

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ

Зав. каф. “Комп'ютерні науки”

доц. _____ Сергій ШАРОВ

“17” лютого 2025 р.

Пояснювальна записка

до кваліфікаційної роботи здобувача СВО Магістр
(ступінь вищої освіти)

на тему: «Експертна система для прогнозування результату контрольного
заходу

52/4КНД.11323044.000000ПЗ

Виконав: здобувач вищої освіти 2 курсу, групи 21МБКН
спеціальності 122 Комп'ютерні науки
за ОПП Комп'ютерні науки

(шифр і назва спеціальності та ОПП)

_____ Нікіта ПАВЛЕНКО

(підпис)

Керівник доц. _____ Сергій ШАРОВ

(підпис)

Нормоконтроль, ст.викл. _____ Ольга ЗІНОВ'ЄВА

(підпис)

Рецензент, доц. _____ Ганна ЧЕРНОВА

(підпис)

Запоріжжя – 2025 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет імені Дмитра
Моторного

Факультет: Енергетики і комп'ютерних технологій

Кафедра: Комп'ютерні науки

Ступінь вищої освіти: Магістр

Спеціальність: 122 Комп'ютерні науки

ОПП: Комп'ютерні науки _____

ЗАТВЕРДЖУЮ

Завідувач кафедри_КН

к.пед.н., доц. _____ Сергій ШАРОВ

“04” _жовтня_ 2024_ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Павленку Нікіті Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Експертна система для прогнозування результату контрольного заходу»

керівник проєкту Шаров Сергій Володимирович, к.пед.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “23” вересня 2024 року № 442-С 2.

Строк подання здобувачем вищої освіти роботи 14 лютого 2025

3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проєктування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)_____

1. Опис предметної області та аналіз існуючих рішень.

2. Специфікація вимог до експертної системи.

3. Вибір та обґрунтування технологій розробки експертної системи.

4. Дослідна експлуатація експертної системи.

5. Тестування експертної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)
Діаграма використання, діаграми IDEF0, розкадровка інтерфейсу, структурна схема, візуальні компоненти інтерфейсу, алгоритм роботи, інтерфейс програми

Презентація – 11 слайдів

1. Титульний слайд

2. Предмет, об'єкт, мета

3. Завдання дослідження

- 4. Порівняльна характеристика аналогів програмного продукту
- 5. Діаграма варіантів використання
- 6. Інструментальні засоби
- 7. Інтерфейс і робота експертної системи
- 8. Інтерфейс і робота експертної системи
- 9. Тестування
- 10. Висновок
- 11. Дякую за увагу

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 01 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	12.09.2024	
2	Специфікація вимог до експертної системи	06.10.2024	
3	Проектування експертної системи	16.10.2024	
4	Обґрунтування інструментальних засобів	05.11.2024	
5	Розробка експертної системи	10.12.2024	
6	Тестування та налагодження експертної системи	20.01.2025	
7	Підпис керівником роботи	14.02.2025	
8	Підпис завідувачем кафедри	17.02.2025	

Здобувач вищої освіти

Нікіта ПАВЛЕНКО
(власне ім'я та прізвище)

Керівник кваліфікаційної роботи

Сергій ШАРОВ
(власне ім'я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра: 90 с., 30 рис., 8 табл., 46 посилань, 3 додатки.

Актуальність. На сьогодні сфера застосування інформаційно-комунікаційних технологій (ІКТ) є доволі широкою та різноманітною. Вона охоплює широкий спектр завдань – від створення електронних документів до складного системного аналізу та підтримки прийняття рішень. Важливим етапом розвитку ІКТ стало створення й використання експертних систем (ЕС) у таких галузях, як економіка, виробництво, державне управління. Одним з напрямків використання експертних систем є освітня діяльність, де вони можуть виконувати окремі функції викладачів, надавати експертні рішення у межах певної дисципліни. Експертні системи мають значну перевагу у здатності приймати рішення в ситуаціях, де відсутній чіткий попередньо сформований алгоритм.

Об'єкт дослідження: прикладне програмне забезпечення для прогнозування контрольних заходів.

Предмет дослідження: експертна система для прогнозування результату контрольного заходу.

Мета роботи: розробити експертну систему для прогнозування контрольного заходу.

Задачі дослідження:

- проаналізувати наявні програмні рішення для управління навчання;
- розглянути інструментальні засоби для розробки експертної системи;
- створити технічне завдання на розробку експертної системи прогнозування контрольного заходу;
- на основі складеного технічного завдання розробити експертну систему, скласти інструкцію користувача та виконати тестування.

Практичне значення дослідження полягає в розробці експертній системи, яка дозволить отримувати підготовленість студента до контрольного заходу.

Ключові слова: БД, ЕС, ПЗ, JSON.

SUMMARY

Qualification work of master's degree: 90 p., 30 figures, 8 tables, 46 references, 3 appendices.

Relevance. Today, the scope of information and communication technologies (ICT) is quite wide and diverse. It covers a wide range of tasks - from the creation of electronic documents to complex system analysis and decision support. An important stage in the development of ICTs was the creation and use of expert systems (ES) in such industries as economics, manufacturing, and public administration. One of the areas where expert systems are used is in education, where they can perform certain functions of teachers and provide expert solutions within a particular discipline. Expert systems have a significant advantage in their ability to make decisions in situations where there is no clear pre-formed algorithm.

Object of research: application software for forecasting control measures.

Subject of research: an expert system for forecasting control measures.

Purpose of the study: to develop an expert system for forecasting a control measure.

Research objectives:

- analyse the existing software solutions for learning management;
- to consider the tools for developing an expert system;
- to create a technical task for the development of an expert system for forecasting a control event;
- to develop an expert system based on the terms of reference, to create a user manual and to perform testing.

The practical significance of the study is to develop an expert system that will allow to obtain the student's readiness for the control event.

Keywords: DATABASE, ES, SOFTWARE, JSON.

ЗМІСТ

ВСТУП	9
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	12
1.1 Особливості використання експертних систем в освіті.....	12
1.2 Аналіз існуючих рішень	16
1.3 Технічне завдання на розробку експертної системи для прогнозування результату контрольного заходу	26
2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ЕКСПЕРТНОЇ СИСТЕМИ	33
2.1 Концептуальна модель використання експертної системи прогнозування контрольних заходів	33
2.2 Функціональна модель експертної системи прогнозування контрольних заходів	37
2.3 Специфікація функціональних та нефункціональних вимог	39
3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ	42
3.1 Загальний огляд технологічного стеку	42
3.2 Обґрунтування вибору мови програмування	48
3.3 Інструменти для розробки.....	54
4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ	60
4.1 Особливості програмної реалізації експертної системи	60
4.2 Функціональні можливості та інструкція користувача експертної системи	70
4.3 Тестування експертної системи.....	75
ВИСНОВКИ.....	84
СПИСОК ЛІТЕРАТУРИ.....	86
ДОДАТОК А (Лістинг коду програми)	91
ДОДАТОК Б (Стаття)	97
ДОДАТОК В (Сертифікат).....	100

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

ЕС – експертна система.

ПЗ – програмне забезпечення.

JSON – JavaScript Object Notation.

ВСТУП

Актуальність. На сьогодні сфера застосування інформаційно-комунікаційних технологій (ІКТ) є доволі широкою та різноманітною. Вона охоплює широкий спектр завдань – від створення електронних документів до складного системного аналізу та підтримки прийняття рішень. Важливим етапом розвитку ІКТ стало створення й використання експертних систем (ЕС) у таких галузях, як економіка, виробництво, державне управління. Одним з напрямків використання експертних систем є освітня діяльність, де вони можуть виконувати окремі функції викладачів, надавати експертні рішення у межах певної дисципліни. Експертні системи мають значну перевагу у здатності приймати рішення в ситуаціях, де відсутній чіткий попередньо сформований алгоритм.

Експертні системи є потужним інструментом у сучасній освіті, що сприяють індивідуалізації навчального процесу, підвищенню ефективності викладання та полегшенню управління освітніми завданнями. Вони дозволяють аналізувати знання учнів, адаптувати матеріали до їхніх потреб, автоматизувати оцінювання та забезпечувати миттєвий зворотний зв'язок. Крім того, такі системи активно використовуються в спеціалізованих галузях, як-от медицина чи інженерія, завдяки можливостям симуляції складних процесів. Вони також допомагають у дистанційному навчанні, створюючи інтерактивне середовище, де здобувачі освіти отримують підтримку від віртуальних наставників. Попри виклики, такі як вартість впровадження та питання етичності, експертні системи мають значний потенціал для розвитку освіти та створення нових можливостей для навчання.

У зв'язку з необхідністю автоматизації процесу прогнозування контрольних заходів та оптимізації ухвалення рішень, тему дослідження було сформульовано як **«Експертна система для прогнозування результату контрольного заходу»**.

Об'єкт: прикладне програмне забезпечення для прогнозування контрольних заходів.

Предмет: експертна система для прогнозування результату контрольного заходу.

Мета: розробити експертну систему для прогнозування контрольного заходу.

Сформована мета, об'єкт і предмет дослідження дозволили поставити такі **завдання**:

1. Проаналізувати наявні програмні рішення для управління навчання.
2. Розглянути інструментальні засоби для розробки експертної системи.
3. Створити технічне завдання на розробку експертної системи прогнозування контрольного заходу.

4. На основі складеного технічного завдання розробити експертну систему, скласти інструкцію користувача та виконати тестування.

При реалізації поставлених задач було використано наступні **методи**: теоретичні: аналіз наукової та спеціальної літератури з експертних систем, освітнього процесу, програмування, практичні: програмна реалізація експертної системи.

Практичне значення дослідження полягає в розробці експертній системи, яка дозволить отримувати підготовленість студента до контрольного заходу.

Апробація результатів здійснювалася шляхом опублікування статті «Особливості використання експертних систем в освіті» у збірнику матеріалів Всеукраїнської науково-практичної конференції «Сучасні комп'ютерні та інформаційні системи та технології» (м. Запоріжжя, 2024).

Структура і обсяг роботи. Дипломна робота складається з переліку умовних скорочень, вступу, чотирьох розділів, загальних висновків, списку використаних джерел (46 найменувань). Перший розділ присвячений аналізу особливостей використання експертних систем в освіті, аналізу існуючих рішень та складання технічного завдання. Другий розділ присвячений

побудови концептуальної і функціональних моделей та специфікацій функціональних та нефункціональних вимог. Третій розділ присвячено написанню загального огляду технологічного стеку, обґрунтування вибору мови програмування та вибору інструментів розробки. Четвертий розділ присвячено написанню особливостей програмної реалізації, функціональних можливостей та інструкції користувача експертної системи та її тестуванню. Загальний обсяг роботи складає 90 сторінок.

РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Особливості використання експертних систем в освіті

На сьогодні сфера застосування інформаційно-комунікаційних технологій (ІКТ) є доволі широкою та різноманітною. Вона охоплює широкий спектр завдань – від створення електронних документів до складного системного аналізу та підтримки прийняття рішень. Важливим етапом розвитку ІКТ стало створення й використання експертних систем (ЕС) у таких галузях, як економіка, виробництво, державне управління. Одним з напрямків використання експертних систем є освітня діяльність, де вони можуть виконувати окремі функції викладачів, надавати експертні рішення у межах певної дисципліни.

Під поняттям «Експертна система» розуміється програмний засіб, що виконує функції експерта при вирішенні задач в конкретній предметній галузі. За своєю суттю вони є одним з типів інтелектуальних систем та призначені для генерування правильних та ефективних рішень у конкретній предметній галузі [25, с.74]. Експертні системи мають значну перевагу у здатності приймати рішення в ситуаціях, де відсутній чіткий попередньо сформований алгоритм. В свою чергу, експертні рішення формуються в умовах невизначеності, неповноти чи неоднозначності вихідної інформації, на основі структурованих даних, які надаються у вигляді умовиводів (правил прийняття рішень), що зберігаються у базі знань [34, с. 10].

Характерною особливістю ЕС є здатність до генерування нових знань на основі вже відомих, вирішення практичних завдань та пояснення логіки отриманих рішень. На основі знань, отриманих від експертів, система формує відповідні рішення, які можуть містити ідентифікацію ситуації, формування причини, прогнозування або рекомендації для вибору дій у конкретній

предметній галузі [21, с. 254]. При цьому вдалі рішення накопичуються в системі та використовуються для підвищення ефективності майбутніх висновків.

До основних переваг ЕС можна віднести наступні:

- створення умов для розповсюдження людського досвіду та забезпечення експертної підтримки навіть за відсутності експерта.
- забезпечення можливості одночасного аналізу значного обсягу інформації.
- унеможливлення упередженості та суб'єктивного впливу користувача на формування експертних рішень.
- модульна структура ЕС дозволяє легко адаптувати систему до змін.
- створення потужного інструментарію для постановки складних запитань та пошуку ефективних рішень з можливістю подальшої інтеграції з іншими програмними засобами, системами управління базами даних тощо.
- виконання окремих функцій людини, завдяки чому відбувається заощадження часу на виконання інших завдань [28, с.29].

Експертні системи можна поділити на декілька типів в залежності від призначення. Наприклад, наступним етапом розвитку традиційних інформаційних систем є інформаційна експертна система (ІЕС) – це сукупність методів і засобів організації, накопичення та застосування інформаційних ресурсів і знань для вирішення складних задач у конкретній галузі. Однак, як і будь-які технології, ІЕС мають певні обмеження. Їхні основні недоліки пов'язані з високими фінансовими витратами на розробку та наповнення бази знань. Крім того, ІЕС недостатньо ефективні для розв'язання нестандартних завдань та формування нових концепцій чи правил [34, с. 10].

Штучний інтелект активно використовується в інтелектуальних інформаційних системах (ІС), які представляють собою комплекс програмних, лінгвістичних, логіко-математичних засобів тощо. Такі системи забезпечують обробку, накопичення й використання великих обсягів даних, автоматизують процес прийняття управлінських рішень на основі аналізу поточних та

попередніх станів системи. Їхня ключова особливість полягає в акумуляції складних технологічних рішень, які сприяють автоматизації обробки інформації та розробці альтернативних варіантів вирішення поставлених задач [10, с. 307].

Використання технологій штучного інтелекту (ШІ) в освітній діяльності відкриває нові форми, засоби й підходи до організації освітнього процесу. Інтеграція різних підходів до моделювання знань та інтелектуалізація програмного забезпечення навчального призначення надає можливість підвищити якість навчання за рахунок автоматизації процесу навчання та формування певних експертних рішень та рекомендацій. Це досягається завдяки інтеграції в базу знань експертної системи додаткової довідкової інформації, яка допомагає формувати відповіді на запитання здобувачів освіти [28, с. 29]. Крім того, оскільки технології ШІ базуються на моделюванні досягнутого здобувачем вищої освіти рівня знань, це дозволяє забезпечити більш точну й адаптивну оцінку його навчальних досягнень [25, с. 74].

Одним із перспективних напрямів використання штучного інтелекту в дистанційному навчанні є створення експертно-навчаючих систем (ЕНС). Під цим поняттям розуміються програмні засоби, які об'єднують ключові компоненти експертних систем і орієнтовані на навчальні завдання. ЕНС базуються на використанні комплексу методів (інтелектуальних, педагогічних, дидактичних та ін.), що дозволяє моделювати поведінку вчителя або викладача [10, с. 307].

У системі дистанційної освіти впровадження таких систем відкриває нові можливості для адаптивного навчання, коли під час формування блоку навчальних матеріалів для вивчення враховуються індивідуальні здібності здобувачів освіти. Завдяки використанню адаптивних алгоритмів і програм, такі системи стимулюють творче мислення здобувачів освіти і посилюють значущість їхньої самостійної роботи. У роботі [24, с. 268] зазначається, що такі системи структурно можуть містити основний розділ (зазвичай він містить навчально-методичні матеріали) та розділ інтелектуального керування,

що забезпечує адаптивне управління освітнім процесом. Зміст навчально-методичних матеріалів включає не лише знання експертів у певній предметній галузі, але й методику їх подання, забезпечуючи інтеграцію навчальних і методичних знань у єдине інформаційне середовище [6, с. 144].

Слід зазначити, що процес вибору педагогічних принципів і способів реалізації ЕНС не має формалізованого підходу. Однак під час проєктування та створення таких систем доцільно спиратися на педагогічні принципи, ефективність яких доведено у розробці інформаційних технологій навчання. Це дозволить забезпечити ефективне управління навчанням і підвищити якість освіти, адаптуючи навчальні матеріали до індивідуальних потреб кожного здобувача освіти. У роботі [31, с. 39] вказується на таких принципах як: принцип центрування на особистості, принцип пріоритетності педагогічного підходу при проєктуванні освітнього процесу з використанням ЕНС, принцип педагогічної доцільності застосування нових інформаційних технологій навчання, принцип свободи вибору пізнавальної траєкторії та унікальності індивідуальної пізнавальної діяльності здобувачів освіти, принцип різноманітності, принцип відкритості та розвитку ЕНС, принцип професійної спрямованості, принцип «прецеденту» або «випадку» (цей принцип покладено в основу кейс-методу), принцип індивідуальної комфортності та емоційної рівноваги, принцип розподіленості навчального матеріалу.

Слід додати, що одним з напрямків використання ІКТ в освітньому процесі є оцінювання навчальних досягнень здобувачів вищої освіти. Систематична перевірка знань здобувачів освіти є ключовим елементом ефективної організації освітнього процесу. Завдяки використанню ІКТ оцінювання рівня навчальних відбувається з достатньо високою деталізацією та в короткий термін за рахунок автоматичної обробки інформації [13, с. 86]. Крім того, контроль знань та сформованих компетентностей можна здійснювати під час поточних занять, контрольних заходів тощо. Вибір методів контролю залежить від конкретних завдань, що ставляться викладачем. Важливо враховувати, що ці методи можуть використовуватися у різних

формах контролю (тестування, відкриті відповіді та ін.). Кожен з методів свої переваги, недоліки та особливості використання, що дозволяє адаптувати їх до специфіки навчальних цілей окремої дисципліни [9, с. 61].

Систематичний контроль знань можна організовувати за допомогою відповідного програмного забезпечення, зокрема експертних систем. Це дозволить не тільки накопичувати інформацію про результати оцінювання, а й сформувати певні рекомендації для конкретних здобувачів освіти, проаналізувати успішність академічної групи з конкретної дисципліни тощо. Крім того, важливого значення набуває прогнозування результатів контрольних заходів, що може впливати на змістове наповнення контрольних заходів, вибір форм здійснення контролю тощо. У подальших дослідженнях ми плануємо розглянути можливість розробки експертної системи для здійснення прогнозування результатів контрольного заходу.

1.2 Аналіз існуючих рішень

Останнім часом актуальність розробки експертних систем для прогнозування контрольних заходів стрімко зростає через необхідність підвищення ефективності прийняття рішень у різних сферах діяльності. Сфера застосування інформаційно-комунікаційних технологій (ІКТ) охоплює широкий спектр завдань – від автоматизації бізнес-процесів до підтримки стратегічного планування. Важливим кроком у розвитку ІКТ стало створення експертних систем (ЕС), які успішно застосовуються в економіці, виробництві, державному управлінні та освіті.

У контексті прогнозування контрольних заходів експертні системи можуть забезпечувати підтримку прийняття рішень на основі аналізу великих обсягів даних, моделювання ризиків та оцінки можливих сценаріїв розвитку подій. В освітній діяльності такі системи здатні допомагати планувати та оцінювати результати навчального процесу, прогнозувати успішність учнів,

оптимізувати навчальні плани та розклад контрольних заходів. Це дозволяє не лише зменшити навантаження на викладачів, але й підвищити об'єктивність оцінювання та ефективність навчання [2, с. 14].

Основою сучасного процесу розвитку суспільства виступає інформатизація освіти. Варто зазначити, що цей процес повинен випереджати інформатизацію інших сфер суспільної діяльності, оскільки саме в освіті формуються соціальні, психологічні та професійні основи для розвитку суспільства [2, с. 14].

На сьогодні інформаційні технології займають провідне місце у системі освітнього процесу. Зокрема, дослідження успішності здобувачів освіти тісно пов'язані з діяльністю закладів вищої освіти, що дає змогу здійснювати детальний моніторинг успішності та рейтингу студентів, а також аналізувати ефективність засвоєння навчального матеріалу з дисциплін [2, с. 14].

Результати моніторингу дозволяють визначити дисципліни, де спостерігається низький рівень засвоєння матеріалу, що зумовлює необхідність проведення додаткових консультацій для студентів. Крім того, забезпечується зворотний зв'язок, який включає надання інформації про результати навчання та досягнення [2, с. 14].

Кількість студентів, які успішно проходять іспити, є дуже низькою. Отже, важливо, щоб викладач міг якнайшвидше виявити студентів, які мають високі шанси на успіх або ризикують зазнати невдачі, щоб відповідно коригувати свою підтримку під час навчання [40, с. 293-294]. Окрім особистої думки викладача, інструмент, який міг би робити прогнози щодо ймовірності успіху студента на національних іспитах, був би значною підтримкою. Це також корисно для керівників шкіл і розробників навчальних планів [40, с. 294]. Попри прийнятні результати, можливості системи можуть бути розширені завдяки використанню більш складних методів класифікації, таких як нечіткі правила, нейронні мережі або гібридні підходи [40, с. 298]

Результати студентів мають велике значення для закладів вищої освіти. Прогнозування академічної успішності студентів є важливим для кожного

освітнього закладу, який прагне підвищити успішність студентів і зменшити рівень невдач у майбутньому [41, с. 1]. Дослідження результатів студентів у комп'ютерному відділі виявило високий відсоток провалів у курсі статистики для студентів першого рівня, який у деякі роки досягав 48.06% [41, с. 12].

Оцінювання якості екзаменаційних робіт університету є важливим для здобуття інституційної та програмної акредитації. Наразі екзаменаційні роботи оцінюються вручну, що може бути виснажливим, тривалим і в деяких випадках непослідовним процесом [33, с. 2]. Через зростаючий інтерес до процесів освітнього оцінювання та студентського оцінювання вищі навчальні заклади стикаються з необхідністю забезпечення якісного складання та оцінювання екзаменаційних робіт, які мають відповідати визначеним стандартам [33, с. 3].

Багато університетів використовують ІКТ-платформи для обчислення результатів студентів. Ці системи не враховують особливості окремих факультетів і департаментів, що призводить до помилок у підрахунках та можливих маніпуляцій [37, с. 9-10]. Деякі університети створюють алгоритми під свої потреби, але вони залишаються фіксованими та не можуть адаптуватися до змін у правилах і політиках різних закладів [37, с. 11-12]. Покладання виключно на ІКТ-відділи створює ризики маніпуляції даними, які можна зменшити шляхом децентралізації процесу обчислення через експертну систему [37, с. 12-13]. Використання експертних систем дозволяє формувати логічні правила, базуючись на встановлених нормативах університетів та національних стандартів, що забезпечує об'єктивність обчислень [37, с. 13].

Програмне забезпечення для оцінювання Gradescope дозволяє студентам отримувати більш швидкі та детальні відгуки про свої роботи, а викладачам - бачити детальну аналітику завдань і питань. Це простий спосіб приймати роботи в цифровому форматі, щоб зберегти оригінал роботи та забезпечити швидкий і легкий перегляд з будь-якого місця [19].

Coursera є освітньою компанією, яка пропонує онлайн - курси, які відомі як MOOCs - Масові відкриті онлайн-курси – це інтернет-курси з

великомасштабною інтерактивною участю та відкритим доступом через інтернет (рис. 1.1).

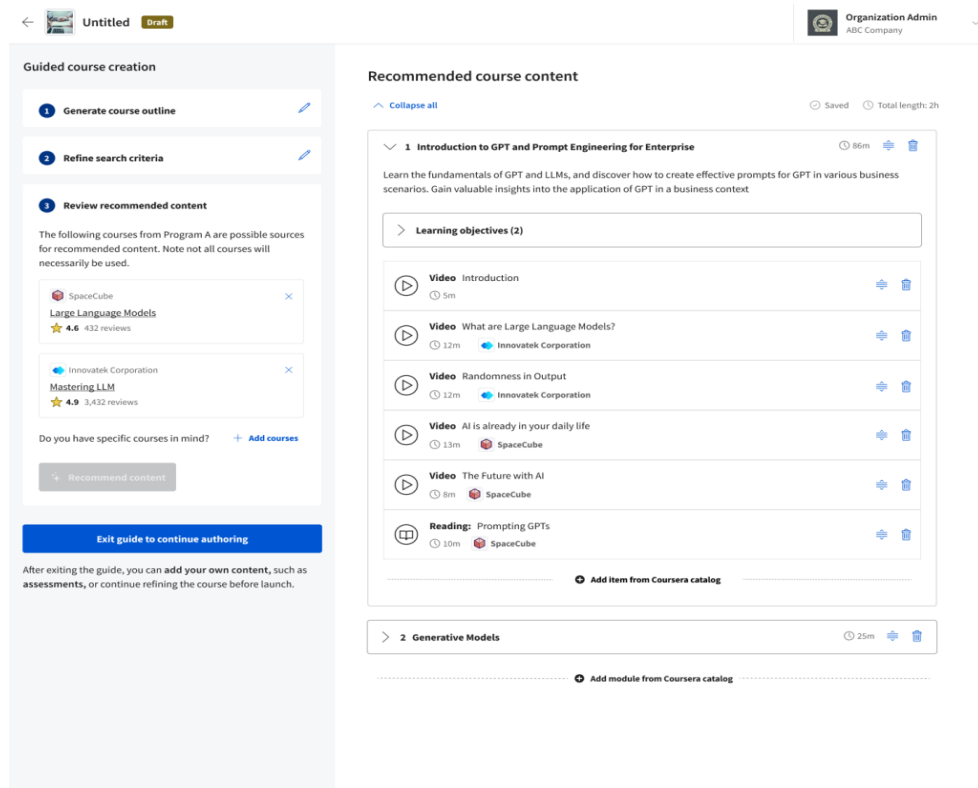


Рисунок 1.1 – Вигляд сайту Coursera

Переважна більшість онлайн-курсів є на англійській мові з субтитрами на кількох мовах. Coursera працює в партнерстві з декількома університетами і пропонує такі курси, як фізика, машинобудування, медицина, біологія, математика, бізнес, інформатика тощо [19].

Coursera була заснована в 2012 році двома професорами комп'ютерних наук Стенфордського університету і в даний час є однією з провідних компаній МООС. Замість створення власного освітнього контенту, Coursera співпрацює з кращими університетами світу та іншими організаціями, щоб зробити курси доступними. Завдяки таким партнерським зв'язкам Coursera пропонує широкий спектр курсів в області бізнесу, науки, інформаційних технологій і тощо. Ці заняття зазвичай проводяться за розкладом, а деякі курси вимагають групового співробітництва, всі вони проводяться в режимі онлайн

[19]. В даний час практично будь-яка людина без знань або досвіду може пройти відмінні курси через курси Coursera. Деякий час назад освоєння нової професії було важким і дорогим. Цей новий, ефективний і економічний спосіб навчання стає все більш доступним через Інтернет. Курси Coursera також можна проходити через мобільний пристрій. Навчання на Coursera є асинхронним, тобто студенти можуть переглядати лекції чи робити домашні завдання в різний, зручний для них час, на відміну від синхронних курсів, в яких студенти та викладач взаємодіють одночасно у чітко визначеному вузькому часовому проміжку [19].

Кожен онлайн курс на Coursera містить інформацію про проект, викладачів, програму курсу, рецензії на курс, параметри реєстрації та відповіді на часті запитання. Після успішного проходження курсу користувачі можуть отримати сертифікат. Вони можуть скористатися функцією «Поділитись» та повідомити про свої успіхи друзям, родині, зацікавленим роботодавцям. Також користувачі Coursera мають можливість навчатися за навчальними програмами бакалавра чи магістра, після чого отримують відповідний диплом [45, с. 118].

Платформа Open edX – це система керування навчанням (LMS) із відкритим вихідним кодом, яка дає змогу організаціям у всьому світі створювати індивідуальні та захоплюючі платформи онлайн-навчання. Платформа Open edX надає масово масштабовану програмну технологію навчання, що лежить в основі edX (рис. 1.2) [32].

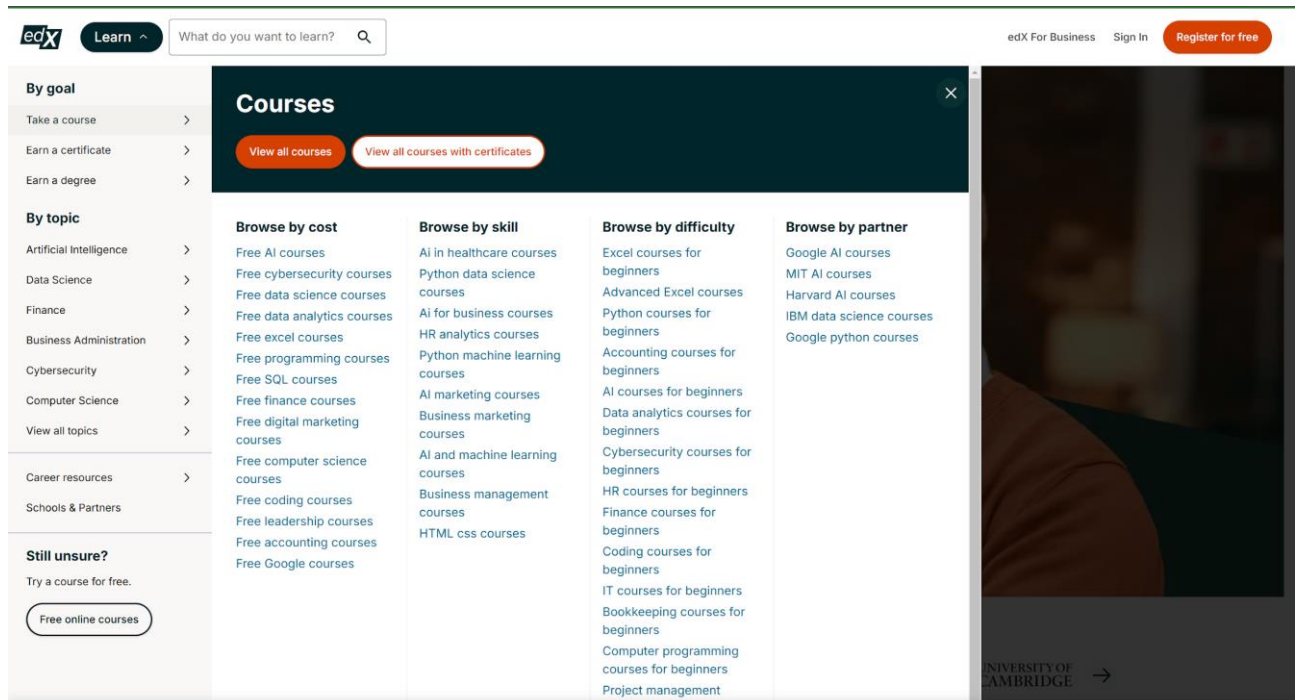


Рисунок 1.2 – Вигляд сайту платформи edX

Програмне забезпечення Open edX підтримує кілька платформ електронного навчання, пропонуючи корисні функції, які роблять онлайн-навчання набагато доступнішим і зручнішим. Оскільки він має відкритий код, він постійно розвивається та вдосконалює свої пропозиції за допомогою надійної групи технологічних партнерів. Гарвардський університет і MIT створили проект Open edX для відомої навчальної платформи edX, а Google і Microsoft швидко підтримали його (рис. 1.3) [32].

1. Масштабованість. Платформа Open edX описує себе як безкоштовну систему керування курсами з відкритим вихідним кодом, яка використовується для розміщення масових відкритих онлайн-курсів у всьому світі, а також для розміщення менших навчальних модулів. Він розроблений для самостійного навчання, а також може обслуговувати кількох учнів на платформі одночасно. Можливість приймати та керувати багатьма учнями допомогла команді Open edX налагодити контакти з університетами, корпораціями та державними установами.

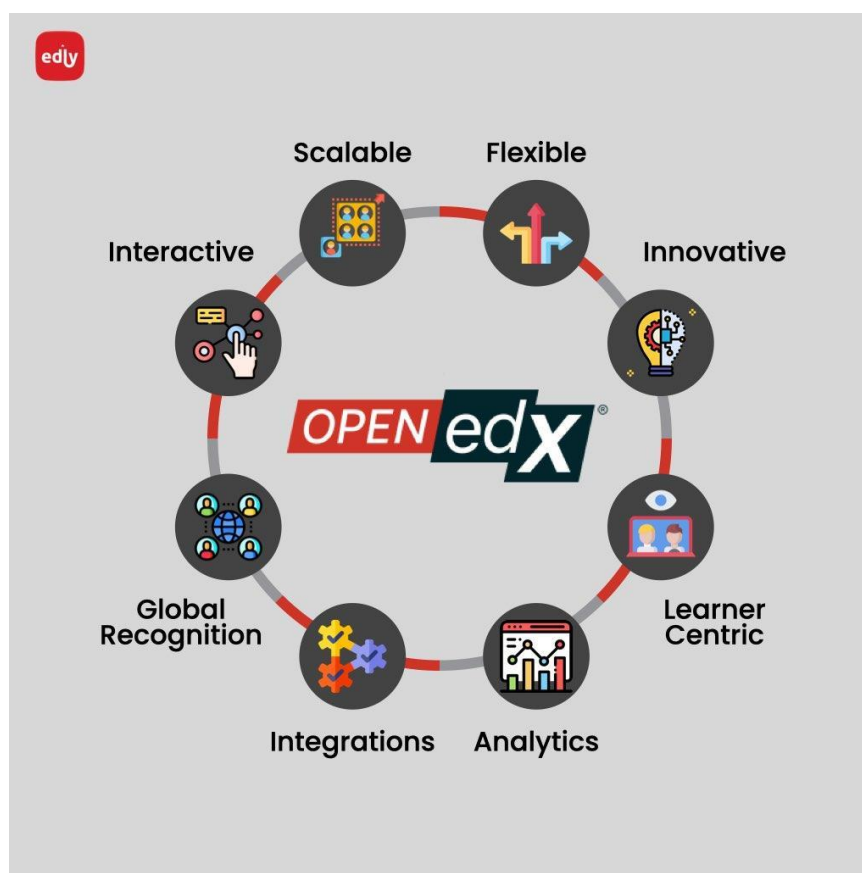


Рисунок 1.3 – Унікальність платформи Open edX

2. Інтерактивний і гнучкий. На відміну від інших платформ електронного навчання, платформа Open edX може вийти за рамки простої взаємодії в онлайн-класі за допомогою плагінів xBlock. Ці плагіни XBlock надають користувачам Open edX величезну гнучкість. Автори курсів, адміністратори та персонал високо цінують їх, оскільки XBlocks покращують їхній досвід авторства та полегшують розробку новаторської платформи електронного навчання.

3. Потужні інструменти та інтеграції. Спільнота Open edX ділиться корисною інформацією на передові теми, які приваблюють вищі навчальні заклади та університети. Інструменти Open edX LTI можна використовувати для зв'язку із зовнішнім програмним забезпеченням; крім того, API, що використовуються програмним забезпеченням Open edX, ще більше покращують роботу користувача.

4. Глобальне визнання та бренд. Платформа Open edX визнана в усьому світі завдяки таким престижним закладам, як Гарвард і MIT. Він має величезну базу користувачів, і великі гіганти, такі як Google, Microsoft, Amazon Web Services і IBM, є її основною частиною. Провідні університети Північної Америки, Близького Сходу та Китаю також використовують платформу Open edX, щоб надавати своїм студентам цікавий та інтерактивний навчальний контент.

5. Інноваційна. Потужний інтерфейс з відкритим вихідним кодом платформи Open edX дозволяє найгеніальнішим умам світу робити внесок у платформу; отже, інновації відбуваються постійно. Покращений інтерфейс і корисний досвід програмного забезпечення Open edX є набагато вдосконаленішим і сучаснішим, ніж його конкуренти.

6. Орієнтація на учня. Платформа заохочує навчання різними способами, наприклад, за допомогою опитувань, вікі, соціальних мереж, відео та вправ. Велика різноманітність типів контенту дозволяє розробникам контенту створювати контент, який підвищує навички та компетенції студентів.

7. Потужна аналітика. Статистика та аналітика дають платформі Open edX всебічну перевагу над конкурентами. Можна створювати звіти для виявлення проблем, з якими стикаються учні під час курсу. Статті, надані у звітах, можуть бути використані для покращення курсу та можуть допомогти розробникам курсу, надаючи зворотний зв'язок у реальному часі [32].

Udemy - це платформа, яка дозволяє викладачам створювати онлайн-курси на бажані теми. Використовуючи засоби розробки курсів Udemy, вони можуть завантажувати відео, презентації, PDF-файли, аудіо, ZIP-файли і живі класи для створення курсів. Викладачі також можуть залучати та взаємодіяти з користувачами за допомогою онлайн-форумів (рис. 1.4) [14].

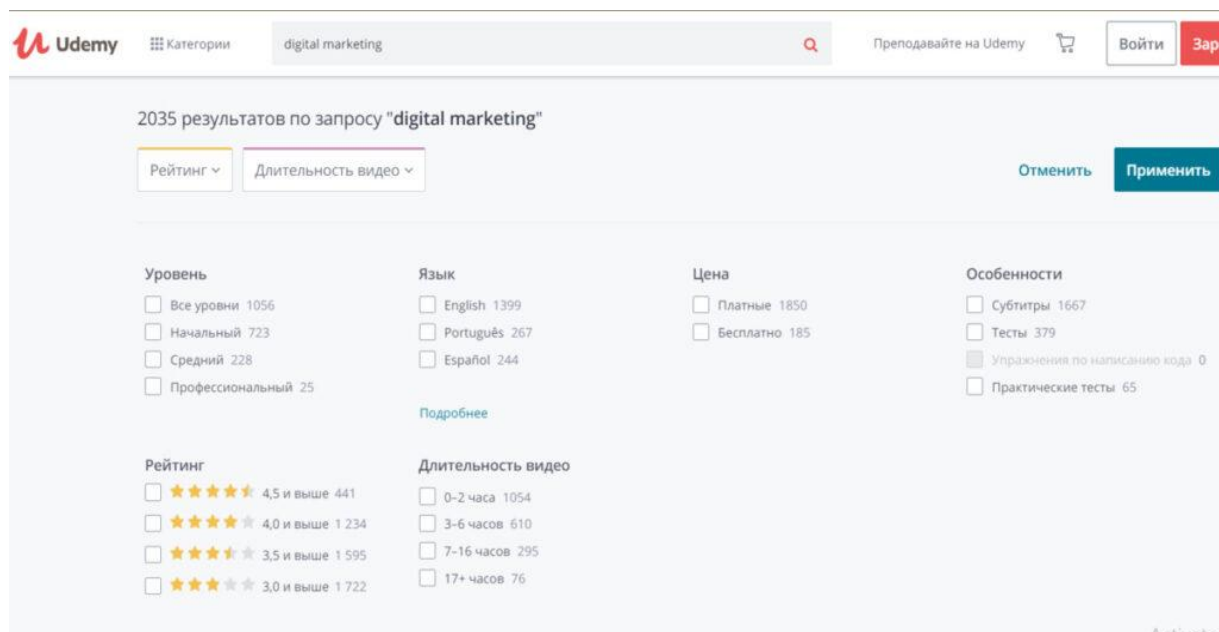


Рисунок 1.4 – Платформа Udemу

Курси пропонуються по широкій категорії, включаючи бізнес та підприємництво, науковці, мистецтво, здоров'я та фітнес, мова, музика та технології. Udemу також пропонує Udemу для бізнесу, надаючи бізнесу доступ до цільового набору з понад 7000 навчальних курсів на теми від тактики цифрового маркетингу до офісної продуктивності, дизайн, управління, програмування, і більше. Завдяки Udemу for Business організації також можуть створювати власні навчальні портали для корпоративного навчання [14].

Порівняння освітніх платформ показано у табл. 1.1.

Таблиця 1.1 – Порівняльна таблиця

Критерій	udx	Coursera	Udemу
Оцінювання	Завдання, тести, проекти	Завдання, тести, проекти	Автоматизовані тести
Прогнозування успішності	Відсутнє	Обмежене	Відсутнє

Ціна	Безкоштовно з платними сертифікатами	Безкоштовно з платними сертифікатами	Переважно платно
------	--	--	---------------------

PASS (Predicting Ability of Students to Succeed) – експертна система для прогнозування ймовірності успішного складання національних іспитів у Греції. Вона орієнтована на учнів технічних і професійних шкіл, допомагаючи виявити тих, хто має високий потенціал, і підтримати студентів із ризиком невдачі.

Система базується на платформі CLIPS і використовує базу знань із правил та фактів. Прогнозування відбувається через початковий і проміжний етапи, використовуючи коефіцієнти впевненості (CF) від 0 до 1. Формула CF враховує вагові коефіцієнти, що визначаються під час тренування системи, забезпечуючи точність, чутливість та специфічність.

Порівняння з логістичною регресією показало, що PASS забезпечує схожу точність, але краще збалансовує оцінювання успішності студентів. Вона допомагає викладачам визначати учнів із високим потенціалом або ризиком відрахування.

Однак система має кілька обмежень. Вона створена спеціально для освітньої системи Греції, базується на застарілій платформі та використовує обмежену формулу CF. Відсутність адаптивного навчання та залежність від ручного оновлення знижують її ефективність. Крім того, невелика вибірка даних (201 запис) обмежує її масштабованість.

PASS фокусується лише на прогнозуванні успішності на іспитах. Для сучасних освітніх систем потрібні гнучкіші моделі з підтримкою великих даних, інтеграцією з навчальними платформами та ширшим функціоналом, включаючи академічне консультування й управління навчальними траєкторіями.

Необхідність створення нової системи впливає з потреби у сучасних методах машинного навчання, підтримці великих даних та ширшому

функціоналі для академічного консультування, прогнозування ризику відрахування та вибору навчальної траєкторії. Більше того, сучасні освітні платформи, які ми розглянули, взагалі не підтримують функції прогнозування результатів контрольних заходів, таких як іспити чи модульні тести. Це суттєвий недолік, оскільки можливість прогнозувати академічні результати дозволяє вчасно виявляти студентів, які можуть зазнати труднощів у навчанні, і надавати їм необхідну підтримку.

Отже, створення нової експертної системи з використанням сучасних технологій, інтеграції великих даних і адаптивних алгоритмів стане вагомим внеском у розвиток освітніх технологій. Це дозволить підвищити точність прогнозів, розширити функціонал та інтегрувати систему в навчальні платформи для підтримки освітнього процесу на новому рівні.

1.3 Технічне завдання на розробку експертної системи для прогнозування результату контрольного заходу

ЗМІСТ

1. Вступ

1.1 Загальні визначення

1.2 Призначення документу

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

2.2 Характеристика об'єктів автоматизації

3. Вимоги до програмного засобу

3.1 Вимоги до оформлення інтерфейсу

3.2 Вимоги до зберігання даних і мов програмування

3.3. Вимоги до розділення доступу

3.4 Вимоги до програмного та технічного забезпечення

4. Стадії і етапи розробки програмного засобу

4.1 Склад і зміст робіт із розробки програмного засобу

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи програми

5.2. Вимоги до введення програмного засобу в дію

1. Вступ

1.1 Загальні визначення

Повна назва програмного продукту: експертна система для прогнозування результату контрольного заходу.

Замовник – організація, яке замовляє створення програмного засобу.

Виконавець – програміст, який розробляє програмний засіб та встановлює його на комп'ютер Замовника, в даному випадку це студент 6 курсу очного відділення Павленко Н.А.

JSON (JavaScript Object Notation) – це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передавання структурованої інформації.

C++, Qt Creator – мова програмування та середовище візуальної розробки додатків, що дозволяють розробити програмний засіб за невеликий період часу.

1.2 Призначення документу

У цьому документі приводиться повний перелік вимог до програмної реалізації та впровадження програмного засобу. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

1. Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.

2. Замовник згоден зі всіма положеннями цього Технічного Завдання.

3. Замовник не має права вимагати від Виконавця в рамках поточного Договору виконання робіт або надання послуг, прямо не описаних у справжньому Технічному Завданні.

4. Виконавець зобов'язується виконати роботи в об'ємі, вказаному в справжньому Технічному Завданні.

5. Замовник не має права вимагати від Виконавця дотримання яких-небудь форматів і стандартів, якщо це не вказано в справжньому Технічному Завданні.

6. Все неоднозначності, виявлені в справжньому Технічному завданні після його підписання, підлягають двосторонньому узгодженню між Сторонами. В процесі узгодження можуть бути розроблені додаткові вимоги, які оформляються додатковою угодою до Договору і відповідним чином оцінюються.

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

Предметом розробки є експертна система, що прогнозує результат контрольного заходу.

Програмний засіб призначений для збирання та аналізу даних про підготовку студентів, зокрема інформації про виконання лабораторних робіт, результати тестувань, рівень теоретичних знань та інші важливі фактори, які можуть впливати на успішність. Система забезпечує автоматичну обробку введених даних, прогнозує ймовірність успішного складання контрольного заходу та надає рекомендації щодо покращення підготовки.

2.2 Характеристика об'єктів автоматизації

Багато закладів освіти стикаються з проблемою об'єктивного оцінювання готовності студентів до контрольних заходів. Це стосується як окремих дисциплін, так і загальної підготовки студентів. Успішне проходження контрольних заходів залежить від багатьох факторів, таких як рівень виконаних лабораторних робіт, результати теоретичних тестів та якість засвоєння матеріалу.

Одним із важливих аспектів у забезпеченні успішності студентів є якісне управління даними про їхню підготовку. Використання експертної системи

для прогнозування результату контрольного заходу дозволяє автоматизувати цей процес, мінімізуючи ризики суб'єктивності.

Експертна система забезпечує автоматизоване робоче місце викладача або керівника дисципліни. Основні функції системи включають накопичення, збереження та обробку інформації про виконання лабораторних робіт, результати тестів, рівень теоретичних знань, тощо. Завдяки використанню комп'ютерної техніки значна частина рутинних операцій автоматизується, тоді як викладач має змогу сконцентруватися на прийнятті відповідальних рішень, що потребують творчого підходу.

Система дозволяє зберігати детальну інформацію про кожного студента, включаючи кількісну та якісну оцінку їхньої підготовки. На основі проаналізованих даних експертна система прогнозує ймовірність успішного складання контрольного заходу, формує рекомендації для покращення результатів та дозволяє роздрукувати потрібні звіти.

Таким чином, використання такого програмного засобу значно підвищує ефективність управління процесом підготовки студентів, сприяючи об'єктивності оцінювання та підвищенню загальної якості освітнього процесу.

3. Вимоги до програмного засобу

3.1 Вимоги до оформлення інтерфейсу

Графічний інтерфейс програмного засобу повинен бути легким (розрахованим на користувача з початковим рівнем інформаційної культури та навичок роботи за комп'ютером), але в той же час функціональним. Вікно для введення тексту та оцінок повинен мати мінімальну кількість елементів управління (кнопок, пунктів меню, полів введення, ...) (рис. 1.5).

Предмет

Група

ПІБ студента

Вхідні данні

Кількість виконаних лабораторних робіт - %

Статус виконаних робіт

Оцінка [0-100]	Тип роботи

Теоретичні тести

Оцінка [0-100]	Список тем

Оцінити

Вихідні данні

Ймовірність допуску до сесії - 25%

Рекомендації

Зберегти у файл

Рисунок 1.5 – Інтерфейс головного вікна

При виборі предмету повинно автоматично заповнюватися поля у блоці вхідні дані, після цього можна заповнювати оцінками конкретного студента. Для збереження вихідних даних у файл повинно відкривається типове вікно збереження файлу (рис. 1.6).

Шлях до файлу

Список файлів

Зберегти

Відмінити

Рисунок 1.6 – Вікно збереження файлу

3.2 Вимоги до зберігання даних і мов програмування

Усі дані про кожен предмет повинні зберігатися у файлах json, які знаходять разом з нашою експертною системою. Для створення експертної системи повинно використовуватися середовище розробки QtCreator та мова програмування C++. Зберігання вихідних даних повинні виконуватися у текстовий файл.

3.3 Вимоги до розділення доступу

Програма повинна розповсюджуватися з конфігураційними файлами для кожного предмета який хоче працювати з експертною системою та файл який містить список саме назв файлів-предметів.

3.4 Вимоги до програмного та технічного забезпечення

Для функціонування програмного засобу необхідне наступне програмне та мінімальне апаратне забезпечення:

- Операційна система – Windows XP та більш пізні версії;
- Framework 4.5;
- Конфігураційні файли з предметами у форматі json;
- процесор – Intel Pentium IV 1 Ghz або більш потужний;
- оперативна пам'ять – 1 Гб RAM або більше;
- жорсткий диск – 256 Гб або більше.

4. Стадії і етапи розробки програмного засобу

4.1 Склад і зміст робіт із розробки програмного засобу

Розробка експертної системи для прогнозування контрольного заходу припускає виконання наступних етапів:

1. Аналіз наочної області, яка буде реалізована в програмному засобі, в даному випадку данні з якими потрібно працювати у додатку.
2. Зберігання потрібної інформації для роботи програмного засобу.
3. Вибір інструментального середовища, за допомогою якого буде створений програмний продукт.
4. Створення структури конфігураційних файлів.

5. Створення модулів експертної системи прогнозування контрольного заходу.

6. Тестування і перевірка працездатності програмного продукту в реальних умовах.

7. виправлення помилок роботи програми і проведення завершального показу Замовнику.

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи програми

Приймання експертної системи для прогнозування до експлуатації складається з наступної перевірки:

1. Вибір предмету для якого буде виконуватися аналіз.

2. Введення ПІБ студента та його групи у разі необхідності зберегти результат для конкретного студента у текстовий файл (не обов'язкові поля).

3. Перевірка можливих допустимих даних у поля вводу оцінок, введення оцінок.

4. Отримання результату аналізу експертної системи у вигляді вірогідності допуску до сесії від 0 до 100 відсотків та рекомендації щодо покращення цього показника.

5. Коректне зберігання вихідної інформації у файл.

6. Очищення всіх вихідних даних при зміні предмету та оновлення таблиць з актуальними роботами для предмету.

5.2. Вимоги до введення програмного засобу в дію

Підготовка програмного засобу до експлуатації припускає:

1. Отримання Замовником експертної системи для прогнозування контрольного заходу у вигляді додатку, необхідних бібліотек/допоміжних файлів для коректної роботи додатку та приклади конфігураційних файлів з предметами.

2. Показ відповідальному представнику Замовника програмного засобу.

РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ЕКСПЕРТНОЇ СИСТЕМИ

2.1 Концептуальна модель використання експертної системи прогнозування контрольних заходів

Як відомо, перед створенням потужного програмного забезпечення за допомогою конкретного інструментального середовища необхідно виконати етап проєктування його структури та функціональності. Ця технологія являє собою набір методів проєктування та інструментів, що дозволяють моделювати предметну область у зручній візуальній формі, аналізувати створену модель на всіх етапах розробки й супроводу інформаційних систем, а також створювати програмні рішення, які відповідають вимогам кінцевих користувачів [7, с. 223]. Цей процес повинен забезпечувати детальний і чіткий опис логіки додатка, який відповідає системним вимогам та враховує можливі обмеження. Водночас у процесі розробки та експлуатації інформаційної системи потреби користувачів можуть змінюватися або уточнюватися, що ускладнює як розробку, так і подальший супровід таких систем [30, с. 10].

Під час проєктування ми будемо використовувати список основних скорочень і термінів, наведений у табл. 2.1.

Таблиця 2.1 – Глосарій

Термін	Опис терміну
1	2
1 Основні поняття та категорії проєкту	
Програмне забезпечення	комплекс програм для обробки інформації разом із програмною документацією, яка потрібна для їх використання

Продовження таблиці 2.1

1	2
UML (Unified Modeling Language)	уніфікована мова моделювання, яка використовується для проектування, аналізу та документування програмних систем, а також для візуалізації структур і процесів у різних сферах діяльності
Інтегрована середа розробки	набір програмних інструментів, що забезпечує всі необхідні функції для створення, компіляції, зв'язування та налагодження програмного коду
Програма	сукупність команд, що визначають дії, які повинен виконати комп'ютер
Фреймворк	програмні інструменти, що полегшують розробку та підтримку складних або ресурсомістких проєктів
Qt	кросплатформний фреймворк для розробки програмного забезпечення, який дозволяє створювати додатки з графічним інтерфейсом користувача і без нього (наприклад, серверні або консольні програми)
Qt Creator	інтегрована середа розробки для створення додатків із використанням фреймворку Qt

Більшість об'єктно-орієнтованих моделей можна створити за допомогою уніфікованої мови моделювання UML (Unified Modeling Language). У UML усі аспекти моделі складної системи представлені у вигляді спеціальних графічних конструкцій, які називаються діаграмами. Діаграма (diagram) у цьому контексті – це графічне зображення, що відображає набір

елементів моделі у вигляді зв'язного графа, де вершини та ребра (дуги) мають певну семантику [29, с. 6].

Візуальне моделювання в UML можна розглядати як процес поступового переходу від загальної та абстрактної концептуальної моделі початкової системи до логічної, а згодом і до фізичної моделі програмної системи. Для цього спершу створюється модель у вигляді діаграми варіантів використання (use case diagram), яка відображає функціональне призначення системи, тобто визначає, що саме система буде виконувати під час свого функціонування. Ця діаграма є початковим концептуальним представленням або концептуальною моделлю системи на етапах її розробки та впровадження.

Діаграма варіантів використання експертної системи прогнозування контрольних заходів наведена на рис. 2.1



Рисунок 2.1 – Діаграма варіантів використання

Створення діаграми варіантів використання спрямоване на досягнення таких цілей:

- визначення меж і контексту предметної області, що моделюється, на початкових етапах проектування системи;

- формування загальних вимог до функціональної поведінки проектованої системи;
- створення початкової концептуальної моделі системи для подальшої деталізації у вигляді логічних і фізичних моделей;
- підготовка документації для забезпечення ефективної взаємодії між розробниками, замовниками та користувачами системи.

Основна ідея цієї діаграми полягає у представленні системи як множини сутностей (акторів), які взаємодіють із нею через варіанти використання. Актором (actor) називається будь-яка зовнішня сутність, яка взаємодіє з системою. Це може бути людина, технічний пристрій, програма або інша система, що впливає на модельовану систему згідно з визначенням розробника.

Варіант використання (use case) слугує для опису послуг, які система надає актору. Кожен варіант використання визначає конкретний набір дій, що виконує система під час взаємодії з актором, при цьому не уточнюючи, як саме реалізовано цю взаємодію.

У загальному вигляді діаграма варіантів використання є спеціалізованим графом, що використовується для графічного представлення варіантів використання, акторів, інтерфейсів та взаємозв'язків між ними. Окремі компоненти діаграми можуть бути поміщені у прямокутник, який символізує проектовану систему в цілому.

Розкадровка – це логічний і концептуальний опис функціональних можливостей системи в межах конкретного сценарію, який враховує необхідну взаємодію між системою та її користувачами. На основі розкадровки мають бути сформульовані вимоги, що підлягають узгодженню із зацікавленими сторонами.

Розкадровка виконана за допомогою середовища розробки QtCreator (рис. 2.2).

The image shows a screenshot of a software interface for an expert system. It features several input fields and sections:

- Top Section:** Includes a dropdown menu for "Назва предмету" (Subject Name), input fields for "Номер групи" (Group Number) and "ПІБ студента" (Student's Full Name), and a "Вхідні данні" (Input Data) label.
- Lab Work Section:** Contains a progress bar for "Кількість виконаних лабораторних робіт - 0%" (Number of completed lab works - 0%), a "Статус виконаних робіт" (Status of completed works) label, and a table with columns "Оцінка [0-100]" (Grade [0-100]) and "Тип роботи" (Type of work).
- Theoretical Tests Section:** Includes a "Теоретичні тести" (Theoretical tests) label, a table with columns "Оцінка [0-100]" (Grade [0-100]) and "Список тем" (List of topics), and a "Вихідні данні" (Output Data) label.
- Recommendations Section:** Features a "Рекомендації" (Recommendations) label, a progress bar for "Ймовірність допуску до сесії - 0%" (Probability of admission to the session - 0%), and a "Зберегти у файл" (Save to file) button.

Рисунок 2.2 – Розкадровка інтерфейсу експертної системи

2.2 Функціональна модель експертної системи прогнозування контрольних заходів

IDEF0 – це мова моделювання бізнес-процесів (BPML), яка використовується для опису рішень, дій та діяльності організації або системи (IDEF0, 1993). Отримана модель відображає знання про те, як функціонує система, процес або організація. IDEF0 описує конкретні кроки виконання процесу та взаємозв'язки, що виникають у його межах. Крім того, вона фіксує потоки інформації, що є результатом цих взаємозв'язків. Модель IDEF0 включає набір синтаксичних компонентів, необхідних для інтеграції бізнес-процесів. До цих компонентів належать блоки, стрілки та діаграми. Блоки позначають функції, які визначаються як дії, процеси або перетворення.

Стрілки відображають дані або об'єкти, пов'язані з функціями. Формат IDEF0 також забезпечує основу для управління конфігурацією моделі [46, с. 110].

Основна перевага добре задокументованої методології полягає у структурованості, яку вона забезпечує. Завдяки наявності такої методології новачки можуть виконувати поставлені завдання більш ефективно та результативно. Формалізована методологія створює чітку структуру, яка сприяє повторюваності успіхів і забезпечує кращу інтеграцію роботи кількох розробників [44, с. 442].

Дана експертна система написана за допомогою мови програмування C++. Функціональна модель першого рівня представлена на рис. 2.3.



Рисунок 2.3 – Контекстна діаграма експертної системи в нотації IDEF0

Основна функція системи – прогнозування контрольного заходу для студента. На вхід подається: предмет, ПІБ студента, конфігураційні файли, оцінки по тестам, відсоток виконаних лабораторних робіт та оцінки профільних робіт. Стрілки управління це функції: побудови рекомендацій, зчитування конфігураційних файлів та збереження рекомендацій. На виході

отримуємо: рекомендації та збережені рекомендації у файлі. У ролі механізму виступає користувач.

На рис. 2.4 представлена декомпозиція головної функції системи «Прогнозування контрольного заходу».

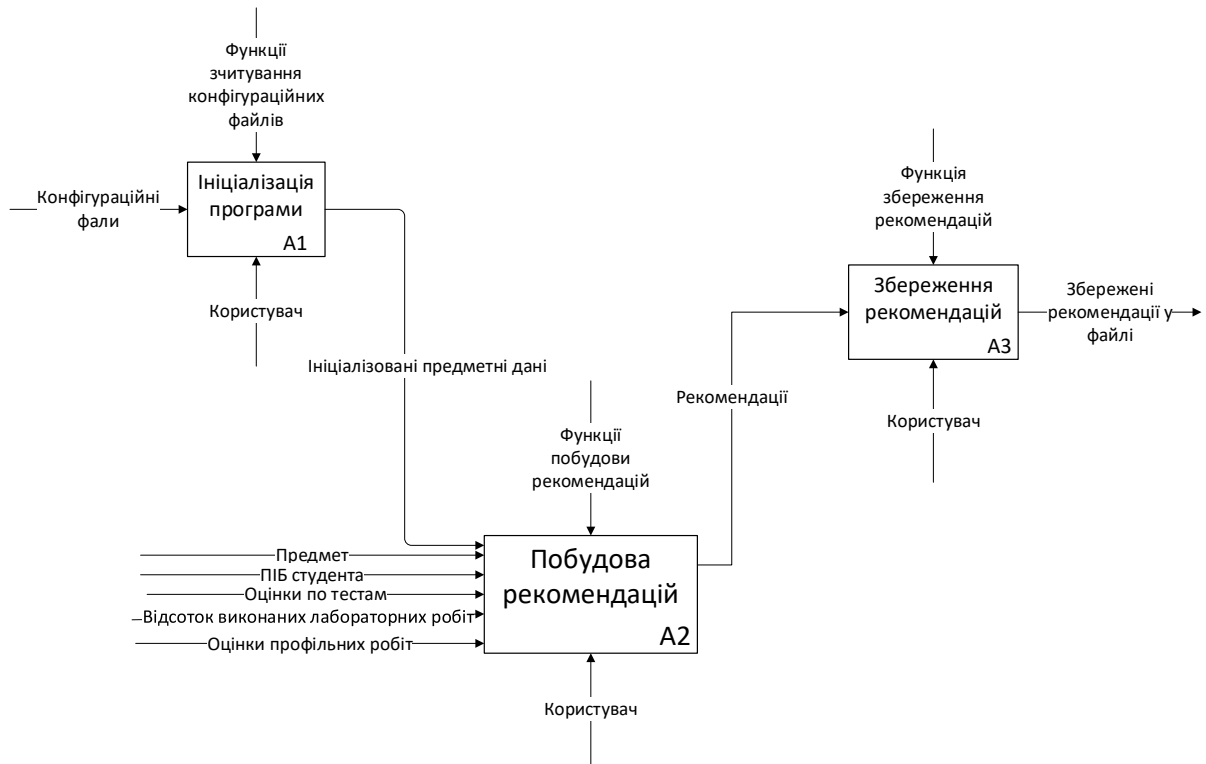


Рисунок 2.4 – Декомпозиція контекстної діаграми в нотатції IDEF0

На діаграмі зображено функції:

- ініціалізації програми. На вхід функції подаються конфігураційні файли, на вихід ініціалізовані предметні дані;
- побудова рекомендацій;
- збереження рекомендацій

2.3 Специфікація функціональних та нефункціональних вимог

Список та опис усіх функціональних вимог розроблюваної системи и атрибутів вимог наведено у табл. 2.2

Таблиця 2.2 – Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
01	Вибір предмету	Обов.	Середня	Розробник
02	Введення % виконаних лабораторних робіт	Обов.	Середня	Розробник
03	Виставлення оцінок по тестах	Обов.	Середня	Розробник
04	Виставлення оцінок по профільним роботам	Обов.	Середня	Розробник
05	Введення ПІБ студента	Необов.	Легка	Розробник
06	Побудова рекомендацій	Обов.	Висока	Розробник
07	Збереження рекомендацій	Обов.	Середня	Розробник

До нефункціональних вимог відносяться всі вимоги, Які не стосуються функціональності проекту. Детальний опис нефункціональних вимог наведено у табл. 2.3

Таблиця 2.3 – Специфікація нефункціональних вимог

№	Назва вимоги	Характеристика
1	2	3
1 Застосовність		
1.1	Час для навчання користувачів	Від 10 хв. (при наявності інструкції)

Продовження таблиці 2.3

1	2	3
1.2	Вимірюваний час відгуку для типових кнопок та меню	До 1с.
2 Надійність		
2.1	Середній час безвідмовної роботи	Безтерміново.
2.2	Середнє напрацювання до ремонту ПК	Безтерміново.
3 Робочі характеристики		
3.1	Швидкодія для транзакції	Від 0.5 до 2х с.
3.2	Продуктивність (в кількості одночасних користувачів)	Необмежено.
4 Проектні обмеження		
4.1	Мова програмування	C++
5	Вимоги до документації, призначеної для користувача, і до системи допомоги	Вимоги вказані в ТЗ
6 Інтерфейси		
6.1	Інтерфейси користувача	Згідно з ТЗ.
6.2	Програмні інтерфейси	Вказуються замовником.
6.3	Комунікаційні інтерфейси	Вказуються замовником.
7	Вимоги до ліцензування	Згідно з ТЗ.
8	Вживані стандарти	Згідно з нормативними документами та ГОСТами

РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ

3.1 Загальний огляд технологічного стеку

Для розробки експертної системи використано мову програмування C++.

C++ перевершує інші мови програмування завдяки спрощенню процесів розробки, підтримки, оновлення та повторного використання програм. Основні переваги мови включають використання покажчиків, динамічних структур даних, об'єктів, а також механізми приховування даних, успадкування та перевантаження. Завдяки цим можливостям C++ забезпечує гнучкість та зручність у роботі для програмістів [38, с. 95].

Особливостей C++ є підтримка комплексних чисел, які мають такий самий синтаксис, як і звичайні числа у математичних виразах. Динамічні структури даних, такі як вектори, списки, дерева та графи, доступні через стандартну бібліотеку шаблонів (STL), що значно спрощує роботу з їх реалізацією [38, с. 95].

C++ є об'єктно-орієнтованою мовою програмування (ООП), де програма організована навколо об'єктів. Об'єкт складається з двох частин: внутрішнього подання даних і інтерфейсу. Інтерфейс включає функції, які взаємодіють із іншими частинами програми. Лише ці функції мають доступ до внутрішнього подання даних, що забезпечує приховування даних. Завдяки цьому програма має чітку структуру, яку легко розробляти, документувати та оновлювати [38, с. 95].

Однією з ключових переваг ООП у C++ є механізм успадкування. Це дозволяє уникнути повторного визначення схожих об'єктів з нуля, замість цього створюючи нові об'єкти, які успадковують властивості існуючих і додають лише необхідні зміни. Крім того, C++ підтримує перевантаження операторів і функцій, що дозволяє використовувати однаковий синтаксис для

різних типів об'єктів. Наприклад, оператор `*` може застосовуватися для множення матриць, а функція `print(A)` – для виведення даних незалежно від типу об'єкта, будь то комплексне число, вектор чи інший тип [38, с. 95].

C++ значно розширює можливості програмування, роблячи його потужним інструментом для створення ефективних та структурованих програм [38, с. 95].

C++ – це об'єктно-орієнтована мова програмування (ООР), яка була розроблена Б'ярне Страуструпом на початку 1980-х років. Вона є однією з перших мов, яка успішно інтегрувала принципи об'єктно-орієнтованого програмування з традиційним процедурним підходом, успадкованим від мови C. Завдяки цьому програмісти мають можливість писати код як у стилі C, так і з використанням об'єктно-орієнтованих концепцій, таких як класи, об'єкти, успадкування, інкапсуляція та поліморфізм. Ця гнучкість робить C++ чудовим прикладом гібридної мови програмування, яка поєднує переваги обох парадигм [39, с 498].

Однією з унікальних рис C++ є її здатність адаптуватися до різних підходів програмування, дозволяючи розробникам створювати як прості програми, орієнтовані на продуктивність, так і складні, структуровані системи, що базуються на принципах об'єктно-орієнтованого програмування. Ця універсальність робить мову C++ популярною у багатьох галузях, включаючи розробку ігор, системного програмного забезпечення, вбудованих систем і наукових обчислень [39, с 498].

Втім, для початківців C++ може бути складною мовою, оскільки об'єктно-орієнтоване програмування вводить багато нових понять і термінів, які можуть бути незрозумілими навіть тим, хто має значний досвід роботи з іншими мовами програмування. Наприклад, такі концепції, як класи, об'єкти, успадкування або поліморфізм, можуть викликати труднощі у розумінні, оскільки вони вимагають іншого способу мислення порівняно з процедурним підходом [39, с 498].

Додаткові труднощі виникають через те, що різні об'єктно-орієнтовані мови програмування використовують різну термінологію для опису схожих концепцій. Це може заплутати новачків і ускладнити процес вивчення. Наприклад, терміни "методи" в одній мові можуть називатися "функціями-членами" в іншій, що створює бар'єри для тих, хто переходить з однієї мови програмування на іншу [39, с 498].

Таким чином, C++ є потужною, але складною мовою, яка вимагає глибокого розуміння як об'єктно-орієнтованих принципів, так і процедурного програмування. Однак її універсальність і гнучкість роблять її незамінним інструментом для професійних розробників [39, с 498].

C++ характеризується такими ключовими особливостями, як об'єкти, класи, абстракція, інкапсуляція, успадкування, повторне використання коду, поліморфізм, динамічне зв'язування та передача повідомлень. Додаткові переваги мови включають акцент на роботі з даними, а не процедурами, поділ програми на об'єкти, інтеграцію функцій і даних у єдину структуру. Завдяки механізмам приховування даних запобігається їх випадкова зміна іншими функціями чи об'єктами. Доступ до даних здійснюється через спеціальні функції, що дозволяє забезпечувати взаємодію між об'єктами. Структури даних представлені у вигляді об'єктів, а розробка програм базується на методології «знизу догори» [39, с 499].

Для графічного інтерфейсу використовується фреймворк Qt.

Qt – це сучасний кросплатформний фреймворк із відкритим вихідним кодом, який широко використовується для розробки програмного забезпечення на різних типах пристроїв. Його створено для роботи як з настільними комп'ютерами, так і з мобільними пристроями та вбудованими системами, що робить його універсальним інструментом у розробці. Історія Qt розпочалася ще у 1990 році, і за цей час фреймворк зазнав численних удосконалень, перетворившись на один із найбільш популярних і функціональних інструментів для програмування [36, с. 17].

Сьогодні розробка Qt здійснюється в межах спільноти "The Qt Project", яка об'єднує розробників з усього світу. Основою фреймворку є мова програмування C++, що забезпечує високу продуктивність та гнучкість у створенні застосунків. Важливим елементом екосистеми Qt є власне інтегроване середовище розробки – Qt Creator, яке включає зручні інструменти для програмування та проектування графічного інтерфейсу користувача. Зокрема, Qt Designer, що входить до складу Qt Creator, дозволяє створювати інтерфейси візуально, без необхідності писати код вручну [36, с. 17].

Однією з ключових переваг Qt є його можливість працювати на різних платформах, що включає операційні системи для настільних комп'ютерів (Windows, macOS, Linux), мобільні платформи (Android, iOS) та вбудовані системи. Завдяки цьому Qt стає все більш популярним серед розробників, які прагнуть створювати багатоплатформні рішення з мінімальними зусиллями щодо адаптації коду. Інтеграція з технологіями Qt/QML, орієнтованими на створення динамічних та інтуїтивних інтерфейсів, робить фреймворк особливо привабливим для мобільних застосунків [36, с. 17].

Окрім своєї технічної універсальності, Qt також відзначається активною підтримкою спільноти та наявністю великої кількості документації, що робить його доступним для вивчення навіть новачкам у програмуванні. Завдяки всім цим перевагам Qt є не лише інструментом для розробки, але й потужною платформою, яка дозволяє створювати складні, високоякісні та естетично привабливі застосунки. Це робить Qt перспективною технологією, яка активно використовується в сучасному світі програмування. [36, с. 17].

Розробка Qt була започаткована у 1991 році Гавардом Нордом і Еріком Чамбе-Енгом. Назва Qt походить від красивої літери Q у шрифті Navard Emacs і слова "toolkit". Компанія, що розвивала Qt, була зареєстрована 4 березня 1994 року під назвою «Quasar Technologies», згодом перейменована на «Troll Tech», «Trolltech», а після придбання компанією Nokia отримала назву «Qt Software». Сьогодні бібліотеку Qt використовують такі компанії, як AMD, Valve,

AutoDesk, Siemens, Kitware, Lucasfilm тощо. Вона є основою для таких проектів, як Skype, Google Earth, Adobe Photoshop Album і KDE [36, с. 76].

Qt є проектом з відкритим кодом із публічним Git-репозиторієм, що дозволяє стежити за розробкою, вносити корективи та пропонувати покращення. Однією з ключових особливостей Qt є компілятор метаоб'єктів (Meta-Object Compiler, moc), який обробляє розширення C++ у Qt. Moc читає заголовкові файли C++ і генерує вихідний код для підтримки механізмів сигналів і слотів, інформації про типи в час виконання та системи динамічних властивостей [42, с. 76].

Для створення програм із використанням бібліотек Qt найзручніше працювати в середовищі розробки Qt Creator, яке підтримує компілятори Microsoft VC++ і GCC, а також налагоджувач GDB. Qt Creator дозволяє працювати як із класичним C++, так і з QML (мова метаоб'єктів Qt) для створення вікон. Для зручної розробки графічного інтерфейсу Qt Creator включає інструмент Qt Designer, який дозволяє створювати діалогові вікна та форми в режимі WYSIWYG (What You See Is What You Get), використовуючи мишу. Це значно спрощує розробку графічних елементів на високому рівні абстракції [36, с. 76].

У ролі бази даних виступає конфігураційні json файли.

JSON (JavaScript Object Notation) – це популярний, легкий і напівструктурований формат даних, який базується на типах даних мови програмування JavaScript. За останні роки JSON став основним форматом обміну даними в мережі Інтернет, здобувши значну популярність як у веб-технологіях, так і серед дослідників баз даних. Завдяки своїй гнучкості JSON широко інтегрується не лише в традиційні реляційні бази даних, але й активно використовується в нереляційних (NoSQL) базах даних. Для роботи з JSON вже розроблені спеціалізовані мови запитів, такі як JSONiq і SQL++, які забезпечують зручне виконання запитів до JSON-структур [43, с. 012101].

Модель даних є основою будь-якого формату даних, включно з JSON. Вона визначає базові структури та семантику даних, що лежать в основі

формату. Як основоположний елемент, модель даних має вирішальне значення для багатьох технологій управління інформацією, таких як індексування, виконання запитів, пошук, мапування, інтеграція даних та аналіз даних (data mining). Від того, наскільки ефективно та потужно буде спроектована модель даних для JSON, залежить успішність вирішення суміжних дослідницьких задач, таких як інтеграція даних із різних джерел, побудова точних і масштабованих пошукових систем, оцінка якості даних тощо [43, с. 012101].

На сьогодні дослідники запропонували кілька підходів до моделювання JSON-документів, серед яких найбільш поширеними є моделі, засновані на деревоподібних структурах. Такі моделі дозволяють відображати вкладеність і ієрархічну природу JSON-даних, що відповідає їхній структурі у форматі ключ-значення та масивів [43, с. 012101].

Дослідження, спрямовані на розробку та вдосконалення моделей даних для JSON, відкривають широкі перспективи для подальших досліджень у галузі управління інформацією. Вони є критично важливими для розробки більш ефективних і універсальних технологій обробки даних у сучасних інформаційних системах, особливо враховуючи зростання популярності напівструктурованих форматів у різних галузях застосування [43, с. 012101].

Експертна система реалізована у вигляді додатку та конфігураційних файлів (рис. 3.1).

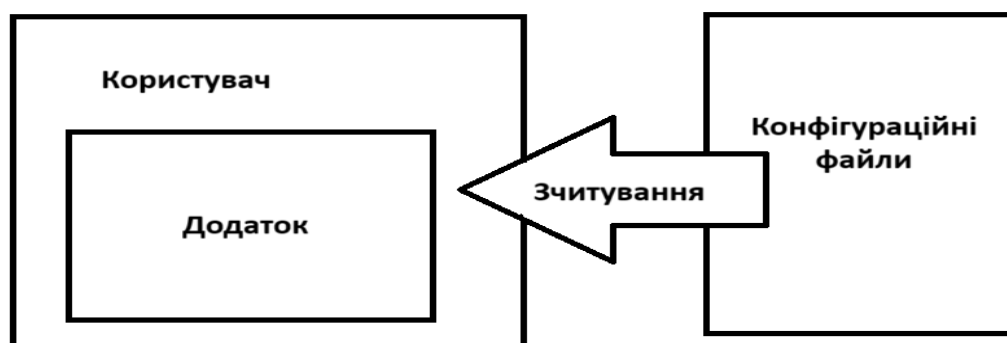


Рисунок 3.1 – Структурна схема експертної системи

У додатку відображається інтерфейс для роботи з експертною системою, а експертна система використовує конфігураційні файли для заповнення необхідних візуальних елементів у додатку та аналізу введених даних.

3.2 Обґрунтування вибору мови програмування

Одним із основних викликів під час створення експертних систем є вибір мови програмування, яка відповідатиме вимогам стабільності, продуктивності, масштабованості та інтеграції з іншими компонентами. С++ як мова, що поєднує об'єктно-орієнтований підхід і низькорівневий контроль, ідеально підходить для таких систем. Її застосування є обґрунтованим у розробці сучасних програмних продуктів, включно з високотехнологічними рішеннями, наприклад, прогнозувальними системами [11, с. 52].

Широкий спектр застосувань. С++ використовується для створення операційних систем, прикладних програм, систем управління базами даних і навіть ігрових рушіїв. Експертні системи часто поєднують елементи з різних категорій, тому гнучкість мови є ключовою перевагою.

Наявність потужних інструментів. Зокрема, Microsoft Visual Studio, Qt Creator та інші середовища розробки забезпечують ефективну розробку, тестування та налагодження програмного забезпечення. Наприклад, Qt активно використовується для створення інтуїтивних графічних інтерфейсів, які можуть бути необхідні в експертних системах для зручного відображення результатів аналізу [18, с. 155].

Нові стандарти. Постійне вдосконалення стандартів С++ (зокрема, С++17 і С++20) дає програмістам нові засоби для спрощення розробки, наприклад, завдяки впровадженню функціонального програмування та вдосконаленню стандартної бібліотеки. Це забезпечує актуальність мови для вирішення сучасних задач [3, с. 250].

Конкретні переваги для експертних систем:

- робота з великими обсягами даних. Прогнозувальні системи обробляють значну кількість даних, виконуючи складні математичні операції. Завдяки ефективному управлінню пам'яттю та швидкій компіляції коду C++ стає важливим інструментом для таких операцій;

- стабільність у реальному часі. Для експертних систем, що потребують роботи в режимі реального часу, критично важливими є мінімізація затримок і забезпечення стабільної роботи, чого легко досягти за допомогою C++;

- інтеграція з іншими системами. Можливість роботи з базами даних, зовнішніми сенсорами та іншими програмними модулями через API або драйвери є важливою перевагою C++. Його багатоплатформність дозволяє створювати рішення, які працюють у різних середовищах [3, с. 250].

Експертна система прогнозування контрольного заходу вимагає ретельного підходу до структурування, що включає:

- чітку модульність. Наприклад, модулі введення даних, аналізу і прогнозування мають бути реалізовані окремо для забезпечення їхньої масштабованості;

- ефективність алгоритмів. C++ дозволяє оптимізувати обчислення завдяки використанню низькорівневих інструкцій і спеціалізованих бібліотек, таких як Eigen для лінійної алгебри або Boost для обробки складних структур даних;

- зручність обслуговування. Завдяки об'єктно-орієнтованому підходу програма стає зрозумілою і легкою у підтримці, що важливо для тривалого життєвого циклу експертної системи.

Незважаючи на численні переваги, мова має і певні недоліки:

- складність налагодження. Система управління пам'яттю вимагає від розробника уважності, щоб уникнути витоків або інших помилок;

- довгий час розробки. Створення програми на C++ зазвичай займає більше часу через деталізованість процесу, однак це компенсується високою продуктивністю готового продукту;

– потреба у високій кваліфікації програмістів. С++ вимагає глибокого розуміння принципів роботи комп'ютера, що може бути викликом для молодих спеціалістів [18, с. 155].

Мова С++ залишається актуальною завдяки своїм оновленням та потужним інструментам. Для експертних систем прогнозування це ідеальний вибір через:

- гнучкість у реалізації алгоритмів машинного навчання та аналізу даних;
- підтримку багатопотокових обчислень. Це дозволяє системі працювати швидше на багатоядерних процесорах;
- можливість інтеграції з іншими мовами програмування, наприклад Python для швидкого прототипування моделей.

Таким чином, використання С++ для розробки експертної системи є виправданим і обґрунтованим вибором, який поєднує продуктивність, гнучкість і стабільність.

Фреймворк Qt є одним із найбільш потужних і гнучких інструментів для створення програмного забезпечення, який забезпечує високу ефективність і кросплатформність. У контексті розробки експертної системи прогнозування контрольного заходу Qt пропонує безліч переваг, таких як швидка розробка, надійність та інтеграція з сучасними технологіями, включаючи роботу з JSON-файлами для зберігання та обробки даних.

1. Кросплатформність. Однією з головних переваг Qt є можливість створювати програми, що працюють на різних операційних системах: Windows, macOS, Linux, Android, iOS та інших. Для експертної системи це означає, що програмний продукт можна легко адаптувати для роботи як на серверних платформах, так і на мобільних пристроях або спеціалізованих вбудованих системах. Розробка одного універсального коду значно скорочує витрати часу і ресурсів на підтримку програмного забезпечення для різних платформ [26, с. 241].

2. Гнучкість у розробці. Qt має великий набір модулів, які відповідають за специфічні функції, від роботи з інтерфейсами до обробки даних і мережевого підключення. Наприклад Qt Quick та QML дозволяють створювати динамічні інтерфейси користувача з інтерактивними анімаціями, які можуть бути зручними для відображення прогнозів у реальному часі. Механізм сигналів та слотів, що вбудований у Qt, спрощує роботу з подіями та значно знижує складність програмного коду.

3. Швидкість розробки. Qt Designer надає візуальний інструмент для створення графічного інтерфейсу без необхідності писати код вручну. Це суттєво прискорює процес розробки, дозволяючи програмістам зосередитися на функціональних аспектах експертної системи, а не на розробці інтерфейсу.

4. Робота з JSON. Для роботи з даними в експертних системах JSON-файли є ефективним форматом зберігання та обміну інформацією. У Qt робота з JSON-файлами реалізується через класи, які забезпечують: просте читання і зберігання структурованих даних, гнучке редагування JSON-об'єктів та масивів, легке використання для конфігурації системи, зберігання результатів аналізу чи обміну інформацією з іншими сервісами.

JSON-файли є гнучким форматом для обміну даними, і Qt забезпечує повну підтримку роботи з ними. Експертна система може використовувати JSON для таких задач: Збереження конфігурації: JSON-файли можуть містити налаштування системи, які легко редагуються. Дані аналізу або прогнозування можуть бути структуровані в JSON, що забезпечує зручний формат для передачі між різними модулями системи. Збереження дій користувача або попередніх прогнозів у JSON-файлах дозволяє аналізувати тренди або переглядати історію роботи системи.

Недоліки Qt:

- великі розміри додатків Qt додає велику кількість бібліотек до створених програм, що збільшує їх розмір. Це може стати проблемою для мобільних платформ або вбудованих систем із обмеженими ресурсами;

- ліцензування Для комерційного використання Qt потрібна платна ліцензія, що може збільшити витрати на розробку;
- потреба у висококваліфікованих розробниках Хоча Qt забезпечує значну гнучкість, для повного використання його можливостей потрібна висока кваліфікація програмістів, особливо при роботі з його розширеними модулями.

JSON (JavaScript Object Notation) є популярним форматом для зберігання й передачі даних завдяки його простоті та універсальності [35с. 123].

Переваги JSON-файлів

1. Простота формату. JSON має зрозумілу ієрархічну структуру на основі пар "ключ-значення". Його можна легко читати і писати як людьми, так і машинами. Ця характеристика робить JSON зручним для створення та редагування конфігурацій, навіть без спеціалізованих інструментів [35, с. 123]. Ця структура дозволяє зберігати складні правила, що описують контрольні заходи, у вигляді вкладених об'єктів і масивів.

2. Гнучкість і динамічність. JSON не вимагає суворої схеми (на відміну від реляційних баз даних). Це означає, що можна легко додавати нові поля або змінювати формат без потреби в значних змінах системи. У контексті експертних систем, які часто працюють зі змінними наборами параметрів, це є ключовою перевагою [35, с. 124].

3. Портативність. JSON-файли є незалежними від платформи. Вони можуть бути передані між системами без необхідності в специфічній інфраструктурі. Це робить їх ідеальними для експертних систем, які можуть працювати в різних середовищах або інтегруватися з іншими інструментами.

4. Легкість інтеграції. Враховуючи використання фреймворку qt, то JSON легко буде інтегрувати.

5. Простота доступу. Для невеликих обсягів конфігурацій JSON забезпечує високу швидкість читання, оскільки зчитування файлу на диску є швидшим і простішим, ніж надсилання SQL-запиту до сервера бази даних.

Недоліки JSON-файлів:

- неефективність при великих обсягах. JSON-файли не оптимізовані для роботи з великими обсягами даних. У таких випадках читання, модифікація або пошук стають менш ефективними через потребу у парсингу всього документа;
- обмеженість запитів. На відміну від SQL, JSON не підтримує складні запити, такі як фільтрація, агрегація чи об'єднання даних. Для обробки складних конфігурацій потрібні додаткові інструменти (наприклад, бібліотеки типу JSONPath);
- відсутність транзакцій. Якщо конфігурація змінюється в реальному часі, JSON не забезпечує транзакційності, тобто можливості виконати декілька змін у конфігурації атомарно. Це може призвести до ризиків втрати цілісності даних;
- масштабованість. У великих системах керування сотнями конфігурацій JSON може стати незручним, оскільки кожен файл потрібно обробляти окремо. Крім того, не існує централізованого контролю над файлами.

Переваги SQL для конфігурацій:

- потужні запити. SQL підтримує складну обробку даних, включаючи фільтрацію, агрегацію та об'єднання;
- транзакційність. Реляційні бази даних забезпечують атомарність операцій, що гарантує збереження цілісності даних навіть у разі збоїв;
- масштабованість. SQL добре працює з великими обсягами даних, що робить його ідеальним для систем, які потребують централізованого управління даними;
- безпека. Реляційні бази даних надають розширені механізми контролю доступу та захисту даних.

Недоліки SQL:

- складність налаштування. Для використання SQL потрібне розгортання СУБД, що додає витрати на інфраструктуру та адміністрування;

– складність внесення змін. У SQL зміна структури таблиць (наприклад, додавання нового поля) може вимагати суттєвої перебудови даних або навіть простою системи.

Чому JSON-файли підходять для конфігурацій?

1. Гнучкість. Експертні системи прогнозування контрольних заходів потребують частого оновлення конфігураційних даних. JSON дозволяє легко додавати нові параметри, не порушуючи структуру вже існуючих;

2. Придатність для автономних систем. JSON-файли можна використовувати локально, без потреби в сервері бази даних, що є перевагою в розподілених системах або автономних рішеннях;

3. Легкість у розробці. Для невеликих систем із невеликою кількістю конфігурацій JSON є оптимальним вибором, оскільки його легко обробляти безпосередньо в коді, без звернення до серверів бази даних.

Для конфігурацій у системах прогнозування контрольних заходів JSON-файли пропонують значну простоту, гнучкість і портативність. Вони дозволяють швидко вносити зміни до налаштувань без потреби в складній інфраструктурі чи серйозних витратах на адміністрування. Хоча SQL має переваги для складних запитів і роботи з великими даними, у випадку конфігурацій, які рідко оновлюються і мають фіксовану структуру, JSON є більш зручним і ефективним вибором.

3.3 Інструменти для розробки

У сучасну епоху інформаційних технологій освоєння мов програмування та вибір ефективних інструментів розробки є невід’ємною складовою ІТ. Зокрема, актуальним є питання вивчення мови програмування C++ та визначення найкращих інтегрованих середовищ розробки (IDE) [20].

Розробка програмного забезпечення вимагає вибору відповідного інтегрованого середовища розробки (IDE), яке забезпечить ефективну роботу,

зручність та необхідні інструменти для конкретних завдань. Розглянемо три популярні IDE: CLion, Visual Studio та Qt Creator, їхні особливості, переваги та недоліки, а також обґрунтуємо вибір Qt Creator для розробки експертної системи прогнозування контрольного заходу [27].

CLion – це кросплатформене IDE від компанії JetBrains, призначене для розробки на мовах C та C++. Воно підтримує CMake як основну систему збірки та надає розширені можливості для редагування коду, рефакторингу та налагодження [15].

Переваги CLion:

- розумне автодоповнення коду: CLion аналізує код у реальному часі, пропонуючи контекстно-залежні підказки, що підвищує продуктивність розробника;
- підтримка рефакторингу: Інструменти для безпечного перейменування змінних, функцій та інших елементів коду;
- інтеграція з системами контролю версій: Підтримка Git, SVN та інших систем, що спрощує командну роботу;
- підтримка плагінів: Можливість розширення функціональності за допомогою плагінів від JetBrains та спільноти.

Недоліки CLion:

- високі вимоги до ресурсів: CLion може споживати значну кількість оперативної пам'яті та процесорного часу, особливо на великих проєктах;
- обмежена підтримка інших мов: Хоча CLion підтримує C та C++, для розробки на інших мовах можуть знадобитися додаткові плагіни або інші IDE;
- відсутність повної інтеграції з Qt: Для розробки Qt-додатків можуть виникати труднощі з налаштуванням та інтеграцією, що потребує додаткових зусиль.

Visual Studio – це потужне IDE від Microsoft, яке підтримує широкий спектр мов програмування, включаючи C, C++, C#, Python та інші. Воно широко використовується для розробки під Windows, але також підтримує кросплатформену розробку [16].

Переваги Visual Studio:

- потужний налагоджувач: Visual Studio надає розширені можливості для налагодження, включаючи аналіз пам'яті, продуктивності та інші інструменти;
- інтеграція з Azure та іншими сервісами Microsoft: Спрощує розробку хмарних додатків та використання інших сервісів екосистеми Microsoft;
- розширюваність: Підтримка великої кількості плагінів та розширень для додаткової функціональності;
- зручний інтерфейс: Інтуїтивно зрозумілий інтерфейс, який можна налаштувати під потреби розробника [4, с. 56].

Недоліки Visual Studio:

- великий розмір та вимоги до системи: Visual Studio займає значний обсяг дискового простору та вимагає потужного обладнання для плавної роботи;
- складність налаштування для кросплатформеної розробки: Хоча Visual Studio підтримує кросплатформену розробку, налаштування може бути складним та вимагати додаткових інструментів;
- вартість: Професійні та корпоративні версії Visual Studio є комерційними продуктами з відповідною вартістю.

Qt Creator – це кросплатформене IDE, розроблене спеціально для роботи з фреймворком Qt, який використовується для створення графічних інтерфейсів та кросплатформених додатків на C++.

Переваги Qt Creator:

- глибока інтеграція з Qt: Qt Creator оптимізований для розробки з використанням Qt, надаючи інструменти для створення інтерфейсів, роботи з сигналами та слотами, а також управління ресурсами;
- кросплатформеність: Підтримка розробки та розгортання додатків на різних платформах, включаючи Windows, Linux та macOS;
- легкість та швидкість: Qt Creator є відносно легким IDE з швидким запуском та низькими вимогами до системних ресурсів;

- вбудовані інструменти для дизайну інтерфейсу: Інтегрований дизайнер форм дозволяє швидко створювати та редагувати графічні інтерфейси.

Недоліки Qt Creator:

- обмежена підтримка інших фреймворків: Qt Creator оптимізований для роботи з Qt, і використання його для інших фреймворків може бути менш ефективним;
- менша кількість плагінів: Порівняно з Visual Studio, Qt Creator має меншу екосистему плагінів та розширень;
- обмежені можливості налагодження: Хоча Qt Creator надає базові інструменти для налагодження, вони можуть бути менш потужними порівняно з Visual Studio.

Обґрунтування вибору Qt Creator для експертної системи прогнозування контрольного заходу.

1. Глибока інтеграція з Qt. Qt забезпечує простий і ефективний інструментарій для створення графічних інтерфейсів, які є невід’ємною частиною експертної системи. Завдяки цьому розробка інтуїтивно зрозумілого інтерфейсу для введення даних, відображення прогнозів та інтерактивної взаємодії з користувачем стає значно простішою. Вбудовані інструменти Qt Designer дозволяють розробляти інтерфейси без написання додаткового коду.

2. Кросплатформеність. Qt Creator дозволяє створювати додатки, які працюють на різних платформах (Windows, Linux, macOS) без необхідності значних змін у коді. Це критично важливо для експертної системи, яка може використовуватися на різних операційних системах залежно від потреб клієнтів або організацій [12].

3. Підтримка C++. Оскільки експертна система прогнозування контрольного заходу вимагає реалізації складних алгоритмів, ефективність і швидкодія коду стають критичними. Мова програмування C++, яка лежить в основі Qt Creator, дозволяє реалізовувати високопродуктивні рішення з низькорівневим доступом до апаратних ресурсів.

4. Модульність та розширюваність. Qt Creator спрощує організацію коду в окремі модулі, що дозволяє легко змінювати або додавати нові функції до експертної системи. Наприклад, можна розділити систему на модулі прогнозування, аналізу даних та управління базами знань. Це підвищує зручність розширення функціональності та підтримки програми.

5. Інтегровані інструменти налагодження. Хоча Qt Creator не має такого потужного налагоджувача, як Visual Studio, він надає необхідний функціонал для швидкого знаходження помилок у коді, особливо під час роботи з додатками Qt. Інтегрований налагоджувач підтримує як локальні, так і віддалені сесії налагодження, що важливо для розробки та тестування експертних систем [17].

6. Спрощення роботи з базами даних. Експертні системи часто використовують бази даних для зберігання знань, правил або історичних даних. Qt має вбудовану підтримку для роботи з різними системами управління базами даних (SQL, SQLite та іншими), що робить його зручним для реалізації подібного функціоналу.

7. Легкість налаштування та використання. На відміну від Visual Studio, яке потребує тривалого налаштування для роботи з Qt або іншими кросплатформеними бібліотеками, Qt Creator готовий до роботи «з коробки». Це економить час розробників та дозволяє швидше перейти до безпосередньої реалізації проекту.

Переваги використання Qt Creator для експертної системи:

- інтуїтивний інтерфейс розробки: Qt Creator надає інтуїтивно зрозумілі інструменти для створення графічного інтерфейсу користувача;
- потужні засоби для роботи з графікою: Завдяки Qt можливе створення візуалізацій та діаграм, що підвищує зручність використання системи;
- швидкий цикл розробки: Завдяки готовим шаблонам, засобам автоматизації та підтримці CMake розробка та тестування системи прискорюються;

- кросплатформеність: Одна кодова база може бути використана для створення додатків під різні операційні системи.
- вбудовані інструменти налагодження та аналізу продуктивності: Це дозволяє вчасно виявляти вузькі місця у продуктивності системи.

Недоліки використання Qt Creator:

- менша кількість плагінів у порівнянні з Visual Studio: Розширення функціональності може бути обмеженим у порівнянні з іншими IDE;
- обмежена інтеграція з деякими інструментами сторонніх виробників: Наприклад, для інтеграції з нестандартними бібліотеками може знадобитися додаткове налаштування;
- менш потужний налагоджувач: Для складних проєктів може не вистачати можливостей аналізу, доступних у Visual Studio;
- початкова складність для новачків: Хоча Qt Creator простий у використанні, початківці можуть потребувати часу для ознайомлення з його функціоналом.

Qt Creator є оптимальним вибором для розробки експертної системи прогнозування контрольного заходу завдяки своїй тісній інтеграції з фреймворком Qt, який забезпечує легкість створення інтерфейсів, кросплатформеність і потужну підтримку роботи з даними. Хоча Visual Studio та CLion мають свої переваги, Qt Creator забезпечує баланс між продуктивністю, зручністю та ефективністю розробки для задач, що включають прогнозування, обробку даних і інтерактивну взаємодію з користувачем.

Qt Creator поєднує у собі гнучкість, продуктивність та сучасний інструментарій, що робить його ідеальним вибором для створення надійної, масштабованої та продуктивної експертної системи.

РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ

4.1 Особливості програмної реалізації експертної системи

Кожен програмний продукт створюється за певним алгоритмом, що включає послідовність етапів і має свій життєвий цикл. Життєвий цикл програмного забезпечення охоплює весь період існування проєкту, системи чи продукту – від моменту виникнення ідеї до припинення його використання. Під час цього циклу визначаються, реалізуються та вдосконалюються процеси, які складають основу програмного продукту. Для проєктів, що розраховані на широке застосування або комерційне впровадження, зміст процесів і завдань повинен відповідати затвердженим стандартам життєвого циклу. Дотримання цих стандартів є не лише показником якості експертної системи, а й умовою її успішного функціонування. Кожен етап життєвого циклу може бути деталізований через додаткові підетапи, що дозволяє більш гнучко розподіляти завдання та оптимізувати їх виконання.

У межах розробки експертної системи для прогнозування контрольних заходів було виконано кілька важливих етапів:

1. Ознайомлення зі специфікою роботи та аналіз існуючих програмних засобів. На цьому етапі досліджувалася діяльність користувачів, аналізувалися вже наявні програмні рішення з метою визначення оптимальної структури та функціональних можливостей майбутнього програмного продукту.

2. Розробка структури конфігураційних файлів для збереження даних. Цей етап виявився критично важливим, оскільки від правильної структури конфігураційних файлів залежала як повнота збереження даних, так і можливість їхньої обробки в подальшому. Важливо враховувати, що після впровадження програмного продукту у реальні умови внесення змін у структуру файлів стає значно складнішим завданням, тому такі рішення мають бути продуманими на етапі проєктування.

3. Створення інтерфейсних елементів. Особлива увага приділялася розробці вікон для введення, редагування та виведення інформації. При цьому враховувалися ергономічні вимоги, що висуваються до побудови зручного графічного інтерфейсу.

4. Розробка графічного інтерфейсу користувача. Було створено інтуїтивно зрозумілий і зручний інтерфейс, який забезпечує ефективну взаємодію користувача з програмним забезпеченням.

Розробка експертної системи здійснювалася в середовищі візуальної розробки QtCreator із використанням мови програмування C++. У проєкті інформаційної системи було реалізовано низку функціональних форм (див. табл. 2.1). Задіяний підхід дозволив створити програмний продукт, що відповідає сучасним вимогам якості та зручності використання.

Один з головних методів які використовуються в програмі, а саме зчитування даних предметів.

```
void MainWindow::LoadData()
{
    //Зчитування списку файлів з предметами
    const auto subjects_cfgs = ParseJsonFile("subjects").value("subjects").toArray();
    //Зчитування кожного файлу по циклу
    for (const auto& subject_cfg : subjects_cfgs)
    {
        //Парсинг файлу
        auto subject_data = m_subject_loader->Load(subject_cfg.toString());
        //Зберігання даних предмету до списку
        m_subjects_data.insert(subject_data.first, subject_data.second);
    } }
```

Далі важливим методом є побудова рекомендацій.

```
void SubjectProcessor::Handle(QSharedPointer<Subject> subject_data_provider)
{
```

```

//Підрахунок відсотку виконання учбової програми
    m_global_done_procent = 0;
//Індексація рекомендацій
    int recomendation_index = 1;
//Вихідна змінна для рекомендацій
    QString global_recomendation;
//Отримання рекомендацій по лабораторних роботах
    QString labs_recomendation =
BuildLabsRecomendation(subject_data_provider->GetLabsData());
    if (!labs_recomendation.isEmpty())
    {
//Запис рекомендацій по лабораторних роботах у загальні рекомендації
        global_recomendation += QString::number(recomendation_index++) + ") " +
labs_recomendation;
    }
// Отримання рекомендацій по тестам
    QString tests_recomendation =
BuildTestsRecomendation(subject_data_provider->GetTestsData());
    if (!tests_recomendation.isEmpty())
    {
//Запис рекомендацій по тестам у загальні рекомендації
        global_recomendation += QString::number(recomendation_index++) + ") " +
tests_recomendation;
    }
//Отримання рекомендацій по інших роботах
    QString works_recomendation =
BuildWorksRecomendation(subject_data_provider->GetWorksData());
    if (!works_recomendation.isEmpty())
    {
//Запис рекомендацій по інших роботах у загальні рекомендації

```

```

        global_recomendation += QString::number(recomendation_index++) + ") " +
works_recomendation;
    }
//Відображення загальних рекомендацій
    m_ui->recomendationEdit->setPlainText(global_recomendation);
//Підрахунок та встановлення вірогідності(відсотка) підготовленості
студента до контрольного заходу
    m_ui->resultProgress->setValue(m_global_done_procent /
static_cast<float>(subject_data_provider->GetMaxPoints() *
subject_data_provider->GetMinAccess() / 100.0f) * 100);}

```

Серед можливих варіантів баз даних було обрано «реалізацію на основі JSON-файлів», оскільки цей підхід не вимагає встановлення додаткового програмного забезпечення для перегляду даних.

JSON базується на двох фундаментальних типах структур даних:

1. Колекція пар «ключ/значення». Ця концепція реалізується в різних мовах програмування під різними назвами: об'єкт, запис, структура, словник, хеш, іменованний список або асоціативний масив.

2. Упорядкований список значень. У багатьох мовах програмування він представлений у вигляді масиву, вектора, списку або послідовності.

Ці структури даних є універсальними, і майже всі сучасні мови програмування підтримують їх у тій чи іншій формі. Тому логічно, що формат даних, незалежний від конкретної мови програмування, побудований саме на основі цих широко використовуваних структур.

Користувачі можуть легко створювати JSON-файли із заданими параметрами, необхідними для роботи експертної системи, і одразу використовувати їх, просто помістивши у відповідну папку разом із бінарним файлом.

Приклад JSON файлу для предмета «Вища математика»:

```

{
    "subject_name": "Вища математика",

```

```

"min_access": 60,
"max_points": 200,
"labs":
{
    "weight": 30,
    "min_requirement": 60
},
"works":
[
    {
        "name": "Індивідуальне домашнє завдання",
        "weight": 50,
        "min_requirement": 60
    }
],
"tests":
[
    {
        "name": "Визначники",
        "weight": 15,
        "min_requirement": 60
    },
    {
        "name": "Матриці та СЛАР",
        "weight": 15,
        "min_requirement": 60
    }
]
}

```

Параметри JSON файлу:

– subject_name (назва предмету);

- min_access (мінімальний відсоток балів, який потрібен);
- max_points (максимальна кількість балів);
- labs (блок лабораторних робіт);
- works (блок спеціальних робіт: реферат, домашня робота і тощо);
- tests (блок тестів);
- name (назва спеціальних робіт/тестів/);
- weight (кількість балів за виконання тесту/роботи/лабораторних);
- min_requirement (мінімальний відсоток отримання за роботу/тесту/лабораторних).

Всі JSON файли зчитуються на початку роботи програми кожного разу. На рис. 4.1-4.2 відображені форми ЕС з нумерацією їх елементів.

Оцінка підготовленості студента до контрольного заходу

1

Номер групи 2 ПІБ студента 2

Вхідні данні

Кількість виконаних лабораторних робіт - 0% 3

Статус виконаних робіт

Оцінка [0-100] Тип роботи

4

Теоретичні тести

Оцінка [0-100] Список тем

4

Оцінити 5

Вихідні данні

Ймовірність допуску до контрольного заходу - 0% 6

Рекомендації

7

5 Зберегти у файл

Рисунок 4.1 – Інтерфейс ЕС

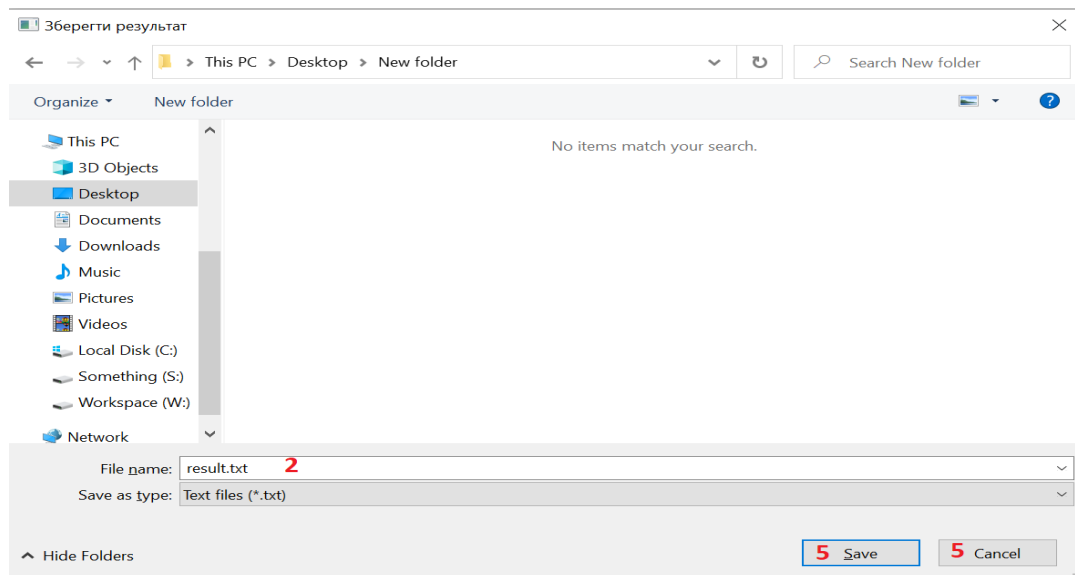


Рисунок 4.2 – Форма збереження рекомендацій

Опис елементів меню:

1. ComboBox – елемент вибору;
2. LineEdit – поле вводу;
3. SpinBox – поле накрутки значення;
4. TableWidget – таблиця;
5. Button – кнопка;
6. ProgressBar – прогрес;
7. TextEdit – поле відображення тексту (запис через клавіатуру заборонено).

На рис. 4.3 зображено елемент інтерфейсу, який забезпечує вибір предмета та ініціалізацію інтерфейсу відповідними даними про нього.

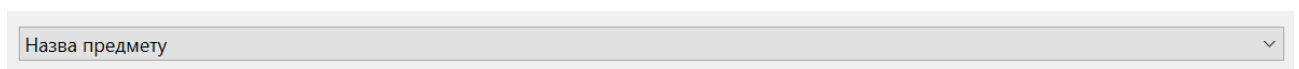


Рисунок 4.3 – Вибір предмету

На рис. 4.4 представлені текстові поля, які дозволяють ввести ПІБ студента та зазначити групу (останнє поле є необов'язковим для заповнення).

Номер групи	ПІБ студента
-------------	--------------

Рисунок 4.4 – Інформація про студента

На рис. 4.5 зображено елемент інтерфейсу, призначений для встановлення відсотка виконаних лабораторних робіт. Значення можна змінювати за допомогою кнопок, розташованих праворуч, або вводити вручну, встановивши курсор у поле біля символу відсотка.

Вхідні дані	
Кількість виконаних лабораторних робіт	55%

Рисунок 4.5 – Виконання лабораторних робіт

На рис. 4.6 продемонстровано таблицю, яка дозволяє задавати оцінки в межах від 0 до 100 для кожної окремої спеціальної роботи.

Статус виконаних робіт		
	Оцінка [0-100]	Тип роботи
1	0	Домашнє завдання

Рисунок 4.6 – Таблиця спеціальних робіт

На рис. 4.7 представлена таблиця, яка використовується для встановлення оцінок у діапазоні від 0 до 100 для кожного тесту.

Теоретичні тести		
	Оцінка [0-100]	Список тем
1	0	Безпека та сучасні тенденції розвитку баз даних
2	0	Логічне та фізичне проектування баз даних
3	0	Локальні та віддалені бази даних. Розподілені бази даних

Рисунок 4.7 – Виконання тестів

На рис. 4.8 продемонстровано кнопку, яка відповідає за аналіз введених даних щодо предмета та формування рекомендацій.

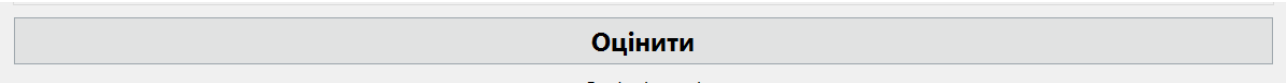


Рисунок 4.8 – Кнопка побудови рекомендацій

На рис. 4.9 зображено індикатор, що відображає прогрес ймовірності допуску.

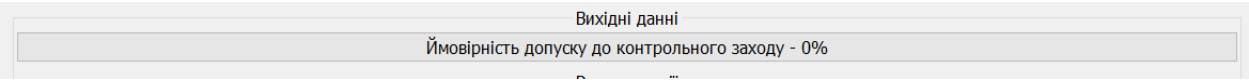


Рисунок 4.9 – Прогрес допуску до сесії

На рис. 4.10 представлено елемент інтерфейсу, призначений для відображення рекомендацій щодо підвищення ймовірності допуску до сесії.

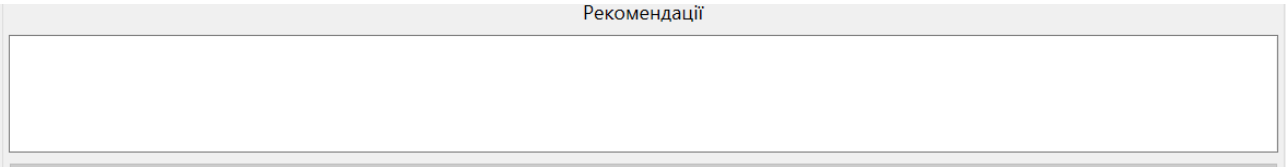


Рисунок 4.10 – Прогрес допуску до сесії

На рис. 4.11 показана кнопка за допомоги якої можна зберегти рекомендації у файл (доступно коли є хоч якісь рекомендації).

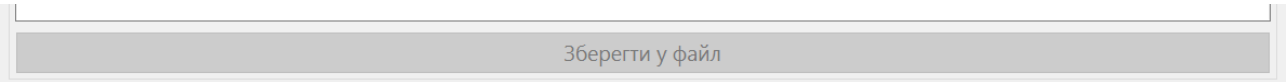


Рисунок 4.11 – Кнопка збереження рекомендацій у файл

На рис. 4.12 зображено текстове поле, за допомогою якого задається назва файлу, в який зберігаються рекомендації.



Рисунок 4.12 – Назва файлу

На рис. 4.13 зображено кнопки управління для вибору та підтвердження форми збереження файлу.

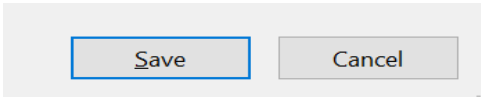


Рисунок 4.13 – Кнопки збереження файлу

На рис. 4.14 представлено алгоритм роботи експертної системи.

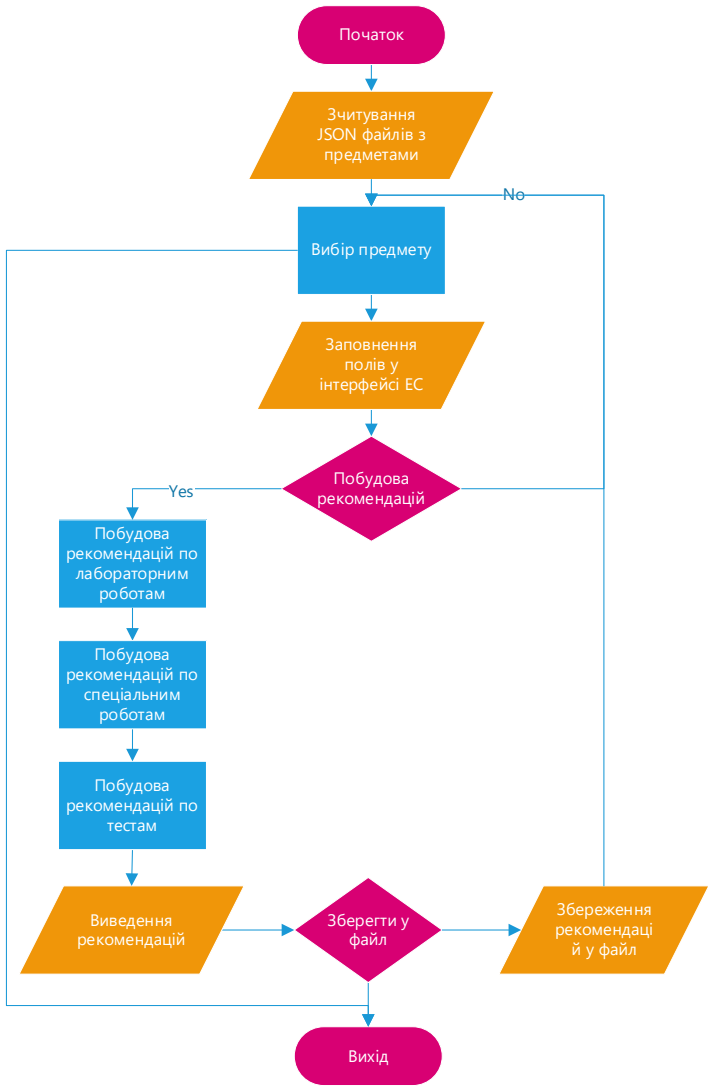


Рисунок 4.14 – Алгоритм роботи ЕС

На початку виконання програми завантажуються конфігураційні файли у форматі JSON, які містять дані про предмети. Формат JSON базується на двох основних типах структур даних. Перший тип представляє колекцію пар ключ-значення, яка в різних мовах програмування реалізується як об'єкт, запис, структура, словник, хеш, асоціативний масив або іменований список. Другий тип – це впорядкований набір значень, який зазвичай реалізується у вигляді масиву, списку, вектора або послідовності [5]. Після завантаження інформації користувач може обрати потрібний предмет для прогнозування та заповнити відповідні поля й таблиці в програмному інтерфейсі. Коли користувач захоче побудувати рекомендацію, то програма зробить аналіз вхідних даних. Спочатку проаналізує данні по лабораторним роботам та випише рекомендацію, далі так сам для спеціальних робіт та тестів. Після того як завершиться побудова рекомендацій відбудеться відображення у інтерфейсі програми. За необхідністю можна зберегти виведену рекомендацію у файл. Або заповнити по іншому поля у додатку чи вибрати новий предмет і повторити теж саме.

4.2 Функціональні можливості та інструкція користувача експертної системи

Після запуску додатку, буде відображено основний інтерфейс експертної системи (рис. 4.15).

Рисунок 4.15 – Основний інтерфейс експертної системи

- визначити % виконаних лабораторних робіт, через спеціальне поле прокрутки. Значення можна як ввести так і докрутити використовуючи мишку та кнопки управління з лівої сторони поля;
- виставити оцінки за спеціальні роботи заповнивши таблицю «Статус виконаних робіт»;
- виставити оцінки за теоретичні тести, заповнивши таблицю «Теоретичні тести»;
- оцінити/з прогнозувати готовність студента до контрольного заходу за допомогою кнопки «Оцінити»;
- зберегти рекомендації у файл натиснувши кнопку «Зберегти у файл».

Для того щоб отримати рекомендації та побачити на скільки студент готовий до контрольного заходу потрібно виконати наступні дії:

1. Вибрати предмет для заповнення таблиць інтерфейсу. Вигляд інтерфейсу після вибору предмета «Бази даних» (рис. 4.16).

Оцінка підготовленості студента до контрольного заходу

Бази даних

Номер групи: ПІБ студента

Вхідні дані

Кількість виконаних лабораторних робіт - 0%

Статус виконаних робіт

Оцінка [0-100]	Тип роботи
1 0	Домашнє завдання

Теоретичні тести

Оцінка [0-100]	Список тем
1 0	Безпека та сучасні тенденції розвитку баз даних
2 0	Логічне та фізичне проектування баз даних
3 0	Локальні та віддалені бази даних. Розподілені бази даних
4 0	Основні поняття курсу. Моделі даних. Мови запитів до реляційної моделі даних

Оцінити

Вихідні дані

Ймовірність допуску до контрольного заходу - 0%

Рекомендації

Зберегти у файл

Рисунок 4.16 – Вигляд інтерфейсу для предмету «Бази даних»

2. Ввести номер групи та ПІБ студента. Не є обов'язковим полем для заповнення.
3. Виставити % виконаних лабораторних робіт.
4. Заповнити перший стовпчик таблиці «Статус виконаних робіт» оцінками.
5. Заповнити перший стовпчик таблиці «Теоретичні тести» оцінками.
6. Натиснути кнопку «Оцінити». Приклад заповнених вхідних даних відображено на рис. 4.17.

Оцінка підготовленості студента до контрольного заходу

Бази даних

635п Павленко Нікіта Андрійович

Вхідні дані

Кількість виконаних лабораторних робіт - 50%

Статус виконаних робіт

Оцінка [0-100]	Тип роботи
1 45	Домашнє завдання

Теоретичні тести

Оцінка [0-100]	Список тем
1 15	Безпека та сучасні тенденції розвитку баз даних
2 60	Логічне та фізичне проектування баз даних
3 30	Локальні та віддалені бази даних. Розподілені бази даних
4 90	Основні поняття курсу. Моделі даних. Мови запитів до реляційної моделі даних

Оцінити

Вихідні дані

Ймовірність допуску до контрольного заходу - 0%

Рекомендації

Зберегти у файл

Рисунок 4.17 – Заповнені вхідні данні у додатку

Після натискання кнопки «Оцінити» буде виконано прогнозування та побудова рекомендацій щодо покращення ймовірності допуску до контрольного заходу (рис. 4.18).

Оцінка підготовленості студента до контрольного заходу

Бази даних

635п Павленко Нікіта Андрійович

Вхідні дані

Кількість виконаних лабораторних робіт - 50%

Статус виконаних робіт

Оцінка [0-100]	Тип роботи
1 45	Домашнє завдання

Теоретичні тести

Оцінка [0-100]	Список тем
1 15	Безпека та сучасні тенденції розвитку баз даних
2 60	Логічне та фізичне проектування баз даних
3 30	Локальні та віддалені бази даних. Розподілені бази даних
4 90	Основні поняття курсу. Моделі даних. Мови запитів до реляційної моделі даних

Оцінити

Вихідні дані

Ймовірність допуску до контрольного заходу - 72%

Рекомендації

1) Здати лабораторні роботи не менше 60%.
 2) Тести
 Повторити теми:
 - Безпека та сучасні тенденції розвитку баз даних;
 - Локальні та віддалені бази даних. Розподілені бази даних;
 3) Роботи
 Прездати/доробити:
 - Домашнє завдання;

Зберегти у файл

Рисунок 4.18 – Результат виконання прогнозування

Для збереження рекомендацій необхідно натиснути на кнопку «Зберегти у файл». Після чого відкриється вікно збереження рекомендацій у файл (рис. 4.19).

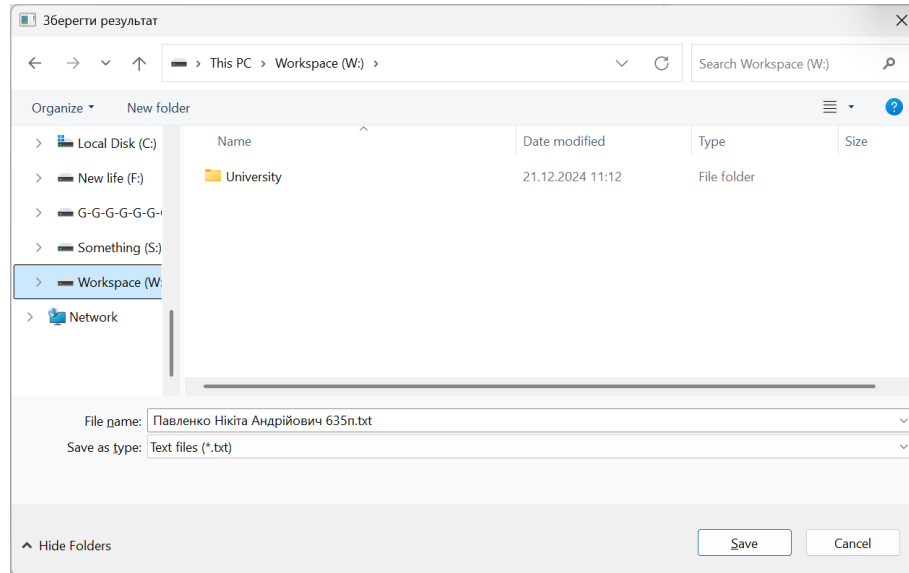


Рисунок 4.19 – Вікно збереження рекомендацій у файл

Після цього з’являються можливості:

- вибрати папку для збереження файлу;
- змінити назву файлу;
- зберегти файл натиснувши кнопку «Save»;
- відмінити зберігання, натиснувши кнопку «Cancel».

Функціонал експертної системи дозволяє виконувати прогнозування та формувати рекомендації для різних предметів. Отримана інформація може бути корисною як викладачам, так і студентам. За потреби користувачі мають можливість створювати необмежену кількість конфігураційних файлів для предметів.

4.3 Тестування експертної системи

Забезпечення якості програмного забезпечення (ПЗ) є однією з основних задач розробників. Для визначення якості продукту важливо враховувати кілька ключових аспектів:

- якість визначається як міра відповідності системи, компонента чи процесу заданим вимогам, очікуванням або потребам користувача. Відображення надійності системи забезпечується через використання так званих атрибутів якості, які описують її властивості;
- тестування є найефективнішим способом забезпечення якості ПЗ, яке допомагає виявляти помилки та неузгодженості. Його мета – знайти та локалізувати помилки, а також сприяти їх усуненню.

У минулому, коли ПЗ створювалося переважно для наукових або військових потреб, розробники самотійно виконували тестування. Цей процес був трудомістким, оскільки всі етапи та результати документувалися, а методи тестування обмежувалися статичними й динамічними підходами. Проте ефективність таких підходів була низькою через високий рівень людського фактору та ризиків [23, с. 186].

Сучасне тестування значно еволюціонувало, ставши окремою складовою процесу розробки зі своїми функціями, принципами й рівнями. Тепер більшість розробників і компаній визнають важливість тестування, інтегруючи написання тестів до процесу розробки нового функціоналу. Це дозволяє впевнитися у відсутності конфліктів між новим і вже існуючим кодом [23, с. 187].

Тестування охоплює не лише перевірку коду на працездатність і стійкість до відмов, але також оцінює дизайн, зручність, інтуїтивність, якість документації, а також процеси встановлення і видалення програми.

Одним із найпопулярніших методів тестування є створення тестових сценаріїв. Цей підхід дозволяє перевірити всі аспекти функціонування певного фрагмента коду, включаючи позитивні та негативні сценарії. Такі тести,

створені розробниками або окремими спеціалістами, забезпечують передбачуваність і стабільність роботи продукту [23, с. 187].

Проте існують аргументи щодо недоцільності повного тестування:

- велика кількість можливих вхідних даних і результатів;
- значна кількість проходів по програмі;
- суб'єктивність специфікації, коли деякі помилки є задумом розробників.

Попри це, тестування залишається важливим, оскільки виявлення дефектів на ранніх етапах економить час і ресурси.

Сучасне тестування значно еволюціонувало, ставши окремою складовою процесу розробки зі своїми функціями, принципами й рівнями. Тепер більшість розробників і компаній визнають важливість тестування, інтегруючи написання тестів до процесу розробки нового функціоналу. Це дозволяє впевнитися у відсутності конфліктів між новим і вже існуючим кодом.

Тестування охоплює не лише перевірку коду на працездатність і стійкість до відмов, але також оцінює дизайн, зручність, інтуїтивність, якість документації, а також процеси встановлення і видалення програми. Методи тестування включають формальні та неформальні підходи. Формальна перевірка допомагає довести відсутність дефектів у межах обраного методу, хоча повна гарантія коректності ПЗ неможлива через людський фактор. Тестування також включає як автоматизовані, так і ручні техніки, кожна з яких має свої переваги та обмеження [1, с. 8].

Види тестування:

1. Юніт-тестування – перевірка окремих методів/функцій.
2. Модульне тестування – перевірка класів.
3. Інтеграційне тестування – оцінка взаємодії між компонентами.
4. Системне тестування – перевірка ПЗ як єдиного цілого.
5. Приймальне тестування – підтвердження готовності продукту до використання.

Під час тестування може оцінюватися:

- відповідність вимогам, визначеним проектувальниками та розробниками;
- правильність реакції програми на всі можливі вхідні дані;
- здатність виконувати функції у встановлені часові рамки;
- зручність використання;
- сумісність з іншими програмними продуктами та операційними системами;
- відповідність завданням замовника.

Якість програмного забезпечення є суб'єктивною категорією, яка не має абсолютного значення. Тому процес тестування, який передбачає своєчасне виявлення помилок і дефектів, не може повністю гарантувати коректність роботи програми. Тестування виконує функцію порівняння стану і поведінки продукту зі специфікацією [22, с. 94].

Традиційний підхід до тестування програмного забезпечення базується на попередньому визначенні критеріїв тестування та використанні допоміжних інструментів. Критерії тестування представляють собою предикати, які відповідають визначеному набору тестових випадків, що використовуються для створення тестових даних. Структурні критерії зосереджуються на особливостях архітектури програмного забезпечення, його коді та базуються на графічному відображенні потоку управління в програмі [8, с. 110].

Однак, на відміну від звичайного програмування, паралельне програмування вимагає врахування специфічних аспектів, таких як взаємозв'язок між змінними й потоками, синхронізація та недетермінованість, що ускладнює процес тестування. У зв'язку з цим тестування програмного забезпечення повинно враховувати як лінійні аспекти створення коду, так і особливості його паралелізації [8, с. 110].

Першим тестуванням яке буде проводитись для експертної системи це юніт-тести. Тести будуть для виконані для методу LoadSubjects. План тестів наведений у табл. 4.1.

Таблиця 4.1 – План юніт-тестів для методу LoadSubjects

№ Тесту	Опис	Вхідні дані	Очікуваний результат	Примітки
1	2	3	4	5
1	Перевірка правильної обробки непустого JSON-файлу	JSON-файл із ключем subjects, що містить масив строк: ["math", "data_base"]	Метод m_subject_loader->Load викликається двічі з параметрами "math" і "data_base". Дані додаються до m_subjects_data	Має перевіряти, чи всі дані з файлу завантажуються коректно
2	Обробка порожнього JSON-файлу	JSON-файл із ключем subjects, що містить порожній масив: []	Метод m_subject_loader->Load не викликається. m_subjects_data залишається порожнім	Має перевіряти, чи метод обробляє порожні дані без помилок
3	Обробка відсутнього ключа subjects	JSON-файл без ключа subjects	Метод ParseJsonFile повертає порожнє значення або викидає виняток залишається незмінним	Має перевіряти, чи система коректно обробляє неправильну структуру файлу

Продовження таблиці 4.1

1	2	3	4	5
4	Обробка невалідного JSON-файлу	Некоректний JSON-файл (наприклад, пошкоджений формат)	Метод ParseJsonFile повертає порожнє значення або викидає виняток	Має забезпечувати, що метод LoadData не викликає аварійного завершення програми

Результат проведення юніт тестів представлений у табл. 4.2.

Таблиця 4.2 – Результат проведення юніт-тестів

№ тесту	Вхідні дані	Очікуваний результат	Отриманий результат	Статус
1	2	3	4	5
1	Правильний JSON	Дані завантажені, m_subject_loader->Load викликається двічі	Дані завантажені, метод викликано двічі	Успішно
2	Порожній JSON	Дані не завантажуються, m_subjects_data залишається порожнім	Дані не завантажуються, m_subjects_data порожній	Успішно

Продовження таблиці 4.2

1	2	3	4	5
3	Відсутній ключ subjects	Дані не завантажуються, m_subjects_data залишається незмінним	Дані не завантажуються, m_subjects_data незмінний	Успішно
4	Некоректний JSON	Програма не аварійно завершується	Програма не аварійно завершується	Успішно

Наступним етапом тестування є модульне тестування класу SubjectProcessor, план тестів відображено у табл. 4.3.

Таблиця 4.3 – Опис модульного тестування

№ Тесту	Опис	Вхідні дані	Очікуваний результат	Примітки
1	2	3	4	5
1	Перевірка обробки коректних даних для лаборатор них робіт	labs_data з min_requirement= 50, weight=30. Відсоток виконання = 40	m_global_done_pro cent збільшений на 12%. Текст: "Здати лабораторні роботи не менше 50%.\n"	Перевіряє логіку BuildLabsRec omendation
2	Перевірка, коли лаборатор ні виконані	labs_data з min_requirement= 50, weight=30. Відсоток виконання = 60	m_global_done_pro cent збільшений на 15%. Текст рекомендації порожній	Перевірка граничних значень

Продовження таблиці 4.3

1	2	3	4	5
3	Перевірка тестів, які не розпочаті	Таблиця тестів містить тест з рейтингом 0	Рекомендація включає "Вивчити теми: - [Назва тесту];"	Перевіряє логіку BuildTestsRecommendation
4	Тести з низьким рейтингом	Тест із рейтингом 20, мінімально необхідно 50	Рекомендація включає "Повторити теми: - [Назва тесту];". m_global_done_percent збільшений відповідно до рейтингу	-
5	Роботи, які не виконані	Таблиця робіт містить роботу з рейтингом 0	Рекомендація включає "Здати: - [Назва роботи];"	Перевірка логіки BuildWorksRecommendation
6	Повна обробка даних (лабораторні, тести, роботи)	Всі дані передаються з валідними значеннями	Глобальна рекомендація містить текст для лабораторних, тестів і робіт. Прогрес-бар оновлений коректно	Перевірка інтеграції

Продовження таблиці 4.3

1	2	3	4	5
7	Некоректні дані в таблиці тестів	Тест відсутній у tests_data	Виняток не кидається, m_global_done_percent не змінюється	Обробка відсутності даних
8	Перевірка прогрес-бару	Сумарні бали: 70/100	Значення прогрес-бару = 70	Перевіряє коректність оновлення UI

Результат проведення модульного показано у табл. 4.4.

Таблиця 4.4 – Результат проведення модульного тестування

№ Тесту	Очікуваний результат	Отриманий результат	Статус
1	2	3	4
1	Текст рекомендації: "Здати лабораторні роботи не менше 50%"	Текст рекомендації: "Здати лабораторні роботи не менше 50%."	Успішно
2	Текст рекомендації: "" (порожній)	Текст рекомендації: ""	Успішно
3	Рекомендація включає "Вивчити теми: - Test1"	Текст рекомендації: ""	Успішно
4	Рекомендація включає "Повторити теми: - Test2"	Рекомендація включає "Повторити теми: - Test2"	Успішно
5	Рекомендація включає "Здати: - Work1"	Рекомендація включає "Здати: - Work1"	Успішно

Продовження таблиці 4.4

1	2	3	4
6	Коректне формування глобальної рекомендації, прогрес- бар = 75%	Глобальна рекомендація коректна, прогрес-бар = 75%	Успішно
7	Тест відсутній у tests_data Виняток не кидається, m_global_done_procent незмінний	Виняток не кинуто, m_global_done_procent незмінний	Успішно
8	Прогрес-бар = 70%	Прогрес-бар = 70%	Успішно

ВИСНОВКИ

Сучасні технології відіграють важливу роль в освітньому процесі, автоматизуючи прогнозування успішності студентів та підготовку до контрольних заходів, таких як іспити чи тестування. Впровадження таких систем дозволяє викладачам та адміністрації оптимізувати навчальний процес, надаючи можливість вчасного реагування на можливі труднощі учнів і підвищуючи загальну ефективність підготовки.

У рамках проекту було розроблено експертну систему для прогнозування результатів контрольних заходів, що базується на використанні технологій C++, Qt та JSON. Система забезпечує автоматичний аналіз навчальних даних, формуючи прогнози успішності та рекомендації щодо підготовки до іспитів. Завдяки модульній архітектурі її можна легко адаптувати до змінюваних вимог навчальних закладів та специфічних потреб окремих користувачів.

В основі роботи системи лежить модуль прогнозування, який аналізує історичні дані про успішність учнів або студентів, їхню активність протягом навчального періоду, складність контрольного заходу та інші ключові показники. Інтерфейс користувача, розроблений на базі Qt, забезпечує інтуїтивно зрозумілу взаємодію з системою, дозволяючи отримувати доступ до прогнозів, налаштовувати параметри аналізу та переглядати детальні рекомендації щодо підготовки.

Система використовує JSON для ефективного зберігання, обміну та інтеграції даних із зовнішніми системами, такими як електронні журнали та бази даних успішності. Це дає змогу динамічно оновлювати інформацію та підвищувати гнучкість системи.

Результати тестування продемонстрували високу точність прогнозів, що стало можливим завдяки комплексному аналізу різних параметрів. Продуктивність системи залишалася стабільною навіть при обробці великої

кількості вхідних даних, а інтерфейс був оцінений як зручний та простий у використанні для викладачів та адміністрації.

Розроблена експертна система відповідає сучасним освітнім викликам і сприяє підвищенню результативності навчального процесу. Її можна ефективно застосовувати для моніторингу успішності, планування підготовки до іспитів та своєчасного прийняття рішень щодо підтримки студентів чи учнів.

Таким чином, мета дипломної роботи, що полягала у розробці експертної системи для прогнозування контрольного заходу та перевірки коректності її роботи, була досягнута, а всі поставлені у ході виконання дипломної роботи задачі виконані.

СПИСОК ЛІТЕРАТУРИ

1. Авраменко, А. С.; Авраменко, В. С.; Косенюк, Г. В. Тестування програмного забезпечення: навч. посіб. 2017. С. 284.
2. Бандурка О., Свинчук О., Кривда Д. Інформаційна система дослідження успішності здобувачів освіти. Збірник матеріалів XI Всеукраїнської науково-практичної конференції молодих вчених «Наукова молодь-2023». 2023. С. 14.
3. Бородіна О. О. Використання мови програмування C++ при створенні сучасних програмних продуктів: дис. канд. техн. наук. Чернігівський національний технологічний університет, 2019. С. 250-252.
4. Бородіна О. О., Гафіяк А. М. Алгоритмізація та програмування: навчальний посібник. 2018. С. 240.
5. Вступ в JSON. JSON: веб-сайт. URL: <https://www.json.org/json-uk.html>
6. Затхей, В. А. Використання експертних систем в дистанційному навчанні. *Системи обробки інформації*. 2015. № 4. С. 144–146.
7. Зінов'єва О. Г. Використання CASE-засобів для проектування інформаційних систем. *Українські студії в європейському контексті*. 2023. № 7. С. 220–227.
8. Лановий О. Ф. Візуалізація в методах тестування програмного забезпечення : дис. канд. техн. наук. 2017. С. 110–111.
9. Лубко Д. В. Використання ІКТ для контролю рівня навчальних досягнень учнів і студентів. *Інформаційно-комунікаційні технології навчання*. 2016. С. 60–65.
10. Лубко Д. В., Шаров С. В. Напрямки використання інтелектуальних систем в освітньому процесі. *Українські студії в європейському контексті*. 2021. №3. С. 305–310.
11. Месюра В. Ш., Яровий А. А., Арсенюк І. Р. Експертні системи. Частина 1: навч. посіб. Вінниця: ВНТУ, 2006. 114 с.

12. Огляд Qt Creator IDE. CompareCamp: веб-сайт. URL: <https://comparecamp.com/qt-creator-ide-review-pricing-pros-cons-features>
13. Олійник Т. О. Використання інформаційно-комунікаційних технологій для оцінювання рівня навчальних досягнень студентів ВНЗ. *Інформаційні технології і засоби навчання*. 2014. № 42(4). С. 85–93.
14. Освітня платформа Udemy. Інформативні технології: веб-сайт. URL: <https://it.pdpu.edu.ua/osvitni-platformy/udemy>
15. Порівняння CLion і Qt Creator. StackShare: веб-сайт. URL: <https://www.stackshare.io/stackups/clion-vs-qt-creator>
16. Порівняння Visual Studio і Qt Creator. StackShare: веб-сайт. URL: <https://www.stackshare.io/stackups/qt-creator-vs-visual-studio>
17. Порівняння Visual Studio IDE та Qt Creator IDE. FinancesOnline: веб-сайт. URL: <https://comparisons.financesonline.com/visual-studio-ide-vs-qt-creator-ide>
18. Притика О. В., Юрченко А. О. Про особливості мови програмування C+. *Інформаційні технології в професійній діяльності*. 2021. С. 155–156.
19. Про онлайн навчання на платформі Coursera. Одеський обласний центр зайнятості: веб-сайт. URL: <https://ode.dcz.gov.ua/publikaciya/pro-onlayn-navchannya-na-platformi-coursera>
20. Прухницький, В. С., Махенько, Я. Д., Стельмашенко, Я. А. (2024) Щодо питання вибору оптимальних інтегрованих середовищ розробки в ІТ-освіті. In: Актуальні питання сучасної інформатики: Матеріали доповідей VIII Всеукраїнської науково-практичної конференції з міжнародною участю «Сучасні інформаційні технології в освіті та науці», 16-17 листопада 2023 р, Житомир. С. 175–178.
21. Ротань О. С. Особливості побудови та організації експертних систем. *Автоматизація та комп'ютерно-інтегровані технології*. 2021. С. 253–255.
22. Руда О. А., Моденов Ю. Б. Методи та моделі тестування програмного забезпечення. *Problems of Informatization and Control*. 2013. Т. 2, № 42. С. 93–98.

23. Рудик І., Шевчук Б. Потреба тестування програмного забезпечення. *Матеріали конференцій МНЛ*. 2022. С. 186–188.
24. Рукіна О. М. Використання експертної системи в системі дистанційного навчання. *Ukrainian Journal of Educational Studies and Information Technology*. 2017. С. 267–270.
25. Сафонов К. С., Сердюк І. М. Передумови використання експертних систем в освітньому процесі. *Стан освітнього процесу в умовах викликів сьогодення*. 2021. С. 74–75.
26. Степанов Д. В. Інструментарій Qt для розробки програмного забезпечення. *Кібербезпека в сучасному світі*. 2020. С. 220–224.
27. ТОП-7 популярних IDE для програмування на C++. ITVDN: веб-сайт. URL: <https://itvdn.com/ua/blog/article/cplsppls-top7>
28. Шаров С. В., Еміратлі А. Р. Використання експертних систем для підтримки навчального процесу. *Science, research, development, technics and technology*. 2018. С. 28–31.
29. Шаров С. В., Скрипка С. О. Використання мови UML для інфологічного моделювання реляційної бази даних. *Інформаційне суспільство*. 2014. № 7. С. 5–9.
30. Шаров С. В., Шамардак О. А. Концептуальне моделювання інформаційної системи за допомогою мови UML. *Інформаційне суспільство*. 2014. № 7. С. 10–14.
31. Шевчук О. Б. Педагогічні принципи проектування та розробки експертних систем навчання. *Наукові записки ТНПУ*. 2016. №1. С. 38–44.
32. Що робить платформу Open edX® унікальною? Open Edx платформа: веб-сайт. URL: <https://openedx.org/uk/blog/what-makes-the-open-edx-platform-unique/>
33. Ahmed, H. M. M., Sorour, S. E. Classification-driven intelligent system for automated evaluation of higher education exam paper quality. *Education and Information Technologies*. 2024. С. 1-27.

34. Arkhipova T. L. Інтелектуальні інформаційні системи в економіці та освіті. *Journal of Information Technologies in Education (ITE)*. 2008. № 1. С. 9-12.
35. Bourhis P., et al. JSON: data model, query languages and schema specification. Proceedings of the 36th ACM symposium on principles of database systems. 2017. С. 123–135.
36. Caro Valadez C. R. Developing an application user interface: comparing the use of web technologies versus QT and QML. 2017. С. 1-41.
37. Emmanuel, U., Collins, U. A Theoretical Framework for an Intelligent Result Processing Expert System in Nigerian Universities. Vol. 5. С. 9-16.
38. Erricolo D. C++, a better language for engineering applications. *IEEE Antennas and Propagation Magazine*. 2000. Т. 42, № 4. С. 95–101.
39. Giri W., Giri S., Shakya S. An Overview of C and C++: A Case of Engineering Faculty. 2019. С. 497–501.
40. Hatzilygeroudis, I., Karatrantou, A., Pierrakeas, C. PASS: An expert system with certainty factors for predicting student success. *Knowledge-Based Intelligent Information and Engineering Systems: 8th International Conference, KES 2004, Wellington, New Zealand*. 2004, С. 292–298.
41. Khairy, D., Ahmed, H. M. M., Sorour, S. E. Prediction of student exam performance using data mining classification algorithms. *Education and Information Technologies*. 2024, С. 1-25.
42. Kucher V. The development of a cross platform by using Qt framework. *Перспективи розвитку професійно спрямованих мовних компетенцій в сучасній науці*. 2016. С. 76–78.
43. Lv T., Yan P., He W. Survey on JSON data modelling. *Journal of Physics: Conference Series*. IOP Publishing, 2018. С. 012101.
44. Presley A., Liles D. H. The use of IDEF0 for the design and specification of methodologies. Proceedings of the 4th industrial engineering research conference. 1995. С. 442–448.

45. Sharov S. et al. Analysis of the MOOC Capabilities for Student Training in the Humanities. *International Journal of Emerging Technologies in Learning (IJET)*. 2021. T. 16. № 22. C. 113–128.
46. Tsironis L., Gentsos A., Moustakis V. Empowerment the IDEF0 modeling language. *International Journal of Business and Management*. 2008. T. 3, № 5. C. 109–118.

ДОДАТОК А

Лістинг коду програми

Subject.hpp

```

#pragma once
#include <types.hpp>
#include <QJsonObject>
#include <QMap>
#include <QString>
class Subject
{
public:
    void SetLabsData(const LabsData& labs_data);
    const LabsData& GetLabsData() const;
    void SetWorksData(const QMap<QString, WorkData>& works_data);
    const QMap<QString, WorkData>& GetWorksData() const;
    void SetTestsData(const QMap<QString, TestData>& tests_data);
    const QMap<QString, TestData>& GetTestsData() const;
    void SetMaxPoints(const int max_points);
    int GetMaxPoints() const;
    void SetMinAccess(const int min_access);
    int GetMinAccess() const;
protected:
    QMap<QString, WorkData> m_works_data;
    QMap<QString, TestData> m_tests_data;
    LabsData m_labs_data;
    int m_max_points { 0 };
    int m_min_access { 0 };};

```

Subject.cpp

```

#include <subject.hpp>
void Subject::SetLabsData(const LabsData& labs_data)
{
    m_labs_data = labs_data;
}
const LabsData& Subject::GetLabsData() const
{
    return m_labs_data;
}
void Subject::SetWorksData(const QMap<QString, WorkData>& works_data)
{
    m_works_data = works_data;
}
const QMap<QString, WorkData>& Subject::GetWorksData() const
{
    return m_works_data;
}
void Subject::SetTestsData(const QMap<QString, TestData>& tests_data)
{
    m_tests_data = tests_data;
}

```

```

const QMap<QString, TestData>& Subject::GetTestsData() const
{
    return m_tests_data;
}
void Subject::SetMaxPoints(const int max_points)
{
    m_max_points = max_points;
}
int Subject::GetMaxPoints() const
{
    return m_max_points;
}
void Subject::SetMinAccess(const int min_access)
{
    m_min_access = min_access;
}
int Subject::GetMinAccess() const
{
    return m_min_access;
}

```

SubjectLoader.hpp

```

#pragma once
#include <types.hpp>
#include <QString>
#include <QPair>
#include <QSharedPointer>

class Subject;

class SubjectLoader
{
public:
    QPair<QString, QSharedPointer<Subject>> Load(const QString& config_file_name) const;
private:
    LabsData LoadLabs(const QJsonObject &cfg) const;
    QMap<QString, WorkData> LoadWorks(const QJsonObject &cfg) const;
    QMap<QString, TestData> LoadTests(const QJsonObject &cfg) const;
}

```

SubjectLoader.cpp

```

#include <subject_loader.hpp>
#include <subject.hpp>
#include <tools.hpp>
#include <QFile>
#include <QMap>
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonArray>
#include <exception>
QPair<QString, QSharedPointer<Subject>> SubjectLoader::Load(const QString&
config_file_name) const
{
    const auto config = ParseJsonFile(config_file_name);
}

```

```

    QSharedPointer<Subject> subject = QSharedPointer<Subject>::create();
    subject->SetLabsData(LoadLabs(config));
    subject->SetWorksData(LoadWorks(config));
    subject->SetTestsData(LoadTests(config));
    subject->SetMaxPoints(config.value("max_points").toInt());
    subject->SetMinAccess(config.value("min_access").toInt());
    const auto subject_name = config.value("subject_name").toString();
    return qMakePair(subject_name, subject);
}

LabsData SubjectLoader::LoadLabs(const QJsonObject& cfg) const
{
    const auto labs_cfg = cfg.value("labs").toObject();
    LabsData labs_data;
    labs_data.weight = labs_cfg.value("weight").toInt();
    labs_data.min_requirement = labs_cfg.value("min_requirement").toInt();
    return labs_data;
}

QMap<QString, WorkData> SubjectLoader::LoadWorks(const QJsonObject& cfg) const
{
    QJsonArray works_cfg = cfg.value("works").toArray();

    QMap<QString, WorkData> works_data;
    WorkData work_data;
    foreach (const auto work, works_cfg)
    {
        work_data.weight = work.toObject().value("weight").toInt();
        work_data.min_requirement = work.toObject().value("min_requirement").toInt();
        works_data.insert(work.toObject().value("name").toString(), work_data);
    }
    return works_data;
}

QMap<QString, TestData> SubjectLoader::LoadTests(const QJsonObject& cfg) const
{
    QJsonArray tests_cfg = cfg.value("tests").toArray();

    QMap<QString, TestData> tests_data;
    TestData test_data;
    foreach (const auto test, tests_cfg)
    {
        test_data.weight = test.toObject().value("weight").toInt();
        test_data.min_requirement = test.toObject().value("min_requirement").toInt();
        tests_data.insert(test.toObject().value("name").toString(), test_data);
    }
    return tests_data;
}

```

MainWindow.hpp

```

#pragma once
#include <QMainWindow>
#include <QMap>
QT_BEGIN_NAMESPACE
namespace Ui {
class MainWindow;
}

```

```

QT_END_NAMESPACE
class SubjectProcessor;
class Subject;
class SubjectLoader;
class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
private slots:
    void on_submitButton_clicked();
    void on_subjectComboBox_currentIndexChanged(int index);
    void on_saveButton_clicked();
    void on_recomendationEdit_textChanged();
private:
    void LoadData();
    void InitSubject(const QString subject_name);
    void InitSubjects();
private:
    QMap<QString, QSharedPointer<Subject>> m_subjects_data;
    QSharedPointer<SubjectProcessor> m_subject_processor;
    QSharedPointer<SubjectLoader> m_subject_loader;
    QSharedPointer<Ui::MainWindow> m_ui;};

```

MainWindow.cpp

```

#include <main_window.hpp>
#include <./ui_main_window.h>
#include <qglobal.h>
#include <subject_loader.hpp>
#include <subject.hpp>
#include <subject_processor.hpp>
#include <tools.hpp>
#include <QFile>
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonValueRef>
#include <QJsonArray>
#include <QTextStream>
#include <QFileDialog>
#include <QDataStream>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , m_ui(new Ui::MainWindow)
{
    m_ui->setupUi(this);
    m_subject_loader = QSharedPointer<SubjectLoader>::create();
    m_subject_processor = QSharedPointer<SubjectProcessor>::create(m_ui);
    LoadData();
    InitSubjects();
}
MainWindow::~MainWindow()

```

```

{
    m_subjects_data.clear();
}
void MainWindow::InitSubject(const QString subject_name)
{
    auto subject_data = m_subjects_data.value(subject_name);
    m_ui->worksTable->setRowCount(0);
    m_ui->testsTable->setRowCount(0);
    int index = 0;
    foreach (auto work, subject_data->GetWorksData().keys())
    {
        m_ui->worksTable->insertRow(index);
        const auto work_name_item = new QTableWidgetItem(work);
        work_name_item->setFlags(work_name_item->flags() ^ Qt::ItemIsEditable);
        m_ui->worksTable->setItem(index, 1, work_name_item);
        const auto work_rating_item = new QTableWidgetItem(QString::number(0));
        m_ui->worksTable->setItem(index, 0, work_rating_item);
        ++index;
    }
    index = 0;
    foreach (auto test, subject_data->GetTestsData().keys())
    {
        m_ui->testsTable->insertRow(index);
        const auto test_name_item = new QTableWidgetItem(test);
        test_name_item->setFlags(test_name_item->flags() ^ Qt::ItemIsEditable);
        m_ui->testsTable->setItem(index, 1, test_name_item);
        const auto test_rating_item = new QTableWidgetItem(QString::number(0));
        m_ui->testsTable->setItem(index, 0, test_rating_item);
        ++index;
    }
}
void MainWindow::InitSubjects()
{
    m_ui->subjectComboBox->clear();
    foreach (const auto& subject, m_subjects_data.keys())
    {
        m_ui->subjectComboBox->addItem(subject);
    }
}
void MainWindow::on_submitButton_clicked()
{
    const auto subject = m_subjects_data[m_ui->subjectComboBox->currentText()];
    m_subject_processor->Handle(subject);
}
void MainWindow::on_subjectComboBox_currentIndexChanged(int index)
{
    const auto subject_number = m_ui->subjectComboBox->currentIndex();
    if (subject_number >= 0)
    {
        InitSubject(m_ui->subjectComboBox->itemText(subject_number));
    }
    m_ui->recomendationEdit->setPlainText("");
    m_ui->resultProgress->setValue(0);
}

```

```

void MainWindow::on_saveButton_clicked()
{
    const auto group_number_text = m_ui->groupNumberEdit->text();
    const auto student_name = m_ui->studentNameEdit->text();
    const auto recomendation_text = m_ui->recomendationEdit->toPlainText();
    QString file_path = "result";
    if (!recomendation_text.isEmpty())
    {
        if (!student_name.isEmpty() && !group_number_text.isEmpty())
        {
            file_path = QDir::currentPath() + '/' + student_name + ' ' + group_number_text;
        }
        QString filename = QFileDialog::getSaveFileName(
            this,
            tr("Зберегти результат"),
            file_path,
            tr("Text files (*.txt)") );
        if( filename.isNull() )
        {
            qDebug() << filename;
            return;
        }
        QFile recomendation_file(filename);
        if (recomendation_file.open(QIODevice::WriteOnly | QIODevice::Text))
        {
            QTextStream out(&recomendation_file);
            out << recomendation_text;
            out.flush();
            recomendation_file.close();
        }
    }
}

void MainWindow::on_recomendationEdit_textChanged()
{
    if (m_ui->recomendationEdit->toPlainText().isEmpty())
    {
        m_ui->saveButton->setDisabled(true);
    }
    else
    {
        m_ui->saveButton->setDisabled(false);
    }
}

void MainWindow::LoadData()
{
    try
    {
        const auto subjects_cfgs = ParseJsonFile("subjects").value("subjects").toArray();
        for (const auto& subject_cfg : subjects_cfgs)
        {
            auto subject_data = m_subject_loader->Load(subject_cfg.toString());
            m_subjects_data.insert(subject_data.first, subject_data.second);
        }
    }
    catch (...)
}

```

ДОДАТОК Б

Стаття

*Матеріали IV Всеукраїнської науково-практичної Інтернет-конференції
«Сучасні комп'ютерні та інформаційні системи і технології»*

УДК 004.8:37.018.43

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ЕКСПЕРТНИХ СИСТЕМ В ОСВІТІ

Павленко Н.А., здобувач вищої освіти *e-mail: mapmejiad.ua@gmail.com*
Науковий керівник: к.пед.н., доцент Шаров С.В.
Таврійський державний агротехнологічний університет імені Дмитра
Моторного

Актуальність та постановка проблеми. На сьогодні сфера застосування інформаційно-комунікаційних технологій (ІКТ) є доволі широкою та різноманітною. Важливим етапом розвитку ІКТ стало створення й використання експертних систем (ЕС) у таких галузях, як економіка, виробництво, державне управління. Одним з напрямків використання експертних систем є освітня діяльність, де вони можуть виконувати окремі функції викладачів, надавати експертні рішення у межах певної дисципліни.

Метою нашого дослідження є висвітлення переваг та особливостей використання експертних систем, зокрема в освітньому процесі.

Основні матеріали дослідження. Під поняттям «Експертна система» розуміється програмний засіб, що виконує функції експерта при вирішенні задач в конкретній предметній галузі. За своєю суттю вони є одним з типів інтелектуальних систем та призначені для генерування правильних та ефективних рішень у конкретній предметній галузі [7, с.74]. Експертні системи мають значну перевагу у здатності приймати рішення в ситуаціях, де відсутній чіткий попередньо сформований алгоритм. В свою чергу, експертні рішення формуються в умовах невизначеності, неповноти чи неоднозначності вихідної інформації, на основі структурованих даних, які надаються у вигляді умовиводів (правил прийняття рішень), що зберігаються у базі знань [10, с. 10].

Характерною особливістю ЕС є здатність до генерування нових знань на основі вже відомих, вирішення практичних завдань та пояснення логіки отриманих рішень. На основі знань, отриманих від експертів, система формує відповідні рішення, які можуть містити ідентифікацію ситуації, формування причини, прогнозування або рекомендації для вибору дій у конкретній предметній галузі [5, с. 254]. При цьому вдалі рішення накопичуються в системі та використовуються для підвищення ефективності майбутніх висновків.

До основних переваг ЕС можна віднести наступні:

- створення умов для розповсюдження людського досвіду та забезпечення експертної підтримки навіть за відсутності експерта;
- створення потужного інструментарію для постановки складних запитань та пошуку ефективних рішень з можливістю подальшої інтеграції з іншими програмними засобами, системами управління базами даних тощо;
- виконання окремих функцій людини, завдяки чому відбувається заощадження часу на виконання інших завдань тощо.

Експертні системи можна поділити на декілька типів в залежності від призначення. Наприклад, наступним етапом розвитку традиційних інформаційних систем є інформаційна експертна система (ІЕС) – це сукупність методів і засобів організації, накопичення та застосування інформаційних ресурсів і знань для вирішення складних задач у конкретній галузі. Однак, як і будь-які технології, ІЕС мають певні обмеження. Їхні основні недоліки пов'язані з високими

фінансовими витратами на розробку та наповнення бази знань. Крім того, ІЕС недостатньо ефективні для розв'язання нестандартних завдань та формування нових концепцій чи правил [10, с. 10].

Штучний інтелект активно використовується в інтелектуальних інформаційних системах (ІС), які представляють собою комплекс програмних, лінгвістичних, логіко-математичних засобів тощо. Такі системи забезпечують обробку, накопичення й використання великих обсягів даних, автоматизують процес прийняття управлінських рішень на основі аналізу поточних та попередніх станів системи. Їхня ключова особливість полягає в акумуляції складних технологічних рішень, які сприяють автоматизації обробки інформації та розробці альтернативних варіантів вирішення поставлених задач [3, с. 307].

Використання технологій штучного інтелекту (ШІ) в освітній діяльності відкриває нові форми, засоби й підходи до організації освітнього процесу. Інтеграція різних підходів до моделювання знань та інтелектуалізація програмного забезпечення навчального призначення надає можливість підвищити якість навчання за рахунок автоматизації процесу навчання та формування певних експертних рішень та рекомендацій. Це досягається завдяки інтеграції в базу знань експертної системи додаткової довідкової інформації, яка допомагає формувати відповіді на запитання здобувачів освіти [8, с. 29].

Одним із перспективних напрямів використання штучного інтелекту в дистанційному навчанні є створення експертно-навчальних систем (ЕНС). Під цим поняттям розуміються програмні засоби, які об'єднують ключові компоненти експертних систем і орієнтовані на навчальні завдання. ЕНС базуються на використанні комплексу методів (інтелектуальних, педагогічних, дидактичних та ін.), що дозволяє моделювати поведінку вчителя або викладача [3, с. 307].

У системі дистанційної освіти впровадження таких систем відкриває нові можливості для адаптивного навчання, коли під час формування блоку навчальних матеріалів для вивчення враховуються індивідуальні здібності здобувачів освіти. Завдяки використанню адаптивних алгоритмів і програм, такі системи стимулюють творче мислення здобувачів освіти і посилюють значущість їхньої самостійної роботи. У роботі [6, с. 268] зазначається, що такі системи структурно можуть містити основний розділ (зазвичай він містить навчально-методичні матеріали) та розділ інтелектуального керування, що забезпечує адаптивне управління освітнім процесом. Зміст навчально-методичних матеріалів включає не лише знання експертів у певній предметній галузі, але й методику їх подання, забезпечуючи інтеграцію навчальних і методичних знань у єдине інформаційне середовище [1, с. 144].

Однак під час проектування та створення ЕНС доцільно спиратися на педагогічні принципи, ефективність яких доведено у розробці інформаційних технологій навчання. У роботі [9, с. 39] вказується на таких принципах як: принцип центрування на особистості, принцип пріоритетності педагогічного підходу при проектуванні освітнього процесу з використанням ЕНС, принцип педагогічної доцільності застосування нових інформаційних технологій навчання, принцип свободи вибору пізнавальної траєкторії та унікальності індивідуальної пізнавальної діяльності здобувачів освіти, принцип різноманітності та інші.

Слід додати, що одним з напрямків використання ІКТ в освітньому процесі є оцінювання навчальних досягнень здобувачів вищої освіти. Завдяки використанню ІКТ оцінювання рівня навчальних відбувається з достатньо високою деталізацією та в короткий термін за рахунок автоматичної обробки інформації [4, с. 86]. Крім того, контроль знань та сформованих компетентностей можна здійснювати під час поточних занять, контрольних заходів тощо. Вибір

методів контролю залежить від конкретних завдань, що ставляться викладачем. Важливо враховувати, що ці методи можуть використовуватися у різних формах контролю (тестування, відкриті відповіді та ін.). Кожен з методів свої переваги, недоліки та особливості використання, що дозволяє адаптувати їх до специфіки навчальних цілей окремої дисципліни [8, с. 61]. Крім того, важливого значення набуває прогнозування результатів контрольних заходів, що може впливати на змістове наповнення контрольних заходів, вибір форм здійснення контролю тощо. У подальших дослідженнях ми плануємо розглянути можливість розробки експертної системи для здійснення прогнозування результатів контрольних заходів.

Висновки. Отже, експертні системи є одним з видів інтелектуальних систем, що можуть використовуватися в освітньому процесі. Їх переваги в основному полягають в накопиченні знань експертів та формуванні на цій основі ефективних рішень. Крім того, експертні системи здатні адаптуватися до індивідуальних особливостей користувачів та надавати релевантні блоки навчального матеріалу для вивчення.

Список використаних джерел:

1. Затхей, В.А. Використання експертних систем в дистанційному навчанні. *Системи обробки інформації*. 2015. № 4. С. 144–146.
2. Лубко Д.В. Використання ІКТ для контролю рівня навчальних досягнень учнів і студентів. *Інформаційно-комунікаційні технології навчання : зб. мат. Всеукраїнської науково-практичної Інтернет-конференції (16-17 березня 2016 р., м. Умань)*. С. 60–65.
3. Лубко Д.В., Шаров С.В. Напрямки використання інтелектуальних систем в освітньому процесі. *Українські студії в європейському контексті: зб. наук. пр.* 2021. №3. С. 305–310.
4. Олійник Т.О. Використання інформаційно-комунікаційних технологій для оцінювання рівня навчальних досягнень студентів ВНЗ. *Інформаційні технології і засоби навчання*. 2014. № 42(4). С. 85–93.
5. Ротань О.С. Особливості побудови та організації експертних систем. *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: матеріали Всеукраїнської науково-практичної Інтернет-конференції (11-21 березня 2021 р., м. Черкаси)*. 2021. С. 253–255.
6. Рукіна О.М. Використання експертної системи в системі дистанційного навчання. *Ukrainian Journal of Educational Studies and Information Technology*. 2017. Т. 5. № 1. С. 267–270.
7. Сафонов К.С., Сердюк І.М. Передумови використання експертних систем в освітньому процесі. *Стан освітнього процесу в умовах викликів сьогодення: матеріали Міжнародної науково-практичної конференції (12 лютого 2021 р., м. Дніпро)*. 2021. С. 74–75.
8. Шаров С.В., Еміратлі А.Р. Використання експертних систем для підтримки навчального процесу. *Science, research, development, technics and technology: monografia pokonferencyjna* (London, 27.02. 2018). 2018. №2. С. 28–31.
9. Шевчук О.Б. Педагогічні принципи проектування та розробки експертних систем навчання. *Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка*. Серія: Педагогіка. 2016. №1. С. 38–44.
10. Arkhipova T.L. Інтелектуальні інформаційні системи в економіці та освіті. *Journal of Information Technologies in Education (ITE)*. 2008. № 1. pp. 9-12.

ДОДАТОК В

Сертифікат

