

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

**ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОДУЛЯ ЕЛЕКТРОННОГО
ДОКУМЕНТООБІГУ З ДИНАМІЧНОЮ МАРШРУТИЗАЦІЄЮ
НА ПЛАТФОРМІ KSU24**

Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: здобувач 4 курсу 12-441 групи
Спеціальності: 121 Інженерія програмного
забезпечення
Освітньо-професійної програми Інженерія
програмного забезпечення
Диннік Михайло Андрійович

Керівник: кандидат технічних наук, доцент
Шишко Людмила Станіславівна

Рецензент: кандидат педагогічних наук,
доцентка кафедри загальнофахової підготовки
та морської безпеки Херсонської державної
морської академії
Зайцева Т.В.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Аналіз предметної області та існуючих рішень . . .	6
1.1. Аналіз предметної області електронного документообігу	6
1.2. Огляд існуючих систем електронного документообігу . .	7
1.3. Порівняльний аналіз існуючих рішень	9
1.4. Обґрунтування вибору інструментів розробки	10
1.5. Постановка задачі	12
РОЗДІЛ 2. Проектування модуля електронного документообігу	14
2.1. Архітектура системи	14
2.2. Проектування моделі даних	17
2.3. Проектування системи динамічної маршрутизації	20
2.4. Проектування життєвого циклу документа	21
2.5. Проектування інтеграцій із зовнішніми системами	25
РОЗДІЛ 3. Реалізація модуля електронного документообігу . .	30
3.1. Реалізація серверної частини	30
3.2. Реалізація клієнтської частини	31
3.3. Реалізація системи генерації документів	34
3.4. Реалізація електронного підпису	38
3.5. Демонстрація роботи системи	40
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ	51
Додаток А	51

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій суттєво змінює підходи до організації діловодства та документообігу в установах різного рівня. Заклади вищої освіти, як складні організаційні структури з великою кількістю внутрішніх та зовнішніх документів, потребують ефективних інструментів автоматизації роботи з документами. Щорічно університет обробляє тисячі документів: контракти зі студентами, накази, довідки, вхідну та вихідну кореспонденцію. Традиційний паперовий документообіг є повільним, непрозорим та схильним до помилок, що призводить до затримок у прийнятті рішень та ускладнює контроль виконання.

Проблема автоматизації документообігу набуває особливої актуальності в контексті цифрової трансформації освіти в Україні. Впровадження електронних підписів через застосунок «Дія», інтеграція з Єдиною державною електронною базою з питань освіти (ЄДЕБО) та перехід до безпаперових технологій створюють потребу в сучасних програмних рішеннях, здатних забезпечити повний цикл роботи з електронними документами.

Існуючі системи електронного документообігу, такі як Alfresco, Paperless-ngx або комерційні українські рішення, мають ряд обмежень при застосуванні в університетському середовищі: відсутність інтеграції з національними сервісами (Дія, ЄДЕБО), негнучкі механізми маршрутизації, неможливість налаштування під специфічні бізнес-процеси закладу освіти. Це обумовлює необхідність розробки спеціалізованого модуля документообігу з динамічною маршрутизацією, інтегрованого в існуючу університетську платформу.

Мета дослідження — проєктування та реалізація модуля електронного документообігу з динамічною маршрутизацією на платформі

KSU24 для автоматизації процесів роботи з документами у закладі вищої освіти.

Об’єкт дослідження — процес електронного документообігу у закладі вищої освіти.

Предмет дослідження — програмний модуль електронного документообігу з динамічною маршрутизацією на платформі KSU24.

Завдання дослідження. Для досягнення поставленої мети необхідно вирішити такі завдання:

- а) проаналізувати предметну область електронного документообігу та існуючі програмні рішення;
- б) визначити функціональні та нефункціональні вимоги до модуля документообігу;
- в) спроектувати архітектуру модуля, модель даних та систему динамічної маршрутизації;
- г) реалізувати серверну та клієнтську частини модуля;
- д) забезпечити інтеграцію з зовнішніми системами (Дія, ЄДЕБО, Google Drive);
- е) продемонструвати роботу розробленого модуля.

Методи дослідження. У роботі використано такі методи: аналіз та порівняння існуючих рішень для обґрунтування вибору підходу; об’єктно-орієнтоване проектування для побудови архітектури системи; моделювання за допомогою UML-діаграм для візуалізації структури та поведінки системи; метод декомпозиції для розбиття складної задачі на підзадачі.

Практична значущість. Розроблений модуль електронного документообігу впроваджено в університетську платформу KSU24, що забезпечує автоматизацію процесів створення, погодження, підписання та архівування документів. Модуль використовується для генерації

контрактів зі студентами на основі даних ЄДЕБО, створення довідок, організації вхідної та вихідної кореспонденції з повним аудитом усіх дій.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1. Аналіз предметної області електронного документообігу

Електронний документообіг (ЕДО) — це система організації роботи з документами в електронній формі, що забезпечує створення, зберігання, передавання, погодження та підписання документів без використання паперових носіїв [30]. Теоретичні засади організації ЕДО розглянуто в роботах [35]. На відміну від простого зберігання файлів, ЕДО передбачає наявність чітко визначених маршрутів руху документів, механізмів контролю виконання та аудиту всіх операцій.

У контексті закладу вищої освіти документообіг має ряд специфічних особливостей, що відрізняють його від документообігу в комерційних організаціях:

- **різноманітність типів документів** — університет працює з контрактами студентів, наказами, довідками, вхідною та вихідною кореспонденцією, кожен з яких має власний життєвий цикл та набір учасників;
- **складна організаційна структура** — документи проходять через деканати, кафедри, ректорат, бухгалтерію та інші підрозділи з різними рівнями доступу;
- **масовість операцій** — під час вступної кампанії необхідно одночасно генерувати та підписувати сотні контрактів;
- **інтеграція з державними системами** — необхідність взаємодії з ЄДЕБО [36] для отримання даних про студентів та з сервісом «Дія» [37] для електронного підпису;

- **юридична значущість** — документи повинні мати кваліфікований електронний підпис (КЕП), що забезпечує їх юридичну силу [31].

Процеси документообігу в університеті можна класифікувати за типом маршрутизації:

Лінійна маршрутизація передбачає послідовне проходження документа через визначені етапи. Наприклад, заява студента проходить шлях: реєстрація → розгляд деканом → затвердження ректором → виконання. Кожен наступний крок починається лише після завершення попереднього.

Паралельна маршрутизація дозволяє одночасне виконання кількох процесів. Наприклад, при підготовці наказу про відрахування одночасно можуть працювати деканат (перевірка академічної заборгованості) та бухгалтерія (перевірка фінансової заборгованості).

Ієрархічна маршрутизація передбачає створення підпроцесів у межах основного процесу. Наприклад, масова генерація контрактів створює батьківський процес із дочірніми процесами для кожного студента.

Ключовою вимогою до сучасної системи ЕДО є *динамічна маршрутизація* — можливість створювати, редагувати та повторно використовувати маршрути документів без зміни програмного коду [1, 10]. Це дозволяє адміністраторам системи самостійно налаштовувати бізнес-процеси під потреби університету.

1.2. Огляд існуючих систем електронного документообігу

На ринку існує низка систем електронного документообігу, як відкритих, так і комерційних. Розглянемо найбільш релевантні для університетського середовища.

Alfresco Community Edition [13] — відкрита платформа для управління корпоративним контентом (ECM), розроблена на Java. Alfresco надає потужні засоби роботи з документами: версіонування, повнотекстовий пошук, інтеграцію з офісними пакетами, механізми робочих процесів на базі Activiti BPMN [2]. Система має модульну архітектуру та підтримує розгортання в хмарному середовищі. Однак Alfresco є складною у налаштуванні, потребує значних серверних ресурсів, не має інтеграції з українськими державними сервісами та орієнтована переважно на англомовних корпоративних користувачів.

Paperless-ngx [19] — відкрита система управління документами, написана на Python (Django) та JavaScript (React). Paperless-ngx спеціалізується на оцифруванні паперових документів із використанням OCR-розпізнавання, автоматичній класифікації документів за допомогою алгоритмів машинного навчання та організації документів за тегами, кореспондентами та типами. Система легко розгортається через Docker-контейнери [8]. Проте Paperless-ngx не підтримує робочі процеси з маршрутизацією, не має механізмів електронного підпису та не призначена для багатокористувацького документообігу з розмежуванням прав.

Вчасно [32] — хмарний сервіс для електронного документообігу, орієнтований на обмін документами між контрагентами. Підтримує КЕП, інтеграцію з бухгалтерськими системами та автоматичну відправку звітності до державних органів. Сервіс не призначений для внутрішнього документообігу організації, не підтримує складну маршрутизацію та не має інтеграції з ЄДЕБО [36].

Документ.Онлайн [33] — комерційний хмарний сервіс електронного документообігу, орієнтований на обмін документами між контрагентами. Підтримує кваліфікований електронний підпис через акредитовані центри сертифікації ключів, створення документів із шаблонів системи та налаштування маршрутів погодження з визначенням учасників

та їхніх ролей. Сервіс інтегрується з 1С та використовує Microsoft Azure для зберігання. Проте система не підтримує складну маршрутизацію з паралельними чи ієрархічними процесами, не має інтеграції з ЄДЕБО та орієнтована на міжорганізаційний обмін, а не на внутрішній документообіг.

АСКОД Онлайн [27] — комерційна українська система електронного документообігу від компанії «Softline», що широко використовується в державних установах та закладах освіти. Система забезпечує автоматизацію службового документообігу, опрацювання кореспонденції, звернень громадян та адміністративних послуг. Підтримує електронний цифровий підпис, налаштовувані маршрути руху документів, контроль виконання, протоколювання дій у захищеному журналі та інтеграцію з Microsoft Office і бухгалтерськими системами. Однак АСКОД не підтримує паралельну маршрутизацію та ієрархію процесів, не має інтеграції з ЄДЕБО та «Дія» [37], а генерація документів із шаблонів обмежена базовими можливостями.

NextCloud з модулем Workflow [18] — відкрита платформа для зберігання та спільної роботи з файлами. Модуль Workflow дозволяє створювати прості правила автоматизації (переміщення файлів, сповіщення), проте не забезпечує повноцінної маршрутизації документів з багатокроковим погодженням та підписанням.

1.3. Порівняльний аналіз існуючих рішень

Для обґрунтованого вибору підходу до розробки проведемо порівняльний аналіз розглянутих систем за ключовими критеріями, важливими для університетського документообігу (табл. 1.1).

Таблиця 1.1

Порівняльний аналіз систем електронного документообігу

Критерій	Alfresco	Paperless-ngx	Вчасно	Док. Онлайн	АСКОД Онлайн	KSU24
Маршрутизація	BPMN	Відсутня	Базова	Базова	Налашт.	Динам., віз. редактор
Підпис КЕП	Ні	Ні	Так	Так	Так	Так
Інтегр. з Дія	Ні	Ні	Ні	Ні	Ні	Так
Інтегр. з ЄДЕБО	Ні	Ні	Ні	Ні	Ні	Так
Генерація з шабл.	Обмеж.	Ні	Ні	Так	Обмеж.	WYSIWYG
Відкритий код	Так	Так	Ні	Ні	Ні	Так
Паралел. маршрут.	Так	Ні	Ні	Ні	Ні	Так
Ієрархія проц.	Ні	Ні	Ні	Ні	Ні	Так
Аудит дій	Так	Обмеж.	Обмеж.	Базов.	Так	Повний
Укр. локаліз.	Частк.	Ні	Так	Так	Так	Так

Як видно з таблиці 1.1, жодна з існуючих систем не забезпечує повного набору функцій, необхідних для університетського документообігу. Alfresco має потужну BPMN-маршрутизацію, але не інтегрована з українськими сервісами. Paperless-ngx спеціалізується на оцифруванні, а не на маршрутизації. АСКОД Онлайн — зріле українське рішення з підтримкою КЕП, проте не має ієрархії процесів та інтеграції з ЄДЕБО. Вчасно та Документ.Онлайн орієнтовані на міжорганізаційний обмін, а не на внутрішній документообіг із складними маршрутами. Жодна система не поєднує динамічну маршрутизацію з візуальним редактором, інтеграцію з «Дія» та ЄДЕБО, генерацію документів із WYSIWYG-шаблонів та ієрархічні процеси одночасно. Це підтверджує доцільність розробки власного модуля.

1.4. Обґрунтування вибору інструментів розробки

Модуль електронного документообігу розроблено як частину існуючої університетської платформи KSU24, що обумовило вибір основних технологій.

Серверна частина. Платформа KSU24 побудована на фреймворку Django [7] (Python), тому серверна частина модуля документообігу

також реалізована на Django. Цей фреймворк забезпечує потужну ORM для роботи з PostgreSQL [20], вбудовану систему автентифікації та авторизації, адміністративну панель для управління даними та широку екосистему бібліотек. Для виконання асинхронних задач (масова генерація контрактів) використовується Celery [5] з Redis [25] як брокером повідомлень.

Клієнтська частина. Фронтенд платформи KSU24 побудований на React [24] з бібліотекою компонентів Ant Design [3] та TanStack Query [28] для управління серверним станом. Бібліотека AntV X6 [4] для побудови графових редакторів уже була частиною проєкту і була використана для реалізації візуального редактора маршрутів документів.

Редактор шаблонів. Для реалізації WYSIWYG-редактора шаблонів документів було обрано бібліотеку TipTap [29], побудовану на основі ProseMirror [21]. Вибір обумовлений такими перевагами:

- розширювана архітектура, що дозволяє створювати власні типи вузлів (змінні, блокові змінні);
- підтримка структурованого JSON-формату для зберігання контенту;
- можливість обмеження редагування окремих ділянок документа (захист структури шаблону);
- активна спільнота та регулярні оновлення.

Альтернативні редактори (CKEditor [6], Quill [23], Slate [26]) були відхилені через обмежені можливості створення кастомних вузлів, менш гнучку архітектуру розширень або відсутність підтримки ізолюваного редагування окремих зон документа.

Зберігання файлів. Для зберігання файлів документів використовується Google Drive API [11], що забезпечує надійне хмарне зберігання, дедуплікацію файлів за SHA-256 хешем та контроль доступу.

Зберігання шаблонів і маршрутів. Шаблони документів та конфігурації маршрутів зберігаються в системі JARS — внутрішньому сервісі платформи KSU24 для управління структурованими даними з підтримкою версіонування та контролю доступу.

1.5. Постановка задачі

На підставі проведеного аналізу сформульовано такі вимоги до модуля електронного документообігу.

Функціональні вимоги:

- а) створення, редагування та видалення документів з підтримкою різних типів (контракти, довідки, накази, кореспонденція);
- б) динамічна маршрутизація з візуальним редактором маршрутів, що підтримує лінійні, паралельні та ієрархічні процеси;
- в) багатокрокове погодження документів із типами кроків: перегляд, затвердження, валідація, підписання, виконання, реєстрація, архівування;
- г) генерація документів із WYSIWYG-шаблонів з автоматичною підстановкою змінних;
- д) електронний підпис через інтеграцію з сервісом «Дія»;
- е) імпорт даних студентів з ЄДЕБО для генерації контрактів;
- ж) система сповіщень про дії з документами;
- з) повний аудит усіх операцій із документами;
- и) система коментарів з підтримкою вкладених відповідей та вкладень.

Нефункціональні вимоги:

- а) масштабованість — підтримка одночасної обробки сотень документів;

- б) безпека — розмежування прав доступу, шифрування конфігурацій інтеграцій;
- в) розширюваність — можливість додавання нових типів документів та змінних без зміни коду;
- г) інтеграційність — API-сумісність з іншими модулями платформи KSU24.

РОЗДІЛ 2

ПРОЄКТУВАННЯ МОДУЛЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

2.1. Архітектура системи

Модуль електронного документообігу спроектовано з використанням багат шарової архітектури [15], що забезпечує розділення відповідальності між компонентами та полегшує підтримку та розширення системи. Концептуальну архітектуру модуля наведено на рис. 2.1.

Серверна частина побудована на фреймворку Django та складається з чотирьох основних шарів: шар API, шар бізнес-логіки, шар сервісів та шар даних. Клієнтська частина реалізована як React SPA із використанням TanStack Query для кешування серверного стану та AntV X6 для візуального редактора маршрутів. Взаємодія між клієнтом і сервером здійснюється через REST API у форматі JSON. Система інтегрована із зовнішніми сервісами: ЄДЕБО, Дія, Google Drive. Асинхронні задачі (генерація контрактів, підписання, масові сповіщення) виконуються через Celery.

Архітектура складається з чотирьох шарів.

Шар API (API Layer) містить 72 файли ендпоінтів, організованих у 12 функціональних груп: документи, центр документів, контракти, довідки, універсальна генерація, процеси, стани документів, коментарі, сповіщення, типи документів, файли Google Drive та загальні дані. Кожен ендпоінт реалізовано в окремому файлі, що спрощує навігацію та підтримку коду. Шар відповідає за валідацію вхідних даних через Pydantic-схеми, перевірку прав доступу та маршрутизацію запитів до відповідних компонентів бізнес-логіки.

Концептуальна архітектура модуля електронного документообігу

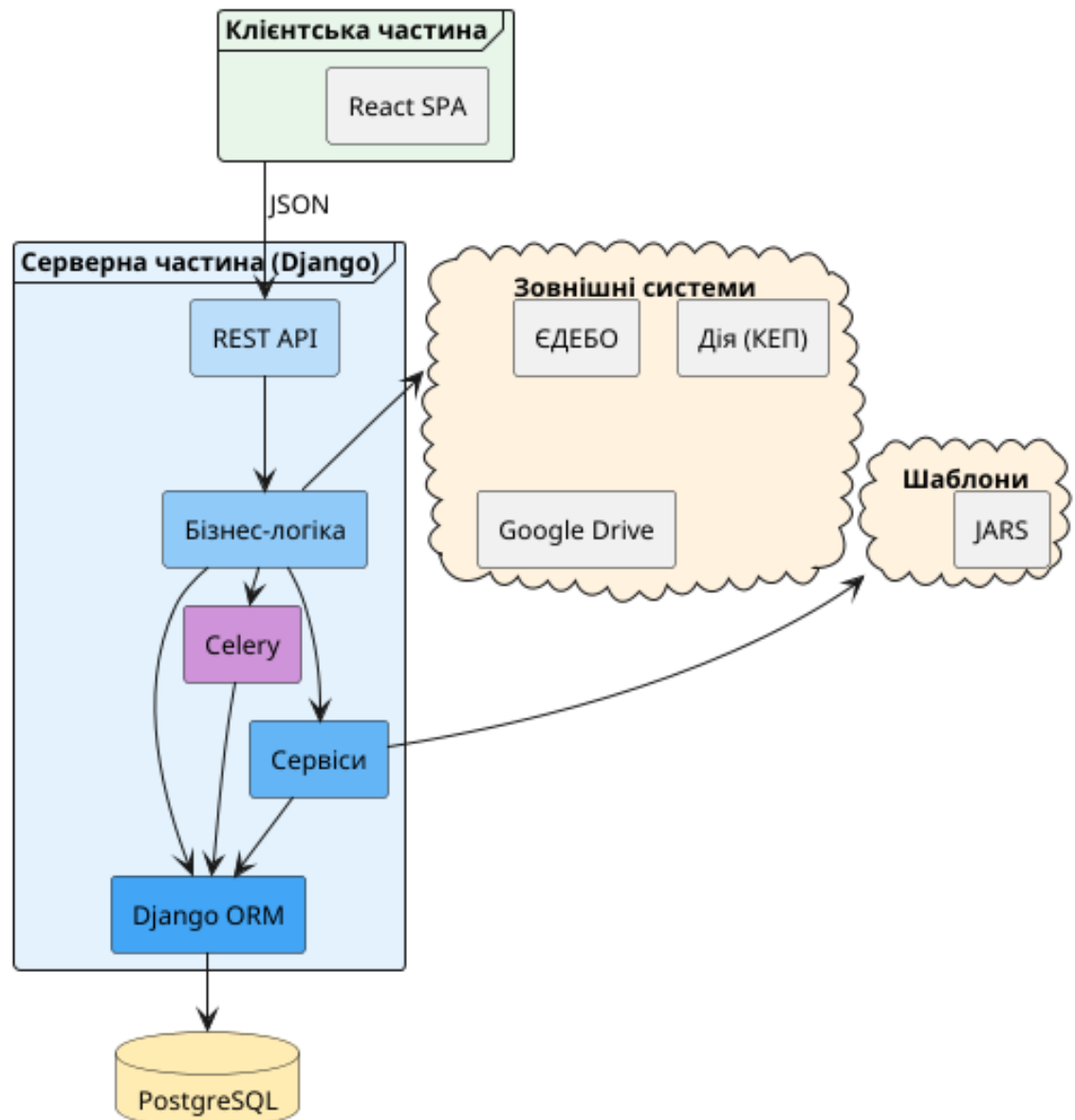


Рисунок 2.1. Концептуальна архітектура модуля електронного документообігу

Шар бізнес-логіки (Features Layer) містить основну функціональність системи:

- *DocumentFlowFeature* — управління життєвим циклом документа, запуск маршрутів, обробка кроків, інтеграція з електронним підписом;
- *DocumentHubFeature* — управління необробленими документами до їх прив'язки до маршруту;

- *UniversalDocumentFeature* — генерація документів із шаблонів з підстановкою змінних;
- *NotificationActionFeature* — створення та управління сповіщеннями;
- *ContractsHubFeature* — генерація контрактів на основі даних ЄДЕБО;
- *CertificatesHubFeature* — генерація довідок про навчання.

Шар сервісів (Services Layer) забезпечує допоміжну функціональність:

- *DocumentFlowService* — форматування TipTap-контенту, генерація XML для підпису;
- *UniversalContextService* — побудова контексту змінних для генерації документів;
- *ProcessService* та *ProcessStepService* — серіалізація даних процесів та кроків;
- підсистема резолверів змінних (*Variable Resolvers*) — реєстр функцій для обчислення значень змінних шаблону.

Шар даних (Data Models Layer) містить 15 моделей Django ORM, об'єднаних у чотири групи: основні моделі, моделі центру документів, допоміжні моделі та моделі аудиту. Усі моделі успадковуються від базової моделі *DataObject*, що забезпечує UUID-первинні ключі, м'яке видалення, автоматичні мітки часу та повну історію змін.

Між серверною та клієнтською частинами взаємодія здійснюється через REST API [9, 16]. Клієнтська частина побудована на React та використовує TanStack Query для кешування та синхронізації серверного стану.

Загальну схему виконання маршруту документа наведено на рис. 2.2.

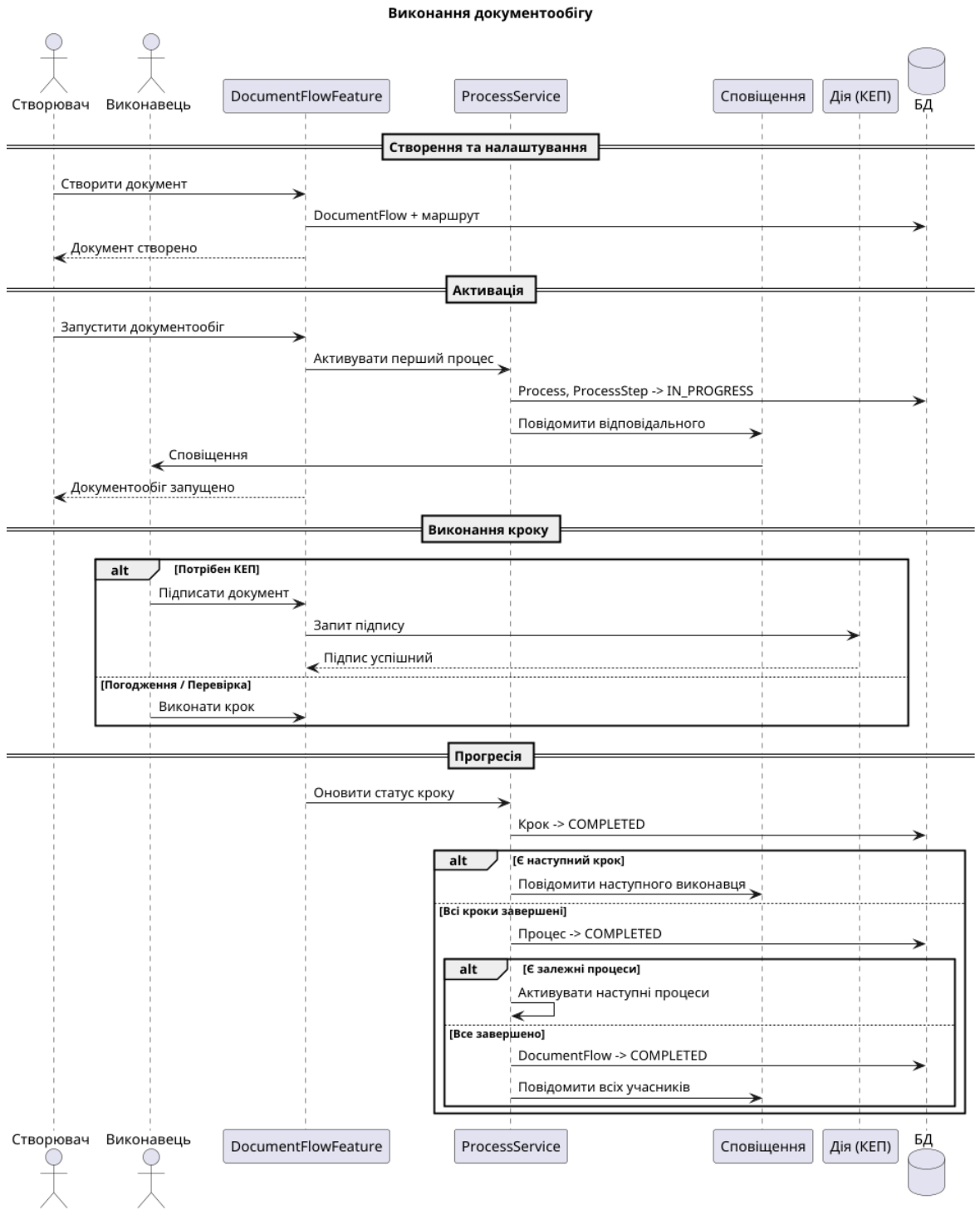


Рисунок 2.2. Діаграма послідовності виконання маршруту документа

2.2. Проектування моделі даних

Модель даних модуля документообігу містить 15 моделей, організованих у чотири групи (рис. 2.3).

Основні моделі визначають ядро системи документообігу:

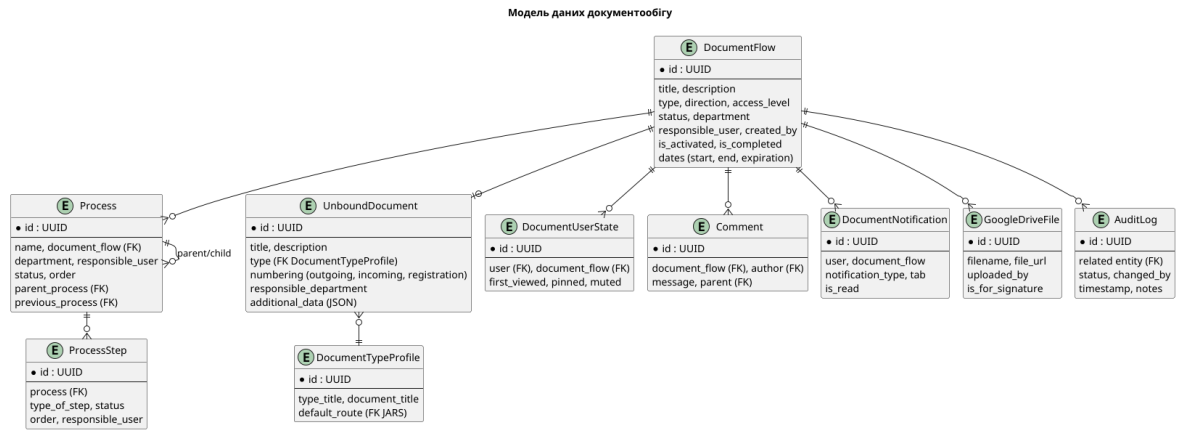


Рисунок 2.3. ER-діаграма моделі даних модуля документообігу

DocumentFlow — центральний контейнер документообігу, що пов'язує документ із маршрутом, файлами та учасниками. Модель містить: назву та опис документа, класифікацію (напрямок, рівень доступу), зв'язки з файлами через M2M-відношення (основні файли, файли для підпису, пов'язані контейнери), генеричний зовнішній ключ для прив'язки до довільної сутності (студент, працівник, підрозділ) та статус документа.

Process — процес у маршруті документа. Кожен процес належить до конкретного документообігу та може мати залежності від інших процесів через зв'язок із попередніми процесами. Це дозволяє моделювати як послідовні, так і паралельні процеси. Для ієрархічних процесів передбачено посилання на батьківський процес. Спеціальний прапорець визначає, чи зупиняється весь маршрут при відхиленні цього процесу.

ProcessStep — окремий крок у процесі. Кроки виконуються послідовно за порядковим номером. Кожен крок має тип (перегляд, затвердження, валідація, підписання тощо), статус та прапорці активації та завершення.

Моделі центру документів забезпечують роботу з документами до їх прив'язки до маршруту:

UnboundDocument — необроблений документ, що існує до прив’язки до маршруту документообігу. Модель містить гнучке JSON-поле для зберігання специфічних даних залежно від типу документа, систему нумерації (вхідний, вихідний, реєстраційний номери), класифікацію та поля управління (відповідальний підрозділ, виконавець, термін виконання).

DocumentTypeProfile — профіль типу документа, що визначає стандартний маршрут, відповідальний підрозділ, термін зберігання та ієрархію типів.

Допоміжні моделі реалізують комунікацію та взаємодію:

Comment — система коментарів з підтримкою форматованого тексту у форматі TipTap JSON, вкладених відповідей через посилання на батьківський коментар, M2M-зв’язків для одержувачів та вкладень.

DocumentUserState — стан документа для конкретного користувача: дата першого та останнього перегляду, закріплення та вимкнення сповіщень. Унікальне обмеження на пару (користувач, документ) гарантує один запис на користувача.

DocumentNotification та *DocumentNotificationSummary* — система сповіщень. Індивідуальні сповіщення зберігають тип, вкладку відображення та статус прочитання. Агрегована модель зберігає прапорець наявності непрочитаних сповіщень для ефективних запитів до бази даних.

GoogleDriveFile — метадані файлів з полями: оригінальне ім’я, URL, SHA-256 хеш для дедуплікації, прапорці підпису та відхилення.

Моделі аудиту забезпечують повне відстеження змін: *AuditLogDocumentFlow*, *AuditLogProcess*, *AuditLogProcessStep* — кожна модель зберігає попередній та поточний статуси, автора зміни, мітку часу та коментар.

2.3. Проектування системи динамічної маршрутизації

Динамічна маршрутизація є ключовою підсистемою модуля, що дозволяє створювати та редагувати маршрути документів без зміни програмного коду.

Концепція маршруту. Маршрут — це JSON-шаблон, що зберігається в системі JARS і визначає, які процеси необхідно створити, в якому порядку їх виконувати, які кроки входять до кожного процесу та хто є відповідальним. Маршрути підтримують версіонування та прив'язку до профілів типів документів.

Структура маршруту. Маршрут складається з масиву процесів, кожен з яких містить: ідентифікатор, назву, підрозділ, відповідальну особу, список кроків із типами та порядком виконання, масив залежностей від інших процесів та прапорець зупинки при відхиленні. Маршрут підтримує підстановку змінних, що розв'язуються під час створення документообігу.

Три рівні зв'язків між процесами забезпечують гнучкість маршрутизації:

Рівень 1 — залежності між процесами. М2М-зв'язок між процесами визначає порядок виконання. Процес може розпочатися лише після завершення всіх процесів, від яких він залежить. Це дозволяє моделювати як послідовні ланцюжки, так і паралельне розгалуження з подальшим злиттям. Для візуалізації використовується рекурсивний алгоритм обчислення рівнів процесів, що групує процеси за глибиною залежностей.

Рівень 2 — дочірні процеси. Зовнішній ключ дозволяє делегувати виконання підзадач окремим підпроцесам, кожен з яких має власну відповідальну особу та набір кроків. Це використовується, наприклад,

при масовій генерації контрактів, де батьківський процес контролює загальний стан, а дочірні процеси обробляють окремих студентів.

Рівень 3 — пов'язані контейнери. M2M-зв'язок між документо-обігами дозволяє відстежувати зв'язки між пов'язаними документами, наприклад, між контрактом та додатковою угодою.

Візуальний редактор маршрутів на клієнтській частині побудований на бібліотеці AntV X6 і забезпечує графічний інтерфейс для створення та редагування маршрутів. Процеси відображаються як прямокутні вузли з інформацією про назву, підрозділ, відповідальну особу та кроки. Зв'язки між вузлами відповідають залежностям процесів. Редактор підтримує повноекранний режим, групування вузлів, вставку існуючих маршрутів та збереження через API.

Організація маршрутів реалізована через деревоподібну структуру папок з контекстним меню для створення, перейменування, видалення папок та переміщення маршрутів. Підтримується перетягування та сортування за українським алфавітом.

Загальну схему універсальної генерації документів наведено на рис. 2.4.

2.4. Проектування життєвого циклу документа

Життєвий цикл документа в системі визначається скінченним автоматом (State Machine) з чітко визначеними станами та дозволеними переходами (рис. 2.5).

Стани документообігу (DocumentFlow):

- *Чернетка (Draft)* — початковий стан після створення. Документ можна редагувати, прикріплювати файли, налаштовувати маршрут;

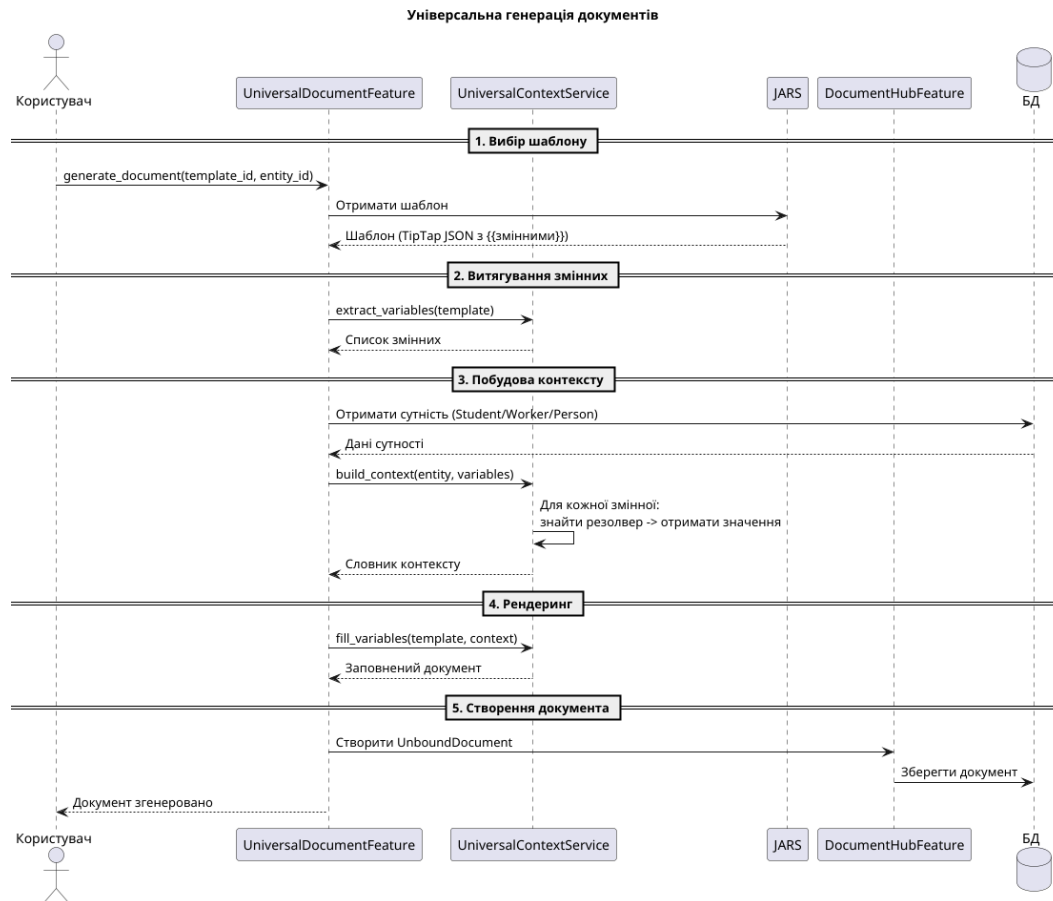


Рисунок 2.4. Діаграма процесу універсальної генерації документів

- *В роботі (In Progress)* — маршрут запущено, процеси виконуються відповідно до залежностей;
- *Завершено (Completed)* — усі процеси успішно завершені;
- *Відхилено (Rejected)* — один із процесів відхилив документ;
- *Архівовано (Archived)* — документ переведено в архів після завершення;
- *Видалено (Deleted)* — м'яке видалення документа.

Стани процесу (Process):

- *Очікує (Pending)* — процес створено, але попередні залежності ще не завершені;
- *В роботі (In Progress)* — всі залежності виконано, кроки процесу активні;
- *Завершено (Completed)* — усі кроки процесу виконано;

Стани життєвого циклу документа



Рисунок 2.5. Діаграма станів життєвого циклу документа

- *Відхилено (Rejected)* — один із кроків відхилив документ.

Стани кроку (ProcessStep):

- *Очікує (Pending)* — крок ще не активний;
- *В роботі (In Progress)* — крок активний, очікує дії відповідальної особи;

- *Завершено (Completed)* — крок успішно виконано;
- *Відхилено (Rejected)* — крок відхилено з обов'язковим коментарем;
- *Пропущено (Skipped)* — крок пропущено за логікою маршруту.

Алгоритм запуску маршруту:

- а) Перевірити, що всі процеси мають призначених відповідальних осіб.
- б) Змінити статус DocumentFlow на «В роботі».
- в) Визначити процеси без залежностей (процеси першого рівня).
- г) Активувати ці процеси та їхні перші кроки.
- д) Створити сповіщення для відповідальних осіб.

Алгоритм завершення кроку:

- а) Змінити статус кроку на «Завершено» та записати в аудит.
- б) Якщо є наступний крок у процесі — активувати його.
- в) Якщо це був останній крок — змінити статус процесу на «Завершено».
- г) Перевірити, чи є процеси, що залежали від завершеного, та чи всі їхні залежності тепер виконані — якщо так, активувати їх.
- д) Якщо всі процеси маршруту завершені — змінити статус DocumentFlow на «Завершено».

Обробка відхилення: При відхиленні кроку створюється обов'язковий коментар із причиною. Якщо процес налаштовано на зупинку при відхиленні, весь документообіг переводиться у стан «Відхилено». Відхилений документ може бути повернутий на доопрацювання з повторним проходженням маршруту.

2.5. Проектування інтеграцій із зовнішніми системами

Модуль документообігу інтегрований з чотирма зовнішніми системами [12] (рис. 2.6).

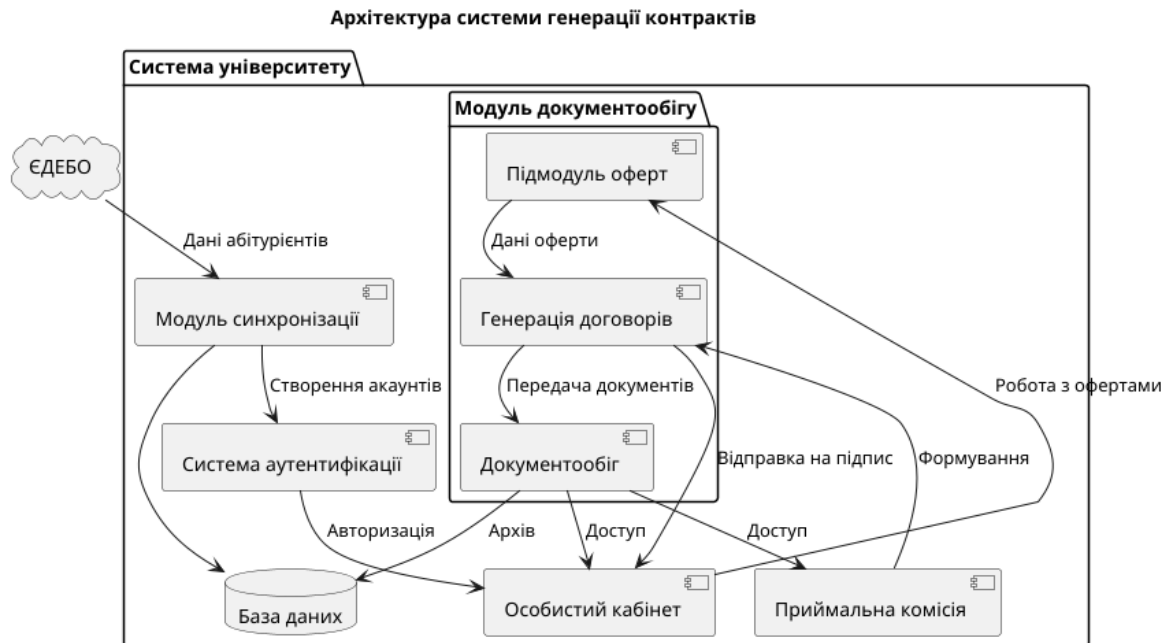


Рисунок 2.6. Компонентна діаграма системи з інтеграціями

Інтеграція з ЄДЕБО. Єдина державна електронна база з питань освіти є джерелом даних про абітурієнтів та студентів [34]. Дані надходять через проміжний кеш у системі JARS, де зберігаються у вигляді моделі *EdeboEntrant* з персональними даними, інформацією про освітню програму, форму навчання, джерело фінансування та успішність. Ці дані використовуються для автоматичної генерації контрактів зі студентами.

Інтеграція з JARS. Внутрішній сервіс платформи KSU24 [38] виконує роль сховища шаблонів документів (TipTap JSON), конфігурацій маршрутів (JSON), історичних даних студентів та згенерованих документів. JARS забезпечує версіонування документів, контроль доступу на основі capabilities та підтримку різних типів документів.

Інтеграція з Дія. Сервіс «Дія» забезпечує накладання кваліфікованого електронного підпису (КЕП) на документи. Модуль документо-

обігу використовує існуючий на платформі додаток `diia` для взаємодії з API сервісу «Дія» та додаток `document_management` для управління життєвим циклом документа як об'єкта підписання (оригінал, запит на підпис, підписаний документ). Процес підписання складається з таких етапів:

- а) модуль документообігу зберігає оригінал документа та генерує його XML-версію;
- б) обчислюється хеш документа та через додаток `diia` формується запит на підписання;
- в) API Дія повертає `deeplink` — користувач переходить в застосунок «Дія» або на сторінку ID.GOV.UA для підпису файловим ключем КЕП;
- г) після підписання «Дія» надсилає `callback` з підписаними даними;
- д) модуль документообігу отримує розшифровані дані та зберігає підписаний документ із метаданими сертифіката (ПІБ, РНОКПП, видавець, серійний номер, термін дії).

Інтеграція з Google Drive. Хмарне сховище використовується для зберігання файлів документів та підписаних документів. Реалізовано дедуплікацію файлів: при завантаженні обчислюється SHA-256 хеш, і якщо файл з таким хешем вже існує, створюється посилання замість повторного завантаження.

Діаграму варіантів використання системи генерації контрактів наведено на рис. 2.7, діаграму послідовності процесу генерації контракту — на рис. 2.8, а діаграму станів абітурієнта — на рис. 2.9.

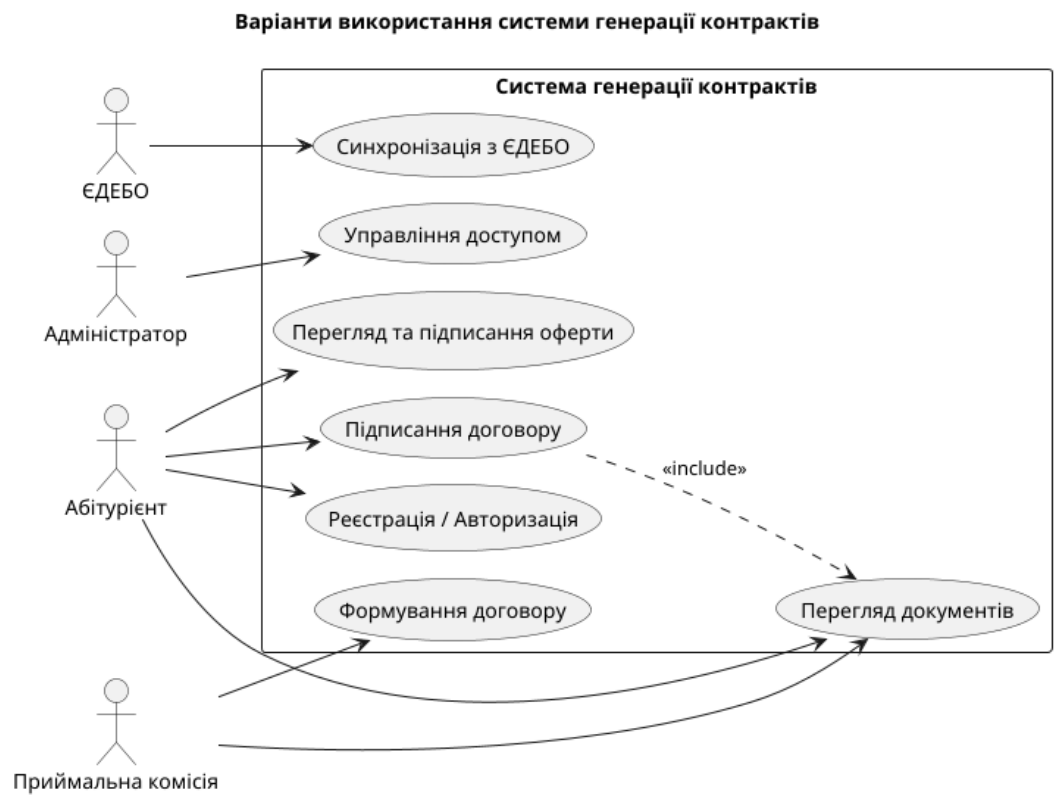


Рисунок 2.7. Діаграма варіантів використання системи генерації контрактів

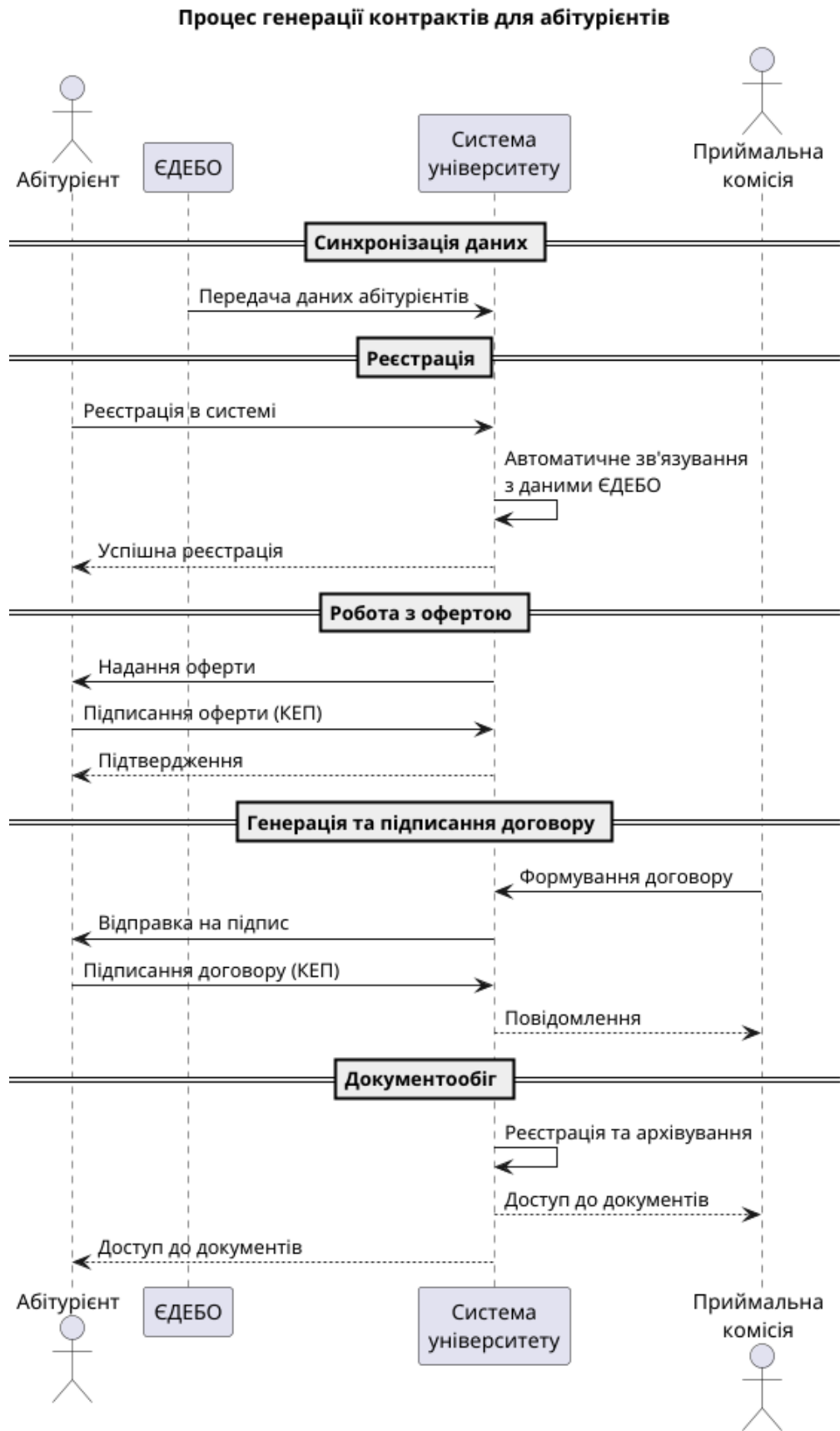


Рисунок 2.8. Діаграма послідовності процесу генерації контракту абітурієнта

Стани абітурієнта в процесі оформлення договору



Рисунок 2.9. Діаграма станів абітурієнта в процесі оформлення контракту

РОЗДІЛ 3

РЕАЛІЗАЦІЯ МОДУЛЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

3.1. Реалізація серверної частини

Серверна частина модуля реалізована як окремий Django-додаток [17] з чотиришаровою архітектурою, описаною в розділі 2.

Реалізація моделей даних. Усі 15 моделей успадковуються від спільного базового класу, що забезпечує UUID-первинні ключі, м'яке видалення, автоматичні мітки часу створення та модифікації, а також повну історію змін. Статуси документів, процесів та кроків зберігаються через механізм налаштовуваних довідників — групи значень, керованих через адміністративну панель, що дозволяє змінювати набір статусів без модифікації коду.

Для оптимізації продуктивності запитів до бази даних [14] застосовуються механізми попереднього завантаження пов'язаних об'єктів Django ORM. Агрегована модель сповіщень із прапорцем наявності непрочитаних повідомлень забезпечує ефективне відображення лічильників без складних агрегаційних запитів.

Реалізація API. REST API містить 72 ендпоінти, організованих у 12 груп. Кожен ендпоінт реалізовано в окремому файлі для зручності навігації. Вхідні та вихідні дані валідуються через Pydantic-схеми [22]. Перевірка прав доступу реалізована через окремий модуль, що забезпечує декларативний контроль дозволів.

Основні групи ендпоінтів охоплюють: CRUD-операції з документообігом, управління необробленими документами, універсальну генерацію документів, генерацію контрактів на основі даних ЄДЕБО, управління сповіщеннями та запуск маршрутів.

Реалізація бізнес-логіки. Шар Features містить основну логіку системи. Основний компонент бізнес-логіки реалізує: створення документообігу з прив'язкою до маршруту, запуск маршруту з активацією перших процесів, обробку завершення та відхилення кроків з каскадною активацією залежних процесів, інтеграцію з електронним підписом Дія. Компонент управління документами забезпечує роботу з необробленими документами, автоматичне створення документообігу та генерацію реєстраційних номерів.

Асинхронні задачі. Для масових операцій використовується Celery. При генерації контрактів для великої кількості студентів (наприклад, під час вступної кампанії) задача виконується асинхронно у фоновому режимі, щоб не блокувати основний потік обробки запитів. Celery-задачі створюють дочірні процеси для кожного студента в межах батьківського процесу генерації.

Система аудиту. Кожна зміна статусу документа, процесу або кроку фіксується у відповідній моделі аудиту з інформацією про: попередній статус, новий статус, автора зміни, мітку часу та коментар. Це забезпечує повну прозорість та можливість відстеження всієї історії документа.

Схему потоку сповіщень у системі наведено на рис. 3.1.

3.2. Реалізація клієнтської частини

Клієнтська частина реалізована на React з використанням Ant Design для UI-компонентів та TanStack Query для управління серверним станом.

Модуль документообігу. Основний інтерфейс складається з таких компонентів:

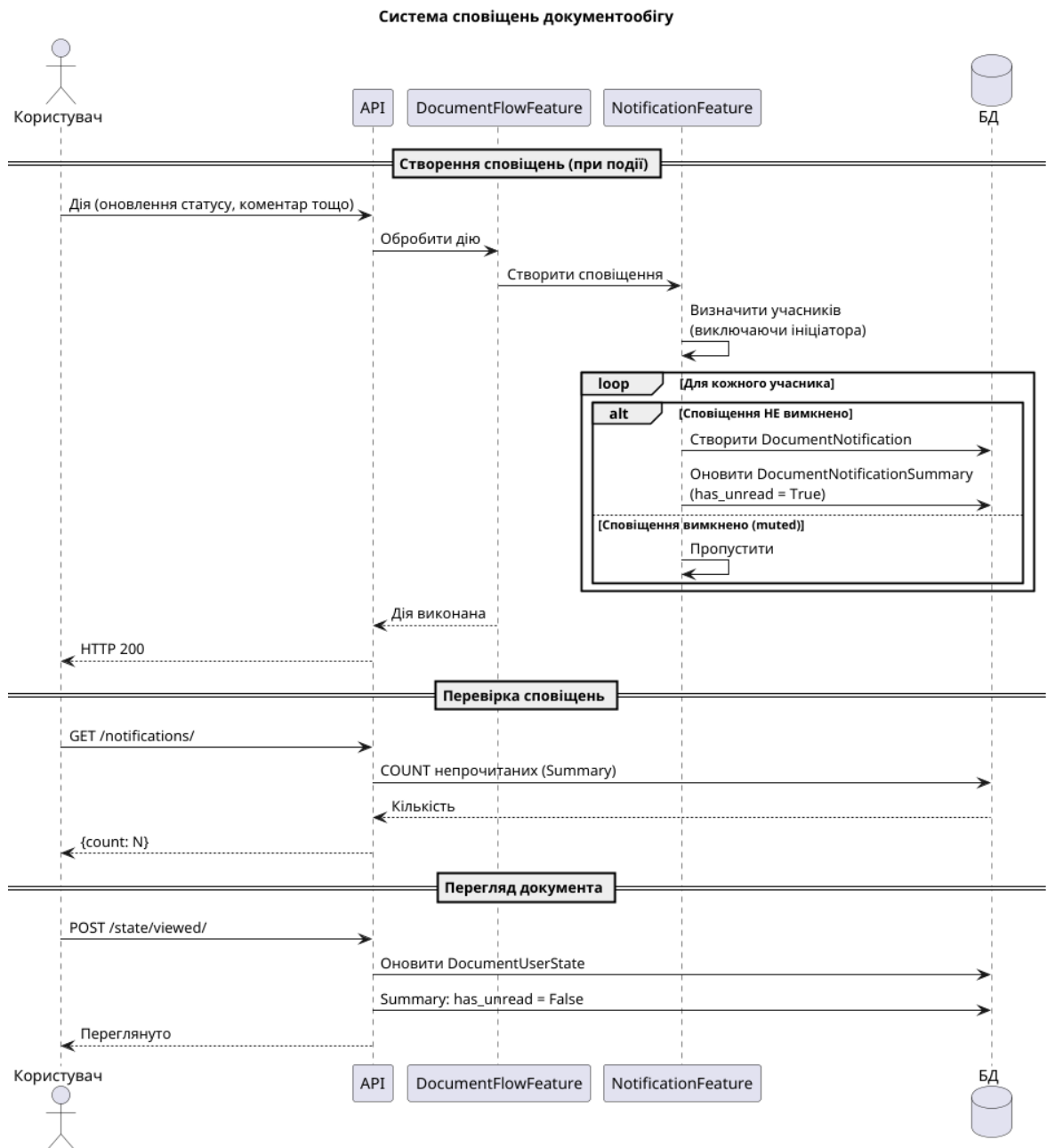


Рисунок 3.1. Діаграма потоку системи сповіщень

- форма створення нового документообігу з полями для типу документа, напрямку, терміну дії, підрозділу, рівня доступу, завантаження файлів та вибору маршруту з візуальним відображенням;
- сторінка перегляду документа з метаданими, вкладками для історії процесу, коментарів та глобальної історії;
- панель дій документа, що динамічно відображає кнопки залежно від поточного кроку (перегляд, затвердження, підписання тощо).

Для кроків типу «перегляд» реалізовано автоматичне підтвердження з 5-секундним таймером.

Візуальний редактор маршрутів. Реалізований на основі бібліотеки AntV X6, редактор забезпечує графічний інтерфейс для створення та редагування маршрутів документів. Редактор відображає процеси як прямокутні вузли графу з інформацією про назву, підрозділ, відповідальну особу та список кроків. Зв'язки між вузлами відповідають залежностям процесів. Для редагування властивостей окремого процесу реалізовано бічну панель з полями назви, підрозділу, відповідальної особи, залежностей та списку кроків.

Організацію маршрутів реалізовано через деревоподібну структуру папок з підтримкою контекстного меню, перетягування, вбудованого редагування назв та сортування за українським алфавітом.

Візуалізація статусу маршруту. Візуалізація статусу відображає поточний стан виконання маршруту у вигляді згрупованих рівнів процесів. Для обчислення рівнів використовується рекурсивний алгоритм, що аналізує залежності між процесами та визначає глибину кожного процесу. Процеси на одному рівні відображаються горизонтально, статуси позначаються кольоровими індикаторами.

Управління необробленими документами. Інтерфейс надає таблицю документів користувача зі статусами (чернетка, оброблений, видалений), лічильниками сповіщень та переходом до редагування. Форма створення необробленого документа містить поля для класифікації, нумерації, метаданих та вкладень.

Система сповіщень та коментарів. Числові лічильники сповіщень (з обмеженням 99+) відображаються на вкладках та рядках таблиць. Система коментарів підтримує вкладені відповіді необмеженої глибини через рекурсивний рендеринг із захистом від циклічних посилань.

Модуль налаштувань. Адміністративна частина включає: управління профілями типів документів, управління групами змінних для шаблонів, управління шаблонами документів з вкладковим інтерфейсом для трьох типів (документи, контракти, сніпети), управління шаблонами відповідей та налаштування правил контрактів з 11 полями для фільтрації за параметрами студента.

3.3. Реалізація системи генерації документів

Система генерації документів побудована на WYSIWYG-редакторі TipTap та забезпечує створення документів із шаблонів з автоматичною підстановкою змінних.

Редактор шаблонів. Реалізовано повнофункціональний редактор з 14 стандартними розширеннями TipTap (базовий набір, стилі тексту, шрифти, розміри, кольори, вирівнювання, зображення, списки, таблиці, висота рядка) та трьома кастомними розширеннями:

VariableNode — інлайновий вузол для вставки змінних виду `{{variable_name}}`. Підтримує типи: текст, число, дата, місяць, рік, вибір зі списку. В режимі заповнення змінна відображається як відповідний елемент введення (текстове поле, випадаючий список, вибір дати).

RichVariable — блоковий вузол для змінних з форматованим вмістом. Забезпечує ізольований контекст редагування з власною панеллю інструментів.

ReadOnlyNodes — розширення, що захищає структуру шаблону від змін у режимі заповнення. Перехоплює події клавіатури, вставки та перетягування за межами зон змінних.

Режими роботи редактора:

- *admin* — створення шаблонів із розміткою змінних;
- *fill* — заповнення шаблону, де змінні стають полями введення;
- *edit* — редагування раніше заповненого документа;
- *view* — перегляд готового документа з підтримкою друку та QR-кодів підпису.

Універсальна генерація. Серверна підсистема універсальної генерації працює за таким алгоритмом (рис. 3.2, 3.3):

- а) отримання HTTP-запиту з ідентифікаторами шаблону та необробленого документа;
- б) перевірка прав доступу;
- в) завантаження TipTap JSON-шаблону з JARS та необробленого документа з бази даних;
- г) витяг усіх змінних `{{variable}}` з JSON-структури шаблону;
- д) визначення типу сутності (студент, працівник, особа) та завантаження відповідних даних;
- е) побудова контексту: для кожної змінної знаходиться відповідний резолвер з реєстру, витягується та форматується значення;
- ж) заміна всіх змінних у JSON-шаблоні значеннями з контексту;
- з) збереження згенерованого документа в JARS.

Підсистема резолверів змінних. Для розв’язування змінних реалізовано реєстр резолверів на основі декораторів Python. Кожен резолвер — це функція, що отримує контекст (дані сутності) та повертає відформатоване значення. Наприклад, окремі резолвери витягують повне ім’я студента, назву факультету чи поточну дату у відповідному форматі. Додавання нової змінної потребує лише створення декорованої функції, без зміни існуючого коду.

Генерація контрактів. Спеціалізований модуль для генерації контрактів зі студентами використовує дані з ЄДЕБО: персональні дані, освітню програму, форму навчання, джерело фінансування. Сервіс кон-

Діяльність універсальної генерації документів

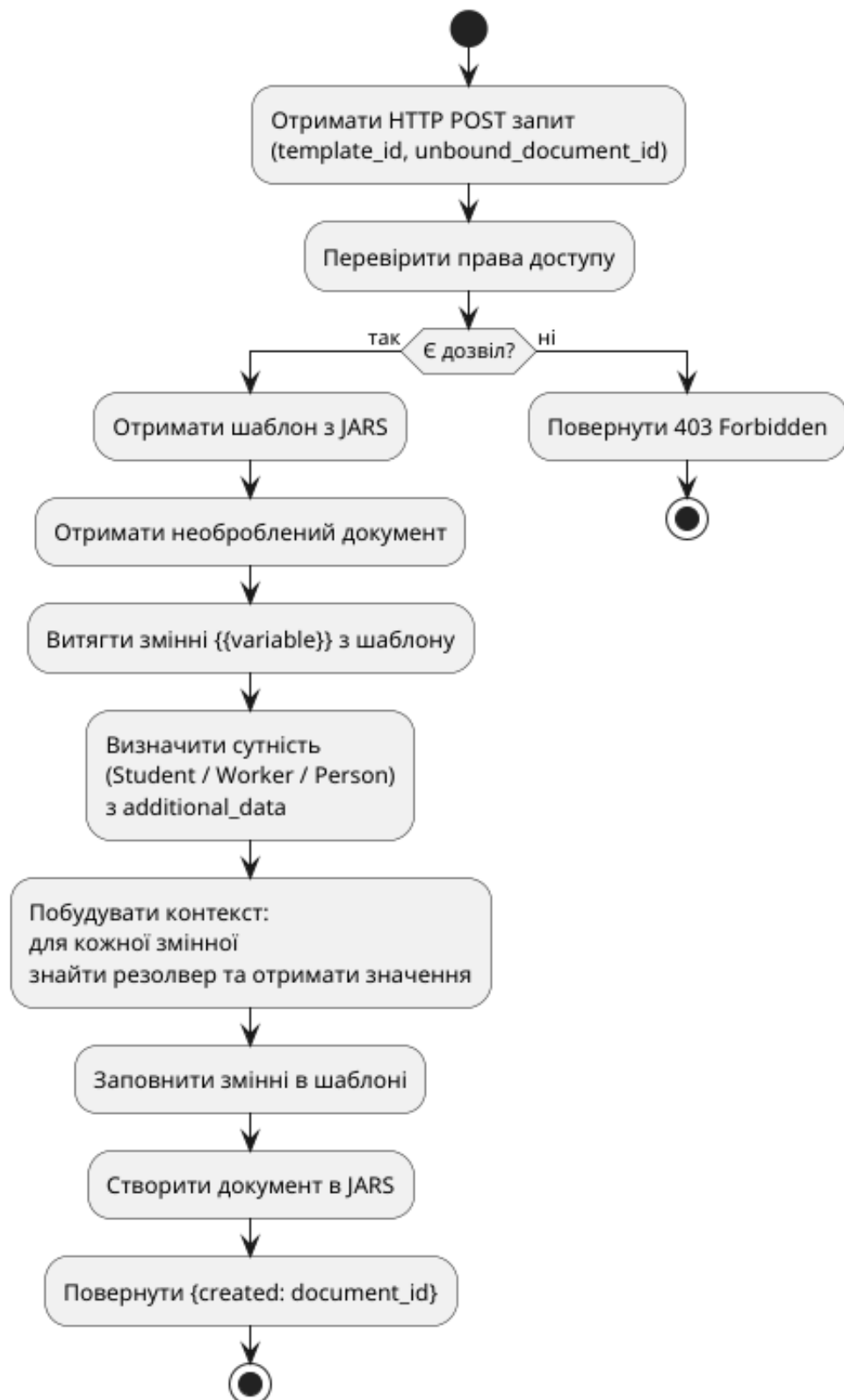


Рисунок 3.2. Діаграма діяльності універсальної генерації документів
тексту обчислює додаткові значення: вік студента, щомісячну оплату,

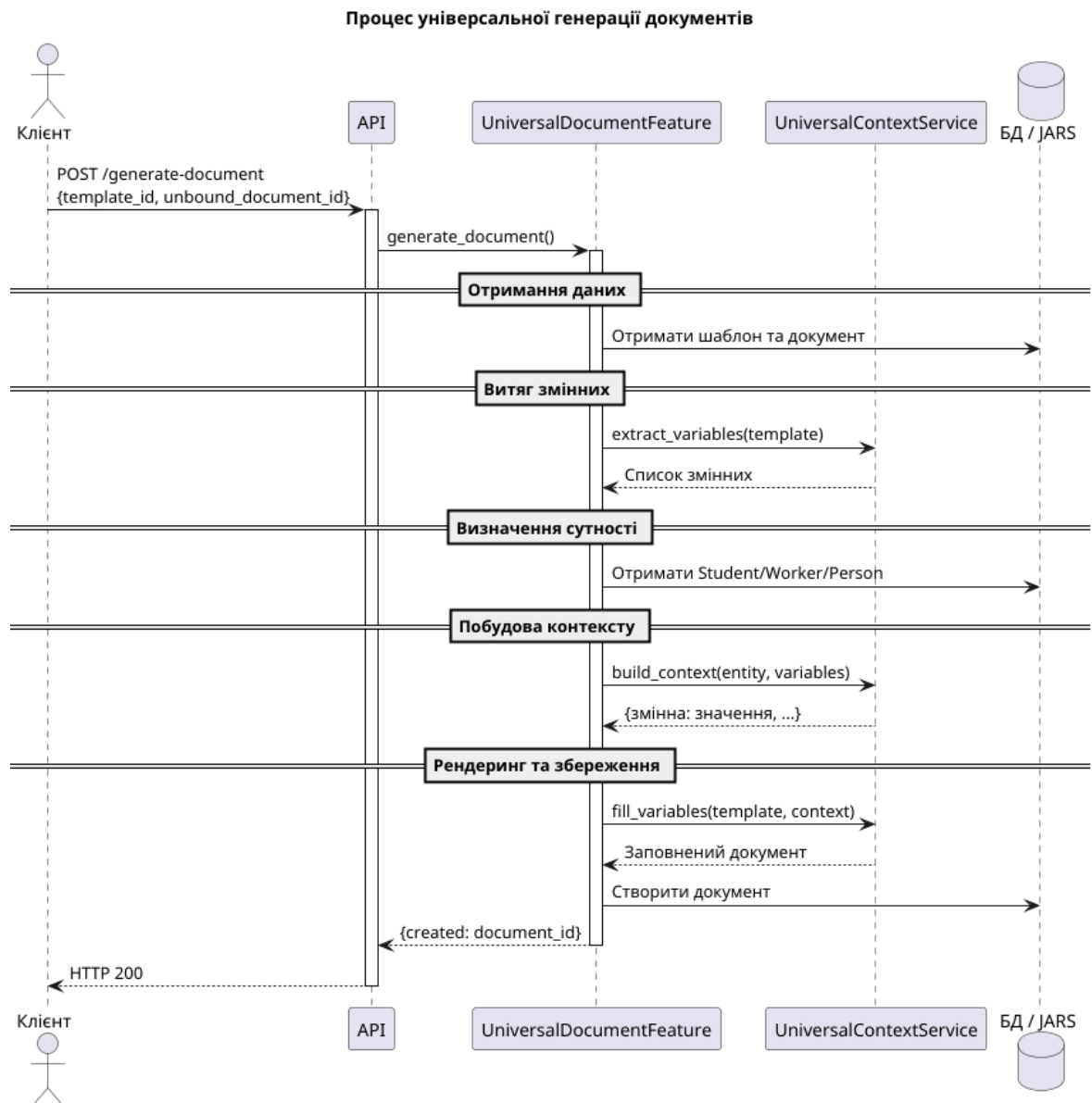


Рисунок 3.3. Діаграма послідовності універсальної генерації документів

загальну вартість навчання, статус акредитації програми. Сервіс правил автоматично підбирає відповідний шаблон контракту за параметрами студента (вікова група, форма навчання, рівень освіти, курс).

3.4. Реалізація електронного підпису

Модуль електронного підпису забезпечує інтеграцію з сервісом «Дія» для накладання кваліфікованого електронного підпису на документи.

Архітектура підсистеми підпису. Для підписання документів модуль документообігу використовує існуючий на платформі Django-додаток *diia*, який забезпечує взаємодію з API сервісу «Дія» (реєстрація гілок закладу, створення запитів на підписання, обробка callback-відповідей). Модуль документообігу інтегрується з ним через додаток *document_management*, що містить моделі життєвого циклу документа як об'єкта підписання:

- *DigitalDocumentOriginal* — оригінальний непідписаний документ із метаданими (вхідний/вихідний номери, дати, назва, зміст, файл, теги, тип, коротке посилання);
- *DigitalDocumentSignatureRequest* — запит на підписання документа, що пов'язує оригінал із підписаним результатом, містить автора, підписанта, статус та підтримує ланцюжок повторних підписів через самопосилання;
- *DigitalDocumentSigned* — підписаний документ із файлом підпису та повними метаданими сертифіката (серійний номер, дати видачі та закінчення, дані видавця та власника сертифіката).

Процес підписання складається з таких кроків:

- а) модуль документообігу генерує XML-версію документа на основі TipTap JSON-контенту;
- б) створюється об'єкт *DigitalDocumentOriginal* з оригіналом документа;
- в) обчислюється хеш документа та створюється *DigitalDocumentSignatureRequest*;

- г) запит передається до додатку *diia*, який формує запит на підписання з хешем та надсилає його до API Дія;
- д) API Дія повертає *deerlink*, який передається користувачу для переходу в застосунок «Дія» або на сторінку ID.GOV.UA для підпису через файловий ключ КЕП;
- е) користувач підписує документ обраним способом;
- ж) «Дія» надсилає *callback* з підписаними даними, який обробляється додатком *diia*;
- з) модуль документообігу отримує розшифровані дані підпису та зберігає підписаний документ (*DigitalDocumentSigned*) з метаданими сертифіката;
- и) крок процесу в документообігу переводиться у стан «Завершено».

На клієнтській частині підписання реалізовано через покрокові інтерфейси. Компонент *SigningFileMode* забезпечує попередній перегляд PDF-документа перед підписанням. Після підписання компонент *SignedDocumentInfo* відображає детальну інформацію про сертифікат (ПІБ власника, РНОКПП, видавець, серійний номер, термін дії) та генерує QR-код розміром 136×136 пікселів для мобільної верифікації. Компонент *SignedDocument* надає список підписаних документів з можливістю завантаження.

Підтримка ручного підпису. Для випадків, коли електронний підпис через «Дія» недоступний, передбачено прапорець *is_manually_signed* у моделі файлу, що дозволяє позначити документ як підписаний вручну.

Управління файлами. Компонент *FileUploaderModal* підтримує два режими: завантаження нових файлів (з валідацією типу PDF/DOCX, обмеженням до 99 файлів) та вибір із існуючих файлів або форм JARS. Файли завантажуються через проксі-сервер з паралельним обчисленням SHA-256 хешу для дедуплікації. Для перегляду реалізовано ав-

томатичне визначення типу файлу: PDF-файли відображаються через *PdfViewFile* з навігацією, масштабуванням та одно-/двосторінковим режимами, DOCX-файли — через *DocxViewer*.

Система публічних ofert. Для абітурієнтів реалізовано модуль публічних ofert із двома режимами: студентський (заповнення даних з інтеграцією документів Дія) та адміністративний (дворівневе затвердження: основне та військове для абітурієнтів чоловічої статі).

3.5. Демонстрація роботи системи

Для демонстрації функціональності розробленого модуля розглянемо основні інтерфейси системи та типовий сценарій використання — генерацію контракту зі студентом.

Візуальний редактор маршрутів (рис. 3.4) надає графічний інтерфейс для створення та редагування маршрутів документів. Процеси відображаються як вузли графу з інформацією про назву, підрозділ та кроки, а зв'язки між ними — як ребра, що визначають залежності.

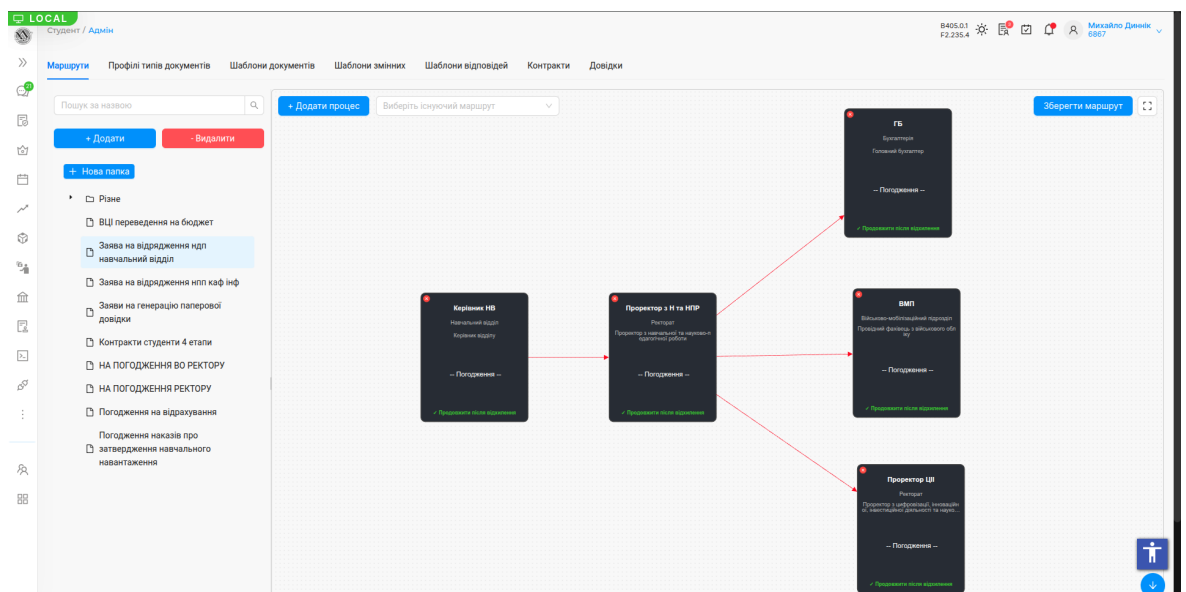
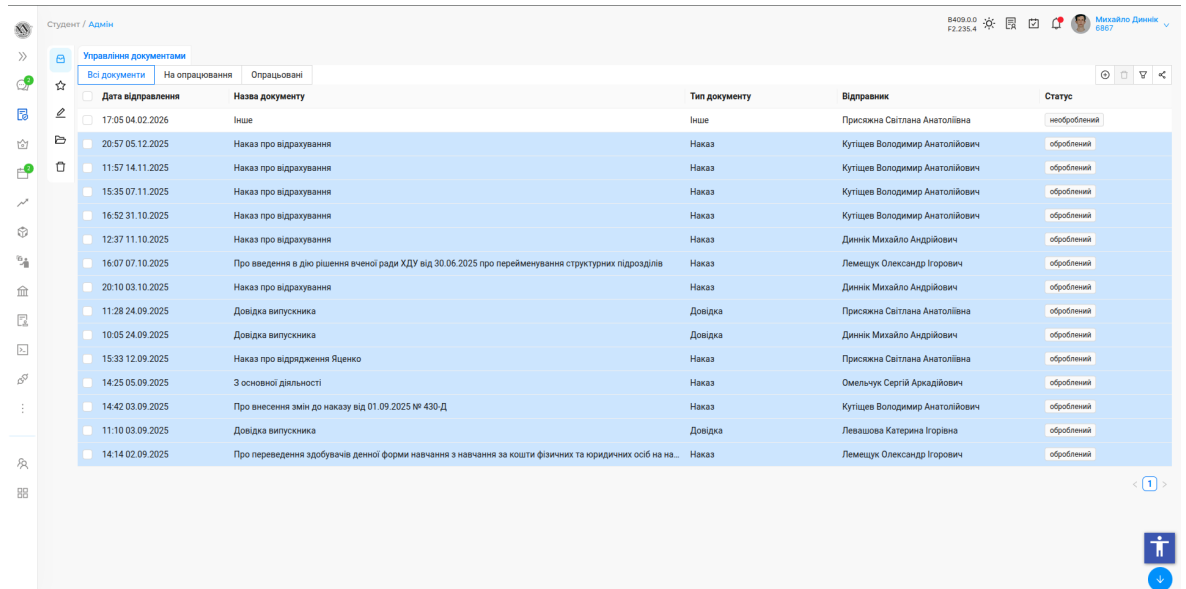


Рисунок 3.4. Візуальний редактор маршрутів

Центр документів (рис. 3.5) відображає список документів користувача з можливістю фільтрації за статусом, типом та напрямком. Кожен рядок містить основну інформацію: назву, тип, статус, відповідальний підрозділ та лічильник непрочитаних сповіщень.



Дата відправлення	Назва документу	Тип документу	Відправник	Статус
17.05.04.02.2026	Інше	Інше	Присяжна Світлана Анатоліївна	необработанный
20.57.05.12.2025	Наказ про відрахування	Наказ	Кутіщев Володимир Анатолійович	обработанный
11.57.14.11.2025	Наказ про відрахування	Наказ	Кутіщев Володимир Анатолійович	обработанный
15.35.07.11.2025	Наказ про відрахування	Наказ	Кутіщев Володимир Анатолійович	обработанный
16.52.31.10.2025	Наказ про відрахування	Наказ	Кутіщев Володимир Анатолійович	обработанный
12.37.11.10.2025	Наказ про відрахування	Наказ	Диннік Михайло Андрійович	обработанный
16.07.07.10.2025	Про введення в дію рішення вченої ради ХДУ від 30.06.2025 про перейменування структурних підрозділів	Наказ	Лемешук Олександр Ігорович	обработанный
20.10.03.10.2025	Наказ про відрахування	Наказ	Диннік Михайло Андрійович	обработанный
11.28.24.09.2025	Довідка випускника	Довідка	Присяжна Світлана Анатоліївна	обработанный
10.05.24.09.2025	Довідка випускника	Довідка	Диннік Михайло Андрійович	обработанный
15.33.12.09.2025	Наказ про відрахування Яценко	Наказ	Присяжна Світлана Анатоліївна	обработанный
14.25.05.09.2025	З основної діяльності	Наказ	Омельчук Сергій Аркадійович	обработанный
14.42.03.09.2025	Про внесення змін до наказу від 01.09.2025 № 430-Д	Наказ	Кутіщев Володимир Анатолійович	обработанный
11.10.03.09.2025	Довідка випускника	Довідка	Левашова Катерина Ігорівна	обработанный
14.14.02.09.2025	Про переведення здобувачів денної форми навчання з навчання за кошти фізичних та юридичних осіб на...	Наказ	Лемешук Олександр Ігорович	обработанный

Рисунок 3.5. Центр документів

Список вступників (рис. 3.6) відображає дані абітурієнтів, синхронізовані з ЄДЕБО: ПІБ, дату народження, курс, форму навчання, відділення, контактну особу, рівень освіти, тип навчання та поточний статус. Вкладки дозволяють фільтрувати записи за стадією обробки.

Сторінка перегляду документа (рис. 3.7) містить метадані документа, візуалізацію поточного стану маршруту з кольоровими індикаторами статусів процесів.

Інтерфейс електронного підпису (рис. 3.8) надає попередній перегляд документа та можливість підписання через сервіс «Дія». Після підписання відображається інформація про сертифікат із QR-кодом для верифікації.

Модуль налаштувань (рис. 3.9) надає адміністраторам інтерфейс для конфігурації системи документообігу. Модуль містить сім вкладок:

LOCAL

Сторінка / Адмін

8405.0.1
F2.335.4

<

Рисунок 3.6. Список вступників з даними ЄДЕБО

Управління документами **Наказ про відрахування** ×

Скасувати Зберегти

Тип документу / Заголовок справи: Наказ / Про відрахування / Наказ про відрахування

Індекс: 66-66

Заголовок електронного документу: Наказ про відрахування

Термін виконання: 08-12-2025

Напрям документу: Внутрішній

Реєстраційний номер: [Оберіть дату]

Вихідний номер: [Оберіть дату]

Вихідна дата: [Оберіть дату]

Відділ: Загальний відділ

Відповідальний виконавець: Купіца Володимир Анатолійович

Відправник: [Оберіть дату]

Контактна особа відправника: [Оберіть дату]

Опис документу / завдання: Наказ_про_відрахування_2025-12-05

Прикріплені документи: 20.58.11 05/12/2025 [Дія підпису] Наказ_про_відрахування_20251205_203951.pdf

Наказ про відрахування Відкрити документ В процесі

Дата початку: 05.12.2025 21:34

Бухгалтерія: Відповідальна особа: Попова Ірина Михайлівна 1. Погодження Очікується

Юридичний відділ: Відповідальна особа: Нікітська Ірина Олександрівна 1. Погодження Очікується

Уповноважений з АД: Відповідальна особа: Савкович Олександр Олександрович 1. Погодження Очікується

Проректор з ННП: Відповідальна особа: Мальшикова Лар'я Сергіївна 1. Погодження Очікується

Рисунок 3.7. Сторінка перегляду документа з візуалізацією маршруту

- *Маршрути* — візуальний редактор маршрутів з деревом папок та графічним редактором процесів;
- *Профілі типів документів* — налаштування типів документів із прив'язкою стандартного маршруту, підрозділу та терміну зберігання;
- *Шаблони документів* — управління WYSIWYG-шаблонами для генерації документів та конфігуратор фрагментів для повторного використання частин шаблонів;

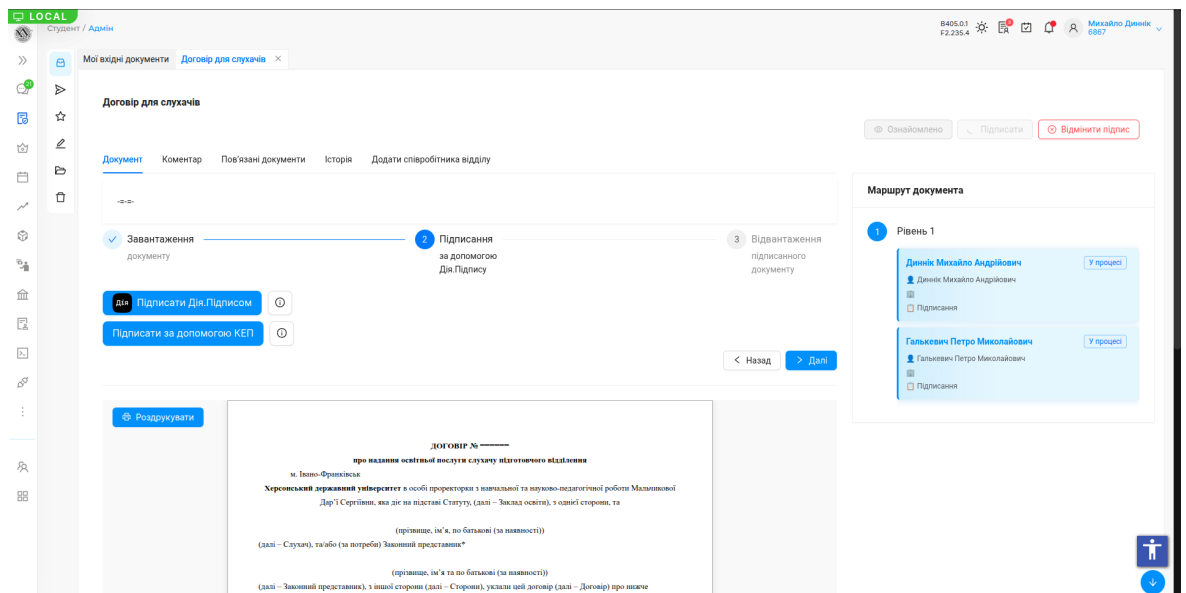


Рисунок 3.8. Інтерфейс електронного підпису через «Дія»

- *Шаблони змінних* — групи змінних для підстановки в шаблони з визначенням типу та назви;
- *Шаблони відповідей* — типові відповіді для швидкого заповнення коментарів при погодженні та відхиленні;
- *Контракти* — правила автоматичного підбору шаблонів контрактів за параметрами студента;
- *Довідки* — налаштування генерації довідок різних типів.

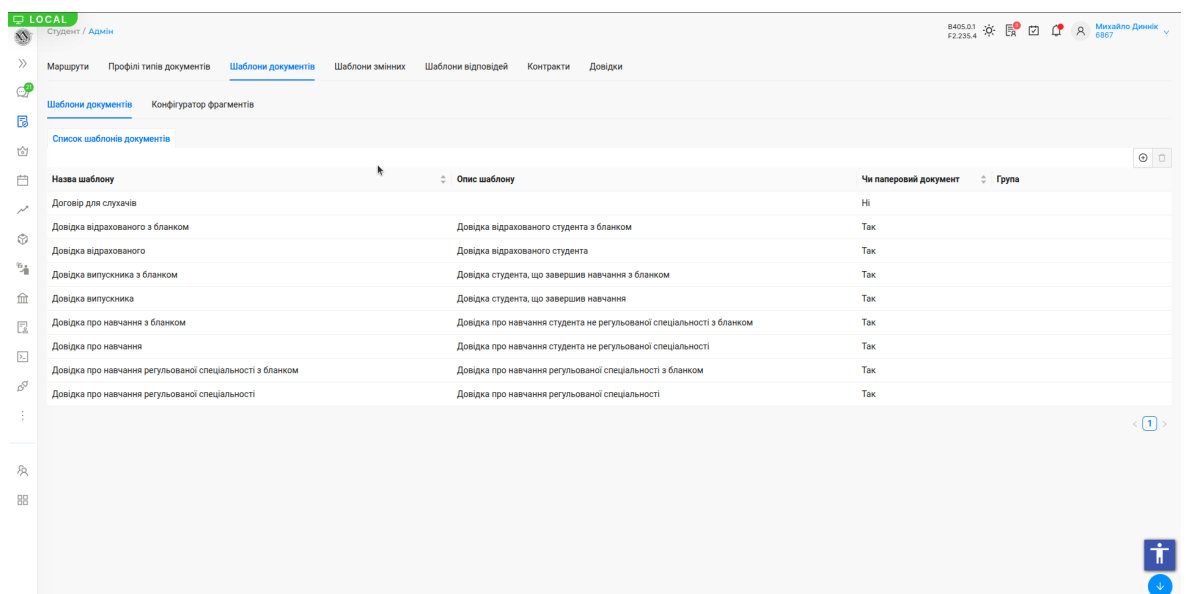


Рисунок 3.9. Модуль налаштувань документообігу

Сценарій генерації контракту. Розглянемо типовий сценарій використання системи.

Крок 1. Створення маршруту. Адміністратор відкриває візуальний редактор маршрутів та створює маршрут для контракту. Маршрут містить три процеси: «Перегляд студентом» (крок: перегляд), «Підписання студентом» (крок: підпис через Дія), «Підписання деканом» (крок: підпис через Дія). Процес підписання студентом залежить від процесу перегляду, а процес підписання деканом — від процесу підписання студентом.

Крок 2. Генерація контракту. Під час вступної кампанії адміністратор приймальної комісії відкриває модуль контрактів. Система відображає список абітурієнтів, дані яких синхронізовані з ЄДЕБО. Адміністратор обирає абітурієнта та ініціює генерацію контракту. Система автоматично підбирає шаблон контракту за параметрами студента (вікова група, форма навчання, джерело фінансування), заповнює шаблон даними та створює документообіг з обраним маршрутом.

Крок 3. Проходження маршруту. Студент отримує сповіщення, переглядає контракт, після чого отримує deeplink для підписання через «Дія». Після підписання студентом система автоматично активує наступний процес, і декан отримує сповіщення для підписання. Після підписання деканом контракт переходить у стан «Завершено».

Усі дії фіксуються в аудиті: система зберігає інформацію про кожну зміну статусу, автора зміни та час виконання. Це забезпечує повну прозорість процесу та можливість відстеження будь-яких проблем. Детальний приклад використання системи з покроковими ілюстраціями наведено у додатку А.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було спроектовано та реалізовано модуль електронного документообігу з динамічною маршрутизацією на платформі KSU24. Отримано такі результати:

- а) Проведено аналіз предметної області електронного документообігу у закладі вищої освіти. Визначено специфічні вимоги університетського середовища: різноманітність типів документів, складна організаційна структура, масовість операцій, інтеграція з державними системами та юридична значущість документів. Виконано порівняльний аналіз шести існуючих рішень (Alfresco, Paperless-ngx, Вчасно, Документ.Онлайн, АСКОД Онлайн, NextCloud), що підтвердив відсутність системи, яка повністю задовольняє потреби університету.
- б) Спроектовано архітектуру модуля на основі чотиришарової моделі (API, бізнес-логіка, сервіси, дані) з 15 моделями даних. Розроблено систему динамічної маршрутизації, що підтримує лінійні, паралельні та ієрархічні процеси з трьома рівнями зв'язків. Спроектовано скінченний автомат життєвого циклу документа та інтеграції з зовнішніми системами.
- в) Реалізовано серверну частину модуля на Django з 72 API-ендпоінтами у 12 функціональних групах, системою аудиту та асинхронними задачами Celery для масових операцій. Реалізовано клієнтську частину на React з візуальним редактором маршрутів на AntV X6, WYSIWYG-редактором шаблонів на TipTap з трьома кастомними розширеннями та модулями управління документами, сповіщеннями та коментарями.
- г) Забезпечено інтеграцію з сервісом «Дія» для кваліфікованого електронного підпису, з ЄДЕБО для отримання даних студентів, з

Google Drive для хмарного зберігання файлів та з JARS для управління шаблонами та маршрутами.

- д) Реалізовано систему універсальної генерації документів із шаблонів з автоматичною підстановкою змінних через реєстр резолверів, що дозволяє додавати нові типи документів без зміни існуючого коду.

Розроблений модуль впроваджено в університетську платформу KSU24 та забезпечує автоматизацію повного циклу роботи з документами: від створення та маршрутизації до підписання та архівування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Aalst W. M. P. van der, Hee K. M. van.* Workflow Management: Models, Methods, and Systems. — Cambridge : MIT Press, 2002. — С. 384.
2. Activiti – Open source business automation. — 2026. — URL: <https://www.activiti.org/> (дата зверн. 15.01.2026).
3. Ant Design – A design system for enterprise-level products. — 2026. — URL: <https://ant.design/> (дата зверн. 15.01.2026).
4. AntV X6 – JavaScript diagramming library. — 2026. — URL: <https://x6.antv.antgroup.com/en> (дата зверн. 15.01.2026).
5. Celery – Distributed task queue. — 2026. — URL: <https://docs.celeryq.dev/en/stable/> (дата зверн. 15.01.2026).
6. CKEditor 5 – Modern rich text editor. — 2026. — URL: <https://ckeditor.com/ckeditor-5/> (дата зверн. 15.01.2026).
7. *Django Software Foundation.* Django documentation. — 2026. — URL: <https://docs.djangoproject.com/en/5.0/> (дата зверн. 15.01.2026).
8. *Docker Inc.* Docker documentation. — 2026. — URL: <https://docs.docker.com/> (дата зверн. 15.01.2026).
9. *Fielding R. T.* Architectural Styles and the Design of Network-based Software Architectures : дис. ... док. / Fielding R. T. — University of California, Irvine, 2000. — С. 162.
10. Fundamentals of Business Process Management / M. Dumas [та ін.]. — 2nd. — Berlin : Springer, 2018. — С. 527.
11. *Google Cloud.* Google Drive API documentation. — 2026. — URL: <https://developers.google.com/drive/api> (дата зверн. 15.01.2026).

12. *Hohpe G., Woolf B.* Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. — Boston : Addison-Wesley, 2003. — C. 736.
13. *Hyland Software.* Alfresco Community Edition documentation. — 2026. — URL: <https://docs.alfresco.com/> (дата зверн. 15.01.2026).
14. *Kleppmann M.* Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. — Sebastopol : O'Reilly Media, 2017. — C. 614.
15. *Martin R. C.* Clean Architecture: A Craftsman's Guide to Software Structure and Design. — Boston : Prentice Hall, 2017. — C. 432.
16. *Massé M.* REST API Design Rulebook. — Sebastopol : O'Reilly Media, 2011. — C. 112.
17. *Newman S.* Building Microservices: Designing Fine-Grained Systems. — 2nd. — Sebastopol : O'Reilly Media, 2021. — C. 612.
18. Nextcloud – Open source content collaboration platform. — 2026. — URL: <https://nextcloud.com/> (дата зверн. 15.01.2026).
19. Paperless-ngx – Document management system. — 2026. — URL: <https://docs.paperless-ngx.com/> (дата зверн. 15.01.2026).
20. *PostgreSQL Global Development Group.* PostgreSQL 16 documentation. — 2026. — URL: <https://www.postgresql.org/docs/16/> (дата зверн. 15.01.2026).
21. ProseMirror – A toolkit for building rich-text editors. — 2026. — URL: <https://prosemirror.net/docs/> (дата зверн. 15.01.2026).
22. Pydantic – Data validation using Python type annotations. — 2026. — URL: <https://docs.pydantic.dev/latest/> (дата зверн. 15.01.2026).

23. Quill – Your powerful rich text editor. — 2026. — URL: <https://quilljs.com/> (дата зверн. 15.01.2026).
24. React – The library for web and native user interfaces. — 2026. — URL: <https://react.dev/> (дата зверн. 15.01.2026).
25. *Redis Ltd.* Redis documentation. — 2026. — URL: <https://redis.io/docs/> (дата зверн. 15.01.2026).
26. Slate – A completely customizable framework for building rich text editors. — 2026. — URL: <https://www.slatejs.org/> (дата зверн. 15.01.2026).
27. *Softline.* АСКОД — автоматизована система контролю та організації документообігу. — 2026. — URL: <https://askod.ua/> (дата зверн. 15.01.2026).
28. TanStack Query – Powerful asynchronous state management. — 2026. — URL: <https://tanstack.com/query/latest> (дата зверн. 15.01.2026).
29. TipTap – The headless rich text editor framework. — 2026. — URL: <https://tiptap.dev/docs> (дата зверн. 15.01.2026).
30. *Верховна Рада України.* Про електронні документи та електронний документообіг: Закон України № 851-IV від 22.05.2003. — 2003. — Відомості Верховної Ради України.
31. *Верховна Рада України.* Про електронну ідентифікацію та електронні довірчі послуги: Закон України № 2155-VIII від 05.10.2017. — 2017. — Відомості Верховної Ради України.
32. *Вчасно.* Вчасно — сервіс електронного документообігу. — 2026. — URL: <https://vchasno.ua/> (дата зверн. 15.01.2026).

33. *Документ.Онлайн*. Документ.Онлайн — хмарний сервіс електронного документообігу. — 2026. — URL: <https://document.online/> (дата зверн. 15.01.2026).
34. *Лемешук О. І., Юрчук Ю. Ю., Сенчишен Д. О.* Віртуальні освітні середовища: допомога абітурієнтові в кризових умовах // Збірник наукових праць «Педагогічні науки». — 2025. — № 109. — С. 60—69. — DOI: 10.32999/ksu2413-1865/2025-109-8.
35. *Матвієнко О. В., Цивін М. Н.* Основи організації електронного документообігу. — Київ : Центр учбової літератури, 2008. — С. 112.
36. *Міністерство освіти і науки України.* Єдина державна електронна база з питань освіти. — 2026. — URL: <https://info.edbo.gov.ua/> (дата зверн. 15.01.2026).
37. *Міністерство цифрової трансформації України.* Дія. Державні послуги онлайн. — 2026. — URL: <https://diia.gov.ua/> (дата зверн. 15.01.2026).
38. *Сенчишен Д. О., Лемешук О. І.* Тимчасові та постійні рішення інтеграції сторонніх сервісів у віртуальне освітнє середовище KSU24 // Вимірювальна та обчислювальна техніка в технологічних процесах. — 2025. — № 1. — С. 235—243. — DOI: 10.31891/2219-9365-2025-81-29.

ДОДАТКИ

Додаток А

Приклад використання: оформлення довідки випускника

Розглянемо реальний сценарій роботи модуля електронного документообігу — оформлення довідки випускника. Цей процес демонструє повний цикл документообігу: від створення документа та проходження маршруту погодження до генерації документа за шаблоном та завантаження підписаного файлу.

Етап 1. Підготовка маршруту. Адміністратор відкриває візуальний редактор маршрутів, де створено маршрут для довідки випускника (рис. А.1). Маршрут містить п'ять процесів погодження: факультет ФКС (погодження кошторисної), відділ кадрів, ректор, бухгалтерія та кадри (створення наказу) — кожен з кроками ознайомлення та погодження. Залежності між процесами визначають порядок проходження.

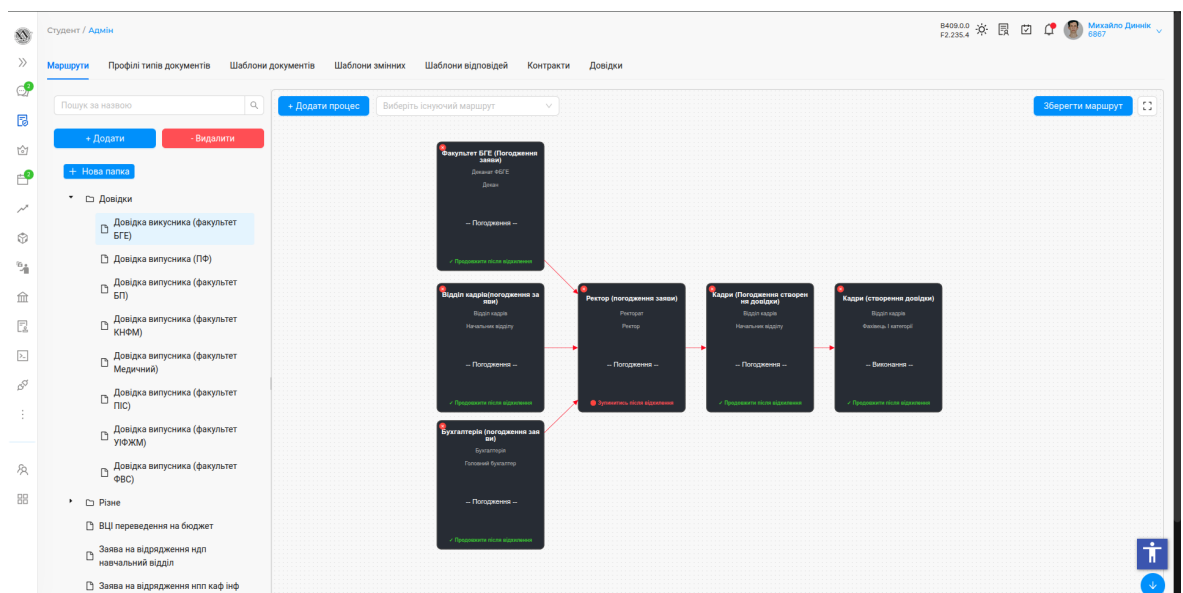


Рисунок А.1. Маршрут для довідки випускника у візуальному редакторі

Етап 2. Створення необробленого документа. Адміністратор створює новий необроблений документ типу «Довідка / Випускник /

Довідка випускника» (рис. А.2). Заповнюються метадані: напрям документа (вхідний), відділ, відповідальний виконавець, опис та термін виконання. Система автоматично присвоює вхідний номер та дату реєстрації.

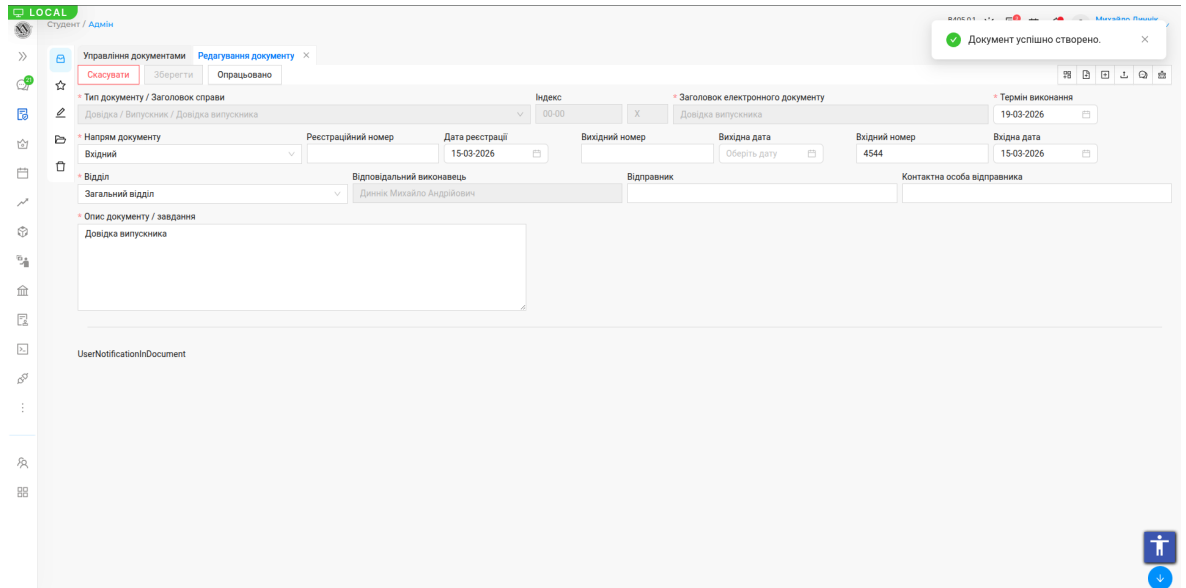


Рисунок А.2. Створення необробленого документа

Етап 3. Завантаження файлу. До документа прикріплюється файл для підпису через модальне вікно роботи з файлами (рис. А.3). Система розділяє файли на дві категорії: «Для підпису» (основний документ) та «Додаткові файли» (супровідні матеріали).

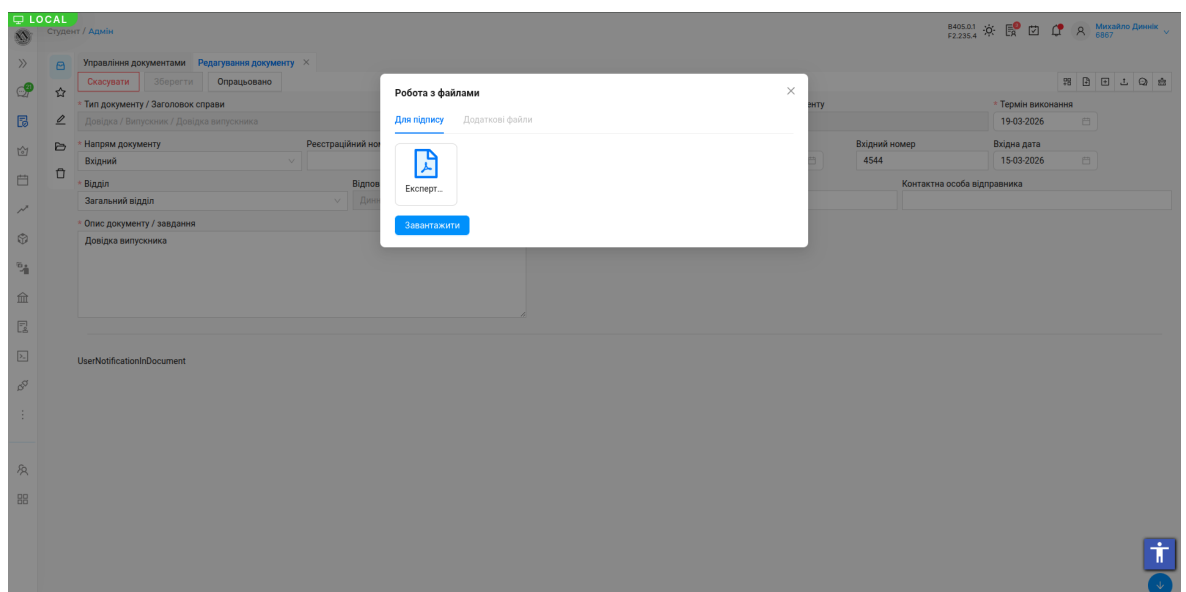


Рисунок А.3. Модальне вікно завантаження файлу

Після успішного завантаження файл відображається у секції «Прикріплені документи» з міткою «для підпису» (рис. А.4).

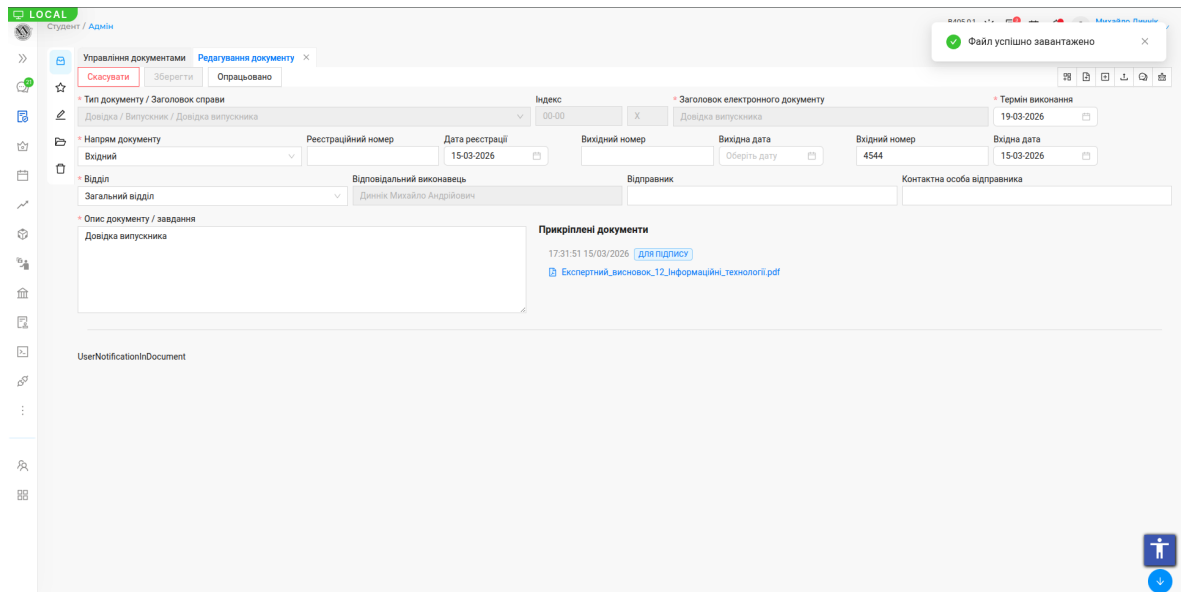


Рисунок А.4. Документ з прикріпленням файлом

Етап 4. Створення документообігу. Адміністратор натискає кнопку «Створити внутрішній документ» (рис. А.5), що ініціює перехід від необробленого документа до повноцінного документообігу з маршрутом.

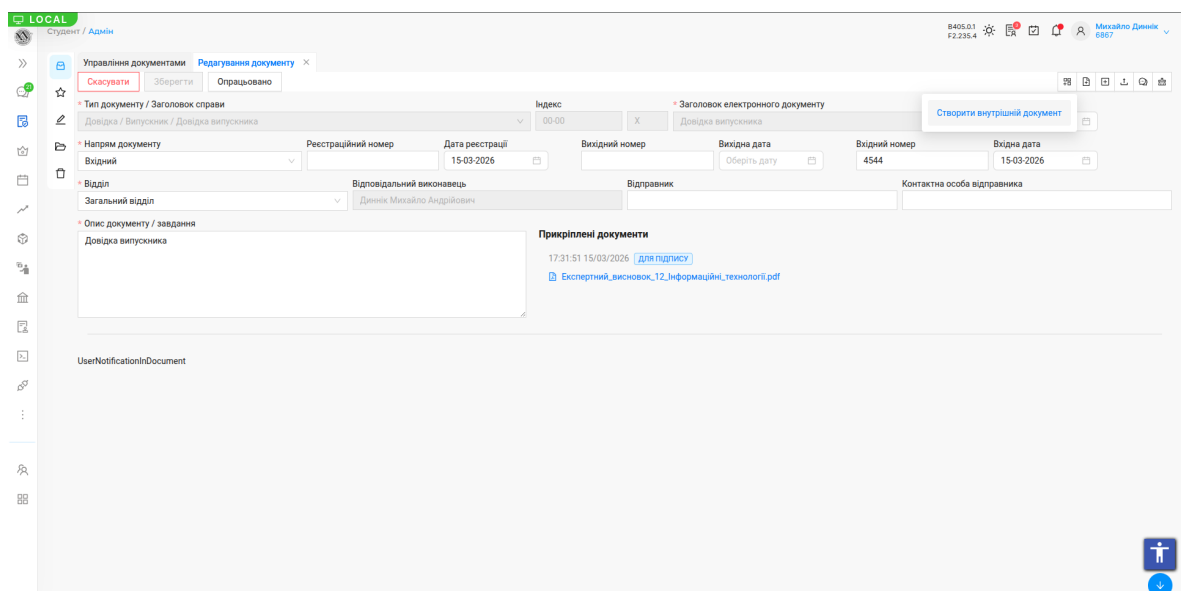


Рисунок А.5. Ініціювання створення документообігу

Відкривається форма створення документообігу (рис. А.6), де відображається обраний маршрут з усіма процесами та відповідальними

особами. Адміністратор перевіряє учасників кожного процесу та підтверджує створення.

The screenshot shows the 'LOCAL' system interface for document management. The top bar indicates the user is 'Михайло Деник' (Mikhail Denik) with ID 6867. The main workspace is divided into several sections:

- Управління документами** (Document Management): Includes buttons for 'Скасувати' (Cancel) and 'Зберегти' (Save).
- Довідка випусника** (Graduate Certificate): Fields for 'Направ документу' (Document Direction), 'Термін виконання' (Execution Term), 'Тип документа' (Document Type), 'Відділ' (Department), and 'Рівень доступу' (Access Level).
- Довідка випусника** (Graduate Certificate): Fields for 'Індекс' (Index), 'Номер документа' (Document Number), 'Назва документа' (Document Name), and 'Відповідальна особа' (Responsible Person).
- Довідка випусника** (Graduate Certificate): A section for 'Додати пов'язаний документ' (Add related document) and 'Маршрути' (Routes).
- Довідка випусника** (Graduate Certificate): A section for 'Додати процес' (Add process).
- Факкультет ФВС (погодження заяви)** (Faculty FVS (Statement Approval)): A section for 'Відповідальний' (Responsible) with a dropdown menu.
- Відділ кадрів (погодження заяви)** (Department of Personnel (Statement Approval)): A section for 'Відповідальний' (Responsible) with a dropdown menu.
- Бухгалтерія (погодження заяви)** (Accounting (Statement Approval)): A section for 'Відповідальний' (Responsible) with a dropdown menu.
- Кадри (створення довідки)** (Personnel (Creation of Certificate)): A section for 'Відповідальний' (Responsible) with a dropdown menu.

Рисунок А.6. Форма створення документообігу з візуалізацією маршруту

Етап 5. Запуск маршруту. Після створення документообігу відображається сторінка документа з повною інформацією: метадані, прикріплені файли та маршрут з процесами у стані «Очікується» (рис. А.7). Адміністратор натискає кнопку «Почати маршрут» для запуску проходження.

The screenshot shows the 'LOCAL' system interface for document management. The top bar indicates the user is 'Михайло Деник' (Mikhail Denik) with ID 6867. The main workspace is divided into several sections:

- Управління документами** (Document Management): Includes buttons for 'Скасувати' (Cancel), 'Зберегти' (Save), and 'Опрацьовано' (Processed).
- Довідка / Випусник / Довідка випусника** (Certificate / Graduate / Graduate Certificate): Fields for 'Тип документа / Заголовок справи' (Document Type / Case Title), 'Індекс' (Index), 'Заголовок електронного документа' (Electronic Document Title), and 'Термін виконання' (Execution Term).
- Довідка / Випусник / Довідка випусника** (Certificate / Graduate / Graduate Certificate): Fields for 'Направ документу' (Document Direction), 'Регістраційний номер' (Registration Number), 'Дата реєстрації' (Registration Date), 'Вихідний номер' (Output Number), 'Вихідна дата' (Output Date), 'Відправник' (Sender), and 'Контактна особа відправника' (Sender's Contact Person).
- Відділ** (Department): A dropdown menu for 'Відділ' (Department).
- Відповідальний виконавець** (Responsible Executive): A dropdown menu for 'Відповідальний виконавець' (Responsible Executive).
- Відправник** (Sender): A dropdown menu for 'Відправник' (Sender).
- Контактна особа відправника** (Sender's Contact Person): A dropdown menu for 'Контактна особа відправника' (Sender's Contact Person).
- Опис документа / завдання** (Document Description / Task): A section for 'Опис документа / завдання' (Document Description / Task).
- Прикріплені документи** (Attached Documents): A section for 'Прикріплені документи' (Attached Documents) with a list of documents and a 'Почати маршрут' (Start Route) button.
- Довідка випусника** (Graduate Certificate): A section for 'Довідка випусника' (Graduate Certificate) with a 'Почати маршрут' (Start Route) button.
- Дата початку: Invalid Date** (Start Date: Invalid Date): A section for 'Дата початку: Invalid Date' (Start Date: Invalid Date).
- Факкультет ФВС (погодження заяви)** (Faculty FVS (Statement Approval)): A section for 'Відповідальна особа' (Responsible Person) with a dropdown menu.
- Відділ кадрів (погодження заяви)** (Department of Personnel (Statement Approval)): A section for 'Відповідальна особа' (Responsible Person) with a dropdown menu.
- Бухгалтерія (погодження заяви)** (Accounting (Statement Approval)): A section for 'Відповідальна особа' (Responsible Person) with a dropdown menu.

Рисунок А.7. Документообіг створено, маршрут готовий до запуску

Етап 6. Погодження учасниками. Відповідальна особа першого процесу отримує сповіщення та відкриває документ. На сторінці відображається попередній перегляд PDF-файлу та візуалізація маршруту з кольоровими індикаторами статусів (рис. А.8). Учасник може ознайомитися з документом та погодити його.

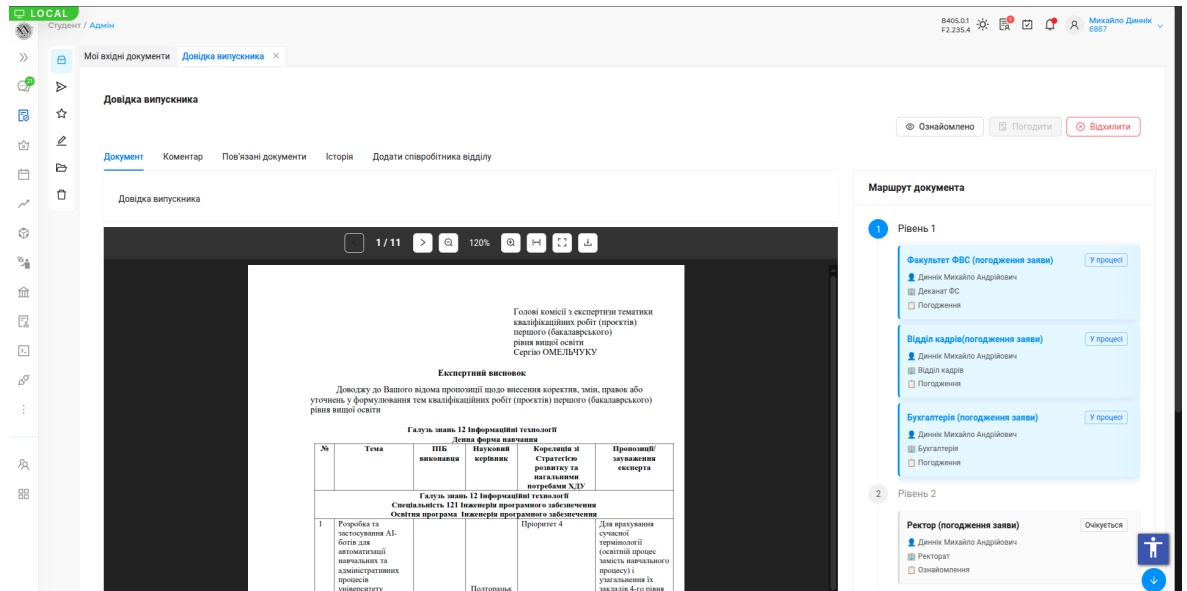


Рисунок А.8. Перегляд документа учасником процесу

Після погодження першим учасником система автоматично активує наступний процес у ланцюжку (рис. А.9). Візуалізація маршруту оновлюється — завершені процеси позначаються відповідним кольором.

Процес погодження продовжується послідовно для кожного підрозділу відповідно до залежностей маршруту. Кожен учасник отримує сповіщення, переглядає документ та погоджує.

Етап 7. Генерація документа за шаблоном. Після завершення всіх процесів погодження адміністратор ініціює генерацію довідки. Система пропонує заповнити дані профілю студента: ПІБ, категорія та освітня програма (рис. А.10).

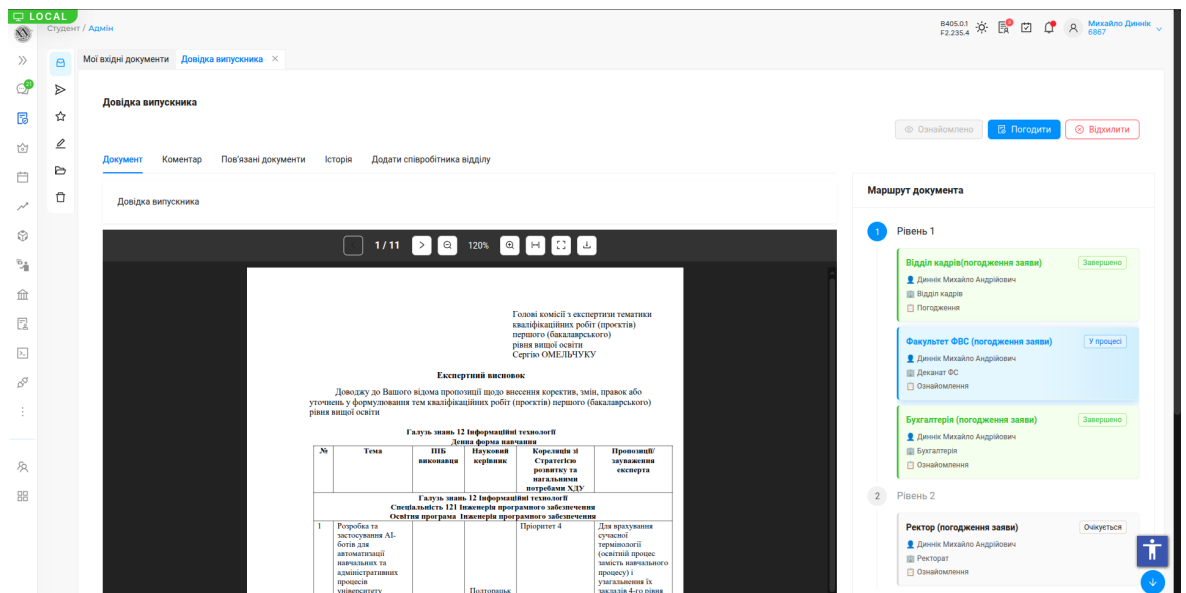


Рисунок А.9. Маршрут після погодження першим учасником

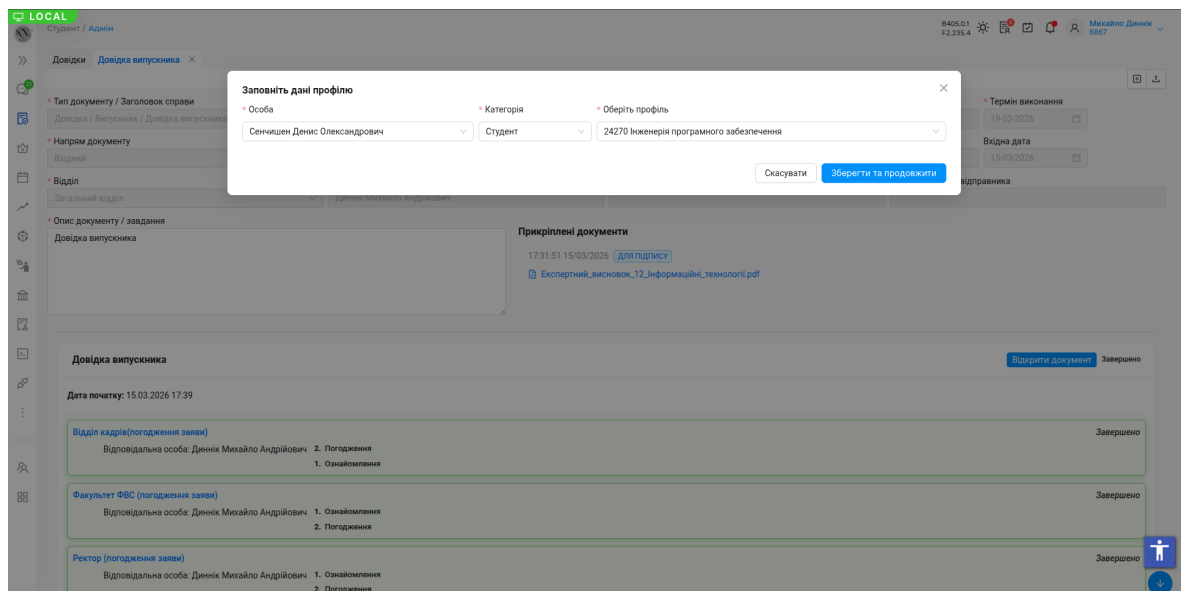


Рисунок А.10. Заповнення даних профілю для генерації довідки

Далі відкривається вікно вибору шаблону документа (рис. А.11). Для довідки випусника доступні варіанти: «Довідка випусника з бланком» (паперова) та «Довідка випусника» (паперова).

Система генерує довідку, автоматично підставляючи дані студента у шаблон (рис. А.12): ПІБ, дата народження, номер наказу про зарахування, спеціальність, освітня програма, факультет, рівень вищої освіти, форма навчання та джерело фінансування.

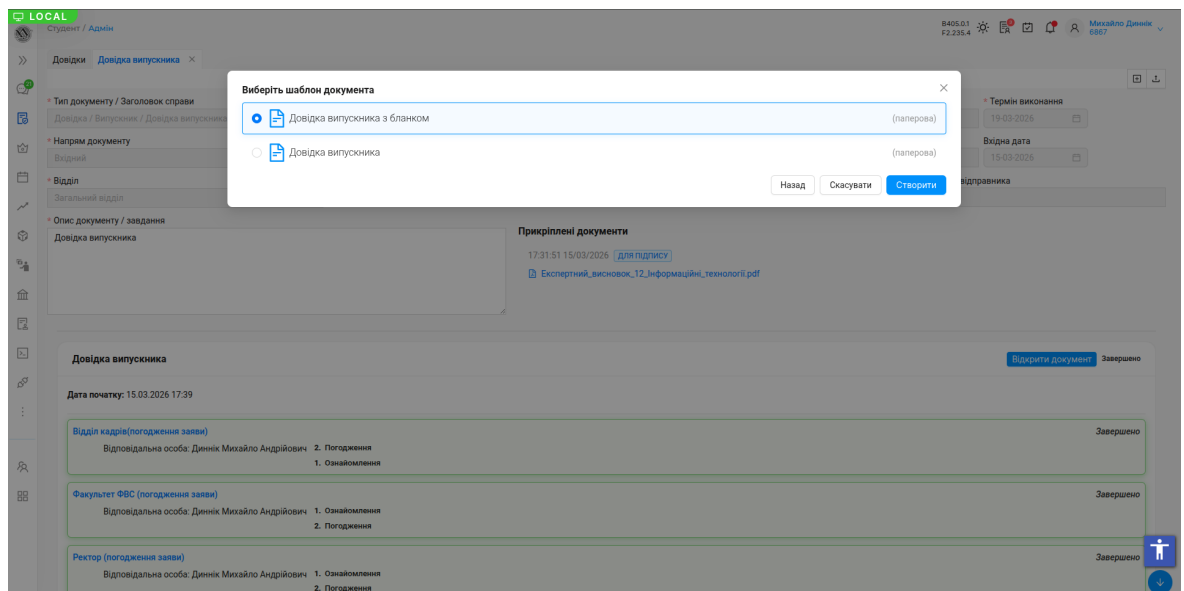


Рисунок А.11. Вибір шаблону документа

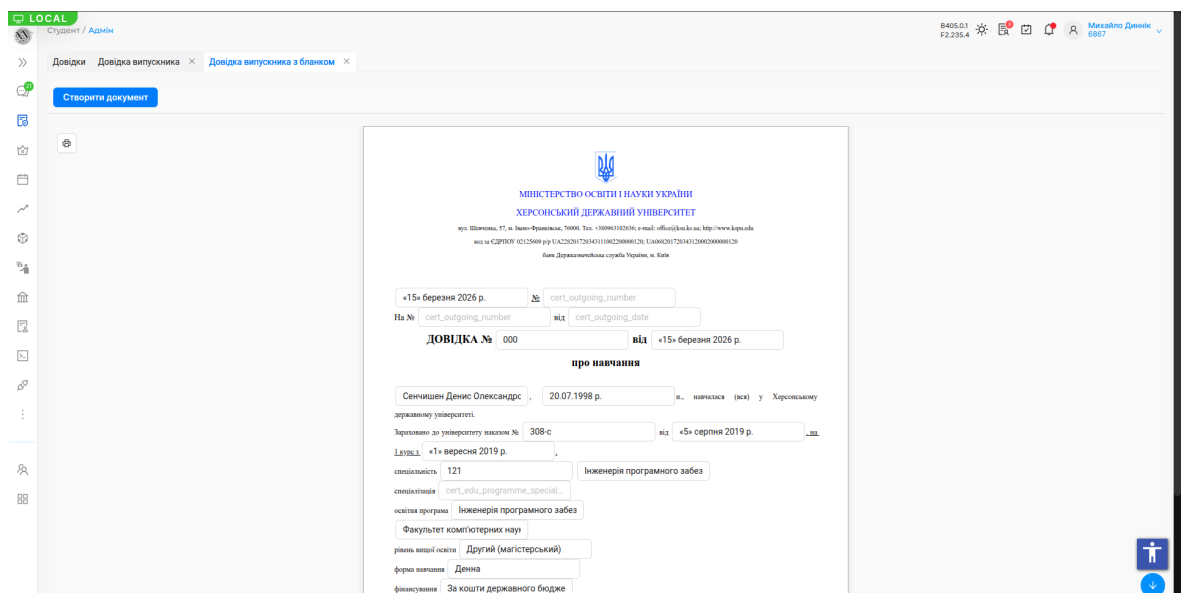


Рисунок А.12. Згенерована довідка випусника з автоматично заповненими даними

Етап 8. Перегляд у модулі довідок. Згенерована довідка відображається у модулі «Довідки» (рис. А.13), де кожен рядок містить інформацію про студента, профіль, період навчання, тип довідки та поточний статус. Кольорова індикація рядків дозволяє швидко ідентифікувати стан кожної довідки.

Дата відправлення	Назва довідки	Особа	Категорія	Профіль	Період / Фінансування / Навчання	Тип довідки	Заява	Довідка
17.30 15.03.2026	Випускник	Сенчишен Денис Олександров...	Студент	24270 Інженерія програмного забезпечення	2019 - 2020 / За кошти державного бюджету / Денна	Паперова	Звернуто	Згенеровано
20.17 26.02.2026	Випускник	Сенчишен Денис Олександров...	Студент	24270 Інженерія програмного забезпечення	2019 - 2020 / За кошти державного бюджету / Денна	Паперова	Відхилено	Згенеровано
16.17 26.02.2026	Випускник	Сенчишен Денис Олександров...	Студент	24270 Інженерія програмного забезпечення	2019 - 2020 / За кошти державного бюджету / Денна	Паперова	Відхилено	Згенеровано
11.18 24.02.2026	Випускник	-	-	-	- / - / -	-	В процесі	-
21.04 18.02.2026	Випускник	Сенчишен Денис Олександров...	Студент	24270 Інженерія програмного забезпечення	2019 - 2020 / За кошти державного бюджету / Денна	Паперова	Відхилено	Згенеровано
11.28 24.09.2025	Випускник	Перов Микола Михайлович	Студент	23691 Готельно-ресторанна справа	2019 - 2023 / За кошти фізичних та юридичних осіб / Д...	Паперова	Звернуто	Згенеровано
10.05 24.09.2025	Випускник	Сенчишен Денис Олександров...	Студент	24270 Інженерія програмного забезпечення	2019 - 2020 / За кошти державного бюджету / Денна	Паперова	Звернуто	Підписано
11.10 03.09.2025	Випускник	-	Студент	24270 Інженерія програмного забезпечення	- / - / -	Паперова	Звернуто	Підписано

Рисунок А.13. Модуль «Довідки» зі списком згенерованих довідок

Етап 9. Завантаження підписаного документа. Після друку та підписання паперової довідки адміністратор завантажує скановану копію підписаного документа через відповідне модальне вікно (рис. А.14).

Завантажте підписаний файл

Завантажте підписаний файл

Довідка ...

Вибірть файл

Завантажити

Довідка випускника

Дата початку: 15.03.2026 17:39

Відділ кадрів(погодження заяви)

Відповідальна особа: Деник Михайло Андрійович

2. Погодження

1. Ознайомлення

Факультет ФБС (погодження заяви)

Відповідальна особа: Деник Михайло Андрійович

2. Погодження

1. Ознайомлення

Ректор (погодження заяви)

Відповідальна особа: Деник Михайло Андрійович

2. Погодження

1. Ознайомлення

Рисунок А.14. Завантаження підписаного файлу

Етап 10. Завершення документообігу. Фінальний стан докумен-тообігу відображає повну картину (рис. А.15): усі процеси маршруту мають статус «Завершено», в секції «Прикріплені документи» міститься оригінальний файл, а в секції «Електронні документи» — згенерована довідка з бланком. Повна історія дій зберігається в аудиті документа.

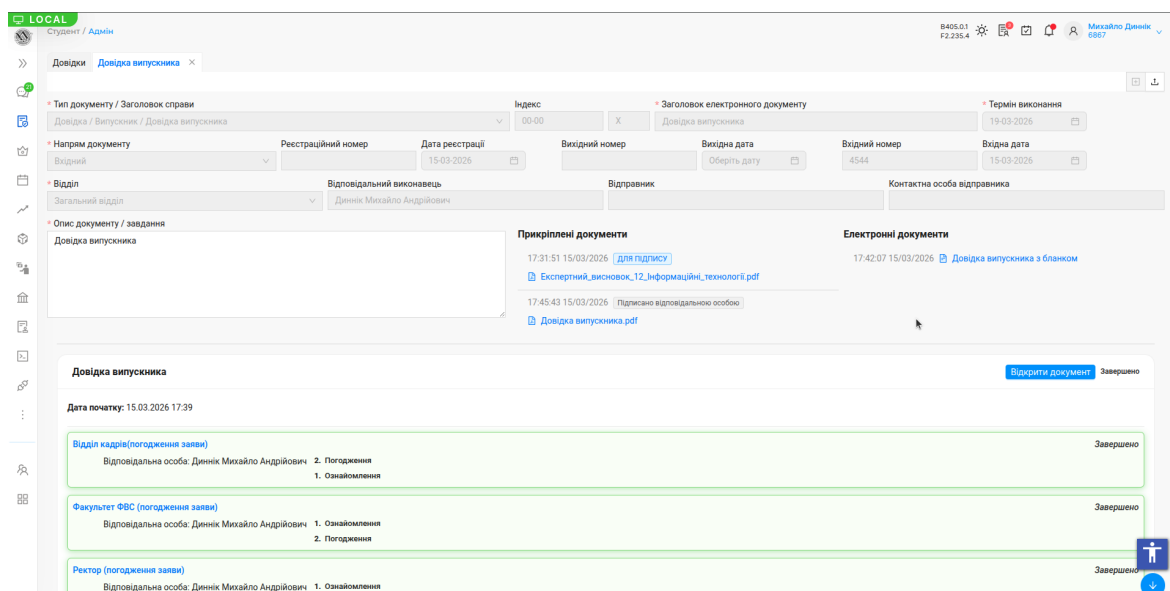


Рисунок А.15. Завершений документообіг з усіма підписами та документами

Наведений приклад демонструє основні можливості модуля електронного документообігу: візуальне створення маршрутів, автоматичну маршрутизацію з каскадною активацією процесів, генерацію документів за шаблонами з підстановкою даних, роботу з файлами та повний аудит дій.

Підписання документа кваліфікованим електронним підписом

Розглянемо процес підписання документа через сервіс «Дія» з використанням кваліфікованого електронного підпису (КЕП).

Крок 1. Завантаження документа. Відповідальна особа отримує сповіщення та відкриває документ. Степпер відображає поточний етап — «Завантаження документу» (рис. А.16). Система показує прев'ю документа та інформацію про файл для підпису.

Крок 2. Вибір способу підписання. Після завантаження степпер переходить на етап «Підписання за допомогою Дія.Підпису» (рис. А.17). Користувач обирає один із двох варіантів: «Підписати Дія.Підписом» або «Підписати за допомогою КЕП».

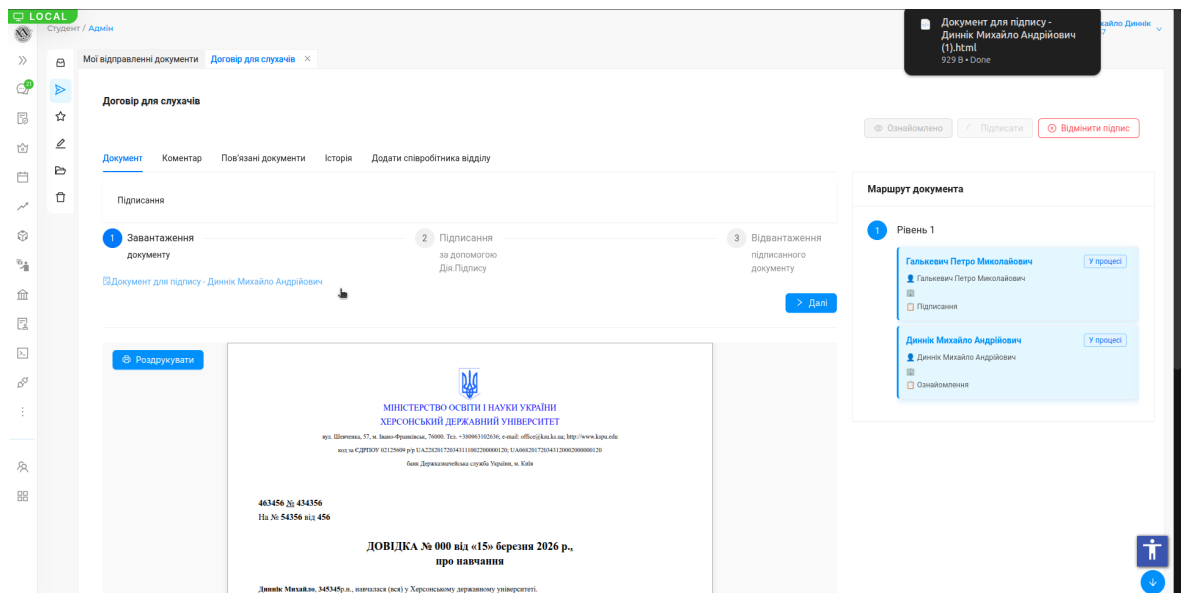


Рисунок А.16. Етап завантаження документа для підпису

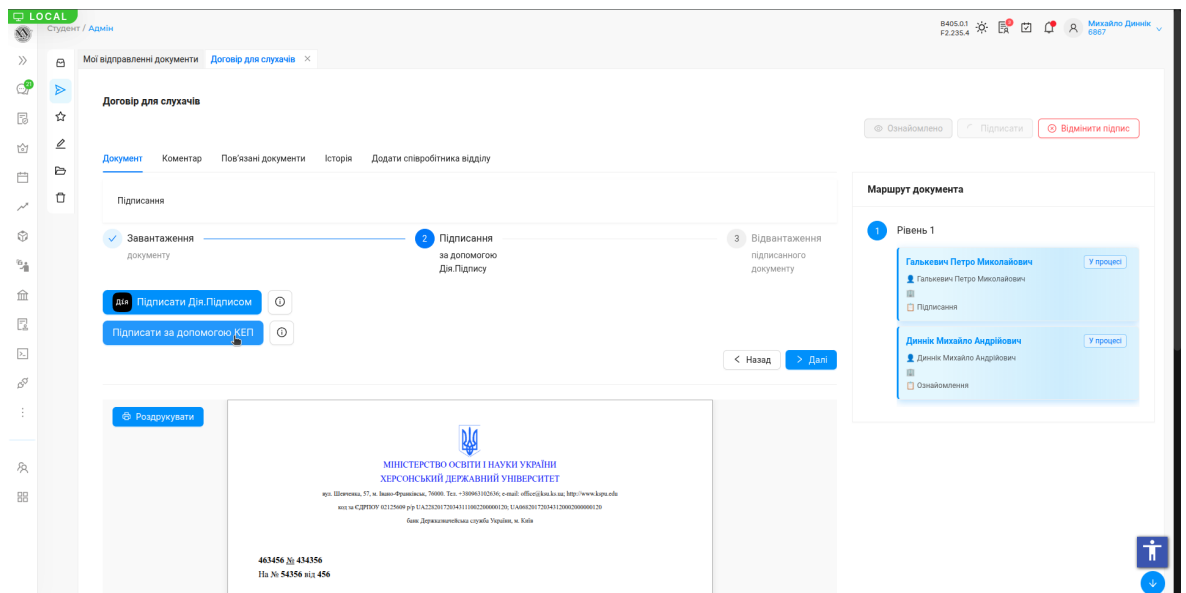


Рисунок А.17. Вибір способу підписання документа

Крок 3. Підписання через ID.GOV.UA. При виборі КЕП система перенаправляє на сторінку ID.GOV.UA (рис. А.18), де доступні три варіанти: електронний підпис (файловий ключ), Дія.Підпис-UA та Дія.Підпис-EU.

Крок 4. Результат підписання. Після успішного підписання система відображає підтвердження (рис. А.19): файл з підписом (.p7s), оригінальний файл та протокол створення і перевірки кваліфікованого електронного підпису з даними підписувача.

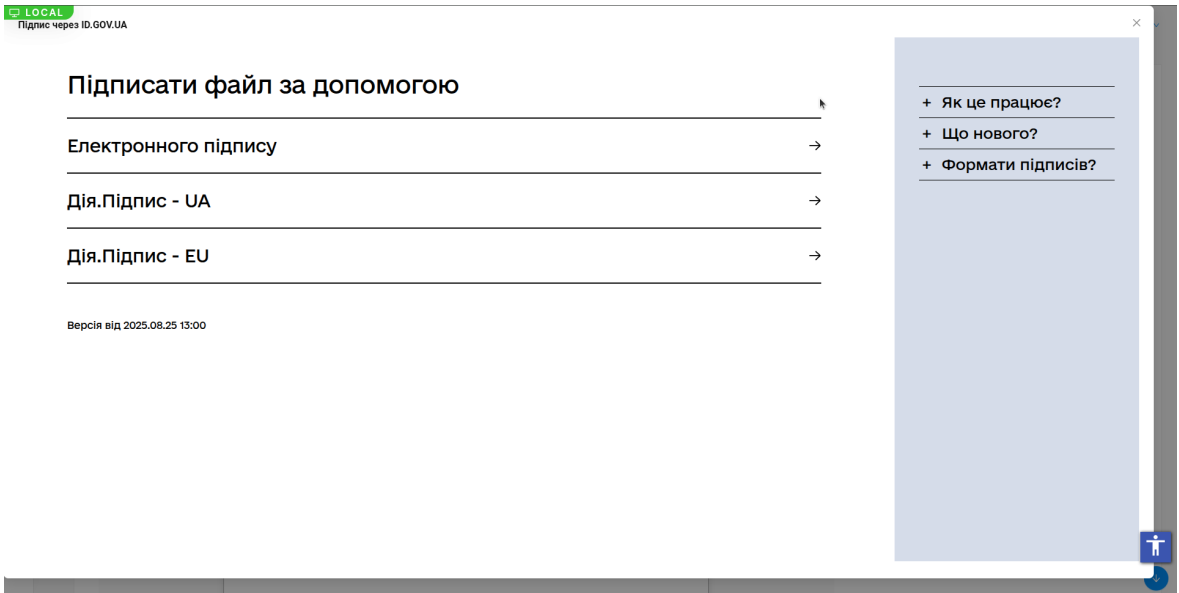


Рисунок А.18. Вибір типу підпису на сторінці ID.GOV.UA

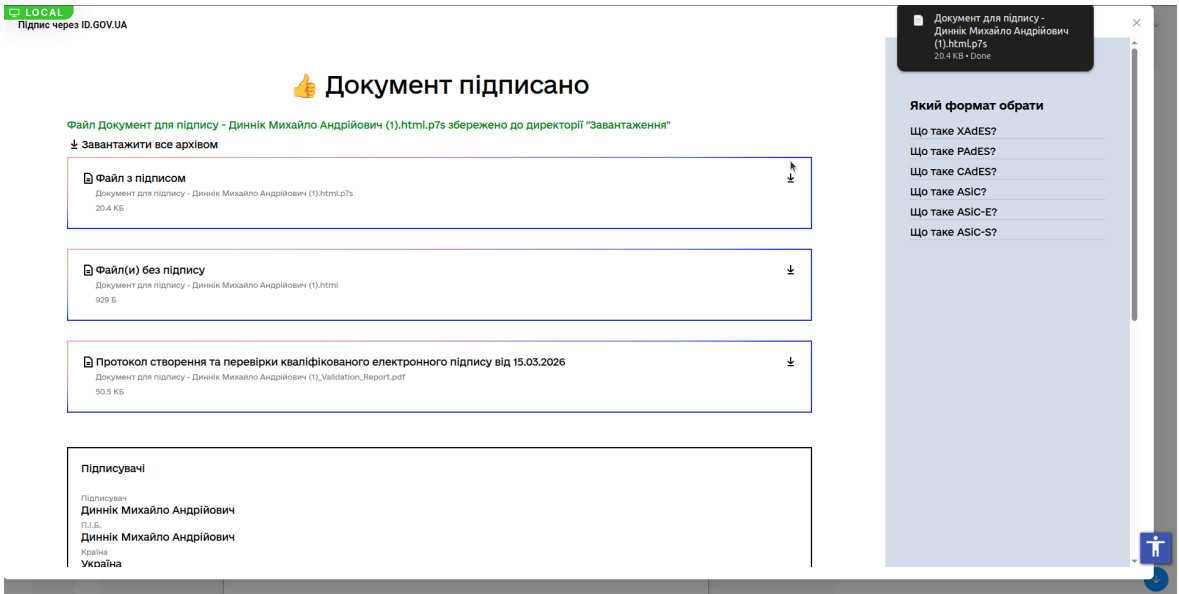


Рисунок А.19. Результат підписання: файли та протокол перевірки КЕП

Крок 5. Відвантаження. Степпер переходить на фінальний етап — «Відвантаження підписаного документу» (рис. А.20). Підписаний файл завантажується в систему.

Крок 6. Завершення. Документ успішно підписано (рис. А.21). Система оновлює статус процесу та відображає підтвердження. Крок підписання завершено, маршрут продовжує виконання.

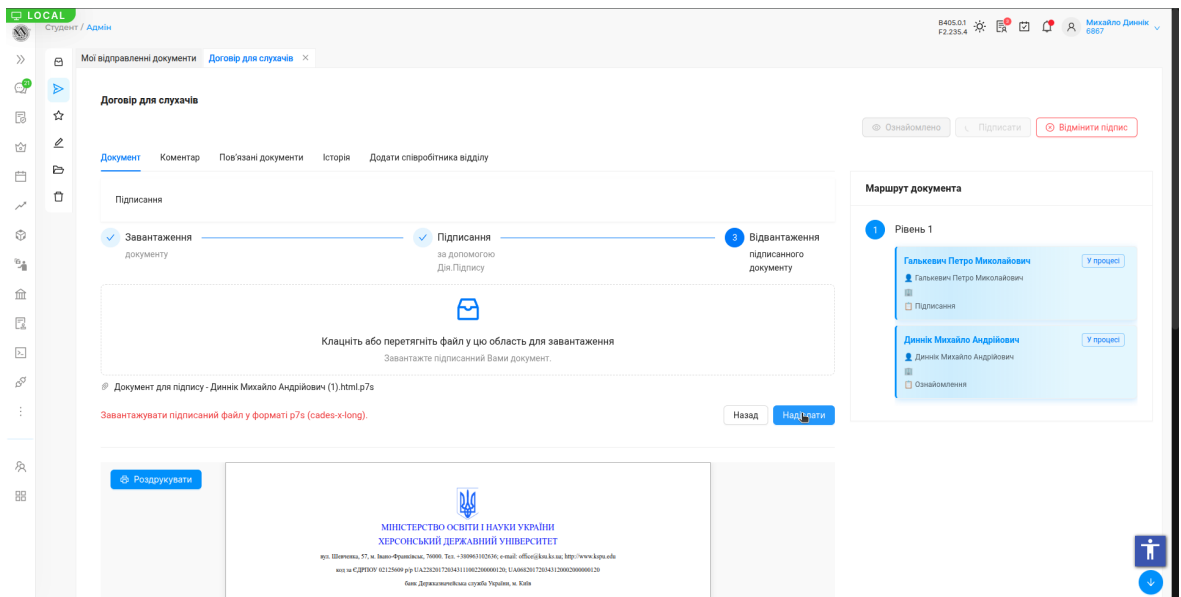


Рисунок А.20. Відвантаження підписаного документа

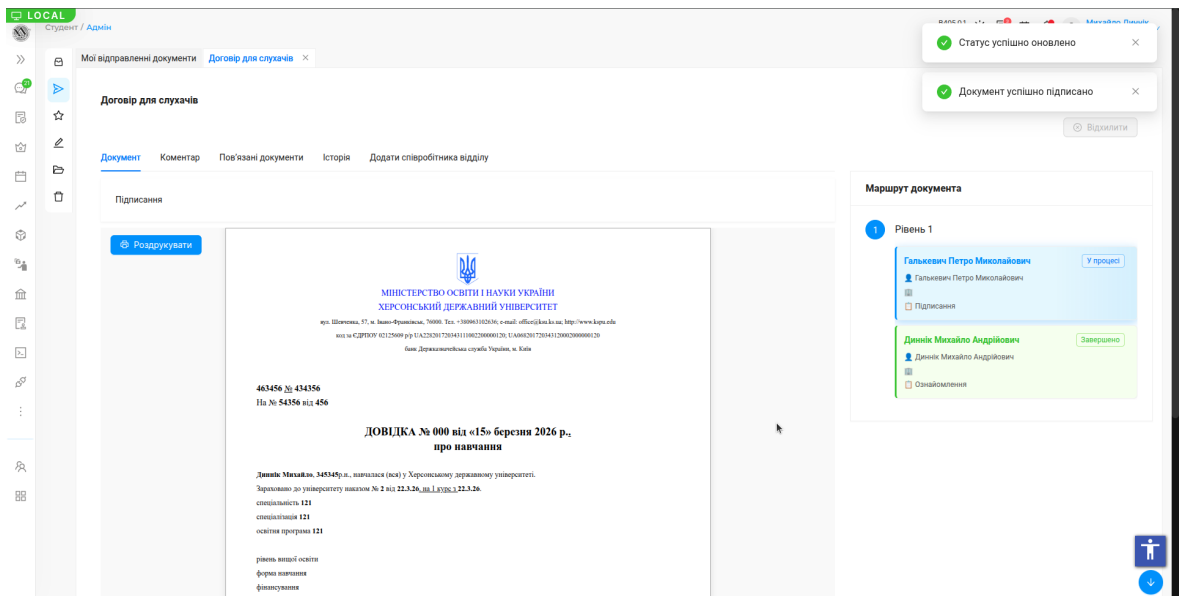


Рисунок А.21. Документ успішно підписано кваліфікованим електронним підписом