**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

До лабораторної роботи №1

**На тему: “Формування технічного завдання як результат аналізу та визначення вимог”**

**З дисципліни:** *“Вступ до інженерії програмного забезпечення”*

**Лектор:**

к. т. н. доцент каф. ПЗ

Левус Є. В.

**Виконав:**

ст. гр. ПЗ-16

Кабачок Т. О.

**Прийняв:**

асист. каф. ПЗ

Легкий М. В.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2025 р.

Львів – 2025

**Тема:** Формування технічного завдання як результат аналізу та визначення вимог.

**Мета:** Навчитися складати технічне завдання до розробки програми.

**Теоретичні відомості**

12, 18, 38

**12. Які документи використовують для опису вимог?**

* Список вимог – короткий опис основних вимог до продукту
* Прототип – макет системи, вимоги до дизайну
* Сценарії використання – опис усіх взаємодій користувача з системою
* Діаграми прецедентів – звʼязки між користувачами системи та функціями
* Технічне завдання (специфікація вимог) – вихідний документ для розробки, в якому описані основні цілі розробки, список принципових вимог, терміни та етапи розробки, регламент характеристик готового продукту

**18. Яка вимога вважається верифікованою? Наведіть правильний та неправильний приклад.**

Верифікована вимога – це та вимога, яку можна однозначно перевірити тестуванням коду.

* Правильна вимога – Програма повинна приймати рядок довжиною від 1 до 255 символів англійського алфавіту.
* Неправильна вимога – Програма повинна коректно обробляти введений користувачем текст

**38. Що мається на увазі, коли говорять про естетичність інтерфейсу користувача?**

Під поняттям естетичність зазвичай мається на увазі стриманість у використанні кольорів, декору шрифтів та стильовим обʼєднанням усіх сторінок інтерфейсу. Підтримується принцип «Краще менше, але краще».

**Постановка завдання**

**Умова.** Скласти технічне завдання (концептуальний рівень) до програми згідно індивідуального варіанту (№1-30) за поданим вище планом. Представити зображення UI для демонстрації взаємодії користувача з програмою для кожної функції з пункту 4.1. З пункту 4.4 зазначити дві нефункціональні вимоги. З пункту 6 вказати документи, які будуть створені. З пункту 7 вказати два інформаційні джерела за темою програмування.

**Варіант завдання.** З клавіатури ввести послідовність записів, які містять дані про результати сесії студентів групи: <Прізвище>, <Імʼя>, <Дата народження>, <Список екзаменаційних оцінок>. Роздрукувати введені дані у вигляді таблиці, а також подати інформацію згідно варіанту. Передбачити зчитування вхідних даних з файлу та запис результатів у файл. Роздрукувати список студентів, які отримали оцінки 4 і 5 на іспитах, у спадному порядку за віком. Визначити двох наймолодших студентів серед них. Вилучити зі списку дані про студентів, які не мають оцінки 2.

**Технічне завдання**

1. **Загальні положення:**

1.1. Найменування роботи: Програмне забезпечення для ведення обліку успішності студентів за результатами сесійного контролю.

1.2. Умовне позначення: Student score list.

1.3. Найменування замовника та розробника:

Замовник: Національний університет «Львівська політехніка».

Розробник: Кабачок Тарас Олександрович

1.4. Терміни початку та закінчення роботи:

Початок: 24.02.2025

Кінець: 13.03.2025

1. **Призначення системи:**

2.1. Цілі створення програмного забезпечення: ведення обліку успішності студентів за результатами сесійного контролю.

2.2. Основні очікувані результати: готова програма для ведення таблиці з оцінками студентів за сесію, з можливістю сортування студентів.

2.3. Область застосування даного продукту: вищі навчальні заклади.

1. **Об’єкти даних:**

Файл містить дані про студентів, де кожен запис включає прізвище, ім’я, дату народження та п’ять оцінок. Прізвище та ім’я складаються виключно з літер латинського алфавіту (a-z, A-Z) і мають довжину від 1 до 50 символів. Дата народження подається у форматі DD.MM.YYYY. Оцінки представлені п’ятьма цілими числами в діапазоні від 0 до 5. Кожен студент записаний в окремому рядку.

1. **Вимоги до програмного забезпечення:**

4.1. Функціональні вимоги:

* відкриття файлу з даними про успішність студентів або створення нового.
* відображення даних у вигляді таблиці.
* можливість внесення змін до даних про студентів.
* можливість додавання нового студента.
* можливість видалення студента.
* пошук студента за його прізвищем чи імʼям.
* сортування студентів від наймолодшого до найстаршого.
* визначення студентів з оцінками лише 4 та 5.
* визначення двох наймолодших студентів які мають оцінки лише 4 і 5.
* видалення даних про студентів, які не мають оцінки 2.

4.2. Системні вимоги:

* 64-бітний або 32-бітний з тактовою частотою від 1 ГГц, підтримка архітектури x86, x86\_64 або ARM.
* відеокарта не менше ніж 128 мегабайтів памʼяті;
* 40 МБ вільного простору;
* ОС:  мінімальні вимоги :
  + - macOS 13.4.1, Windows 7, Linux з ядром 5.4

4.3. Нефункціональні вимоги:

* Помилка при спробі відкриття файлу, що не відповідає вимогам (файл .xlsx, створений в цій же програмі).
* Швидкодія – час відповіді системи не перевищує 1 секунди.

4.4. Вимоги до користувацького інтерфейсу:

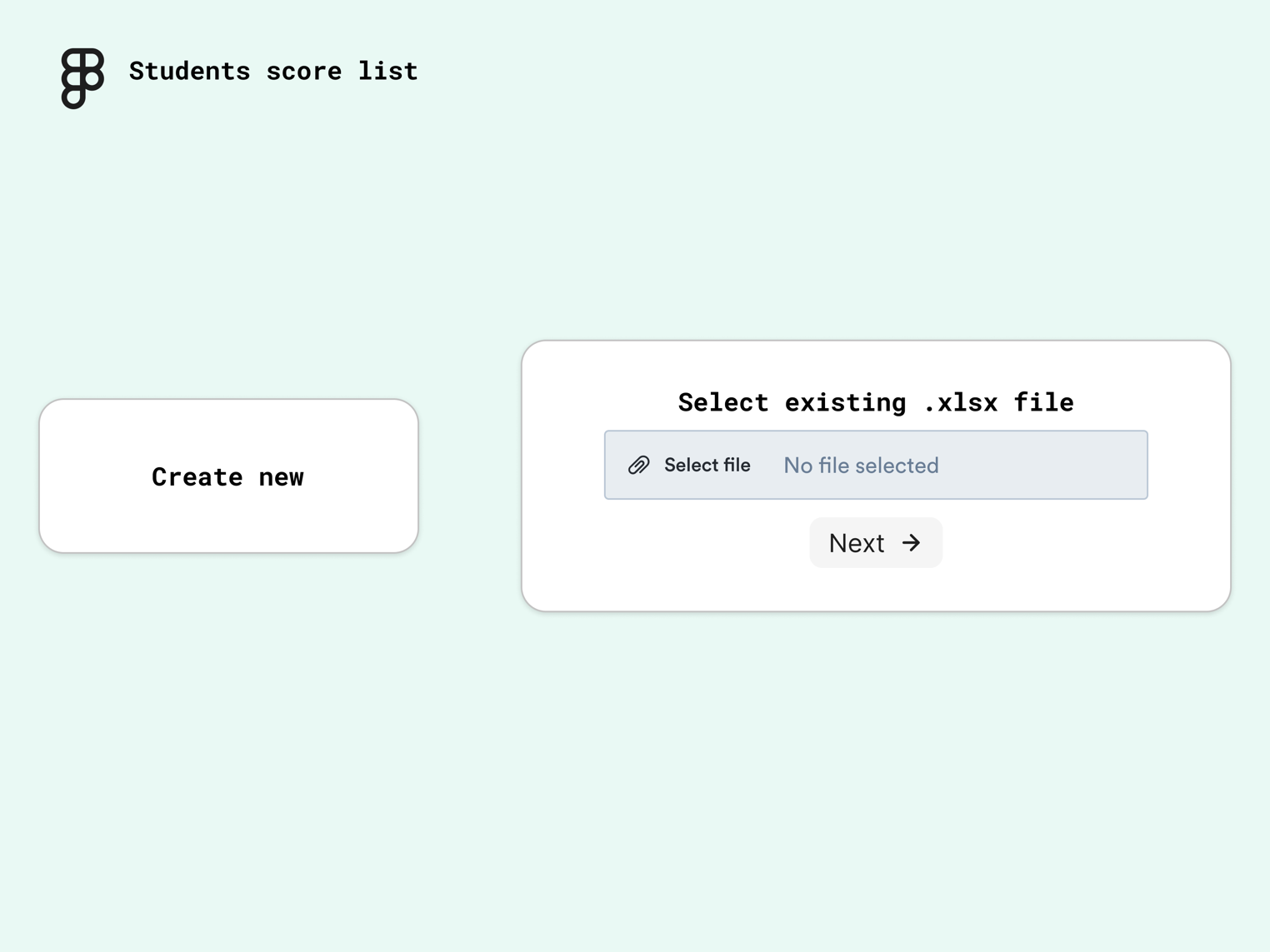


Рис. 1. Головне вікно програми після запуску

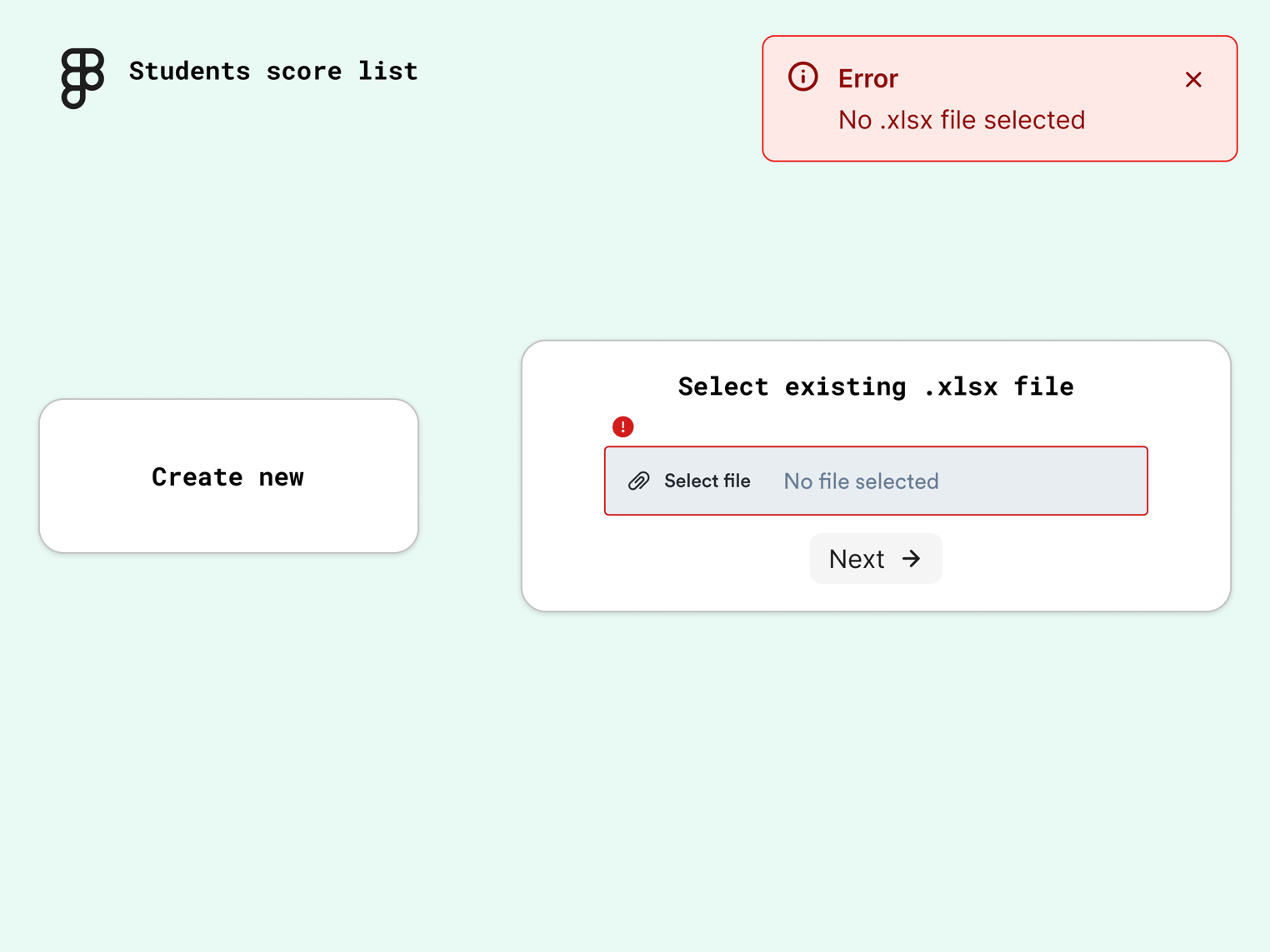


Рис. 2. Нефункціональна вимога програми, перевірка на коректність завантажуваного файлу



Рис. 3. Процес відкриття чи створення файлу супроводжується візуальними елементами завантаження

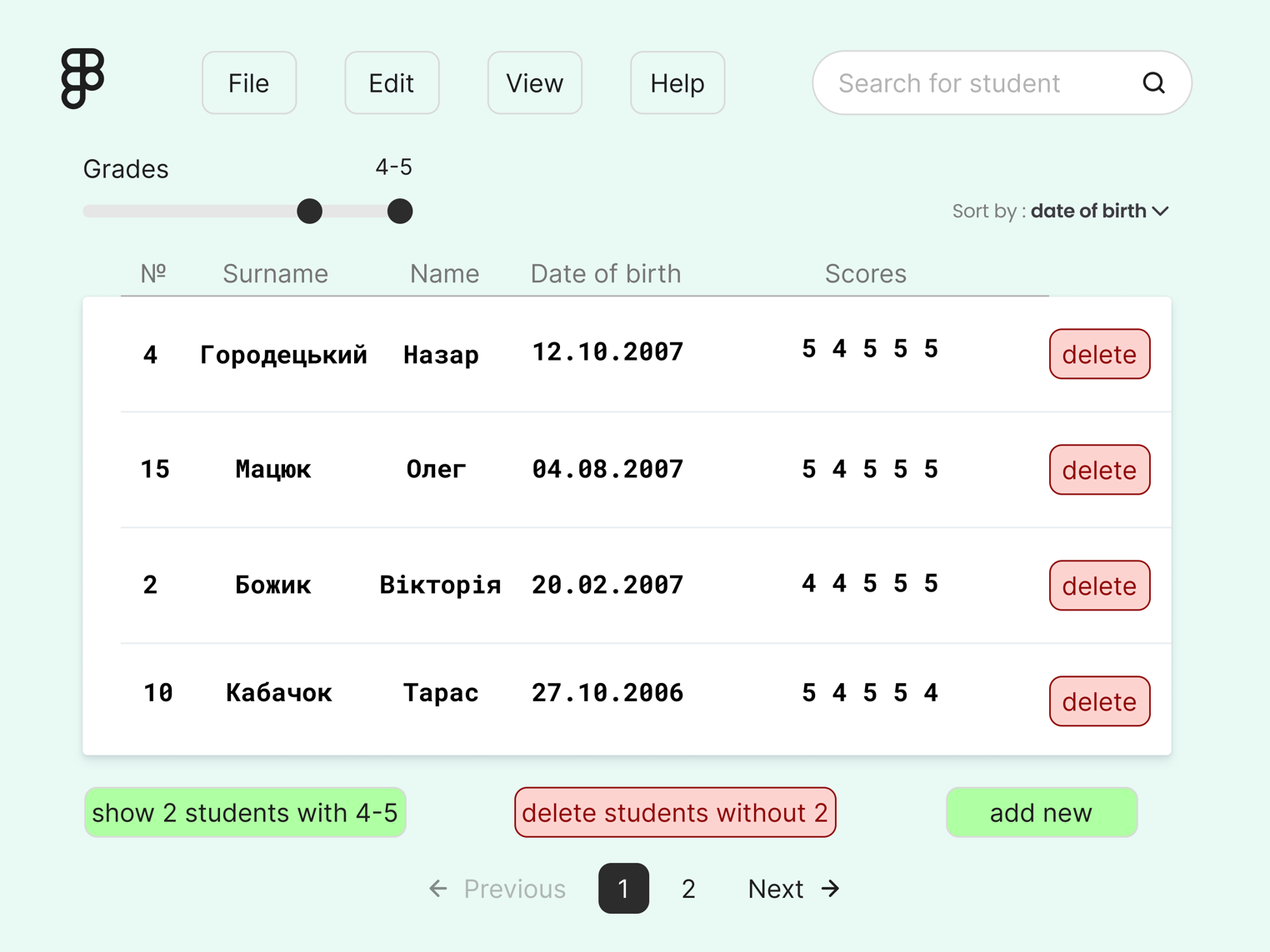


Рис. 4. Головне вікно відкритого файлу з таблицею студентів

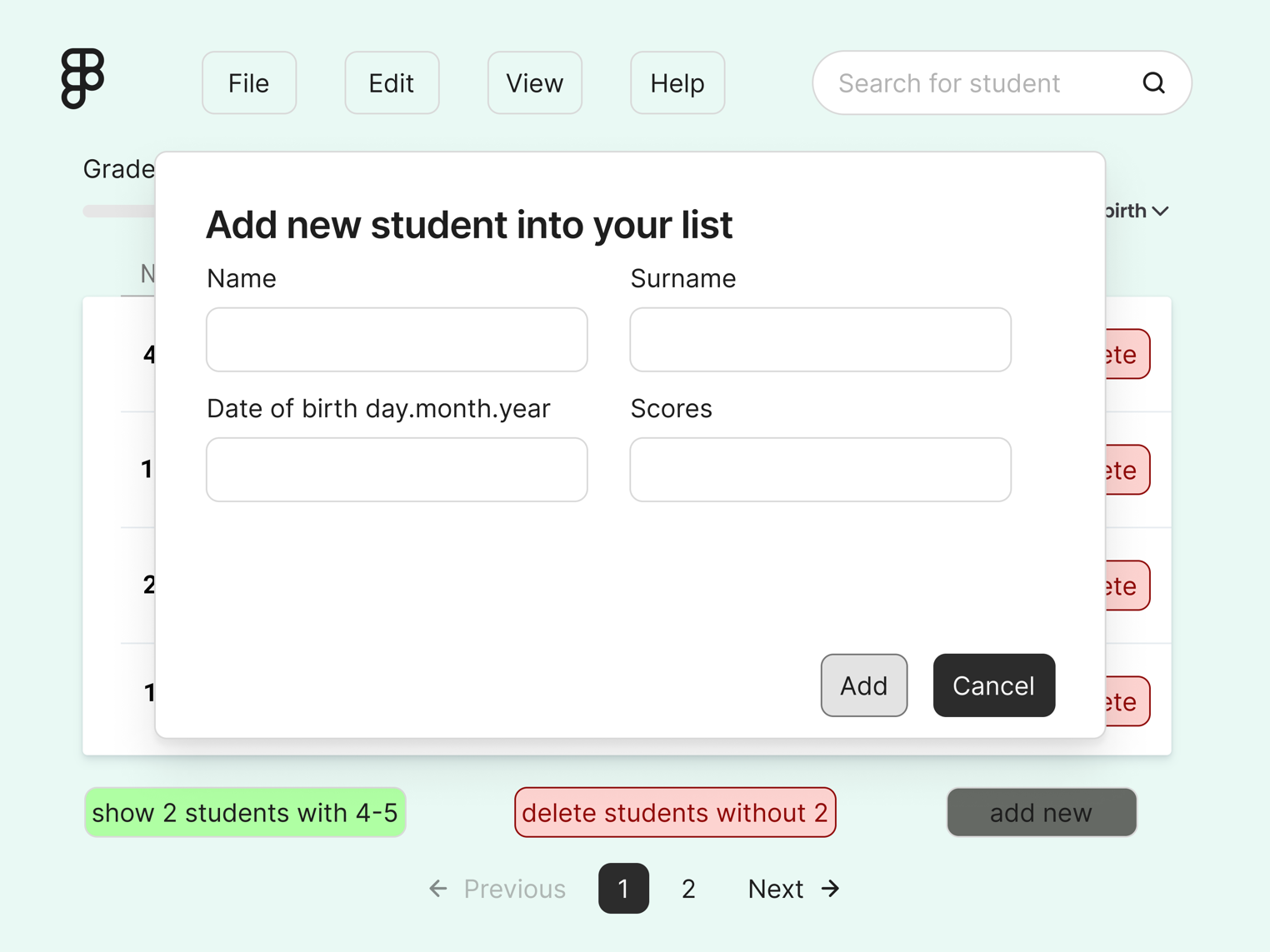


Рис. 5. Доступні опції при натисканні кнопки add new (додавання нового студенту до списку)

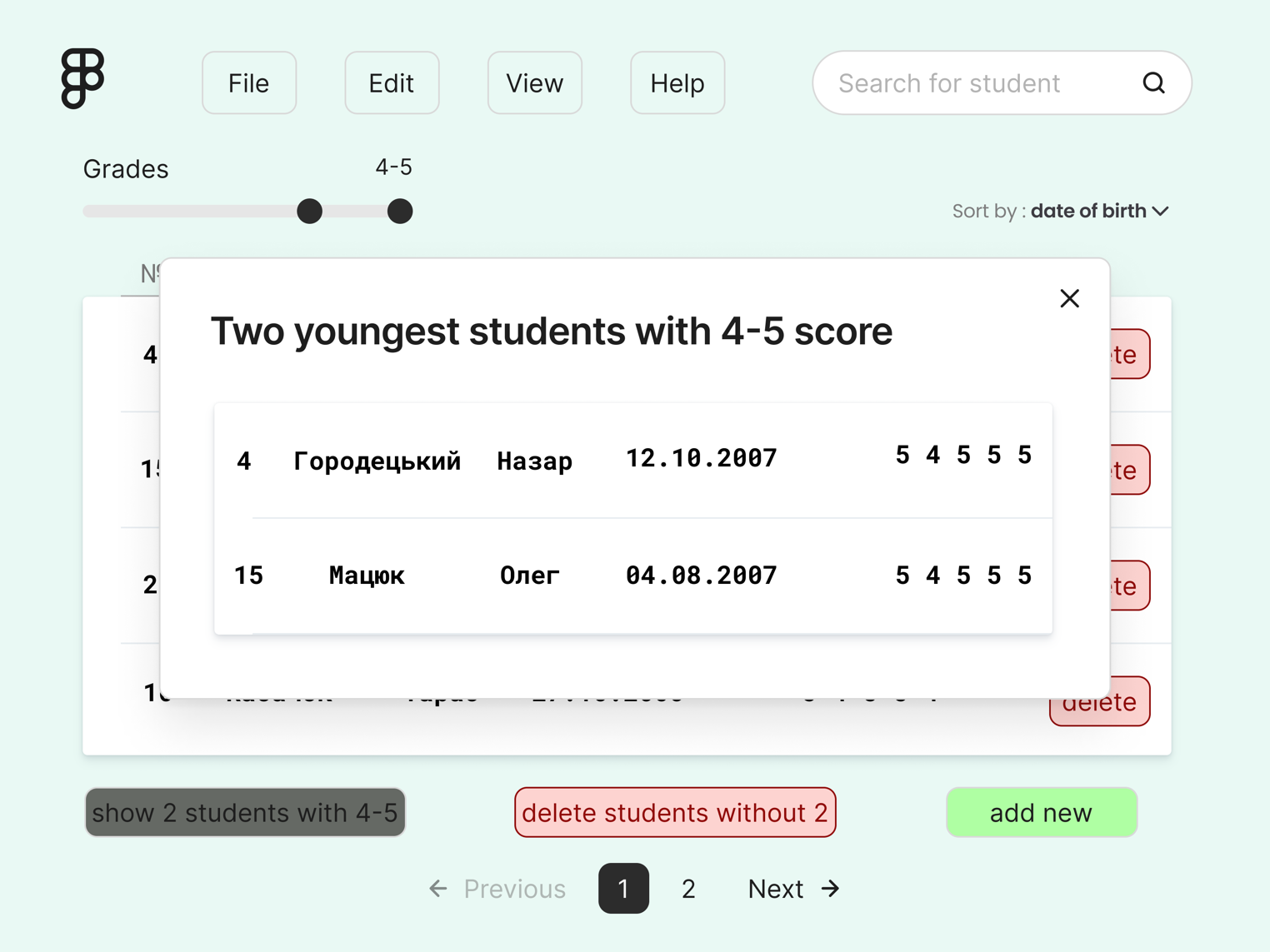


Рис. 6. Окреме вікно, що демонструє програма при натисканні кнопки show 2 students with 4-5 (на екрані зображається окреме вікно з двома наймолодшими студентам які мають оцінки лише 4 та 5)

* 1. **Стадії розробки:**

|  |  |
| --- | --- |
| Етап розробки | Характеристика |
| Аналіз та специфікація вимог | Збирання та аналіз вимог до програми. Визначення та формулювання функціональних та нефункціональних вимог. |
| Проектування | Розробка архітектури програми, планування інтерфейсу, вибір структур даних, технологій та інструментів для розробки. |
| Кодування | Написання програмного коду відповідно до проектної документації. Реалізація вимог та інтеграція компонентів. |
| Тестування | Реалізація тестів для перевірки роботи програмного забезпечення, виявлення та виправлення помилок. |
| Експлуатація | Встановлення програмного забезпечення на компʼютерне забезпечення замовника. |
| Супровід | Регулярне обслуговування в консультативній формі для замовника. Оновлення програми, допомога у вирішення проблем. |

* 1. **Вимоги до програмної документації:**
* Технічне завдання (ТЗ)
  + Опис можливостей та обмежень програми.
  + Вибір мови програмування та вимоги до пристрою.
  + Способи взаємодії з іншими системами (якщо потрібно).
* Блок-схеми
  + Графічне зображення логіки програми.
* Код програми
  + Зрозуміло написаний код.
  + Коментарі для пояснення роботи.
* Тестування
  + Перелік тестів та їх результати.
  + Очікуваний та реальний результат.
  1. **Посилання**
* Левус Є. В. Життєвий цикл програмного забезпечення: навчальний посібник Є. В.  Левус, Т. A. Марусенкова. – Львів: Видавництво Львівської політехніки, 2017. – 208 с.
* Левус Є. В. Вступ до програмного забезпечення: навчальний посібник Є. В. Левус, Н. Б. Мельник. - Львів: Видавництво Львівської політехніки, 2018. – 248 с.

**Висновки**

Під час виконання даної лабораторної роботи я ознайомився з життєвим циклом програмного забезпечення, навчився складати власне технічне завдання до розробки програми. Окрім цього спробував себе у ролі UI дизайнера, та створив концепцію інтерфейсу користувача для програми поданої мені в 10 варіанті.