

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Навчально-науковий інститут інформаційних технологій та бізнесу
Кафедра інформаційних технологій та аналітики даних

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра

на тему: **«Розробка дизайн-системи та користувацького інтерфейсу веб-додатку адміністрування розкладу занять з урахуванням принципів WCAG та адаптацією під мобільні пристрої»**

Виконав: студент 4 курсу, групи КН-42
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Строзюк Роман Володимирович

Керівник: викладач кафедри інформаційних технологій та
аналітики даних *Мацевич Денис Володимирович*

Рецензент: кандидат технічних наук, доцент,
доцент кафедри прикладної математики
Донецького національного університету
імені Василя Стуса
Загоруйко Любов Василівна

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри інформаційних технологій та аналітики даних
_____ (проф., д.е.н. Кривицька О.Р.)

Протокол № ____ від « ____ » _____ 2026 р.

Острог, 2026

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: Розробка дизайн-системи та користувацького інтерфейсу веб-додатку адміністрування розкладу занять з урахуванням принципів WCAG та адаптацією під мобільні пристрої

Автор: студент спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки» Строзюк Роман Володимирович

Науковий керівник: Керівник: викладач кафедри економіко-математичного моделювання та інформаційних технологій Мацевич Денис Володимирович

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 52 (кількість сторінок роботи) с., 10 (кількість рисунків) рис., 0 (кількість таблиць) табл., 4 (кількість додатків) додатків, 35 (кількість джерел) джерел.

Ключові слова: ДИЗАЙН-СИСТЕМА, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ВЕБДОДАТОК, РОЗКЛАД ЗАНЯТЬ, ДОСТУПНІСТЬ, WCAG, МОБІЛЬНА АДАПТАЦІЯ, FIGMA, ПРОТОТИПУВАННЯ, UI/UX.

Короткий зміст праці: Кваліфікаційна робота присвячена проектуванню користувацького інтерфейсу та розробці комплексної дизайн-системи для вебдодатка адміністрування розкладу занять. Метою роботи є створення візуально консистентного, інтуїтивно зрозумілого та інклюзивного середовища для ефективного управління навчальним розкладом. У процесі виконання роботи досліджено наявні програмні рішення та виявлено їхні UX/UI недоліки. Розроблено інформаційну архітектуру та мапу шляху користувача (User Flow). У середовищі Figma створено масштабовану дизайн-систему (UI Kit) із стандартизованими компонентами. Особливу увагу приділено забезпеченню інклюзивності: інтерфейс спроектовано із суворим дотриманням міжнародних стандартів вебдоступності WCAG (забезпечення необхідного контрасту, читабельності та навігації). Реалізовано адаптацію макетів під мобільні пристрої відповідно до принципів Mobile First. Практичним результатом є інтерактивний високоточний прототип системи, повністю підготовлений до передачі у Frontend-розробку.

ABSTRACT
of the qualification
work for a Bachelor's degree

Theme: *Development of a design system and user interface for a timetable administration web application, considering WCAG principles and mobile adaptation*

Author: *Student of degree programme 122 Computer Science within the vocational education programme 'Computer Science' Roman Volodymyrovych Stroziuk*

Supervisor: *Lecturer of the Department of Economic and Mathematical Modeling and Information Technologies Denys Volodymyrovych Matsevych*

Defended on “.....”..... 20__.

Explanatory note to the qualification work: *52 (number of pages) pp., 41 (number of figures) figs., 10 (number of tables) tables, 4 (number of appendices) appendices, 35 (number of sources) sources.*

Keywords: *DESIGN SYSTEM, USER INTERFACE, WEB APPLICATION, TIMETABLE, ACCESSIBILITY, WCAG, MOBILE ADAPTATION, FIGMA, PROTOTYPING, UI/UX.*

Abstract: *This qualification work is devoted to the design of a user interface and the development of a comprehensive design system for a timetable administration web application. The aim of the work is to create a visually consistent, intuitive, and inclusive environment for effective management of the educational schedule. The research involves analyzing existing software solutions and identifying their UX/UI shortcomings. The information architecture and user flow have been developed. A scalable design system (UI Kit) with standardized components was created in Figma. Special attention was paid to ensuring inclusivity: the interface was designed in strict compliance with the international web accessibility standards WCAG (ensuring required contrast, readability, and navigation). Adaptation of layouts for mobile devices has been implemented in accordance with Mobile First principles. The practical result is a high-fidelity interactive prototype of the system, fully prepared for handoff to frontend development.*

ЗМІСТ

ЗМІСТ	4
ВСТУП	6
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ТА АНАЛІЗ ПРЕДМЕТНОГО СЕРЕДОВИЩА	9
1.1. Аналіз процесів адміністрування навчального розкладу та потреб користувачів	9
1.2. Огляд наявних програмних рішень та аналогів інтерфейсів управління розкладом	10
1.3. Обґрунтування вибору стандартів доступності WCAG та принципів Mobile First	12
1.4. Постановка задачі проектування	14
ВИСНОВКИ ДО РОЗДІЛУ	16
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ДИЗАЙН-СИСТЕМИ	18
2.1. Аналіз інформаційних об'єктів та структури даних інтерфейсу системи адміністрування	18
2.2. Проектування інформаційної архітектури та сценаріїв користувацької взаємодії	20
2.3. Математичне та алгоритмічне обґрунтування параметрів дизайн-системи (колірні контрасти, модульні сітки)	22
2.3.1. Алгоритмічне забезпечення інклюзивності (WCAG)	22
2.3.2. Математична модель просторової системи (Spatial System)	23
ВИСНОВКИ ДО РОЗДІЛУ	24
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ	26
3.1. Обґрунтування вибору інструментів розробки (Figma та екосистема дизайну)	26
3.2. Технічні вимоги до функціонування та середовища розгортання інтерфейсу	28
3.3. Розробка та опис компонентної бази дизайн-системи (UI Kit)	30
3.4. Прототипування та керівництво користувача з експлуатації вебдодатка	33
ВИСНОВКИ ДО РОЗДІЛУ	37
ЗАГАЛЬНІ ВИСНОВКИ	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41
ДОДАТКИ	44

ВСТУП

У сучасному цифровому суспільстві ефективна організація освітнього процесу безпосередньо залежить від якості та надійності інформаційних систем управління. В умовах стрімкої цифровізації закладів вищої освіти, зокрема Національного університету «Острозька академія», виникає гостра потреба у переході від застарілих монолітних систем до сучасних вебрішень. Одним із найбільш критичних та високонавантажених елементів управління є адміністрування розкладу занять — процес, що характеризується високим рівнем динамічності та великою кількістю взаємозалежних змінних.

Традиційні підходи до розробки освітнього ПЗ часто призводять до «інтерфейсної фрагментації», де відсутність єдиних інженерних стандартів дизайну спричиняє появу складних, перевантажених та неадаптивних інтерфейсів. Це не лише знижує продуктивність роботи адміністраторів, а й створює бар'єри для користувачів із порушеннями зору чи моторики через ігнорування стандартів інклюзивності. В сучасних умовах користувач очікує від академічних платформ рівня UX/UI, що відповідає кращим світовим комерційним продуктам.

Ефективне розв'язання цих проблем лежить у площині розробки та впровадження **масштабованих дизайн-систем**. Використання компонентно-орієнтованого підходу дозволяє забезпечити візуальну та функціональну консистентність, значно прискорити процес Frontend-розробки та закласти фундамент для легкої підтримки продукту. Особливої актуальності набуває інтеграція міжнародних стандартів доступності WCAG та реалізація стратегії Mobile First, що гарантує безперешкодний доступ до даних розкладу з будь-якого пристрою

Метою даної кваліфікаційної роботи є розробка комплексної дизайн-системи та проектування користувацького інтерфейсу вебдодатка адміністрування розкладу занять із суворим дотриманням принципів доступності WCAG та повною адаптацією під мобільні пристрої.

Для досягнення поставленої мети необхідно виконати такі основні завдання:

- провести глибокий аналіз предметної області та дослідити UX/UI недоліки існуючих рішень для управління розкладом;
- спроектувати інформаційну архітектуру додатка та розробити сценарії користувацької взаємодії (User Flow) для ключових акторів системи;
- розробити архітектуру дизайн-системи (UI Kit), що включає систему дизайн-токенів, бібліотеку високомодульних компонентів та правила їхньої взаємодії;
- створити високоточні макети інтерфейсу для десктопної та мобільної версій вебдодатка;
- імплементувати у процес проектування вимоги стандартів WCAG 2.1 (рівень AA), забезпечивши високий рівень інклюзивності інтерфейсу;
- реалізувати інтерактивний високоточний прототип системи у середовищі Figma для верифікації проєктних рішень та підготовки до етапу розробки.

Об'єкт дослідження: користувацькі інтерфейси освітніх вебдодатків та інженерні підходи до побудови систем візуального дизайну.

Предмет дослідження: методи та засоби проектування адаптивної дизайн-системи та інклюзивного інтерфейсу вебдодатка адміністрування розкладу занять.

Таким чином, реалізація даного проєкту дозволить перейти від стихійного створення екранних форм до системного проектування освітніх інтерфейсів. Це не

лише підвищить ефективність адміністрування навчального процесу, а й забезпечить створення сучасного, інклюзивного та технологічно досконалого інструменту, доступного для кожного учасника академічної спільноти.

РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ТА АНАЛІЗ ПРЕДМЕТНОГО СЕРЕДОВИЩА

1.1. Аналіз процесів адміністрування навчального розкладу та потреб користувачів

Процес формування навчального розкладу в закладах вищої освіти, зокрема в умовах цифровізації Національного університету «Острозька академія», є багатопараметричною задачею, що вимагає опрацювання значних обсягів слабкоструктурованих даних. Цей процес охоплює координацію навчальних планів, облік аудиторного фонду, моніторинг навантаження викладачів та дотримання санітарно-гігієнічних норм щодо розкладу студентів.

З погляду інженерії інтерфейсів (UI/UX), ключовою проблемою існуючих систем є «інтерфейсна фрагментація» — відсутність єдиної дизайн-системи. Застарілі рішення оперують розрізненими табличними структурами, які не мають спільної логіки взаємодії. Це не лише ускладнює когнітивне сприйняття, а й унеможливорює швидку адаптацію інтерфейсу під різні типи пристроїв чи потреби інклюзивності. Таким чином, виникає потреба не просто у створенні нових панелей адміністрування, а у розробці цілісного візуального та функціонального фреймворку (дизайн-системи), який забезпечить консистентність даних.

Аналіз предметного середовища дозволяє виокремити три ключові групи акторів, чії потреби формують архітектуру майбутньої дизайн-системи:

1. **Адміністратори (диспетчери розкладу).** * *Проектний фокус:* Потребують складних високомодульних компонентів (сітки розкладу, drag-and-drop контейнери, багаторівневі фільтри).
 - *Вимоги до системи:* Мінімізація часу на масові операції та візуальна валідація конфліктів у реальному часі.
2. **Викладачі.** * *Проектний фокус:* Потребують персоналізованих інформаційних блоків (картки занять, дашборди).

- *Вимоги до системи:* Висока читабельність на мобільних пристроях та швидкий доступ до змін у розкладі через систему сповіщень.
3. **Студенти.** * *Проектний фокус:* Потребують максимально спрощених навігаційних патернів.
- *Вимоги до системи:* Пріоритет на Mobile First архітектуру, інтуїтивне перемикання між циклами навчання (чисельник/знаменник) та робота інтерфейсу в умовах обмеженого екранного простору.

Особливе місце в розробці дизайн-системи посідає дотримання міжнародних стандартів доступності WCAG. Це вимагає закладення на рівні дизайн-токенів (базових параметрів системи) правил щодо контрастності кольорів (коефіцієнт не менше 4.5:1), масштабованості шрифтів без руйнування макета та створення достатніх сенсорних зон для мобільної взаємодії. Такий підхід перетворює інтерфейс із простого інструменту перегляду на інклюзивне середовище, доступне для всіх учасників освітнього процесу.

Функціональну модель взаємодії акторів із компонентами системи та межі їхнього доступу формалізовано за допомогою UML-діаграми використання (Use Case), яку наведено у Додатку А (рис. А.1).»

Логіку найбільш трудомісткого процесу — формування та публікації розкладу з перевіркою цілісності даних — відображено на UML-діаграмі діяльності (Activity diagram) у Додатку А (рис. А.2).

Отже, розробка сучасної системи управління розкладом вимагає переходу від проектування окремих екранів до створення масштабованої дизайн-системи. Це дозволить замінити громіздкі таблиці адаптивними та доступними компонентами, що забезпечують стабільний користувацький досвід на будь-яких пристроях.

1.2. Огляд наявних програмних рішень та аналогів інтерфейсів управління розкладом

Аналіз сучасного ринку програмного забезпечення для управління освітніми процесами свідчить про глибокий розрив між функціональною складністю алгоритмів та якістю їх візуальної реалізації. Більшість існуючих рішень розроблялися як монолітні системи, де інтерфейс є вторинним продуктом щодо бізнес-логіки. З погляду комп'ютерних наук, головною проблемою таких систем є «інтерфейсний хаос», спричинений відсутністю єдиної дизайн-системи. Для аналізу ці аналоги доцільно класифікувати за рівнем системності їхнього дизайну.

1. Спеціалізовані десктопні комплекси (наприклад, aSc Timetables) Дана категорія систем володіє найпотужнішим математичним апаратом для генерації розкладів, проте демонструє критичну слабкість в архітектурі інтерфейсу.

- **Архітектурний недолік:** Інтерфейс базується на жорстко зафіксованих формах (Legacy UI), що не підтримують компонентний підхід. Це унеможливорює масштабування системи на інші платформи (web, mobile) без повної переробки логіки відображення.
- **UI/UX обмеження:** Веб-модулі таких систем мають низьку консистентність: елементи на десктопі та в браузері сприймаються як різні продукти, що підвищує когнітивне навантаження та порушує цілісність користувацького досвіду.

2. Екосистемні корпоративні рішення (Google Calendar, Microsoft Teams) Ці продукти є прикладом успішного впровадження глобальних дизайн-систем (Material Design та Fluent Design відповідно).

- **Переваги системного підходу:** Завдяки використанню стандартизованих дизайн-токенів ці рішення ідеально адаптовані під мобільні пристрої та повністю відповідають вимогам інклюзивності WCAG.

- **Проблема доменної адаптації:** Попри досконалу дизайн-систему, ці продукти не мають специфічних компонентів для освітньої галузі. У їхній бібліотеці елементів відсутні патерни для відображення специфічних академічних циклів (чисельник/знаменник) або потокових занять. Це робить їх зручними для персонального планування, але непридатними для комплексного адміністрування розкладу ЗВО.

3. Локальні університетські портали (веб-модулі АСУ) Українські академічні системи (наприклад, АСУ «Деканат») найчастіше демонструють повну відсутність системного дизайну. Аналіз виявляє наступні критичні патерни:

- **Інтерфейсна фрагментація:** Через відсутність єдиної UI-бібліотеки сторінки порталу створюються стихійно. Різні радіуси заокруглень, неоднакові колірні коди для схожих дій та розбіжності у типографіці роблять інтерфейс візуально «шумним» та непередбачуваним.
- **Деградація мобільного досвіду:** Відсутність принципів Mobile First у розробці призводить до того, що складні табличні структури стають повністю нечитабельними на малих екранах.
- **Порушення інклюзивності:** Оскільки розробка ведеться без використання дизайн-токенів кольору та контрасту, такі системи практично ніколи не проходять автоматизовану перевірку доступності (наприклад, через Google Lighthouse), залишаючись закритими для користувачів з вадами зору.

Висновки з огляду аналогів: Проведений аналіз підтверджує, що головною причиною низької якості освітніх інтерфейсів є відсутність **спеціалізованої дизайн-системи**, яка б поєднувала потужну академічну логіку з сучасними інженерними стандартами UI/UX. Жодне з існуючих рішень не пропонує масштабованої бібліотеки компонентів, яка б одночасно враховувала специфіку українських ЗВО, принципи Mobile First та сувору відповідність стандартам доступності WCAG. Це обґрунтовує необхідність розробки власного системного

рішення, яке стане фундаментом для створення консистентного та інклюзивного інтерфейсу адміністрування розкладу.

1.3. Обґрунтування вибору стандартів доступності WCAG та принципів Mobile First

Проектування сучасних вебрішень для освітнього середовища потребує врахування глобальних тенденцій у сфері вебтехнологій, серед яких ключовими є інклюзивність та адаптивність. Для розробки інтерфейсу системи адміністрування розкладу було обрано міжнародний стандарт WCAG (Web Content Accessibility Guidelines) та методологію проектування Mobile First.

Стандарти WCAG 2.1 (та їх оновлення) є основоположними для створення доступного вебконтенту. Вибір на користь цих стандартів обґрунтований наступними факторами:

- 1. Соціальна відповідальність та інклюзивність.** Університетська спільнота є різномірною за своїм складом і включає користувачів із постійними або тимчасовими порушеннями здоров'я (зору, слуху, моторики). Дотримання WCAG (зокрема рівня AA) гарантує, що розклад занять буде доступним для студентів зі зниженим зором (через підтримку високого контрасту та масштабування) та користувачів, які використовують допоміжні технології, такі як зчитувачі екрана (screen readers).
- 2. Покращення загального UX.** Більшість вимог доступності (чітка навігація, зрозуміла візуальна ієрархія, передбачуваність інтерфейсу) позитивно впливають на досвід усіх категорій користувачів, зменшуючи когнітивне навантаження та пришвидшуючи пошук необхідної інформації.
- 3. Відповідність державним та міжнародним вимогам.** В Україні цифрова доступність стає обов'язковою умовою для державних та освітніх інформаційних ресурсів (згідно з ДСТУ EN 301 549), що робить використання WCAG необхідним етапом професійної розробки.

Паралельно з доступністю, критично важливим аспектом є стратегія Mobile First. Цей підхід передбачає розробку інтерфейсу, починаючи з найменшого екрана (смартфона) з поступовим розширенням функціональності для десктопних версій. Обґрунтування даного вибору базується на таких положеннях:

- **Пріоритетність мобільного трафіку.** Статистичний аналіз використання освітніх порталів свідчить, що понад 70% звернень до розкладу здійснюються зі смартфонів «на ходу» — під час перерв, у транспорті чи безпосередньо перед початком занять.
- **Фокус на контенті.** Обмежений простір мобільного екрана змушує дизайнера відмовитися від другорядних елементів і зосередитися на головному — часі, місці та назві дисципліни. Це дозволяє уникнути перевантаження інтерфейсу адміністрування, яке є типовою проблемою аналогів.
- **Технологічна готовність (Progressive Enhancement).** Проектування від меншого до більшого дозволяє створити легкий та швидкий інтерфейс, який буде коректно відображатися на слабких пристроях або при нестабільному інтернет-з'єднанні, що є критичним для мобільного використання в умовах кампусу.

Поєднання стандартів WCAG та принципів Mobile First дозволяє створити універсальну дизайн-систему. Такий підхід забезпечує не лише візуальну привабливість, а й високу технічну якість продукту, роблячи систему адміністрування розкладу зручним та безбар'єрним інструментом для кожного учасника навчального процесу.

1.4. Постановка задачі проектування

На основі проведеного комплексного аналізу предметної області та виявлених критичних недоліків у наявних програмних рішеннях, формулюється основна мета даної кваліфікаційної роботи: **розробити масштабовану дизайн-систему та адаптивний користувацький інтерфейс** вебдодатка для адміністрування

навчального розкладу. Проект спрямований на створення інклюзивного цифрового середовища, що базується на принципах системного дизайну, суворій відповідності стандартам WCAG та методології Mobile First.

Для реалізації поставленої мети необхідно вирішити наступний перелік інженерно-проектних завдань:

- **Формалізація інформаційної архітектури:** структурувати логіку вебдодатка та змоделювати карти користувацьких шляхів (User Flow) для ключових сценаріїв адміністрування. Це дозволить забезпечити передбачувану взаємодію користувача з високонавантаженими даними розкладу.
- **Розробка цілісної дизайн-системи (UI Kit):** спроектувати бібліотеку стандартизованих компонентів (атомів, молекул та організмів) у середовищі Figma. Використання інструментарію Variants, Auto Layout та системи дизайн-токенів повинно гарантувати технічну гнучкість та візуальну консистентність усіх модулів системи.
- **Інтеграція стандартів інклюзивності:** імплементувати вимоги міжнародного стандарту WCAG 2.1 (рівень AA) на рівні архітектури компонентів. Це включає налаштування колірних контрастів, правил типографіки та навігаційних патернів для безбар'єрного доступу користувачів із порушеннями зору.
- **Реалізація адаптивності за принципом Mobile First:** розробити систему макетів, де пріоритет надається мобільному інтерфейсу як найбільш затребуваному каналу доступу. Забезпечити автоматичну адаптацію компонентів для планшетних та десктопних версій без втрати функціональності.
- **Побудова інтерактивного високоточного прототипу:** створити у Figma клікабельну модель додатка (High-Fidelity Prototype) для верифікації проектних рішень, тестування навігаційної логіки та підготовки технічної документації для етапу розробки (Handoff).

Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено ґрунтовний аналіз предметного середовища та специфіки адміністрування навчального розкладу в умовах сучасного університету. За допомогою інструментарію UML-моделювання (діаграм використання та діяльності) було формалізовано рольові моделі акторів та їхні функціональні потреби, що дозволило визначити межі проектування майбутньої системи.

Критичний огляд існуючих програмних продуктів та аналогів інтерфейсів засвідчив наявність системної проблеми — «інтерфейсної фрагментації». Більшість рішень ігнорують принципи системного дизайну, що призводить до створення перевантажених, неадаптивних та недоступних для інклюзивного використання продуктів. Встановлено, що відсутність єдиної дизайн-системи є головним бар'єром для масштабування та якісного оновлення освітнього ПЗ.

Для розв'язання виявлених проблем обґрунтовано необхідність застосування методології Mobile First та стандартів вебдоступності WCAG. Такий підхід дозволить змістити акцент із простого малювання екранних форм на розробку гнучкого інженерного фреймворку, здатного стабільно функціонувати на будь-яких типах пристроїв та бути доступним для кожного учасника навчального процесу.

Результатом аналітичного етапу стала чітка постановка задачі проектування. Сформовані вимоги до інформаційної архітектури, компонентної бази та інклюзивних параметрів системи створюють необхідний базис для практичної розробки дизайн-системи та побудови високоточних прототипів у наступних розділах роботи.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ДИЗАЙН-СИСТЕМИ

2.1. Аналіз інформаційних об'єктів та структури даних інтерфейсу системи адміністрування

Проектування якісного користувацького інтерфейсу та розробка дизайн-системи неможливі без глибокого розуміння структури даних (контенту), якими оперує система. У контексті UI/UX проектування за методологією Content-First, інформаційна модель розглядається як набір візуальних об'єктів та їхніх станів, що формують архітектуру майбутнього додатка.

Аналіз предметного середовища дозволяє виділити дві ключові категорії інформаційних об'єктів дизайн-системи: об'єкти контенту (дані, що відображаються) та візуальні об'єкти (елементи інтерфейсу).

1. Об'єкти контенту (дані системи розкладу) Це сутності предметної області, які безпосередньо диктують вимоги до структури UI-компонентів:

- **Картка заняття (Слот розкладу).** Найбільш інформаційно навантажений об'єкт. Він агрегує в собі назву дисципліни, ПІБ викладачів, номер аудиторії та час. З точки зору UI, цей об'єкт вимагає гнучкого контейнера (Auto Layout), який здатний масштабуватися залежно від обсягу тексту (наприклад, коли у викладача довге прізвище і посада), не руйнуючи загальну модульну сітку екрана.
- **Довідники (Аудиторії, Групи, Викладачі).** Ці масиви даних в інтерфейсі адміністрування трансформуються в об'єкти фільтрації, пошуку та вибору. Це формує вимогу до розробки складних інтерактивних компонентів: випадаючих списків (Dropdowns) з підтримкою скролінгу та полів автодоповнення (Autocomplete).

2. Базові об'єкти дизайн-системи (Дизайн-токени та компоненти) Щоб контентні сутності відображалися консистентно на різних екранах, структура даних самої дизайн-системи формується за принципами Atomic Design:

- **Дизайн-токени (Атоми).** Базові інформаційні константи інтерфейсу, що зберігаються у вигляді стилів Figma. До них належать: колірна палітра з розрахованими рівнями контрасту за WCAG, шкала типографіки (Heading, Body, Caption) та математична модель відступів (Spacing) на базі 8-піксельної сітки.
- **UI-компоненти (Молекули та Організми).** Об'єкти інтерфейсу, зібрані з токенів. Важливою характеристикою структури цих об'єктів є їхні стани (States). Наприклад, об'єкт «Кнопка збереження розкладу» в дизайн-системі містить дані про свої стани: Default (звичайний), Hover (наведення), Disabled (неактивний при помилці валідації) та Loading (завантаження).

Структура даних інтерфейсу також накладає **обмеження на користувацьке введення**, які мають візуалізуватися дизайн-системою:

- **Обробка помилок (Error states):** у разі конфлікту в розкладі (наприклад, аудиторія зайнята), система має змінити стан відповідного UI-компонента (червоне обведення, поява допоміжного тексту).
- **Адаптивні обмеження:** зміна розміру або приховування другорядних об'єктів контенту на екранах мобільних пристроїв (Responsive constraints).

Таким чином, чітка формалізація інформаційних об'єктів (як з боку контенту розкладу, так і з боку дизайн-токенів) дозволяє на етапі прототипування створити логічну, масштабовану та передбачувану систему компонентів, яка легко адаптується під різні пристрої та відповідає вимогам доступності.

2.2. Проектування інформаційної архітектури та сценаріїв користувацької взаємодії

Проектування інформаційної архітектури (ІА) для сучасних вебдодатків виходить за межі простого створення мапи екранів. У контексті розробки дизайн-системи ІА визначає ієрархію, принципи успадкування та логіку взаємодії UI-компонентів. Для системи адміністрування розкладу було застосовано компонентно-орієнтований підхід на базі методології Atomic Design.

Логічна структура дизайн-системи побудована за багаторівневим принципом:

1. **Атоми (Дизайн-токени та базові елементи):** Найнижчий рівень абстракції, що не підлягає подальшому поділу. Включає кольори, типографіку, іконки та сітки.
2. **Молекули:** Прості UI-компоненти, створені з атомів (наприклад, кнопка з іконкою або поле введення з міткою).
3. **Організми:** Складні інтерактивні блоки, що інкапсулюють власну бізнес-логіку відображення (наприклад, картка заняття, навігаційна панель або календарний фільтр).

Для формалізації структури компонентної бази та виявлення зв'язків між елементами інтерфейсу розроблено концептуальну модель у формі UML-діаграми класів (рис. 2.1). Базовим елементом архітектури є абстрактний клас **UI_Component**, який визначає загальні властивості: розміри (ширина, висота) та стан видимості. Ці базові параметри успадковуються конкретними функціональними компонентами, такими як **Button** (Кнопка), **InputField** (Поле введення) та **Card** (Картка). Кожен клас-наслідник містить власні унікальні атрибути (наприклад, стан наведення чи наявність іконки), що в середовищі розробки Figma реалізуються через механізм властивостей та варіантів (Component Properties, Variants).

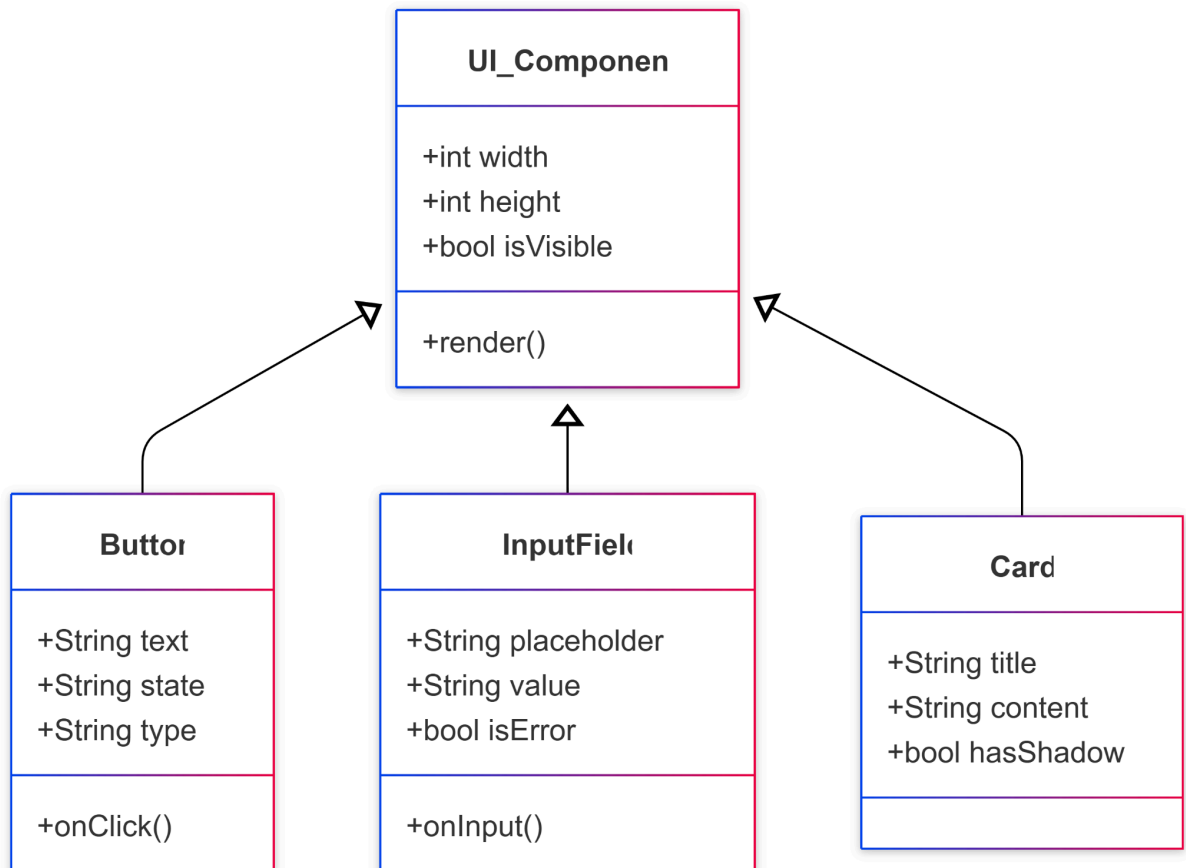


Рис. 2.1. Концептуальна модель ієрархії компонентів дизайн-системи (UML-діаграма класів)

Наступним етапом стало проєктування сценаріїв користувацької взаємодії (User Flows), орієнтоване на забезпечення безшовної та інтуїтивно зрозумілої роботи в кросплатформному середовищі. Для деталізації логіки поведінки системи було розроблено карти шляхів користувача для ключових мікроінтеракцій:

- **Сценарій валідації даних:** При створенні нового заняття компоненти форми автоматично змінюють свій стан (State) з **Default** на **Error** у разі виявлення конфлікту (наприклад, накладання розкладу). Це супроводжується візуальними тригерами, що не покладаються виключно на колір, що є вимогою WCAG.
- **Сценарій мобільної адаптації:** Навігаційні патерни змінюють свою архітектуру залежно від ширини екрана (Viewport). Десктопний сайдбар

(Sidebar) трансформується у мобільне нижнє меню (Bottom Navigation Bar), гарантуючи збереження доступу до ключових функцій адміністрування без горизонтального скролінгу.

2.3. Математичне та алгоритмічне обґрунтування параметрів дизайн-системи (колірні контрасти, модульні сітки)

Візуальна консистентність та інклюзивність інтерфейсу забезпечуються застосуванням математичних та алгоритмічних методів на етапі розробки дизайн-токенів.

2.3.1. Алгоритмічне забезпечення інклюзивності (WCAG)

Для досягнення рівня доступності AA за стандартом WCAG 2.1, колірна палітра дизайн-системи розраховувалася з використанням математичної моделі відносної яскравості (Relative Luminance). Алгоритм перевірки базується на обчисленні коефіцієнта контрастності (CR) між текстом та фоном.

Відносна яскравість простору sRGB обчислюється за формулою: $L = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$

Де R , G , B — лінеаризовані значення колірних каналів. Коефіцієнт контрастності розраховується як: $CR = (L_1 + 0.05) / (L_2 + 0.05)$ (де L_1 — яскравість світлішого кольору, L_2 — темнішого).

У розробленій дизайн-системі усі текстові компоненти (Heading, Body) проходять автоматичну алгоритмічну перевірку, що гарантує показник $CR \geq 4.5:1$ для стандартного тексту та $CR \geq 3.0:1$ для великих заголовків і графічних об'єктів (іконок).

2.3.2. Математична модель просторової системи (Spatial System)

Для забезпечення адаптивності та гармонійного масштабування інтерфейсу (Mobile First) була застосована математична модель 8-піксельної модульної сітки (8pt Grid System).

Суть алгоритму полягає в тому, що всі просторові змінні (зовнішні відступи — margins, внутрішні відступи — paddings, радіуси заокруглення, розміри іконок) є кратними базовому значенню 8. Математично це описується прогресією:

$$S_n = 8 \times n, \text{ де } n \in \mathbb{Z}^+ \text{ (} n \text{ — додатне ціле число).}$$

Використання базового множника 8 обґрунтовано архітектурою сучасних дисплеїв: більшість розмірів екранів є кратними 8. Це гарантує, що при масштабуванні або рендерингу в браузері об'єкти не потраплятимуть на субпікселі, що дозволяє уникнути розмиття елементів (anti-aliasing issues) та забезпечує ідеальну чіткість інтерфейсу адміністрування розкладу як на мобільних пристроях, так і на великих моніторах (Retina/4K).

Використання алгоритмічного підходу при створенні токенів кольору та простору перетворює дизайн-систему з набору графічних макетів на строгу інженерну структуру, оптимізовану для програмної реалізації.

Висновки до розділу 2

У другому розділі було проведено комплексне проектування інформаційної та математичної основи дизайн-системи вебдодатка адміністрування розкладу. Результати проведених досліджень та розробок дозволяють зробити наступні висновки:

1. На основі аналізу предметної області було формалізовано структуру інформаційних об'єктів інтерфейсу. Встановлено, що для забезпечення консистентності даних необхідно оперувати не лише контентними сутностями (картки занять, групи, викладачі), а й системними дизайн-токенами. Це дозволило закласти фундамент для створення гнучких UI-компонентів, що здатні коректно обробляти динамічні дані та різні стани системи.
2. Спроектовано багаторівневу архітектуру дизайн-системи за методологією Atomic Design. Використання компонентно-орієнтованого підходу дозволило структурувати інтерфейс від базових «атомів» до складних «організмів». Розроблена концептуальна модель у формі UML-діаграми класів підтвердила інженерну цілісність системи, де кожен UI-компонент має чітко визначені властивості, методи успадкування та стани, що спрощує подальшу програмну реалізацію.
3. Побудовано сценарії користувацької взаємодії (User Flows), які враховують особливості як десктопних, так і мобільних платформ. Особливу увагу приділено алгоритмам валідації вхідних даних та візуальній сигналізації про помилки, що є критичним для систем адміністрування з високою щільністю інформації.
4. Наведено математичне та алгоритмічне обґрунтування ключових параметрів системи. Використання моделі відносної яскравості для перевірки колірних контрастів гарантує відповідність інтерфейсу стандарту WCAG 2.1 на рівні AA. Впровадження математичної моделі 8-піксельної модульної сітки

забезпечує ідеальну чіткість та адаптивність елементів на екранах із різною щільністю пікселів.

Таким чином, розроблена теоретична та архітектурна база перетворює дизайн-систему на суворий інженерний стандарт. Це створює всі необхідні умови для переходу до третього розділу — безпосередньої практичної реалізації інтерфейсу в середовищі Figma та його тестування.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ

3.1. Обґрунтування вибору інструментарію розробки (Figma та екосистема дизайну)

Реалізація вимог, поставлених під час проєктування інформаційної та математичної архітектури дизайн-системи, вимагає використання сучасного програмного забезпечення, що підтримує компонентно-орієнтований підхід. На сучасному ринку інструментів для UI/UX дизайну основними конкурентами є Sketch, Adobe XD та Figma. Проте аналіз їхніх технічних характеристик показує, що Sketch має суворе обмеження (працює виключно на macOS), а Adobe XD зупинив активний розвиток нових функцій. Тому основним інструментальним середовищем для виконання даної кваліфікаційної роботи було обрано платформу Figma.

Вибір Figma як головного інструменту розробки обґрунтовується такими технічними та архітектурними перевагами:

1. **Хмарна інфраструктура (Cloud-based architecture).** Figma працює у браузері на базі WebGL, що забезпечує абсолютну кросплатформенність. Це усуває проблему несумісності операційних систем і дозволяє проводити тестування прототипів у режимі реального часу безпосередньо на цільових мобільних пристроях за допомогою додатку Figma Mirror.
2. **Алгоритмічне управління макетами (Auto Layout).** Ця функція є ключовою для реалізації адаптивного дизайну (Responsive Design) та імплементації математичної 8-піксельної модульної сітки. Auto Layout дозволяє створювати динамічні контейнери (фрейми), які автоматично змінюють свої розміри залежно від обсягу контенту (наприклад, довжини назви дисципліни чи ПІБ викладача), зберігаючи при цьому математично точні відступи згідно з розробленими правилами.

3. **Об'єктно-орієнтований підхід до компонентів (Variants та Component Properties).** Figma дозволяє будувати UI-елементи за принципами об'єктно-орієнтованого програмування. Базовий компонент (Master Component) виступає в ролі абстрактного класу, а його варіанти (Variants) — як екземпляри об'єкта з різними станами (Hover, Active, Error, Disabled). Застосування булевих властивостей (Boolean properties) та текстових змінних дозволяє значно зменшити кількість дубльованих макетів та оптимізувати вагу файлу дизайн-системи.
4. **Управління дизайн-токенами (Variables/Styles).** Інструментарій локальних змінних дозволяє закласти на рівні проєкту глобальні константи кольорів, радіусів та відступів. Це критично важливо для забезпечення вимог WCAG: будь-яка зміна контрастності кольору для покращення доступності застосовується глобально до всіх пов'язаних компонентів системи в один клік.

Окремим вагомим фактором є відкрита екосистема плагінів Figma та інтегрований режим розробника (Dev Mode). Завдяки Dev Mode спроектовані компоненти автоматично транслюються у фрагменти коду (CSS, Swift, Jetpack Compose), що максимально наближає процес проєктування до реальної Frontend-розробки (Handoff). Для алгоритмічної перевірки інклюзивності використовуються спеціалізовані плагіни (наприклад, Stark або Able), які в реальному часі аналізують коефіцієнт контрастності тексту щодо фону та симулюють різні типи кольорової сліпоти користувачів.

Таким чином, Figma надає вичерпний інженерний інструментарій, необхідний для побудови, тестування та підготовки до розгортання масштабованої дизайн-системи вебдодатка адміністрування розкладу.

3.2. Технічні вимоги до функціонування та середовища розгортання інтерфейсу

Оскільки розроблена дизайн-система є фундаментом для подальшої програмної реалізації вебдодатка (етап Frontend-розробки), критично важливим є формування чітких технічних вимог до середовища розгортання та метрик продуктивності. Проектування інтерфейсу здійснювалося з урахуванням того, що кінцевий продукт функціонуватиме в гетерогенному середовищі (різні пристрої, операційні системи та якості мережевого з'єднання).

Технічні вимоги до середовища функціонування поділяються на чотири основні категорії:

1. Вимоги до апаратного та програмного забезпечення клієнтської частини Вебдодаток повинен коректно відображатися та функціонувати за принципами кросбраузерності.

- **Браузерні рушії:** підтримка сучасних версій браузерів на базі Chromium (Google Chrome, Microsoft Edge), WebKit (Apple Safari) та Gecko (Mozilla Firefox).
- **Роздільна здатність екранів (Breakpoints):** відповідно до стратегії Mobile First, дизайн-система містить адаптивні правила (Media Queries) для трьох базових контрольних точок:
 - *Mobile:* від 320px (забезпечення роботи на пристроях базового рівня, наприклад, iPhone SE).
 - *Tablet:* від 768px (портретна та альбомна орієнтація планшетів).
 - *Desktop:* від 1024px до 1920px (максимальне використання ширини екрана для розгорнутої сітки розкладу).

2. Вимоги до продуктивності інтерфейсу (Core Web Vitals) Оскільки архітектура UI-компонентів безпосередньо впливає на швидкість рендерингу, до

програмної реалізації макетів висуваються вимоги згідно з метриками Google Core Web Vitals:

- **LCP (Largest Contentful Paint):** час відтворення основного контенту (наприклад, таблиці розкладу на поточний день) не повинен перевищувати 2.5 секунди. Для цього в дизайн-системі передбачено компоненти-заглушки (Skeleton Loaders), які відображаються до моменту отримання даних від бекенду.
- **CLS (Cumulative Layout Shift):** показник візуальної стабільності має бути меншим за 0.1. Компоненти дизайн-системи спроектовані з жорстко заданими мінімальними висотами (min-height), що запобігає «стрибкам» інтерфейсу під час динамічного підвантаження інформації про викладачів чи аудиторії.

3. Вимоги до апаратних засобів інклюзивності (Accessibility Tools) Для повноцінної відповідності стандарту WCAG 2.1 (рівень AA), розгорнутий інтерфейс повинен підтримувати взаємодію з допоміжними технологіями:

- **Скрінрідери (Screen Readers):** програмна розмітка повинна дозволяти коректне зчитування розкладу програмами VoiceOver (iOS/macOS), TalkBack (Android) та NVDA (Windows). Для цього в макетах передбачено логіку використання семантичних тегів та атрибутів `aria-label`.
- **Клавіатурна навігація:** інтерфейс повинен повноцінно функціонувати без використання комп'ютерної миші. У дизайн-системі розроблено спеціальні стани `:focus-visible` для всіх інтерактивних елементів (кнопок, полів вводу, карток занять), що забезпечує візуальну індикацію поточного положення користувача при навігації клавішею Tab.

4. Мережеві обмеження Враховуючи специфіку використання додатка (часті запити з мобільних пристроїв у приміщеннях університету, де можливе нестабільне з'єднання 3G/LTE), інтерфейс спроектовано за принципами Progressive Web App

(PWA). Візуальна архітектура передбачає чітку індикацію станів «Offline» (відсутність мережі) та локальне кешування останнього переглянутого розкладу.

Чітке дотримання вищезазначених технічних вимог при перенесенні спроектованої дизайн-системи у програмний код (наприклад, за допомогою фреймворків React або Vue.js) гарантує стабільну, швидку та безбар'єрну роботу вебдодатка для всіх учасників освітнього процесу.

3.3. Розробка та опис компонентної бази дизайн-системи (UI Kit)

Практична реалізація архітектури вебдодатка адміністрування розкладу була виконана у середовищі Figma шляхом створення масштабованої бібліотеки компонентів (UI Kit). Розробка базувалася на методології Atomic Design з активним використанням інструментарію Figma Variables (Змінні), що дозволило побудувати гнучку систему дизайн-токенів із повною підтримкою двох тем інтерфейсу: світлої (Light) та темної (Dark).

Основою візуальної мови розробленої системи стали два фундаментальні рівні: типографіка та колірна палітра.

1. Система типографіки

Як основний шрифт дизайн-системи було обрано гарнітуру Poppins. Цей вибір обґрунтовано її геометричною структурою, відсутністю засічок (Sans-serif) та високим рівнем читабельності на екранах із різною роздільною здатністю, що є критично важливим фактором для відображення щільних масивів табличних даних розкладу. Детальну шкалу типографіки, математично вивірену та поділену на функціональні групи, наведено у Додатку Б (рис. Б.1). До основних груп належать:

- **Headlines (Заголовки):** ієрархічна структура від H1 (44px, 128% Line Height, Medium) до H4 (18px, 132% Line Height, Bold).
- **Main Text (Основний текст):** базові стилі для контенту (Big, Middle, Small text) з використанням накреслень Regular та Light (від 16px до 22px), що

оптимізовані для зниження когнітивного навантаження під час тривалого читання.

- **Buttons (Елементи управління):** окрема група стилів для кнопок трьох розмірів (Large, Medium, Small).

2. Колірна архітектура та семантичні змінні (Variables)

На заміну статичному призначенню кольорів була розроблена дворівнева система дизайн-токенів за допомогою функціоналу Figma Variables (колекції Primary Colors та Token Colors). Базова палітра складається з двох основних спектрів, розбитих на градації від 5 до 100. Специфікацію колірної палітри та архітектуру семантичних змінних для світлого і темного режимів наведено у Додатку Б (рис. Б.2 – Б.3).

До структури палітри входять:

- **Primary (Основні кольори):** спектр теплих бежево-коричневих відтінків, які формують візуальну ідентичність системи.
- **Secondary (Другорядні кольори):** монохромна шкала сірих відтінків для фонів та допоміжного тексту.
- **Сигнальні кольори (Error/Success):** червоний (Error) та зелений (Success) для валідації станів системи.

Для забезпечення автоматичного перемикання тем усі UI-елементи використовують семантичні змінні (Aliases). Логіка трансформації інтерфейсу реалізована на рівні токенів:

- **back (Фони):** токени primary bg, secondary bg, tertiary bg. Наприклад, для світлої теми глобальний фон використовує токен Secondary 10, тоді як у темній він автоматично перемикається на Secondary 90.
- **button (Кнопки):** логіка станів інтегрована на рівні змінних. Стан **enabled** у світлій темі використовує Primary 30, **hover** — Primary 40, **active** — Primary 50, а **disabled** — Primary 10.

3. Просторова система та модульна сітка (Layout Grid)

Для забезпечення візуального балансу та адаптивності було налаштовано систему Layout Grid, яка складається з двох структур. Візуалізацію параметрів 12-колонкової та горизонтальної базової сітки наведено у Додатку Б (рис. Б.4):

- **Вертикальна 12-колонкова сітка:** Інженерний стандарт для десктопних інтерфейсів, що дозволяє гнучко керувати пропорціями (наприклад, розподіл простору між основним розкладом та сайдбаром).
- **Горизонтальна базова сітка:** Побудована з кроком у 8 пікселів (8pt Grid System). Усі просторові змінні (зовнішні та внутрішні відступи) й параметри міжрядкового інтервалу вирівнюються суворо за цими лініями, що гарантує геометричну чіткість інтерфейсу на будь-яких пристроях.

4. Побудова UI-компонентів

На базі розроблених токенів сформовано бібліотеку інтерактивних компонентів (кнопки, інпути, перемикачі). Усі елементи спроектовані з використанням функцій Auto Layout, що забезпечує динамічну зміну розмірів залежно від контенту, та механізму Variants, який дозволяє об'єднати різні стани (Default, Hover, Active, Disabled) в єдиний майстер-компонент. Такий підхід мінімізує кількість графічних артефактів у проекті та значно спрощує подальшу Frontend-розробку. Специфікацію та візуалізацію станів ключових компонентів системи наведено у Додатку Б (рис. Б.5).

3.4. Прототипування та керівництво користувача з експлуатації вебдодатка

На фінальному етапі було створено клікабельний високоточний (High-Fidelity) прототип освітнього порталу «ОАЕdu». Прототип охоплює ключові модулі: автентифікацію, головну сторінку, модуль «Розклад», «Новини» та «Волонтерство».

1. Навігація та архітектура головних екранів

Глобальна навігація системи базується на принципах консистентності,

масштабованості та мінімізації кількості кліків для доступу до критичної інформації. Основні елементи інтерфейсу та архітектуру ключових модулів системи наведено у Додатку В (рис. В.1 – В.4).

- **Глобальний заголовок (Header):** Реалізований як фіксований (Sticky) елемент, що містить логотип освітнього порталу, інтелектуальний пошук за ключовими словами, перемикач тем оформлення, центр сповіщень та розширений блок профілю користувача з відображенням рольової моделі (Студент / Викладач / Адміністратор).
- **Глобальний підвал (Footer):** Спроектований за патерном «Fat Footer» (розширений підвал). Він містить розгорнуту карту сайту для покращення навігації, швидкі посилання на ключові сервіси університету, контакти технічної підтримки, а також посилання на нормативні документи (політика конфіденційності, інклюзивні стандарти).
- **Модуль «Розклад»:** Є найбільш складним функціональним вузлом. Спроектований у вигляді інформаційно-щільної таблиці з підтримкою горизонтального скролінгу для збереження читабельності на малих екранах.
 - *Інтерактивна фільтрація:* Впроваджено систему динамічних фільтрів (дисципліна, тип заняття, аудиторія, викладач), яка працює на стороні клієнта без перезавантаження сторінки.
 - *Навігаційний блок:* Дозволяє швидко перемикатися між тижнями та семестрами, містить функцію експорту обраного розкладу у формати PDF або iCal.
 - *Контекстні підказки:* Кожна клітинка розкладу є інтерактивною — при натисканні відкривається Side-panel з детальною інформацією про заняття, посиланнями на навчальні матеріали (LMS) та мапою корпусу.
- **Модуль «Особистий кабінет користувача» (Student/Faculty Dashboard):** Інтегрований аналітичний простір, де візуалізується статистика користувача. Для студента це дашборд із графіками академічної успішності, трекінгом відвідуваності та накопиченими волонтерськими годинами. Використання

прогрес-барів та інфографіки дозволяє швидко оцінювати особисті досягнення.

- **Модуль «Довідник викладачів» (Faculty Directory):** Інструмент для забезпечення прозорості академічної комунікації. Включає пошукову систему з фільтрацією за кафедрами та науковими ступенями. Профіль кожного викладача містить інтерактивний графік вільних годин для консультацій та кнопки швидкого зв'язку.
- **Інтелектуальний модуль «AI Advisor» (Помічник вибору дисциплін):** Інноваційний функціонал системи, розроблений з урахуванням підходів Data-аналітики. Використовуючи алгоритми предиктивного аналізу, AI-асистент оцінює попередні оцінки студента, його спеціальність та вказані кар'єрні цілі, після чого генерує персоналізовані рекомендації щодо вибору вибіркових дисциплін. Це усуває проблему когнітивного перевантаження при формуванні індивідуального навчального плану.
- **Модулі «Новини» та «Волонтерство»:** Базуються на картковому інтерфейсі (Card-based layout). Картки містять метадані (дати, теги, статус активності), що дозволяє ефективно сканувати великі обсяги інформації. Пошук реалізовано через багаторівневий сайдбар із можливістю сортування за актуальністю та категоріями.

2. Імплементация стандартів WCAG: Панель доступності

Ключовою інновацією, що забезпечує інклюзивність розробленого інтерфейсу відповідно до міжнародних стандартів Web Content Accessibility Guidelines (WCAG 2.1), є інтегрована «Панель доступності» (швидкий виклик через комбінацію клавіш CTRL + U). Візуалізацію інтерфейсу панелі наведено у Додатку Г (рис. Г.1). Цей спеціалізований модуль дозволяє користувачеві персоналізувати інтерфейс у реальному часі без перезавантаження системи:

- **Контраст+ та Насичення:** алгоритмічна зміна значень колірних токенів для забезпечення високого коефіцієнта контрастності для осіб із порушеннями зору.
- **Шрифт+ та XL Збільшення:** динамічне масштабування типографіки (заголовків та основного тексту) без руйнування базової модульної сітки екрана.
- **Режим «Дислексія»:** автоматична заміна базового шрифту на спеціалізовану гарнітуру, що нівелює візуальне скупчення символів та полегшує сприйняття тексту.
- **Спеціальний курсор:** збільшення розміру вказівника та активних зон натискання (hitboxes) для осіб з порушеннями дрібної моторики.

3. Реалізація стратегії **Mobile First** та адаптивність інтерфейсу

Усі макети освітньої екосистеми спроектовано за принципами Responsive Web Design (Адаптивний вебдизайн). Завдяки математично розрахованим правилам поведінки компонентів (Auto Layout) та гнучкій системі брейкпоінтів (Breakpoints), при переході до мобільної версії екрана складні багатоклонкові структури автоматично трансформуються в оптимізований формат (Single-column layout). Приклади трансформації інтерфейсу для мобільних пристроїв наведено у Додатку Г (рис. Г.2 та Г.4). Таке архітектурне рішення гарантує зручний доступ до навчального розкладу зі смартфонів, повністю усуваючи необхідність горизонтального скролінгу та зберігаючи повний функціонал десктопної версії.

Висновки до розділу 3

У третьому розділі було здійснено комплексну практичну реалізацію спроектованої архітектури освітньої екосистеми у хмарному середовищі Figma. Результати виконаної роботи дозволяють сформулювати такі висновки:

- 1. Обґрунтовано ефективність обраного інструментарію.** Застосування математичних алгоритмів Auto Layout та механізму семантичних змінних (Variables) дозволило створити гнучку параметризовану систему, що мінімізує кількість графічних артефактів та значно оптимізує подальший процес Frontend-розробки.
- 2. Практично реалізовано масштабовану компонентну базу (UI Kit).** На основі методології Atomic Design сформовано бібліотеку інтерфейсних елементів. Впровадження 12-колонкової вертикальної та 8-піксельної горизонтальної сіток гарантувало геометричну точність, а розроблена колірна архітектура забезпечила безшовну підтримку світлої та темної візуальних тем.
- 3. Створено високоточний інтерактивний прототип (High-Fidelity).** Спроектовано клікабельну модель освітнього порталу, яка об'єднує інноваційну систему адміністрування розкладу, аналітичний дашборд студента, інтелектуального помічника (AI Advisor) та довідник викладачів. Це підтверджує життєздатність та масштабованість розробленої архітектури.
- 4. Успішно розв'язано стратегічне завдання забезпечення інклюзивності.** Інтеграція інноваційної «Панелі доступності» з алгоритмами динамічної зміни контрастності, масштабування типографіки та режимом «Дислексія» робить розроблений інтерфейс повністю сумісним із міжнародними стандартами WCAG 2.1.
- 5. Визначено технічні вимоги до середовища розгортання.** Враховано ключові метрики продуктивності Core Web Vitals (зокрема LCP та CLS) та правила кросбраузерності, що закладає надійний фундамент для стабільної роботи та

швидкого завантаження високонавантажених сторінок розкладу на будь-яких пристроях.

ЗАГАЛЬНІ ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальне завдання щодо проектування адаптивного користувацького інтерфейсу та комплексної дизайн-системи вебдодатка для адміністрування навчального розкладу. В умовах стрімкої цифровізації освітніх процесів розроблений продукт пропонує сучасний підхід до управління високонавантаженими даними, орієнтуючись на міжнародні стандарти інклюзивності та доступності.

У результаті виконання роботи було досягнуто поставленої мети та отримано такі основні результати:

- 1. Проведено системний аналіз предметної області.** Дослідження існуючих програмних рішень виявило критичну проблему «інтерфейсної фрагментації» та відсутність єдиних дизайн-стандартів в академічних системах. За допомогою інструментарію UML (діаграми використання та діяльності) було формалізовано рольові моделі ключових акторів (студентів, викладачів, адміністраторів) та їхні функціональні потреби, що стало основою для подальшого проектування.
- 2. Спроектовано інформаційну архітектуру та логіку взаємодії.** Визначено структуру ключових об'єктів системи. Розроблено сценарії користувацької взаємодії (User Flows), що забезпечують безшовну роботу з розкладом, алгоритмами валідації конфліктів та інструментами динамічної фільтрації даних.
- 3. Розроблено масштабовану дизайн-систему (UI Kit).** Застосовуючи методологію Content-First та Atomic Design, у хмарному середовищі Figma створено комплексну бібліотеку компонентів. Впровадження інструментарію Auto Layout та семантичних дизайн-токенів (Variables) дозволило створити параметризований фреймворк із повною підтримкою світлої та темної візуальних тем (Light i Dark modes).

4. **Забезпечено імплементацію математичних моделей адаптивності.** Для досягнення гармонійного масштабування інтерфейсу на різних пристроях впроваджено 12-колонкову вертикальну та 8-піксельну горизонтальну базову сітки (Layout Grid). Дизайн системи оптимізовано за стратегією Mobile First, що гарантує безперешкодний доступ до освітніх послуг зі смартфонів.
5. **Реалізовано інклюзивне середовище (WCAG 2.1).** Головним інноваційним рішенням роботи стала розробка інтегрованої «Панелі доступності». Вона дозволяє користувачам персоналізувати інтерфейс у режимі реального часу: алгоритмічно змінювати контрастність колірних токенів, масштабувати типографіку та застосовувати спеціалізовані режими (наприклад, для дислексії). Усі текстові та графічні компоненти пройшли алгоритмічну перевірку на відповідність коефіцієнту контрастності ($CR \geq 4.5:1$).
6. **Створено високоточний інтерактивний прототип (High-Fidelity).** Спроектовано клікабельну модель освітнього порталу «OAEdu», що об'єднує модулі розкладу, новин та волонтерства. Прототип верифікує архітектурні рішення, повністю розв'язує задачу інклюзивності та підготовлений до етапу Handoff для подальшої програмної реалізації (Frontend-розробки).

Таким чином, результати кваліфікаційної роботи мають високу практичну цінність. Запропонована дизайн-система не лише оптимізує процеси адміністрування навчального розкладу, а й створює безбар'єрне цифрове середовище, доступне для всіх учасників освітнього процесу незалежно від їхніх фізичних потреб чи технічних можливостей пристроїв.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. – Київ : ДП «УкрНДНЦ», 2016. – 26 с.
2. ДСТУ ISO 9241-210:2021. Ергономіка взаємодії людина-система. Частина 210. Людиноорієнтоване проектування інтерактивних систем. – Київ : ДП «УкрНДНЦ», 2022.
3. Apple Developer. Human Interface Guidelines. URL: <https://developer.apple.com/design/human-interface-guidelines/> (дата звернення: 15.11.2025).
4. Color.org. A Standard Default Color Space for the Internet - sRGB. URL: <https://color.org/srgb.pdf> (дата звернення: 02.10.2025).
5. Figma Documentation. Creating and organizing components. URL: <https://help.figma.com/hc/en-us/articles/360038662654> (дата звернення: 28.12.2025).
6. Figma Documentation. Explore auto layout properties. URL: <https://help.figma.com/hc/en-us/articles/360040451373> (дата звернення: 14.01.2026).
7. Figma Documentation. Guide to variables in Figma. URL: <https://help.figma.com/hc/en-us/articles/15145852043671> (дата звернення: 05.11.2025).
8. Figma Documentation. Prototyping in Figma. URL: <https://help.figma.com/hc/en-us/articles/360040314193> (дата звернення: 20.03.2026).
9. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. – Addison-Wesley Professional, 2003. – 208 p.
10. Frost B. Atomic Design. – Brad Frost Web, 2016. – 195 p.
11. Google Design. Material Design 3 Guidelines. URL: <https://m3.material.io/> (дата звернення: 10.10.2025).

12. Google Fonts. Poppins Specimen. URL: <https://fonts.google.com/specimen/Poppins> (дата звернення: 03.12.2025).
13. Gothelf J., Seiden J. Lean UX: Designing Great Products with Agile Teams. – O'Reilly Media, 2021. – 216 p.
14. IBM Documentation. UML Use Case Diagrams. URL: <https://www.ibm.com/docs/en/rational-soft-arch/> (дата звернення: 22.01.2026).
15. Interaction Design Foundation. Mobile-First Design. URL: <https://www.interaction-design.org/literature/topics/mobile-first> (дата звернення: 17.02.2026).
16. Interaction Design Foundation. What is Design Systems? URL: <https://www.interaction-design.org/literature/topics/design-systems> (дата звернення: 08.11.2025).
17. InVision. Design Systems Handbook. URL: <https://www.designbetter.co/design-systems-handbook> (дата звернення: 25.10.2025).
18. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. – New Riders, 2014. – 216 p.
19. Laws of UX. A collection of maxims and principles that designers can consider when building user interfaces. URL: <https://lawsofux.com/> (дата звернення: 30.01.2026).
20. Mozilla Developer Network (MDN). CSS Grid Layout. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout (дата звернення: 11.04.2026).
21. Mozilla Developer Network (MDN). Responsive design. URL: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design (дата звернення: 18.12.2025).
22. Nielsen J. Usability Engineering. – Academic Press, 1994. – 362 p.

23. Nielsen Norman Group. 10 Usability Heuristics for User Interface Design. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (дата звернення: 05.02.2026).
24. Nielsen Norman Group. Mobile First Design. URL: <https://www.nngroup.com/articles/mobile-first/> (дата звернення: 22.11.2025).
25. Norman D. The Design of Everyday Things. – Basic Books, 2013. – 368 p.
26. OMG. Unified Modeling Language (UML) Specification. URL: <https://www.omg.org/spec/UML/> (дата звернення: 14.03.2026).
27. Smashing Magazine. A Comprehensive Guide to UI Design. URL: <https://www.smashingmagazine.com/> (дата звернення: 09.02.2026).
28. Spec Network. The 8pt Grid System. URL: <https://builttoadapt.io/intro-to-the-8-point-grid-system> (дата звернення: 04.01.2026).
29. Tidwell J. Designing Interfaces: Patterns for Effective Interaction Design. – O'Reilly Media, 2020. – 578 p.
30. UX Collective. Best Practices for Dashboard Design. URL: <https://uxdesign.cc/> (дата звернення: 27.10.2025).
31. W3C. Web Accessibility Evaluation Tools List. URL: <https://www.w3.org/WAI/ER/tools/> (дата звернення: 16.03.2026).
32. W3C. Web Accessibility Initiative (WAI). URL: <https://www.w3.org/WAI/> (дата звернення: 12.11.2025).
33. W3C. Web Content Accessibility Guidelines (WCAG) 2.1. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 21.02.2026).
34. WebAIM. Contrast Checker. URL: <https://webaim.org/resources/contrastchecker/> (дата звернення: 19.01.2026).
35. WebAIM. Typefaces and Fonts for Accessibility. URL: <https://webaim.org/techniques/fonts/> (дата звернення: 08.03.2026).

ДОДАТКИ

ДОДАТОК А

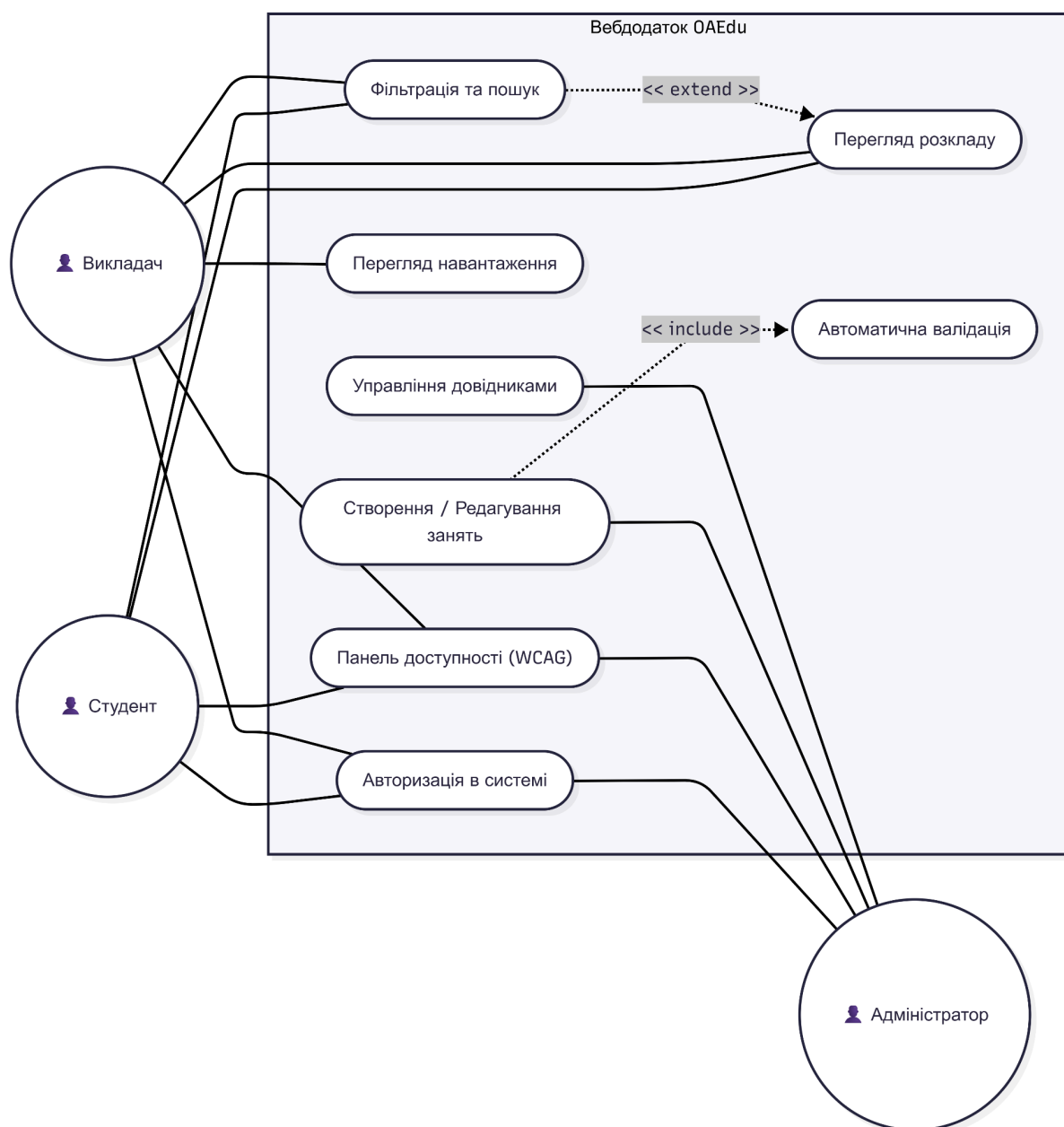


Рис. А.1. UML-діаграма використання (Use Case)

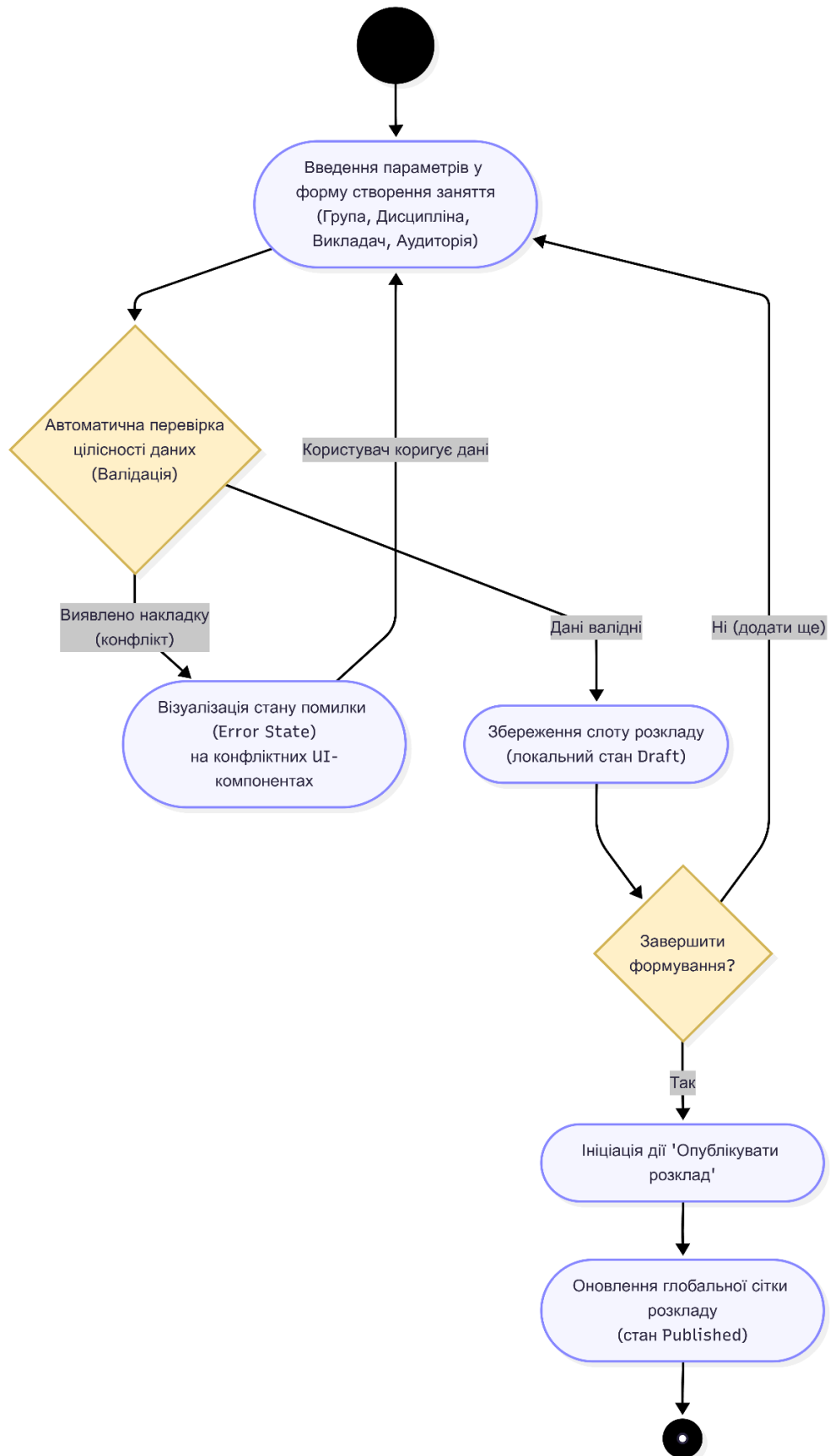


Рис. А.2. UML-діаграма діяльності (Activity).

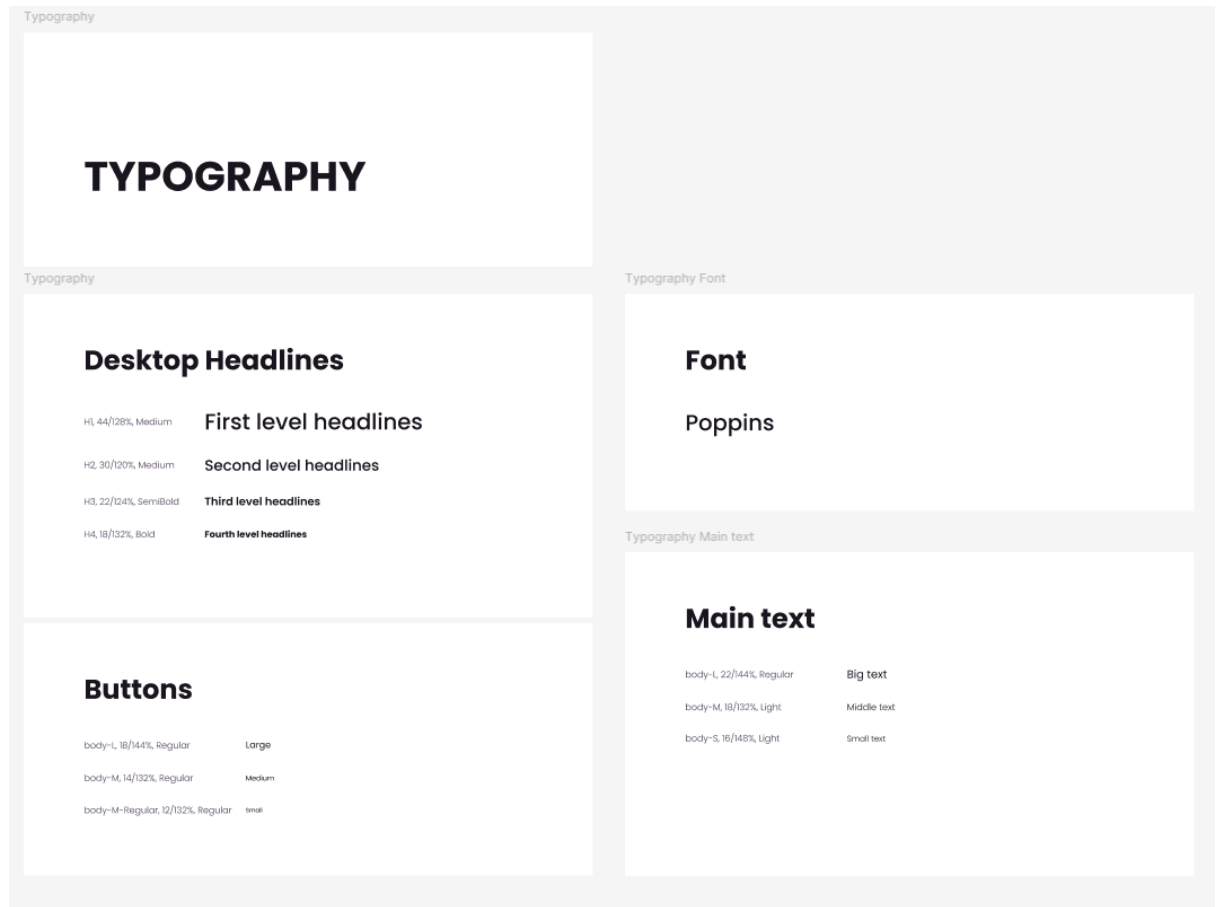


Рис. Б.1. Структура типографіки дизайн-системи (гарнітура Poppins)

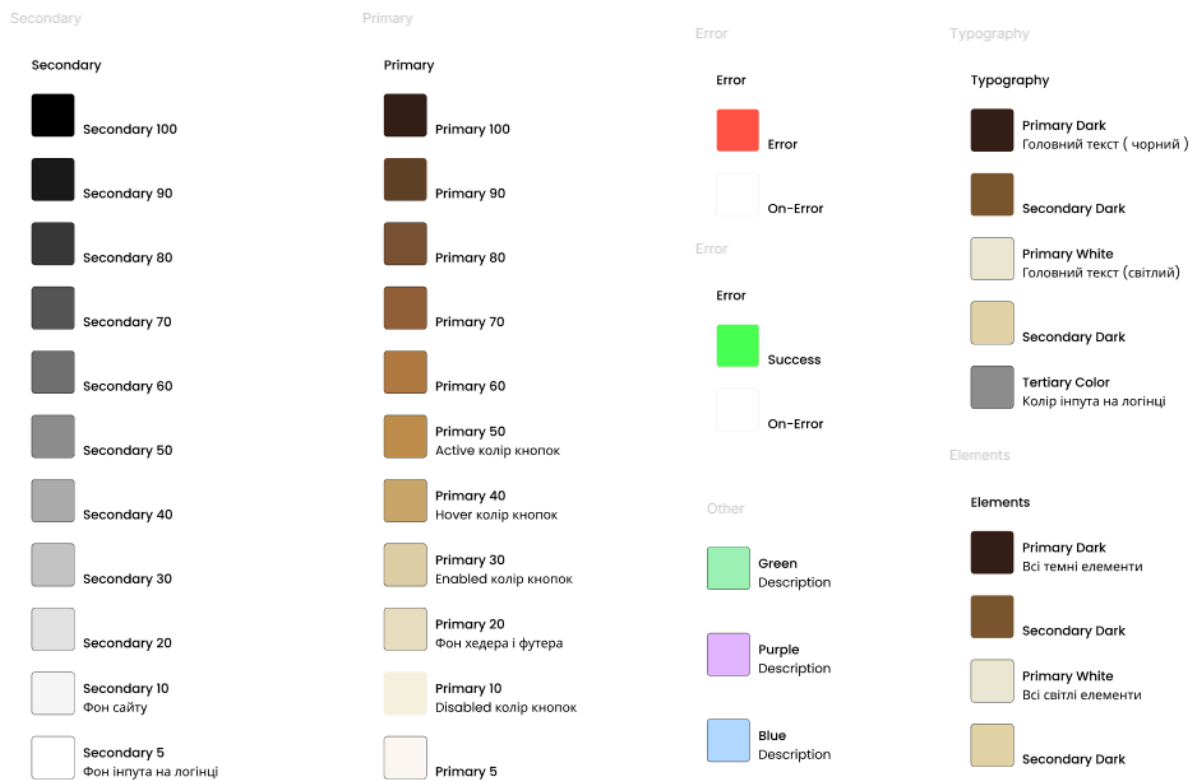


Рис. Б.2. Специфікація базової колірної палітри дизайн-системи

Variables			light	dark
Collections	+	Name		
Primary Colors	32			
Token Colors	15	typography		
Groups	☰	🔄 dark	default/Black	default/White
All	15	🔄 light	default/White	default/Black
typography	5	🔄 secondary dark	default/Secondary Black	default/Secondary White
elements	3	🔄 secondary light	default/Secondary White	default/Secondary Black
back	3	🔄 tertiary	secondary/Secondary 50	secondary/Secondary 50
button	4			
		elements		
		🔄 dark	default/Black	default/White
		🔄 light	default/White	default/Black
		🔄 disabled	secondary/Secondary 50	secondary/Secondary 50
		back		
		🔄 primary bg	secondary/Secondary 10	secondary/Secondary 80
		🔄 secondary bg	primary/Primary 20	primary/Primary 80
		🔄 tertiary bg	secondary/Secondary 5	secondary/Secondary 90
		button		
		🔄 enabled	primary/Primary 30	primary/Primary 70
		🔄 hover	primary/Primary 40	primary/Primary 60
		🔄 active	primary/Primary 50	primary/Primary 50
		+ Create variable		

Рис. Б.3. Налаштування семантичних дизайн-токенів (Figma Variables) для Light/Dark режимів



Рис. Б.4. Структура 12-колонкової та горизонтальної базової сітки (Layout Grid)

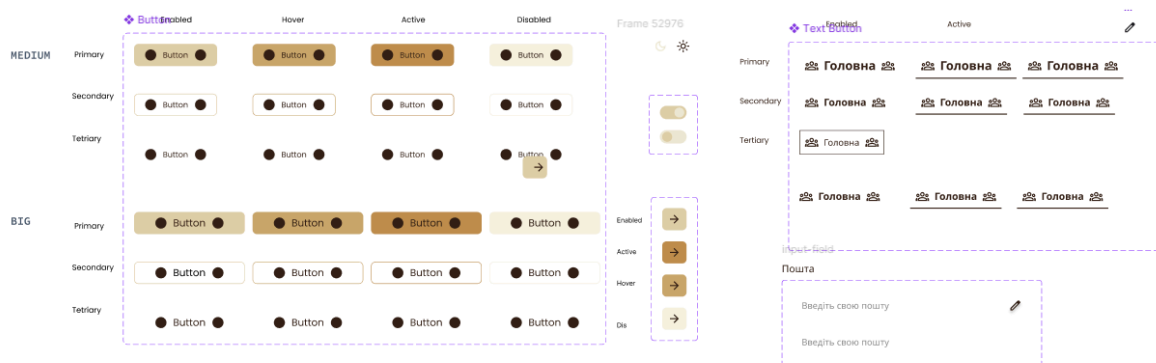


Рис. Б.5. Бібліотека інтерактивних компонентів дизайн-системи та специфікація їхніх станів

OA Edu

Головна Розклад Новини Волонтерство

Строзюк Роман Володимирович roman.strozuk@oa.edu.ua

26.05.2025 – 31.05.2025

Понеділок 26.05.2025 **Вівторок 27.05.2025** Середа 28.05.2025 Четвер 29.05.2025 П'ятниця 30.05.2025 Субота 31.05.2025 Неділя 01.06.2025

#	Час	Дисципліна	Аудиторія	Тип	Викладач	Групи	Пари	Кодове слово
#		Дисципліна	Аудиторія	Тип	Викладач	Групи	Пари	Кодове слово
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk
1	9:00 – 10:40	Криптограф. захист інформації	23	Лабораторна	Коцюк Юрій Анатолійович	Криптографічний захист інформації КН-33	21/26	yurlykotslyk

Кодове слово
 Дисципліна
 Аудиторія
 Викладач
 Групи
 Пари
 Час
 Тип

© 2025 OA Edu .

Telegram Bot

Social media

Application

Рис. В.1. Інтерфейс модуля «Розклад» з інтерактивною сіткою занять та глобальними елементами навігації (Header/Footer)

Рис. В.2. Аналітичний дашборд особистого кабінету користувача

Рис. В.3. Інтерфейс модулів комунікації: «Довідник викладачів», «Новини» та «Волонтерство»

Рис. В.4. Інтерактивний процес роботи модуля «AI Advisor» для рекомендації вибіркокових дисциплін

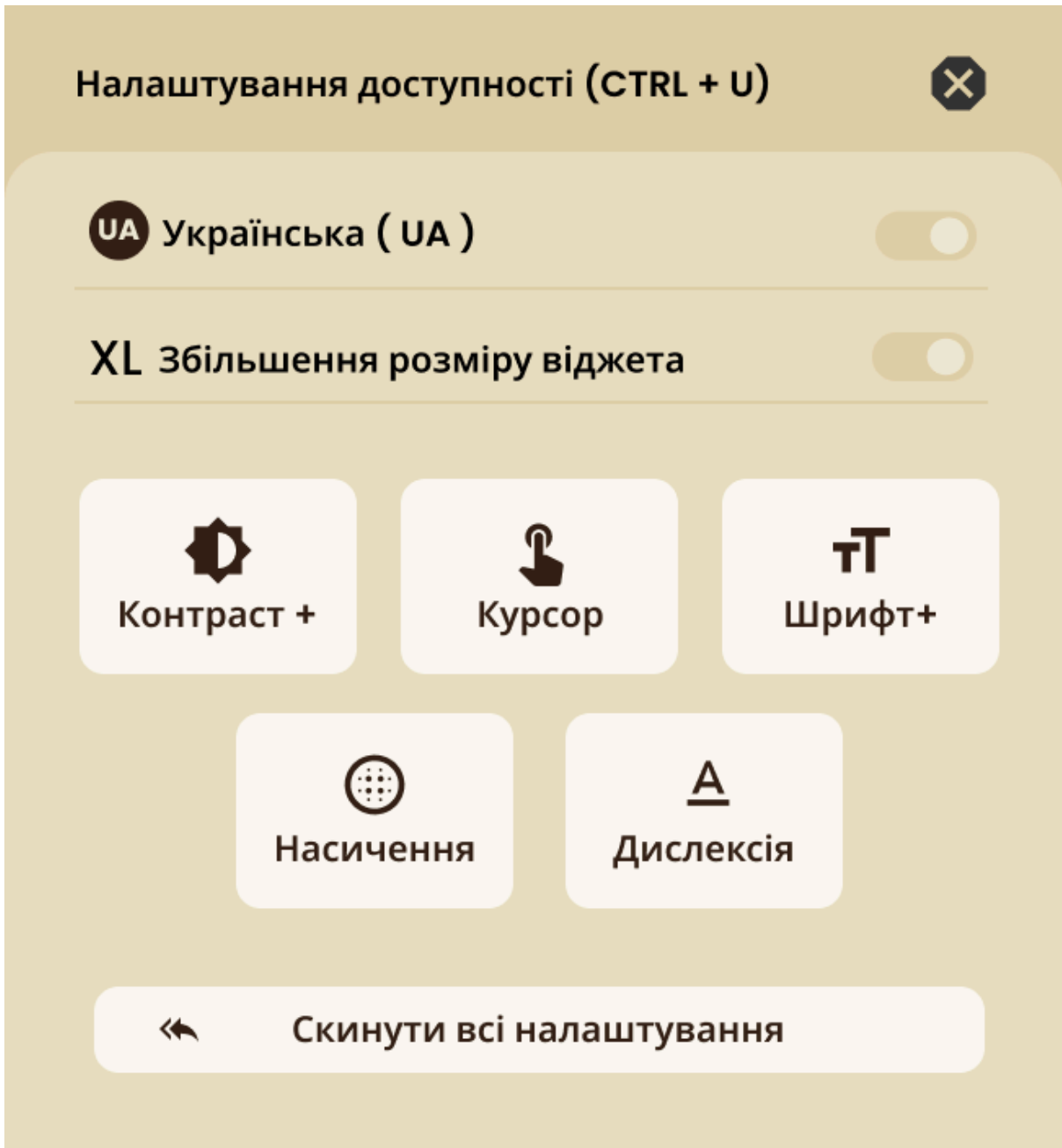


Рис. Г.1. Інтерфейс інтегрованої «Панелі доступності» для персоналізації відображення

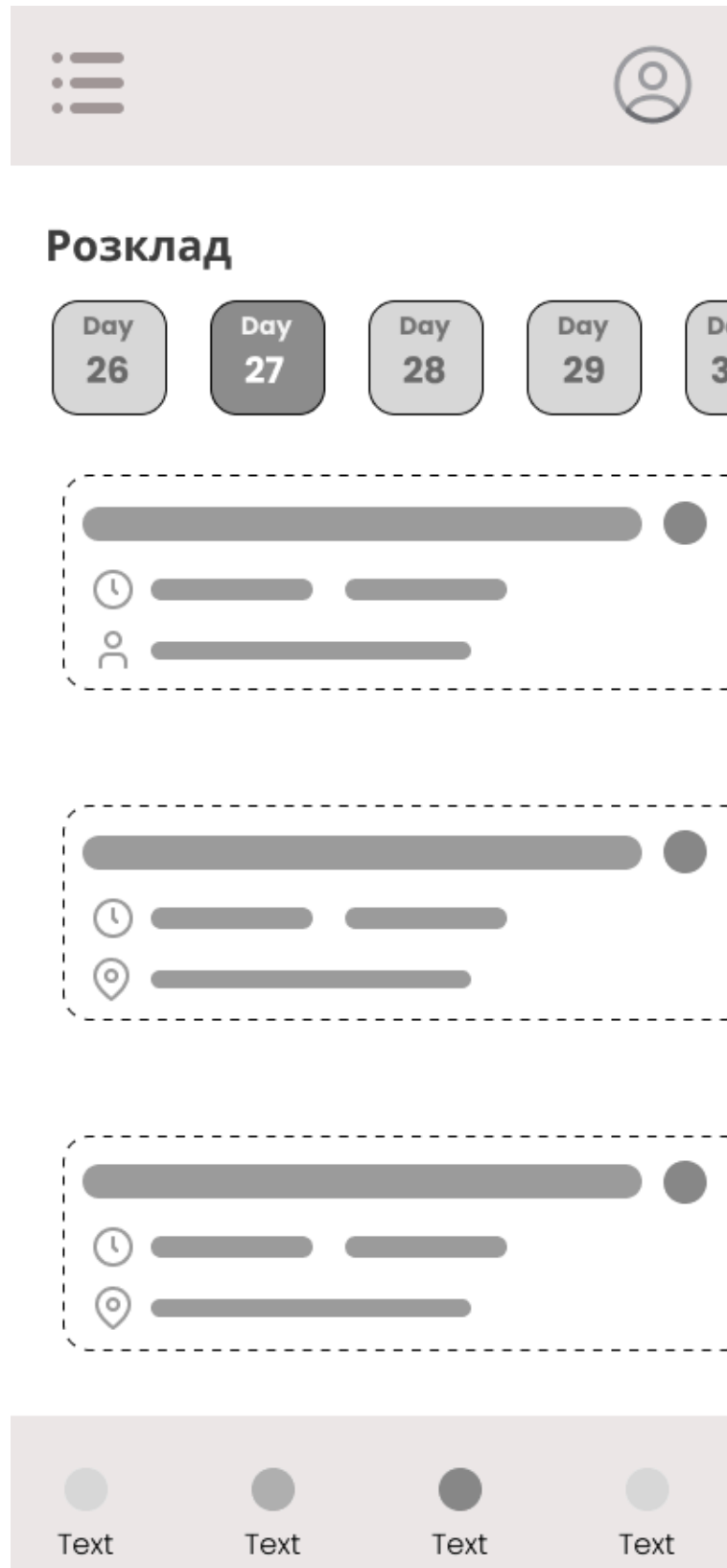


Рис. Г.2. Макет сторінки розкладу мобільного застосунку

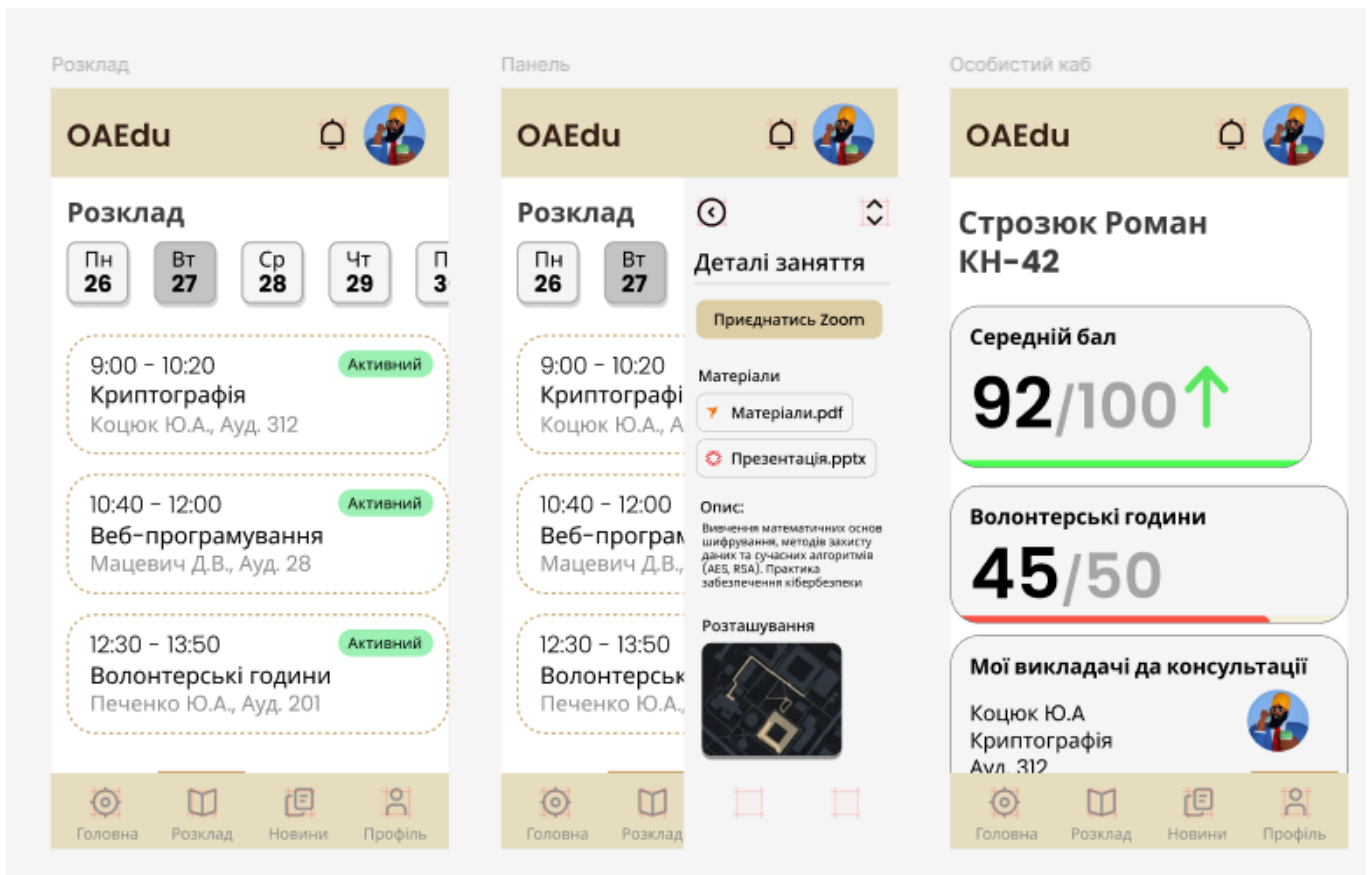


Рис. Г.4. Приклади готових сторінок мобільного додатку