

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Кулиняк Роман Володимирович

**Програмна система автоматизованої перевірки
коректності написання програмного коду на мові C++
/ Software system of automated verification of writing
code in C ++ language correctness**

спеціальність: 6.050102 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерні системи та мережі

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-41
Кулиняк Роман Володимирович

Науковий керівник:
Батько Ю. М.

Випускну кваліфікаційну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Бакалаврська робота містить 69 сторінок пояснюючої записки, 31 рисунок, 12 таблиць, 3 додатки. Обсяг графічного матеріалу складає 2 аркуші формату А3.

Метою бакалаврської роботи є розробка гнучкої програмної системи для автоматизації процесу проведення тестових іспитів, серед студентів, для дисципліни «мова програмування C++».

Дана бакалаврська робота описує підхід до розробки програмної системи опрацювання екзаменаційних тестових робіт, з перевіркою якості наданого студентами програмного коду. Відповідно до предметної області, надається можливість інтеграції тестувальних модулів перевірки якості коду.

Розроблений та впроваджений модуль для трьох етапної перевірки якості вихідного коду мови C++, забезпечує перевірку виконання компіляції, відповідності отриманих результатів з визначним набором вхідних параметрів до очікуваних та наявність критичних помилок в силу виток інформації, використання неініціалізованих змінних, виходу за межі допустимих границь масиву тощо.

Збереження інформації про перебіг проведення тестового іспиту виконується за використанням серверного логувальника. Збереження даних конкретного тестового іспиту виконується за використання єдиної бази даних.

Ключові слова: СОКЕТ З'ЄДНАННЯ, АРІ, CONCURRENCY, СЕРВЕРНА ЧАСТИНА, КЛІЄНТСЬКА ЧАСТИНА, КОМПІЛЯТОР, СТАТИЧНИЙ АНАЛІЗАТОР.

RESUME

Bachelor work contains 69 pages of explanatory note, 31 pictures, 12 tables, 3 applications. The volume of graphic material is 2 sheets of A3 format.

The purpose of the bachelor's work is to develop a flexible software system for automating the process of conducting test exams, among students, for the discipline "programming language C ++".

This bachelor's paper describes the approach to developing a software system for examining test work, with the quality assurance provided by students of the code. According to the subject area, it is possible to integrate test quality modules of the code.

The developed and implemented module for three-step verification of the quality of the source code of C ++, provides verification of the implementation of compilation, the correspondence of the results with a significant set of input parameters to the expected and the existence of critical errors due to the turn of information, the use of non-initialized variables, beyond the permissible limits of the array etc.

Saving the information about the course of the test exam is carried out using the server logger. Saving the data of a specific test exam is performed using a single database.

Keywords: SOCKET CONNECTION, API, CONCURRENCY, SERVER PART, CLIENT PART, COMPILER, STATIC ANALYZER.

ЗМІСТ

Вступ.....	10
1 Моніторинг та оцінювання знань на основі автоматизованих систем перевірки вихідного коду.....	9
1.1 Оцінювання знань студентів.....	9
1.2 Критерії оцінювання знань студентів.....	13
1.3 Програмні системи оцінювання знань	17
1.4 Висновки та постановка задачі.....	20
2 Методи та алгоритми автоматизованого оцінювання знань студентів	22
2.1 Проектування автоматизованої системи оцінювання знань студентів ...	22
2.2 Алгоритм автоматизованого формування тестових завдань для оцінювання знань студентів	25
2.3 Структура бази даних системи автоматизованого оцінювання знань студентів	27
3 Програмна реалізація автоматизованої системи оцінювання знань студентів	32
3.1 Опис структури ПЗ, графічного інтерфейсу та використаних бібліотек	32
3.2 Опис розроблених програмних класів та методів	36
3.3 Тестування та порівняння з аналогами	40
4 Техніко-економічне обґрунтування розробки програмного засобу	46
4.1 Розрахунок витрат на розробку програмного забезпечення	46
4.2 Визначення експлуатаційних витрат.. Ошибка! Закладка не определена.	
4.3 Розрахунок ціни споживання програмного продукту	54
4.4 Визначення показників економічної ефективності.....	56
Висновки.....	58

					БР.КСМ 07117/15.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Кулиняк Р. В.			ПРОГРАМНА СИСТЕМА АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ КОРЕКТНОСТІ НАПИСАННЯ ПРОГРАМНОГО КОДУ НА МОВІ C++	Літ.	Арк.	Аркушів
Перевірив		Батько Ю. М.					8	69
Конс.		Паздрій І. Р.				ТНЕУ. ФКІТ. КСМ-42/1		
Н. Контр.		Гураль І. В.						
Затвер.		Березький О. М.						

Список використаних джерел.....	60
Додаток А Лістинг програмного класу «Server.java».....	63
Додаток Б Світлокопія публікації	66
Додаток В Довідка про використання.....	69

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

Контроль рівня якості знань студентів, в межах навчальної дисципліни, являється однією із складових частин навчального процесу. На основі аналізу проведених тестувань визначається загальний показник успішності та сприйняття студентами пройденого матеріалу [1].

Здійснюючи проектування навчальної дисципліни, основним завданням для кожного викладача є вибір методів та способів контролю навчальних досягнень, визначення критеріїв оцінювання якості засвоєння матеріалу, формування шляхів та способів для покращення процесу вивчення предметної області [2].

Використання системи тестів, як способу перевірки знань, сприяє покращенню організації та збільшенню якості навчального процесу. Тестові завдання повинні розроблятися у відповідності з вимогами представленими до навчальної дисципліни. База тестових завдання повинна охоплювати значну кількість пройденого та засвоєного матеріалу. Загальні результати успішності групи визначають якість вибору методики вивчення навчального матеріалу викладачем.

До переваг автоматизованих способів тестування над класичними слід віднести:

- універсальність при використанні для різноманітних типів організації навчального процесу предметних дисциплін;
- захищеність системи та забезпечення конфіденційності, цілісності та доступності інформації процесу проведення та результатів тестувань;
- масовість в організації проведення тестувань для великої групи людей в межах локальних та глобальних мереж;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

– об'єктивність з визначенням загального положення та умов, неупередженість в процесі тестування та перевірки результатів, виключення об'єктивізму викладача [3].

1 МОНІТОРИНГ ТА ОЦІНЮВАННЯ ЗНАНЬ НА ОСНОВІ АВТОМАТИЗОВАНИХ СИСТЕМ ПЕРЕВІРКИ ВИХІДНОГО КОДУ

1.1 Оцінювання знань студентів

Моніторинг та контроль навчального процесу визначає оцінку ефективності освоєння матеріалу, формування умінь, розвитку та покращення навичок в конкретній предметній області [4]. Для керування процесом вивчення дисципліни на різних етапах, викладач повинен мати достовірні відомості про те, як студенти сприймають і засвоюють пройдений матеріал. Це сприятиме своєчасному коригуванню та внесенню важливих змін до навчального плану.

Тестування, як методика оцінювання та контролю якості рівня підготовки молодих спеціалістів, отримала широке визнання та поширення у сфері діяльності освітніх систем. Окрім традиційних форм тестування, широкого застосування також отримало комп'ютерне (автоматизоване) тестування, що відповідає загальній тенденції в модернізації та комп'ютеризації сучасної системи освіти [5, 6].

Автоматизоване тестування представляє собою більш справедливий спосіб визначення рівня знань. Воно забезпечує студентам рівні умови складення іспиту (процес контролю та оцінювання), чим практично виключає суб'єктивізм викладача.

Беручи до уваги актуальність технології автоматизованого тестування та її унікальність, в порівнянні з багатьма іншими методиками оцінювання знань, можна визначити перелік основних завдань тестування:

– розвиток інтересу у студентів до дисциплін, за якими передбачається тестування;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- розвиток навичок самостійного опрацювання матеріалу, при підготовці до тестувань;
- виховання свідомої та особистої відповідальності студентів за результати своєї підготовки;
- визначення якості знань, розвитку та покращення умінь і навичок серед студентів відповідно до передбачених дисциплін;
- допомога студентам у визначенні рівня досягнутих результатів навчання і планування подальшої підготовки;
- допомога викладачам в оцінці індивідуальних результатів навчання і розвитку студентів, розширення спектру контрольних-вимірювальних матеріалів, коригування методики навчання [7].

Станом на сьогоднішній день, розроблено безліч програмних засобів призначених для проведення тестування. Доступні засоби з наперед визначеним переліком тестових завдань або ж програмні-оболонки для самостійного внесення. Кожен з типів містить свої переваги та недоліки. Однак, оцінюючи загальну картину всієї різноманітності готових рішень, важко виділити та назвати один універсальний засіб, який однаково добре справлявся б з списком поставлених вимог [8].

Системи перевірки знань надають чотири загальні форми для постановки тестових задач.

а) Завдання закритої форми – необхідно вибрати один або декілька варіантів відповіді із запропонованих. При умові хорошого складення завдань закритої форми, всі запропоновані варіанти повинні бути однаково привабливими. Приклад завдання закритої форми зображено на рисунку 1.1.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Якими домішками легують кварцове скло для збільшення його показника заломлення?

1. Бор (В)

2. Фтор (F)

3. Фосфор (P)

4. Германій (Ge)

Затвердити

Рисунок 1.1 – Приклад тестового завдань закритої форми

б) Завдання відкритої форми – необхідно заповнити поле пропуску однозначною короткою відповіддю або розгорнутим висловленням думки. Приклад завдання відкритої форми з коротким варіантом відповіді зображено на рисунку 1.2.

1) Який рН водного розчину солі $Mn(NO_3)_2$?

Відповідь:

Затвердити

Рисунок 1.2 – Приклад тестового завдань відкритої форми з короткою відповіддю

Приклад завдання відкритої форми з розгорнутим варіантом відповіді зображено на рисунку 1.3.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1. Психологічні прийоми, що дають змогу успішно побудувати свою відповідь на іспиті і подолати хвилювання:

Відповідь:

Затвердити

Рисунок 1.3 – Приклад тестового завдань відкритої форми з розгорнутою відповіддю

в) Завдання на відповідність – необхідно встановити відповідності між наданими варіантами з правого та лівого стовпчика. Часто також включають завідомо неправильні “зайві” варіанти відповіді. Приклад завдання на відповідність зображено на рисунку 1.4.

1. Між магнітними величинами та одиницями вимірювання:

Магнітна індукція	Обрати... Обрати Вебер Ньютон А*м ² Ват Тесла Генрі
Магнітний потік	
Магнітний момент	

Рисунок 1.4 – Приклад тестового завдань на відповідність

г) Завдання на визначення правильної послідовності – необхідно розмістити елементи в правильній логічній послідовності відповідно до подій, термінів, значень абощо. Приклад завдання на визначення відповідності зображено на рисунку 1.5.

Стилі староруської літератури:				
– монументальний історизм				
– експресивно-емоційний				
– барокко				
– епічний				
ВІДПОВІДІ:				
1 _____;	2 _____;	3 _____;	4 _____;	

Рисунок 1.5 – Приклад тестового завдань на визначення правильної послідовності

Вище описані форми використовуються для створення тестових завдань різноманітного рівня складності. В більшості випадків, всі вони, за виключенням завдань відкритої форми з розгорнутою відповіддю, проходять перевірку виконуваною системою. Перевірку завдань відкритої форми з розгорнутою відповіддю (коли важливий людський фактор сприйняття) бере на себе викладач. Або ж у випадку застосування даної форми, для внесення вихідного коду чи інших допустимих завдань, система може коректно опрацьовувати його власноруч, без залучення викладача [9, 10].

1.2 Критерії оцінювання знань студентів

Основна мета тестування полягає у вирішенні студентом поставлених завдань, для аналізу та визначення його рівня знань в конкретній темі предметної області. Список завдань формується викладачем. Мета тестування досягається за умов визначення критеріїв оцінювання наданих студентом відповідей. В результаті чого отримується загальна оцінка, яка відповідає визначеній системі оцінювання і описує знання студента як: хороші, добрі, задовільні, незадовільні, погані абощо.

Оцінювання результату проходження тестового завдання – означає встановлення рівня виконання поставлених завдань, відповідно до очікуваних

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

результатів. Оцінюється як системність та узагальненість засвоєних знань, так і рівень розвитку спеціальних предметних навичок та вмінь, універсальних компетенцій, включаючи індивідуальні параметри та особистісні компоненти, в сукупності визначаючи і характеризуючи навчальні досягнення студента [11].

Система оцінювання знань варіюється у відповідності до країни та типу навчальних закладів.

Школи та коледжі Америки використовують п'ятибальну шкалу оцінювання [12]. ВНЗ користуються 100 бальною, наряду з буквеною, де оцінки нижче 60-ти процентів (нижче С-) являються непрохідними. Представлення американської системи оцінювання знань наведено в таблиці 1.1.

Таблиця 1.1 – Американська система оцінювання знань

Оцінка (буквенна система)	Бали GPA	Процентний вміст балів	Оцінка (традиційна система)
Університети			Школи та коледжі
A	4,00	95-100	5
A-	3,67	90-94	

Продовження таблиці 1.1

B+	3,33	85-89	4
B	3,00	80-84	
B-	2,67	75-79	
C+	2,33	70-74	3
C	2,00	65-69	
C-	1,67	60-64	
D+	1,33	55-59	
D	1,00	50-54	
F	0,00	0-49	2

Замінивши стару радянську систему, восени 2000 року, Україна визначила свою нову 12-бальну шкалу оцінювання для учнів шкіл. Коледжі, технічні

училища та вищі навчальні заклади використовують 5-бальну та 100-бальну систему [13]. Представлення української шкільної системи оцінювання знань наведено в таблиці 1.2.

Таблиця 1.2 – Українська шкільна система оцінювання знань

Нова школа	Стара школа
12	5+
11	5
10	5-
9	4+
8	4
7	4-
6	3+
5	3
4	3-
3	2+
2	2
1	1

Представлення української системи оцінювання знань ВНЗ наведено в таблиці 1.3.

Таблиця 1.3 – Українська система оцінювання знань ВНЗ

Залік	100-бальна шкала	Оцінка ECTS	
Зараховано	90-100	A	Відмінно
	82-89	B	дуже добре
	74-81	C	Добре
	64-73	D	Задовільно
	60-63	E	Достатньо
не зараховано	35-59	FX	Незадовільно
	1-34	F	Неприйнятно

Оцінка «А». Відмінні знання матеріалу теми. Точне та грамотне використання предметної термінології. Проявлення знань додаткової літератури. Лаконічне та послідовне викладення думки з незначною кількістю дрібних помилок. Високий рівень розуміння та хороші практичні навички.

Оцінка «В, С». Хороше засвоєння пройденого матеріалу. Чітке розуміння основ. Допускаються певні неточні та помилки середнього та нижче середнього рівня важкості. Погано структуроване викладення основних ідей теорії. Достатнє розуміння для продовження та подальшого вивчення наступних тем.

Оцінка «D, E». Середнє розуміння пройденого матеріалу. Наявність серйозної та більш дрібніших помилок. Допускання неточностей у формулюванні визначень, правил, термінології. Недостатньо повне розуміння основних ідей теорії. Відповідь задовольняє мінімальні критерії оцінювання.

Оцінка «FX». Матеріал засвоєний погано. Допущена значна кількість грубих помилок. Часткове розуміння фундаментальних (базових) аспектів теми. Залік в такому випадку не зараховується. Студент відправляється на повторну перездачу.

Оцінка «F». Матеріал не засвоєний, студент відмовляється відповідати на тему чи виявляє незнання її основних положень. Залік в такому випадку не зараховується. До перездачі студент не допускається. Необхідне повторне проходження дисципліни [14, 15].

Бально-рейтингова система активізує роботу студентів в межах навчального курсу, змушує їх систематично і регулярно готуватися до занять, підвищує здорову конкуренцію серед групи і їх відповідальність за свою навчальну діяльність. Сприяє розвитку студентами навичок самостійної роботи, раціональному плануванню навчального процесу викладачами та організації навчальної системи університету в цілому.

Застосування тестового контролю та систем рейтингового оцінювання знань є найбільш ефективними сучасними методами контролю успішності студентів [16].

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

1.3 Програмні системи оцінювання знань

Програмні системи оцінювання знань класифікуються за п'ятьма ознаками:

1. За цілями проведення – інформаційні, діагностичні, навчальні, мотиваційні, атестаційні.

2. За процедурою створення – стандартизовані або нестандартизовані. Стандартизовані тести повинні містити специфікацію та визначені характеристики, попередньо затверджені та рекомендовані до виконання. Стандартизований тип тестування відноситься до професіонального тестування.

3. За наявністю зворотного зв'язку – адаптивні або традиційні системи тестування. Комп'ютерна система, яка реалізує адаптивний варіант тестування, задає питання з урахуванням відповідей тестованого на раніше задані системою тестування в цьому ж тесті питання.

4. За особливостями технічної реалізації – локальні або інтерактивні з використанням web-інтерфейсу. До першої групи будуть віднесені програми, що працюють на комп'ютері локально. Друга група представляє клієнт-серверні рішення з використанням web-інтерфейсу, що дозволяє домогтися незалежності від програмної і апаратної платформ, організувати взаємодію будь-якої кількості віддалених користувачів з системою за допомогою мережі Internet.

5. За вартістю розробки або придбання – університетські та комерційні. Серед використовуваних систем можуть зустрічатися ексклюзивні (написані програмістами самого навчального закладу) або комерційні з ліцензією на продукт.

Беручи до уваги різноманітність потреб, не існує програмного рішення, яке б задовольняло і універсально застосовувалося для усіх видів тестувань. Специфіка окремих реалізацій включає одну властивість, виключаючи іншу [17]. Тому більшість сконструйованих систем суттєво відрізняються один від одного.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Для порівняння можливостей різних систем були відібрані шість програмних реалізацій систем тестування, котрі найчастіше використовуються для організації тестування знань. Серед відібраних програм: MyTestXPro, x-TLS, INDIGO, Moodle, OpenTest і Let's test. Результати порівняння програмних реалізацій тестування знань наведено в таблиці 1.4.

Аналізуючи порівняння слід зазначати, що серед найбільш використовуваних програмних реалізацій, абсолютна більшість розповсюджуються за умов платної ліцензії та несуть витрати для покупки університетом [18]. Окрім цього, вони не передбачають доопрацювання нових модулів та інтегрування власних блоків за потребами викладача та навчальної дисципліни. Продукт з безкоштовною ліцензією та можливістю доопрацювання модулів містить ряд недоліків для впровадження [17].

Серед іноземних аналогів, на базі open-source платформи GitHub, ведеться розробка нового програмного засобу, який має гнучку структуру. Наявний функціонал буде забезпечувати систему доступну до налаштувань, в основі якої використовується тестування програмного коду [20]. Для тестування вихідного програмного коду інтегруються бібліотеки з наявним статичним та динамічним тестуванням.

Таблиця 1.4 – Порівняння програмних реалізацій тестування знань

Можливості	Програмна реалізація тестування знань					
	MyTestXPro	X-TLS	INDIGO	Moodle	OpenTest	Let's test
Додаткові види тестових питань, окрім основних	X	-	-	-	-	-
Можливість налаштування шкали оцінок	X	X	X	X	X	X
Можливість імпорту питань	X	X	X	X	X	X
Можливість експорту таблиць з результатами	X	-	X	X	X	X
Захист ключів до тестів і даних користувачів	X	X	X	X	X	X
Безкоштовне розповсюдження ліцензійної версії	-	X	-	-	-	-

Можливість доопрацювання модулів програми, інтегрування власних блоків	-	X	-	-	-	-
Можливість зміни дизайну інтерфейсу програми, що тестується	-	-	X	X	X	X
Налаштування розкладу часу проведення тестування	-	-	X	X	X	X
Встановлення додаткового анкетування під час тестування	-	-	-	X	X	X

Беручи до уваги специфіку проведення тестувань дисципліни “мова програмування С”, жоден з варіантів не виконує всіх поставлених завдань:

- реалізація ефективного тестування знань;
- точна і здатна до адаптації оцінка результату тестування (статичний аналіз вихідного коду, тестування методом чорного ящика);
- можливість легкого додавання нових завдань (з текстом і коректним кодом реалізації або набором зазначених вхідних/вихідних даних для перевірки);
- можливість проведення тестувань для вибраних груп студентів;
- моніторинг активності (подій) викладачем;
- збереження даних проведення іспиту в єдиній базі;

Програмні платформи для олімпіадних змагань між програмістами, на прикладі “CodeChef” та інших представляють ядро, тобто основний компонент, в якому реалізована лише перевірка вихідного коду без необхідної системи проведення іспиту в цілому [21]. Інтерфейс платформи для олімпіадних завдань “CodeChef” зображено на рисунку 1.6.



Рисунок 1.6 – Інтерфейс платформи “CodeChef”

Комбінація базових підходів до реалізації системи проведення тестувань та ядро перевірки вихідного коду (олімпіадних платформ) формує основу платформи.

1.4 Висновки та постановка задачі

Відповідно до розглянутих способів оцінювання знань студентів, критеріїв оцінювання знань студентів та доступних програмних реалізацій, виноситься перелік задач та цілей для проектування системи. Які будуть визначати загальний підхід до розробки, для визначення архітектури та функціональних можливостей [22].

Задачі та цілі проектування системи:

- реалізувати клієнт-серверну архітектуру в межах локальної мережі;
- розробити програмний інтерфейс сервера, для обробки клієнтів типу: студент та викладач;
- розробити функціонал розподілення тестових завдань в межах студентської групи;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

- розробити функціонал трьох-етапного тестування (компіляція, black-box, статичний аналізатор) програмного коду мовою C++;
- спроектувати базу даних для збереження інформації про перебіг іспиту;
- розробити логувальник для запису активності користувачів;
- розробити алгоритм визначення оцінки складення іспиту, на основі отриманих даних про етапи тестування програмного коду;
- забезпечити ефективну систему для проведення тестових іспитів.

Виконання поставлених цілей, дозволить реалізувати гнучку програмну систему, котра надасть змогу якісно проводити тестувань знань студентів, в межах програмної дисципліни «мова програмування C++».

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

2 МЕТОДИ ТА АЛГОРИТМИ АВТОМАТИЗОВАНОГО ОЦІНЮВАННЯ ЯКОСТІ ЗНАНЬ СТУДЕНТІВ

2.1 Проектування автоматизованої системи оцінювання знань студентів

Відповідно до поставлених завдань, проектуємо архітектуру майбутньої системи оцінювання знань студентів. В основі повинна бути закладена клієнт-серверна архітектура на основі сокет з'єднання [23]. Тобто, клієнт та сервер можуть комунікувати лише в межах локальної мережі. Забезпечивши необхідний функціонал системи, користування сокетами дає змогу обмежити користувача у виході в глобальну мережу Інтернет [24].

Викладачем в аудиторії повинен бути вибраний комп'ютер для запуску сервера. Визначивши перелік груп, тестових завдань та тривалості іспиту, викладач або адміністратор запускає сервер, який готовий приймати та обробляти клієнтські запити.

В загальному вигляді базова схема клієнт-серверної архітектури зображена на рисунку 2.1.



Рисунок 2.1 – Базова схема клієнт-серверної архітектури

Реалізація клієнтської частини передбачає два типи клієнтських з'єднань: студент та викладач [25, 26]. В межах проведення тестувань, сервер надає функціонал для комфортного складення іспиту студентом та контролю проведення викладачем.

Для більш загального представлення функціонування системи сконструйована UML діаграма варіантів використання, в якій представлено відношення між акторами та прецедентами зображено на рисунку 2.2.

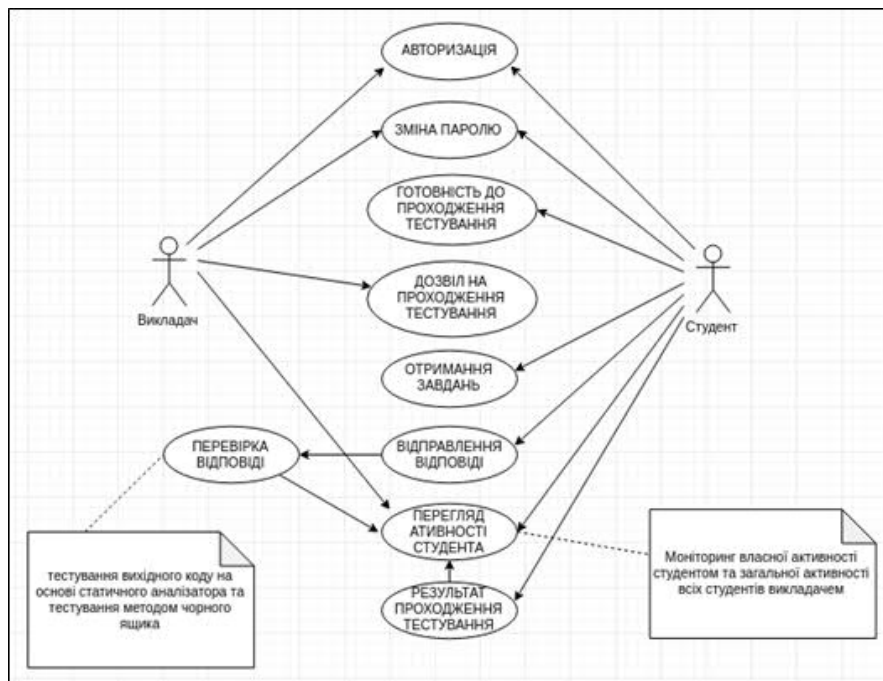


Рисунок 2.2 – Діаграма варіантів використання

Після проходження авторизації студент виконує запит на готовність до проходження тестування. Запит приймається сервером, лише в разі входження студентом до переліку груп, які підлягають тестуванню (задається при старті сервера). Проте тестування не запускається до моменту, поки викладач не затвердить список тестуємих. В разі недопущення викладач має змогу скасувати доступ до проведення тестування окремими студентами.

Отримавши тестові завдання, студентам доступний визначений проміжок часу для відправлення їх рішень. В межах складення тестового іспиту надається дозвіл на використання інтегрованого середовища розробки, текстового редактора, компілятора та документація стандартної бібліотеки мови.

Активність студента (початок/завершення тестування, оцінка виконаного тесту абощо) логуватиметься і відобразатиметься у визначеному на це місці. Кожен студент здатен слідкувати за перебігом здачі свого іспиту. Система

забезпечує відображення активності кожному окремому студенту та загальний моніторинг активності в межах всіх студентів групи для викладача. Це забезпечує конфіденційність проходження тестування для кожного студента, адже кожному доступна інформація лише про своє проходження, та відкритість на всіх етапах проведення тестування. Студенту надається доступ до логу всіх його подій. В разі спірної ситуації або виникнення проблеми чи порушення загальних правил проведення, він сигналізує про це викладачу. Викладач, в свою чергу, містить інформацію про активність всіх студентів. Реалізація даного механізму несе на меті запровадження відкритого, чесного та об'єктивного підходу до проведення тестового іспиту.

Перевірка наданого студентом рішення виконується сервером. До уваги береться, як загальний аналіз валідності вихідних даних програми при визначених вхідних, так і статичний аналізатор котрий виконує перевірку таких важливих аспектів: витік пам'яті, відкритий ресурс, використання неініціалізованої змінної (мова програмування C++ присвоює визначеній але неіціалізованій змінній garbage value) та інші. Ілюстрація механізму етапів перевірки наданого студентом вихідного коду зображено на рисунку 2.3.

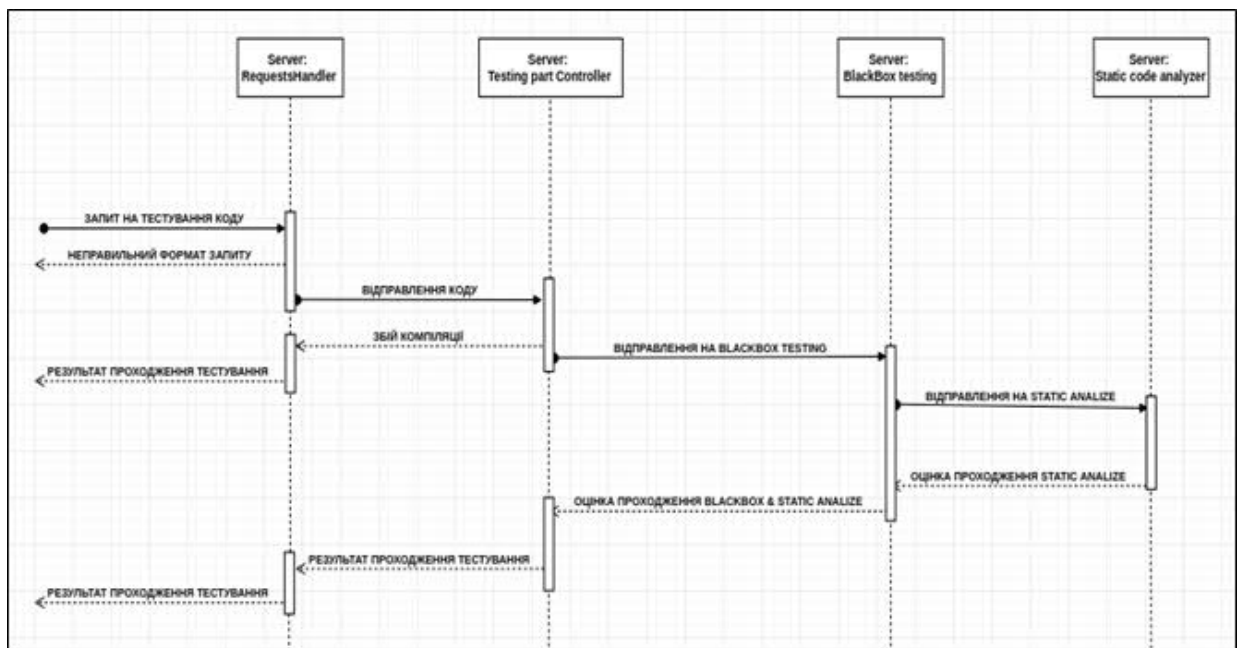


Рисунок 2.3 – Діаграма послідовностей тестування вихідного коду

Опис процесу тестування вихідного коду:

- сформований запит на тестування коду подається на обробник, який перевіряє його валідність та відправляє контролеру тестування;
- першим етапом тестування код проходить компіляцію на перевірку обов'язкової умови відсутності помилок запуску;
- другим етапом виконується перевірка методом чорного ящика, коли при певних вхідних даних перевіряється коректність отриманих вихідних;
- третім етапом виконується статичний аналізатор коду.

На основі пройдених перевірок система формує оцінку якості наданого студентом коду і зберігає дану спробу. Після завершення тестування формується загальна оцінка на основі найкращих спроб.

Частота надсилання запитів перевірки тестового рішення (вихідного коду) не більше одного разу в хвилину. Інтервал часу, за який студент може відлогітисся (у разі виникнення технічних несправностей) та продовжити проходження тесту повторною авторизацією складає три хвилини.

2.2 Алгоритм автоматизованого формування тестових завдань для оцінювання знань студентів

Проведення тестового іспиту можливе за умов вказання студентських груп, пакетів завдань та тривалості даного іспиту. Пакети завдань групуються відповідно до теми і представляють собою множину програмних задач, які будуть відправлені студентам для подальшого розв'язування.

Визначення пакетів тестових завдань відбувається ще до запуску сервера. Адміністратор повинен затвердити з викладачем їх перелік та вказати в параметрах запуску. Після чого системі доступна інформація про список завдань рекомендованих до використання.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура тестового завдання містить готовий програмний код, який використовується сервером для тестування студентських відповідей, та текст завдання доступний студенту для ознайомлення і формування подальших відповідей.

Загальний опис алгоритму формування тестових завдань для студентів містить десять кроків.

Крок 1. Визначення переліку пакетів доступних для використання в межах даного іспиту.

Крок 2. Групування завдань відповідно до теми та рівня складності.

Крок 3. Формування трьох списків доступних для вибірки завдань.

Крок 4. Вибірка випадкового елемента з першого списку доступних завдань.

Крок 5. Позначення вибраного завдання маркером та вилучення із списку.

Крок 6. Вибірка випадкового елемента з другого списку доступних завдань.

Крок 7. Позначення вибраного завдання маркером та вилучення із списку.

Крок 8. Вибірка випадкового елемента з третього списку доступних завдань.

Крок 9. Позначення вибраного завдання маркером та вилучення із списку.

Крок 10. Оновлення переліку доступних завдань у разі вичерпання списків.

Отже, сервер формує списки доступних завдань і за допомогою випадкової вибірки, маркування та оновлення даного списку реалізовується алгоритм формування тестових завдань для кожного студента.

Блок-схема алгоритму вибірки випадкових елементів з масиву зображена на рисунку 2.4.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

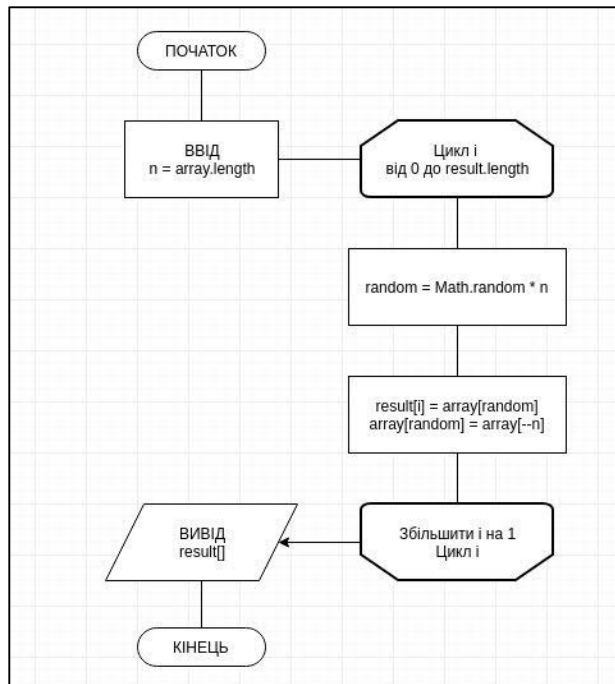


Рисунок 2.4 – Блок-схема алгоритму вибірки випадкових елементів з масиву

Таким чином описується реалізація розробки автоматизованого алгоритму формування тестових завдань для оцінювання знань студентів.

2.3 Структура бази даних системи автоматизованого оцінювання знань студентів

Для збереження інформації про хід проведення іспиту використовується логувальник та база даних. На відміну від логувальника, який зберігає інформацію про будь-яку активність / зміну, до бази даних вноситься важлива сформована інформація. Студенти та викладачі не в змозі вносити / змінювати / оновлювати / видаляти такі дані. База даних виступає в ролі сховища інформації про всіх студентів, викладачів, тестів, проведених іспитів абощо. Доступ до яких має сервер та адміністратор.

Першою важливою сутністю в базі даних виступають “користувачі” (студент та викладач). Для авторизації в системі ви повинні бути заздалегідь

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

внесені в базу даних адміністратором. В іншому випадку вам не вдасться увійти в головне меню. Сутність студента містить: логін (PK), ім'я, прізвище та віднесення до конкретної студентської групи (FK). Сутність викладача містить: логін (PK), ім'я та прізвище. Опис сутностей студент та викладач зображено на рисунку 2.5.

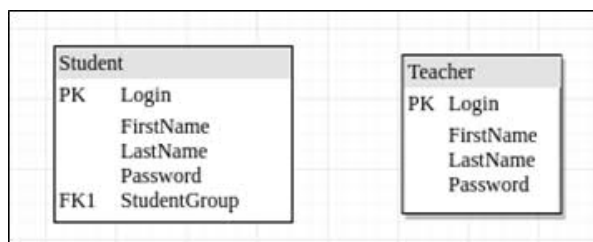


Рисунок 2.5 – Опис сутностей студент та викладач

Наступна сутність “завдання”. Завдання формуються викладачем. До них повинно входити: ідентифікатор (PK), текст умови, реалізований код для тестувань та віднесення до конкретного тестового пакету (FK). Текст умови представляється студенту, в той час коли код використовується на сервері для black-box тестувань. Маючи в наявності правильно описаний код, можна зрівнювати наданий студентом варіант вирішення з оригінальним. Вибравши вхідні дані проводиться порівнювання між вихідними даними оригінального коду та вихідними даними студентського коду. Опис сутності завдання зображено на рисунку 2.6.

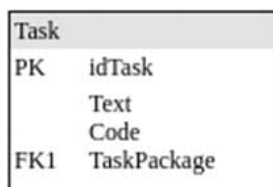


Рисунок 2.6 – Опис сутності завдання

Як вказувалося вище, студенти та завдання групуються в множини у відповідності до віднесення до студентської групи та пакету тестових завдань відповідно. Сутності студентської групи та пакету тестових завдань несуть на

меті з'єднати окремі записи в одну логічну структуру. Опис сутностей та зв'язків студентської групи та тестового пакету зображено на рисунку 2.7.

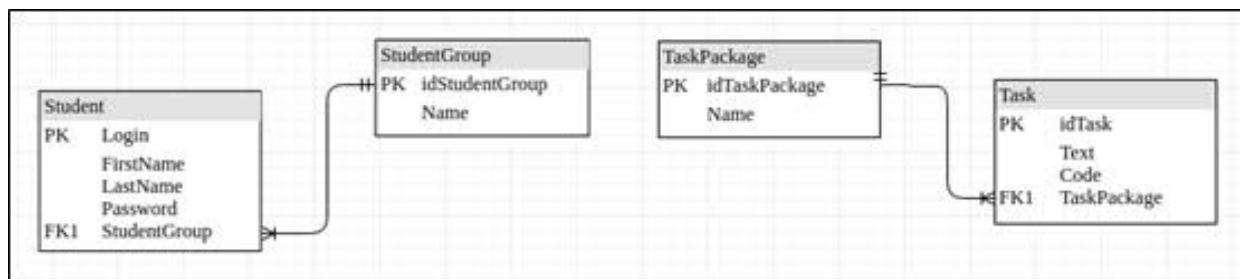


Рисунок 2.7 – Представлення сутності студентської групи та тестового пакету

Внесенням нових даних, до описаних вище сутностей, займається адміністратор. Він повністю відповідальний за представлення цих даних, їх валідність та цілісність.

Системою контролюється важлива сутність відповідальна за реєстрацію нового іспиту. Сутність іспит містить: ідентифікатор (PK), викладача (FK), тривалість іспиту. Здійснюючи запуск тестування система зберігає інформацію про нього в дану таблицю. Опис сутності іспиту зображено на рисунку 2.8.

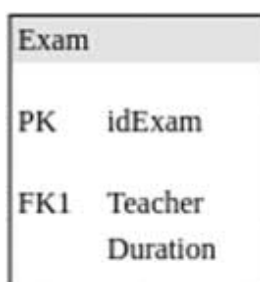


Рисунок 2.8 – Опис сутності іспиту

Окрім визначених в таблиці полів іспит містить відомості про групи (одна / декілька) та пакети завдань (один / декілька) реалізовані за допомогою зв'язку багато до багатьох. Схема відношень між сутностями пов'язаними з іспитом зображена на рисунку 2.9.

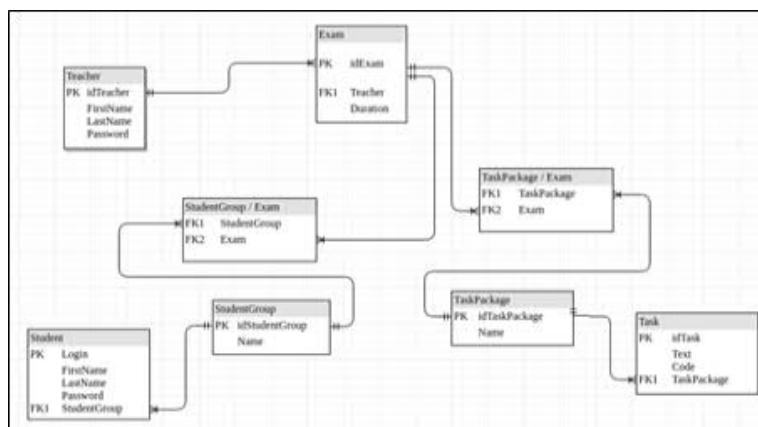


Рисунок 2.9 – Схема відношень між сутностями пов’язаними з іспитом

Визначивши проведення іспиту, ми формуємо таблиці для збереження даних про виконанні студентом завдання, інформацію по кожній спробі та успішне завершення складання іспиту.

Отже, наступна сутність відповідає за процес складання студентом іспиту. В ній містяться поля: ідентифікатор (PK), студент (FK), віднесення до конкретного іспиту (FK), перелік отриманих завдань, спроби виконання тестових завдань та доступний залишок часу. Заповнення даними сутності процесу складання іспиту дозволяє відновити студентом проходження тестування, у разі випадкового відключення. Схема відношень між сутностями пов’язаними з процесом складання зображена на рисунку 2.10.

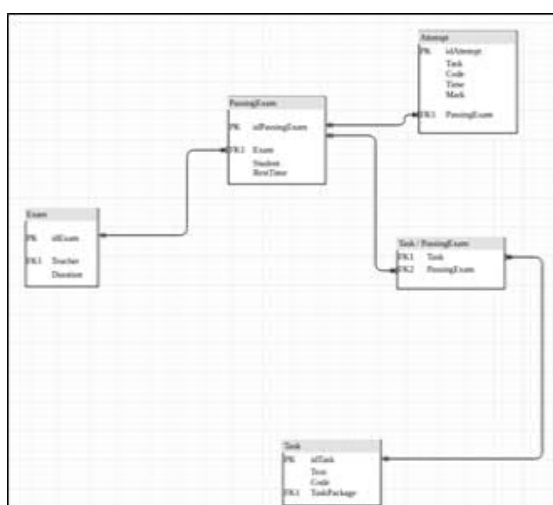


Рисунок 2.10 – Схема відношень між сутностями пов’язаних з процесом складання іспиту

Завершальна сутність складення іспиту використовується для внесення кінцевої оцінки після завершення тестування. Вона містить: ідентифікатор (PK), студента (FK), оцінку та віднесення до іспиту (FK). Опис сутності складеного іспиту зображено на рисунку 2.11

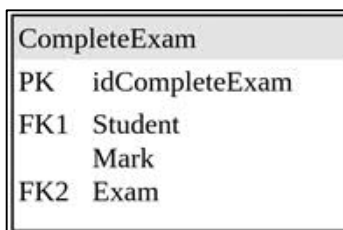


Рисунок 2.11 – Представлення сутності складеного іспиту

Загальна схема всіх сутностей бази даних і відношень між ними зображена на рисунку 2.12.

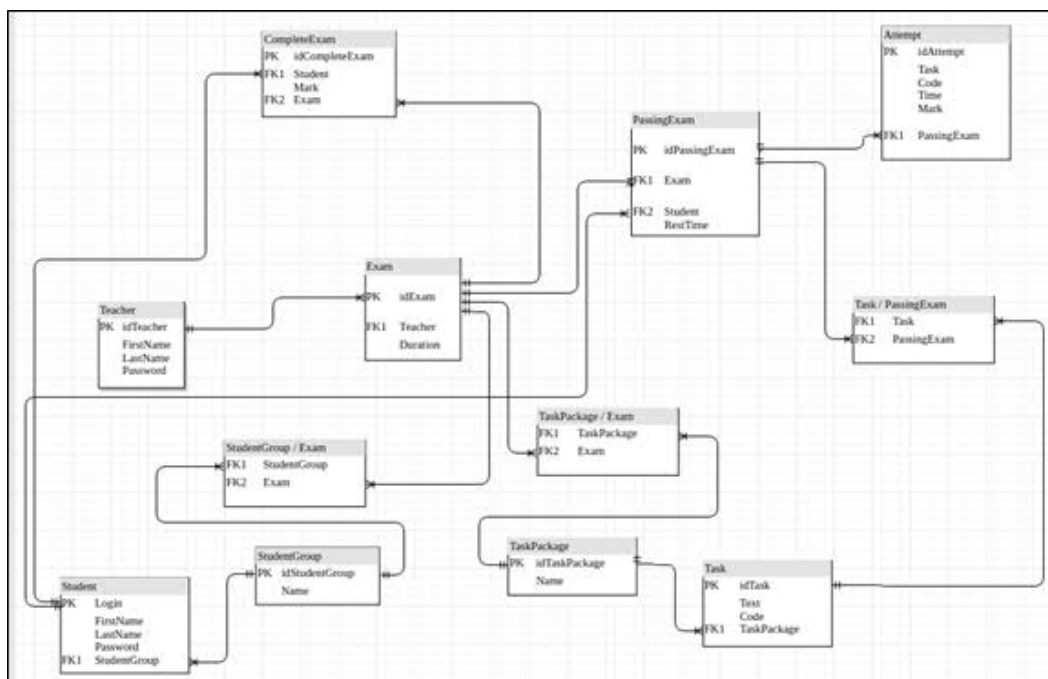


Рисунок 2.12 – Схема відношень між сутностями БД

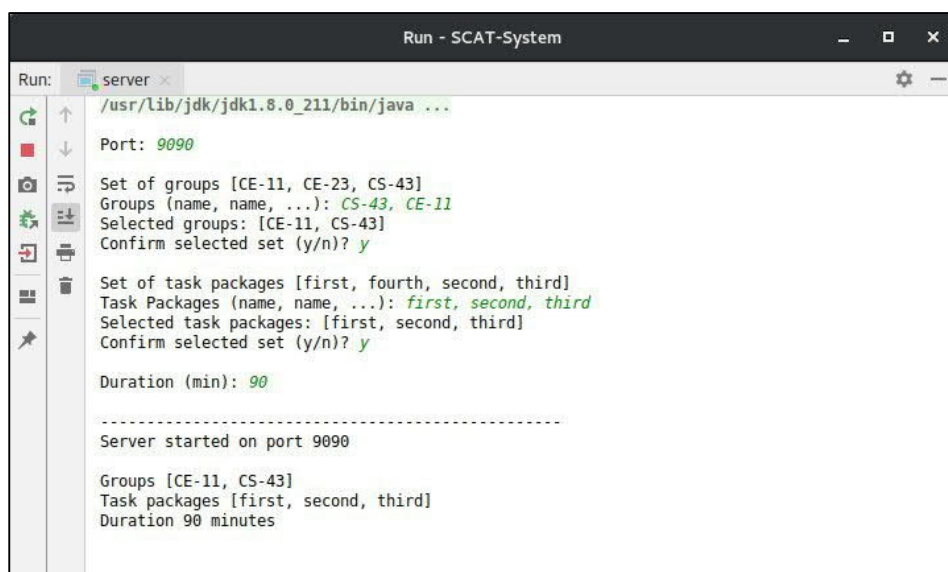
Проектування бази даних це важливий етап розробки програмного забезпечення. Визначивши правильні сутності та відношення між ними, система може організувати запити, вносити дані та змінювати їх.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

3.1 Опис структури ПЗ, графічного інтерфейсу та використаних бібліотек

Загальна форма представлення структури автоматизованої системи оцінювання знань студентів містить: серверну частину, базу даних та клієнтську частину.

Основою та ядром системи виступає серверна частина. Сервер описує загальні правила проведення іспиту. Він відповідальний за збереження, отримання та зміну даних в наявній базі даних і комунікацію з клієнтами для обробки та представлення цих даних. Клієнтська частина спілкується з сервером за наявним API, що наперед визначає чіткі правила взаємодії. Графічне представлення роботи сервера зображено на рисунку 3.1.



```
Run - SCAT-System
Run: server x
/usr/lib/jdk/jdk1.8.0_211/bin/java ...
Port: 9090
Set of groups [CE-11, CE-23, CS-43]
Groups (name, name, ...): CS-43, CE-11
Selected groups: [CE-11, CS-43]
Confirm selected set (y/n)? y
Set of task packages [first, fourth, second, third]
Task Packages (name, name, ...): first, second, third
Selected task packages: [first, second, third]
Confirm selected set (y/n)? y
Duration (min): 90
-----
Server started on port 9090
Groups [CE-11, CS-43]
Task packages [first, second, third]
Duration 90 minutes
```

Рисунок 3.1 – Представлення роботи сервера

Сервер реалізує багатопоточну модель програми. Основний потік виступає в ролі слухача та опрацьовує нові підключення. Кожне нове підключення формує зв'язок між клієнтом та сервером і виконується окремо в новому потоці. При припиненні роботи основного потоку, всі клієнто-орієнтовані потоки отримують

сигнал про необхідність коректного завершення поточного з'єднання та звільнення використовуваних ресурсів.

В межах роботи клієнт-серверної архітектури база даних відіграє ключову роль. Вона використовується для збереження всієї важливої інформації про процес проведенні іспиту, також в ній зберігаються дані про студентів, викладачів, перелік завдань і тому подібне. Для роботи з цією базою даних адміністратор працює з клієнтською частиною, яка відповідальна лише за зв'язок з базою даних і надає функціонал для внесення, читання, зміни та видалення доступних даних. Основне вікно клієнтської частини адміністратора зображено на рисунку 3.2.

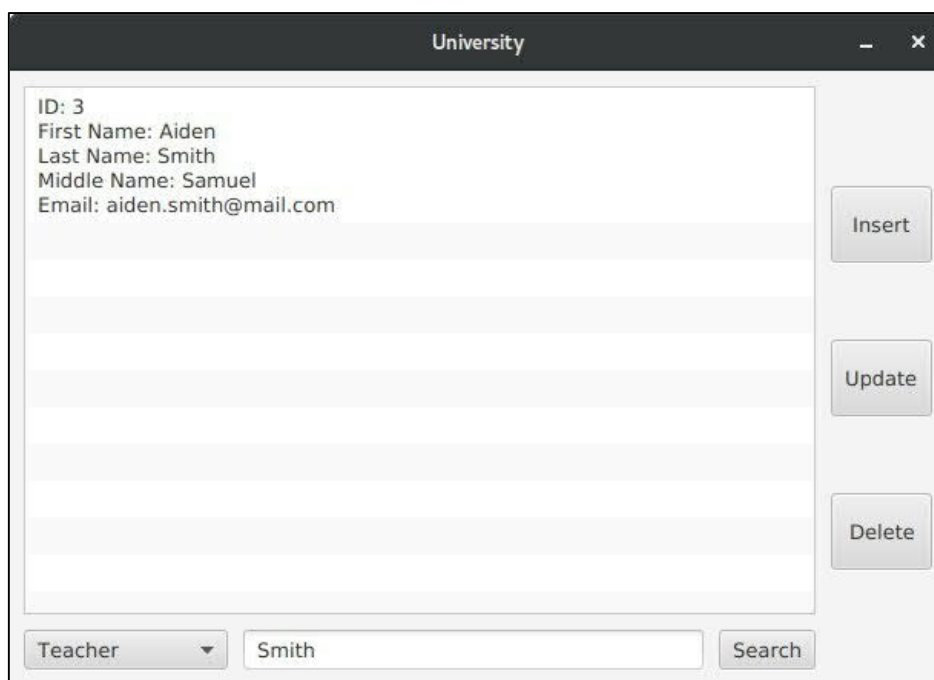


Рисунок 3.2 – Основне вікно клієнтської частини адміністратора

Внесення нових даних про студентів, викладачів, тестові завдання та формування групи студентів чи тестових завдань (пакетів) відноситься до відповідальності адміністратора. Внесення нових даних про процес проведення іспиту відноситься до відповідальності серверної частини.

Сервер опрацьовує вже сформовані дані з доступної бази даних та не містить зв'язку з клієнтською частиною адміністратора. Клієнтська частина

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

викладача та студента напряму залежить від стану роботи сервера. Вони формують підключення після запуску нового іспиту.

Роль викладача в межах іспиту базується на визначенні переліку студентів, доступних до складання іспиту та безпосереднього запуску даного іспиту. Після чого він слідкує за перебігом проведення та отримує дані нових подій (розподілення завдань студентам, опрацювання сервером відправлених студентських відповідей, загальна оцінка та інформація про завершення складання іспиту). Основне вікно клієнтської частини викладача зображено на рисунку 3.3.

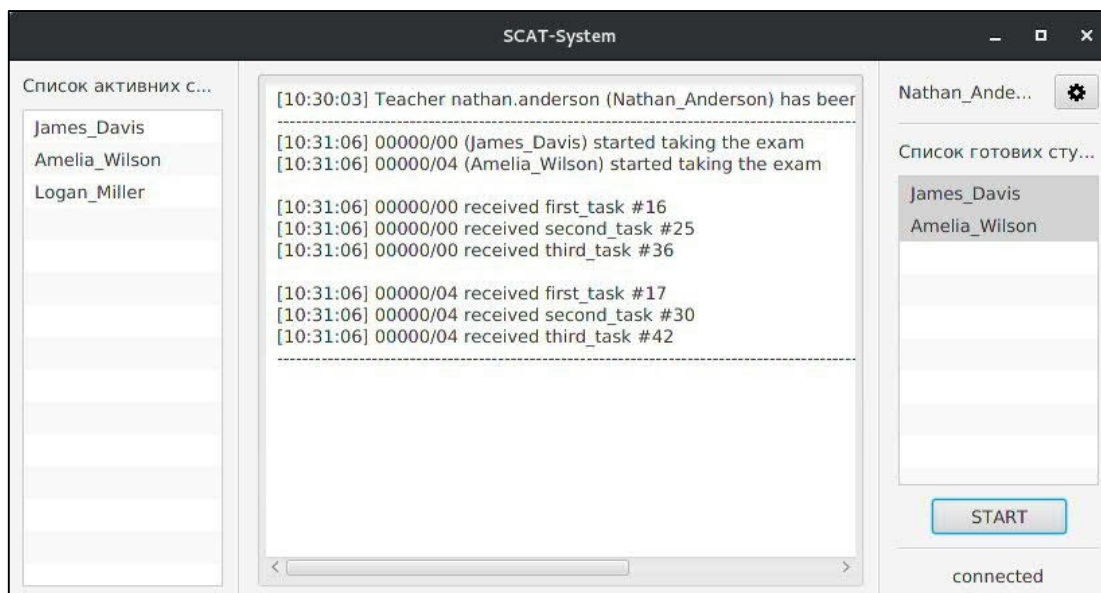


Рисунок 3.3 – Основне вікно клієнтської частини викладача

Опис графічного інтерфейсу клієнтської частини викладача:

- у верхньому правому куті міститься ім'я викладача та вікно налаштувань;
- ліва сторона вікна містить список студентів, які є активними (вдала автентифікація);
- права сторона вікна містить список студентів які очікують на отримання завдань і початок складання іспиту (викладач повинен затвердити список очікуючих студентів та натиснути кнопку початку іспиту “START”).

- після чого у вікні легування подій, яке знаходиться посередині, викладач має змогу переглядати всю доступну інформацію про процес проведення тестового іспиту;
- в правому нижньому куті міститься інформація про стан зв'язку з сервером.

Студент проходить автентифікацію на сервері та вноситься до списку активних. Після чого він подає сигнал про свою готовність до складання іспиту. Отримавши завдання та після успішного запуску процесу проведення іспиту, студент надає файлову відповідь на тестове завдання серверу, за яку формується оцінка. Завершивши процес складання іспиту, формується загальна оцінка для студента, на основі отриманих відповідей до тестових завдань. Основне вікно клієнтської частини викладача зображено на рисунку 3.4.

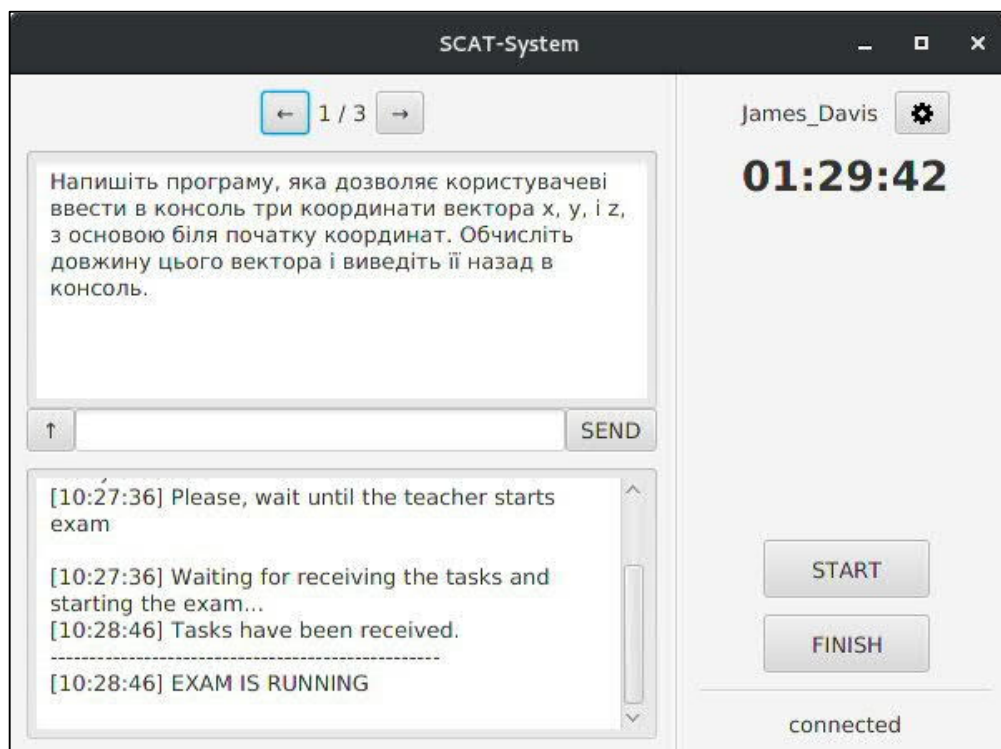


Рисунок 3.4 – Основне вікно клієнтської частини студента

Опис графічного інтерфейсу клієнтської частини викладача:

- у верхньому правому куті міститься ім'я студента та вікно налаштувань;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

- нижче розміщене інформаційне поле наявного залишку часу для складання іспиту;
- нижче доступні кнопки для старту (інформування про готовність до складання) та фінішу (передчасне завершення складання іспиту);
- ліва верхня сторона містить вікно для роботи з отриманими тестовими завданнями (доступний перегляд та відправлення файлу відповіді для визначеного завдання з дерева файлової директорії комп'ютера);
- ліва нижня сторона містить вікно логування подій процесу складання іспиту;
- в правому нижньому куті міститься інформація про стан зв'язку з сервером.

Клієнтська частина викладача та студента взаємодіють з сервером на основі визначеного API для кожного. При втраті зв'язку з сервером необхідна повторна авдентифікація, після чого доступне відновлення процесу проведення тестового іспиту.

Окрім наявного функціоналу та розроблених алгоритмів, для перевірки якості коду засобами статичного аналізатора, використовується бібліотека “crrcheck”. Основним її призначенням є знаходження помилок програми, які можуть не виявитися на етапі компіляції, проте несуть значні недопрацювання і спричиняють такі проблеми як витік пам'яті, вихід за межі масиву, використання неініціалізованих змінних, відсутність коректного завершення опрацювання використовуваних ресурсів.

3.2 Опис розроблених програмних класів та методів

Серверна частина відповідає за опрацювання клієнтських запитів, взаємодіє з базою даних та містить складну бізнес-логіку, розбиту на модулі, для опрацювання всієї необхідної інформації.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

UML діаграму відношень між складовими модулями системи представлено у БР.КСМ.07117/15.00.00.000 С1.

Модульна частина запуску сервера описує правила за якими проводиться базове налаштування сервера. Для коректного запуску необхідно визначити порт, перелік студентських груп, перелік тестових завдань та тривалість іспиту. Після чого потік виконання передається модулю опрацювання вхідних з'єднань.

Модуль опрацювання вхідних з'єднань отримує серверний сокет-слухач, який очікує нових підключень, обробляє їх і передає з'єднання в новому потоці модулю взаємодії з користувачем.

Модуль взаємодії з користувачем містить контролер, який направлений безпосередньо на обробку вхідних запитів і відправлення результату виконання цих запитів. Вдало пройшовши авдентифікацію потік взаємодії передає керування класу, відповідального за тип підключення (student/teacher).

Опрацювання подальших запитів визначено АРІ яке доступне для даного підключення. Загальне представлення діаграми класів описаних модулів зображено на рисунку 3.5.

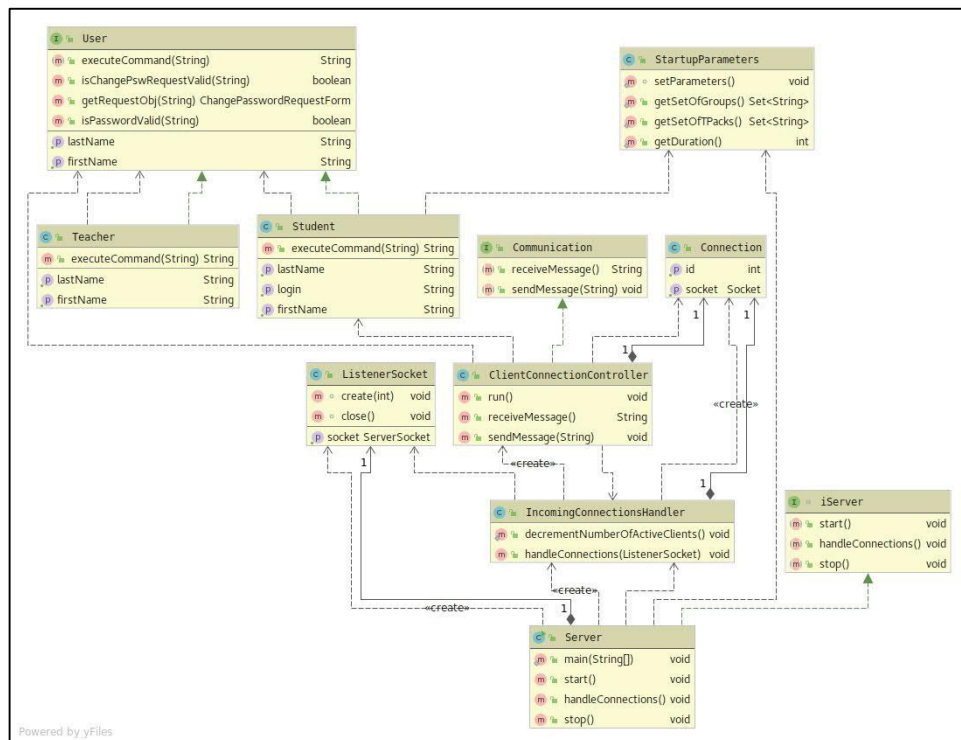


Рисунок 3.5 – Загальне представлення діаграми класів серверної частини

Повне представлення відношень між програмними класами серверної частини представлено в БР.КСМ.07117/15.00.00.000 С2.

Інтерфейс iServer визначає загальні методи для запуску, роботи та зупинки виконання сервера. Імплементация інтерфейсу iServer клас Server. Лістинг програмного класу Server представлено в додатку А.

Метод start() відповідальний за почерговий виклик методів для:

- реєстрації JDBC драйвера;
- перевірки наявного підключення до бази даних;
- визначення стартових параметрів запуску;
- створення серверного сокета-слухача;
- відображення інформації про запуск сервера на консоль.

Метод handleConnections() створює новий об'єкт IncomingConnectionHandler та передає керування методу handleConnections (socket:ListenerSocket). Даний метод відповідальний за почерговий виклик методів для:

- перевірки отриманих вхідних параметрів;
- отримання нового клієнтського підключення;
- створення об'єкту Connection з ідентифікатором та сокет об'єктом нового підключення;
- перевірка завантаженості сервера;
- виведення інформації про нове підключення і передача обробнику підключень.

Цикл описаних викликів виконується доки listenerSocket не закритий, тобто слухач доступний для використання і отримання нових підключень. У разі виникнення помилки на якомусь з етапів, через втрату підключення, генеруються та обробляються типи відповідних помилок для логування, на консоль виводиться відповідне повідомлення, звільнюються ресурси та передається виконання початку методи (для обробки наступного коректного підключення).

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

У випадку виникнення критичних помилок, які не дозволяють продовжувати процес роботи сервера, генерується `ServerFatalException` і виконання після перехоплення даного типу `checked exception` переходить до метода `stop()`.

Метод `stop()` відповідальний за почерговий виклик методів для:

- звільнення всіх ресурсів;
- посилення сигналів на припинення роботи всіх клієнто-орієнтованих потоків;
- закриття серверного сокета-слухача;
- відображення інформації про зупинку сервера на консоль.

Події запуску сервера, наявних стартових параметрів, опрацювання з'єднань, можливі помилки або що, записуються в лог файли, які архівуються на комп'ютері і дають змогу переглянути процес роботи сервера.

Модуль взаємодії з базою даних визначає всі наявні способи внесення, отримання чи оновлення даних. Загальне представлення діаграми класів модуля взаємодії з базою даних зображено на рисунку 3.6.

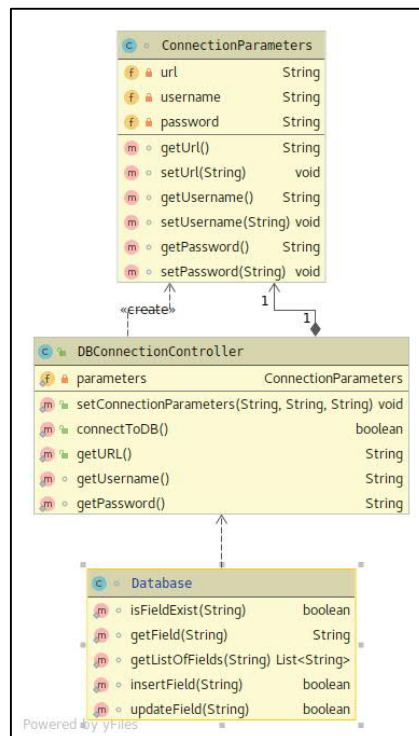


Рисунок 3.6 – Загальне представлення діаграми класів модуля БД

Клас DataBase містить інтерфейс для безпосереднього зв'язку з базою даних. Він містить методи, які виконують SQL запити для отримання значення, списку значень, перевірка наявності даних, внесення та оновлення даних.

Клас ConnectionParameters містить проперті (url, username, password) для налаштування зв'язку з JDBC драйвером.

Клас DBConnectionController представляє контролер для встановлення зв'язку між системою та JDBC драйвером, для подальшого зв'язку з БД.

3.3 Тестування та порівняння з аналогами

Клієнтська частина користувачів (викладач, студент) отримує зв'язок з сервером на основі вказаного IP адресу та порту. При запуску, параметри отримують значення за замовчуванням. У випадку невдалої спроби підключення до сервера, користувач має можливість змінити значення параметрів, за допомогою діалогового меню налаштувань параметрів з'єднання зображеного на рисунку 3.7.

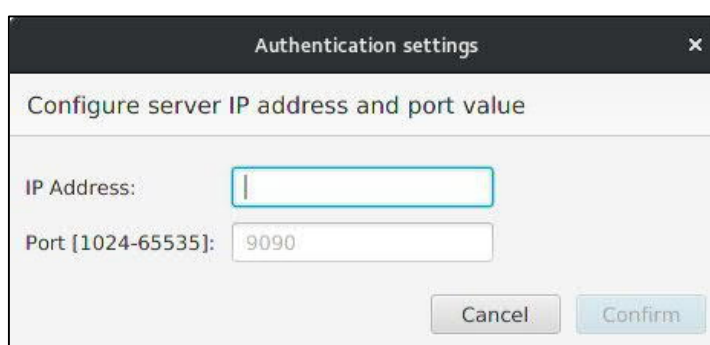


Рисунок 3.7 – Діалогове меню налаштування параметрів з'єднання

За наявності інформаційного поля та діалогового меню налаштування, користувач має змогу виявити помилки підключення та здійснити необхідні дії, для їх усунення. Таким чином надається змога для запуску декількох серверів на

різних машинах, для одночасного проведення тестувань в різних групах, в межах різних аудиторій (локальних мереж).

Після налаштування вдалого зв'язку з сервером, наступним етапом тестування клієнтської частини буде процес авторизації та авдентифікації користувача в системі. Вдалих вхід користувача в систему можливий лише за дотримання всіх визначених правил. Процес авторизації та авдентифікації користувача в системі передбачає:

- користувач може здійснити вхід в систему, лише за наявності власного логіна та пароля (логін студента відповідає ідентифікаційному номеру його залікової книжки, логін викладача відповідає його електронній пошті в домені навчального закладу);
- пароль користувача встановлюється за замовчування (зміна паролю доступна в вікні налаштувань головного меню);
- не допускається повторна реєстрація та вхід в систему користувача, який отримав підключення та активний на момент відправлення запиту.

Відображення стану опрацювання запиту на вхід в систему міститься в інформаційному полі авторизації користувача рисунок 3.8.

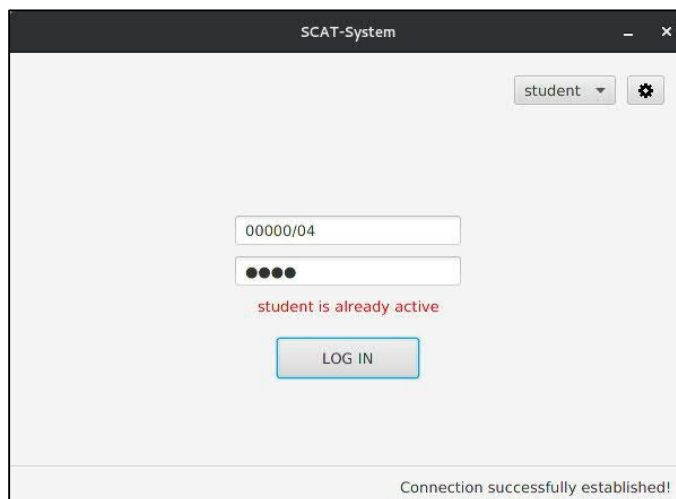


Рисунок 3.8 – Тестування спроби повторної авторизації студента

Авторизація викладача потребує додаткового дозволу з сторони серверного процесу. Таким чином, при опрацюванні коректної заявки на

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

підключення, серверна частина видає інформацію про спробу підключення викладача, з вказанням IP адресу та порту. Вхід викладача в систему можливий, лише за підтвердження даної спроби. Відображення процесу отримання додаткового дозволу для авторизації викладача зображено на рисунку 3.9.

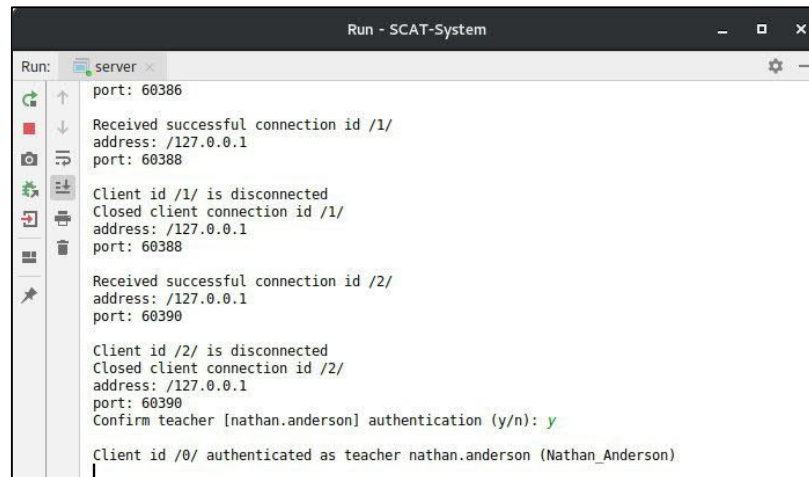


Рисунок 3.9 – Отримання дозволу на авторизацію викладача

Авторизація студента, який не входить в перелік груп, в межах даного іспиту, не обмежується сервером. Проте такий студент не може потрапити в список студентів очікуючих на початок іспиту рисунку 3.10.

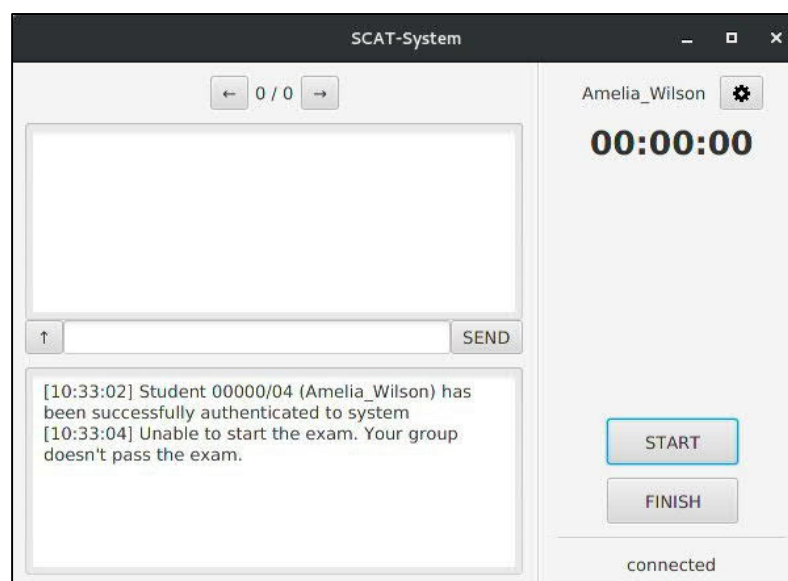


Рисунок 3.10 – Тестування спроби складання іспиту студентом група якого, не входить в перелік груп, в межах певного іспиту

Наступним важливим етапом тестування клієнтської частини студента, розглядається можливість відновлення процесу складання іспиту, у випадку аварійного завершення програми або втрати зв'язку.

Визначивши перелік студентів, допущених до складання іспиту, та запуск іспиту викладачем, створює новий запис в базі даних, який містить всю важливу інформацію про процес проведення. Отримавши тестові завдання та значення залишку часу, клієнтська частина студента, кожного визначеного інтервалу, проводить оновлення даних про доступний залишок часу. Таким чином забезпечується механізм відновлення екзамену, при втраті студентом зв'язку

рисунок 3.11.

Принцип роботи SCAT-System передбачає проведення єдиного тестового іспиту в межах нового запуску серверної частини. Перехід студента в список очікування на запуск іспиту викладачем можливе лише до старту даного іспиту, в іншому випадку студент не допускається до здачі. Проте, відновлення процесу складання іспиту доступне всім студентам, яких внесено в перелік. Відновлення передбачає отримання набору тих ж тестових завдань, та залишку часу з можливою похибкою в 10 секунд. Також, це жодним чином не впливає на отримані бали за результатом попередніх спроб, виконаного одного чи декількох завдань до втрати з'єднання.

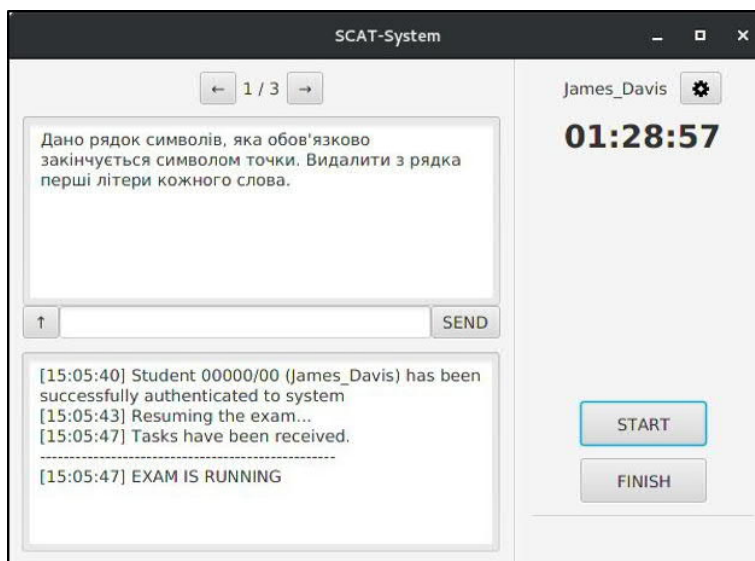


Рисунок 3.11 – Механізм відновлення екзамену, при втраті студентом зв'язку

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Після вичерпання залишку часу, або ж при достроковому завершенні складання іспиту, формується загальна оцінка, на основі здійснених спроб для кожного тестового завдання рисунок 3.12. Повторне складання іспиту в межах даного запуску сервера неможливе.

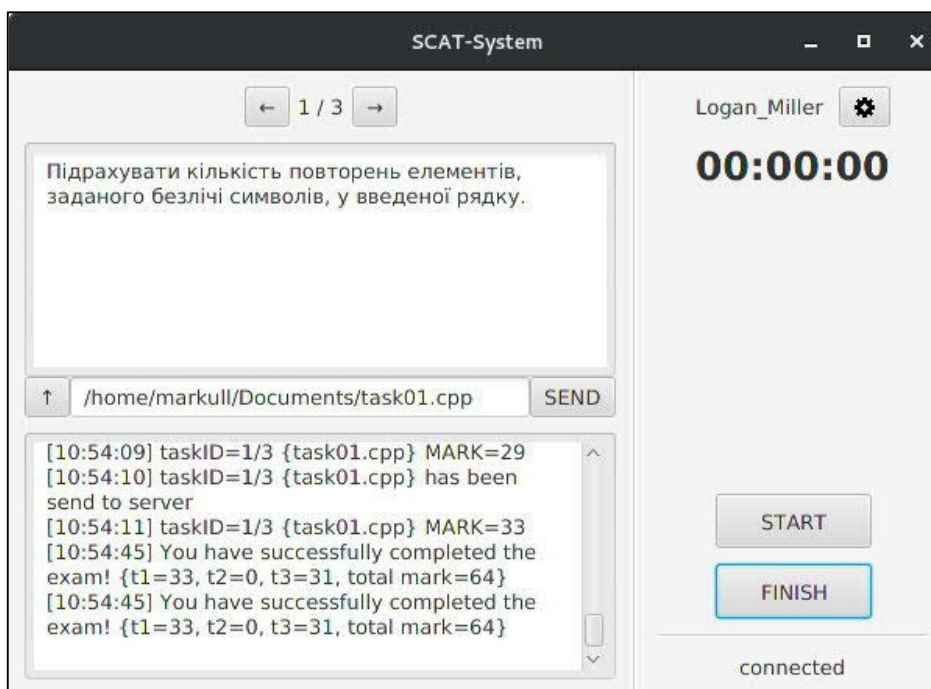


Рисунок 3.12 – Завершення складання тестового іспиту

Частота надсилання запитів перевірки тестового рішення (вихідного коду) визначена з інтервалом не більше однієї спроби в хвилину. Інтервал часу, за який студент може відновити процес проходження тестового іспиту складає три хвилини.

Ліміт надсилання запитів забезпечує серверу захист від перенавантаження кількості оброблювальних процесів. Також обмежує в кількості спроб для одного студента при проведенні визначеного екзамену. Інтервал відновлення студентом процесу проходження тестового іспиту забезпечує контроль над неправомірною поведінкою, у випадку використання функціональної можливості відновлення іспиту в спробах обману та збільшення доступного часу. Сервер здійснює перевірку цих умов, виводить вікно з відповідним повідомленням та забороняє

виконувати запити на перевірку тестового рішення та відновлення процесу проведення тестування у випадку невиконання визначених умов.

На основі розробленої системи проведено порівняння з аналогами. Результати порівнянь наведено в таблиці 3.1.

Таблиця 3.1 – Результати порівнянь SCAT-System з аналогами

Можливості	SCAT-System	Програмна реалізація аналогів					
		MyTestXPro	X-TLS	INDIGO	Moodle	OpenTest	Let's test
Додавання нових видів тестових завдань, окрім основних	X	-	-	-	-	-	-
Налаштування шкали оцінок	-	X	X	X	X	X	X
Експорт таблиці з результатами	X	X	X	X	X	X	X
Безкоштовне розповсюдження ліцензійної версії	X	-	X	-	-	-	-
Доопрацювання модулів програми, інтегрування власних блоків	X	-	X	-	-	-	-
Зміна дизайну інтерфейсу програми	-	-	-	X	X	X	X
Налаштування розкладу часу проведення тестувань	-	-	-	X	X	X	X

SCAT-System містить ряд переваг та недоліків в порівнянні з аналогами. Більшість аналогів розповсюджується на умовах платної комерційної ліцензії. Це робить їх закритими для змін і доопрацювань. Такі програмні рішення також не дають змоги вносити нові види тестових завдань. У випадку open-source проекту SCAT-System підтримка та зміна доступна всім охочим. Гнучка архітектура дозволяє допрацьовувати нові модулі для тестових іспитів з певної предметної області. Інформація про тестові іспити зберігається в університетській базі даних і є захищеною від стороннього втручання.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

В цьому розділі бакалаврської роботи (БР) проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання програмного забезпечення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблене програмне забезпечення призначене для автоматизації процесу проведення тестового іспиту з інтегрованим модулем трьохетапної перевірки якості вихідного C++ коду.

4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (K) включають:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн;

K_2 – витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди (B_{ϕ});
- витрати на покупні вироби ($B_{пв}$);

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- витрати на придбання спецобладнання для проведення експериментальних робіт (O_B);
- накладні витрати (H);
- інші витрати (I).

4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці програмного забезпечення задіяні наступні спеціалісти – розробники, а саме – керівник проекту, студент-дипломник, консультант техніко-економічного розділу.

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн
Керівник ВКР, старший викладач	6397
Консультант техніко-економічного розділу, доцент	7293
Студент	1400

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год (приймаємо 168 год).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год	Витрати на розробку, грн
Керівник ВКР, старший викладач	20	47,6	951,93
Консультант техніко-економічного розділу, доцент	2	63,81	127,62
Студент	100	8,3	833,33
Разом		1912,89	

4.1.2 Відрахування на соціальні заходи

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

$$B_{\phi} = \frac{20,5}{100} \cdot 1912,89 = 392,14 \text{ грн.} \quad (4.4)$$

4.1.3 Розрахунок витрат на матеріали та комплектуючі

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

Найменування купованих виробів	Одиниця виміру	Ціна, Грн.	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
Папір (формат А4)	уп	110,0	1	110,00	11,0	121,0
Ручка кулькова	шт	13,0	2	26,00	2,6	28,6
Олівець простий	шт	2,0	1	2,00	0,2	2,2
Зошит, 18 арк	шт	5,0	2	10,00	1	11,0
Тонер для принтера	уп	47	1	47	4,7	51,7
Разом						214,5

4.1.4 Витрати на використання комп'ютерної техніки

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу ІВМ РС/АТХ вартість години роботи становить 5,2 грн. Середній щоденний час роботи на комп'ютері 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера, грн.
Проведення досліджень та оформлення їх результатів	75	390
Оформлення техніко-економічного розділу	10	52
Оформлення ВКР	18	93,6
Разом	103	535,6

4.1.5 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1912,89 = 2869,34 \text{ (грн)}. \quad (4.5)$$

4.1.6 Інші витрати

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 1912,89 \cdot 0,1 = 191,29 \text{ (грн)}. \quad (4.6)$$

Витрати на розробку програмного забезпечення складають:

$$K_I = B_{ОП} + B_{\Phi} + B_{ПВ} + H + I, \quad (4.7)$$

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

$$K_1 = 1912,89 + 392,14 + 214,5 + 2869,34 + 191,29 = 5580,17 \text{ (грн)}. \quad (4.8)$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.9)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн/год.

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 40 днів. Середній щоденний час роботи на комп'ютері 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому:

$$K_2 = 5,2 \cdot 103 = 535,6 \text{ грн}. \quad (4.10)$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 2.5).

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн
Витрати на оплату праці	1912,89
Відрахування у спеціальні державні фонди	392,14
Витрати на куповані вироби	214,5
Накладні витрати	2869,34
Інші витрати	191,28
Витрати на відлагодження і дослідну експлуатацію програмного продукту	535,6
Разом	6115,77

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.11)$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ – вартість роботи комп'ютера для розробки програмного забезпечення (аналог), грн.

Річні експлуатаційні витрати B_{en} визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} \cdot N_{\Pi}, \quad (4.12)$$

де N_{Π} – періодичність експлуатації ПЗ (аналог), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i, \quad (4.13)$$

де i – категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i – кількість працівників i -ої категорії, осіб;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0(1+b)}{m}, \quad (4.14)$$

де c_i^0 – основна місячна заробітна плата працівника i -ої категорії, грн;

b – коефіцієнт, який враховує додаткову заробітну плату;

m – кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмного забезпечення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає:

$c^0 = 1400$ грн. Тоді:

$$c_1 = \frac{1400(1+0)}{22 \cdot 8} = 7,95 \text{ грн/год} \quad (4.15)$$

Трудомісткість підготовки даних для програмного забезпечення складає 1 год., для аналога 1,5 год.

Таблиця 4.6 – Розрахунок витрат на підготовку даних та реалізацію програмного забезпечення на комп'ютері

Час роботи співробітників, год	Середньогодинна заробітна плата, грн/год	Витрати, грн.
Проектне рішення		
1	7,95	13,15
Аналог		
1,5	11,93	19,73

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t \cdot S_{MG}, \quad (4.16)$$

де t – витрати машинного часу для реалізації програмного продукту, год;

S_{MG} – вартість однієї години роботи комп'ютера, грн/год.

$$E_{2\Pi} = 1 \cdot 5,2 = 5,2 \text{ (грн)}; \quad E_{2A} = 1,5 \cdot 5,2 = 7,8 \text{ (грн)} \quad (4.17)$$

$$E_{\Pi} = 7,95 + 5,2 = 13,15 \text{ (грн)}; \quad E_A = 11,93 + 7,8 = 19,73 \text{ (грн)} \quad (4.18)$$

$$B_{E\Pi} = 13,15 \cdot 252 = 3314,94 \text{ (грн)}; \quad B_{EA} = 19,73 \cdot 252 = 4972,41 \text{ (грн)} \quad (4.19)$$

4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання – це витрати на придбання і експлуатацію програмного продукту за весь строк його служби:

$$Ц_{C(\Pi)} = Ц_{\Pi} + B_{(E)NPV}, \quad (4.20)$$

де $Ц_{\Pi}$ – ціна придбання програмного продукту, грн.

$$Ц_{\Pi} = K \left(1 + \frac{\Pi_p}{100}\right) + K_0 + K_k, \quad (4.21)$$

де K – кошторисна вартість;

Π_p – рентабельність;

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

K_0 – витрати на прив'язку та освоєння програмного забезпечення на конкретному об'єкті, грн;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

$$C_{II} = 6115,77 \cdot (1+0,3) = 7950,5 \text{ (грн)}. \quad (4.22)$$

Вартість витрат на експлуатацію програмного забезпечення (за весь час його експлуатації), грн:

$$B_{eПpv} = \sum_{t=0}^T \frac{B_{eП}}{(1+R)^t} \quad (4.23)$$

де $B_{eП}$ – річні експлуатаційні витрати, грн;

T – термін служби програмного забезпечення, років;

R – річна ставка проценту банку.

$$B_{EПPV} = \sum_{t=1}^5 \frac{3314,94}{(1+0,08)^t} = 13235,61 \text{ (грн)} \quad (4.24)$$

$$B_{EAPV} = \sum_{t=1}^5 \frac{4972,41}{(1+0,08)^t} = 19853,42 \text{ (грн)} \quad (4.25)$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

$$C_{СП} = 7950,5 + 13235,61 = 21186,11 \text{ (грн)} \quad (4.26)$$

Аналогічно визначається ціна споживання для аналогу:

$$C_{СА} = 7000,0 + 19853,42 = 26853,42 \text{ (грн)} \quad (4.27)$$

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмного продукту:

$$E_{PP} = Ц_{П} - Ц_{А} \quad (4.28)$$

$$E_{PP} = 7950,5 - 7000,0 = 950,5 \text{ (грн)} \quad (4.29)$$

Річний економічний ефект в сфері експлуатації:

$$E_{КС} = B_{ЕА} - B_{ЕП} \quad (4.30)$$

$$E_{КС} = 4972,41 - 3314,94 = 1657,47 \text{ (грн)} \quad (4.31)$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{ЕКС} = \sum_{t=1}^T E_{ЕКС} (1+R)^{T-t} \quad (4.32)$$

$$\Delta E_{ЕКС} = \sum_{t=1}^5 165747 \cdot (1+0,08)^{5-t} = 9723,73 \text{ (грн)} \quad (4.33)$$

Сумарний ефект складає:

$$E = E_{PP} + \Delta E_{ЕКС} = 950,5 + 9723,73 = 10674,23 \text{ (грн)} \quad (4.34)$$

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 4.7 – Показники економічної ефективності програмного забезпечення

Найменування	Одиниці вимірювання	Значення показників	
		Базовий варіант	Новий варіант
Капітальні вкладення	грн.	-	6115,77
Ціна придбання	грн.	7000,0	7950,5
Річні експлуатаційні витрати	грн.	13235,61	19853,42
Ціна споживання	грн.	21186,11	26853,42
Економічний ефект в сфері проектування	грн.	-	950,5
Економічний ефект в сфері експлуатації	грн.	-	1657,47
Додатковий ефект в сфері експлуатації	грн.	-	9723,73
Сумарний ефект	грн.	10674,23	

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Здійснено порівняння з існуючим аналогом, і цим показано, що вказане програмне забезпечення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, зручність. Згідно проведеного економічного обґрунтування зазначене програмного забезпечення є конкурентоздатним. Крім того, отримано економічний ефект у розмірі 10674,23 грн. і тому розробка і впровадження цього програмного забезпечення є економічно доцільними.

ВИСНОВКИ

В даній бакалаврській роботі було спроектовано та реалізовано систему проведення тестових іспитів з наявністю інтегрованого модуля трьох-етапної перевірки якості програмного коду на мові C++.

Система базується на клієнт-серверній архітектурі в межах локальної комп'ютерної мережі (з використанням сокетів для обміну даними). Опрацьовує запити для користувачів типу адміністратор, викладач та студент за наявним API, що визначає чіткі правила взаємодії. Містить функціонал:

- розподілення тестових завдань;
- трьох-етапного тестування якості програмного коду на мові C++;
- визначення кінцевої оцінки складення тестового іспиту;
- ефективного збереження даних про перебіг проведення тестувань.

Реалізована система забезпечує гнучкий механізм автоматизованого тестування. Відкритість для інтеграцій нових програмних модулів, дозволяє розширювати перелік предметних областей доступних для обробки системою. Єдина база та наявність логувальника подій забезпечує прозорість проведення тестових іспитів для студентів та викладачів.

Результати розробки системи доповідалися на інтернет-конференції «Науково-практична конференція інтелектуальних комп'ютерних систем та мереж» (додаток Б).

Розроблена система має практичне значення, що підтверджено довідкою про використання (додаток В).

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Заболотна Г.М. Стандартизовані методи тестування знань студентів. Львів: ВНТЛ, 2004. 320 с.
2. Яковець Ф.М., Лещенко Л.Ю., Супрун О.Л. Методика моніторингу та оцінювання знань : довідник. Луцьк, 2011. 453 с.
3. Бондарець Д.В. Методика моніторингу та оцінювання знань. Рівне: Знання, 2009. 254 с.
4. Литовченко Н.В., Мельничук О.Д., Лесів В.О. Системні підходи до визначення ступеня засвоєння навчальної дисципліни : довідник. Одеса, 2001. 545 с.
5. Ромашко Л.В., Климчук А.М. Педагогіка : навч. посіб. Вид. 2-ге, переробл. і допов. Львів, 2014. 283 с.
6. Драган Т.Д., Ворона С.П., Дмитрук Н.В. Вимоги до контролю та оцінювання навчальних досягнень студентів вищих навчальних закладів : довідник. Рівне, 2010. 430 с.
7. Клієнт-серверна система для безпечного обміну приватними повідомленнями/ Левчук В.Л., Шевчук Б.О., Артеменко К.Б., Когут Н.А. Київ : Основи, 2003. 215 с.
8. Войтенко Д.А., Івасів В.А., Кириленко С.С. Способи впровадження автоматизованих тестувань : довідник. Вінниця, 2012. 183 с.
9. Коваль О.І., Михайлюк Н.Н., Журавель Т.В. Алгоритми автоматизованих тестувань : довідник. Харків, 2005. 310 с.
10. Пилипчук В.В., Волкова М.Д. Сучасні методи оцінки якості вихідного коду : довідник. Київ, 2014. 390 с.
11. Матвійчук Л.А. Використання комп'ютерної техніки для перевірки якості знань. Львів: Свічадо, 2003. 296 с.
12. Хоменко Л.М., Ткач І.М., Яковенко П.П. Програмні системи оцінювання знань : довідник. Луцьк, 2010. 223 с.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

13. Тестування програмного забезпечення / Гончарук В.А., Марчук З.Т., Антонюк М.В., Григоришин Я.Б. Одеса : Маяк, 2015. 434 с.
14. Федик Г. Є. Test Driven Development як методика розробки програмного коду / пер. з англ. М. Верба та ін. Дніпро : ДВУ, 2006. 873 с.
15. Маслій Д. П. Модель архітектури «клієнт-сервер» / пер. з англ. А. Зубченко Київ : Молодь, 2016. 383 с.
16. Вербіцький О.В. Клієнт-серверні системи. Тернопіль: Джура, 2000. 323 с.
17. Олешко В. С. Багатопоточність на Java / пер. з англ. В. Поворозник Київ : КНЕУ, 2008. 530 с.
18. Антонов В. В. Багатопоточність та синхронізація / пер. з англ. К. Сулима Київ : КНЕУ, 2004. 393 с.
19. Рудюк В. А. Підходи до проектування захищеної бази даних / пер. з англ. Р. Коновалов Луцьк : Вежа Друк, 1997. 998 с.
20. Система оцінювання знань: веб-сайт. URL: http://ua.wikipedia.org/wiki/Система_оцінювання_знань (дата звернення: 23.02.2019).
21. Платформи для організації олімпіад по програмуванню: веб-сайт. <https://habr.com/ua/company/cloud4u/blog/346838/> (дата звернення: 25.02.2019).
22. Шкала оцінювання ECTS: веб-сайт. URL: <http://law.nau.edu.ua/uk/caphedras/46-8estimate> (дата звернення: 13.03.2019).
23. Клієнт серверна архітектура та ролі серверів: веб-сайт. URL: <http://www.eco-live.com.ua>(дата звернення: 08.04.2019).
24. Локальна комп'ютерна мережа: веб-сайт. URL: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/21/21.html> (дата звернення: 09.04.2019).
25. Статичний та динамічний аналіз коду: веб-сайт. URL: <https://www.viva64.com/ru/b/0248/> (дата звернення: 10.04.2019).
26. Тестування програмного забезпечення: веб-сайт. URL: <http://lits.ua/qa-school/test-base-course> (дата звернення: 20.05.2019).

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

27. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» підготовки 6.050102 комп'ютерна інженерія/ І.Р. Паздрій Тернопіль: ТАНГ, 2014. 37 с.

28. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2016. 65с.

29. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

					БР.КСМ 07117/15.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61