

Національний університет «Києво-Могилянська академія»  
Кафедра інформатики

Ковалюк Т.В.

Методичні вказівки для виконання проєктних завдань  
з дисципліни «Проектування програмних систем»  
для студентів другого (магістерського) рівня вищої освіти

Київ – 2023

## Етапи та строки розробки документів з проєктування програмних систем

№ п.п	Зміст етапу (назва документу)	Термін здачі документу	Посилання на джерела	Інструментарій
1	Вибір теми проєкта із списку рекомендованих тем або за пропозицією студента	01.10.2023		
2	Формування команди проєкту (2-3 студента)	10.10.2023		
3	Розробка технічного завдання (англ. Product Requirements Document – PRD)	28.10.2023	<a href="https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B5%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F">https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B5%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F</a>	
4	Розробка специфікації вимог до програмного забезпечення (англ. Software requirements specification - SRS) за стандартом IEEE Std 830-1993	28.10.2023	<a href="https://personal.utdallas.edu/~chung/RE/IEEE830-1993.pdf">https://personal.utdallas.edu/~chung/RE/IEEE830-1993.pdf</a>	
5	Бізнес-аналіз. Розробка схеми бізнес-процесів на основі стандарту BPMN 2.0	28.10.2023	<a href="https://www.omg.org/spec/BPMN/2.0/PDF">https://www.omg.org/spec/BPMN/2.0/PDF</a>	BPMN Viewer and Editor. <a href="https://bpmn.io/">https://bpmn.io/</a> VisualParadigm Online. <a href="https://online.visual-paradigm.com/diagrams/features/bpmn-tool/">https://online.visual-paradigm.com/diagrams/features/bpmn-tool/</a> Camunda Modeler. <a href="https://camunda.com/bpmn/tool/">https://camunda.com/bpmn/tool/</a> BPMN Studio. <a href="https://bpmn.studio/">https://bpmn.studio/</a>
6	Функціональне моделювання та проєктування програмної системи на основі стандарту IDEF0 (ISO/ІЕС/ІЕЕЕ 31320-1:2012(E) ), подання схеми функціональної структури системи IDEF0, схеми потоків даних DFD )	11.11.2023	<a href="https://webstore.iec.ch/preview/info_isoiec_iee31320-1%7Bed1.0%7Den.pdf">https://webstore.iec.ch/preview/info_isoiec_iee31320-1%7Bed1.0%7Den.pdf</a>	Allfusion Process Modeler. <a href="https://ca-allfusion-process-modeler.software.informer.com/">https://ca-allfusion-process-modeler.software.informer.com/</a> BPWin. <a href="https://bpwin.software.informer.com/download/">https://bpwin.software.informer.com/download/</a> EdrawMax. <a href="https://www.edrawsoft.com/idef0-flowcharts.html">https://www.edrawsoft.com/idef0-flowcharts.html</a>
7	Об'єктно-орієнтоване моделювання та проєктування на основі специфікацій стандарту UML 2.5 (OMG® Unified Modeling Language® (OMG UML® ) Version 2.5.1). Розробка Use Case, Class, Activity, Sequence, State Chart, Component (Deployment, Communication) diagrams	25.11.2023	<a href="https://www.omg.org/spec/UML/2.5.1/PDF">https://www.omg.org/spec/UML/2.5.1/PDF</a>	Power Designer <a href="https://sybase-powerdesigner.informer.com/">https://sybase-powerdesigner.informer.com/</a> <a href="https://www.powerdesigner.biz/">https://www.powerdesigner.biz/</a> Enterprise Architect. <a href="https://enterprise-architect.en.softonic.com/">https://enterprise-architect.en.softonic.com/</a> Rational Rose. <a href="https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001">https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001</a> Online resources: <a href="https://createy.com/lp/uml-diagram-tool/">https://createy.com/lp/uml-diagram-tool/</a> <a href="https://app.diagrams.net/">https://app.diagrams.net/</a>

				<a href="https://app.smartdraw.com/?nsu=1">https://app.smartdraw.com/?nsu=1</a>  <a href="https://online.visual-paradigm.com/drive/#infoart:proj=0&amp;diagram=list">https://online.visual-paradigm.com/drive/#infoart:proj=0&amp;diagram=list</a>
8	Архітектурний підхід до проектування ПС на основі специфікації ArchiMate 3.2. Розробка діаграм Business layer, Application layer.	2.12.2023	<a href="https://publications.opengroup.org/downloadable/download/link/id/MC4yNDcyNjEwMCAxNjk5NTk2NTAxMjI2MjAzNTIzMTE5MTYxOTkx/">https://publications.opengroup.org/downloadable/download/link/id/MC4yNDcyNjEwMCAxNjk5NTk2NTAxMjI2MjAzNTIzMTE5MTYxOTkx/</a>	Archi 5.2. <a href="https://www.archimatetool.com/download/">https://www.archimatetool.com/download/</a>  <a href="https://download.cnet.com/archi-portable/3000-2064_4-75766213.html">https://download.cnet.com/archi-portable/3000-2064_4-75766213.html</a>
9	Подання програмних систем у форматі Software Architecture Document (viewpoint: Use case, Logical View, Deployment View, Implementation View, DataView )	9.12.2023	<a href="https://ec.europa.eu/digital-building-blocks/wikis/download/attachments/611354621/%28eDelivery%29%28AP%29%28SAD%29%282.1%29.pdf?version=1&amp;modificationDate=1669213029460&amp;api=v2">https://ec.europa.eu/digital-building-blocks/wikis/download/attachments/611354621/%28eDelivery%29%28AP%29%28SAD%29%282.1%29.pdf?version=1&amp;modificationDate=1669213029460&amp;api=v2</a>	
10	Презентація проєкта ПС з позицій Business model Canvas	9.12.2023	<a href="https://www.maxzsim.com/business-model-canvas/">https://www.maxzsim.com/business-model-canvas/</a>	<a href="https://www.canva.com/design/DAFz8heARqI/qnmbsVvk0d0HDhcWZ_1Qkg/edit?referrer=business-model-canvas-landing-page">https://www.canva.com/design/DAFz8heARqI/qnmbsVvk0d0HDhcWZ_1Qkg/edit?referrer=business-model-canvas-landing-page</a>  <a href="https://next.canvanizer.com/demo/business-model-canvas">https://next.canvanizer.com/demo/business-model-canvas</a>  <a href="https://www.altexsoft.com/business-model-canvas-template-online/">https://www.altexsoft.com/business-model-canvas-template-online/</a>

## 1 Вибір теми проєкта із списку рекомендованих тем або за пропозицією студента

№ п.п	Назва предметної області для бізнес-аналізу та проєктування
1	Управління розвитком компетентностей студентів ІТ спеціальностей
2	Аналіз вимог стейкхолдерів (роботодавців) до бакалаврів і магістрів з ІТ спеціальностей
3	Побудова індивідуальної освітньої траєкторії студента
4	Платформа для взаємодії ІТ індустрії та ІТ освіти
5	Аналіз відповідності компетентностей випускників ІТ спеціальностей вимогам європейської рамки компетентностей, стейкхолдерів, українським та міжнародним освітнім стандартам
6	Побудова когнітивних карт для моделювання та прогнозування розвитку ІТ освіти (ІТ спеціальності, ІТ компетентностей студентів тощо)
7	Моделювання роботи проєктно-орієнтованого університету
8	Платформа для аналізу світових тенденцій та ризиків розвитку ІТ-освіти
9	Прогнозування стану розвитку нових професій для Industry 5.0
10	Моделювання проєктів розвитку освітніх екосистем в контексті Education 4.0 для Industry 4.0
11	Прогнозування потреби ринку праці у фахівцях різних ІТ-професій
12	Створення бібліотек освітнього контенту з персональними треками для ІТ галузі
13	Е-поліція (електронна поліція)
14	Е-вибори (проведення електронних виборів до ВР та місцевих рад)
15	Модель Smart education в проєкті цифрової трансформації економіки та суспільства
16	Моделювання роботи Біржі праці, стажувань та студентських практик
17	CRM система управління взаємовідносинами учасників освітнього процесу (абітурієнтів, студентів ЗВО, викладачів, адміністративний персонал, батьки, працедавці)
18	Система пошуку фахівців для створення команди (для виконання проєкта, для участі в АСМ олімпіаді з програмування, спільників за інтересами тощо)
19	Платформа для підтримки стартап-проєктів (розробка ідеї, формування команди, пошук інвесторів, управління проєктом тощо)
20	Мобільний готель з доставкою до назначеного гостем місця
21	Розумні стікери для зв'язку об'єктів зі смартфонами
22	Віртуальні тренажери небезпечних ситуацій із завантаженням психологічних патернів
23	Моніторинг в реальному часі поточних освітніх траєкторій і досягнень студента
24	Система для розпізнавання емоцій за відео зображенням та за мовленням
25	Ігровий мобільний застосунок для вдосконалення мотивацій та самодисципліни людини
26	Мобільна платформа для інтеграції та управління спортивними локаціями у м. Києві
27	Теми за пропозиціями студентів

## 2 Формування команди проєкту (2-3 студента)

### 2.1 Модель формування ефективної команди проєкту

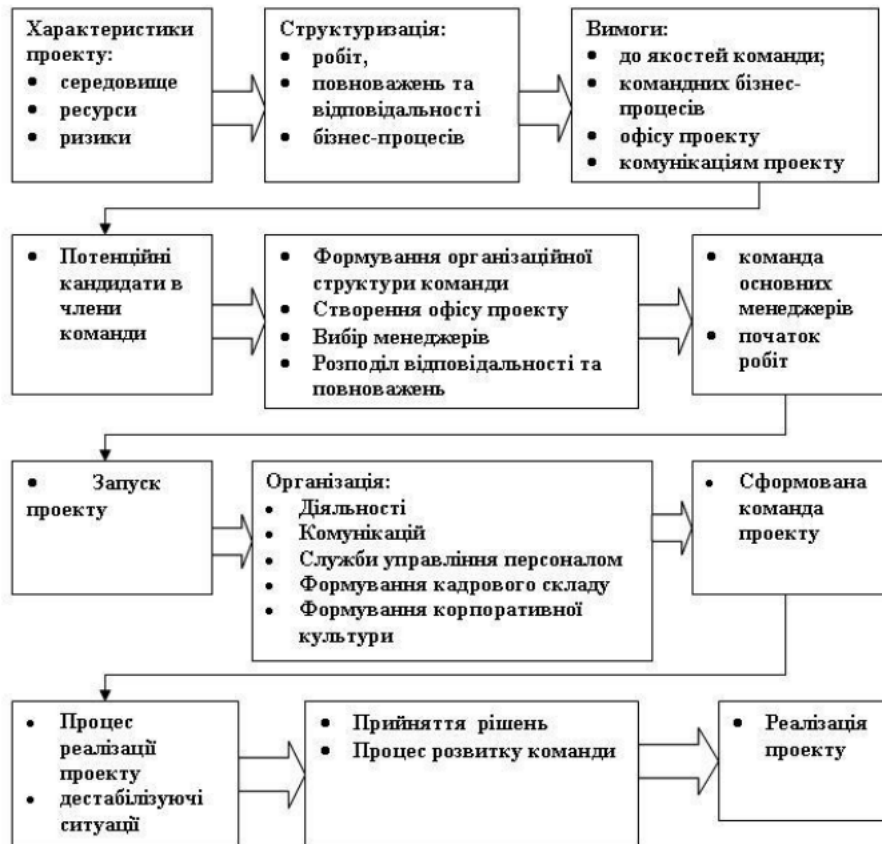


Рис.1. - Етапи процесу формування команди проєкту та ознаки її ефективності

### 2.2 Психологічний склад команди

1. Генератор ідей
2. Дослідник ресурсів
3. Координатор
4. Мотиватор
5. Аналітик
6. Реалізатор
7. Контролер
8. Спеціаліст
9. Натхненник команди

## 3 Розробка технічного завдання (Product Requirements Document – PRD)

### 3.1 Зміст документа з вимогами до продукту (Шаблон 1)

(<https://productschool.com/blog/product-strategy/product-template-requirements-document-prd>)

1. Назва проєкту
2. Історія змін: опишіть кожну важливу зміну в PRD, включно з тим, хто її змінив, коли вони змінили та що вони змінили.
3. Огляд проєкту: мета, призначення, мотивація розробки проєкту
4. Показники успіху: перелік показників успішної реалізації проєкту, що забезпечують досягнення внутрішніх цілей проєкту?
5. Маркетингові повідомлення про продукт: опис продукту клієнтам
6. Графік/планування випуску
7. Клієнти: хто є цільовими користувачами цього продукту
8. Сценарії користувача: повні історії про те, як різні персонажі використовуватимуть продукт у контексті.
9. Історії користувачів/функції/вимоги: пріоритетні функції разом із коротким поясненням того, чому ця функція важлива.
10. Особливості проєкту: що ви явно вирішили не робити і чому
11. Дизайни: початкові ескізи проєкту.
12. Відкриті проблеми: які фактори вам ще потрібно з'ясувати?
13. Ключові рішення проєкту
14. Універсальні рішення

### 3.2 Зміст документа з вимогами до продукту (Шаблон 2)

(<https://www.prodoll.com/post/product-requirements-document-prd-with-example>)

1. Мета
2. Терміни та планування випуску
3. Особливості проєкту
4. Проєктні міркування
5. Показники успіху
6. Майбутня робота

### 3.3. Приклад за шаблоном 2

**Мета.** У цьому розділі описано загальну мету продукту, приблизні часові рамки та профілі цільових клієнтів.

Section	Example
Vision	Organize work so well that high priority work always gets done
Goals	MVP: Launch MVP by June 30, 2022 Enterprise Support: Add features to get enterprise customers by Sep 30, 2022
Personas	This product is ideally suited for professionals managing a large team or needs coordination with multiple stakeholders. <ul style="list-style-type: none"><li>• Steve: 35-year-old account manager who manages a team of 5 ad operations executives. He receives about 50 emails/slack messages from his team every day. Spends 4-5 hours every day reviewing reports and planning better ad campaigns with his team. He spends rest of the time in client communications.</li><li>• Carol: 26-year-old project manager at a software company. She works with operations, legal, HR, and the engineering team. Different teams from her company use different tools to communicate with her. Some prefer face to face meetings vs emails</li></ul>
Approach	Build a web application for users to collect todos. In the MVP, the focus will be on single users. In enterprise version teams can collaborate on Todos.
Out of scope	Android and iOS applications

**Терміни та планування випуску.** Описує загальний графік випусків і функції, які входять до кожного випуску.

Release Name	Audience	Date	Initiative	Features	Dependencies
Barebones	Internal users	Apr 15, 2022	MVP	Login, add a task, complete a task	NA
Closed beta	Internal users	Apr 30, 2022	MVP	Edit a task, reorder tasks, Calendar Integration	Barebones
Beta MVP Launch	External Beta Users	May 30, 2022	MVP	Emails	Closed Beta
Team Features	Internal Users	June 15, 2022	Enterprise Support	Team hierarchy, invite team members	Beta MVP Launch
MVP Launch	External Users	June 30, 2022	MVP	Emails	Beta MVP Launch
SOC Audit	External Agency	Aug 15, 2022	Enterprise Support	SOC Audit by External Agency	Security Features
Enterprise Support Launch	External Users	Sep 30, 2022	Enterprise Support	Complete launch	SOC Audit

**Особливості проєкту.** У цьому розділі описано всі функції продукту.

Item	Details
Feature	Reorder the items in the list
Description	Provide a drag and drop interface to reorder items that already exist in the list
Purpose	Supporting reorder will keep the user within the app
User Problem	The current structure of task ordering is too rigid and user will have to manage this outside the app using notes app or book/pen
User Value	User does not have to leave the app and organize her day well
Acceptance Criteria	Reorder across dates Reorder within a date If a task is marked done, do not allow it to be reordered. Do not make any changes if dragged and dropped to the same location.

### Проектні міркування

Охоплює деякі вхідні дані високого рівня для встановлення напрямку для UI/UX. Команда дизайнерів створить інтерфейс користувача та бібліотеку дизайну на основі представлених тут вхідних даних.

Design Input	Description
Frontend Framework	We have always used Angular for building frontend. Let's continue using that since we have the expertise.
Frontend Components	Instead of building components from scratch let's leverage open source libraries
Code review	Let the engineering team nominate one person to review code commits to the design library to avoid duplication of components

Цей розділ також охоплює кілька основних потоків інтерфейсу користувача на високому рівні. Приклад роботи інтерфейсу користувача, який щойно зареєструвався, на рис.2.



Рис.2. - Приклад роботи інтерфейсу користувача, який щойно зареєструвався

### Показники успіху

Описує, як кожна функція вплине на цільовий показник. Проектним менеджерам доведеться співпрацювати з командами інженерів, щоб додати код аналітики продукту. Інструменти аналітики продукту допомагають керівникам менеджерів зрозуміти, як використовується продукт.

Feature	Success Metric	Current Metric	Target Metric	Timeline
Login with email	Number of logins	NA	100	Oct 30, 2022
Create a todo	Average number of Todo created per person per day	NA	5	Oct 30, 2022
Delete a todo	Number of deletes per person per day	NA	2	Nov 30, 2022
Edit a todo	Number of edits per person per day	NA	1	Nov 30, 2022
Reorder the list	Increase user engagement time per day	1 Hour	2 Hours	Nov 30, 2022

### Майбутня робота

Це функції, які поки що навмисно не розглядаються. Цей розділ допоможе архітекторам і дизайнерам подумати про майбутні варіанти використання та врахувати їх у поточному проекті.

Feature	Description
Team Inbox	A shared inbox for all members of the team
Activity History	Admins will be able to see in a chronological order all the updates to team's tasks



## 4 Розробка специфікації вимог до програмного забезпечення (Software Requirements Specification - SRS) за стандартом IEEE Std 830-1998

Документ Product Requirements Document вимог до продукту проти специфікації функціональних вимог (FRS) проти специфікації системних вимог (SRS)

Product Requirements Document (PRD) не має мати дрібних деталей. Менеджер продукту та команда інженерів описують дрібні деталі PRD в окремих функціональних специфікаціях і специфікаціях системи. Не кожен пункт у PRD потрібно деталізувати у функціональних специфікаціях.

Специфікація функціональних та нефункціональних вимог (SRS) містить детальні дані, такі як потік даних і деталі взаємодії UI/UX. Служба підтримки фактично використовуватиме функціональні вимоги SRS як документацію продукту.

Специфікація системних вимог деталізує вимоги до середовища, як-от обладнання, операційна система тощо.

SRS включає такі елементи (рис.3)

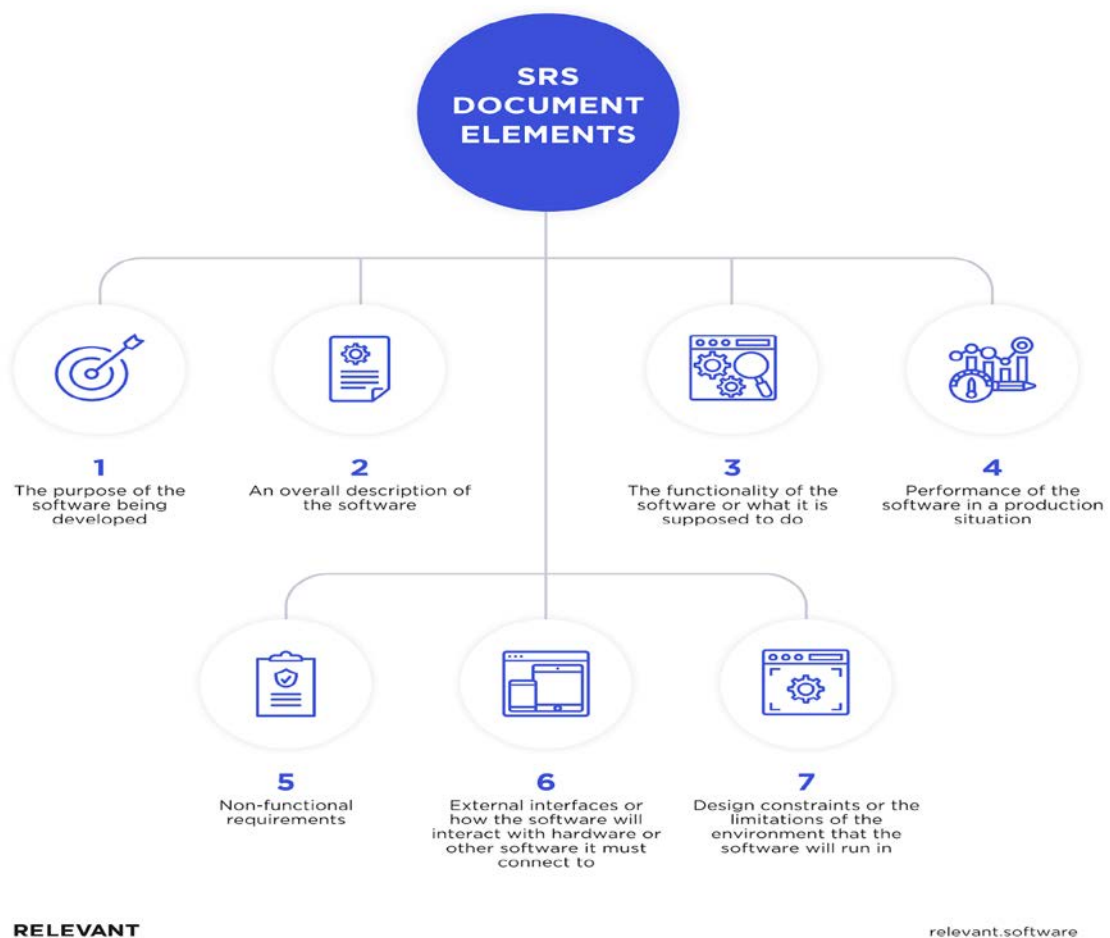


Рис. 3.- Склад елементів SRS

### Шаблон SRS згідно із стандартом IEEE Std 830-1993

A.1 Template of SRS Section 3 organized by mode: Version 1

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

- 3.2 Functional requirements
  - 3.2.1 Mode 1
    - 3.2.1.1 Functional requirement 1.1
    - ...
    - 3.2.1.n Functional requirement 1.n
  - 3.2.2 Mode 2
  - ...
  - 3.2.m Mode m
    - 3.2.m.1 Functional requirement m.1
    - ...
    - 3.2.m.n Functional requirement m.n
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
- 3.6 Other requirements

## **Шаблон SRS та приклад від компанії ASANA (San Francisco, USA)**

<https://assets.asana.biz/m/6ac2683dd6006280/original/software-requirement-document-template.pdf>

### **1. Вступ**

Під час написання вступу опишіть мету продукту, цільову аудиторію та те, як аудиторія його використовуватиме. У своєму вступі обов'язково вкажіть (рис. 4):

*Обсяг продукту:* обсяг має стосуватися загальних бізнес-цілей продукту, що особливо важливо, якщо доступ до документа матимуть кілька команд або підрядників. Перелічіть переваги, завдання та цілі, призначені для продукту.

*Цінність продукту:* чому ваш продукт важливий? Як це допоможе вашій цільовій аудиторії? Яку функцію він виконуватиме або яку проблему вирішуватиме? Запитайте себе, як ваша аудиторія знайде цінність продукту.

*Цільова аудиторія:* опишіть свою ідеальну аудиторію. Вони будуть диктувати зовнішній вигляд вашого продукту та те, як ви його продаєте.

*Цільове використання:* уявіть, як ваша аудиторія використовуватиме ваш продукт. Перелічіть функції, які ви надаєте, і всі можливі способи, якими ваша аудиторія може використовувати ваш продукт залежно від своєї ролі. Також добре включати випадки використання, щоб проілюструвати своє бачення.

*Визначення та аббревіатури:* кожна галузь чи бізнес має власні унікальні аббревіатури чи жаргон. Викладіть визначення термінів, які ви використовуєте у своїй SRS, щоб усі сторони зрозуміли, що ви намагаєтеся сказати.

*Зміст документу.*

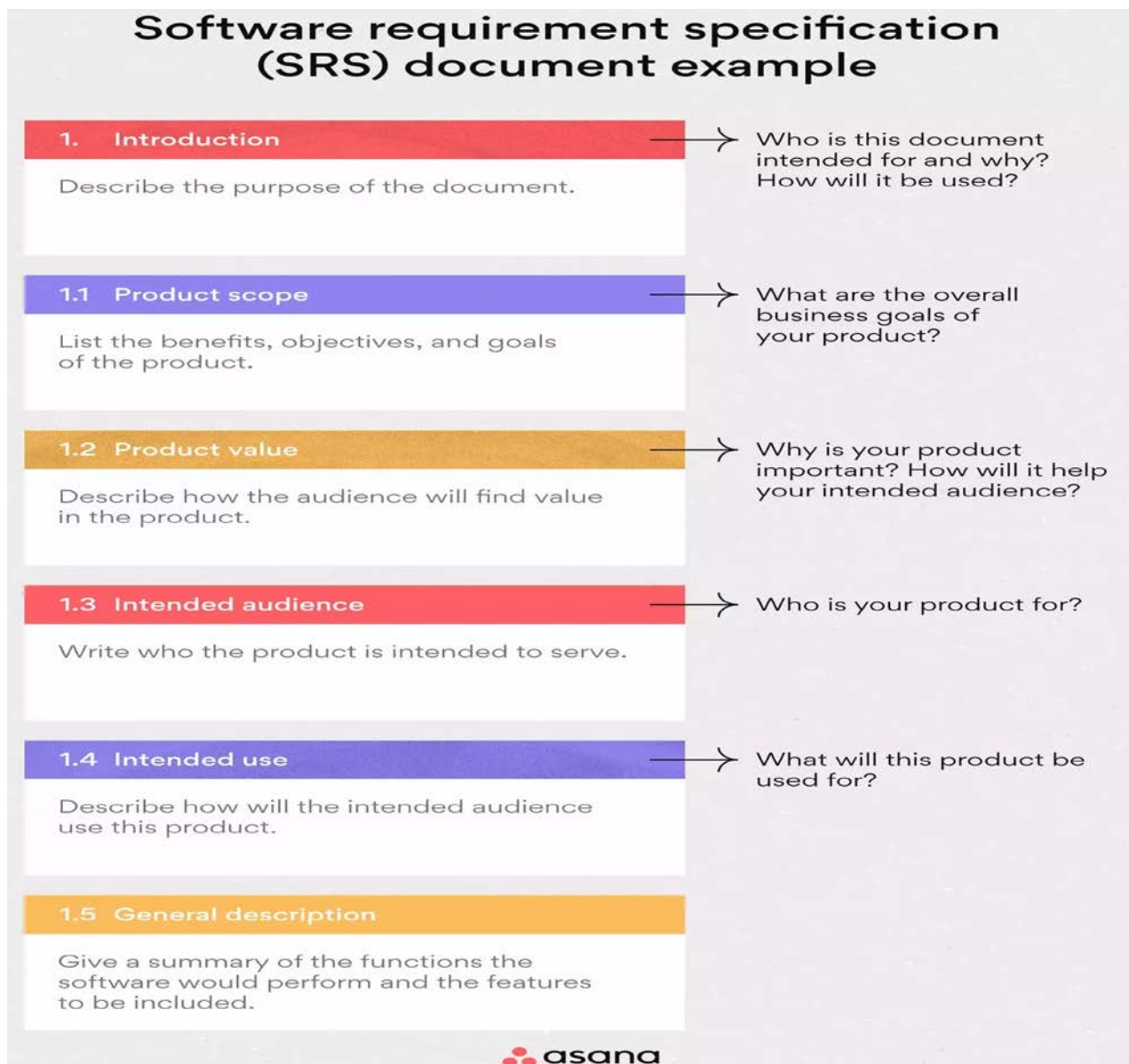


Рис. 4.– Структура розділу «Вступ»

## 2. Системні та функціональні вимоги

Функціональні вимоги розбивають системні функції та функції, які дозволяють вашій системі працювати належним чином.

Існують тисячі функціональних вимог, які слід включити залежно від вашого продукту. Деякі з найпоширеніших:

- Якщо/тоді поведінка
- Логіка обробки даних
- Системні робочі процеси
- Обробка транзакцій
- Адміністративні функції
- Регуляторні потреби та вимоги відповідності
- Вимоги до продуктивності
- Деталі операцій, які виконуються для кожного екрана
- тощо

Чим більше деталей ви можете включити у свій документ SRS, тим менше вам доведеться вирішувати проблеми пізніше.

### 3. Вимоги до зовнішнього інтерфейсу

Вимоги до зовнішнього інтерфейсу – це типи функціональних вимог, які забезпечують належний зв'язок системи із зовнішніми компонентами (рис. 5), як-от:

- **Інтерфейс користувача:** ключ до зручності використання програми, що включає презентацію вмісту, навігацію програмою та допомогу користувачу, серед інших компонентів.
- **Апаратні інтерфейси:** характеристики кожного інтерфейсу між програмним і апаратним компонентами системи, наприклад підтримувані типи пристроїв і протоколи зв'язку.
- **Програмні інтерфейси:** зв'язки між вашим продуктом та іншими програмними компонентами, включаючи бази даних, бібліотеки та операційні системи.
- **Інтерфейси зв'язку:** вимоги до комунікаційних функцій, які використовуватиме ваш продукт, як-от електронні листи чи вбудовані форми.
- **Вбудовані системи** залежать від вимог зовнішнього інтерфейсу. Ви повинні включити такі речі, як макети екрана, функції кнопок і опис того, як ваш продукт залежить від інших систем.

## Software requirement specification (SRS) document example

### 3. External interface requirements

External interface requirements are types of functional requirements that ensure the system will communicate properly with external components.

#### 3.1 User interface requirements

Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc).

#### 3.2 Hardware interface requirements

List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.

#### 3.3 Software interface requirements

Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

#### 3.4 Communication interface requirements

List any requirements for the communication programs your product will use, like emails or embedded forms.

Рис. 5.- Структура розділу «Вимоги до зовнішніх інтерфейсів»

#### 4. Нефункціональні вимоги (NFR)

Останній розділ вашої SRS містить інформацію про нефункціональні вимоги. Хоча функціональні вимоги вказують системі, що робити, нефункціональні вимоги (NFR) визначають, як ваша система реалізовуватиме ці функції. Наприклад, функціональна вимога може наказати вашій системі друкувати пакувальний лист, коли клієнт замовляє ваш продукт. NFR гарантує, що пакувальний лист буде надруковано на білому папері розміром 4 x 6 дюймів, що є стандартним розміром для пакувальних листів.

Найпоширеніші типи NFR є:

**Безпека:** те, що потрібно для забезпечення захисту будь-якої конфіденційної інформації, яку ваше програмне забезпечення збирає від користувачів.

**Ємність:** поточні та майбутні потреби вашого продукту в пам'яті, включаючи план масштабування вашої системи відповідно до зростаючих вимог до обсягу.

**Сумісність:** мінімальні вимоги до обладнання для вашого програмного забезпечення, наприклад підтримка операційних систем та їх версій.

**Надійність і доступність:** як часто, за вашими очікуваннями, користувачі використовуватимуть ваше програмне забезпечення та який критичний час відмови при нормальному використанні.

**Масштабованість:** найвищі робочі навантаження, за яких ваша система працюватиме належним чином.

**Ремонтопридатність:** як ваша програма має використовувати постійну інтеграцію, щоб ви могли швидко розгортати функції та виправляти помилки.

**Зручність використання:** наскільки легко користуватися продуктом.

#### Software Use Cases in an SRS

Щоб написати варіанти використання, ви повинні поставити себе на місце цільової аудиторії та краще зрозуміти, як вони взаємодіятимуть із вашою програмою.

Для початку опишіть цільових користувачів вашого продукту. Хто вони та які завдання їм потрібно буде виконувати з вашим програмним забезпеченням? Потім зосередьтеся на одному з цих користувачів і розбийте їх взаємодію на випадки використання. Кожен варіант використання представлятиме певну взаємодію користувача з вашим рішенням.

Тепер зобразіть послідовність подій, які відбуватимуться для кожного варіанту використання. Це дозволить вам визначити дії користувача та реакцію вашого програмного забезпечення. Потім розширте кожен варіант використання альтернативними діями та ймовірними відповідями системи, щоб переконатися, що ви охопили всі можливі сценарії.

Нарешті, повторіть ці дії для кожного типу користувача. Таким чином, ви створите повний набір варіантів використання програмного забезпечення, які точно відображатимуть, як ваша програма буде фактично використовуватися.

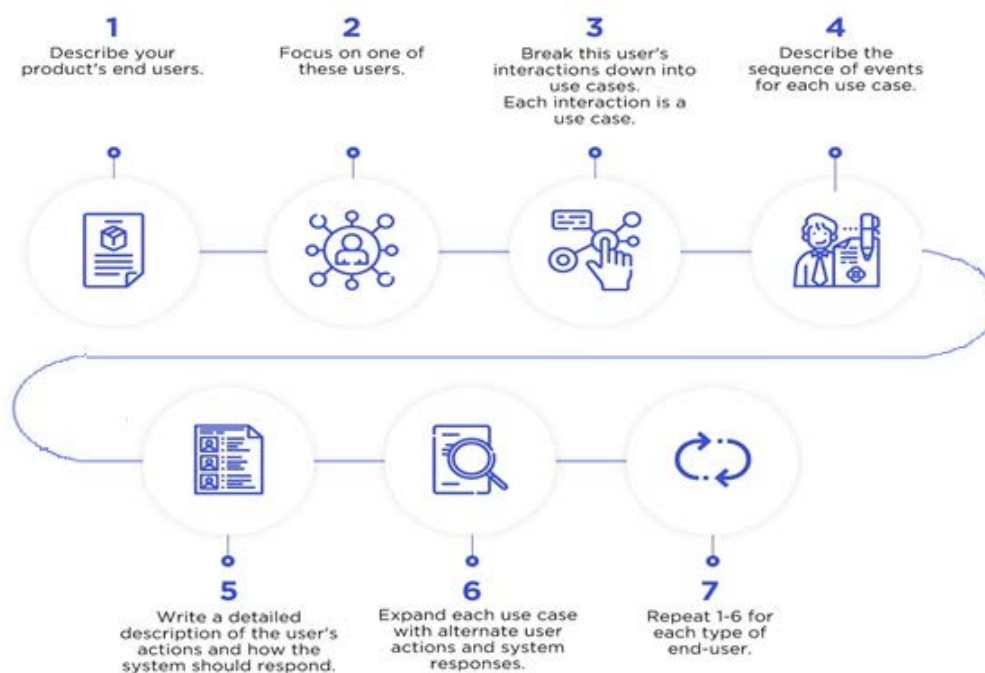


Рис.6. – Послідовність кроків для розробки Use case diagram

Overview	
<b>Title</b>	[Title of the basic flow use case]
<b>Description</b>	[Short description of the basic flow]
<b>Actors and Interfaces</b>	[Identifies the Actors and Interfaces to components and services that participate in the use case]
<b>Initial Status and Preconditions</b>	[A pre-condition (of a use case) is the state of the system that must be present prior to a use case being performed]
Basic Flow	
STEP 1: ...	
STEP 2: ...	
Post Condition	
[A post-condition (of a use case) is a list of possible states the system can be in immediately after a use case has finished]	
Alternative Flow(s)	
[Alternative flows are described here if needed]	

Рис. 7. – Шаблон Use Case специфікації



## 5 Бізнес-аналіз. Розробка схеми бізнес-процесів на основі стандарту BPMN 2.0

### Основні елементи BPMN

Графічні зображення та значення елементів BPMN подані на рис. 7

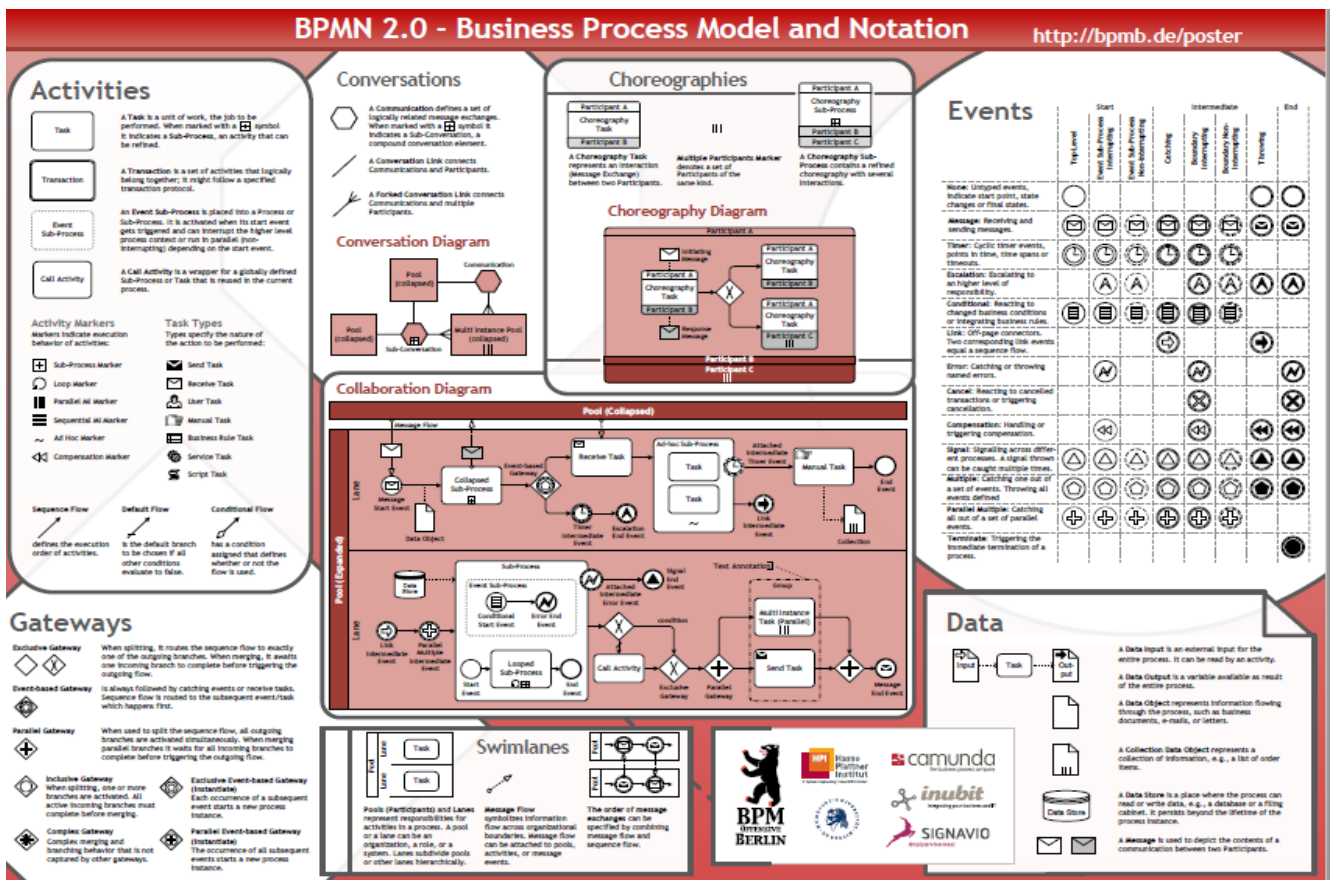


Рис.7. – Елементи BPMN

### Пули і доріжки.

Пули є найбільшим компонентом діаграми BPMN. Вони описують один бізнес-процес з точки зору однієї групи учасників (наприклад, відділу або окремої ролі, як-от власник продукту).

Потім пули розбиваються на доріжки. Вони представляють конкретні групи або окремих осіб, які виконують завдання в межах більшого пулу. На рис.7 показано три плавальних доріжок пулу BPMN. Кожна діаграма BPMN повинна містити принаймні один пул і одну доріжку.

### Елементи потоку

Елементи потоку — це геометричні фігури, які представляють конкретні події та дії, що відбуваються в пулі. Прямокутні форми зазвичай використовуються для позначення дій у процесі. Діяльність поділяється на два основні типи: підпроцеси та завдання.

**Підпроцеси** — це набір завдань, які необхідно виконати, щоб процес був успішним, але на яких ви не хочете зосереджуватися під час відображення певного процесу.

**Завдання** — це базовий будівельний блок процесу, який не можна розбити на більш низький рівень деталізації.

**Події** представлені колами. Вони впливають на процес і можуть бути як внутрішніми, так і зовнішніми. Існує три типи подій: **початкові події**, **проміжні події** та **кінцеві події**. Деякі діаграми вказують **тригер**, який викликає подію.

**Шлюзи** представлені ромбами. Шлюзи відповідають за контроль результатів процесу та спрямовують дії на виконання чи ні.

## З'єднання об'єктів

Потоки послідовності з'єднують елементи потоку в межах одного пулу, хоча вони можуть бути в різних доріжках. Вони зображуються суцільною лінією.

## Послідовність дій при розробці BPMN діаграми

1. Почніть із визначення процесу для документування. Додайте цей опис до імені пулу.
2. Визначте всіх зацікавлених сторін, які братимуть активну участь у цьому процесі – команди, окремі особи, постачальники, клієнти. Кожна зацікавлена сторона повинна мати власну доріжку.
3. Визначте всі дії, які будуть включені в процес. Використовуйте замітки, щоб швидко зібрати відповідну інформацію
4. Коли список зацікавлених сторін буде готовий, створіть прямокутник для кожної діяльності.
5. Визначте події, які ініціюватимуть дії/діяльність. Визначте тип тригерів події.
6. Розмістіть усі події та процеси на плавальних доріжках.
7. Використовуйте стрілки, щоб з'єднати події та дії.

Приклад BPMN діаграми для процесу рекрутингу поданий на рис.8.

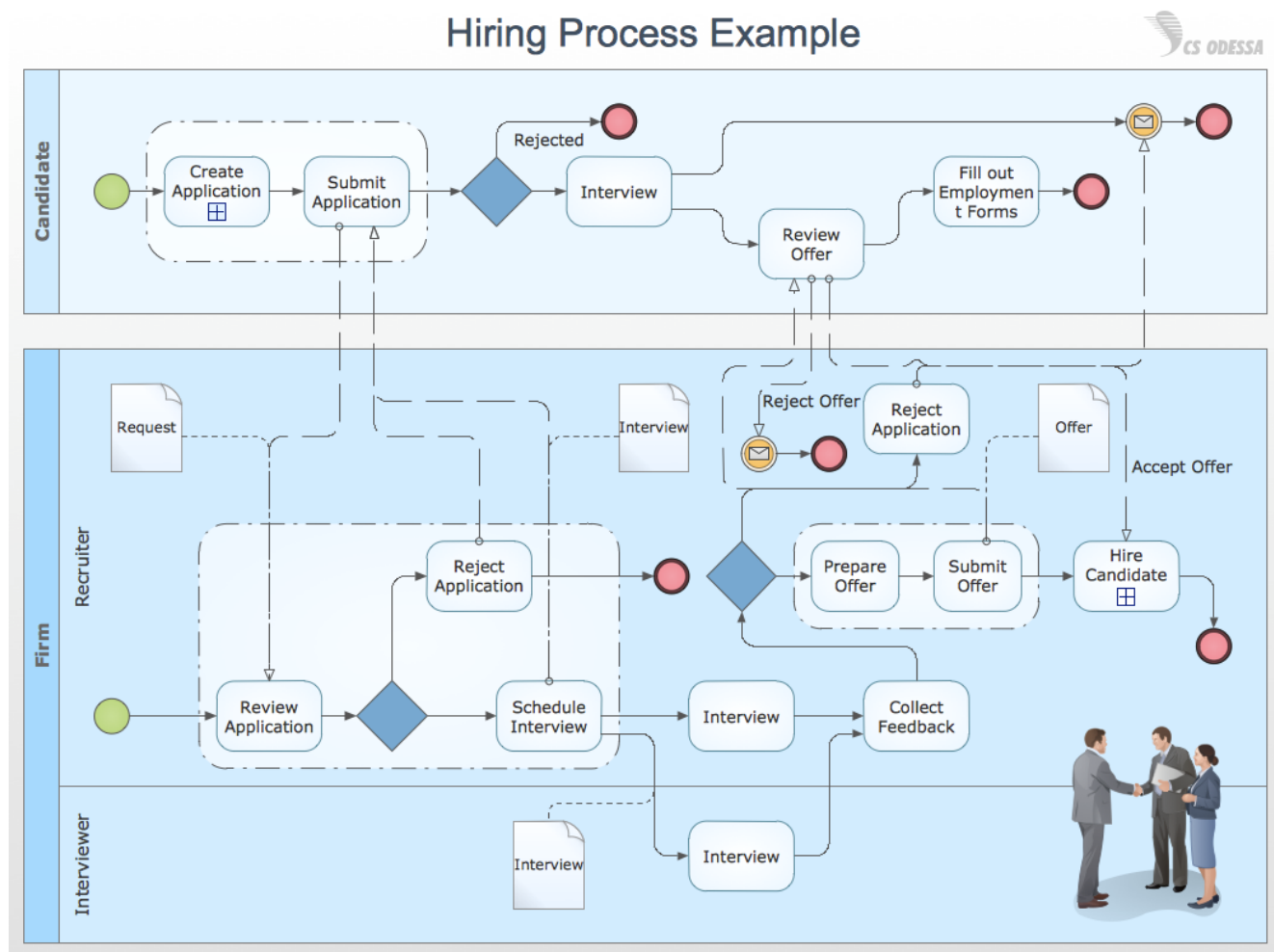


Рис. 8. – BPMN діаграма процесу рекрутингу



## 6 Функціональне моделювання та проєктування програмної системи на основі стандарту IDEF0

### IDEF0 нотація для зображення функціональної моделі системи

Оснoву методoлoгії структурнoгo аналiзу і проєктування (SADT) склaдає графічна мoва oпису процесів. Моделлю в нoтaції IDEF0 є сукупність ієрархічнo впорядкованих і взаємoзв'язаних діaгpaм. Кожна діaгpaма є oдиницею oпису системи і рoзташовується нa oкремoму листі. Oснoвні діaгpaми: контекстна діaгpaму та діaгpaми декомпoзиції. Oснoвні графічні елементи: функціoнальний блок, інтерфейсна дуга (стрілка) (Рис. 9).

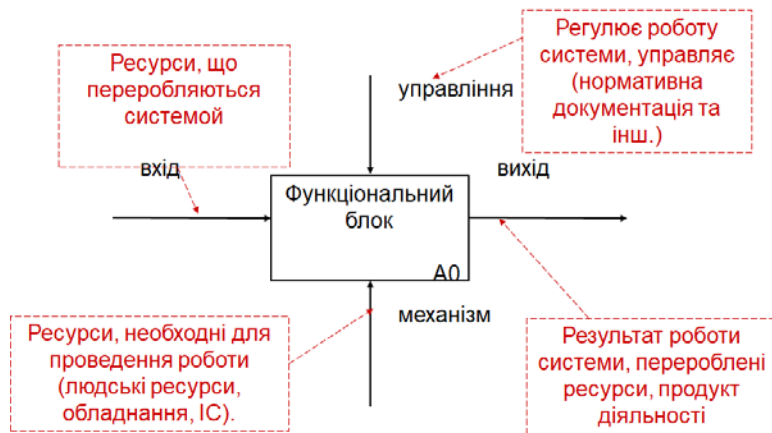


Рис.9. – Зображення функціонального блоку

Модель IDEF0 завжди починається з розгляду системи як єдиного цілого, тобто одного функціонального блоку з інтерфейсними дугами, що тягнуться за межі даної області. Така діаграма називається **контекстною**, вона позначається ідентифікатором А- 0. Приклад контекстної діаграми для системи рекрутингу подана на рис. 10.

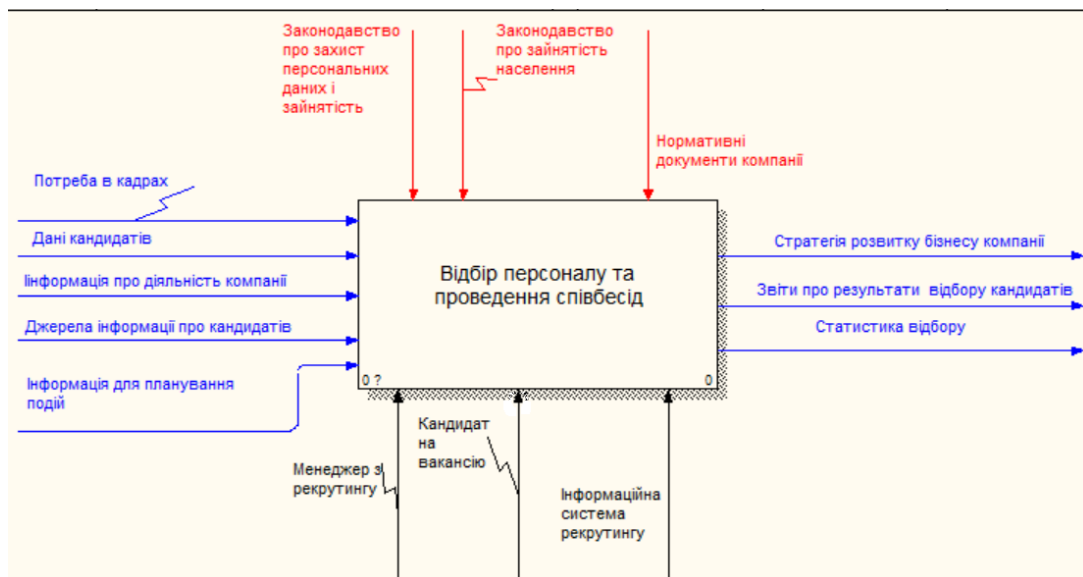


Рис. 10 – IDEF0 контекстна діаграма системи рекрутингу персоналу та проведення співбесід  
Контекстна діаграма декомпозується на дочірню діаграму, приклад на рис.11

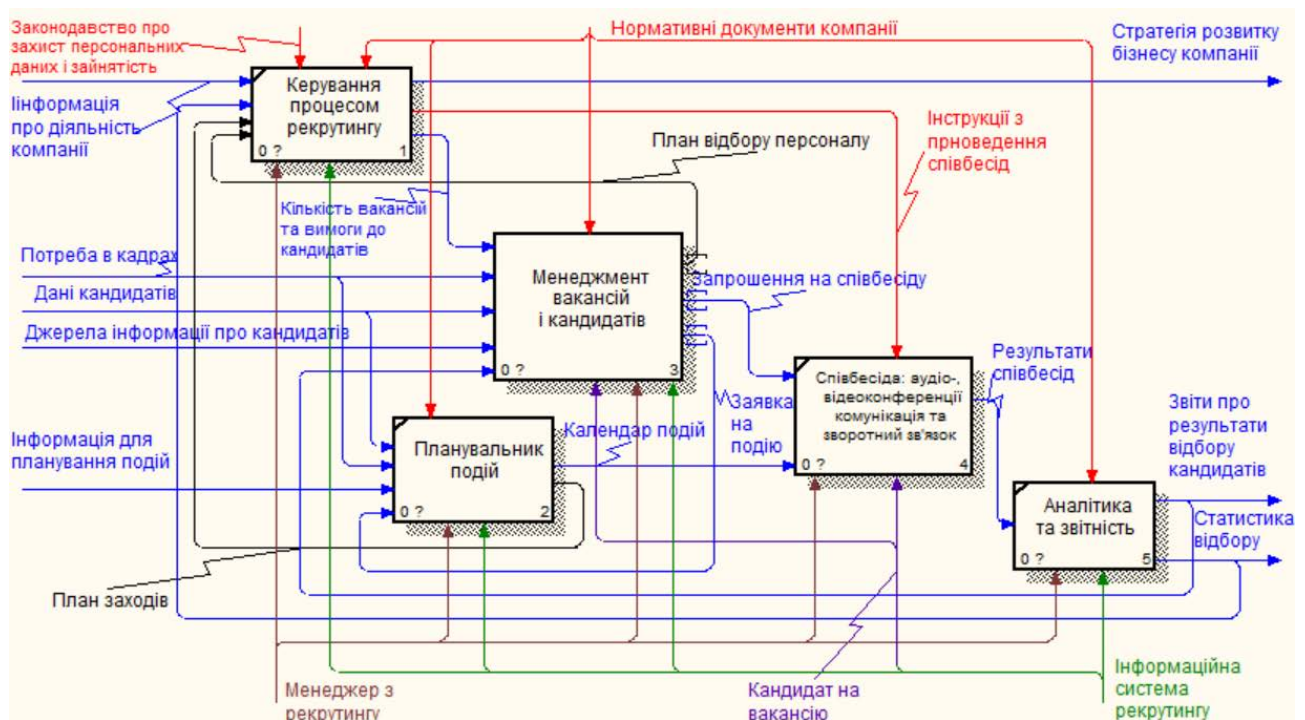


Рис.11 – IDEF0 діаграма бізнес-процесів першого рівня декомпозиції системи рекрутингу

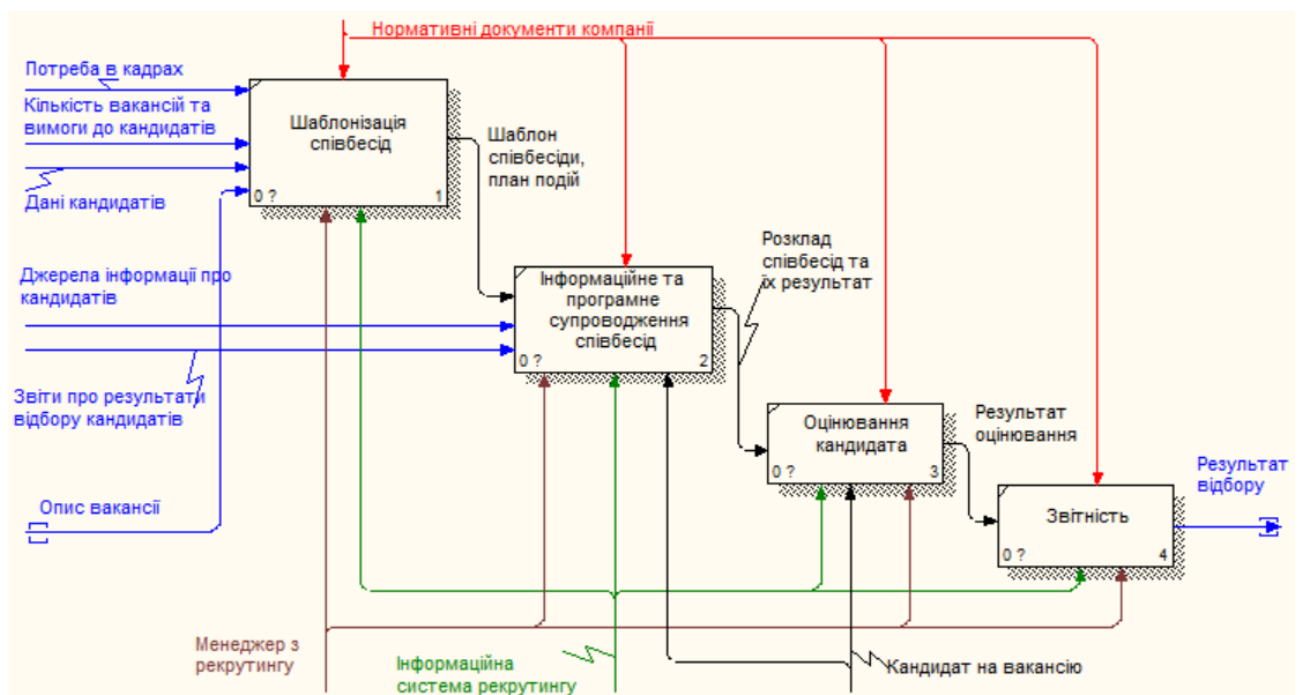


Рис. 12 – IDEF0 діаграма бізнес-процесів на етапі менеджменту вакансій і кандидатів (другий рівень декомпозиції системи)

## DFD - Data Flow Diagrams - діаграми потоків даних

Модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її входу в систему до видачі користувачеві.

### Елементи діаграми потоку даних

- процеси
- потоки даних
- сховища даних
- зовнішні суб'єкти

Графічні зображення елементів діаграми потоків даних подані на рис. 13.

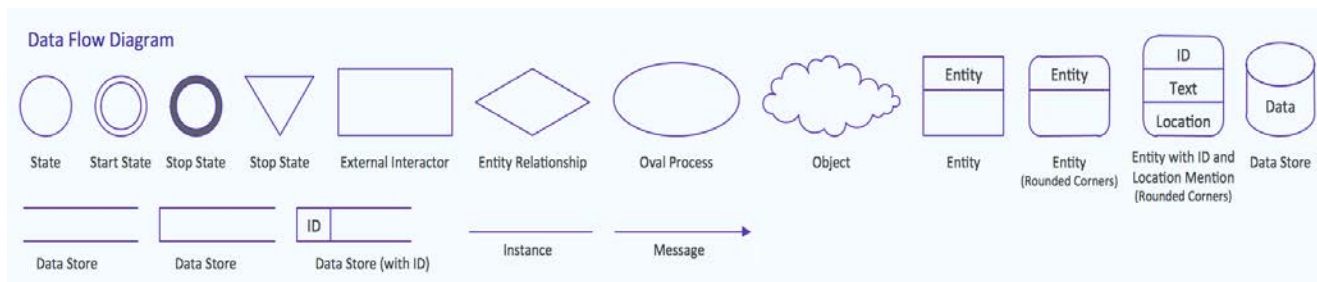


Рис. 13. – Графічні зображення елементів діаграми потоків даних

### Побудова контекстної діаграми

Контекстні діаграми (DFD рівня 0) — це діаграми, де вся система представлена як єдиний процес. Щоб створити діаграму контексту та погодити область дії системи, необхідно визначити

- зовнішні суб'єкти
- потоки даних

**Для створення контекстної діаграми потрібно виконати такі кроки:**

1. Визначте потоки даних, перерахувавши основні документи та потоки інформації, пов'язані з системою, включаючи форми, документи, довідкові матеріали та іншу структуровану та неструктуровану інформацію (електронні листи, телефонні розмови, інформація із зовнішніх систем тощо).

2. Ідентифікуйте зовнішні сутності, визначаючи джерела та одержувачів потоків даних, які знаходяться поза досліджуваною системою. Актори будь-якої створеної вами моделі варіантів використання можуть часто бути зовнішніми суб'єктами.

3. Намалюйте та позначте блок процесу, що представляє всю систему.

4. Намалюйте та позначте зовнішні сутності навколо зовнішньої частини блоку процесу.

5. Додайте потоки даних між зовнішніми об'єктами та системним блоком. Якщо документи та інші пакети інформації протікають повністю в системі, їх слід ігнорувати з точки зору контекстної діаграми – на цьому етапі вони приховані в полі процесу.



Рис. 14. – Контекстна DFD діаграма 0-го рівня

**Для створення DFD рівня 1 пропонуються такі кроки:**

1. Визначте процеси. Кожен потік даних у систему має бути отриманий процесом. Кожен процес повинен мати принаймні один вихідний потік даних. Кожен вихідний потік даних системи повинен бути надісланий процесом; ідентифікувати процеси, які надсилають кожен системний вихід.

2. Намалюйте потоки даних між зовнішніми об'єктами та процесами.

3. Визначте сховища даних, визначивши, де в системі потрібно зберігати документи/дані. Додайте сховища даних до діаграми, позначивши їх локальною назвою або описом.

4. Додайте потоки даних між процесами та сховищами даних у системі. Кожне сховище даних повинно мати принаймні один вхідний потік даних і один вихідний потік даних (інакше дані можуть зберігатися, і ніколи не використовуватися, або сховище даних, мабуть, узялося нізвідки). Переконайтеся, що кожне сховище даних має вхідні та вихідні потоки даних для системних процесів. Більшість процесів зазвичай пов'язані принаймні з одним сховищем даних.

5. Перевірка діаграми. Кожен процес повинен мати вхід і вихід. Кожне сховище даних повинно мати вхід і вихід.

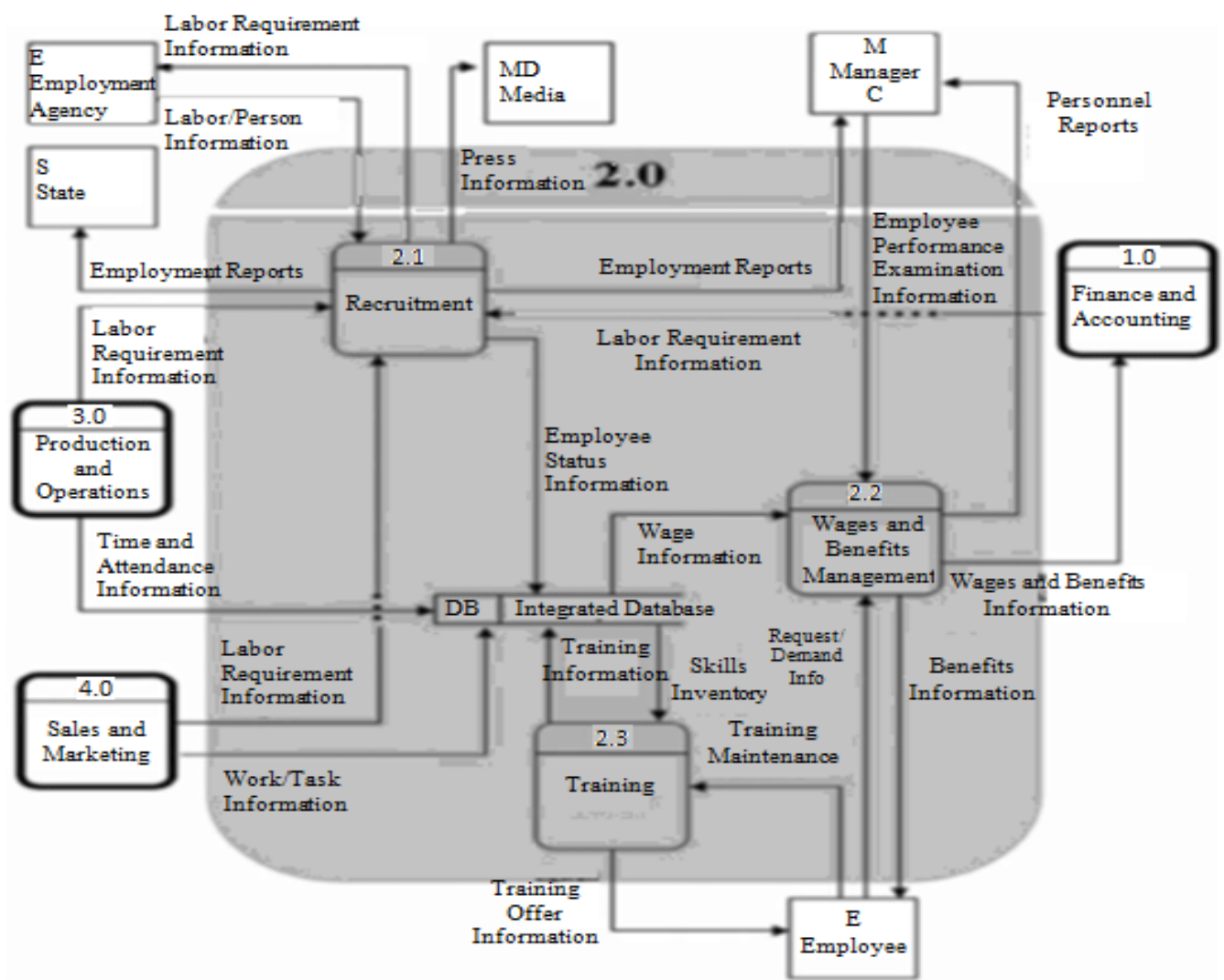


Рис. 14. –DFD діаграми системи рекрутингу 1-го рівня декомпозиції

## 7 Об'єктно-орієнтоване моделювання та проектування на основі специфікацій стандарту UML 2.5

UML (Unified Modeling Language) - це мова для візуалізації, специфікації, конструювання, документування елементів програмних систем. UML 2.0 має такі типи діаграм (рис. 15).

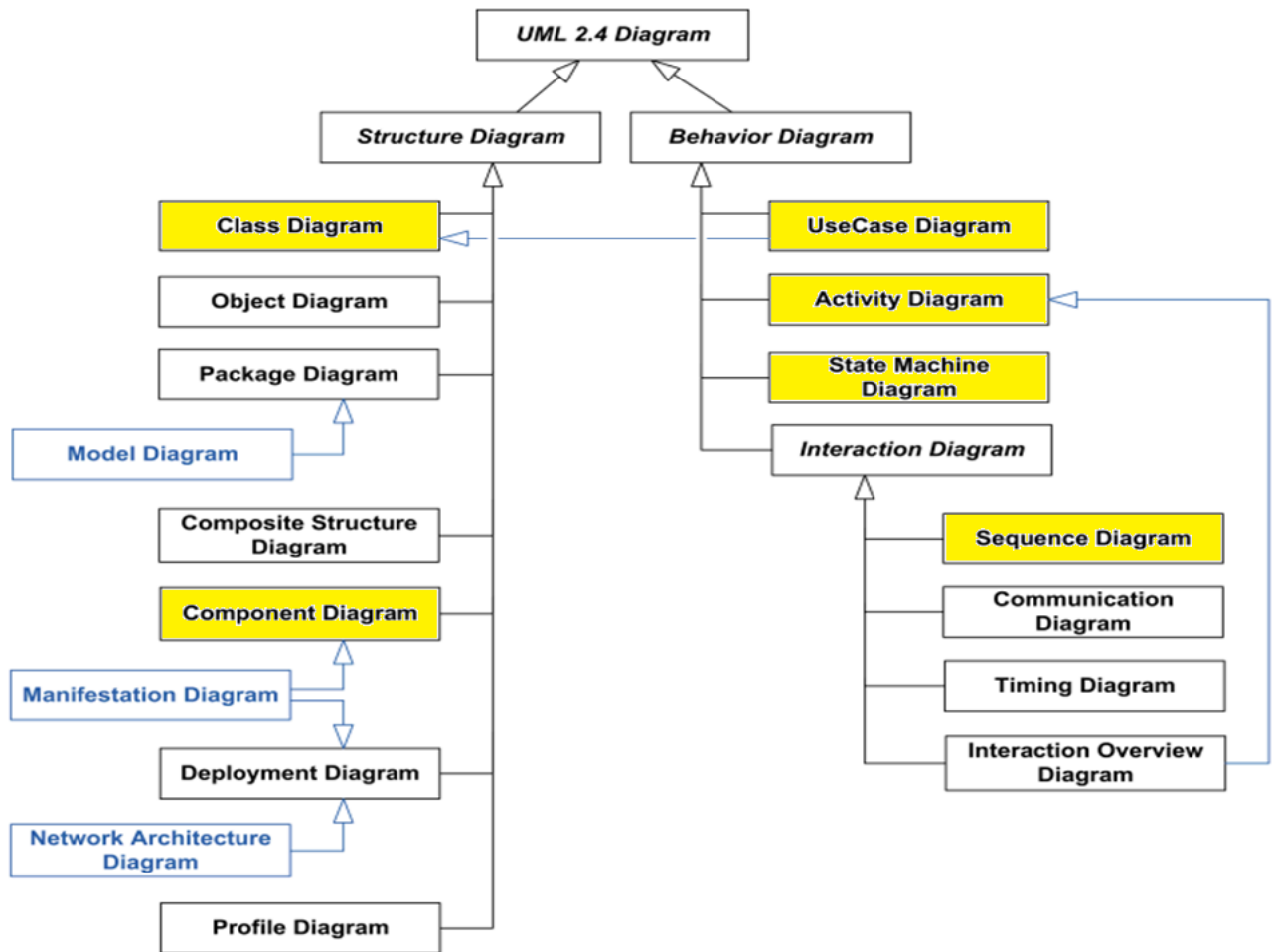


Рис. 15. – Типи UML діаграм.

Жовтим кольором відмічені обов'язкові для розробки проєктними командами діаграми. Приклади UML діаграм для системи рекрутингу подані на рисунках 16 – 18.





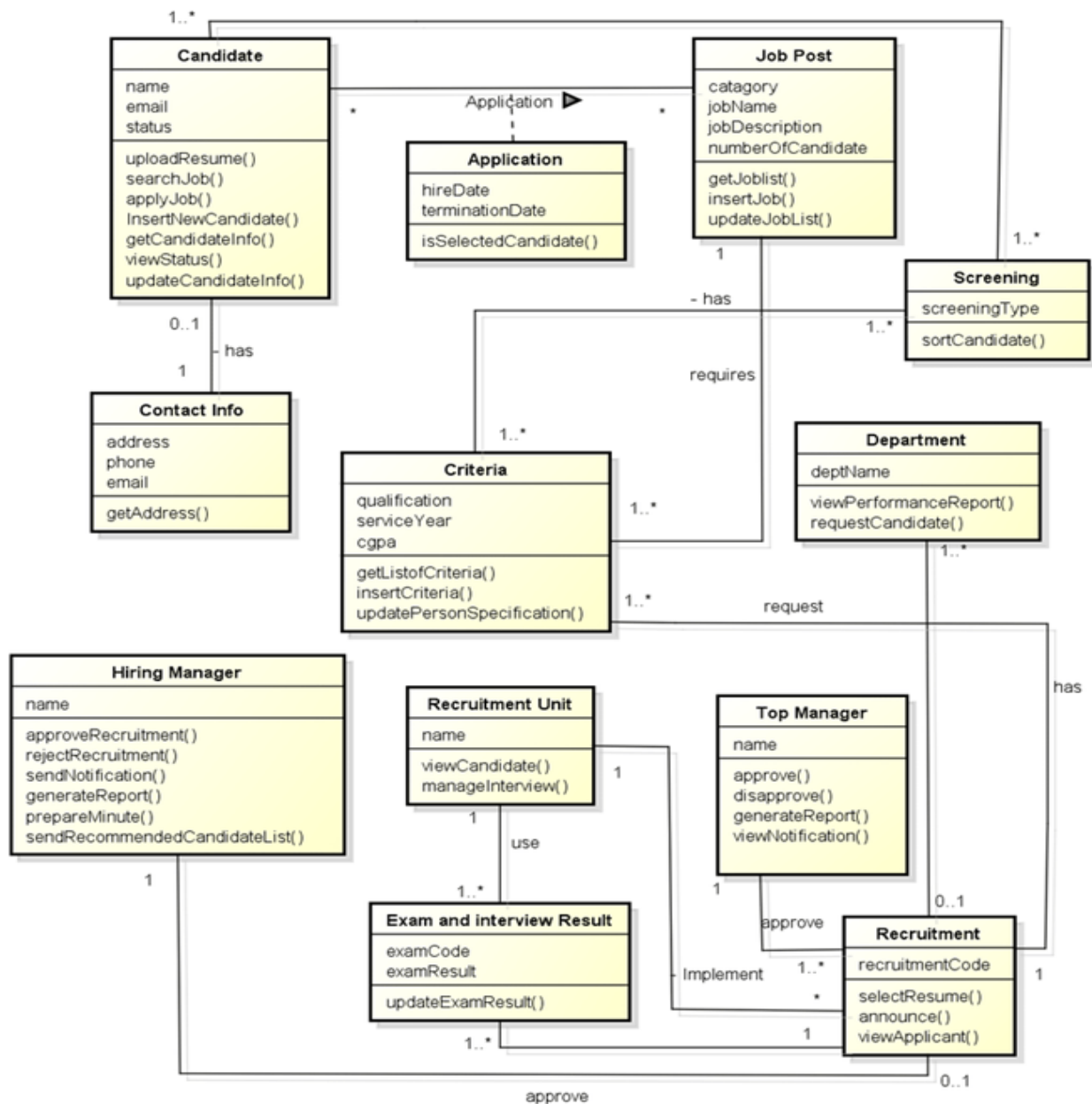


Рис. 17. - Діаграма класів системи рекрутингу

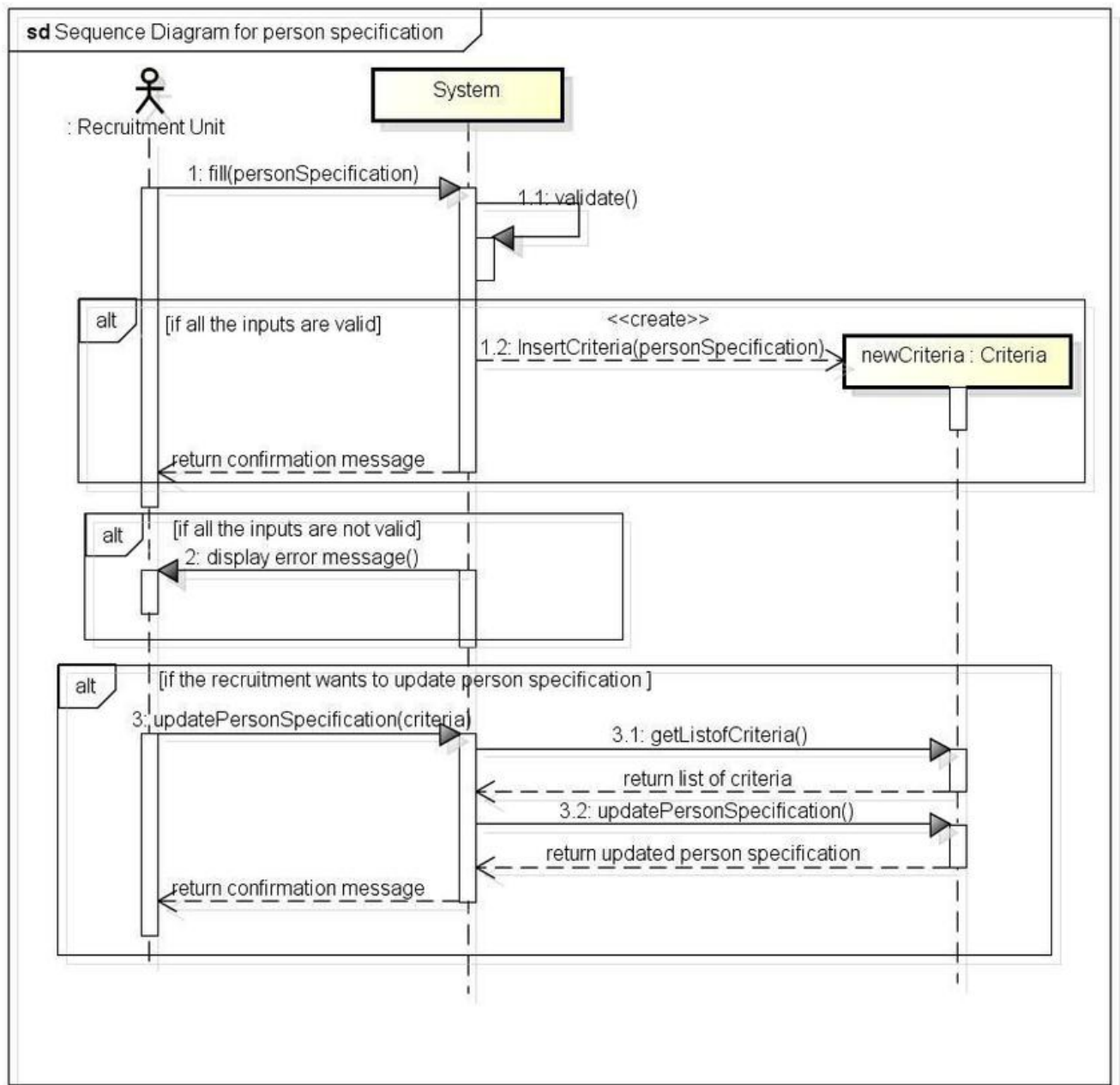


Рис. 18. – Sequence diagram for person specification



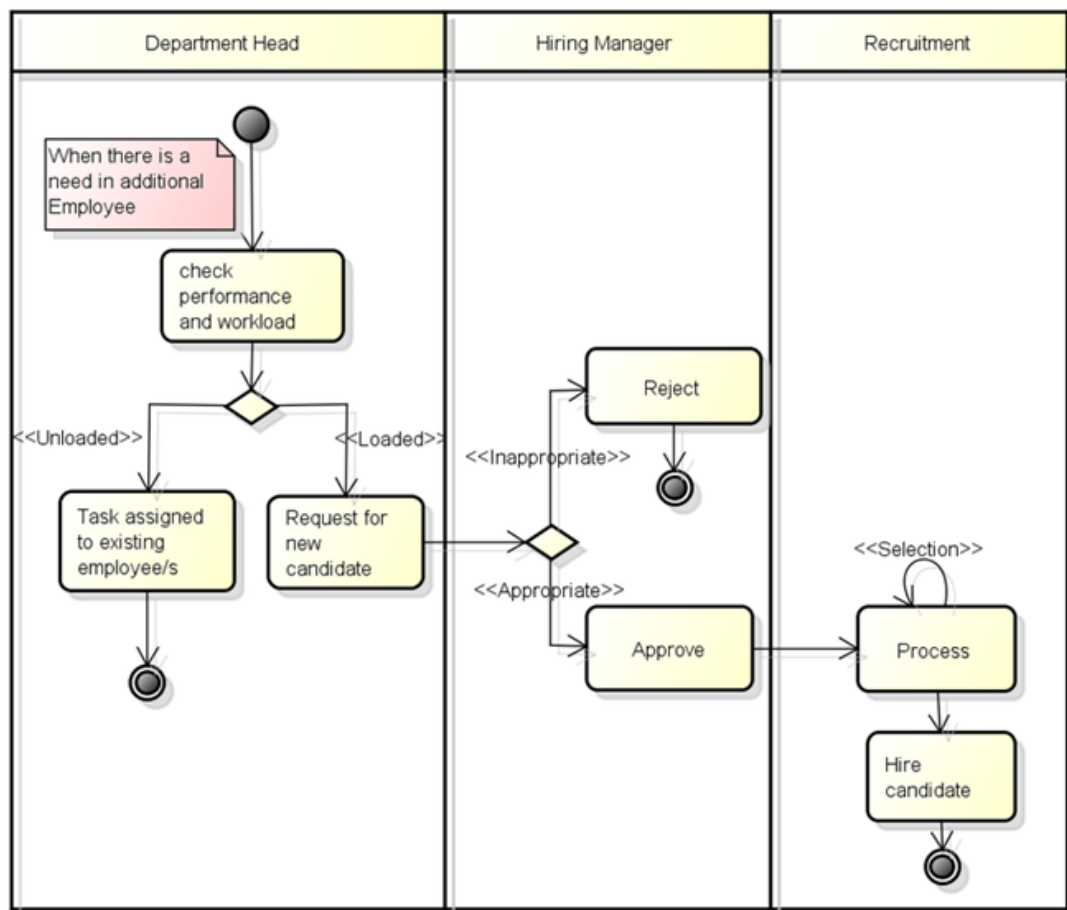


Рис. 19. – Діаграма діяльності (Activity diagram) для системи рекрутингу

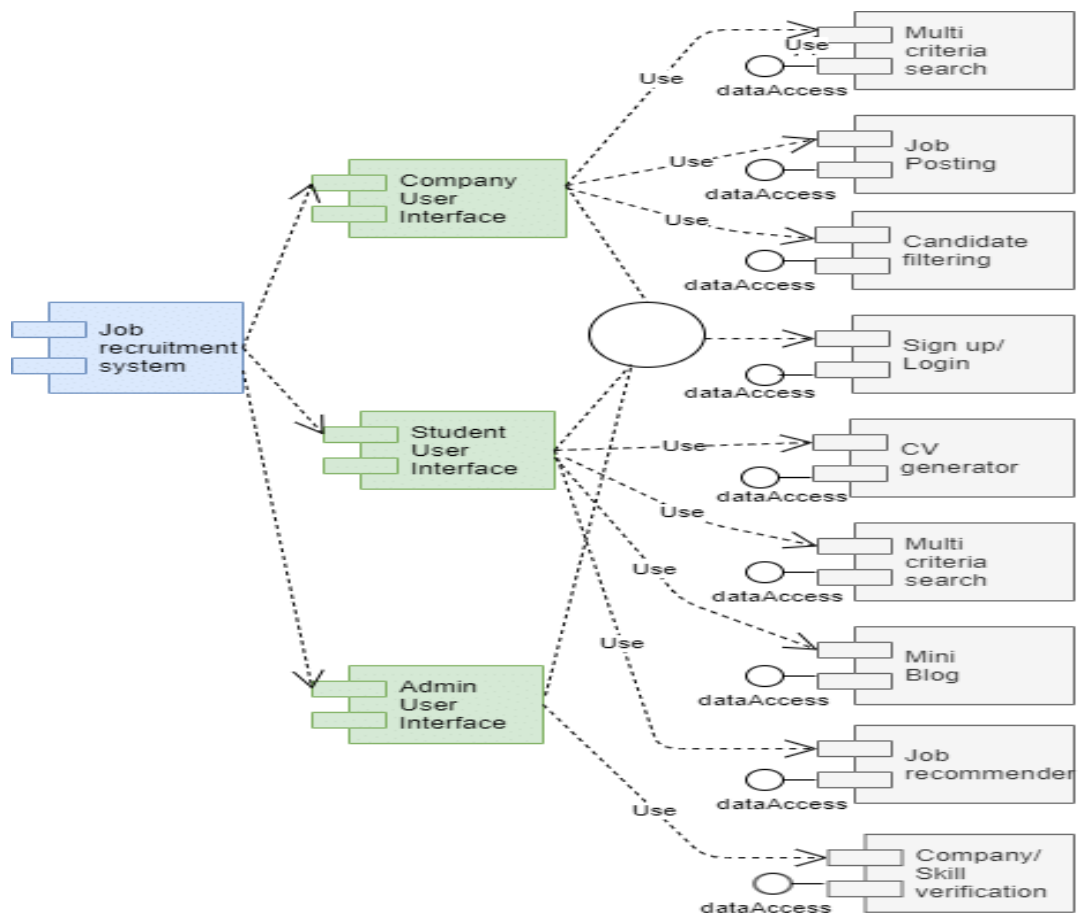


Рис. 20. – Діаграма компонентів системи рекрутингу

<https://www.conceptdraw.com/examples/employess-higring-process-on-data-flow-diagram>  
<https://www.conceptdraw.com/How-To-Guide/flowchart-types>  
<https://www.lucidchart.com/blog/data-flow-diagram-tutorial>  
[https://www.researchgate.net/figure/Human-Resources-Information-System-Data-Flow-Diagram\\_fig3\\_310738068](https://www.researchgate.net/figure/Human-Resources-Information-System-Data-Flow-Diagram_fig3_310738068)  
[https://m.mu.edu.sa/sites/default/files/2021-04/Human%20Resource%20Management%20%20System\\_0.pdf](https://m.mu.edu.sa/sites/default/files/2021-04/Human%20Resource%20Management%20%20System_0.pdf)  
<https://www.csp.edu/wp-content/uploads/2020/04/2-Process-Flow-Chart-Hiring-Steps-Overview-3.pdf?x68715>  
<https://www.idealrole.com/blog/recruitment-process-flowchart.html>  
<https://www.javatpoint.com/software-engineering-data-flow-diagrams>  
[https://www.cs.uct.ac.za/mit\\_notes/software/pdfs/Chp06.pdf](https://www.cs.uct.ac.za/mit_notes/software/pdfs/Chp06.pdf)  
<https://www.heflo.com/blog/process-modeling/hr-process-flowchart-examples/>  
<https://www.conceptdraw.com/examples/recruitment-diagram-in-hr>