

ПРОГРАМНИЙ ДОДАТОК ДЛЯ ОСВІТНЬОГО ПРОЦЕСУ

Анотація: В процесі роботи було створено додаток для автоматизованого робочого місця «Помічник студента» з переліком усіх необхідних функцій для коректної роботи програми у сфері освіти. Запропоновано ієрархію розв'язуваних задач, розроблено модель предметної області та проаналізовано систему, досліджено послідовність вирішення поставлених задач, складено алгоритм функціонування додатку, головного та спливаючих меню, описано специфікацію процесів, описано діаграму розгортання. Розроблено ER діаграму створюваної бази даних, описано зв'язки для проєктованої системи, обґрунтовано структуру створеної бази даних, описано сутності, проаналізовано запити до бази даних, розглянуто звіти для представлення результатів фільтрації, запропоновано інтерфейс користувача. Результатом роботи є відповідне програмне забезпечення.

Ключові слова: Додаток, автоматизоване робоче місце, ієрархія задач, модель предметної області, алгоритм функціонування, специфікація процесів, діаграма розгортання, ER діаграма, бази даних, сутності, зв'язки, звіти для представлення результатів фільтрації, інтерфейс користувача

Опис проблеми

Створення програмного додатку «Помічник студента» дасть змогу студентам переглядати усю необхідну інформацію для навчального процесу, як то: розклад занять, перелік дисциплін та завдань, інформацію по точках семестрового контролю та лабораторних роботах, прогрес навчання, а також перелік актуальних подій.

Використання додатку «Помічник студента» можливе на одному пристрої одним або декількома користувачами. Серед цільової аудиторії можуть бути, як студенти, так і школярі з викладачами, або можливе навіть використання застосунка в якості персонального щоденника..

Оскільки студентам іноді доволі важко слідкувати за змінами в розкладі, вимогами викладачів, прогресом по дисциплінах та здійснювати ефективне планування свого часу, було поставлене завдання створити програмний застосунок, що допоможе їм в розв'язанні цієї проблеми.

Задача досліджень полягає в розробці програмного додатку.

Функціональність запропонованого додатку є затребуваною на сучасному ринку мобільних додатків, не передбачено розгляд та отримання певних фінансових зисків, але потрібно забезпечення функціональної наповненості, оптимальності виконання

функцій призначення та тестування з практичним втіленням отриманих в освітньому процесі знань і вмінь в реальні програмні застосунки.

Набуття і вдосконалення науково – практичних навичок здійснювалось поетапно [4 – 8], корегуючи та поглиблюючи компетенції та навички.

Метою роботи є створення додатку, що забезпечить вивід інформації про заплановані події, завдання, розклад та таким чином допоможе планувати свою активність протягом навчального року. [2, 3].

Основний зміст і результати роботи

Матеріали і методи дослідження. В першу чергу система орієнтована на роботу із користувачами. На рис. 1 наведено деталізовану діаграму прецедентів. Розробка системи буде здійснюватися під потреби студента і працюватиме з наступними елементами даних:

- **Інформація про заплановані завдання:** тип запланованого завдання, власне, інформація, дедлайн, інформація про час, у який можна захистити це завдання.

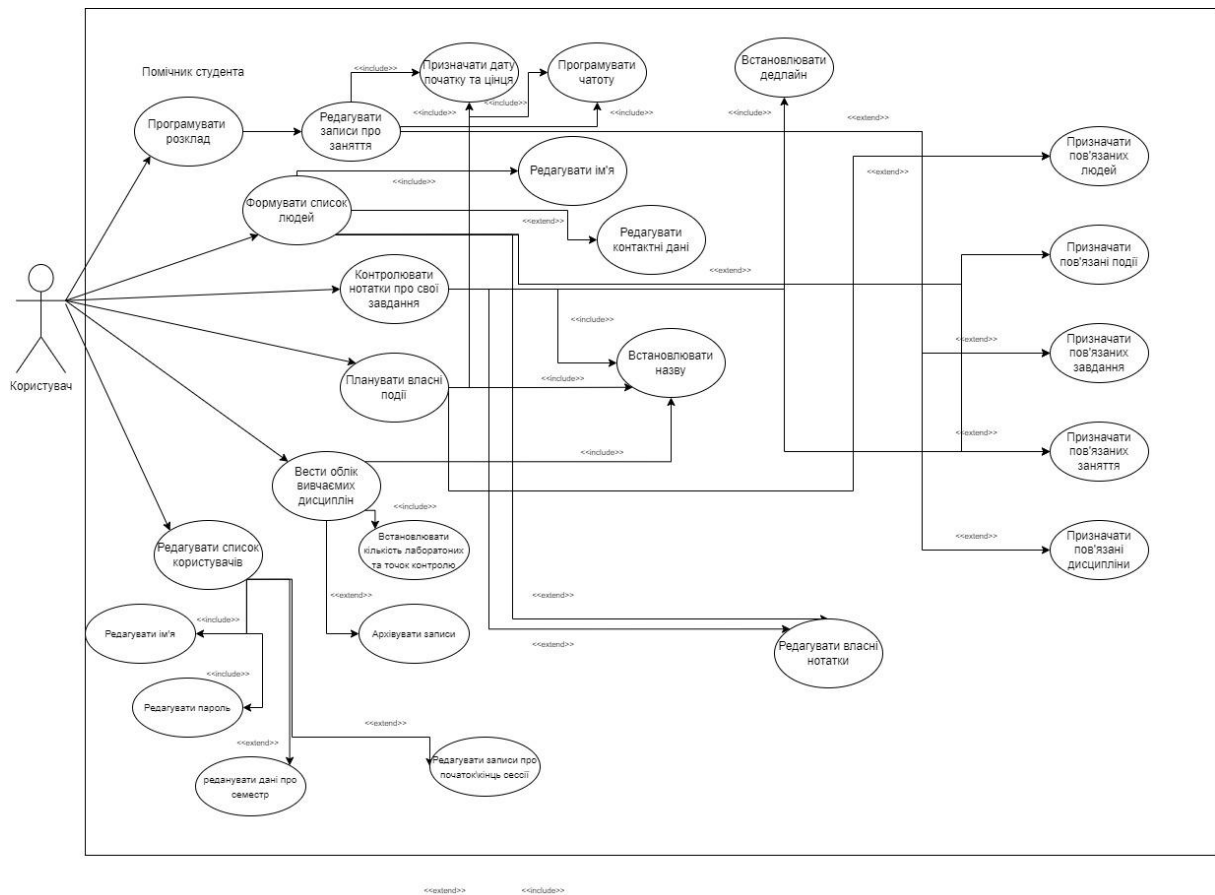


Рис. 1. Деталізована діаграма прецедентів

- **Інформація про заняття:** тип заняття Дата та час початку, дата та час кінця, інформація про викладача або викладачів, інформація про пов'язані завдання, дисципліну, інформація про частоту повторення.

- **Інформація про заплановані події:** назва, опис користувача дата та час початку, дата та час кінця, інформація про пов'язаних людей, дисципліни, завдання, вираз частоти повторень.

- **Інформація про людей:** Повне ім'я нотатки користувача, контактні дані.

- **Інформація про дисципліни:** назва, семестр вивчення, чи є архівованою, якщо так, то з яким підсумковим балом, кількість лабораторних робіт, точок контролю, інформація про користувача що її додав.

- **Інформація про користувачів:** ім'я користувача, пароль, поточний семестр, дата початку та дата кінця сесії.

Саме тому, необхідно перш за все розробити приємний і зрозумілий інтерфейс, для того, щоб користувач зміг правильно реєструватися, переглядати графік лікаря та власну картотеку. Цього можна досягти, лише через діалог з користувачем: при невірних діях користувачу будуть писати повідомлення про помилки [1]. Це необхідно для того, щоб клієнт не витрачав багато часу для ознайомлення з роботою додатку.

Розглянуто звіти для представлення результатів фільтрації. Вибір невиконаних завдань, у яких ще не сплинув час виконання, наведено на рис. 1.

Перш за все необхідно створити базу даних [2], до якої буде входити інформація щодо усіх запланованих подій (на рис. 2 наведено діаграму активності) та дані по користувачах, дисциплінах та викладачах.

Далі формуємо вікно інтерфейсу, що відповідатиме за реєстрацію, по завершенні це діалогове вікно буде перевіряти наявність введених даних в таблиці користувачів у БД і дозволяти або забороняти вхід [3].

Далі на основі введених даних формується головне вікно програми, в якому наявна уся необхідна інформація та є доступ до її редагування, додавання та видалення.

Кожен процес розроблюваного додатку має свої вхідні та вихідні дані, з якими потім будуть взаємодіяти інші процеси, змінювати їх, щоб на виході досягти мети даного додатку (на рис. 3 наведено діаграму послідовностей для будь-якого процесу додавання або редагування). Тобто відбуваються: зміни даних, внесення коректив, доповнення, передача інформації від одного прецедента до іншого.

Діаграму інтеграції моделювання процесів (IDEF) наведено на рис. 4.

На даній діаграмі «дані для заповнення форми» у керуючих структурах першої сутності є списки наявних користувачів, людей, занесених до БД, дисциплін, подій, завдань, занять. Також у такій структурі можуть бути наявні і самі дані для заповнення форми, які треба редагувати.

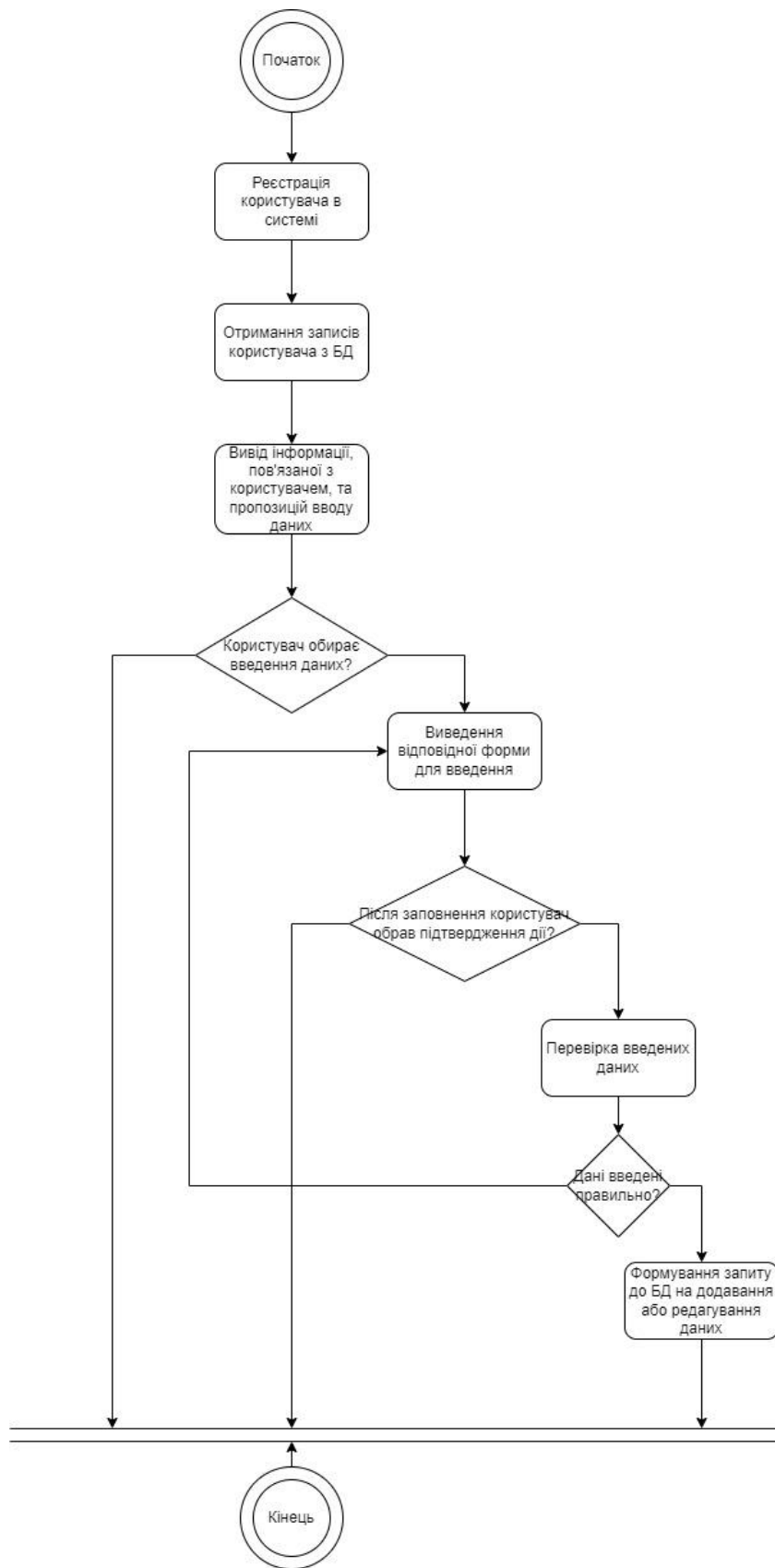


Рис. 2. Діаграма активності

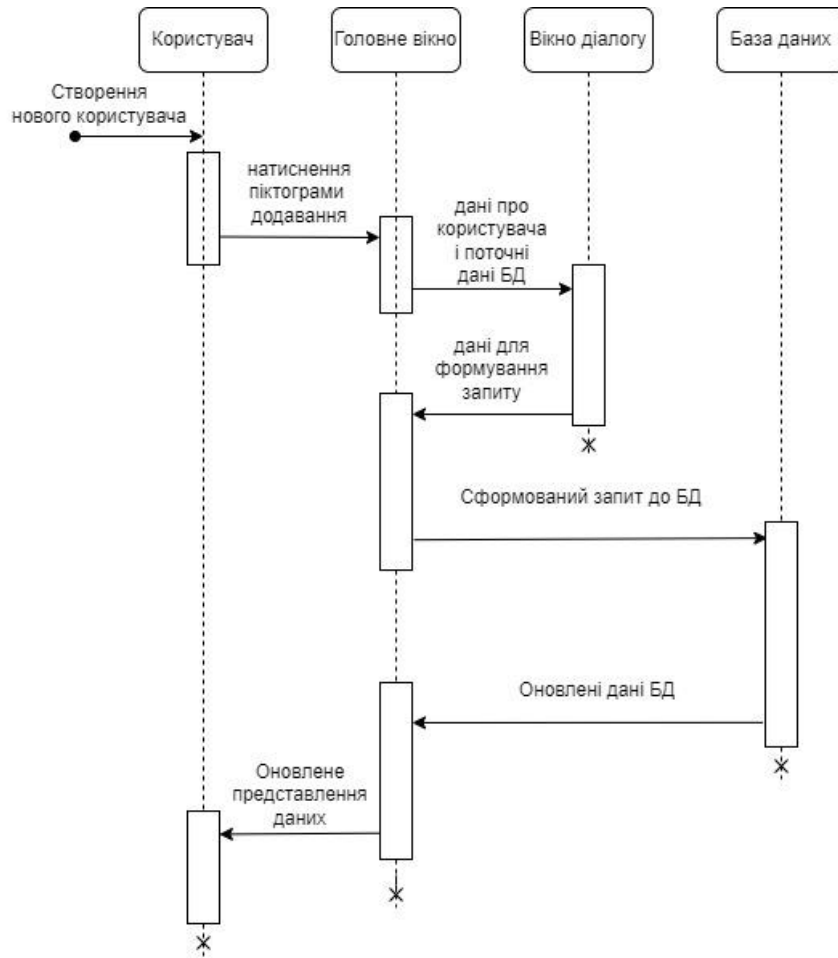


Рис. 3. Діаграма послідовностей для будь-якого процесу додавання або редагування

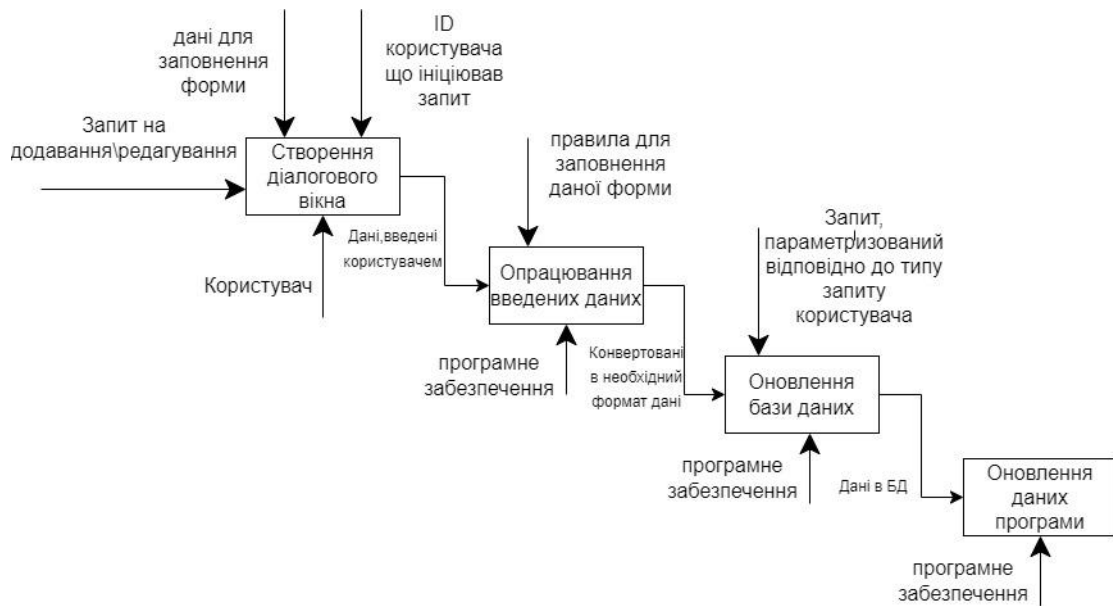


Рис. 4. Діаграма інтеграції моделювання процесів

Діаграму розгортання наведено на рис. 5.



Рис. 5. Діаграма розгортання

Розробляючи інформаційне забезпечення додатку, розглянемо структуру бази даних, та визначимо, що база даних складається з 12 таблиць: users_data – Зберігання даних зареєстрованих користувачів; Discipline – зберігання даних дисципліни; People – зберігання даних людей; Events – зберігання даних подій; Lesson – таблиця занять у розкладі; Task – таблиця завдань; Lesson_types – таблиця-довідник типів занять; Tasks_types – таблиця-довідник типів завдань; Discipline_event – таблиця-коннектор для реалізації зв’язку багато до багатьох між таблицями дисциплін та подій; Event_people – таблиця-коннектор для реалізації зв’язку багато до багатьох між таблицями подій та людей; Lesson_person – таблиця - коннектор для реалізації зв’язку багато до багатьох між таблицями занять та людей; Task_person – таблиця - коннектор для реалізації зв’язку багато до багатьох між таблицями завдань та людей.

Таблиця 1. Опис зв’язків БД

Від	До	Тип зв’язку	Пояснення
Task_types	Lesson	1:N	Один тип – багато записів
Lesson_types	Task	1:N	Один тип – багато записів
task	lesson	1:1	Для кожного завдання є час для його захисту
Lesson	Discipline	1:1	Заняття відбуваються лише з одної дисципліни
Users_data	discipline	1:N	Один користувач може додати багато дисциплін
discipline	event	M:N	Подія може торкатися багатьох дисциплін (конференція з фізики англійською) так і жодної (гітарний вечір у гуртожитку)
Users_data	people	1:N	Один користувач може додати скільки завгодно користувачів(кілька користувачів можуть додати одну і ту саму людину, але для користувача важливі його нотатки для цієї людини)

Від	До	Тип зв'язку	Пояснення
Від	До	Тип зв'язку	Пояснення
people	Event	M:N	Одна людина може бути причетною до багатьох подій і одну подію можуть організовувати кілька людей
people	Task	M:N	До завдання можна призначити як людину, що надала його, так і людину що допомагала або, наприклад, до курсової або диплома може бути причетна комісія з викладачів
people	lesson	M:N	Те саме, але для заняття

На рис. 6 наведено ER-діаграму проектованої системи.

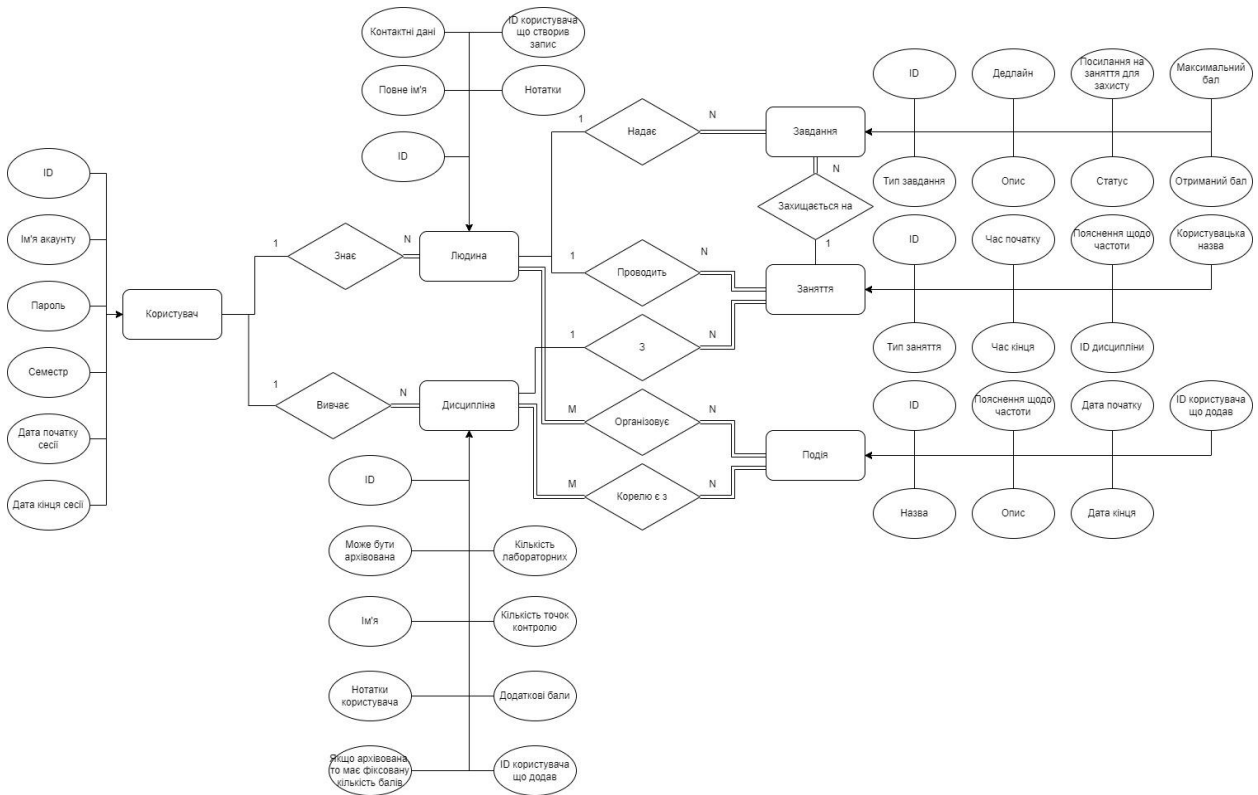


Рис. 6. ER діаграма створеної БД

Структуру створеної БД наведено на рис. 7.

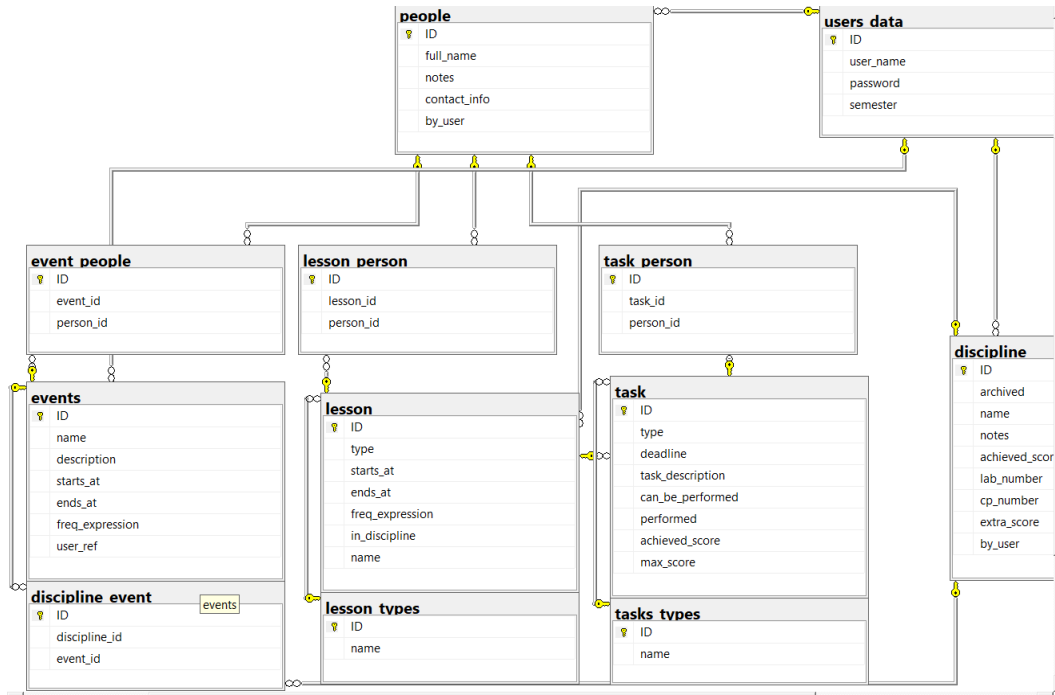


Рис. 7. Структура створеної БД

Описання сутності users_data наведено у табл. 2.

Таблиця 2. Сутність users_data

Ім'я стовця	Тип ключа	NOT NULL	Тип даних	Довжина	Додаткове
ID	PK	+	Int		Identity(1,1)
User_name		+	Varchar	100	
Password		+	Varchar	100	
Semester		-	Int		
Session_starts		-	Datetime		
Session_ends		-	datetime		

Описання сутності discipline наведено у табл. 3.

Таблиця 3. Сутність discipline

Ім'я стовця	Тип ключа	NOT NULL	Тип даних	Довжина	Додаткове
ID	PK	+	Int		Identity(1,1)
Archived		+	Bit		
Name		+	Varchar	100	
Notes		+	Varchar	200	
Achieved_score		+	Float		Default 0
Lab_number		+	Int		Default 0
Cp_number		+	Int		Default 0
Extra_score		+	Float		Default 0
By_user	FK to users_data(ID)	+	Int		

Вибір занесених до БД користувачів наведено на рис. 8 [2, 4].

```
select * from users_data
```

	ID	user_name	password	semester	session_start	session_end
1	1	Natan	1111	5	2022-12-25 00:00:00.000	2022-12-31 00:00:00.000
2	2	Nika	2222	5	2022-12-13 00:00:00.000	2022-12-20 00:00:00.000

Рис. 8. Вибір занесених до БД користувачів

Вибір типів завдань наведено на рис. 9.

```
select * from tasks_types
```

	ID	name
1	3	Дипломна робота
2	8	Збір інформації
3	5	Контрольна
4	2	Курсова робота
5	1	Лабораторна
6	6	Письмове завдання
7	7	Підготовчі заходи
8	4	Реферат

Рис. 9. Вибір типів завдань

Вибір типів занять наведено на рис. 10.

```
select * from lesson_types
```

	ID	name
1	2	Лекція
2	1	Практичне
3	3	Семинар
4	4	Шкільне заняття

Рис. 10. Вибір типів занять

Звіти для представлення вихідної інформації наведено на рис. 11.

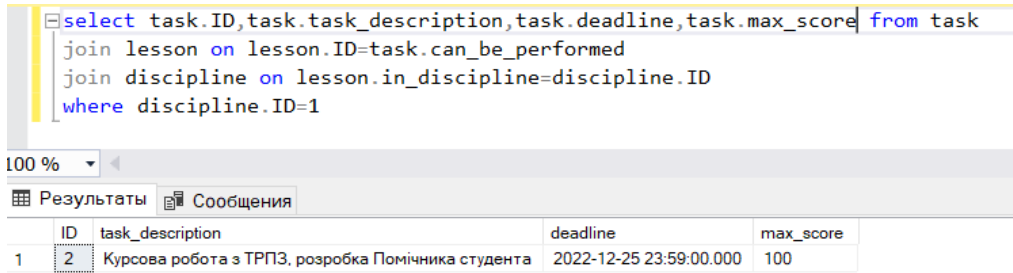


Рис. 11. Вибір завдань з дисципліни з ID 1

Вибір неархівованих дисциплін користувача з ID 1 наведено на рис. 12 [8-11].

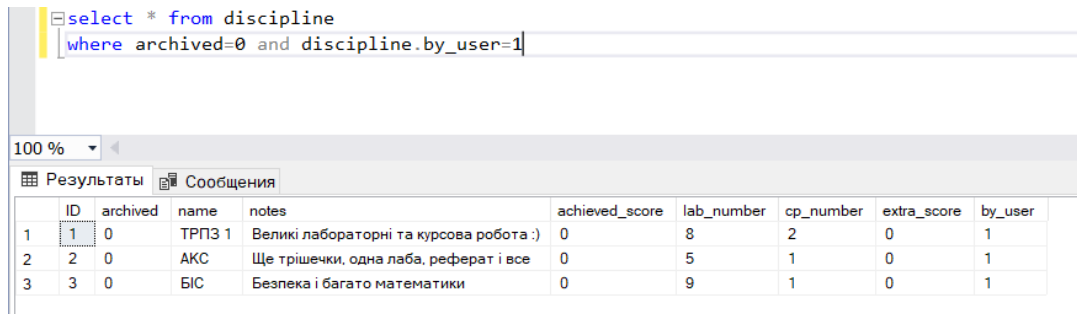


Рис. 12. Вибір неархівованих дисциплін користувача з ID 1

Розглянуто звіти для представлення результатів фільтрації. Вибір невиконаних завдань, у яких ще не сплинув час виконання, наведено на рис. 13.

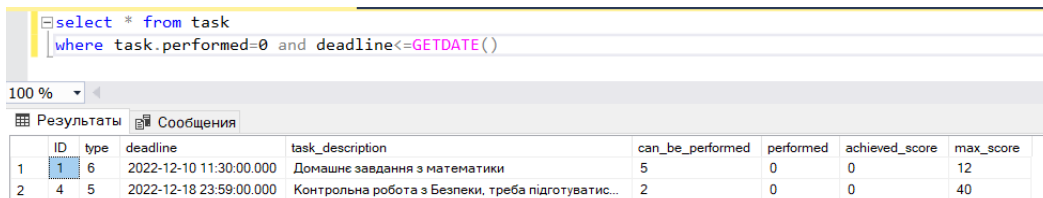


Рис. 13. Вибір невиконаних завдань, у яких ще не сплинув час виконання

Вибір пов'язаних з завданням під індексом 2 людей, наведено на рис. 14. [1, 3]

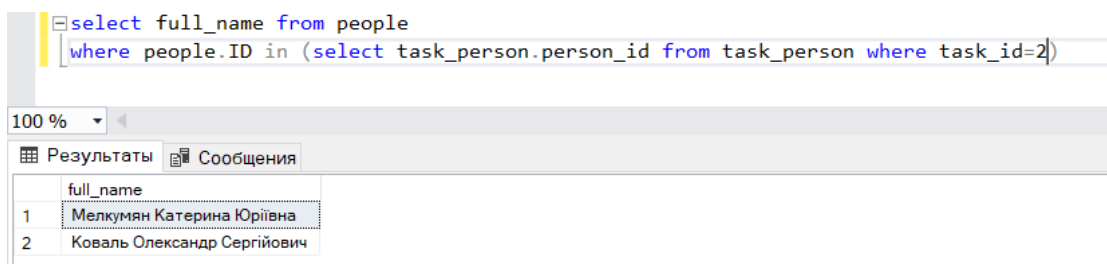


Рис. 14. Вибір пов'язаних з завданням під індексом 2 людей

Інтерфейс користувача був сформований у мінімалістичному стилі. Розробка була спроектована таким чином, аби кожен зміг за короткий час знайти інформацію про усі незакриті завдання, розклад і все інше. Дизайн усіх сторінок аналогічний один одному.

При створенні додатку були використані такі евристики Нільсена [1, 2, 11]:

1. Одноманітність і стандарти. Дотримання однаковості і дотримання стандартів. Всі сторінки у додатку виконані у однаковому стилі, тому користувач розуміє на якому рівні знаходиться і не плутається. Для введення і редагування даних використовуються одні і ті самі форми.

2. Гнучкість і ефективність. Не навантажуюмо досвідчених користувачів зайвою інформацією, надаємо їм можливість здійснювати часто повторювані дії якомога швидше і простіше.

3. Естетичний і мінімалістичний дизайн. Тексти не містять марної або застарілої інформації. Короткі і зрозумілі тексти у поєднанні із мінімалістичним дизайном роблять додаток приємним при використанні.

4. Копії екранних форм.

Головне вікно створеного додатку наведено на рис. 15.

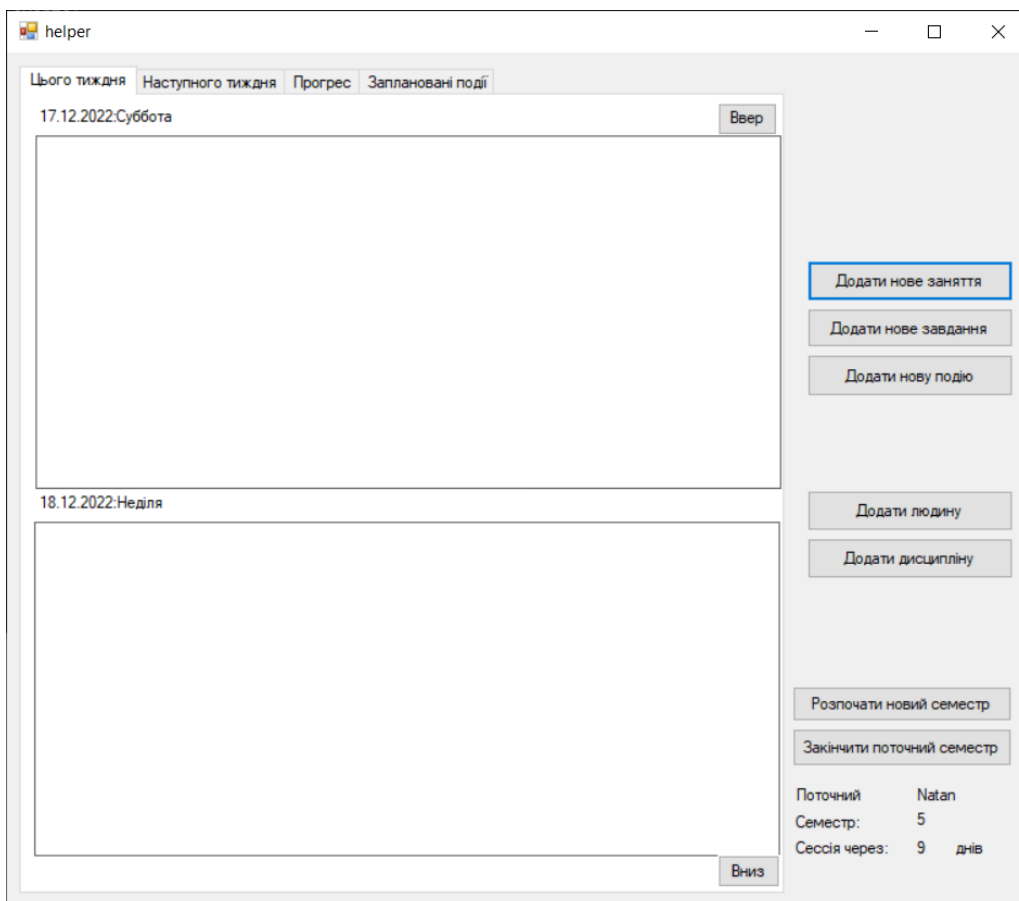


Рис. 15. Вигляд головного вікна

Головний каталог додатку містить наступні файли: Файл проекту *.sln та папку helper з файлами коду проекту: DataStructures.cs – файл з описом структур даних що використовуються при взаємодії з БД; MainForm.cs – опис головного вікна програми і процесів що в ньому відбуваються; MainForm.Designer.cs – файл, в якому розписуються елементи керування головної форми, що не генеруються процедурно; Program.cs – точка входу програми; Progress_report.cs – описуються методи генерації структур, що відображують інформацію за людьми, дисциплінами, подіями; Prompts.cs – опис усіх діалогових форм для отримання даних від користувача; Week_controls.cs – опис парсеру виразів частоти а також структур, що надають інформацію про розклад [4-7].

Інструментарій розробки: Sharpdevelop 4.4 та Microsoft SQL Server Management Studio 18.12.1. Мова: C# + TSQL.

Висновки

Розроблено функціональний додаток для сфери освіти. Запропоновано ієрархію розв'язуваних задач, розроблено модель предметної області та проаналізовано систему, досліджено послідовність вирішення поставлених задач, складено алгоритм функціонування додатку, головного та спливаючих меню, описано специфікацію процесів, описано діаграму розгортання. Розроблено ER діаграму створюваної бази даних та описані відповідні зв'язки між компонентами, обґрунтовано структуру створеної бази даних, описано сутності, проаналізовано запити до бази даних, розглянуто звіти для представлення результатів фільтрації, запропоновано інтерфейс користувача. Результатом роботи є відповідне програмне забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software.* — Addison-Wesley Professional, 1995. — 395.
2. *Мартин Р. Чистий код: створення і рефакторинг за допомогою AGILE.* — Харків: Фабула, 2019. — 416.: іл.
3. *Смолий В.М., Лісовиченко О.І., Борукаєв З.Х. Система підтримки прийняття рішень у радіоелектроніці.,* Київ : Три К, 2020. - 211 с. (Ум. друк. арк. 10,6): ISBN: 978-966-7690-57-1
4. *Смолий В.Н. Подготовка, преобразование и использование данных в процессе производства электронного аппарата // Българско списание за инженерно проектиране, брой 32, април 2017 г. – С.125 – 135.*
5. *Смолий В. Н. Системное моделирование электронных аппаратов и компонентов / В. Н. Смолий // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2015. – № 1(26). – С. 128–136. – Бібліогр.: 8 назв.*
6. *Смолий В. Н. Особенности моделирования электронных аппаратов различного*

назначения и условий эксплуатации / *В. Н. Смолий* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2016. – № 1(28). – С. 116–128. – Бібліогр.: 12 назв.

7. *Смолий В. Н.* Комплексование технических средств производства электронных аппаратов различного назначения и условий эксплуатации / *В. Н. Смолий* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2017. – № 1(30). – С. 147–163. – Бібліогр.: 16 назв.

8. *Smoliy V.* Management conception designer preproduction of electronic vehicles / *V. Smoliy* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2019. – № 1 (34). – С. 113–124. – Бібліогр.: 22 назви.

9. *Смолий В. М.* Модель автоматизованого управління конструкторською підготовкою виробництва електронних апаратів / *В.М. Смолий, Н.В. Смолий* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2022. – № 1(40). – С. 129–133. – Бібліогр.: 5 назв.

10. *Smolij N.* Simulation tools: formal language for cellular automata behavior description / *N. Smolij, O. Lisovychenko, V. Smolij* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2022. – № 2(41). – С. 16–21. – Бібліогр.: 3 назв.

11. *Smolij V.* Graphical shell for constructing user-entered arithmetic functions / *V. Smolij, O. Lisovychenko, N. Smolij* // Адаптивні системи автоматичного управління : міжвідомчий науково-технічний збірник. – 2023. – № 1(42). – С. 115–120. – Бібліогр.: 5 назв.