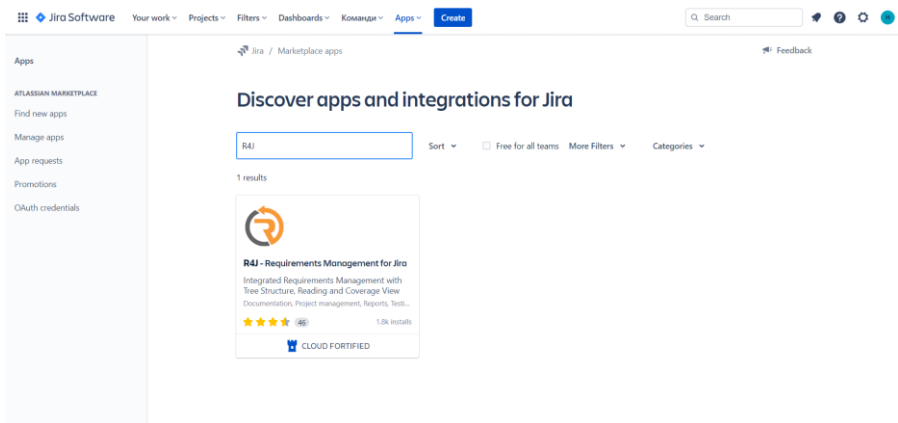


Рисунок 1.4 – Пошук плагіну у «Atlassian Marketplace»

## 1.3 Порядок виконання роботи

### 1.3.1 Створення проєкту

Для створення проєкту необхідно у меню «Projects» обрати пункт «Create project» (рис. 1.5), після чого буде відкрито форму створення нового проєкту (рис. 1.6, 1.7), де необхідно обрати пункт «Kanban» у категорії «Software development».



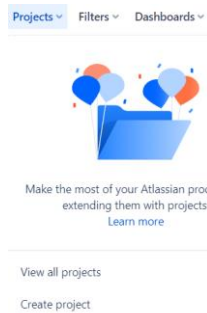


Рисунок 1.5 – Створення нового проєкту

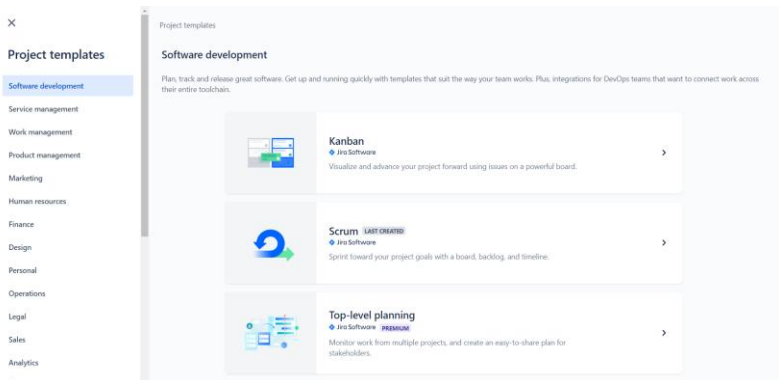


Рисунок 1.6 – Форма створення проєкту (вибір типу)

← Back to project types

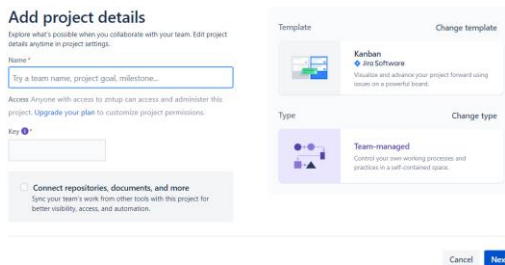


Рисунок 1.7 – Форма створення проєкту (введення назви)

Для роботи з плагіном RJ4 необхідно додати новостворений проєкт до списку проєктів RJ4. Для цього у меню «Apps» необхідно обрати пункт «Manage All Apps», після чого буде відкрито вікно керування R4J. Після цього, необхідно в лівому меню обрати пункт «Getting Started» (рис. 1.8).

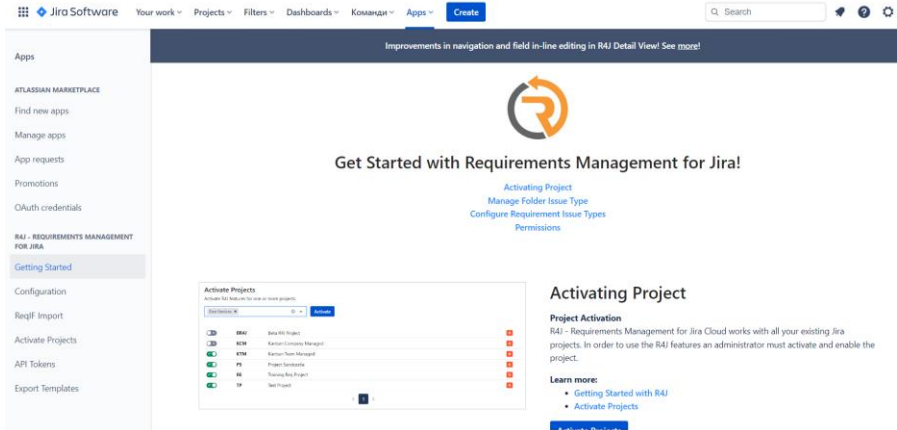


Рисунок 1.8 – Сторінка початку роботи з R4J

На відкритій сторінці (рис. 1.9) необхідно активувати ваш проєкт, за допомогою кнопки «Activate Projects», прогортавши нижче, натиснути кнопку «Specify Folder Issue Type», у вікні, що з’явилося, обрати у списку пункт «Create Issue Type» (Рис. 1.9). У поле «Name» вводимо *Folder*, відмічаємо прапорцем пункт «Add created issue type to all activated projects», натискаємо «Create».

#### Manage Folder Issue Type Dialog

Folder Issue Type

Please select an issue type for folders.

Create issue type

Epic

Рисунок 1.9 – Конкретизація типу Folder для R4J

Гортаємо сторінку в самий низ, натискаємо кнопку «Configure Permission Schemes», після чого відкриється нове вікно (Рис. 1.10), натискаємо кнопку «Permissions», виставляємо права доступу, як на рисунку 1.11.

Name	Projects	Actions
<b>Default Permission Scheme</b> This is the default Permission Scheme. Any new projects that are created will be assigned this scheme.		<a href="#">Permissions</a> <a href="#">Copy</a> <a href="#">Edit</a>

Рисунок 1.10 – Налаштування прав доступу

<b>[R4J] Baseline modification</b> <small>Allows to create/edit/delete baselines.</small>	Any logged in user	Update	Remove
<b>[R4J] Coverage View modification</b> <small>Allows to create/edit/delete views.</small>	Any logged in user	Update	Remove
<b>[R4J] Traceability Matrix modification</b> <small>Allows to create/edit/delete traceability matrices. Works only when restriction is enabled in R4J configuration page.</small>	Any logged in user	Update	Remove
<b>[R4J] Tree modification</b> <small>Allows to add/move/remove items in tree.</small>	Any logged in user	Update	Remove

Рисунок 1.11 – Конфігурація прав доступу

### 1.3.2 Налаштування проєкту до роботи з R4J

Для початку роботи з R4J необхідно провести певні маніпуляції з самим проєктом Jira.

Перейдіть у ваш проєкт за допомогою випадаючого меню «Projects». Після цього необхідно перейти у вкладку «Project settings» (Рис. 1.12).

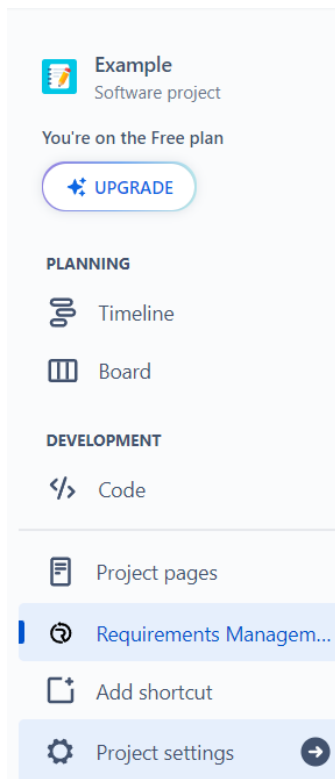


Рисунок 1.12 – Перехід у налаштування проєкту

Після переходу на нову сторінку, в тому ж меню обираємо пункт «Issue types». Натискаємо «Add issue type», «Create issue type».

Створюємо 4 види вимог: Customer Requirements, Functional Requirements, Non-Functional Requirements, Improvements (Рис. 1.13). Також обов’язково створюємо тип Folder, йому можна залишити

стандартні поля. За допомогою перетаскування наповнюємо кожну вимогу полями, як на Рисунку 1.14.

### Create issue type

Name \*

Customer Requirements

Description

Let people know when to use this issue type

Icon



Change icon

Create

Cancel

Рисунок 1.13 – Створення нового типу вимог



Customer Requirements

Edit workflow



Let people know when to use this issue type

Summary

REQUIRED

Priority



Due date



Assignee



Reporter



Environment



Description



Context fields ⓘ

Status

Labels



Рисунок 1.14 – Поля нової вимоги

### 1.3.3 Створення вимог

Після додавання існуючого проекту до плагіну R4J з'являється можливість відкрити огляд проекту за допомогою плагіну. Для цього необхідно у меню «Apps» обрати пункт з назвою власного проекту, після чого відкриється вікно перегляду проекту (за замовчуванням – «Tree View»). Для взаємодії з об'єктами необхідно натиснути праву кнопку миші на об'єкт в меню зліва (рис. 1.9).

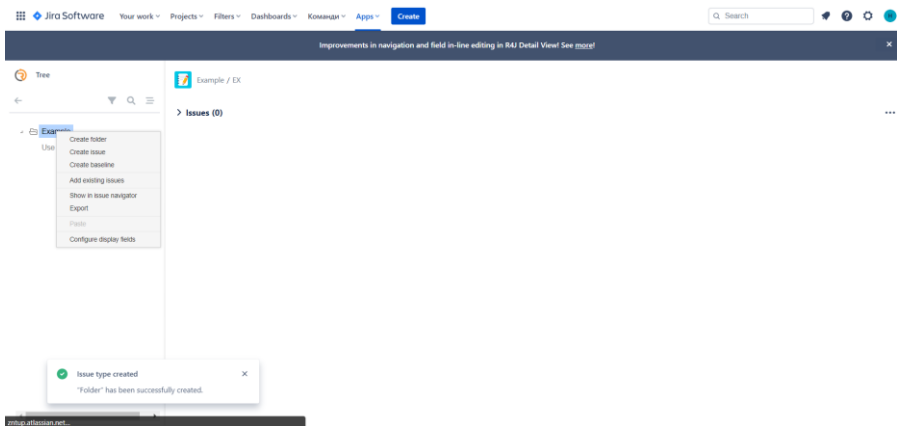


Рисунок 1.15 – «Tree View» проекту

Для створення нової вимоги у контекстному меню об'єкта необхідно обрати пункт «Create» (для додавання вже існуючих, створених за допомогою меню «Issues», оберіть «Add existing issue»). Для кращої організації процесу заздалегідь створіть папки під кожен тип вимог. В проекті є наступні типи вимог:

- користувачка вимога (англ. «Customer requirement») – вимоги, що висуваються певною групою користувачів (авторизовані користувачі, гості, адміністратори, модератори тощо);
- функціональні вимоги (англ. «Functional requirement») – вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система;

- нефункціональні вимоги (англ. «Non-functional requirement») – це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути;

- поліпшення (англ. «Improvement») – вимоги до модифікації вже існуючих функцій системи;

- вимоги до тестування (англ. «Testing») – вимоги до того, які саме тести повинні бути виконані над системою, перш ніж вона може вважатись готовою.

Під час створення нової вимоги можна визначити наступні її параметри:

- заголовок, назва (англ. «Summary») – короткий опис, що дає змогу зрозуміти суть вимоги;

- пріоритет (англ. «Priority») – ступінь важливості виконання вимоги;

- період виконання (англ. «Due Date») – період, у який вимога повинна бути виконана;

- виконавець (англ. «Assignee») – прив’язка виконавця до вимоги;

- доповідач (англ. «Reporter») – прив’язка доповідача до вимоги;

- середовища (англ. «Environment») – опис вимог, яким повинна задовольняти система, для виконання поставленої задачі;

- опис (англ. «Description») – повний, розгорнутий опис вимоги;

- мітки (англ. «Labels») – текстові мітки для фільтрації вимог за ними.

### 1.3.4 Встановлення відношень між вимогами

Для встановлення відношення між двома вимогами необхідно викликати контексте меню вимоги зі списку та натиснути «Link existing issue» (рис. 1.16, 1.17). Іншим варіантом є створення нової вимоги, для якої одразу буде встановлено зв’язок (пункт «Create linked issue»).

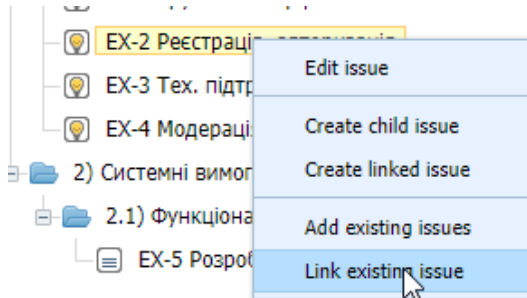


Рисунок 1.16 – Встановлення відношення між вимогами

## Add Link

Рисунок 1.17 – Форма встановлення відношення

Встановлені відношення між вимогами можна візуалізувати за допомогою вбудованих у R4J функцій.

Для візуалізації відношень у вигляді таблиці покриття необхідно обрати відповідний пункт «Coverage» у меню ліворуч (рис. 1.18), після чого натиснути кнопку «Configure View» (зазвичай, на першому рівні розміщують вимоги з користувацьких історій), після чого додати вимоги другого рівня натисненням кнопки «Add Coverage Level» (на другому рівні, як правило, розміщують функціональні і нефункціональні вимоги). Необхідно додати стільки рівнів покриття, скільки рівнів вкладеності у визначених відношеннях вимог є (наприклад, наступним, третім, рівнем може бути визначені вимоги до



тестування, які відносяться до функціональних вимог). Результат зображено на рисунку 1.19.

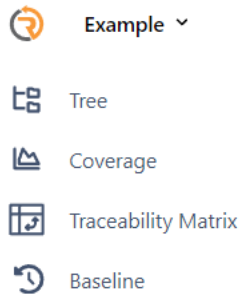


Рисунок 1.18 – Відкриття вкладки покриття

Example	
<div> Add Coverage Level Refresh Clear All Display Fields Export </div>	
Level 1	Level 2
4 Issue(s)	2 Issue(s) Coverage: 25%
<div> EX-4 ↑ OPEN Модерація </div>	Not covered
<div> EX-3 ↑ OPEN Тех. підтримка </div>	Not covered
<div> EX-2 ↑ OPEN Реєстрація, авторизація </div>	<div> EX-5 ↑ OPEN Розроблення реєстрації (relates to) </div>
	<div> EX-6 ↑ OPEN Розроблення авторизації (relates to) </div>
<div> EX-1 ↑ OPEN Зручний інтерфейс </div>	Not covered

Рисунок 1.19 – Результат покриття

Для створення матриці трасування необхідно відкрити вкладку «Tracebilty» у меню ліворуч та натиснути на кнопку «Configure View». У відкритій формі необхідно обрати параметри трасування (рис. 1.20), які потім будуть відображені на екрані (рис. 1.21).

Рисунок 1.20 – Створення матриці трасування

Рисунок 1.21 – Побудована матриця трасування

### 1.3.3 Створення глосарію

Ані Jira, ані R4J не мають вбудованих функцій для створення глосарію, тому для його оформлення створюється нова вимога з назвою «Глосарій», описом якої є список всіх використаних термінів і позначень (рис. 1.22).

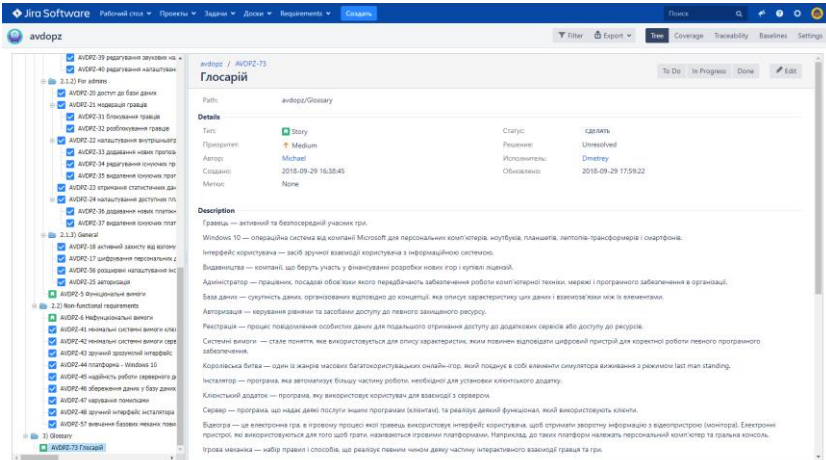


Рисунок 1.22 – Створений глосарій

### 1.4 Завдання на лабораторну роботу

- 1.4.1. Вибрати тему для роботи у викладача (Додаток А).
- 1.4.2. Ознайомитись з інтерфейсом Jira Software, відповідно до завдання створити новий проект, налаштувати властивості проекту.
- 1.4.3. Розробити користувацькі історії (вимоги користувачів).
- 1.4.4. Розробити функціональні та нефункціональні вимоги до проекту.
- 1.4.5. Відобразити декілька графів відношень вимог.
- 1.4.6. Відобразити таблицю покриття вимог.
- 1.4.7. Відобразити матрицю трасування.
- 1.4.8. Оформити звіт з роботи.
- 1.4.9. Відповісти на контрольні запитання.

### 1.5 Контрольні запитання

- 1.5.1. Дайте визначення вимогам до програмного забезпечення.
- 1.5.2. Сформулюйте завдання аналізу вимог.
- 1.5.3. Які вимоги називаються функціональними?
- 1.5.4. Які вимоги називаються нефункціональними?
- 1.5.5. Яке основне призначення Jira Software?
- 1.5.6. Для чого слугує плагін «Requirement Management for Jira – R4J»?
- 1.5.7. Які типи відношень можуть бути між вимогами?
- 1.5.8. Що показує матриця трасування?
- 1.5.9. На що вказує аналіз покриття?
- 1.5.10. Що таке глосарій?

## ЛАБОРАТОРНА РОБОТА №2 ДІАГРАМИ БІЗНЕС ПРОЦЕСІВ

### 2.1 Мета роботи

Ознайомитись з концепцією BPMN 2.0 для конструювання бізнес-діаграм.

### 2.2 Короткі теоретичні відомості

BPMN (англ. «Business Process Model and Notation», модель та нотація бізнес-процесів) – система умовних позначень для моделювання бізнес-процесів. Розроблена Business Process Management Initiative (BPMI) та підтримується Object Management Group після їх злиття в 2005 році.

Для роботи потрібно створити обліковий запис на сервісі Signavio(<https://academic.signavio.com/p/login>). В свою чергу, для його створення вам обов'язково необхідно мати пошту з університетським доменом **@zp.edu.ua**. Для отримання актуальної інформації, щодо того, як отримати таку пошту – зверніться до куратора своєї групи або офіційних ресурсів кафедри.

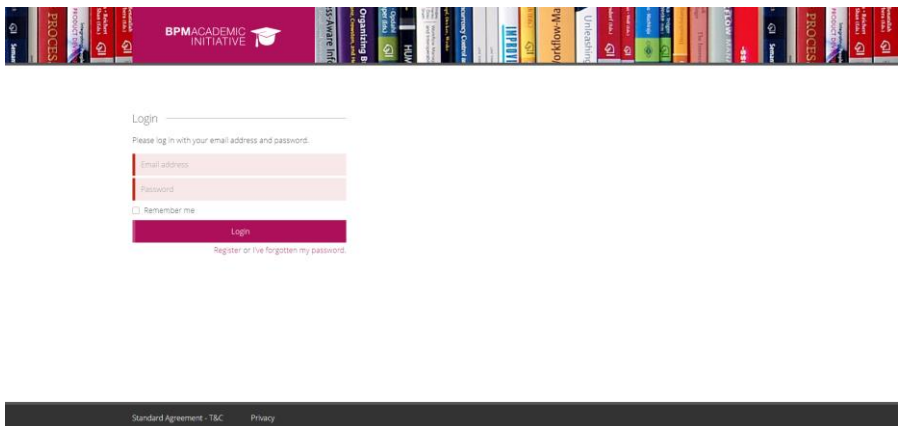


Рисунок 2.1 – Вхід до особистого кабінету Signavio

Для створення BPMN діаграми необхідно використовувати набори візуальних елементів (рис. 2.2).

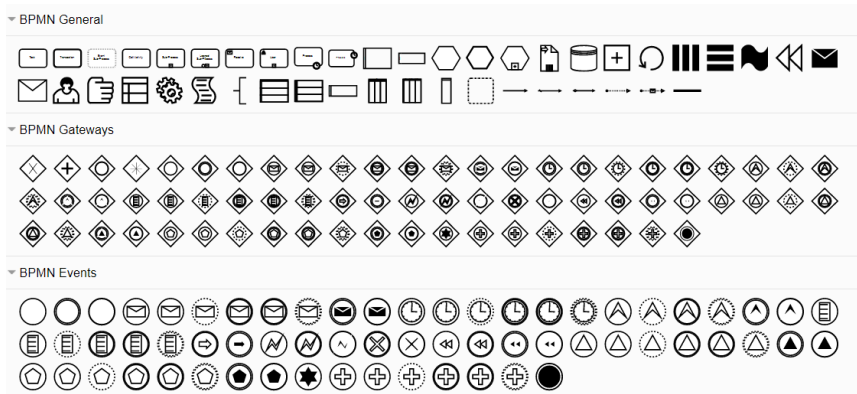


Рисунок 2.2 – Візуальні елементи BPMN

Створення нової діаграми починається з вибору саме, яку діаграму необхідно побудувати.

В нашому випадку це Business Process Diagram (BPMN 2.0) (Рис. 2.3).

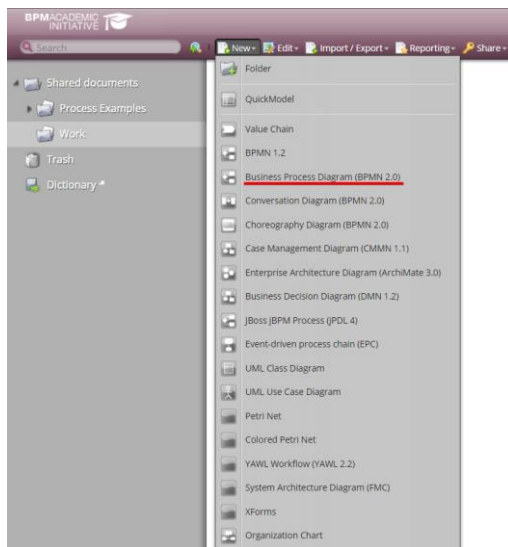


Рисунок 2.3 – Створення нової діаграми

Для побудови простих бізнес-процесів в нотації BPMN 2.0 обираємо «BPMN (Core Elements)», як на рисунку 2.4.

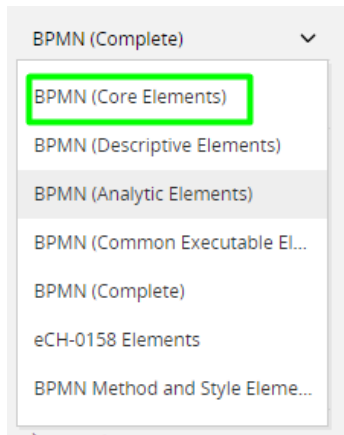


Рисунок 2.4 – Вибір діаграми для побудови простих бізнес-процесів

Приклади простих бізнес-процесів BPMN діаграм зображено на рисунках 2.5 - 2.7.



Рисунок 2.5 – Діаграма простого бізнес-процесу перевірки клієнтів бібліотеки, які потребують нагадування про здачу книги

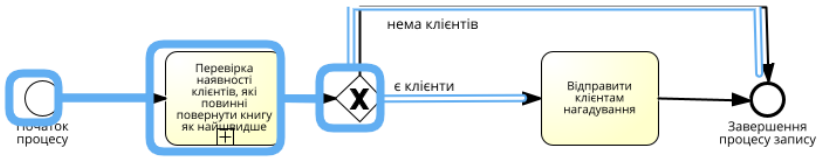


Рисунок 2.6 – Приклад симуляції простого бізнес-процесу перевірки клієнтів бібліотеки, які потребують нагадування про здачу книги

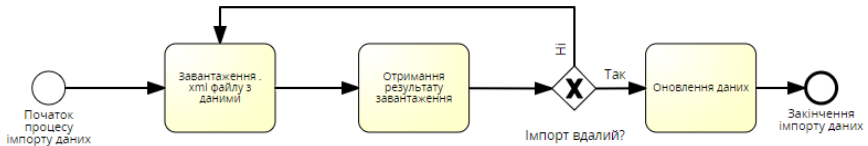


Рисунок 2.7 – Діаграма простого бізнес-процесу

Приклади Business-to-business (B2B) BPMN діаграм зображено на рисунках 2.8 - 2.10.

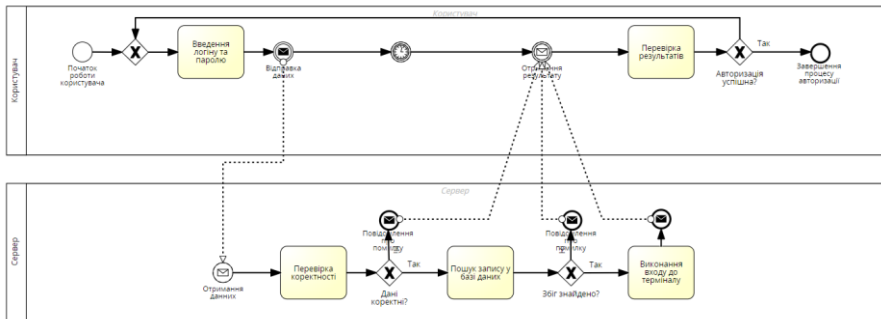


Рисунок 2.8 – Діаграма бізнес-процесу авторизації



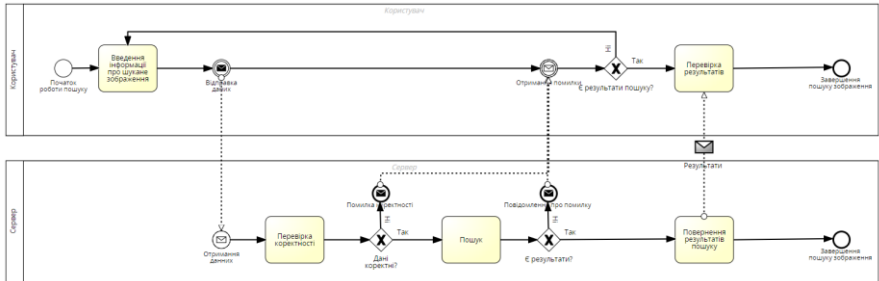


Рисунок 2.9 – Діаграма бізнес-процесу пошуку зображення

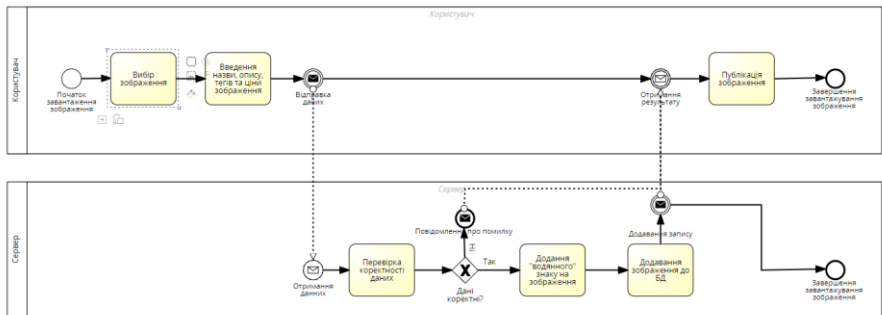


Рисунок 2.10 – Діаграма бізнес-процесу завантаження зображення

Створену діаграму необхідно перевірити на помилки за допомогою Check syntax (Рис. 2.11).

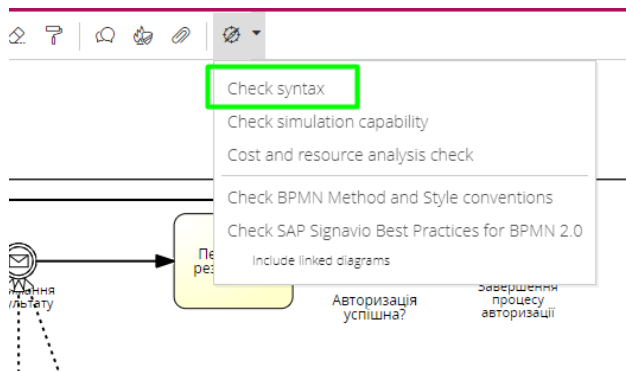


Рисунок 2.11 – Перевірка діаграми на синтаксичні помилки

Далі необхідно провести перевірку вашої діаграми на можливість симуляції за допомогою Check simulation capability (Рис.2.12)

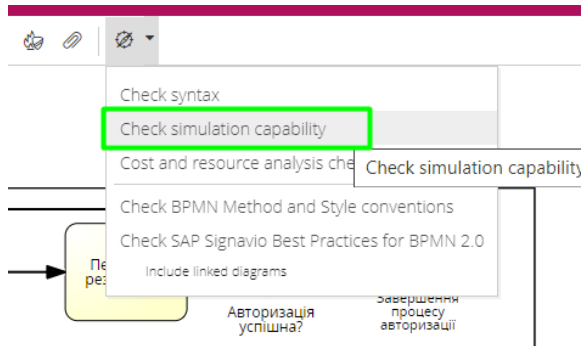


Рисунок 2.12 – Перевірка діаграми на можливість симуляції

Результат перевірки можна побачити на рисунку 2.13.

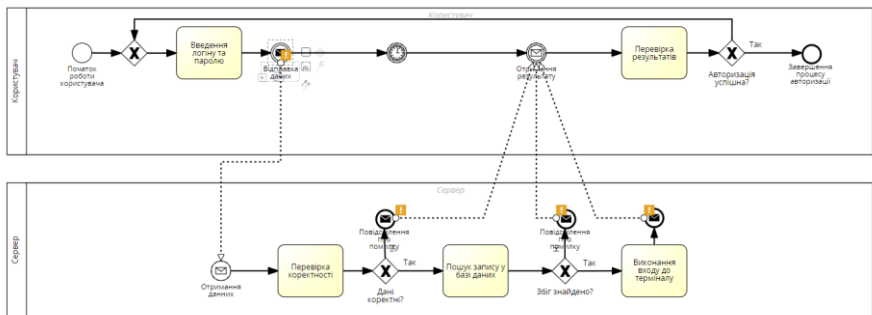


Рис. 2.13 – Результат перевірки діаграми на можливість симуляції

**Зверніть увагу! Сервіс signavio.com попереджає, що некоректно працює з повідомленнями і при симуляції та ігнорує їх.**

Наступним кроком необхідно провести симуляцію вашого бізнес-процесу. Для цього необхідно повернутися до головного меню,



### **2.3 Завдання на лабораторну роботу**

2.3.1 Описати бізнес-процеси для вашого застосунку, використовуючи діаграми BPMN 2.0 (Додаток Б), розробіть не менше трьох простих бізнес-процесів.

2.3.2 Створити Business2Business diagram, що відображає взаємодію повідомлень між процесами.

2.3.4 Зв'язати діаграми з вимогами.

2.3.5 Оформити звіт з роботи.

2.3.6 Відповісти на контрольні запитання

### **2.4 Контрольні запитання**

2.4.1 Що таке BPMN?

2.4.2 Що позначає термін Business2Business Diagram?

2.4.3 Які типи візуальних елементів BPMN існують? Яке їх призначення?

## ЛАБОРАТОРНА РОБОТА №3 ВІЗУАЛІЗАЦІЯ ВИМОГ КОРИСТУВАЧА

### 3.1 Мета роботи

Навчитись розробляти діаграми прецедентів та візуалізувати сценарії.

### 3.2 Короткі теоретичні відомості

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Приклади діаграм прецедентів для описаних у Лабораторній роботі №2 подій («Реєстрація нового користувача», «Пошук зображення» та «Завантаження зображення») зображені на рисунках 3.1 - 3.3.

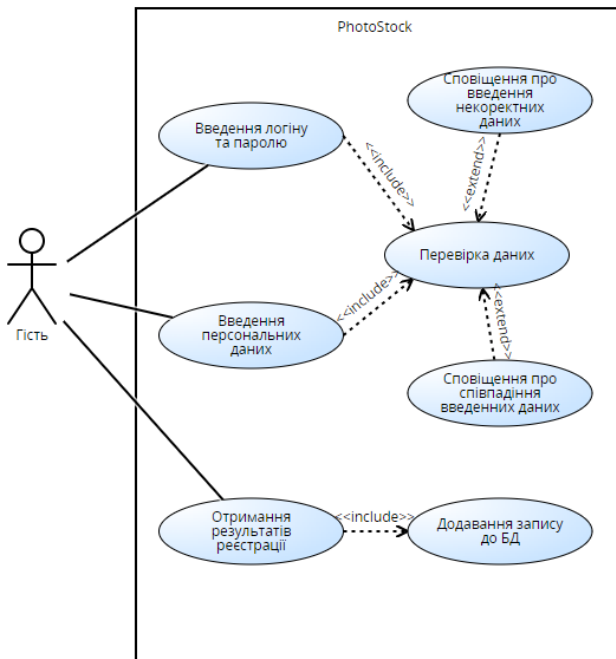


Рисунок 3.1 – Прецедент «Реєстрація нового користувача»

Потік подій для розробленої діаграми прецеденту «Реєстрація нового користувача»:

Передумова: доступ до мережі інтернет, відкрита сторінка реєстрації нового облікового запису.

Основний виконавець – незареєстрований користувач (гість).

Основний потік:

1. Користувач вводить логін та пароль.
  - 1.1 Система «PhotoStock» виконує валідацію даних. У разі виникнення невідповідності - А1. Введення некоректних даних. У разі збігу даних з наявними у БД – А2. Введення наявних даних.
2. Користувач вводить персональних даних.
  - 2.1 Система «PhotoStock» виконує валідацію даних. У разі виникнення невідповідності – А1. Введення некоректних даних.
3. Користувач отримує результати реєстрації.
  - 3.1 Система «PhotoStock» додає новий запис у БД.

Альтернативний потік:

А1. Вхідні дані є некоректними.

1. Користувач отримує сповіщення з помилкою, де вказано, які поля заповненні невірно.
2. Потік повертається до пункту 1 основного потоку.

А2. Логін вже наявний у БД.

1. Користувачу пропонуються варіанти валідного логіну, що ще не наявний в БД.
2. Потік повертається до пункту 2 основного потоку.

Постумова: зареєстрований обліковий запис користувача.

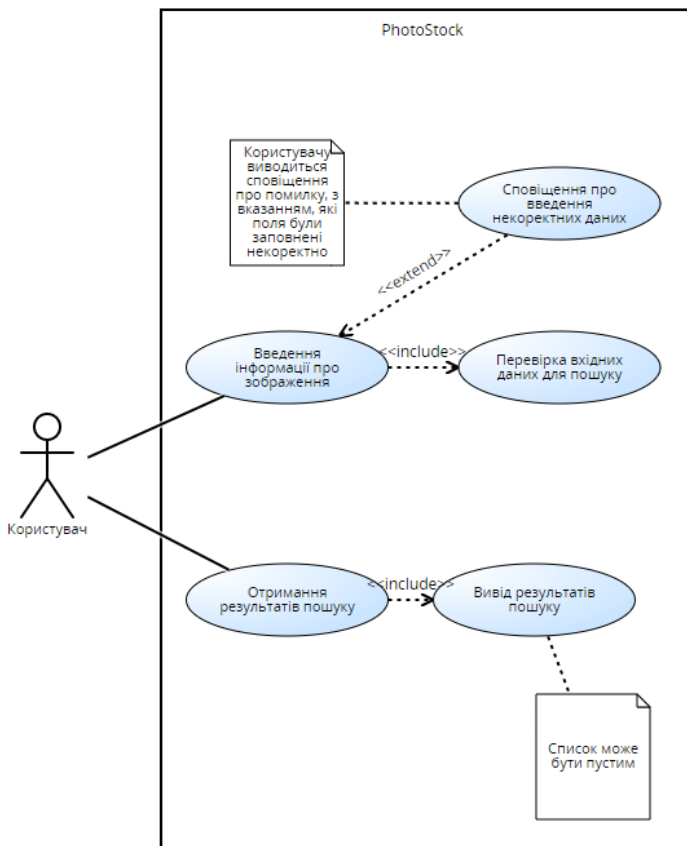


Рисунок 3.2 – Прецедент «Пошук зображення»

Потік подій для розробленої діаграми прецеденту «Пошук зображення»:

Передумова: доступ до мережі Інтернет, відкрита сторінка пошуку на сайті.

Етапи основного потоку подій:

1. Користувач вводить інформації про зображення.

1.1 Виконується перевірка вхідних даних для пошуку.

У разі надання некоректних даних користувачем – А1. Введення некоректних даних.

2. Користувач отримує результати пошуку.

### 2.1 Система «PhotoStock» виводить результати пошуку.

У разі, якщо жодне зображення не задовольняє умови пошуку – система виведе повідомлення про те, що результати пошуку відсутні.

Етапи альтернативного потоку:

A1. Введення некоректних даних.

1. Користувачу виводиться сповіщення про помилку, з вказанням, які поля були заповнені некоректно.

2. Потік повертається до пункту 1.1 основного потоку.

Постумова: знайдене(-і) зображення.

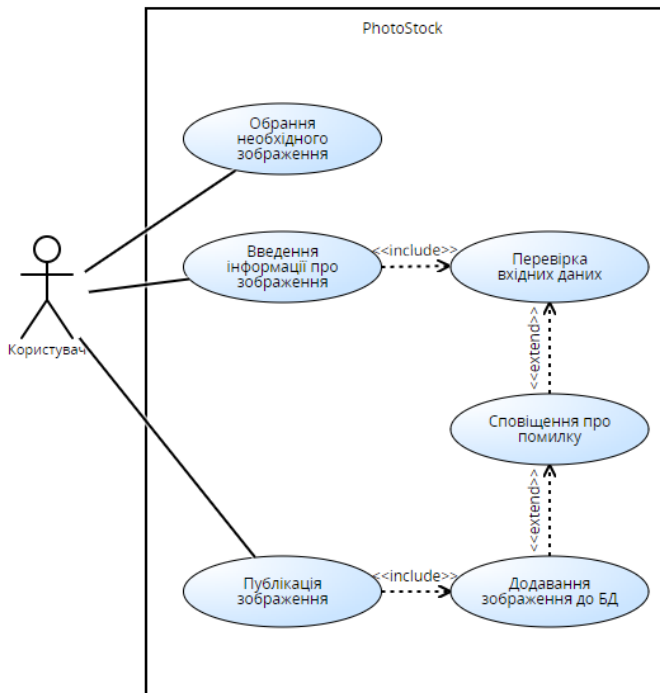


Рисунок 3.3 – Прецедент «Завантаження зображення на сервер»

Потік подій для розробленої діаграми прецеденту «Завантаження зображення»:



Передумова: доступ до мережі інтернет, відкрита сторінка завантаження нового зображення на сайті.

Основний виконавець – користувач.

Етапи основного потоку подій:

1. Користувач обирає необхідне зображення.
2. Користувач вводить інформацію про зображення.

2.1 Перевірка вхідних даних. Якщо вхідні дані не задовольняють умовам – А1. Введення невірних даних.

3. Користувач публікує зображення.

3.1 Система «PhotoStock» додає зображення до БД. Якщо в процесі відбувається помилка – А1. Введення невірних даних.

Етапи альтернативного потоку:

А1. Введення невірних даних.

1. Користувачу виводиться сповіщення про помилку з вказанням, які саме вхідні дані були невірними.

2. Повернення потоку до пункту 2 основного потоку.

Постумова: зображення опубліковане на сайті та додане до БД.

### 3.3 Завдання на лабораторну роботу

3.3.1 Розробити прецеденти для візуалізації вимог користувача (3-5 діаграм). Для опису прецедентів використовувати шаблон Use Case Specification, що містить наступні розділи:

- 1 Use Case Name (Найменування прецеденту)
  - 1.1 Brief description (Короткий опис)
- 2 Flow of Events (Потоки подій)
  - 2.1 Basic flow (Основний потік)
  - 2.2 Alternative flows (Альтернативні потоки)
    - 2.2.1 Alternative flow 1 (Альтернативний потік 1)
    - 2.2.2 Alternative flow 2 (Альтернативний потік 2)
- 3 Special requirements (Особливі вимоги)
- 4 Preconditions (Передумови)
- 5 Postconditions (Постумови)
- 6 Extension points (Крапки розширення)

3.3.2 Оформити звіт з роботи.

## ЛАБОРАТОРНА РОБОТА №4 СТВОРЕННЯ ЕСКІЗІВ

### 4.1 Мета роботи

Навчитись розробляти ескізи та потоки вікон інтерфейсу застосунку.

### 4.2 Короткі теоретичні відомості

Для створення ескізів графічного інтерфейсу можна використовувати будь-який графічний редактор (Paint, Photoshop, GIMP, Paint.NET, Illustrator, Figma, Adobe XD та інші). В нашому випадку будемо використовувати безкоштовний онлайн сервіс Draw.io (<https://www.draw.io/>).

Після відкриття головної сторінки буде запропоновано два основні варіанти подальших дій (рис. 4.1): «Create New Diagram» - створити нову діаграму та «Open Existing Diagram» - відкрити існуючу діаграму.

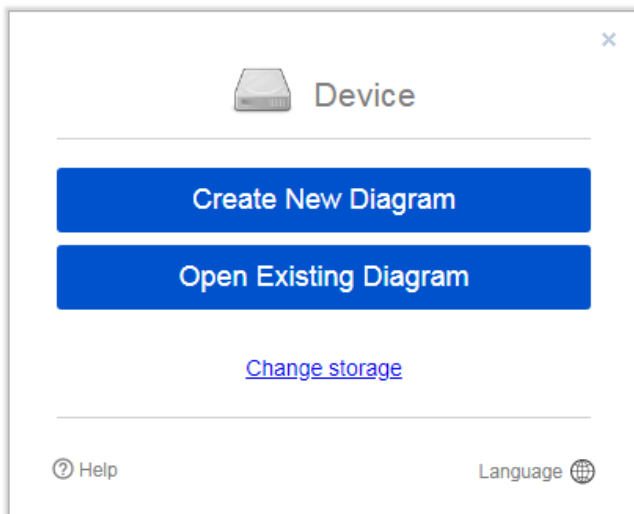


Рисунок 4.1 – Головна сторінка сервісу Draw.io

В нашому випадку потрібно натиснути «Create New Diagram».

На наступному кроці (рис. 4.2) потрібно вказати ім'я майбутньої діаграми з розширенням «.drawio», обрати пустий шаблон («Blank Diagram») та натиснути клавішу «Create».

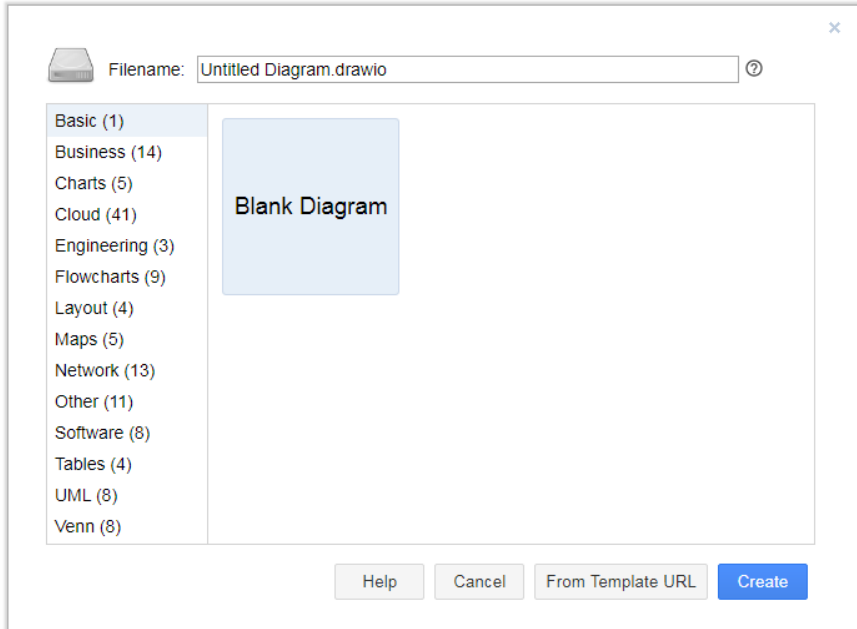


Рисунок 4.2 – Обрання шаблону діаграми

Далі ми опиняємося на сторінці редагування діаграм (рис. 4.3).

Для створення ескізу інтерфейсів можна використовувати примітивні геометричні фігури (рис. 4.4), такі як прямокутник, коло, квадрат, лінії та інші.

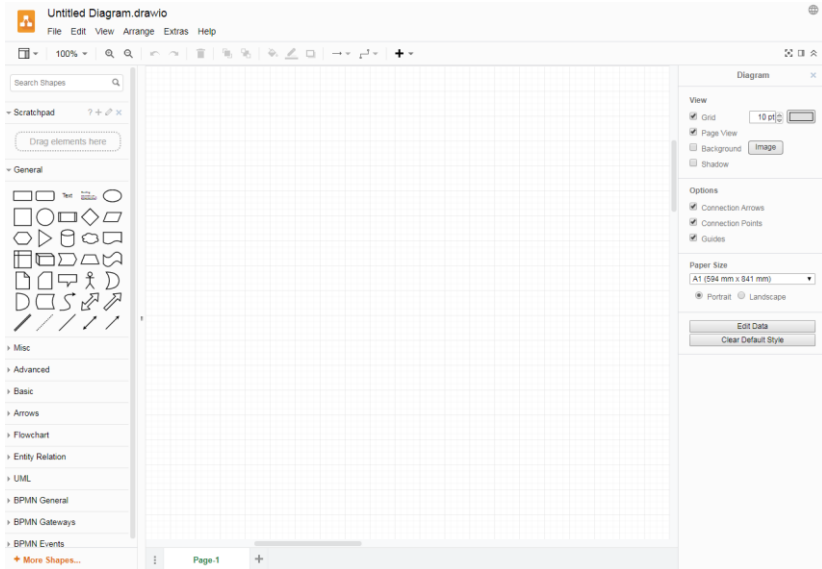


Рисунок 4.3 – Інтерфейс редагування діаграм

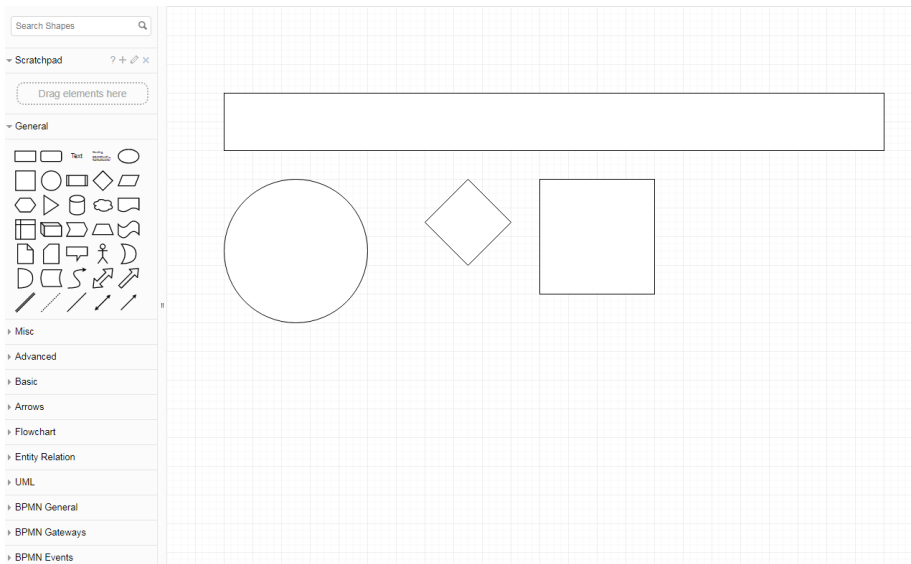


Рисунок 4.4 – Прімітивні елементи для створення діаграм

Для початку необхідно розробити ескізи частин, які будуть повторюватися, наприклад для сайту це «header» (рис. 4.5) та «footer» (рис. 4.6).

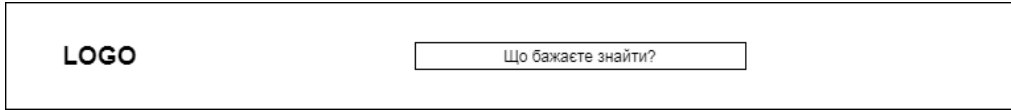


Рисунок 4.5 – Приклад «header» сайту



Рисунок 4.6 – Приклад «footer» сайту

Після створення ескізів, які будуть повторюватися, можна приступити до розробки приблизного зовнішнього виду інтерфейсу (рис. 4.7).

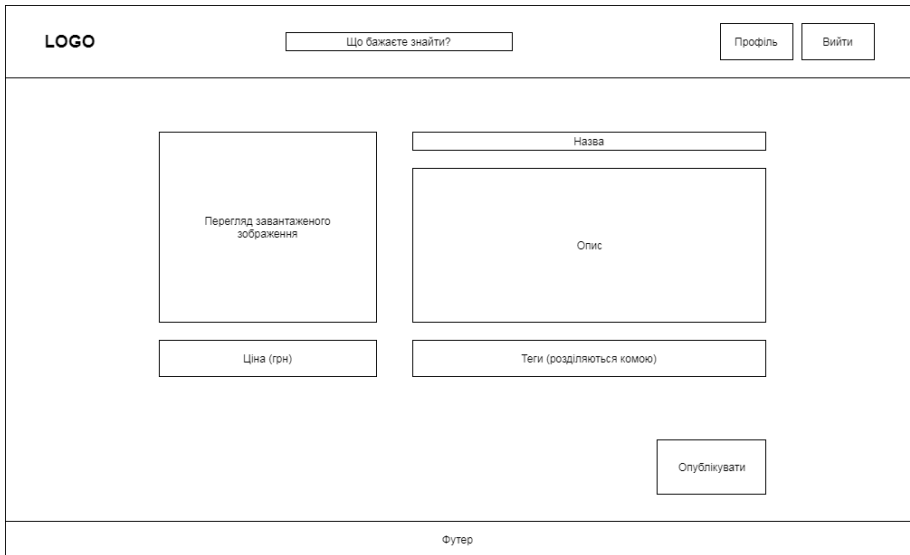


Рисунок 4.7 – Прототип інтерфейсу сторінки завантаження зображення на сайт

Після того, як ескіз інтерфейсу розроблений, необхідно експортувати його на комп'ютер у форматі PNG, для цього необхідно перейти до вкладки «File», потім «Export as» та обрати пункт «PNG...» (рис. 4.8).

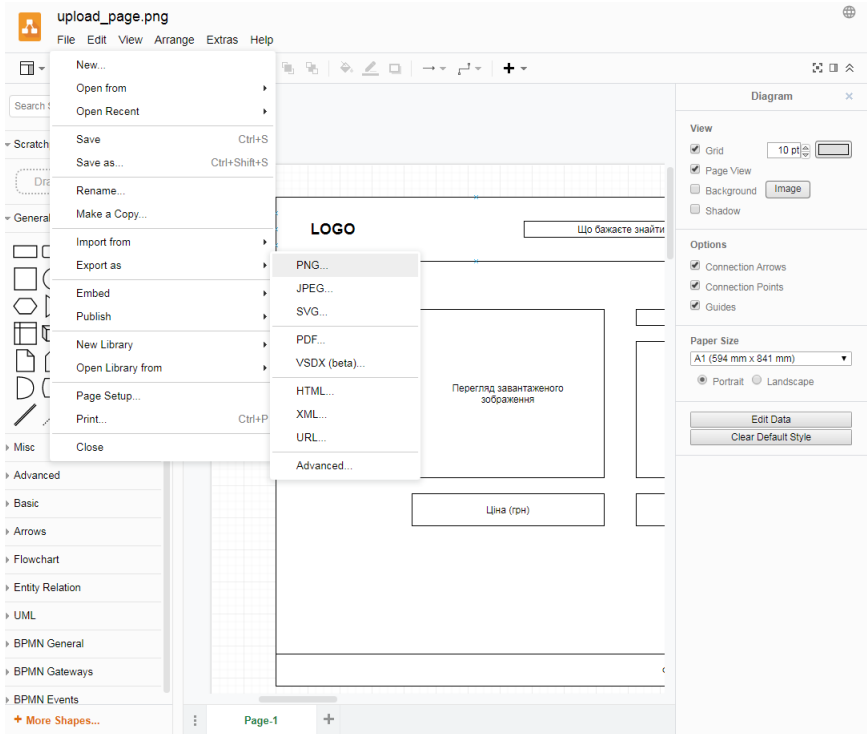


Рисунок 4.8 – Збереження зображення у форматі PNG

### 4.3 Завдання на лабораторну роботу

- 4.3.1 Розробити ескіз інтерфейсу системи, що розробляється.
- 4.3.2 Розробити частини (parts) інтерфейсу користувача.
- 4.3.3 Зібрати розроблені частини в ескізи (sketch) вікон інтерфейсу.
- 4.3.4 Оформити звіт з роботи.

#### **4.4 Контрольні запитання**

- 4.4.1 З якою метою розробляється ескіз інтерфейсу системи?
- 4.4.2 Які елементи можливо додавати до інтерфейсу?
- 4.4.3 Які методи розробки інтерфейсу ви знаєте?
- 4.4.4 З якою метою зв'язують інтерфейс з вимогами?
- 4.4.5 Яким чином з'єднуються частини інтерфейсу?

## Додаток А

### Перелік орієнтовних тем для лабораторних робіт

1. Розробка соціальної мережі.
2. Розробка відео-блогу
3. Розробка Wiki-системи
4. Розробка інтернет магазину
5. Розробка твітера
6. Розробка системи для он-лайн трансляції телепередач
7. Розробка системи для дистанційної освіти
8. Розробка програмного забезпечення для голосового зв'язку (аналогу Skype).
9. Розробка безкоштовної системи миттєвого обміну повідомленнями (аналог Qip)
10. Розробка Dropbox.
11. Розробка поштового клієнта.
12. Розробка безкоштовного онлайн-офісу (аналог Google Docs)
13. Розробка довідково-пошукової системи
14. Розробка АРМ диспетчера таксі
15. Розробка АРМ бібліотекаря
16. Розробка системи деканат
17. Розробка системи бухгалтерського обліку
18. Розробка системи підтримки складського господарства
19. Розробка системи інтернет-банкінгу
20. Розробка системи онлайн-керування розробки ПЗ
21. Розробка аналога графічного редактора в Internet
22. Розробка системи розрахунку страхування для фізичних та юридичних осіб
23. Розробка веборієнтованої системи підтримки групових рішень
24. Розробка експертної системи вибору методів прийняття рішень
25. Розробка системи оцінки якості програмного забезпечення
26. Оцінка надійності газотурбинних установок наземного використання.



# Додаток Б

## Нотація для моделювання бізнес-процесів BPMN 2.0

<http://bpmn.de/poster>

### BPMN 2.0 - Business Process Model and Notation

#### Activities

Task

Transaction

Event

Sub-Process

Call Activity

Activity Markers

Sub-Process Marker

Parallel All Marker

Sequential All Marker

Ad Hoc Marker

Compensation Marker

Task Types

Send Task

Receive Task

User Task

Manual Task

Business Rule Task

Service Task

Script Task

Conditional Flow

Default Flow

Sequence Flow

Inclusive Gateway

Exclusive Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

#### Conversations

A Conversation defines a set of participants and their interactions.

When marked with a [ ] symbol it is a Conversation Pool.

A Conversation is a wrapper for a globally defined Conversation or Sub-Process.

A Conversation is marked with a [ ] symbol.

A Conversation Link connects Conversations and Participants.

Conversation Diagram

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

#### Events

Start

Intermediate

End

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

Event Sub-Process

Non-Interrupting

Interrupting

Standard

#### Choreographies

Participant A

Participant B

Participant C

Participant D

Participant E

Participant F

Participant G

Participant H

Participant I

Participant J

Participant K

Participant L

Participant M

Participant N

Participant O

Participant P

Participant Q

Participant R

Participant S

Participant T

Participant U

Participant V

Participant W

Participant X

Participant Y

Participant Z

Participant AA

Participant AB

Participant AC

Participant AD

Participant AE

Participant AF

Participant AG

Participant AH

Participant AI

Participant AJ

#### Collaboration Diagram

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

Pool (Black Box)

Sub-Process

Conversation

Pool (White Box)

Sub-Process

Conversation

#### Gateways

Exclusive Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

#### Swimlanes

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

#### Activities

Task

Transaction

Event

Sub-Process

Call Activity

Activity Markers

Sub-Process Marker

Parallel All Marker

Sequential All Marker

Ad Hoc Marker

Compensation Marker

Task Types

Send Task

Receive Task

User Task

Manual Task

Business Rule Task

#### Gateways

Exclusive Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

Parallel Gateway

Complex Gateway

Event-based Gateway

#### Swimlanes

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

Information Flow

Message Flow

#### Data

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output

Input

#### Data

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output

Input

Output