

Міністерство освіти і науки України
Державний університет "Одеська політехніка"
Навчально-науковий інститут комп'ютерних систем
Кафедра системного програмного забезпечення

Чечуй Артур Вікторович,
студент групи АС-161

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Програмний засіб для навчання шкільному курсу геометрії

Спеціальність:

121 – Інженерія програмного забезпечення

Освітня програма:

Інженерія програмного забезпечення

Керівник:

Паулін Олег Миколайович,
канд. техн. наук, професор

Одеса – 2021

ЗМІСТ

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	4
АНОТАЦІЯ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ	10
1.1 Опис та аналіз предметної області	10
1.2 Принципи роботи засобів для навчання	14
1.3 Огляд існуючих аналогів для навчання геометрії	17
1.4 SWOT-аналіз проекту	
1.5 Висновки до розділу	19
РОЗДІЛ 2 РОЗРОБКА ЗАСТОСУНКУ З НАВЧАННЯ ГЕОМЕТРІЇ	20
2.1 Створення моделі навчання	20
2.2 Розробка підсистеми навчання	22
2.3 Розробка підсистеми тестування	24
2.4 Висновки до розділу	26
РОЗДІЛ 3 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАСОБУ	27
3.1 Варіанти використання програмного засобу	27
3.2 Вимоги до нефункціональних характеристик засобу	39
3.3 Висновки до розділу	41
РОЗДІЛ 4 ПРОЕКТУВАННЯ ЗАСОБУ З НАВЧАННЯ ГЕОМЕТРІЇ	42
4.1 Проектування архітектури засобу	42
4.2 Основні діаграми роботи	44
4.3 Проектування реляційної бази даних	53
4.4 Розробка прототипів інтерфейсу	57
4.5 Проектування структури класів програмного засобу	60
4.6 Висновки до розділу	63
РОЗДІЛ 5 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУВАННЯ	64
5.1 Інструменти створення навчального засобу	64

5.2 Приклади використання навчального засобу	65
5.3 Висновки за розділом	68
РОЗДІЛ 6 ВИЗНАЧЕННЯ ВЛАСТИВОСТЕЙ ЗАСОБУ ДЛЯ НАВЧАННЯ ГЕОМЕТРІЇ	69
6.1 Тестування функціональності з використанням сценаріїв	69
6.2 Експериментальна перевірка зниження часу на навчання	74
6.3 Висновки до розділу	77
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
Додаток А. ЛІСТИНГ ПРОГРАМИ	81

Міністерство освіти і науки України
Державний університет "Одеська політехніка"
Навчально-науковий інститут комп'ютерних систем
Кафедра системного програмного забезпечення

Рівень вищої освіти: другий (магістерський)

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Любченко В.В.

«__» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Чечую Артуру Вікторовичу, група АС-161

1. Тема роботи: Програмний засіб для навчання шкільному курсу геометрії
Керівник роботи: Паулін Олег Миколайович, канд. техн. наук, професор
затверджені наказом ректора від «25»жовтня 2021 р. № 374-в.
2. Зміст роботи: аналіз існуючих рішень для навчання шкільному курсу геометрії; створення моделі навчання, розробка підсистем навчання та тестування; визначення функціональних та нефункціональних вимог до засобу; проектування програмного засобу; вибір програмних інструментів розробки; прототипування інтерфейсу; програмна реалізація навчальної системи; тестування функціональності системи для визначених варіантів використання, визначення часових властивостей навчального програмного засобу.
3. Перелік ілюстративного матеріалу: згідно зі слайдами презентації.

4. Консультанти розділів роботи

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

5. Дата видачі завдання: « ___ » _____ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	Аналіз існуючих рішень	2.09.2021 – 10.09.2021	вик.
2	Специфікація вимог до програмної системи	11.09.2021 – 23.09.2021	вик.
3	Проектування програмної системи	24.09.2021 – 17.10.2021	вик.
4	Програмна реалізація системи	18.10.2021 – 14.11.2021	вик.
5	Тестування системи	15.11.2021 – 23.11.2021	вик.
6	Оформлення пояснювальної записки та графічного матеріалу	24.11.2021 – 30.11.2021	вик.

Здобувач вищої освіти _____ А.В.Чечуй

Керівник роботи _____ О. М. Паулін

АНОТАЦІЯ

Метою кваліфікаційної роботи є зниження часу на вивчення теоретичних принципів та практичних завдань зі шкільного курсу геометрії. Методи програмної розробки базуються на мові Java, середовищі розробки IntelliJ Idea, мові JavaScript та реляційній базі даних SQLite.

Як результат розроблено програмний продукт для вивчення та отримання практичних навичок роботи з матеріалами шкільного курсу геометрії.

Ключові слова: програмний засіб, навчання, геометрія, завдання, тест, проектування, Java, IntelliJ Idea, JavaScript, SQLite.

ABSTRACT

The purpose of the qualification work is to reduce the time for studying the theoretical principles and practical tasks of the school course of geometry. Software development methods are based on the Java language, the IntelliJ Idea development environment, the JavaScript language and relational database SQLite.

As a result, a software product was developed to study and gain practical skills in working with materials for the school course of geometry.

Keywords: software, training, geometry, tasks, test, designing, Java, IntelliJ Idea, JavaScript, SQLite.

ВСТУП

В останній час різко зростає необхідність підключення до освітнього процесу можливостей використання комп'ютерних систем навчання. Якщо раніше основними аргументами для їх застосування було перевантаження вчителів при великій наповнюваності навчальних груп, та можливість підвищити мотивацію завдяки використанню цікавих та різноманітних комп'ютерних інструментів, то з початком пандемії на COVID-19 велика доля навчання перейшла виключно у онлайн-формат. Примитивна демонстрація вчителями на екрані методичних матеріалів не дозволяє заохотити учнів до опанування знань згідно до шкільного навчального плану. Зростає кількість відстаючих учнів, що не є мотивованими до якісного навчання. Практично неможливим стає застосування індивідуального підходу до учнів та підтримка постійного стеження за їх навчальною діяльністю задля формування стійких навичок опановувати новий матеріал та закріплювати його у ході практичної діяльності.

На цьому тлі виникає гостра необхідність у розробці та подальшому застосуванні програмних засобів, які дозволять покращити процес навчання та знизити кількість невстигаючих учнів. При цьому зрозуміло, що комп'ютерні навчальні системи можна використовувати як окремо, так і в комплексі зі звичайним онлайн-навчанням, у тому числі у STREAM-освіті [1].

При застосуванні спеціальних засобів, що дозволяють автоматизувати деякі навчальні функції, різко зростає мотивація учня, який може вивчати матеріал у припустимому для себе темпі, у зручному для себе час та може порівнювати свої успіхи з результатами навчання інших учнів у разі, якщо програмний засіб надає таку можливість. Тому доцільним є ретельний аналіз та вибір технологій навчання [2] з використанням відповідних методів та моделей [3].

При використанні автоматизованих програмних засобів для навчання вчителів, що веде вказаний шкільний курс, надаються інструменти для налаштування проходження учбового курсу. Це є дуже корисним, та допомагає

скоротити час на навчання при забезпеченні потрібного рівня якості опанування учбових матеріалів.

Метою даної кваліфікаційної роботи є зменшення часу на вивчення учнем шкільного курсу геометрії. Для досягнення цієї поставленої мети потрібно вирішення таких задач:

- опис та аналіз предметної області, основних понять з напрямку створення навчальних засобів, вивчення існуючих аналогів, їх переваг та недоліків;
- створення моделі навчання, що буде основою розроблюваного програмного засобу;
- формалізація функціональних та нефункціональних вимог до навчального засобу;
- створення архітектури, проектування бази даних навчального засобу;
- вибір мови програмування та програмна реалізація засобу;
- тестування працездатності засобу.

Звіт є результатом виконання магістерського кваліфікаційного проектування. Він містить вступ, шість основних розділів, висновки, список використаної літератури та додаток.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Опис та аналіз предметної області

Термін «геометрія» з'явився ще у стародавні дні, та дослівно переводиться як «вимірювання землі». Але зараз геометрія відійшла далеко від свого початкового призначення.

У середній школі курс геометрії починається у 7 класі та продовжується до кінця шкільного навчання (11 чи 12 класи). Матеріали з геометрії входять до складу завдань державної підсумкової атестації та зовнішнього незалежного оцінювання з математики. Здача екзамену та саме бали, отримані на зовнішньому незалежному оцінювання з математики, грають ключову роль для можливості вступу на багато спеціальностей у заклади вищої освіти. Все це лише підкреслює важність якісного вивчення геометрії, починаючи з тих основ, що викладаються у 7 класі.

Шкільна геометрія включає у себе наступні ідеї, що втілюються у відповідні навчальні теми:

- геометричні фігури та особливості їх будови;
- властивості геометричних фігур у планіметрії та стереометрії;
- теорія геометричних побудов та перетворень;
- методи та засоби аналітичної геометрії;
- аксіоматичний метод та теорія логічних умовиводів.

Всі ці ідеї вивчаються поступово, закріплюються прикладами вирішення завдань та етапами тестування знань учнів. Обов'язкові повторення пройденого матеріалу дозволяють підтримувати в активному стані знання щодо раніше вивчених тем.

На рис. 1.1 в узагальненому вигляді показана структура курсу з шкільної геометрії. Кількість класів може відрізнятись в залежності від шкільного навчального плану (11-ти або 12-ти річне навчання у школі). Кількість тем та підтем у кожній темі може налаштовуватись вчителем теж в залежності від

навчального плану та особливостей конкретної школи. Під терміном «порція» розуміється одиниця учбового матеріалу, що може використовуватись як для навчання, так і для тестування з метою перевірки знань учнів. Порцією може бути окремий файл з теорією, зображення, відеоролик, набір тестів, посилання на інтернет-ресурс, тощо.

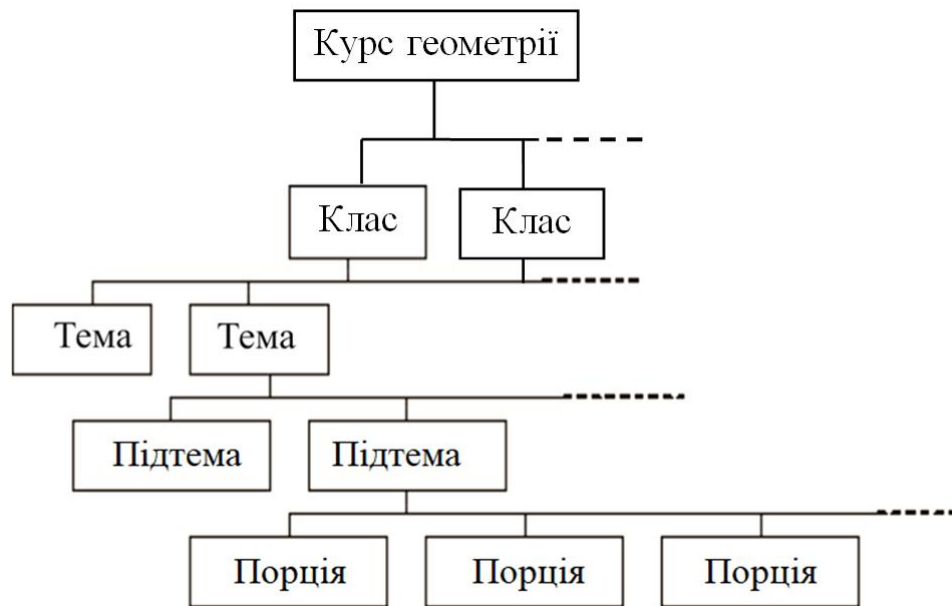


Рисунок 1.1 – Узагальнена структура курсу з шкільної геометрії

За допомогою програмного навчального засобу учні повинні мати можливість опанувати наступні властивості та визначення стосовно геометричних величин:

- мірою довжини є одиничний відрізок, кута – кут у 1 градус, площини – одиничний квадрат, об’єму тіла – одиничний куб;
- визначення міри довжини як позитивного числа, яке показує скільки разів одиничний відрізок можна вмістити у даний вимірюваний відрізок; визначення міри кута як градусної величини, що визначає кількість разів, яку кут в один градус може бути укладений в вимірюваний кут; аналогічне визначення одиничної площини та одиничного кубу як позитивного числа для вимірювання за його допомогою площини чи об’єму фігури;

– нормованість: для довжини це одиничний відрізок, що береться за одиницю виміру, наприклад, 1 мм, 1 см, 1 м; для величини кута це кут з градусною мірою 1 градус; для площини це квадрат зі стороною, що дорівнює одиниці виміру довжини; для об'єму це куб, ребро якого дорівнює одиниці виміру довжини, та приймається за одиницю об'єму;

– інваріантність: визначає, що довжина відрізка не залежить від його розташування, тобто незалежно від орієнтації відрізка у просторі рівним відрізкам відповідають рівні довжини; якщо кути є рівними, то й їх градусні кутові значення є рівними; якщо фігури є рівними, то й їх площини та об'єми мають однакові значення;

– адитивність: якщо деякий відрізок є розбитим на частини, що не перетинаються один з одним, то довжина цього відрізка визначається як сума довжин його частин; кутова міра суми кутів визначається як сума вимірювання їх кутових значень; якщо пласка (двовимірна) фігура складається з інших фігур, що не перетинаються одна з іншою, то її площа складається як сума площин цих фігур; аналогічно, якщо тривимірна (об'ємна) фігура складається з декількох фігур, що не перетинаються, то об'єм цієї фігури складається з суми об'ємів вказаних фігур.

Для можливості тестування ступеня опанування знань учнями визначаються базові типи завдань для метричної, а також афінної геометрії. Наведемо основні з типів таких завдань:

– метричні завдання, що належать до програми 7 класу: ознаками типа завдання є наявність умови зі значеннями довжин відрізків, площин, периметрів, кутів, об'ємів фігур, тощо, а також відношення перпендикулярності чи паралельності між вказаними елементами; виглядом перетворення є рух, при якому задані осьова та центральна симетрії, виконання паралельного перенесення, здійснення повороту; у якості інваріантів перетворення виступають довжина відрізка, об'єм, площа, значення кута, належності точки до прямої, відома паралельність прямих, відомий перетин прямих;

– завдання на подібність (8 класу): містять теж ознаки типу завдання як наявність умови зі значеннями довжин периметрів, відрізків, кутів, площин та об'ємів фігур, а також відомі відношення перпендикулярності чи паралельності між вказаними елементами; виглядом перетворення є гомотетія та подібність; інваріантами перетворень виступають значення кута, відношення периметрів, площин, відрізків подібних фігур та їх об'ємів, також паралельність та перетин прямих ліній;

– афінні завдання у якості умови мають відношення паралельності прямих, відношення довжин відрізків та площин, при яких немає точних значень їх кіл та значень геометричних величин, та ін.; виглядом перетворення є паралельне проектування фігури на площину та відображення як паралельне проектування площини на площину; з точки зору інваріантів перетворення проекцією будь-якої прямої є теж пряма, залишається інваріантність паралельності прямих та площин фігур, а також відношення відрізків на одній й тій же самій прямій та на різних, але паралельних прямих;

– проєктивні завдання у якості умови мають відношення довжин відрізків; виглядом їх перетворення є центральна проєкція; інваріантами є перетин прямих та інцидентність точки і прямої.

1.2 Принципи роботи засобів для навчання

Різні класифікації надають різні методики управління навчальним процесом. Під управлінням процесом навчання будемо розуміти надання учневі завдань у визначеній для нього послідовності. Це дозволяє, з одного боку, уникнути ситуації, коли учень вже навчився вирішувати завдання за деяким шаблоном, але система все одно надає йому інші ідентичні завдання для вирішення. З іншого боку, якщо учень не встигає, доцільно надавати йому завдання на повторення матеріалу, а в деякому випадку, навіть налаштовувати на вивчення попередньої

теми. Як буде реалізований механізм керування навчанням – залежить від обраної методики.

У загальному випадку без деталізації методики керування схема управління системою навчання наведена на рис. 1.2.

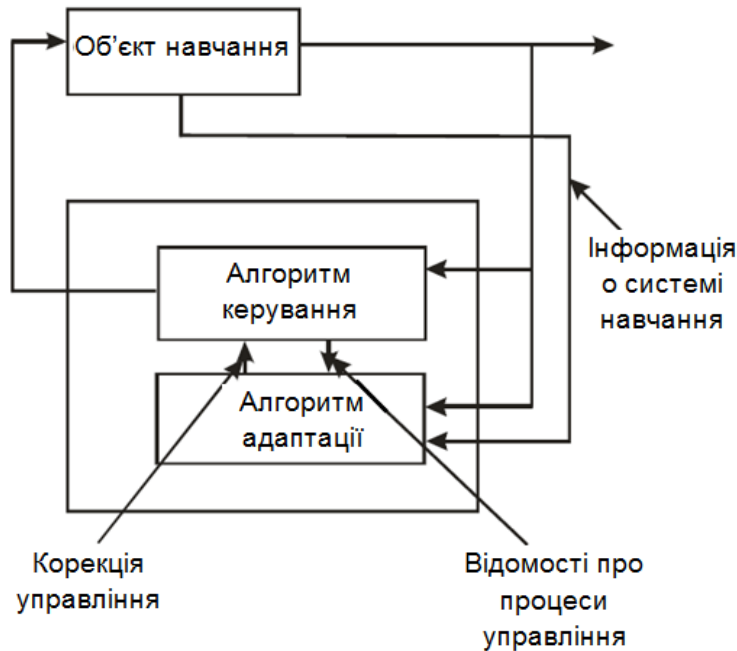


Рисунок 1.2 – Схема управління системою навчання

У рамках даної кваліфікаційної роботи буде використовуватись три відомі методики (алгоритми) управління навчанням:

- послідовне навчання;
- вибіркоче навчання
- адаптивне навчання.

Розглянемо більш детально кожну з методик.

При використанні методики послідовного навчання учень, відповідно до свого класу, проходить всі зазначені теми та підтеми, при цьому опановує теоретичні матеріали та вирішує тестові завдання. На виході кожної теми вчитель отримує оцінки за вирішені учнем тестові завдання, або перевіряє вручну ті завдання, що потребують для перевірки участі вчителя. Кожне завдання може

мати рівень складності – простий, середній або складний. Ця методика відповідає традиційному вивченню шкільного курсу геометрії.

Методика вибіркового навчання є модифікацією методики послідовного навчання з тією різницею, що вчитель обирає ті теми, які потім будуть вивчати учні. Не обрані вчителем теми є схованими від учня, він ніяк не може їх вивчати. Крім того, вчитель може обирати певні теоретичні матеріали та тестові завдання, які повинні бути вирішені учнями. Ця методика дозволяє вивчати окремі теми більш поглиблено (детально), чи при необхідності повертатись до пройденого матеріалу у режимі повторення. Таку методику можна застосовувати для програмних засобів, що дозволяють проводити онлайн математичні гуртки, чи консультативні заняття для відстаючих учнів.

Третя методика – методика адаптивного навчання – є найбільш складною. Її доцільно використовувати у тому разі, коли потрібно оптимізувати процес навчання та скоротити час, що відводиться на опанування навчальних матеріалів, зі збереженням потрібної якості навчання.

Розглянемо основні відомі реалізації систем на основі методики адаптивного навчання. Кожна з них має свою структуру:

- навчальні системи зі зворотним зв'язком: поділяються на імітаційні, тестуючі-навчальні, системи без ігрових елементів, системи з ігровими елементами;
- навчальні системи без зворотного зв'язку: поділяються на презентаційні, тестуючі.

За структурним ознаками взаємодії навчальної системи з користувачем навчальні системи поділяються на два базових класи: розімкнені (без зворотного зв'язку) та замкнуті (зі зворотним зв'язком) системи, які відрізняються принциповим підходом до процесу навчання.

«У розімкнутих навчальних системах не враховуються відгуки учнів на поставлені питання і не коригується послідовність пред'явлення навчального матеріалу в функції міри засвоєння учням досліджуваної теми. Тут лише

виконується певна заздалегідь задана програмним шляхом послідовність викладу уроку або контрольних питань.

Адаптивне навчання і адаптивний тестовий контроль в своїй єдності являють собою сучасний комп'ютерний варіант реалізації принципу індивідуалізації навчання з використанням замкнутої навчальної системи. Адаптивне навчання дозволяє:

- використовувати оцінки, отримані при попередньому адаптивному тестовому контролі для вибору вихідних тематичних блоків навчальної інформації, з якої починається адаптивне навчання;
- забезпечити передачу знань, формування умінь і навичок без пробілів, отримуючи необхідну структуру і бажаний рівень знань, умінь, навичок.

На даний час існують три варіанти проведення адаптивного тестування:

1) пірамідальне тестування - учню дається завдання середньої складності, потім в залежності від правильності відповіді дається завдання легше чи складніше;

2) Flexilevel - контроль починається з будь-якого рівня складності, а потім відбувається поступове наближення до реального рівня підготовленості;

3) Stradaptive - тестування проводиться на основі бази складності завдань, де всі завдання попередньо розділені на рівні складності. Після вірного виконання наступне завдання береться з більш високого рівня складності, після невірної - навпаки. Це вимагає попереднього визначення труднощі всіх завдань.»

Саме методика Stradaptive буде використовуватись у даній роботі при виборі адаптивної форми управління процесом навчання. Це дозволить переходити від однієї теми до наступної лише після повного опанування учнем матеріалу, яка міститься в ньому. При цьому вирішено розробити взяти за основу методику Stradaptive.

Для керування навчальним процесом використовується поняття функції управління. Значення функцій управління для тестових завдань визначається на підставі складності цих завдань на оцінки, отриманої учнем при вирішенні

поточного завдання. Вчитель призначає конкретні значення функцій управління для задач зі шкільного курсу геометрії.

1.3 Огляд існуючих аналогів для навчання геометрії

Аналіз наведених ресурсів та сервісів наданий у табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика програмних аналогів

Назва ПЗ/ Характеристики	Study WithUs	Школа. Геометрия. 7 класс	Web- math	Геометрія (Pocket Edition)	Геометрія. 7-11 класс	Власна розробка
Підтримка всієї шкільної програми	+	-	+	+	+	+
Безкоштовність	+	-	+	-	+	+
Кросплат- форменість	+	-	+	-	-	+
Зручний інтерфейс	+	-	+	+	+	+
Теоретичний матеріал	+	+	+	+	+	+
Тестові завдання	-	-	-	-	+	+
Система підказок	+	+	-	-	-	+
Інтерактивність	+	+	+	+	-	+
Можливість користування з мобільного телефону	+	-	+	+	+	+
Відображення прогресу навчання	+	+	-	-	-	+
Веб- застосування	-	-	+	-	-	+

Наведемо короткий опис програмних аналогів, що дозволяють вивчати шкільний курс геометрії.

Програма «StudyWithUs» є найбільш близьким аналогом до розроблюваного засобу. Вона підтримує учбовий матеріал для усіх класів, є безкоштовною та кросплатформеною, має інтерфейс, який користувач може налаштовувати під себе. Теоретичний матеріал розбитий по темам в залежності від номеру класу. Відсутнє розбиття по підтемам. Також відсутніми є тестові завдання. Є багато інтерактивних підказок та зручне користування прогресом навчання. Програма не є веб-застосуванням, але має свій сайт з описом основних функцій та можливостей.

Програмний застосунок «Школа. Геометрія. 7 клас» є не досить близьким аналогом. Основний недолік – наявність навчального матеріалу тільки за 7 клас, що є обмеженням використання. Програма не є безкоштовною, її можна придбати за підпискою. Вартість підписки складає 250 грн на кожен місяць користування. Інтерфейс програми досить старий та тому незручний. Перевагою програми «Школа. Геометрія. 7 клас» є дуже детальне викладення теоретичного матеріалу і розвинута інтерактивна система підказок. Тестові завдання покривають лише декілька основних теорем, тому загалом можна вважати, що тести відсутні. Програма відображає прогрес у навчанні та навіть має деякі елементи гейміфікації для підвищення мотивації.

«Web-math» є веб-застосунком, що підтримує вивчення геометрії у всіх шкільних класах, є безкоштовним та кросплатформеним. Особливістю викладання теоретичного матеріалу є те, що саме тексту наведено небагато, у більшості тем розробники обмежились кольоровими схемами та рисунками з дуже короткими поясненнями. Це не є дуже зручним. Будь-які тестові завдання та система підказок відсутні. При наведенні на рисунок чи схему вони збільшуються, при бажанні користувач може завантажити їх на власний комп'ютер. У системі не відображається прогрес навчання. Застосування можна використовувати через мобільний браузер на телефоні, при цьому інтерфейс адаптується до мобільного пристрою.

Програма «Геометрія (PocketEdition)» є мобільним застосунком, що дозволяє інтерактивно навчатись геометрії в усіх класах. Програма є платною, її вартість вимірюється десятками доларів в залежності від об'єму додаткового функціоналу. Зручний інтерфейс дозволяє налаштовувати кольорову гаму інтерфейсу та встановлювати потрібну мову. Тестові завдання та система підказок відсутні.

Програмний засіб «Геометрія. 7-11 клас» є повним та безкоштовним. Він містить тестові завдання, але не зберігає прогрес користувача.

Як можна побачити з огляду, більшість програмних ресурсів не мають усіх потрібних можливостей, тому доцільно говорити про необхідність розробки власного навчального засобу для вивчення шкільного курсу геометрії.

1.4 SWOT-аналіз проекту

У роботі проведено SWOT-аналіз проекту (табл. 1.2). SWOT-аналіз зазвичай застосовується для того, щоб визначити сильні та слабкі сторони створюваного проекту, а також можливості та загрози для цього проекту.

Метою проведення такого аналізу є вивчення потенційних перспектив розроблюваного програмного забезпечення.

Таблиця 1.2 – SWOT-аналіз проекту

Переваги:	Недоліки:
<ol style="list-style-type: none"> 1. Низька ціна або безкоштовність продукту. 2. Можливість замінити онлайн-навчання. 3. Простота користування дає можливість використовувати продукт будь-якому вчителю та учню. 	<ol style="list-style-type: none"> 1. Необхідність створювати онлайн-матеріали. 2. Складність долучення вчителів та учнів з поганим інтернет-зв'язком.

Продовження табл. 1.2

<p>Можливості:</p> <ol style="list-style-type: none"> 1. Низьке навантаження на вчителя для випадку автоматичних тестів. 2. Автоматичний збір оцінок. 3. Можливість автоматично створювати звіти. 4. Можливість надавати доступ до звіту з оцінками після кожної теми. 	<p>Загрози:</p> <ol style="list-style-type: none"> 1. Можливі порушення доброчесності серед учнів. 2. Можливі помилки у тестових завданнях. 3. Високе навантаження для вчителя при перевірці завдань в ручному режимі.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Проведений аналіз визначає актуальність та корисність розробки навчального програмного засобу для покращення вивчення шкільного курсу геометрії.

1.5 Висновки до розділу

У цьому розділі проведено опис та аналіз предметної області, визначено загальну структуру шкільного курсу з геометрії. Розглянуто можливі методики для управління процесом онлайн-навчання.

Проведено огляд існуючих аналогів для навчання геометрії: «StudyWithUs», «Школа. Геометрия. 7 класс», «Web-math», «Геометрія (PocketEdition)» та «Геометрія. 7-11 класс». Проведено SWOT-аналіз проекту з визначенням сильних та слабких сторін.

Визначено, що розробка власного навчального засобу є доцільною.

2 РОЗРОБКА ЗАСТОСУНКУ З НАВЧАННЯ ГЕОМЕТРІЇ

2.1 Створення моделі навчання

Перед тим, як розпочати проектування навчального застосунку, потрібно розробити модель навчального процесу. Модель може бути представлена трьома способами:

- графічно;
- у вигляді математичних співвідношень;
- у вигляді узагальнених позначень.

У даній роботі обрано графічний спосіб подання моделі. Вона представлена на рис. 2.1.

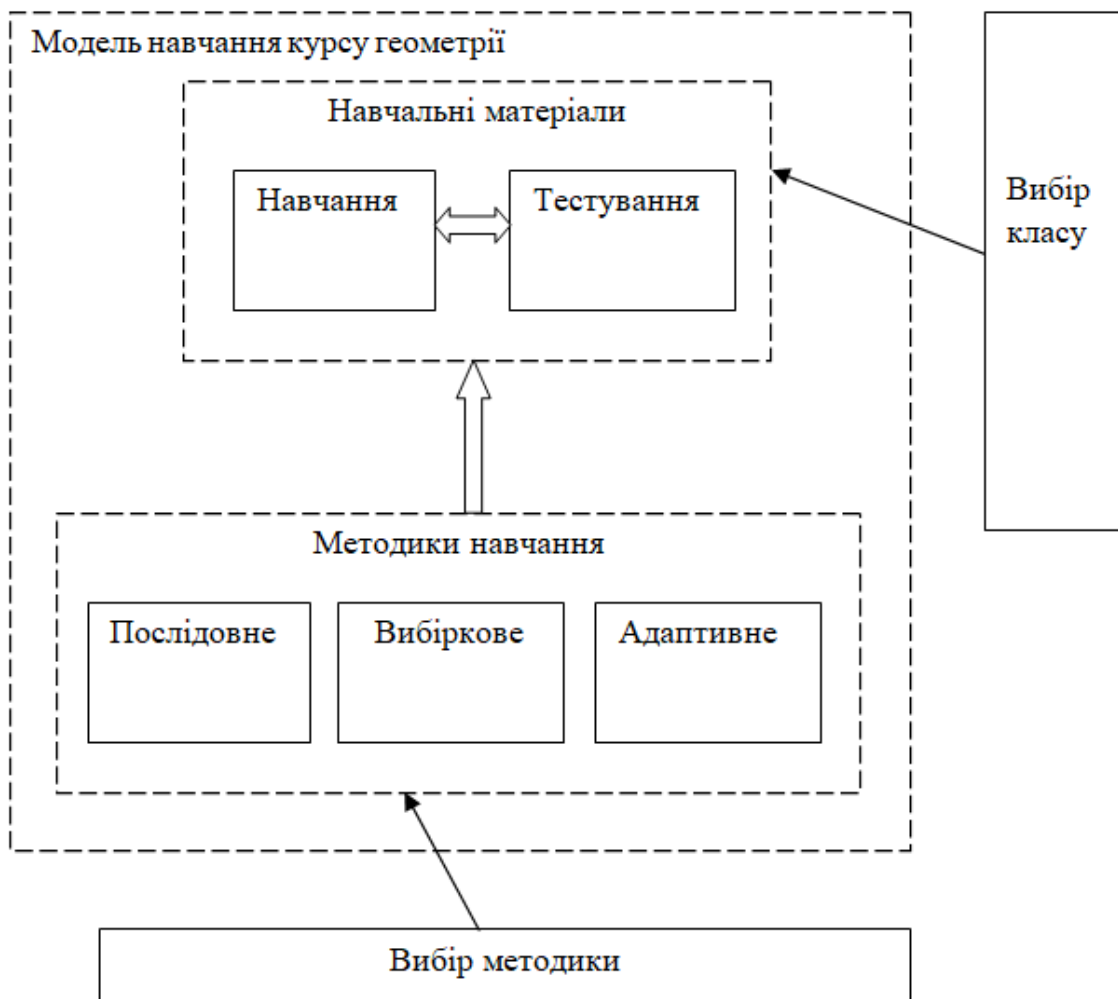


Рисунок 2.1 – Модель навчання шкільному курсу геометрії

Представлена модель включає дві основні частини:

- навчальні матеріали;
- методики навчання.

Навчальні матеріали включають у себе ресурси для опанування тем з геометрії згідно зі шкільною програмою. Матеріали, призначені для навчання, можуть бути наступних типів:

- текстові – звичайний текст у HTML-форматі;
- графічні – зображення, представлені у форматах jpeg, png, bmp;
- відео – відеоролики у форматах mpeg-4 та avi;
- презентації – звичайна або анімована презентація у форматах ppt, pptx.

Тестування знань школярів може виконуватись наступним чином:

- з використанням автоматизованих тестів;
- шляхом перевірки вчителем вручну.

Автоматизовані тести, передбачені системою, повинні включати наступні види:

- тест за принципом «один з багатьох» (лише один з наданих учню варіантів повинен бути обраним);
- тест за принципом «багато з багатьох» (декілька з наданих учню варіантів можуть бути обраними);
- тест за принципом «відповідність» (потрібно зв'язати відношенням відповідності об'єкти з різних списків, під об'єктом може бути використано термін з курсу геометрії, конкретний результат вирішення завдання, графічний об'єкт, тощо).

За кожне правильно вирішене завдання може бути нарахований певний бал. Для тестів за принципом «багато з багатьох» та «відповідність» можуть бути нараховані бали в залежності від ступеня правильності відповіді, що надає можливість врахування балів, якщо на певне питання було надано n правильних відповідей з m при $n < m$.

Перевірка завдань вчителем вручну доцільна у тому разі, якщо завдання передбачає послідовність записів, графічних креслень, розрахунків та умовиводів. Так як структуру відповіді на такі завдання дуже складно формалізувати та, відповідно, алгоритмізувати, то для оцінювання потрібен досвід та участь саме вчителя. Для таких завдань передбачається можливість завантаження учнем рішення у вигляді скрінів.

2.2 Розробка підсистеми навчання

Структурна схема підсистеми навчання наведена на рис. 2.2.

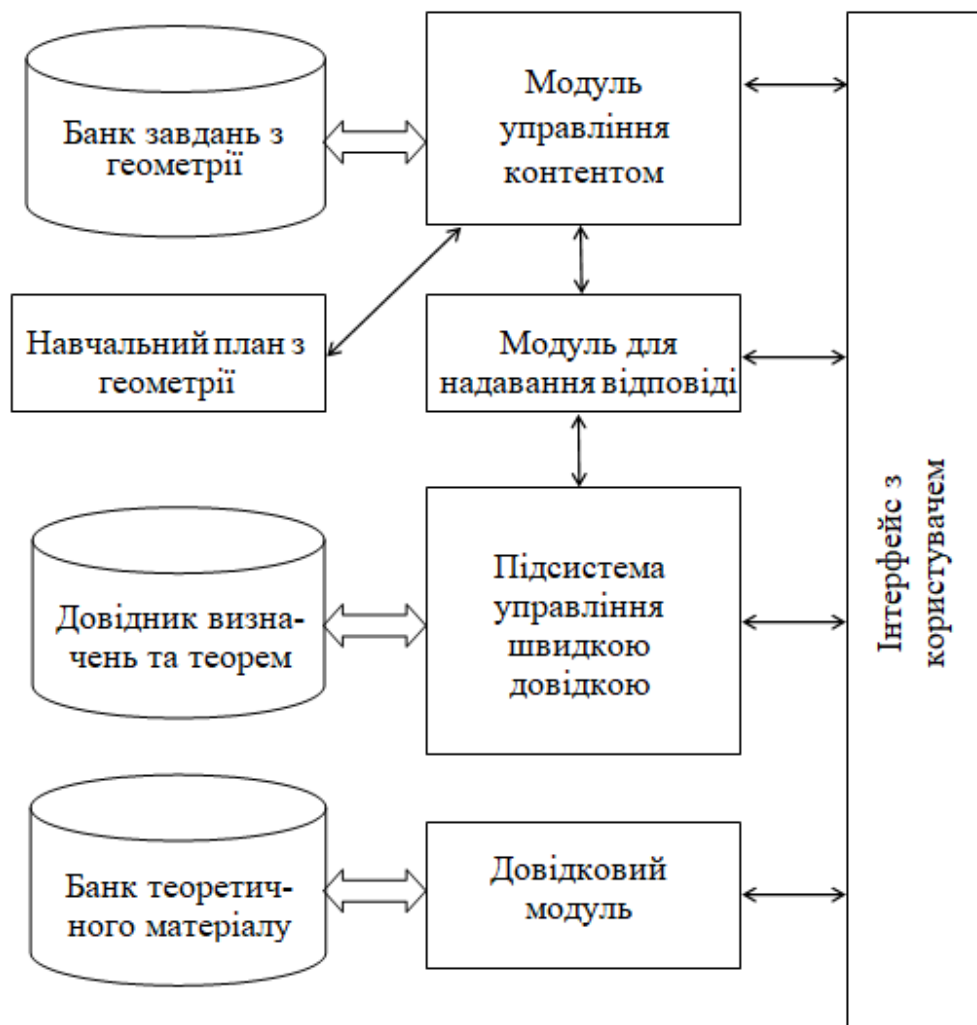


Рисунок 2.2 – Структурна схема підсистеми навчання

Відповідно до навчального плану з геометрії згідно з обраним номером класа (що фактично визначає рівень складності матеріалу), вчитель завантажує у систему контент, тобто завдання з необхідним описом.

Банк завдань містить усі практичні завдання, як автоматизовані тести, так і такі, що розв'язуються вручну учнем та перевіряються вручну вчителем.

Навчальний план з геометрії містить перелік тем та підтем відповідно до класу, у якому вчиться учень. При необхідності можна налаштовувати систему для класів та шкіл з поглибленим вивченням математики. Зрозуміло, що у такому разі навчальний план повинен відрізнитись від звичайного переліком тем та/або підтем з курсу геометрії.

Банк теоретичного матеріалу з геометрії містить всю необхідну сукупність інформації, можливий тип якої вказаний вище. За бажанням пройти теоретичний матеріал учень звертається для відповідної теми та підтеми, які повинні бути заздалегідь завантажені у систему вчителем та перевірені та зручність користування.

У якості швидкої довідки система може надавати основні теореми, визначення та властивості, структуровані згідно з навчальним планом з курсу геометрії.

Взаємодія учнів та вчителів з навчальним застосунком відбувається з використанням інтерфейсу з користувачем. Інтерфейс взаємодіє з наступними блоками застосунку:

- модулем управління контентом – надає можливість вчителю завантажувати, корегувати та видаляти завдання з курсу геометрії;
- модулем для надання відповіді – дозволяє учневі надавати відповіді на тестові питання та завдання, що потребують розгорнутої відповіді;
- підсистема управління швидкою довідкою використовується обома типами користувачів – як учнем, так і вчителем. Учень викликає швидку довідку при створенні відповідей (якщо це не заборонено вчителем, вчитель може заборонити учневі користуватись швидкою довідкою у разі виконання ним

контрольних завдань). Вчитель налаштовує швидку довідку, для цього йому надаються права для завантаження визначень та теорем відповідно до навчального плану, та, при необхідності, їх корегування та видалення, або переном до іншої теми чи підтеми;

– довідковий модуль, до якого також звертаються обидва типи користувачів: учень для опанування нового матеріалу, вчитель – для наповнення системи теоретичним матеріалом та налаштування його розташування та зручності використання.

2.3 Розробка підсистеми тестування

Тепер наведемо більш детальну схему підсистеми тестування, що входить до складу застосунку з навчання шкільному курсу геометрії.

Перевірка навчальних тестових завдань може виконуватись у ручному чи автоматичному вигляді.

Модуль перевірки тестів з геометрії дозволяє автоматично перевіряти тести. Для цього у разі тестів «один-до-багатьох», де лише одна відповідь з усіх може бути правильною, повинен бути вказаний правильний варіант відповіді та оцінка за нього. Для тестів «багато-до-багатьох», при яких правильними можуть бути декілька варіантів, та їх усіх потрібно відмічати, потрібно вказати ці варіанти, а також оцінку за повністю правильну відповідь. У разі, якщо за частково правильну відповідь вчитель хоче теж нараховувати якісь бали, потрібно їх вказати. Для тестів «відповідність» повинні бути встановлені відповідні зв'язки між елементами різних категорій та визначена оцінка за повністю правильну відповідь. Аналогічно тесту «багато-до-багатьох», якщо вчитель бажає нараховувати бали за частково правильну відповідь, потрібно це визначити.

Модуль ручної перевірки завдань вимагає від учня написати рішення вручну та завантажити у систему фото, чи створити текстовий файл з рішенням. Вчитель перевіряє надане рішення та виставляє учню за нього оцінку згідно з встановленим максимальним балом.

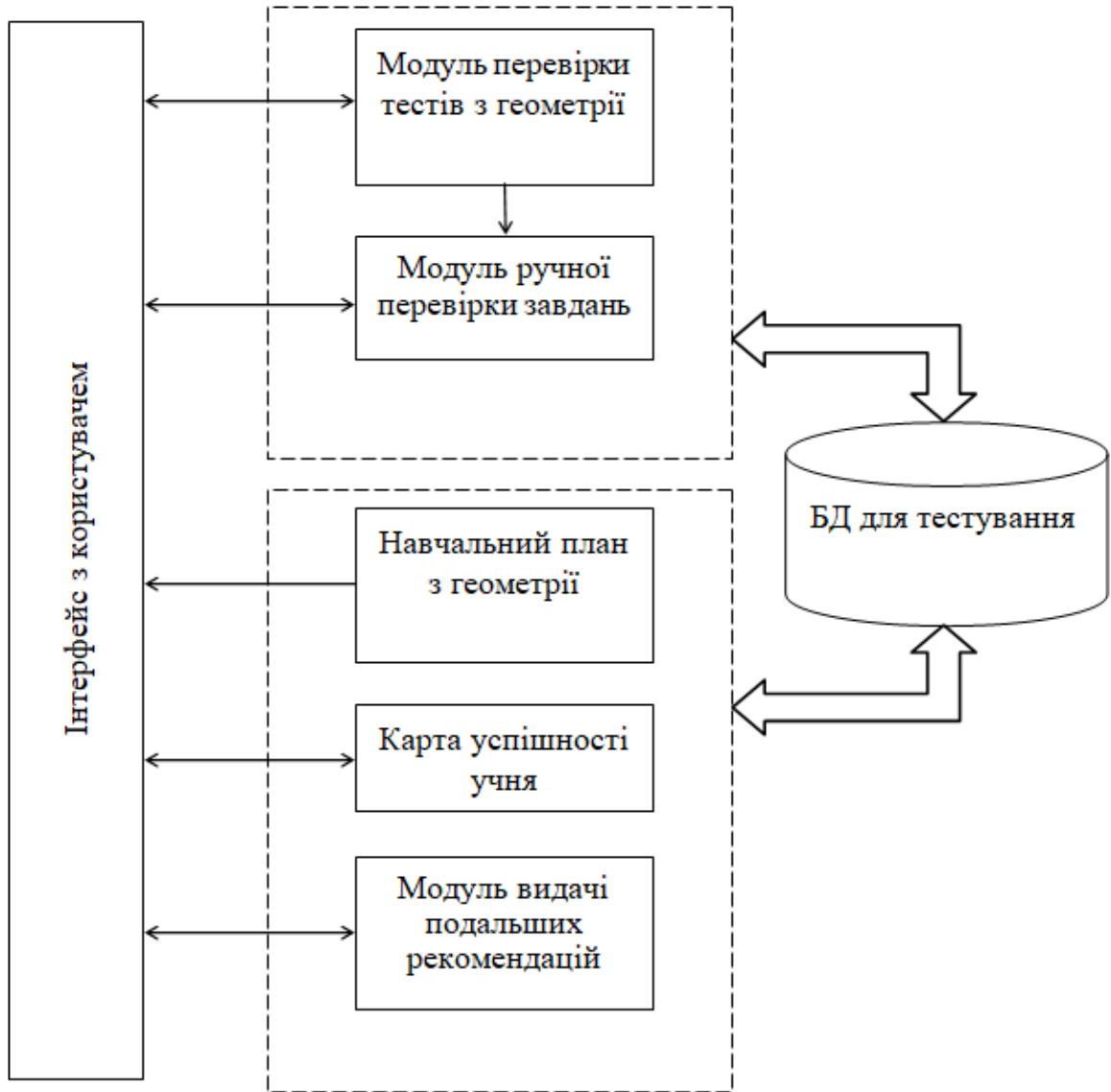


Рисунок 2.3 – Структурна схема підсистеми тестування

Навчальний план з геометрії залежить від номера класа та є основою для переліку тем та підтем, які вивчаються у системі. Цей план можна змінювати за бажанням вчителя, але розуміти, що всі обов'язкові теми повинні бути пройденими. Можна лише додавати нові теми чи більш докладно вивчати вже існуючі теми.

Під картою успішності учня розуміється набір тем та завдань, які учень опанував, з урахування дати та часу проходження завдань, а також оцінок, які отримав учень.

Модуль видачі подальших рекомендацій для послідовної чи вибіркової методики навчання передбачає поступове надання учневі доступних матеріалів та завдань. Для адаптивної методики навчання аналізується оцінка, отримана учнем за завдання, та складність цього завдання. У результаті учню може бути запропоновано повернутись по попередньої теми, повторно вивчити поточну тему, перейти з більш складних завдань на більш прості чи навпаки, чи перейти до наступної теми.

2.4 Висновки до розділу

У цьому розділі проведено розробку за стосунку з навчання геометрії.

Створено модель навчання шкільному курсу геометрії, яка містить навчальні теоретичні та практичні матеріали та методики навчання, які визначають переходи між вивчаємими учнем темами.

Створено та описано структурні схеми для підсистем навчання та тестування.

3 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАСОБУ

3.1 Варіанти використання програмного засобу

Виконаємо розробку та аналіз вимог до розроблюваного навчального програмного засобу [4, 5]. З навчальним засобом можуть працювати 2 категорії користувачів – це Вчитель та Учень. Прийняте розмежування прав доступу до засобу та типова послідовність роботи з навчальною програмою у спрощеному вигляді можна описати наступним чином:

- Вчитель – задає теми, підтеми та навчальні порції, переглядає профілі учнів, організованих у класи, задає методику проходження навчального матеріалу та тестів, переглядає та аналізує кількість та якість проходження учнями тестових завдань;

- Учень – реєструється у програмному засобі відповідно до власного класу та проходить надані навчальні матеріали та тестові завдання згідно з методикою, вказаною вчителем.

Усі дані по ходу навчання дані повинні відображатися на екрані монітора у зручній формі та зберігатися в реляційній базі даних. Проведений аналіз предметної області та загальних вимог до роботи навчального засобу дозволяє виділити таких акторів: «Користувач», від якого далі успадковуються актори «Вчитель» та «Учень».

Для опис функціональних вимог засобу надано UML діаграму варіантів використання (рис. 3.1) та описано сценарії її використання.

«Варіант використання № 1» – «Авторизація користувача»

Основний виконавець: «Неавторизований користувач».

«Передумови»: «Неавторизований користувач» має логін та пароль у навчальному засобі.

«Післяумови»: «Неавторизований користувач» увійшов у навчальний засіб як Вчитель або Учень.

«Основний успішний сценарій».

1. «Неавторизований користувач» заходить в розділ авторизації.

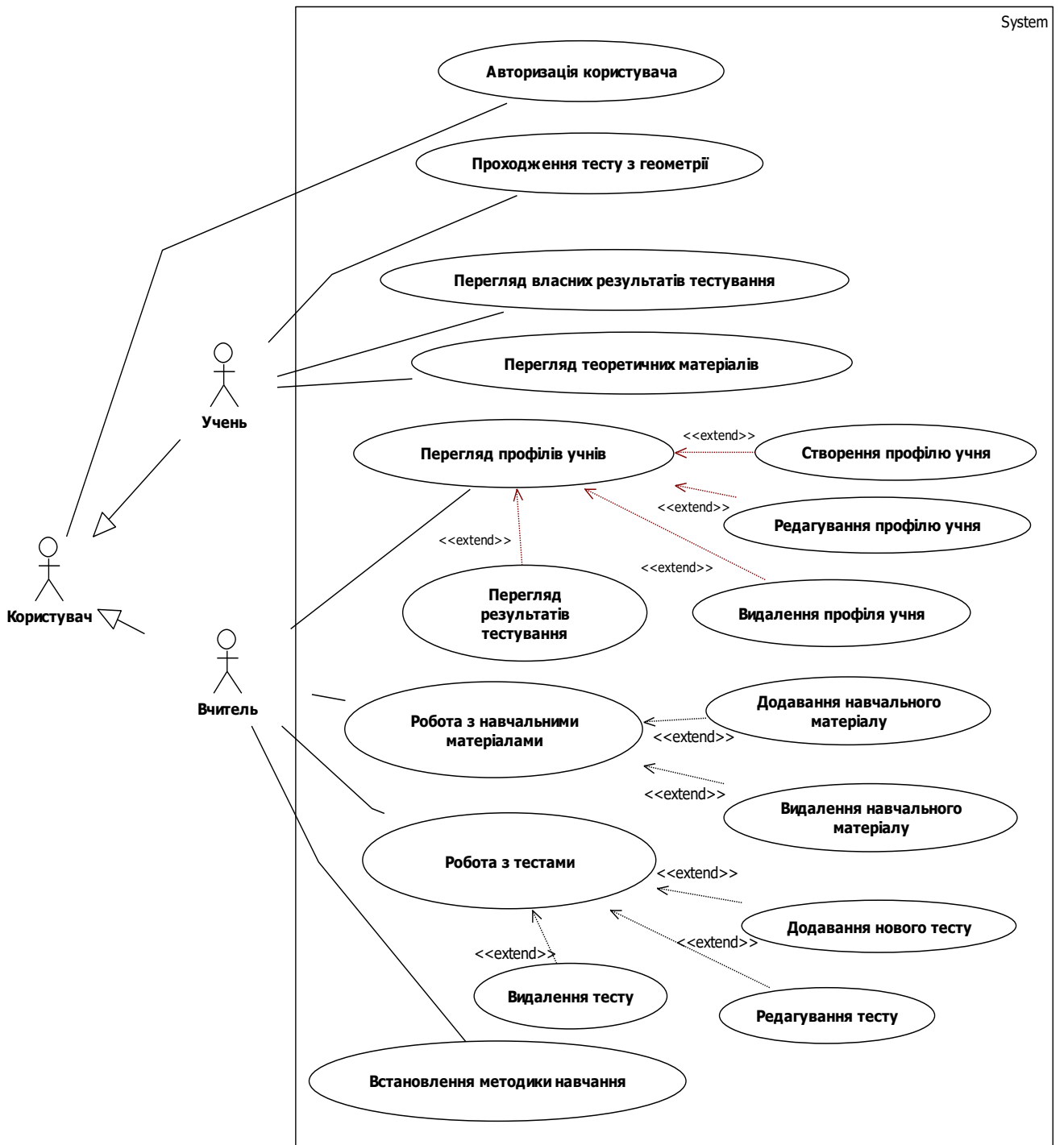


Рисунок 3.1 – Діаграма варіантів використання навчального засобу

1. Якщо «Неавторизований користувач» вчителем, навчальний засіб запитує його ім'я, електронну адресу та номер телефону. Якщо

користувач є учнем, навчальний засіб запитує його ім'я, школу, клас та електронну адресу користувача.

2. «Неавторизований користувач» заповнює дані у необхідних полях програмної форми.
3. Навчальний засіб аналізує дані та перевіряє їх коректність.
4. Навчальний засіб надає права доступу до профілю.
5. Зафіксовано успішний вхід в навчальний засіб.
6. Кінець прецеденту.

«Альтернативний сценарій ВВ № 1».

3а) Не всі програмні поля заповнені.

- 1) Навчальний засіб видає сповіщення про необхідність заповнити всі поля форми.

3б) «Неавторизований користувач» відмовляється вводити логін і пароль.

- 1) Авторизація не завершена. Перехід до п.7.

«Варіант використання № 2» – «Проходження тесту з геометрії»

«Основний виконавець»: Користувач, зареєстрований як Учень.

«Передумови»: Учень бажає пройти тест та дізнатись власні результати.

«Післяумови»: Учень дізнався результати тестування.

«Основний успішний сценарій».

1. Учень натискає кнопку для вибору теста, при цьому програмний засіб відкриває необхідну форму.
2. Навчальний засіб визначає необхідну кількість кроків N для проходження обраного тесту.
3. Крок = 0.
3. Крок = Крок + 1. Учень натискає відповідні кнопки для вибору номерів

тестових завдань.

4. Навчальний засіб відображає завдання на екрані.

5. Учень відповідає на тестові завдання за принципами «один-з-багатьох», «багато-з-багатьох», «відповідність».

6. Якщо ще залишились не пройдені завдання (Крок $<$ N), здійснюється перехід до п. 3.

7. Учень обирає команду «Обчислити результати тестування».

8. Навчальний засіб обчислює значення результатів по завданням.

9. Учень обирає команду «Завершити тест».

10. Кінець прецеденту.

«Альтернативний сценарій ВВ № 2».

1а) Учень відмовляється обирати завдання.

1) Завершення прецеденту.

5а) Учень не відповідає на задані тестові питання.

1) Завершення прецеденту без збереження змін.

«Варіант використання № 3» – «Перегляд власних результатів тестування»

«Основний виконавець»: Користувач, зареєстрований як Учень.

«Передумови»: Учень бажає отримати та переглянути результати власного тестування з геометрії.

«Післяумови»: Результати власних тестів переглянуто.

«Основний успішний сценарій».

1. Учень відкриває форму.

2. Навчальний засіб надає перелік пройдених Учнем тестів з геометрії з вказанням дати та часу завершення тестування.

3. Учень переглядає список тестів з геометрії, обирає певний тест та натискає кнопку ОК.

4. Навчальний засіб відображає результати тестування з геометрії з розподілом за темами та підтемами.

5. Учень обирає команду «Закри́ть».

6. Кінець прецеденту.

«Альтернативний сценарій ВВ № 3».

2а) Немає зв'язку з БД, неможливо завантажити список тестів з геометрії.

1) Навчальний засіб видає сповіщення. Завершення прецеденту.

3а) Учень відмовляється обирати тест з геометрії.

1) Завершення прецеденту.

4а) Немає зв'язку з БД, неможливо завантажити результати тестування.

1) Навчальний засіб видає сповіщення про помилку. Завершення прецеденту.

«Варіант використання № 4» – «Перегляд теоретичних матеріалів»

«Основний виконавець»: Користувач, зареєстрований як Учень.

«Передумови»: Учень має намір переглянути теоретичні матеріали по певній темі.

«Післяумови»: Потрібні матеріали переглянуто.

1. Учень відкриває спеціальну форму для перегляду матеріалів та обирає тему.

2. Навчальний засіб завантажує перелік існуючих навчальних відео та текстових матеріалів з геометрії. Для відеоматеріалів під вікном для демонстрації відео наведений пояснювальний опис.

3. Учень обирає матеріал з геометрії та вивчає його. Для повернення до списку доступних йому тем він натискає кнопку-«стрілку».

4. Кінець прецеденту.

«Альтернативний сценарій ВВ № 4».

1а) Учень не обирає жодну.

1) Завершення прецеденту.

2а) Немає зв'язку з реляційною БД, тому неможливо завантажити список матеріалів з геометрії.

- 1) Навчальний засіб видає сповіщення про виникнення помилки. Кінець прецеденту.

«Варіант використання № 5» - «Перегляд профілів учнів»

«Основний виконавець»: Користувач, зареєстрований як Вчитель.

«Передумови»: Для перегляду, редагування та видалення профілю він повинен бути створений у навчальному засобі. Для створення профіля учня потрібно знати його ім'я та електронну адресу.

«Післяумови»: Профілі учнів, які були створені та активовані у навчальному засобі вчителем, переглянуто.

«Основний успішний сценарій».

1. Вчитель відкриває спеціальну форму перегляду профілів учнів.
2. Навчальний засіб завантажує дані по профілям та надає їх вчителю з можливістю прокрутки цього списку.

3. Вчитель обирає одну з наступних дій:

3.1 Редагування. Для редагування даних певного профіля учня потрібно його виділити та обрати команду «Редактировать». Після цього дані обраного профіля будуть відображені у формі для редагування. По натисканню кнопки «ОК» редагування завершується. При цьому зміст реляційної бази даних оновлюється.

3.2 Видалення. Для видалення даних певного профіля учня потрібно його виділити та натиснути кнопку з зображенням «хрестика». Навчальний засіб видає сповіщення щодо видалення картки. Якщо вчитель підтверджує дії, то обраний профіль учня видаляється та при цьому зміст реляційної БД оновлюється. У разі, якщо вчительне згоден видаляти профіль учня, він залишається у навчальному засобі та зміст бази даних залишається без змін.

3.3 Додавання. Для додавання нового профіля учня вчитель у відповідній програмній формі реєструє для нового учня ім'я та електронну адресу. Далі учень проходить процедуру реєстрації з цими даними, при цьому

він додатково вказує номер чи назву школи та номер класу. Далі вчителю потрібно оновити на екрані список учнів. Після цього доданий профіль з'явиться у списку.

3.4 Перегляд. Для перегляду профілів учнів з результатами їх тестування вчитель обирає профіль учня, який його цікавить, двічі натискає мишкою по певному рядку у таблиці. Далі навчальний засіб надає спеціальну форму з усіма результатами тестування виділеного учня. Для того, щоб повернутись до загального переліку профілів учнів, вчитель повинен натиснути кнопку «Закреть».

4. Далі вчитель обирає команду «Закреть».

5. Кінець прецеденту.

«Альтернативний сценарій ВВ № 5».

2а) Немає зв'язку з реляційною БД, тому неможливо завантажити перелік профілів учнів.

1) Навчальний засіб видає сповіщення про виникнення помилки БД.
Завершення прецеденту.

3.1а) Вчитель вводить невірні дані у профіль учня.

1) Навчальний засіб видає сповіщення про виникнення помилки вводу та вимагає повторного введення даних про учня.

3.1б) Вчитель вводить дані учня у профіль та не погоджується їх зберігати.

1) Завершення процедури редагування профілю учня без змін даних у базі.

3.2а) Вчитель обирає профіль учня та потім не погоджується його видалити.

1) Завершення процедури видалення профіля без змін даних у базі.

3.2б) Відсутній доступ до бази профілів.

1) Навчальний засіб видає сповіщення про виникнення помилки бази.
Завершення процесу видалення без змін даних у базі.

3.3а) При реєстрації нового учня вчитель вводить дані, які не відповідають потрібному формату.

1) Навчальний засіб видає сповіщення про помилку формату та потребує повторного внесення даних.

3.3б) При реєстрації нового учня вчитель вводить його дані такі, що повністю співпадають з даними іншого учня, який вже є зареєстрованим у програмі.

1) Навчальний засіб видає сповіщення про виникнення помилки дублювання даних та потребує повторного внесення даних.

«Варіант використання № 6» – «Робота з навчальними матеріалами»

«Основний виконавець»: Користувач, зареєстрований як Вчитель.

«Передумови»: Вчитель хоче змінити чи замінити навчальні матеріали, завантажені у засобі.

«Післяумови»: Вчителем внесено зміни у перелік навчальних матеріалів, змінений перелік збережено у базі.

«Основний успішний сценарій».

1. Вчитель бажає обрати номер класу та для цього системі задає відповідну команду.
2. Програмний засіб надає йому перелік класів.
3. Вчитель обирає номер зі списку класів.
4. Навчальний засіб надає вчителю список завантажених у системи тем та підтем з геометрії відповідно до обраного класу.
5. Вчитель обирає тему та підтему, для якої потрібно визначити новий навчальний матеріал чи видалити непотрібний матеріал.
 - 5.1 Для того, щоб завантажити новий навчальний матеріал, вчитель натискає кнопку «Вибрати», вказує формат матеріалу, обирає файл з матеріалом (навчальну порцію) та завантажує його до навчального засобу. Якщо завантажується матеріал у відео-форматі, потрібно створити до нього короткий опис для більш зручного користування системою.
 - 5.2 Для видалення матеріалу вчитель двічі натискає мишкою по непотрібному навчальному матеріалу, програма потребує ввести

підтвердження своїх дій. Вчитель підтверджує дію, непотрібний навчальний матеріал видаляється з засобу.

6. Навчальний засіб зберігає оновлені результати у реляційній базі.
7. Вчитель обирає команду «Закри́ть».
8. Кінець прецеденту.

«Альтернативний сценарій ВВ № 6».

5.1a) Обраний для додавання файл з навчальними матеріалами пошкоджено чи обрано файл неприпустимого формату.

- 1) Навчальний засіб надає сповіщення про помилку читання файлу. Завершення роботи без будь-яких змін.

5.2a) Обрано для видалення з системи навчальний матеріал, який використовується іншим користувачем-учнем.

- 1) Видалення матеріалу відкладається до завершення користування файлом всіма користувачами. Після цього матеріал видаляється та оновлюється база навчальних матеріалів.

ба) Немає доступу до реляційної бази.

- 1) Навчальний засіб надає вчителю відповідне сповіщення.
- 2) Завершення роботи з матеріалами без змін у базі.

«Варіант використання № 7» - «Робота з тестами»

«Основний виконавець»: Користувач, зареєстрований як Вчитель.

«Передумови»: Процедури редагування та видалення доступні для тестів, завантажених у навчальний засіб. Для додавання тесту потрібно знати, до яких класу, теми та підтеми він належить.

«Післяумови»: Перелік та зміст навчальних тестів є актуальними.

«Основний успішний сценарій».

1. Вчитель бажає обрати номер класу та для цього задає системі відповідну команду.
2. Програмний засіб надає йому перелік класів.

3. Вчитель обирає номер зі списку класів.
4. Вчитель відкриває форму перегляду тестів, що належать до тем обраного класу.
5. Навчальний засіб завантажує дані по тестам та надає їх вчителю з можливістю прокрутки списку тестів.
6. Вчитель обирає одну з дій:
 - 6.1 Редагування. Для редагування певного тесту потрібно його визначити та обратикоманду «Редактировать». Це веде до відображення даних цього тесту у спеціальній програмній формі. Далі вчитель може додавати чи видаляти питання, змінювати правильні відповіді на питання, змінювати порядок відповідей, встановлювати оцінки для правильної відповіді. Для завершення редагування потрібно натиснути кнопку«ОК». Після цього реляційна база даних оновлюється.
 - 6.2 Додавання. Для того, щоб додати новий навчальний тест, по-перше, потрібно ввести його назву та короткий опис. Далі необхідно навести питання тесту та варіанти відповідей. При цьому обирається тип тесту («один-з-багатьох», «багато-з-багатьох», «відношення»), правильна відповідь та оцінка за правильну відповідь. До питань тесту при необхідності може бути доданий коментар.
 - 6.3 Видалення. Для того, щоб видалити тест, потрібно обрати його з загального списку та натиснути кнопку «хрестик». Навчальний засіб потребує підтвердити видалення навчального тесту. Якщо вчитель підтверджує операцію видалення, вона здійснюється та зміст бази оновлюється. Видалені тести не надаються у подальшому у переліку доступних тестів. Проте якщо будь-який учень успішно виконав такий тест, то результати його тестувань зберігаються у програмі. У випадку, коли вчитель не погоджується видалити навчальний тест, він залишається у базі без змін.
7. Вчитель обирає команду«Закри́ть».
8. Кінець прецеденту.

«Альтернативний сценарій ВВ № 7».

5a) Немає зв'язку з базою, тому неможливо завантажити список тестів.

- 1) Навчальний засіб видає сповіщення про виникнення помилки по тестам. Кінець прецеденту.

6.1a) Для програмного засобу відсутній доступ до тестів.

- 1) Навчальний засіб видає сповіщення про помилку бази даних. Завершення роботи без змін у базі.

6.2a) Вчителем створено тест, але у нього не додано жодного питання.

- 1) Навчальний засіб видає сповіщення про помилку створення тесту та потребує повторного введення даних.

6.3a) Вчитель визначив тест для видалення, але відмовляється підтверджувати свою дію.

- 1) Завершення роботи без змін у базі.

Варіант використання № 8» – «Встановлення методики навчання»

«Основний виконавець»: Користувач, зареєстрований як вчитель.

«Передумови»: Вчитель бажає обрати методику, згідно з якою учні будуть проходити навчання у системі.

«Післяумови»: Методику обрано.

«Основний успішний сценарій».

1. Вчитель бажає обрати методику та для цього задає системі відповідну команду.
2. Програмний засіб надає йому перелік методик: послідовне, вибіркоче чи адаптивне проходження матеріалів.
3. Вчитель обирає методику.
4. У разі, якщо обрано вибіркоче проходження матеріалів, вчителем відмічаються ті матеріали, які необхідно вивчати.
5. Програмний засіб зберігає налаштування методики.
6. Кінець прецеденту.

«Альтернативний сценарій ВВ № 8».

4а) Немає зв'язку з БД, неможливо завантажити список матеріалів з геометрії.

1) Навчальний засіб видає сповіщення. Завершення прецеденту.

5а) Немає зв'язку з БД, неможливо зберегти налаштування методики.

1) Навчальний засіб видає сповіщення про помилку бази. Завершення прецеденту.

Загальний список тем з геометрії відповідає класам, він є постійним та відповідає навчальній програмі. Він повністю покриває повний курс шкільної геометрії.

Навчальний засіб підтримує навчання з геометрії за наступними темами та підтемами [6]:

7 клас

01. Тема 1. Початкові геометричні відомості

02.Тема 2. Трикутники

03.Тема 3. Паралельні прямі

04.Тема 4. Співвідношення між сторонами та кутами трикутників

05.Повторення курсу геометрії 7-го класу

8 клас

01.Повторення

02.Тема 1. Чотирикутники

03.Тема 2. Площа

04.Тема 3. Подібні трикутники

05.Тема 4. Окружність

06.Тема 5. Вектори

07.Повторення курсу геометрії 8-го класу

9 клас

01.Повторення

02.Тема 1. Метод координат

- 03.Тема 2. Співвідношення між сторонами та кутами трикутника. Підтема 1. Синус, косинус та тангенс кута
- 04.Тема 2. Підтема 2. Співвідношення між сторонами та кутами трикутника
05. Тема 2. Підтема 3. Скалярний добуток векторів
- 06.Тема 3. Довжина кола та площа кола. Підтема 1. Правильні багатокутники
- 07.Тема 4. Рух
- 08.Підсумкове повторення курсу геометрії за 7-9 класи
10 клас
- 01.Тема 1. Аксиоми стереометрії та їх наслідки
- 02.Тема 2. Паралельність прямих та площин
- 03.Тема 3.Перпендикулярність прямих та площин Підтема 1.Перпендикулярність прямої та площини
- 04.Тема 3. Підтема 2. Перпендикуляр та похилі. Кут між прямою та площиною
05. Тема 3. Підтема 3. Двогранний кут. Перпендикулярність площин
- 06.Тема 4. Багатогранники
- 07.Тема 5. Вектори у просторі
- 08.Тема 6. Повторення курсу геометрії 10 класу
11 клас
- 01.Тема 1. Метод координат у просторі
- 02.Тема 2. Тіла обертання
- 03.Тема 3. Об'єми тіл
- 04.Повторення

3.2 Вимоги до нефункціональних характеристик засобу

За результатами аналізу даної навчального програмного засобу були виявлені і формалізовані такі нефункціональні вимоги до розроблюваної системи, що представлені нижче.

1. «Практичність» (Usability).

1.1. Навчальний засіб має бути інтуїтивно зрозумілим як учням, так і викладачам.

1.2. Система не повинна вимагати від вчителя спеціальної підготовки, попередня підготовка учня повинна бути мінімальною. Всі функції повинні бути зрозумілими.

1.3. Навчальний засіб повинен мати стандартний та простий інтерфейс користувача, що не дає підстави для подвійних тлумачень при необхідності виконання функцій.

2. «Надійність» (Reliability).

2.1. Навчальний засіб має бути доступним 90% часу днем та 80% часу вночі.

2.2. Не менш ніж 80% збоїв повинні бути усунуто протягом години, не менше ніж 90% - протягом трьох годин.

3. «Продуктивність» (Performance).

Для визначення потрібної продуктивності системи був проведений аналіз завантаженості типових навчальних аналогів.

3.1. При роботі у профілі вчителя в середньому час відповіді має становити 4-5 секунд, а максимальний час – 10 секунд.

3.2. При роботі у профілі учня в середньому час відповіді має становити 3-4 секунди, а максимальний час – 6 секунд.

3.3. Система має підтримувати не менш ніж 500 користувачів при пропускній здатності 95 транзакцій у секунду.

4. «Можливість обслуговування» (Supportability).

Програмний засіб має бути не складно модифікованим для розширення функціоналу та модернізації існуючих функцій.

5. «Обмеження» (Constraints).

Програмний засіб є веб-застосуванням, тому він може відкриватися у будь-якому браузері як зі стаціонарного комп'ютера, так і з мобільного пристрою.

6. «Витрати пам'яті».

База даних системи не повинна перевищувати 1 Тб. Швидкість звертання до даних – 0,5 Мбіт/с, доступність бази повинна бути 24/7.

7. Архітектурні стандарти.

Потрібна підтримка трирівневої архітектури.

8. Збереження та відновлення.

Час на резервне копіювання даних не повинен перевищувати 15 с.

3.3 Висновки до розділу

У даному розділі кваліфікаційній роботі визначено основні специфікації до програмного навчального засобу.

Створено Use Case diagram та описано основні та альтернативні варіанти використання. Усього визначено 8 варіантів використання навчального продукту, деякі з них мають extend-розширення.

Також створено вимоги до основних нефункціональних характеристик навчального засобу.

4 ПРОЕКТУВАННЯ ЗАСОБУ З НАВЧАННЯ ГЕОМЕТРІЇ

4.1 Проектування архітектури засобу

Розроблюваний засіб є веб-застосуванням. Для роботи застосування користувач (учень чи викладач) використовує свій браузер (перелік браузерів не обмежується) та звертається до вебсерверу. В свою чергу вебсервер відповідає за бізнес-логіку системи, яка потребує для зберігання даних реляційну базу даних. Загальний вигляд описаної структури наведений на рис. 4.1.

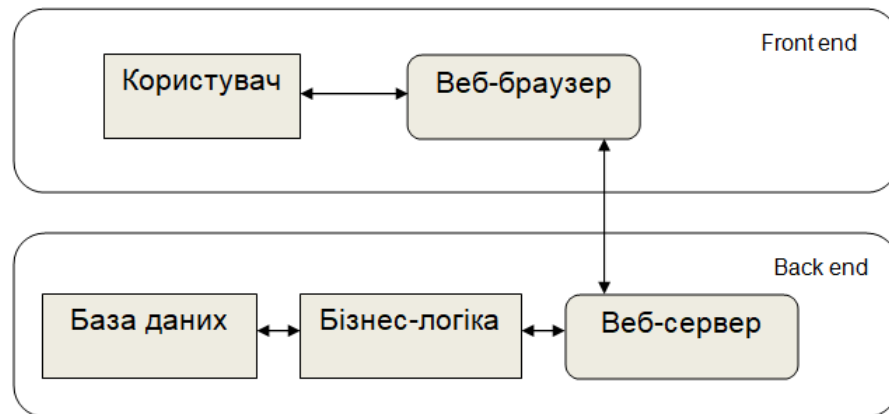


Рисунок 4.1 – Загальний вигляд структури веб-засобу

Наведемо характеристику веб-засобу:

- взаємодіє зі своїми користувачами (вчителі та учні) за допомогою інтерфейса;
- організовує формування вчителем навчальних ресурсів з геометрії;
- дозволяє вчителю створювати текстові, графічні та відеоматеріали та завантажувати їх відповідно до обраного класу та теми;
- дозволяє вчителю створювати автоматичні тести;
- дозволяє учні опановувати теоретичні матеріали з геометрії;
- дозволяє учню тестувати свої знання;
- зберігає результати тестування у базі даних;

– організує управління навчальним онлайн процесом згідно з обраною вчителем методикою керування та оцінкою учня.

Для проектування засобу обрана архітектура MVC («Model-View-Controller»). Використання цієї архітектури дозволяє відокремити дані від засобів їх використання – представлення та обробки. Це дозволяє підвищити гнучкість архітектури та налаштовувати різні програмні інтерфейси для різних категорій користувачів.

Робота навчального засобу з використанням діаграми станів та переходів представлена на рис. 4.2. Схема містить один вхід у систему, два коректні виходи та один помилковий.

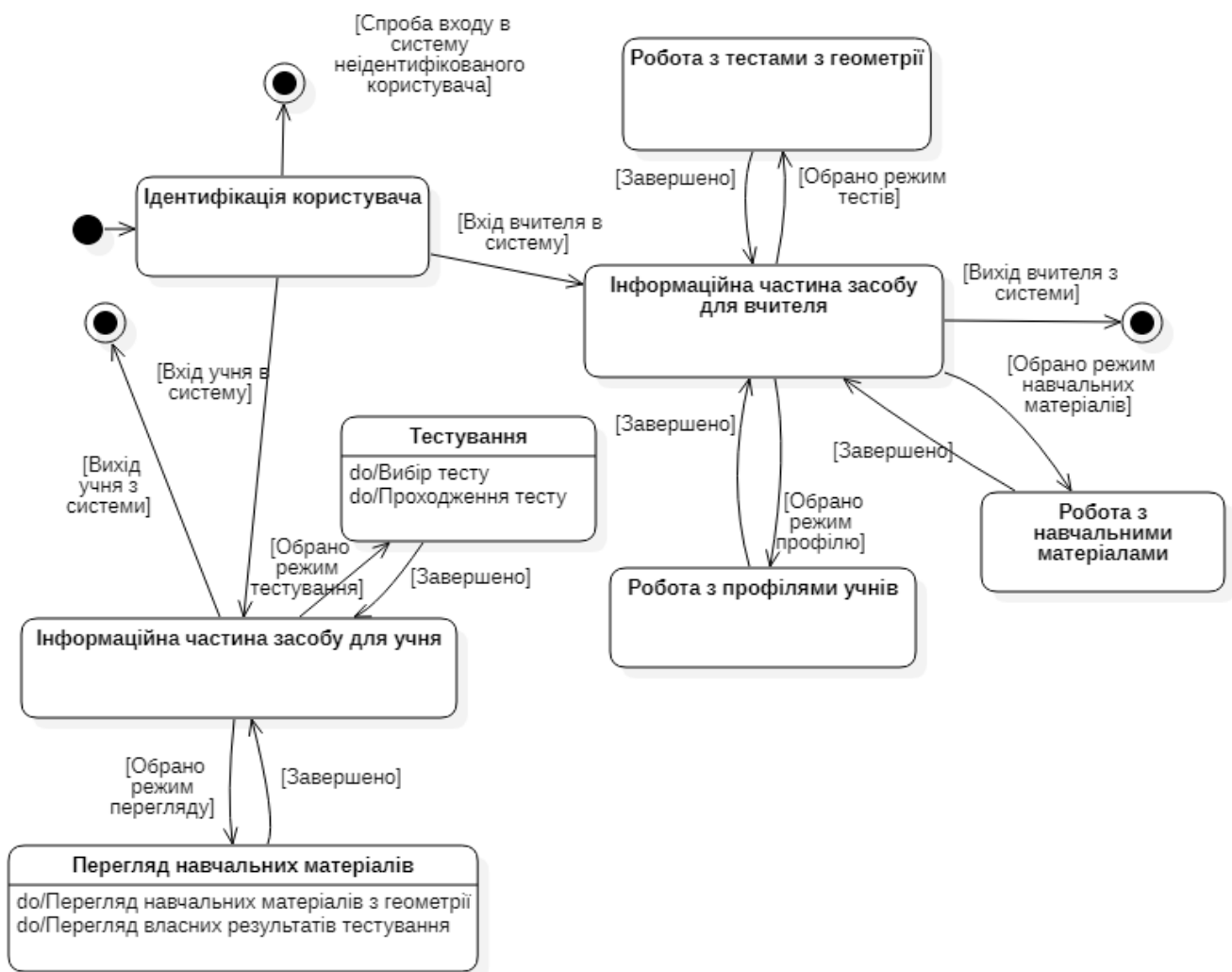


Рисунок 4.2 – Діаграма станів навчального засобу

Згідно з представленою діаграмою станів та переходів, по-перше, система намагається ідентифікувати користувача. Цей користувач може бути ідентифікованим як вчительабоучень. При успішній ідентифікації вчителю та учню надаються певні інформаційні частини засобу. Якщо трапились помилки при спробі ідентифікації, то користувач отримує повідомлення та не може увійти у систему («Спроба входу в систему неідентифікованого користувача»).

При користуванні системою вчителем йому доступна інформаційна частина, що надає можливість переходів з головного стану до станів «Робота з тестами з геометрії», «Робота з профілями учнів», «Робота з навчальними матеріалами», а також повернення з цих станів до головного. За бажанням вчитель виходить з системи зі збереженням усіх введених даних та змін щодо навчального процесу та даних учнів у реляційній базі.

При користуванні засобом учнем, що хоче вивчати геометрію, він отримує інформаційну частину з можливістю переходу до станів «Тестування» та «Перегляд навчальних матеріалів». Аналогічно вчителю,учень може повертатись у головний стан системи по закінченню навчання відповідно до обраного класу або ж тестування. За бажанням учень виходить з системи, при цьому інформація про всі пройдені тести зберігається у його карті успішності.

4.2 Основні діаграми роботи

Для вибору наступного теста система контролю процесу навчання аналізує значення наступних параметрів:

1) оцінка, яку отримав учень у навчальній системі за вирішення поточного завдання з геометрії. Ця оцінка виставляється зазвичай за п'ятибальною шкалою від «1» до «5». Якщо вчитель задає іншу шкалу, то виконується перерахунок оцінки саме у п'ятибальну шкалу. Отже, це дозволяє перевести управління процесом відповідно до одного з режимів: «Режим 1», ..., «Режим 5»;

2) для вказаного класу порядковий номер поточної теми $T(i)$ з геометрії, що містяться у навчальній системі. Це необхідно для визначення завдання з тієї ж

теми, або переходу на наступну тему, або переходу на попередню тему якщо вона існує;

3) величина категорії складності поточного вирішеного завдання: вчителем можуть бути задані категорії С1 – просте завдання, С2 – середнє чи С3 – складне завдання з геометрії.

Тепер розглянемо детальніше існуючі у навчальному засобі режими керування. З переліку режимів «Режим 4» відповідає стандартному режиму управління, а «Режим 1», «Режим 2», «Режим 3» та «Режим 5» – це адаптація процесу навчання до успішності певного учня. Основним завданням системи керування навчанням є надання рекомендації по наступному тесту. Правильний підбір тесту дозволяє підвищити ефективність та прискорити процес вивчення курсу геометрії.

«Режим 1» відповідає критично низькому рівню учня. Це відповідає ситуації нерозуміння викладеного матеріалу, що може бути наслідком як незнання теорії, так і неможливості її застосування до вирішення практичних тестів. У навчальному засобі учбовий матеріал систематизований у виді дерева ієрархії. Для опанування знань з будь-якої теми учень повинен обрати її та використовувати завантажені у ній матеріали.

У навчальному засобі в «Режимі 1» спочатку визначається поточна тема $T(i)$. Після цього з переліку тем обирається попередня тема - $T(i-1)$. Якщо тема $T(i)$ є першою, то керування переводить засіб у «Режиму 2». Якщо ж засіб визначає відсутність попередньої теми, а при цьому тема $T(i)$ не помічена як перша, то це є помилкою. Ця ситуація відповідає порушенню цілісності організації тем.

У разі, якщо у засобі попередня тема $T(i-1)$ є в наявності, то надається перелік нерозв'язаних учнем тестів з категорії С3. При наявності таких тестів засіб обирає один з них таким чином, щоб функція управління мала мінімальне значення. Це відповідає поступовому зростанню складності тестів. Тобто величина функції управління визначається складністю теста, та здійснюється вибір найпростішого тесту з категорії складних.

У разі, якщо тест з категорії С3 є відсутнім, то засіб намагається обрати нерозв'язаний учнем тест спочатку з категорії С2, а якщо не вийшло – то з категорії С1. Та у цьому випадку береться тест з максимальною функцією управління. Це відповідає вибору найскладнішого завдання по-перше з категорії «Середнє», а по-друге з категорії «Просте». У разі невдачі учень не отримує жодного тесту для подальшого навчання. Учень може власноруч повертатись до вже пройдених попередніх тем та підтем та при наявності нерозв'язаних тестів виконувати саме їх.

На рис. 4.3 показано діаграму діяльності «Режиму 1» роботи засобу.

Коли навчальний засіб працює в «Режимі 2», то обов'язково визначається як поточна тема $T(i)$, так і категорія складності для поточного розв'язаного тесту C_j . Далі засіб визначає набір нерозв'язаних учнем тестів, що належать до категорії складності С1. Якщо такі тести є, то серед них обирається такий тест, що має мінімальну функцію керування, це відповідає найпростішому тесту будь-якого типу з геометрії.

Треба взяти до уваги, значення функції керування можуть не бути унікальними, при цьому декілька тестів можуть мати одне і те саме значення функцій управління. У такому випадку можна обирати будь-який з тестів. При пошуку тесту використана умова

```
if (CTRLfunc(n)<min_value) { min_value = CTRLfunc(n);testNumber = n; }
```

що надає відповідний тест. Це означає, що якщо учнем отримана оцінка «2», тоді він отримує найпростіший тест будь-якого типу з поточної для нього теми з категорії С1.

У тому випадку, коли нерозв'язаних тестів у категорії С1 немає, якщо $j=3$ - учень розв'язував складний тест, тоді йому надається тест з категорії середнього рівня складності С2, що має мінімальну функцію управління, тобто є найпростішим з категорії С2. Зрозуміло, що при відсутності таких тестів учень не зможе отримати рекомендації.

Діаграму діяльності «Режиму 2» представлено на рис. 4.4.

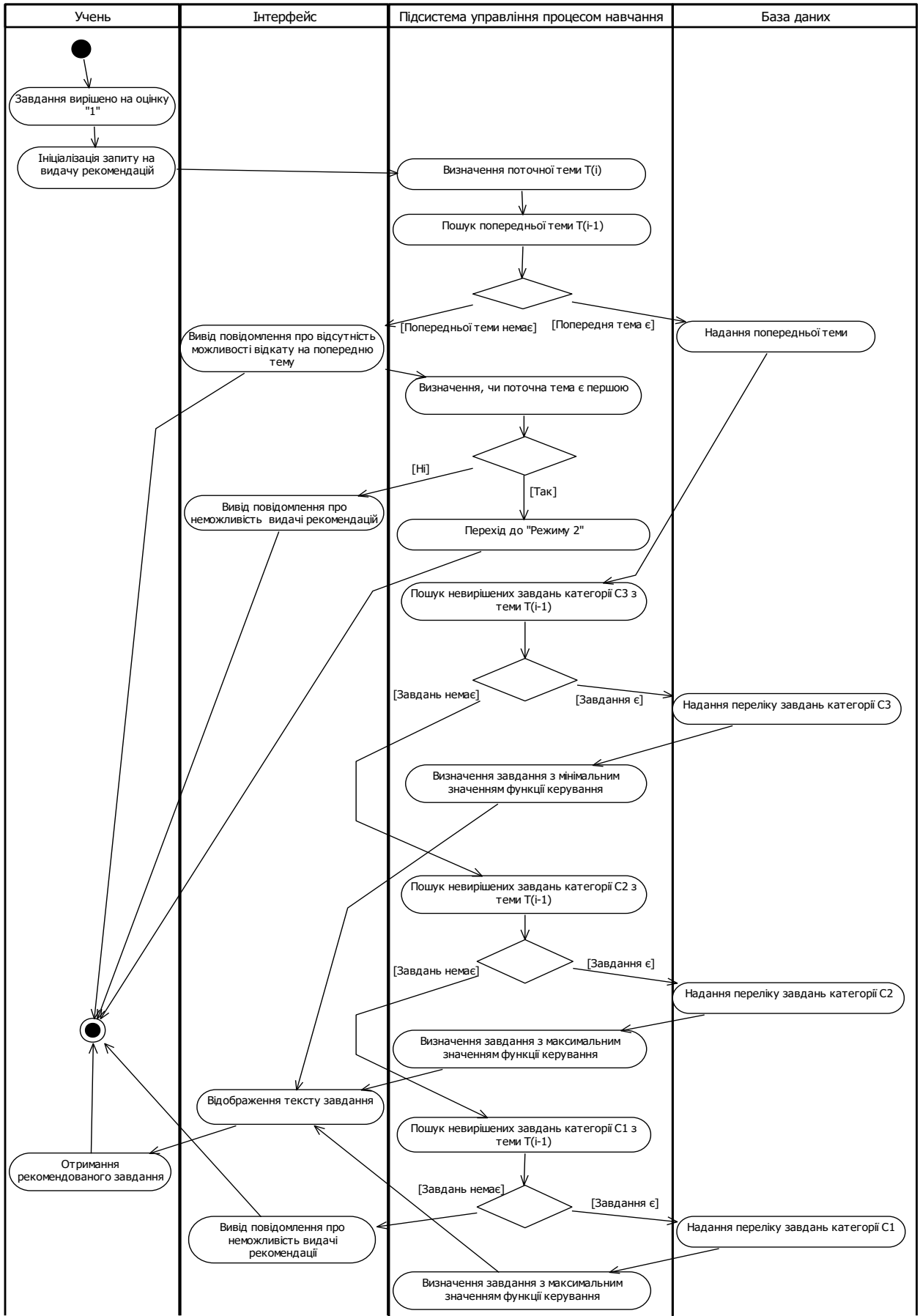


Рисунок 4.3 – Діаграма діяльності при критичному рівні відповіді (режим 1)

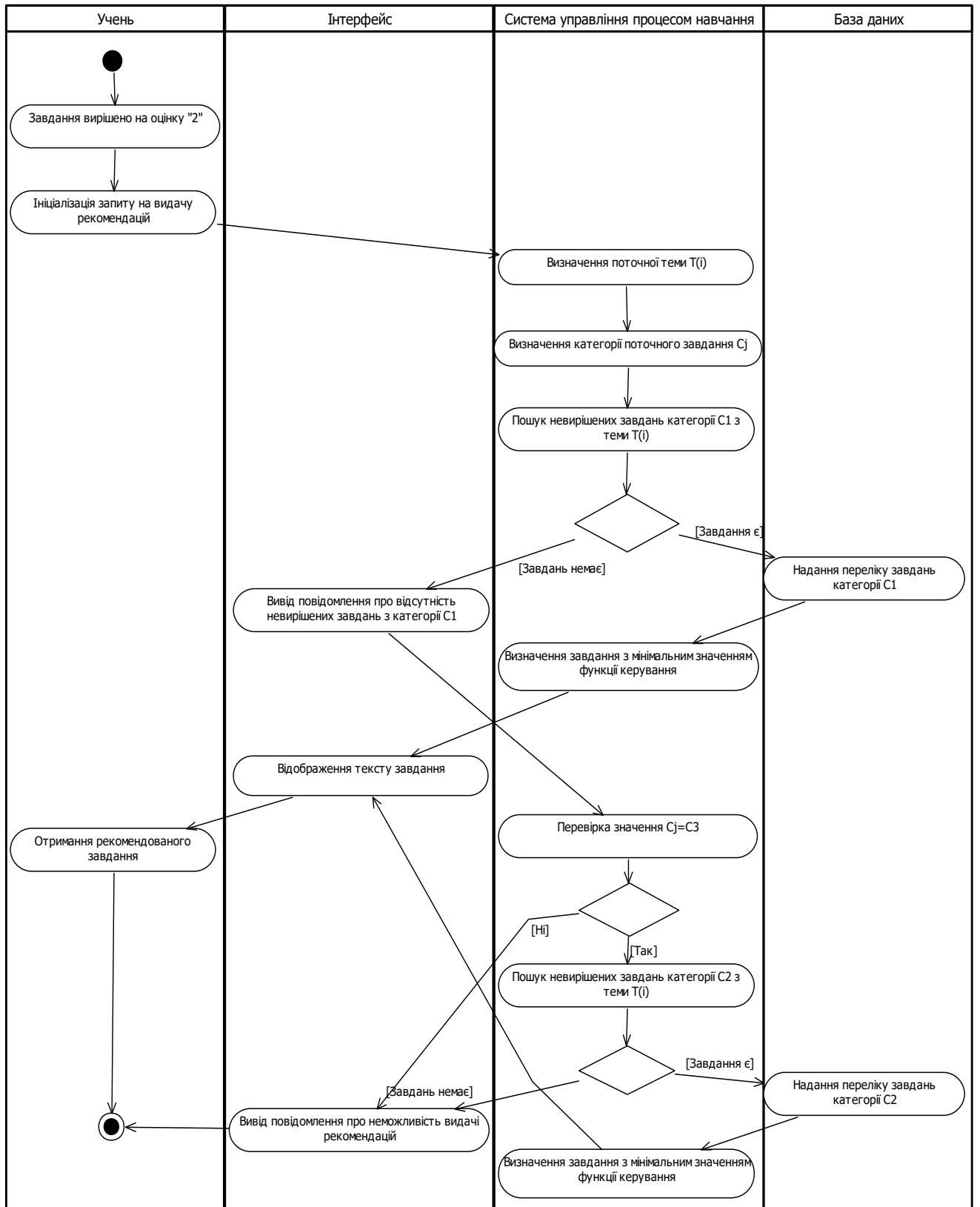


Рисунок 4.4 – Діаграма діяльності при незадовільному рівні відповіді (режим 2)

При функціонуванні засобу у «Режимі 3» теж визначається $T(i)$ - поточна тема - та C_j - поточна категорія складності. Якщо є нероз'язані тести з вказаних теми та категорії, то засіб визначає їх перелік. При цьому береться тест з мінімальною функцією управління. Це відповідає вибору найпростішого тесту з усіх тестів поточної теми й рівня складності.

На рис. 4.5 наведено діаграму діяльності для третього режиму роботи засобу.

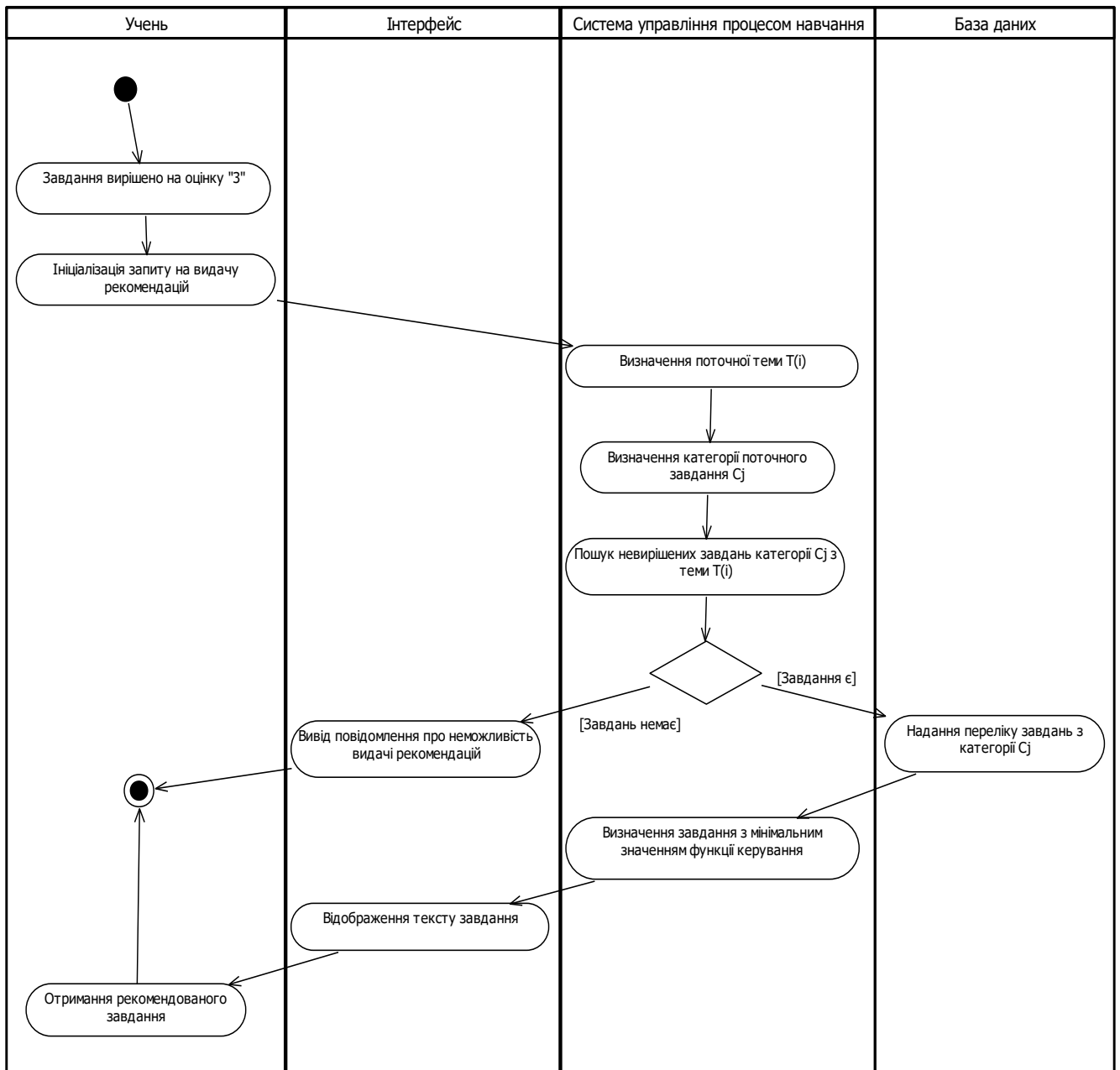


Рисунок 4.5 – Діаграма діяльності при середньому рівні відповіді (режим 3)

Відповідно до адаптивної методики, «Режим 4» є таким стандартним режимом навчання, що відстежує позитивний прогрес навчального рівня учня. При цьому учень не повертається до вже пройдених тем.

Розглянемо особливості цього режиму, що відрізняють його від інших.

Після визначення $T(i)$ та C_j перевіряється наявність нерозв'язаних тестів з C_j . Якщо такі тести відсутні та $j < 3$, то значення j інкрементується, тобто виконується обчислення $j = j + 1$.

Засіб отримує нову категорію, у якій обирається тест з мінімальною функцією управління.

У разі, якщо у C_j маємо $j = 3$, то тоді переходимо до слідуючої теми та беремо C_1 за категорію складності. У C_1 обирається тест з мінімальною функцією управління.

На рис. 4.6 формалізовано роботу засобу у «Режимі 4» з використанням діаграми діяльності.

Нарешті, розглянемо особливості роботи засобу у «Режимі 5». Традиційно навчальним засобом визначаються $T(i)$ і C_j . Якщо учень розв'язав не всі тести з $T(i)$ та C_j , то йому надається список таких тестів. При наявності декількох підходящих тестів з них обирається найскладніше, що має найбільшу функцію управління.

У разі, коли найбільш складний тест з поточної категорії вже є розв'язаним, то обирається тест зі слідуючої за складності категорії з урахуванням найбільшого значення для функції управління.

У випадку, коли всі категорії складності з поточної теми є пройденими, засіб обирає наступну тему з геометрії та встановлює «просту» категорію складності - C_1 . З C_1 обирається тест з максимальною функцією управління. У разі відсутності наступної теми, якщо поточна тема не вважається останньою, учень отримує сповіщення про помилку.

У разі, коли поточна тема вважається останньою, учень отримує сповіщення про те, що всі теми вивчено та навчання завершено.

На рис. 4.7 показано роботу засобу у «Режимі 5».

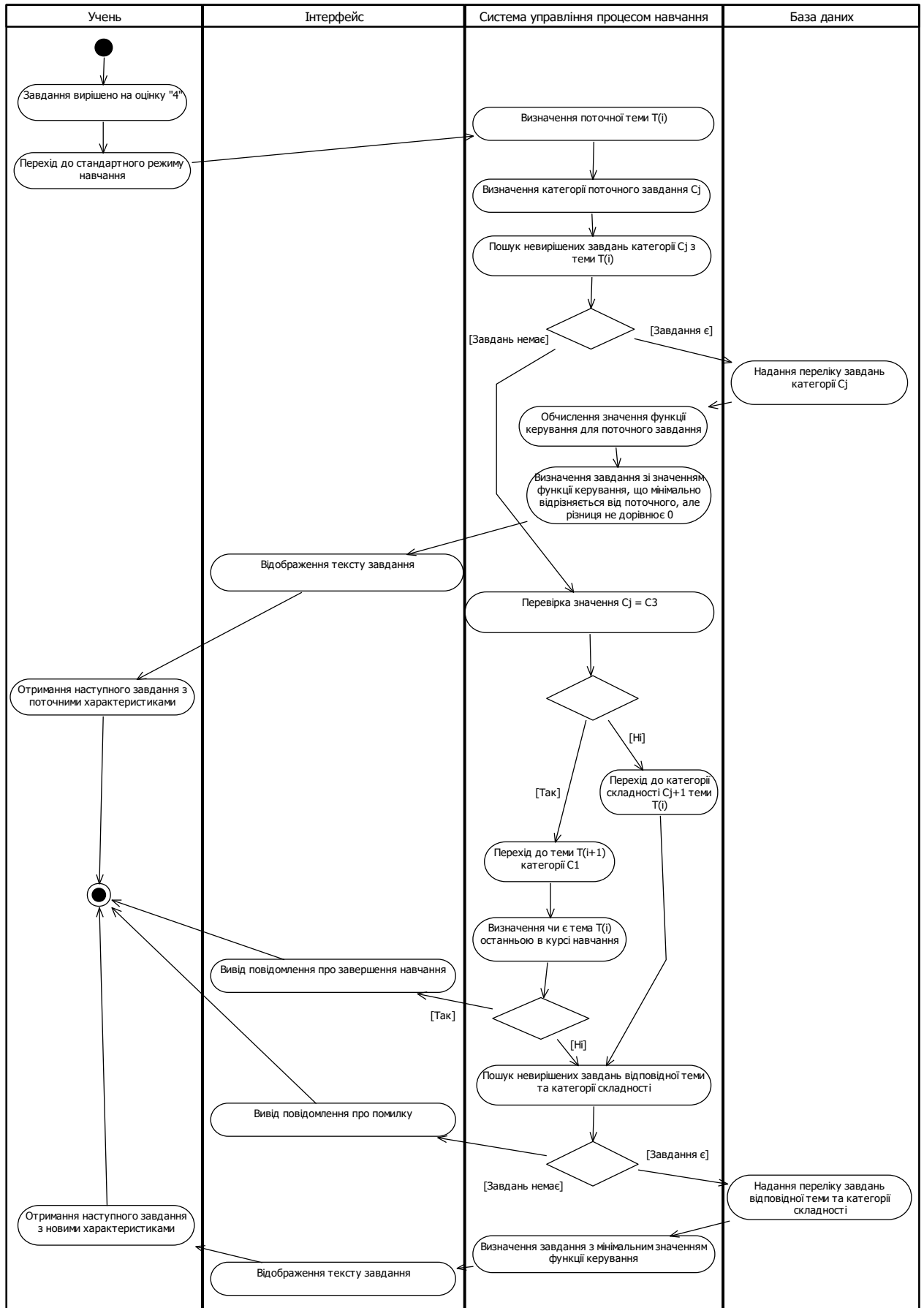


Рисунок 4.6 – Діаграма діяльності при достатньому рівні відповіді (режим 4)

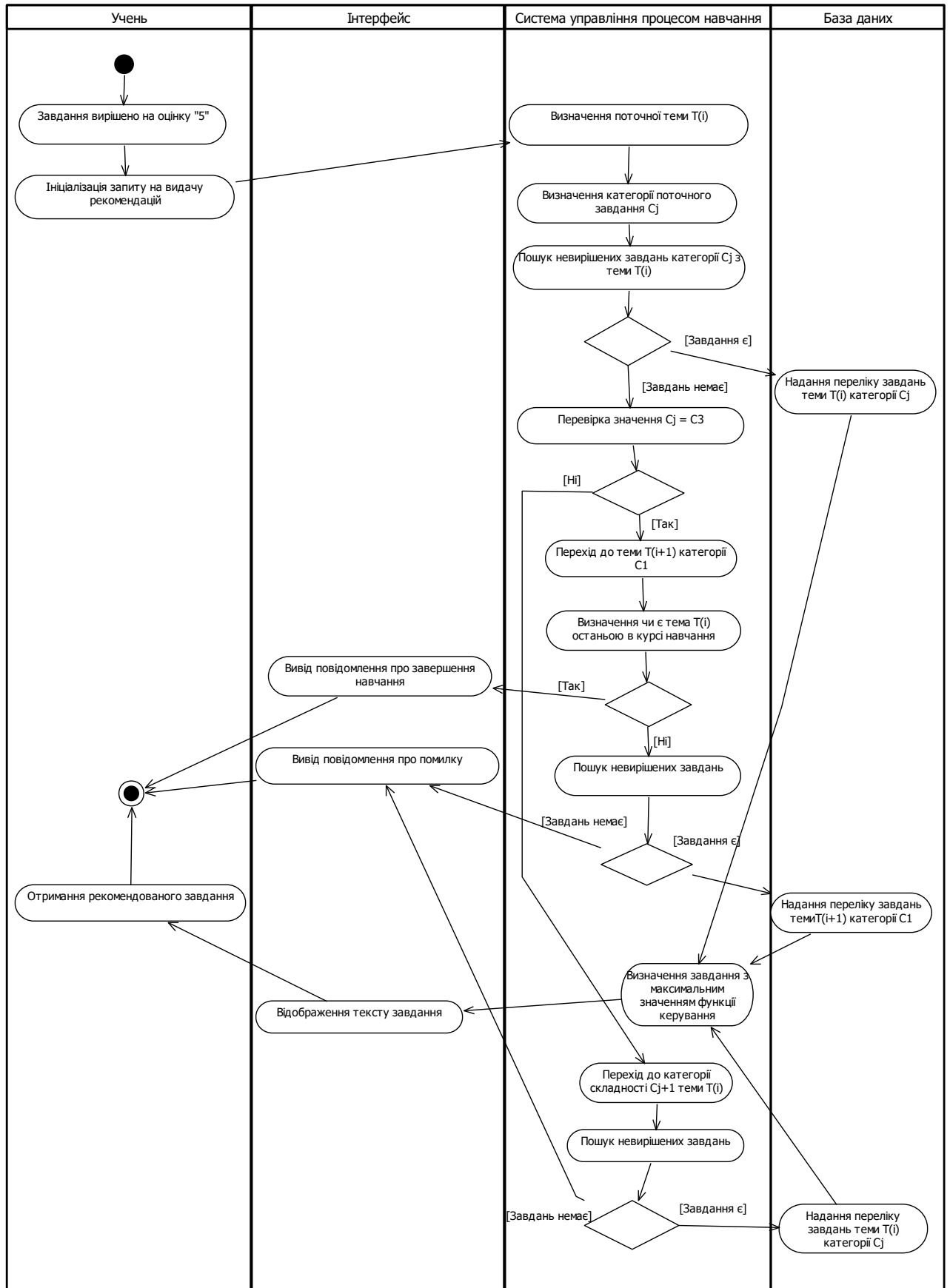


Рисунок 4.7 – Діаграма діяльності при граничному рівні відповіді (режим 5)

4.3 Проектування реляційної бази даних

Виділемо та спроектуємо окремі смислові частини БД [7, 8], що у сукупності потрібні для роботи навчального засобу:

- формування переліку уроків;
- наповнення уроків навчальними матеріалами;
- наповнення уроків тестами;
- організація навчання та тестування учнів.

На рис. 4.8 показана частина схеми БД для формування переліку уроків. Для таблиць бази виконана нормалізація відношень, усі між табличні зв'язки типу «один до багатьох». На схемі присутні такі таблиці:

LearnPlan – навчальний план, який взагалі є стандартним, але може відрізнитись для шкіл з поглибленим вивченням математики; план має порядковий номер `idLearnPlan` (первинний ключ), назву `name` та опис `description`;

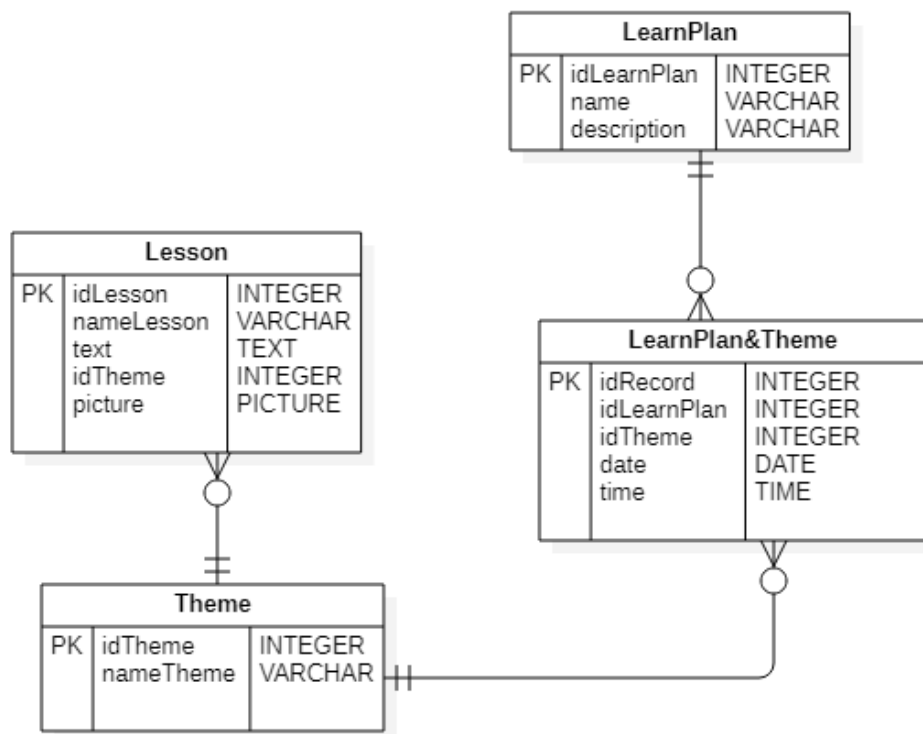


Рисунок 4.8 – Частина схеми БД для формування переліку уроків

Theme – теми з геометрії, що вивчаються відповідно плану; idTheme – номер теми, nameTheme – назва теми;

LearnPlan&Theme – допоміжна таблиця, яка введена для нормалізації відношень (може бути багато навчальних планів, в кожному – кілька тем, та тема може входити у різні плани з дисципліни геометрія); idRecord – номер запису у таблиці, idLearnPlan – ідентифікатор навчального плану, idTheme – ідентифікатор теми, date та time – дата та час додавання теми у план;

Lesson – теоретичні уроки, що входять до складу певних тем; idLesson – ідентифікатор уроку, nameLesson – назва уроку, text – текстове надання теоретичного матеріалу або гіперпосилання на відео урок, idTheme – ідентифікатор теми, до якої належить урок, picture – зображення (при необхідності подання схеми чи рисунку в уроці).

На рис. 4.9 створена частина БД для наповнення уроків навчальними матеріалами.

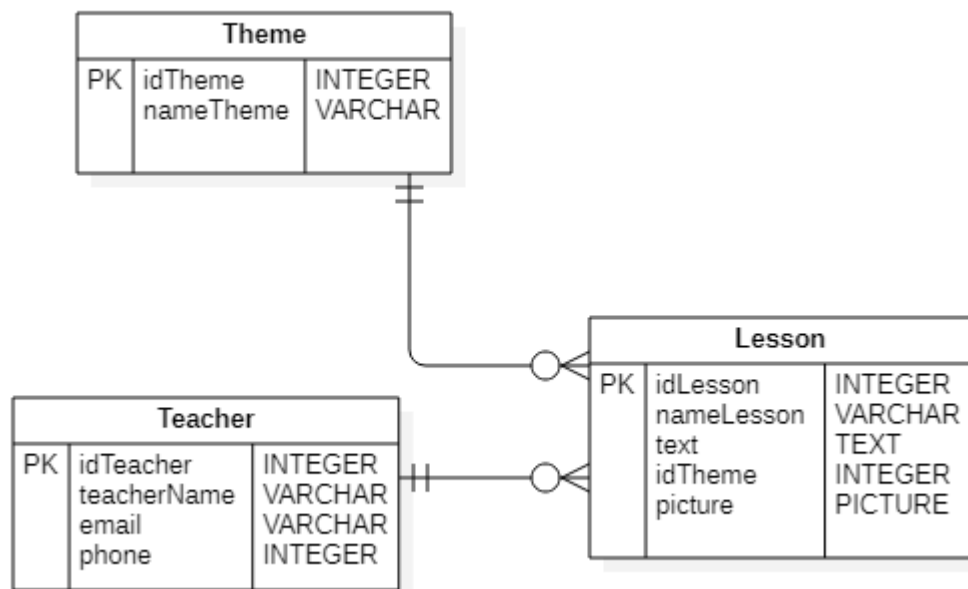


Рисунок 4.9 – Частина схеми БД для наповнення уроків навчальними матеріалами

На схемі, крім Lesson та Theme, присутня таблиця Teacher, яка містить інформацію про вчителів, які організують навчання за допомогою цього засобу.

Вчитель має номер `idTeacher`, ім'я `teacherName`, електронну адресу `email` та номер телефону `phone`.

Наведемо фрагмент БД для наповнення уроків тестами (рис. 4.10).

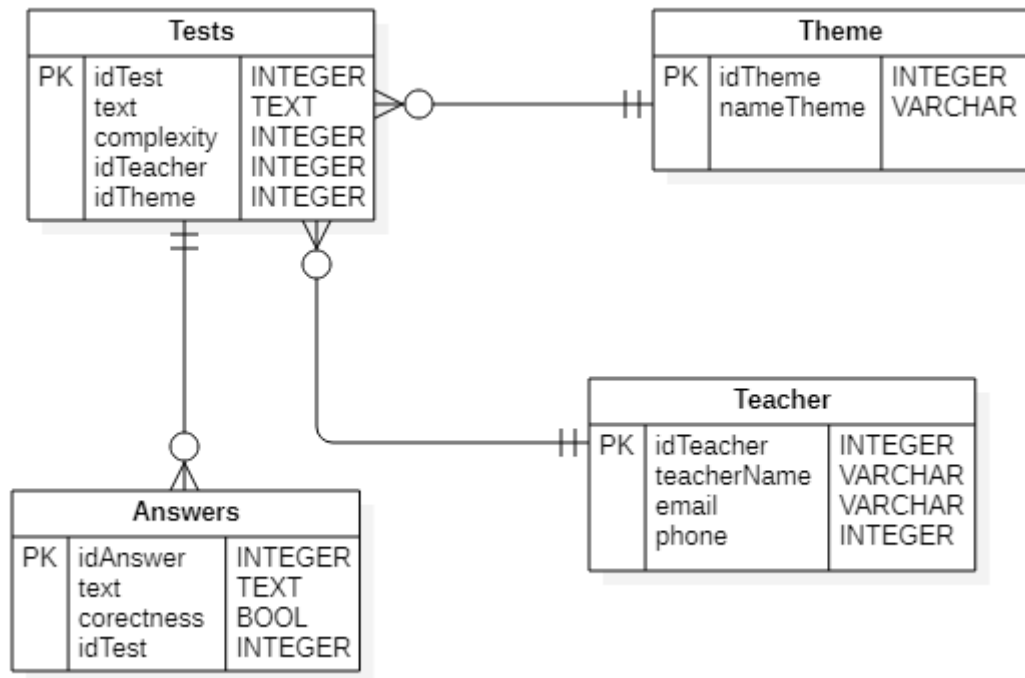


Рисунок 4.10 – Частина схеми БД для наповнення уроків тестами

Таблиця `Tests` містить перевіірочні тести, кожен тест належить до певної теми: `idTest` – номер тесту, `text` – завдання тесту, `complexity` – складність тесту (1 – простий, 2 – середній, 3 – складний), `idTeacher` – ідентифікатор вчителя, який створив тест, `idTheme` – ідентифікатор теми, до якої належить тест.

`Answers` – таблиця з відповідями на тести: `idAnswer` – ідентифікатор варіанту відповіді, `text` – текст варіанту відповіді, `correctness` – чи є відповідь правильною (0 – ні, 1 - так), `idTest` – номер тесту, до якого відноситься відповідь.

Наприкінці, фрагмент схеми для організації навчання та тестування учнів наведений на рис. 4.11.

Таблиця `Pupil` містить інформацію про учнів: `idUser` – ідентифікатор учня, `username` – його ім'я, `email` – електронна адреса учня, `class` та `school` – його клас та школа.

Таблиця TestSession є допоміжною та використовується для нормалізації, тому що один учень може проходити багато тестів, та кожен тест може бути розв’язуваним багатьма учнями: idUser – ідентифікатор учня, idTest – номер теста, datetime – дата та час проходження тесту.

Таблиця Pupil&Theme – аналогічно попередньому випадку, дозволяє нормалізувати відношення «багато до багатьох» між таблицями Pupil та Theme.

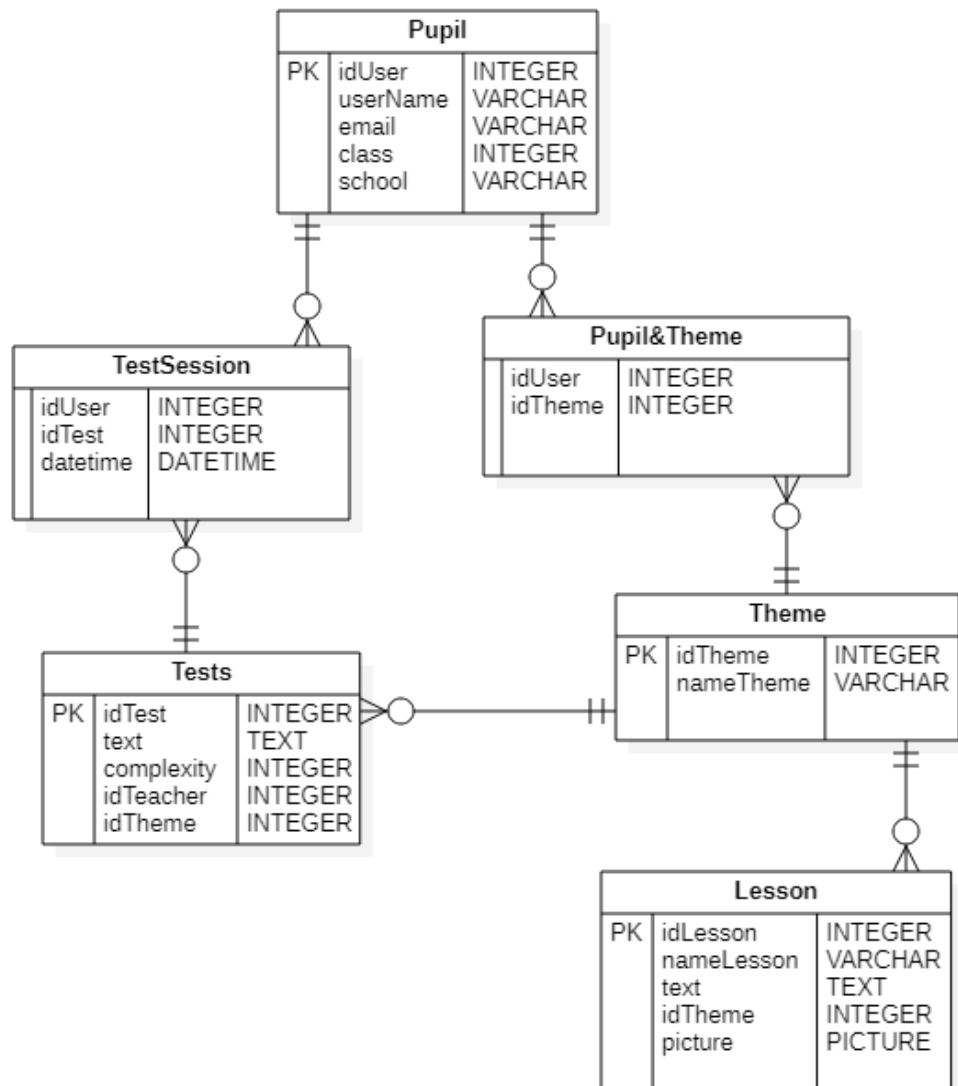


Рисунок 4.11 – Частина схеми БД для організації навчання та тестування

Якщо зібрати всі окремі частини, отримуємо повну схему реляційної бази даних навчального засобу (рис. 4.12).

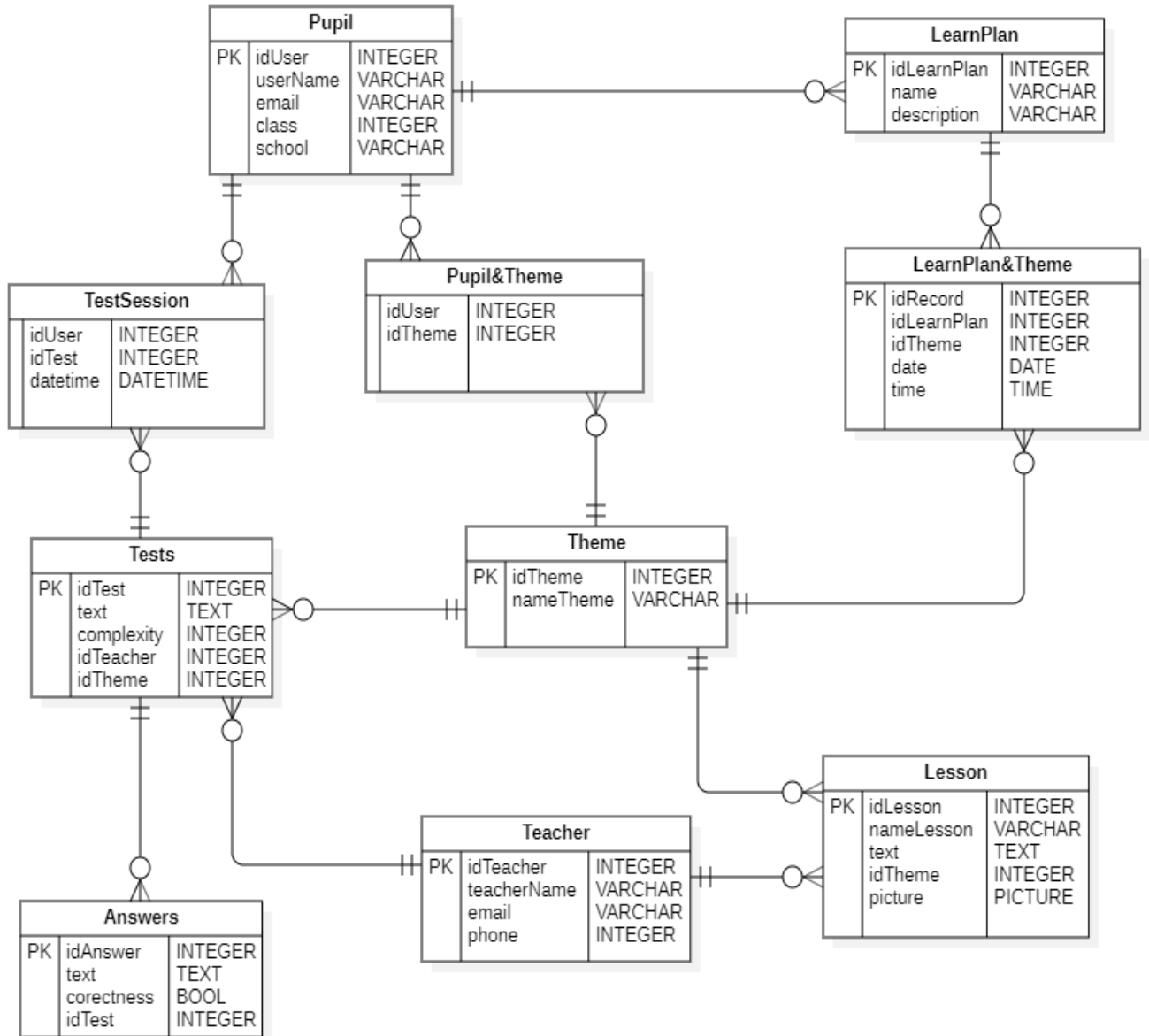
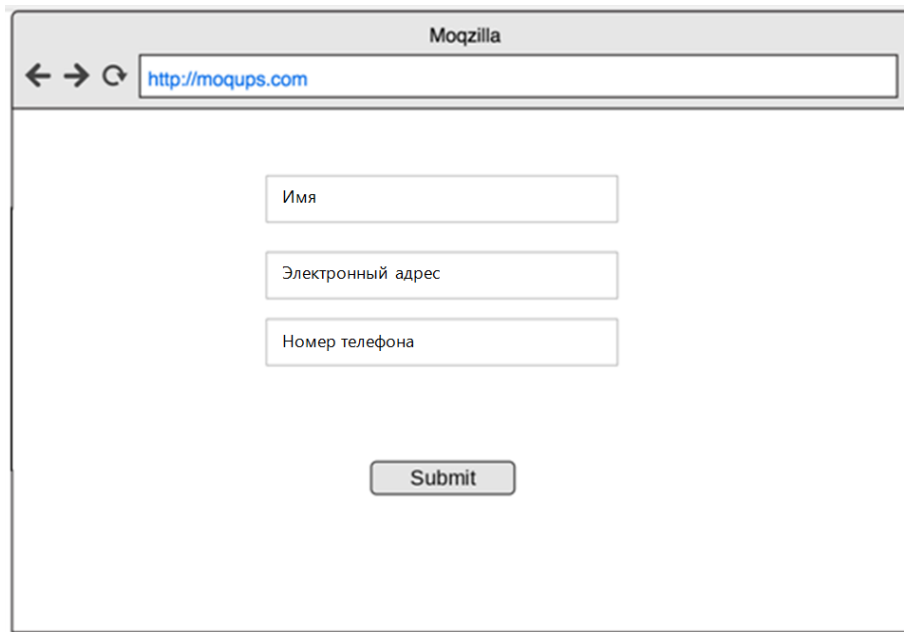


Рисунок 4.12 – Схема реляційної БД

4.4 Розробка прототипів інтерфейсу

Інтерфейс навчального засобу для вчителя та учня повинен відрізнитись. Перед початком програмування інтерфейсу виконано його проектування зі створенням прототипів. У роботі наведені та описані основні прототипи інтерфейсу.

На рис. 4.13 наведено прототип для вікна реєстрації вчителя.




The screenshot shows a Mozilla browser window with the address bar containing <http://moqups.com>. The main content area displays a registration form with three input fields: "Имя" (Name), "Электронный адрес" (Email address), and "Номер телефона" (Phone number). Below these fields is a "Submit" button.

Рисунок 4.13 – Прототип вікна реєстрації вчителя

Для реєстрації учня, на відміну від вчителя, потрібно задати номер свого класу та номер чи назву школи. Прототип для вікна реєстрації учня знаходиться на рис. 4.14.

У власному профілі учень може переглядати власні результати тестування (рис. 4.15). Вчитель може їх також переглянути, якщо зайде на профіль учня.



The screenshot shows a Mozilla browser window with the address bar containing <http://moqups.com>. The main content area displays a registration form with four input fields: "Имя" (Name), "Электронный адрес" (Email address), "Школа" (School), and "Класс" (Class). Below these fields is a "Submit" button.

Рисунок 4.14 – Прототип вікна реєстрації учня

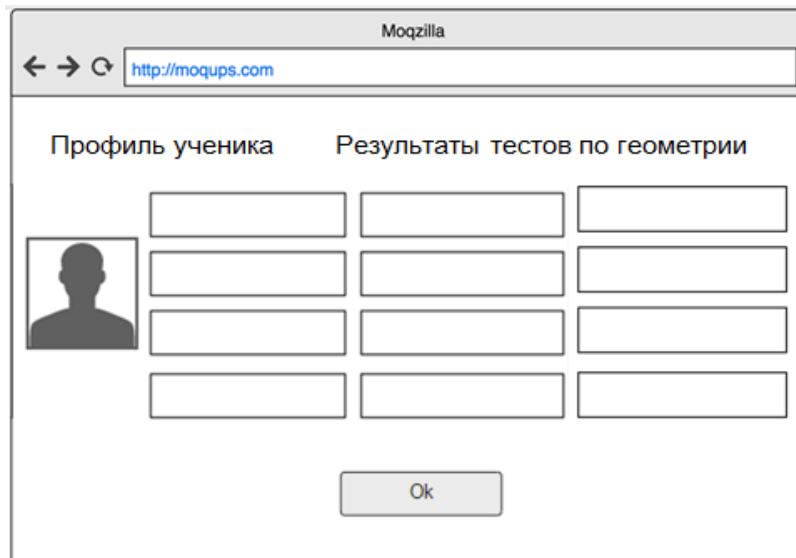


Рисунок 4.15 – Прототип вікна перегляду результатів тестування з геометрії

На рис. 4.16 показано прототип вікна для перегляду учнем теоретичних навчальних матеріалів. У стовпчику зліва наведено назви текстових файлів, зображень та відеоресурсів, які може обрати учень. Для відеофайлів повинен бути присутнім короткий текстовий опис.

Прототип форми, у якій вчитель може завантажувати такі навчальні матеріали у програму, показаний на рис. 4.17.

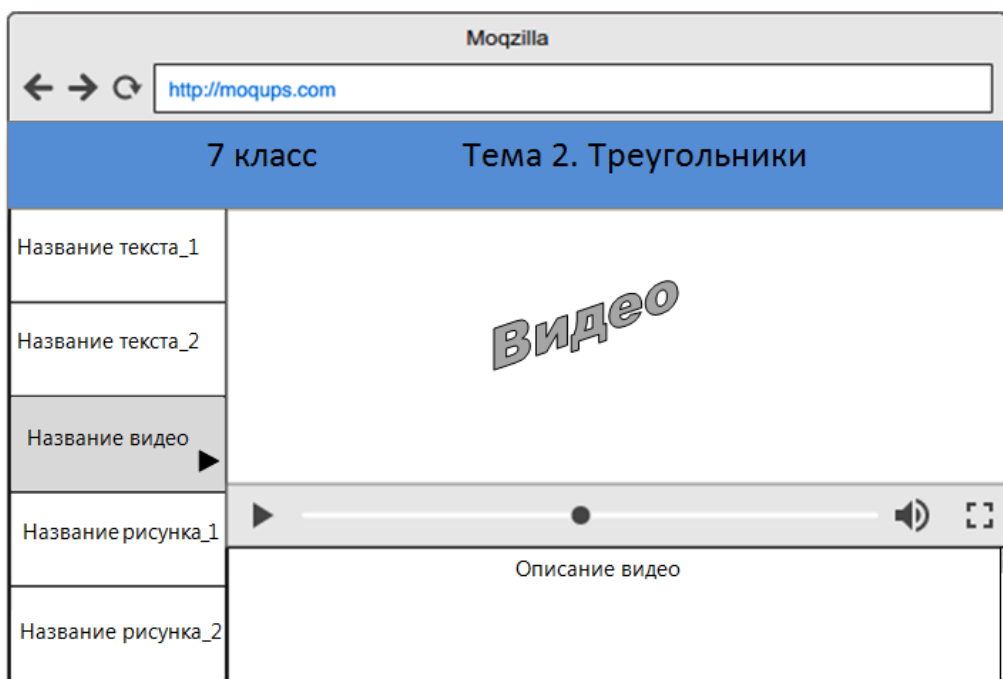


Рисунок 4.16 – Прототип вікна для перегляду учнем навчальних ресурсів

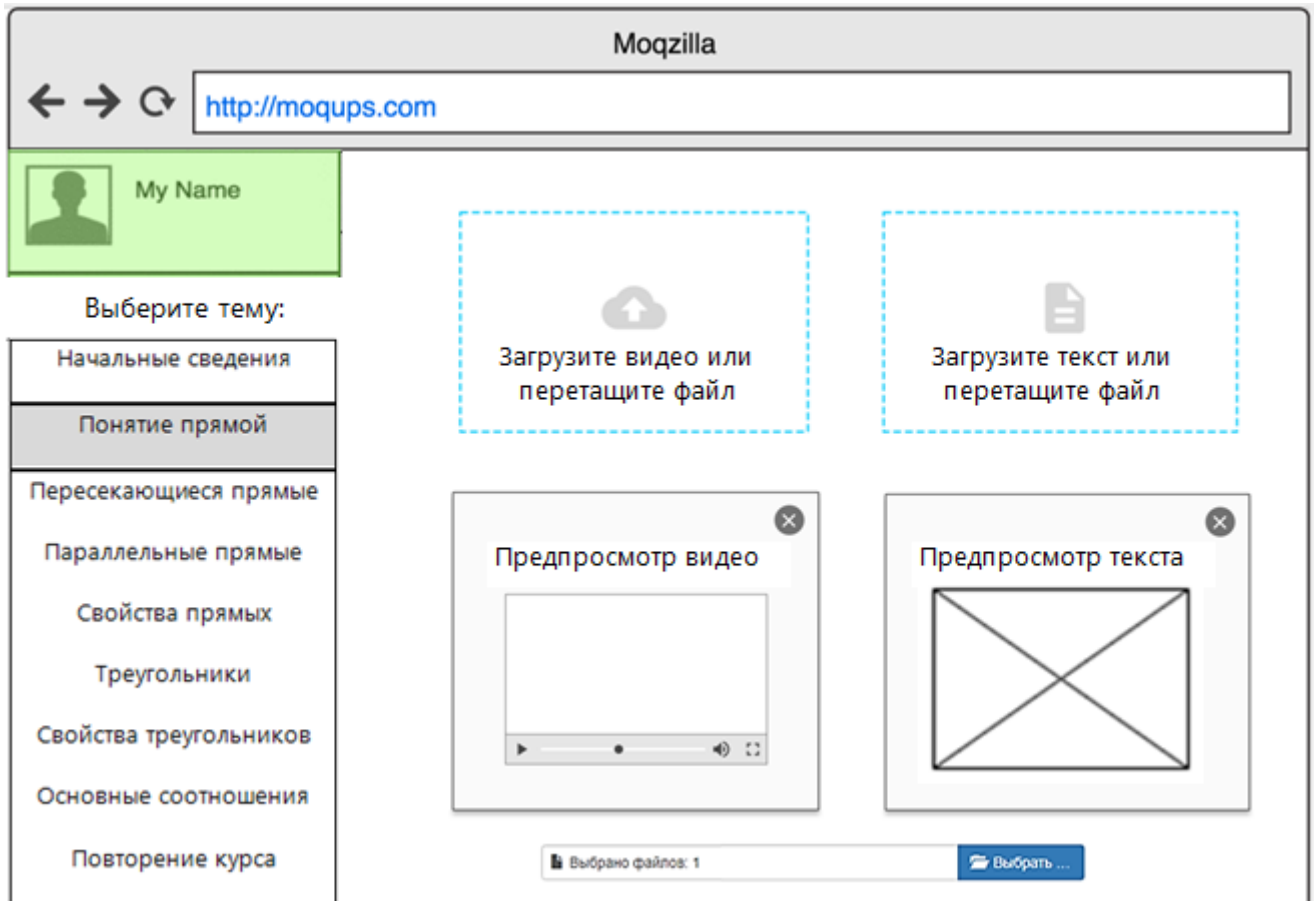


Рисунок 4.17 – Прототип вікна для завантаження вчителем навчальних ресурсів

4.5 Проектування структури класів програмного засобу

Згідно з архітектурою MVC, класи утворюють контролер, модель та вигляд.

До класів контролеру відносяться:

- AuthorController, який відповідає за авторизацію на сервері користувачів;
- TeacherController, що управляє даними вчителів;
- PupilController, який керує інформацією щодо учнів;
- TaskController, що дозволяє керувати завданнями з шкільного курсу

геометрії.

Далі для кожного класу визначимо методи та їх призначення.

AuthorController:

login() – метод для авторизації вчителя та учня.

TeacherController:

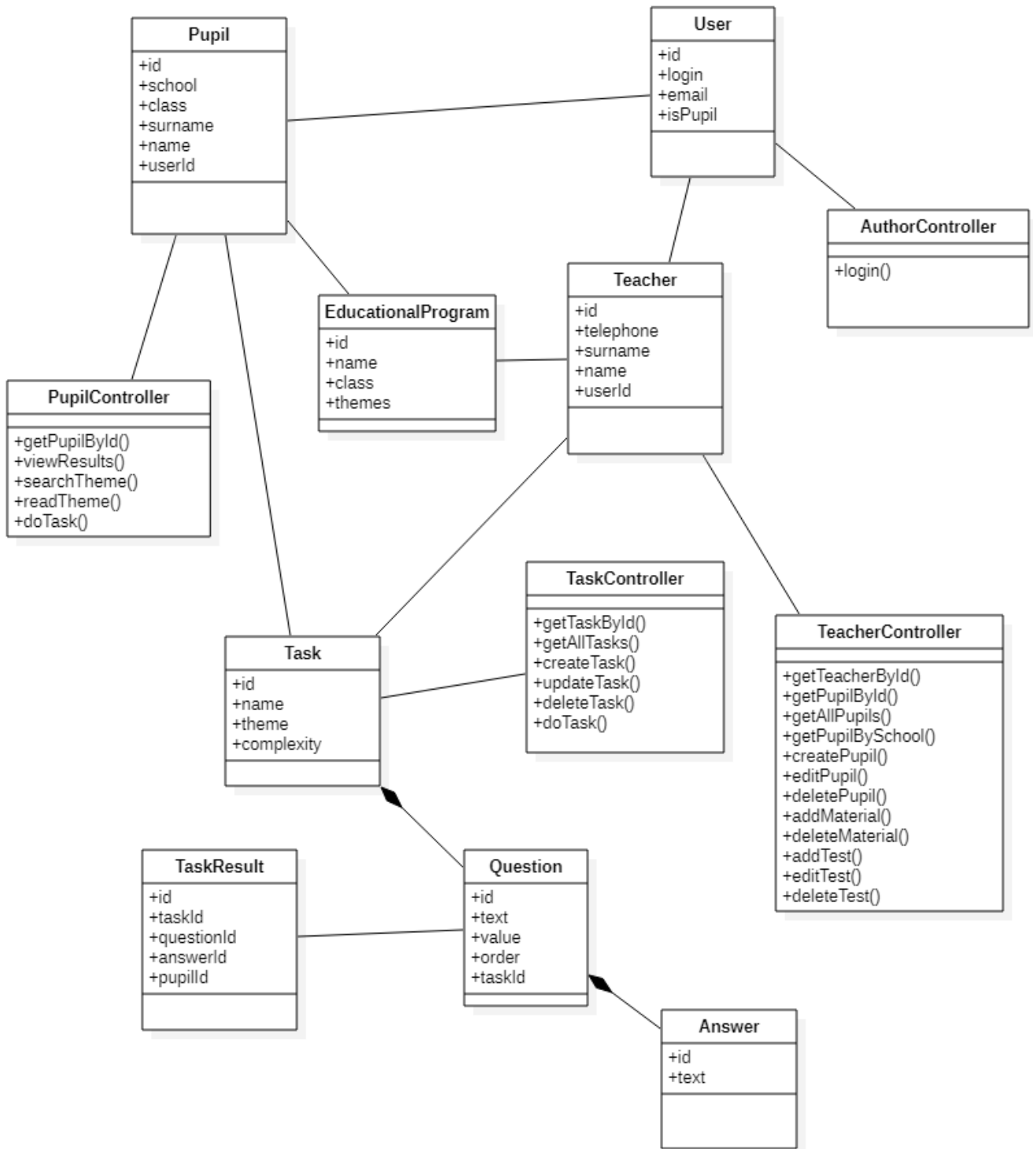


Рисунок 4.18 – Діаграма класів програмного засобу

- `getTeacherById()` – метод для отримання з бази даних інформації щодо вчителя за його `id`-номером;
- `getPupilById()` – метод для завантаження інформації по учню за його `id`-номером;

- getAllPupils() – метод для завантаження інформації до всіх учнів;
- getPupilBySchool() – метод для завантаження інформації до учнів по номеру школи;
- getPupilByClass() – метод завантаження інформації до учнів по номеру класу;
- createPupil() – метод створення запису для нового учня у базі;
- editPupil() – метод для редагування запису для учня у базі;
- deletePupil() – метод видалення з бази запису для учня;
- addTask() – метод додавання вчителем нового завдання з геометрії у систему;
- editTask() – метод для редагування завдання з геометрії;
- deleteTask() – метод видалення з бази завдання з геометрії.

Клас PupilController:

- getPupilById() – метод отримання учнем своєї картки профілю;
- viewResults() – запит на перегляд учнем результатів свого навчання;
- searchTheme() – запит на пошук учнем теми з геометрії;
- readTheme() – запит на перегляд учнем теми з геометрії;
- doTask() – запит на виконання завдання з геометрії.

Клас TaskController:

- getTaskById() – метод для отримання з бази завдання за його id-номером;
- getAllTasks() – метод для завантаження всіх завдань;
- createTask() – метод для створення нового завдання з геометрії;
- updateTask() – метод для редагування завдання з геометрії;
- deleteTask() – метод для видалення завдання з бази даних засобу;
- doTask() – метод, що дозволяє виконувати завдання з геометрії.

Клас User містить наступні дані: ідентифікатор, логін, електронна адреса, чи є користувач учнем.

Клас Teacher містить дані: ідентифікатор, номер телефону, прізвище та ім'я, код користувача, коди пов'язаних з ним груп абітурієнтів.

Клас Pupil містить дані: ідентифікатор учня, номер його школи, номер класу, прізвище та ім'я, код користувача.

Клас EducationalProgram містить дані: ідентифікатор навчальної програми з геометрії, її назву, клас, для якого призначена ця програма, та список тем, які до неї входять.

Між класами Task та Question є відношення композиції. Клас Task включає id-номер завдання, його назву, тему, до якої відноситься завдання, та складність завдання.

Клас Question містить ідентифікатор питання завдання, текст питання, вагу питання (у разі, коли за різні питання надаються різні бали), порядок у завданні та id-номер завдання.

Між класами Question та Answer є відношення композиції. Клас Answer містить id-номер відповіді та її текст.

У класі TaskResult присутні id-номер результату, а також id-номер завдання, питання, відповіді на питання та учня, що розв'язував завдання.

4.6 Висновки до розділу

У розділі 4 виконано проектування засобу з навчання геометрії. Для цього визначено архітектуру засобу та створено діаграму станів. Розроблено основні діаграми, що демонструють роботу системи.

Визначено реляційні таблиці даних та створено структуру бази даних. Розроблено прототипи інтерфейсу для роботи основної частини функціоналу.

Виконано проектування структури класів програмного засобу, надано опис створених класів.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУВАННЯ

5.1 Інструменти створення навчального засобу

«Мова програмування Java є, безсумнівно, однією з найкращих мов програмування, доступних серйозним програмістам. І однією з найважливіших переваг мови Java, це незалежність від платформи, на якій виконуються його програми. Один і той же код можна запускати під керуванням різних операційних систем, таких як Windows, Solaris, Linux і т.д. І це особливо необхідно, коли програми завантажуються через Інтернет, для подальшого виконання на комп'ютерах користувача в абсолютно різних операційних системах, які на них встановлені.» [9]

«JavaScript – це мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні клієнта. Якщо бути точнішим, то JavaScript – це не лише мова програмування на стороні клієнта. Найбільшу популярність JavaScript забезпечило програмування за клієнта.

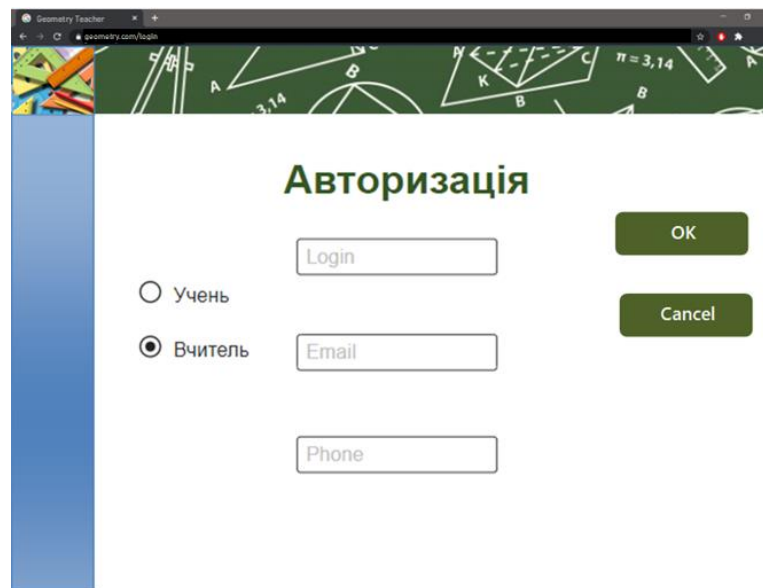
Основна ідея JavaScript полягає у можливості зміни значень атрибутів HTML-контейнерів та властивостей середовища відображення у процесі перегляду HTML-сторінки користувачем. При цьому перезавантаження сторінки не відбувається. Насправді це виявляється у тому, що можна, наприклад, змінити колір фону сторінки чи інтегровану у документ картинку, відкрити нове вікно чи попередження.» [10]

«SQLite – це програмна бібліотека, яка реалізує автономний, безсерверний, транзакційний механізм бази даних SQL із нульовою конфігурацією. SQLite є одним з найбільш швидко зростаючих механізмів баз даних, але він зростає з погляду популярності, а не його розміру. Вихідний код SQLite знаходиться у відкритому доступі. Двигун SQLite не є автономним процесом, як інші бази даних, можна зв'язати його статично або динамічно відповідно до вимог до застосування. SQLite не вимагає окремого серверного процесу або системи для роботи (без сервера). Повна база даних SQLite зберігається в одному міжплатформному файлі на диску.» [11]

5.2 Приклади використання навчального засобу

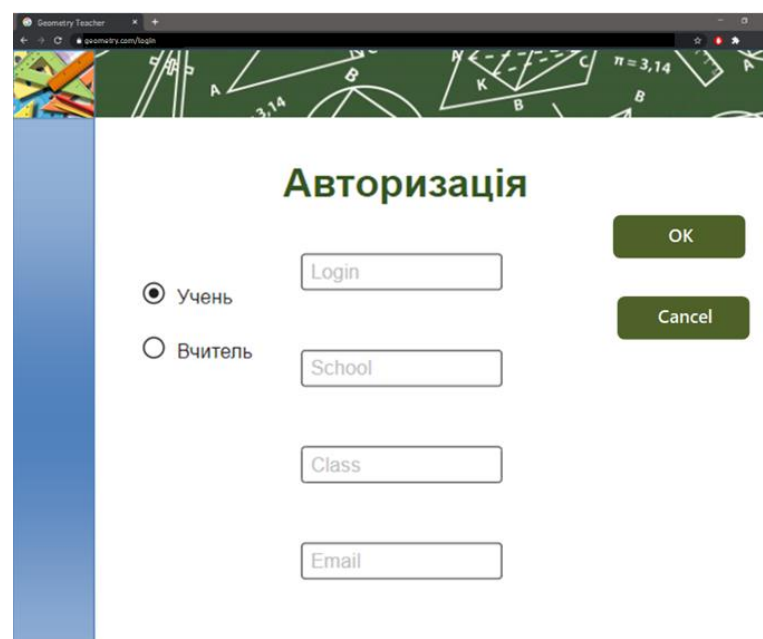
Наведемо у цьому підрозділі екранні форми розроблюваного навчального засобу.

На рис. 5.1 та 5.2 показані вікна для авторизації двох категорій користувачів: вчителя та учня.



The screenshot shows a web browser window titled "Geometry Teacher" with the URL "geometry.com/login". The page has a green header with geometric diagrams and the text "Авторизація". On the left, there are two radio buttons: "Учень" (unselected) and "Вчитель" (selected). To the right of the "Вчитель" option are three input fields: "Login", "Email", and "Phone". On the far right, there are two green buttons: "OK" and "Cancel".

Рисунок 5.1 – Вікно авторизації вчителя



The screenshot shows a web browser window titled "Geometry Teacher" with the URL "geometry.com/login". The page has a green header with geometric diagrams and the text "Авторизація". On the left, there are two radio buttons: "Учень" (selected) and "Вчитель" (unselected). To the right of the "Учень" option are three input fields: "Login", "School", and "Class". Below these is a fourth input field labeled "Email". On the far right, there are two green buttons: "OK" and "Cancel".

Рисунок 5.2 – Вікно авторизації учня

До основного функціоналу програми належить надання теоретичних матеріалів з геометрії. На основі матеріалів для 7-го класу показано приклади екранних форм: рис. 5.3 – з використанням текстового подання матеріалу, рис. 5.4 – з застосуванням рисунків, рис. 5.5 – відео з програванням у стандартному програвачі (можна завантажувати відео з Інтернету).

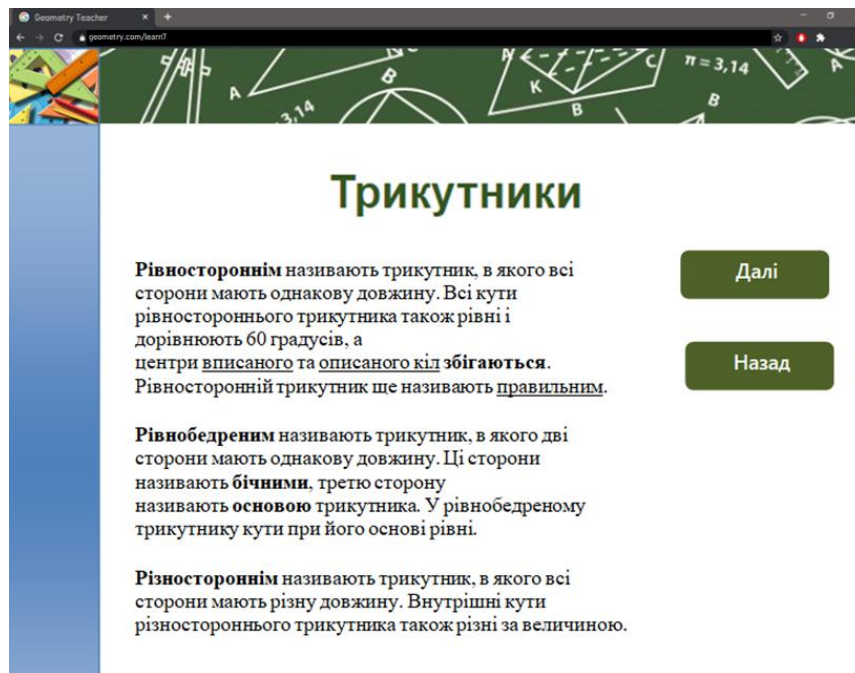


Рисунок 5.3 – Подання навчального матеріалу у вигляді тексту

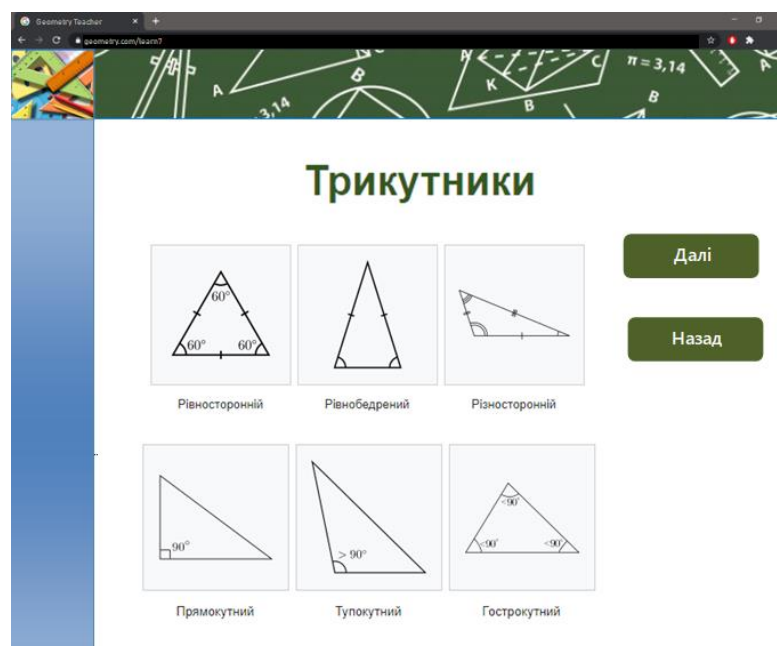


Рисунок 5.4 – Подання навчального матеріалу у вигляді рисунків

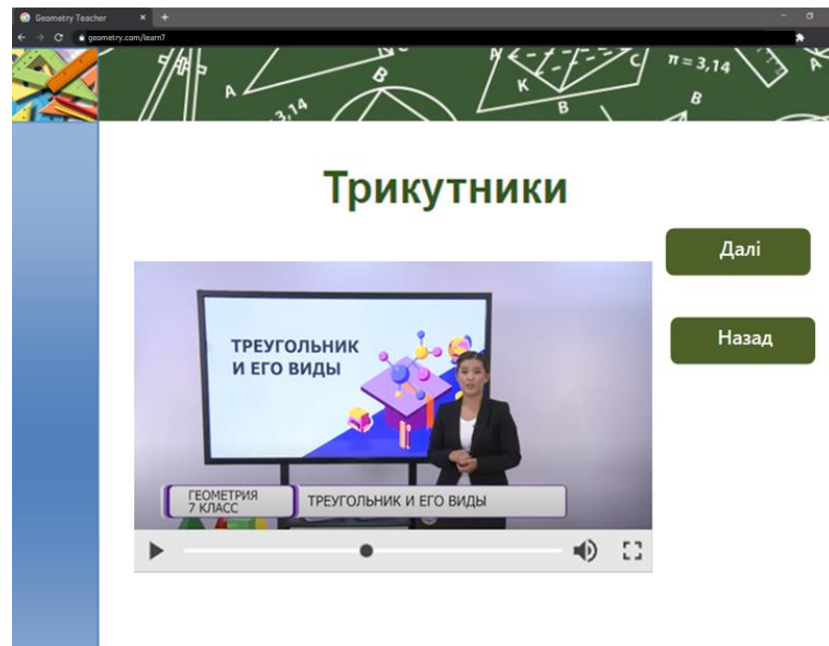


Рисунок 5.5 – Подання навчального матеріалу у вигляді відео

Практична частина представлена задачами та тестами. На рис. 5.6 показан приклад вікна з задачею. Учень може вписувати рішення у наданому текстовому полі або завантажувати відповідь як окремий графічний файл (у цьому разі відповідь може містити на фото як текст, так і рисунки).

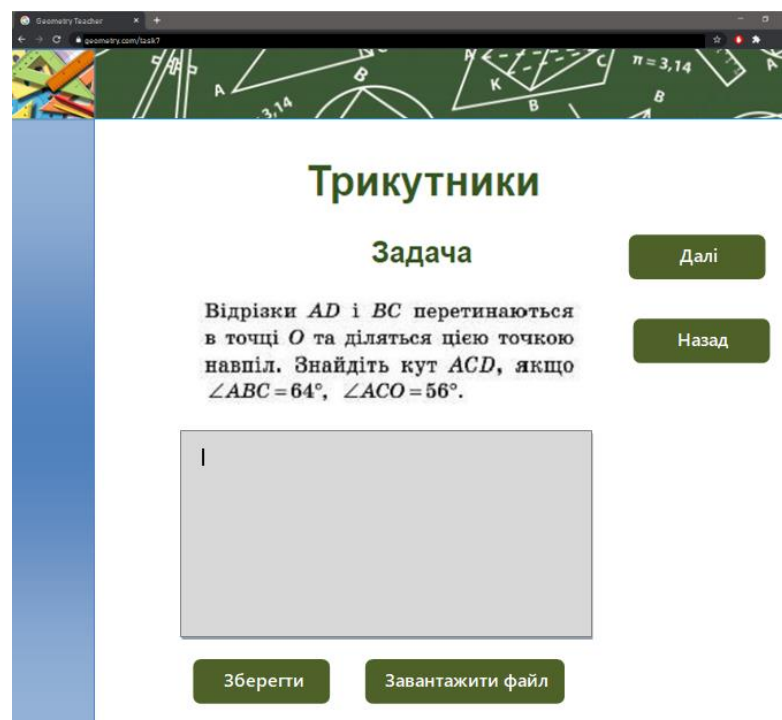


Рисунок 5.6 – Форма для вирішення задач

На рис. 5.7 наведено приклад форми для вирішення тестів. Після вибору вірних відповідей учень повинен натиснути кнопку «Зберегти».

Geometry Teacher

geomoby.com/12d7

Трикутники

Тест - Властивості трикутника

Далі

Які кути лежать навпроти рівних сторін у трикутнику?

Назад

рівні

різні

гострі

прямі

Зберегти

Чому дорівнює зовнішній кут трикутника?

різниці між 180° та відповідним внутрішнім кутом

різниці між 360° та відповідним внутрішнім кутом

180 градусів

90 градусів

Рисунок 5.7 – Форма для вирішення тестів

5.3 Висновки до розділу

У розділі розглянуто інструменти та мови розробки: Java, JavaScript, SQLite та показано їх особливості.

Наведено приклади інтерфейсу для роботи навчального засобу для вивчення шкільного курсу геометрії.

6 ВИЗНАЧЕННЯ ВЛАСТИВОСТЕЙ ЗАСОБУ ДЛЯ НАВЧАННЯ ГЕОМЕТРІЇ

6.1 Тестування функціональності з використанням сценаріїв

Для тестування функціональності системи [12] потрібно спочатку визначити перелік та короткий опис тестових сценаріїв (табл. 6.1).

Таблиця 6.1 – Тестові сценарії

Функціональна вимога	Позначення тестового сценарію	Вхідні дані	Мета проведення тесту
1	2	3	4
Авторизація вчителя та учня	Test scenario1	Авторизаційні дані вчителя та учня	Перевірити коректність авторизації вчителя та учня
Вирішення завдань з геометрії	Test scenario2	Користувач авторизован як учень, завдання доступно для вирішення	Перевірити можливість вирішення практичних завдань
Перегляд результати вирішених завдань	Test scenario3	Користувач авторизован як учень, учень почав навчання	Перевірити можливість зберігати результати навчання
Вивчення теоретичних матеріалів з геометрії	Test scenario4	Авторизація в профілі абітурієнта, наявність матеріалів	Перевірити можливість теоретичних матеріалів з геометрії

Продовження таблиці 6.1

1	2	3	4
Перегляд профілів учнів	Test scenario5	Користувач авторизован як вчитель, в базі є профілі учнів	Перевірити можливість доступу до даних учнів
Робота з навчальними матеріалами	Test scenario6	Користувач авторизован як вчитель, в базу завантажено матеріали	Перевірити можливість доступу до навчальних матеріалів з геометрії та їх зміни
Робота з завданнями з геометрії	Test scenario7	Користувач авторизован як вчитель, в базу завантажено практичні завдання	Перевірити можливість доступу до практичних завдань з геометрії та їх зміни
Встановлення методики навчання	Test scenario8	Користувач авторизован як вчитель, в базу завантажено практичні завдання	Перевірити можливість встановлення порядку виконання практичних завдань
Роздрук картки учня	Test scenario9	Користувач авторизований як вчитель або учень, картка існує у системі	Перевірити можливість отримання роздруківки даних по учню

Для означених тестових сценаріїв створено тесту, результати тестування містяться у табл. 6.2 – 6.10.

Таблиця 6.2 – Test scenario 1

Умова	Вчитель чи учень ввели правильні дані у ході авторизації	T	F
Дія	Надати користувачеві доступ	+	
	Вивести повідомлення про помилку		+
		Успіх	Успіх

Таблиця 6.3 – Test scenario 2

Умова	Учень обирає завдання з геометрії	T	T	T
	Текст завдання завантажується та відображається	T	F	T
	Учень правильно вирішує завдання	T	x	F
Дія	Система перевіряє правильність відповіді	+		+
	Система надає повідомлення про успіх рішення	+		
	Система надає повідомлення про помилку рішення			+
	Помилка завантаження завдання		+	
		Успіх	Успіх	Успіх

Таблиця 6.4 – Test scenario 3

Умова	Користувач авторизован як учень	T	T	T	F
	Учень почав навчання	T	F	T	x
	Дані учня завантажено	T	T	F	x
Дія	Надати учню дані про його навчання	+			
	Результати відсутні		+		+
	Помилка завантаження			+	+
		Усп.	Усп	Усп	Усп

Таблиця 6.5 – Test scenario 4

Умова	Користувач авторизован як учень	T	T	F
	Теоретичні матеріали з геометрії завантажено у систему	T	F	x
Дія	Надати учню матеріали	+		
	Видати повідомлення про помилку		+	+
		Успіх	Успіх	Успіх

Таблиця 6.6 – Test scenario 5

Умова	Вчитель авторизований	T	T	T	F
	Профіль учня є зареєстрованим	T	T	F	x
	Профіль учня завантажений	T	F	x	x
Дія	Відобразити картку учня	+			
	Помилка перегляду профіля		+	+	+
		Успіх	Успіх	Успіх	Успіх

Таблиця 6.7 – Test scenario 6

Умова	Вчитель авторизований	T	T	T	F
	Вчитель визначився з дією	T	T	F	x
	Є підключення до БД	T	F	x	x
Дія	Відображення матеріалів та доступ до роботи з ними	+			
	Помилка авторизації				+
	Надати вчителю повідомлення про необхідність визначення дій з навчальними матеріалами			+	
	Повідомлення про помилку підключення до бази		+		
		Успіх	Успіх	Успіх	Успіх

Таблиця 6.8 – Test scenario 7

Умова	Вчитель авторизований	T	T	T	F	T	T	T
	Вчитель безпомилково додав нове завдання	T			x	F	x	T
	Вчитель безпомилково відредагував завдання		T		x	x	F	T
	Вчитель коректно видалив завдання			T	x	x	x	T
	Підключення до БД встановлено	T	T	T	x	x	x	F
Дія	Збереження БД завдань	+	+	+				
	Помилка авторизації				+			
	Помилка додавання завдання					+		
	Помилка редагування завдання						+	
	Помилка підключення до бази даних							+
		Усп.	Усп.	Усп.	Усп.	Усп.	Усп.	Усп.

Таблиця 6.9 – Test scenario 8

Умова	Користувач авторизован як вчитель	T	T	T	F
	Вчитель обрав методику	T	F	T	x
	Підключення до БД встановлено	T	T	F	x
Дія	Система встановила методику навчання	+			
	Помилка авторизації				+
	Не обрана методика навчання		+		
	Помилка підключення до бази даних			+	
		Усп.	Усп	Усп	Усп

Таблиця 6.10 – Test scenario 9

Умова	Користувач авторизован як вчитель	T	T	T	F
	Вчитель обрав картку учня	T	F	T	x
	Підключення до БД встановлено	T	T	F	x
Дія	Надати вчителю роздруківку	+			
	Не обрана картка		+		
	Помилка підключення до бази даних			+	
	Помилка авторизації				+
		Усп.	Усп	Усп	Усп

Успішне проведення тестів згідно з побудованими сценаріями підтверджує, що розроблений програмний навчальний засіб відповідає поставленим функціональним вимогам.

6.2 Експериментальна перевірка зниження часу на навчання

Для перевірки часу, який витрачається на вивчення матеріалу з геометрії, було поставлено наступний експеримент [13].

Було обрано по одній темі з програми для 7 та для 8 класів загальноосвітньої школи. Було запропоновано вивчення теми для 7 класу двом групам учнів: учні першої групи у кількості 20 осіб не використовували розроблений навчальний засіб, а вивчали тему традиційним способом, учні другої групи у кількості 16 осіб використовували навчальний засіб. Аналогічно було обрано тему з програми для 8 класу та сформовано 2 групи по 12 осіб: 8-класники першої групи не використовували навчальний засіб, а учні другої групи – використовували. Усі учні фіксували час, який їм знадобився для повного опанування теоретичного матеріалу та розв'язання практичних завдань. На рис. 6.1 і 6.2 наведені графіки витрат часу на вивчення теми «Трикутники» для учнів 7 класу, по горизонтальній осі відмічено учнів, по вертикальній – витрачений час у хвилинах.

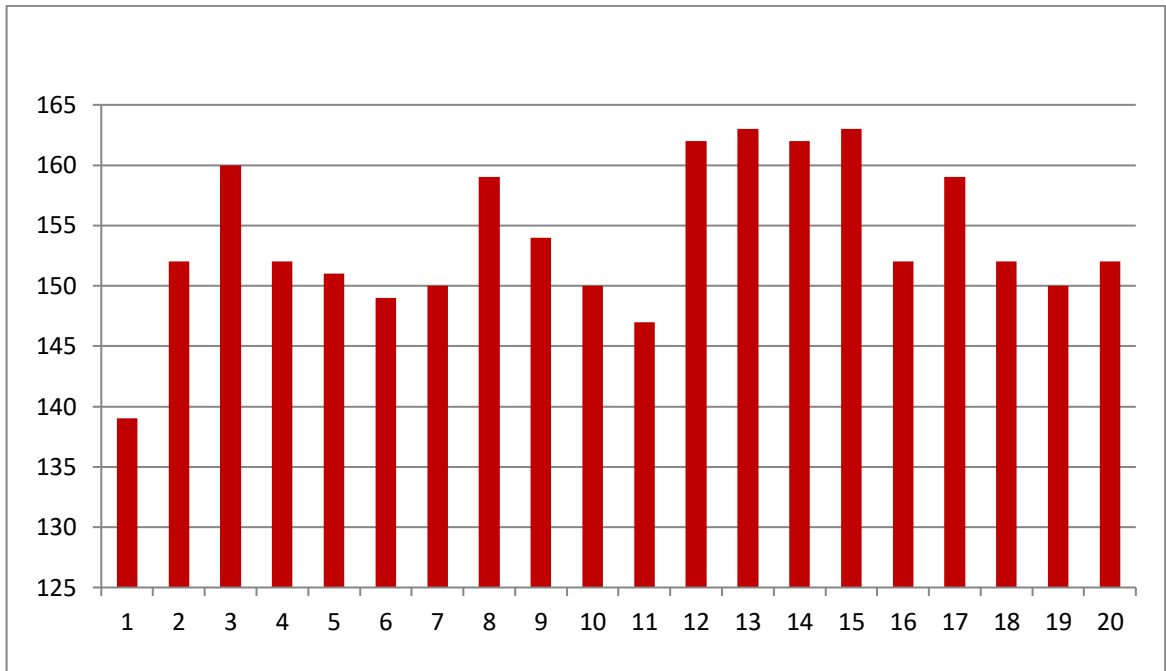


Рисунок 6.1 – Графік розподілу часу для учнів 7 класу без програми

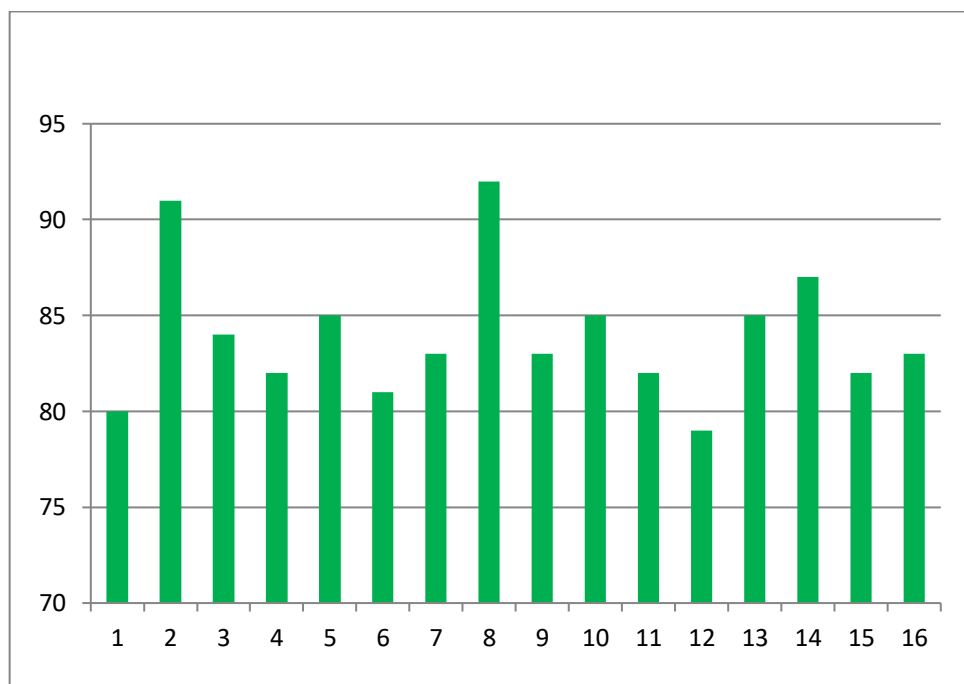


Рисунок 6.2 – Графік розподілу часу для учнів 7 класу з програмою

Аналогічно на рис. 6.3 та 6.4 наведено витрачений час на вивчення теми «Чотирикутники» для учнів 8 класу – без використання програми та з її використанням.

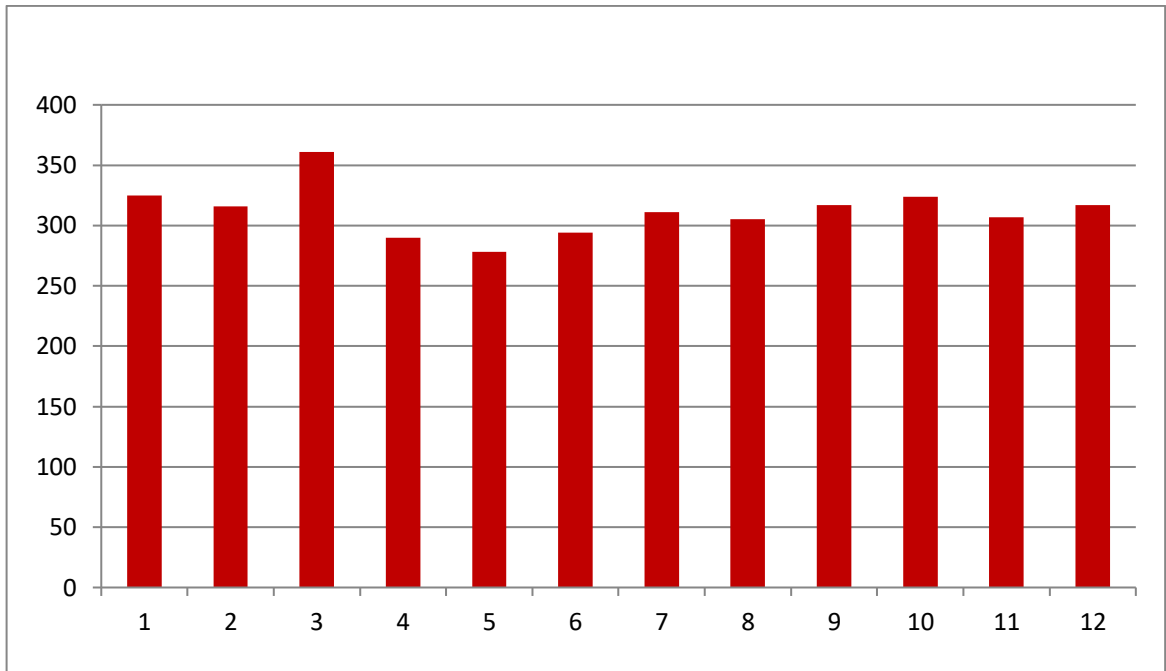


Рисунок 6.3 – Графік розподілу часу для учнів 8 класу без програми

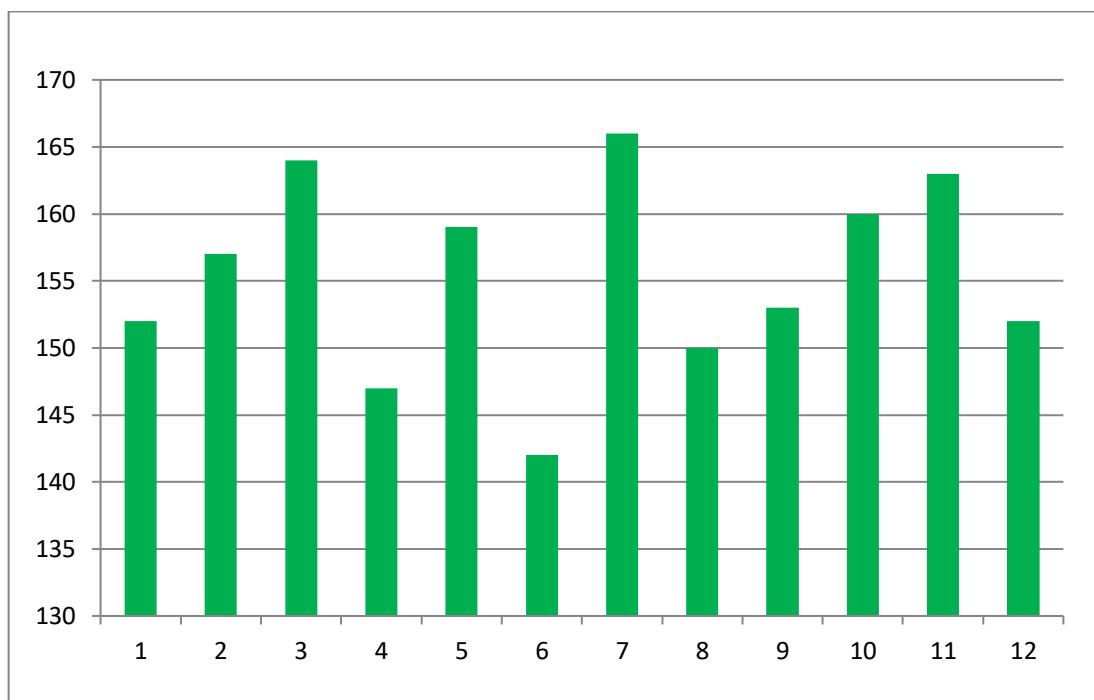


Рисунок 6.4 – Графік розподілу часу для учнів 8 класу з програмою

Усереднені дані щодо груп учнів для кожного класу наведені у табл. 6.11. Як можна побачити, обидва рази спостерігається зменшення часу: у 1,83 рази для 7 класу та у 2,01 рази для 8 класу. Можна вважати, що усереднене зменшення часу складає приблизно 1,9 рази.

Таблиця 6.11 – Аналіз досягнення мети роботи

Середній загальний час	Без використання програми, хв.	З використанням програми, хв.	Зменшення часу, рази
7 клас	154	84	1,83
8 клас	312	155	2,01

6.3 Висновки до розділу

У даному розділі визначено властивості програмного засобу для вивчення геометрії. Проведено тестування функціональності з використанням сценаріїв, усього створено 9 тестових сценаріїв. Для кожного сценарію виконано функціональне тестування та отримано успішні результати.

Проведено експеримент для визначення часу, який потрібен для вивчення певних тем з геометрії. Згідно з отриманими результатами, витрачений час зменшено приблизно у 1,9 рази.

ВИСНОВКИ

У дипломній роботі розроблено програмний засіб для навчання шкільному курсу геометрії. Засіб включає в себе підсистему викладення теоретичного матеріалу та підсистему перевірки знань за допомогою практичних завдань. Проведені експерименти показали, що використання розробленої системи дозволило зменшити час, витрачений на навчання, приблизно у 1,9 разів.

Робота містить вирішення наступних завдань.

У першому розділі проведено аналіз предметної області та визначено програмні аналоги для навчання геометрії.

У другому розділі описано модель навчання, підсистеми навчання та тестування. Також розглядаються методики навчання: послідовне, вибіркоче та адаптивне.

У третьому розділі виконано специфікацію вимог до навчального засобу з геометрії.

Четвертий розділ містить проектування архітектури засобу. Створено основні діаграми роботи та структуру бази даних. Спроектовано прототипи інтерфейсу та структуру програмних класів.

П'ятий розділ відповідає за програмну реалізацію застосування. Розглянуто інструменти розробки та показано форми інтерфейсу програмного засобу.

У шостому розділі визначено властивості засобу для навчання геометрії. Проведено функціональне тестування, виконано експериментальну перевірку зниження часу на навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gumennykova, T. R., Luhova, T. A., Riashchenko, O. I. & Troianovska, Yu. L. “Integration of the Process of Computer Game Development with Augmented Reality in Stream-Education Components”. *Herald of Advanced Information Technology. Publ. Science i Technical*. 2018; Vol.1 No.1:p.49–61. Odesa. Ukraine. DOI: <https://doi.org/10.15276/hait.01.2018.5>
2. Larshin, V. P. & Lishchenko, N. V. “Educational Technology Information Support”. *Herald of Advanced Information Technology. Publ. Science i Technical*. 2019; Vol.2 No.4: p.317–327. Odesa. Ukraine. DOI: <https://doi.org/10.15276/hait.04.2019.8>
3. Oborskyi, H. O., Saveleva, O. S., Stanovska, I. I. & Saukh, I. A. “Information Models and Methods of the Structural Crises Consequences Overcoming in the Educational Space”. *Herald of Advanced Information Technology. Publ. Science i Technical*. 2020; Vol.3 No.3:p. 185–198. Odesa. Ukraine. DOI: <https://doi.org/10.15276/hait.03.2020.7>
4. Л.А. Люшенко Я.В. Хіцко Розробка та аналіз вимог до програмного забезпечення. Компоненти програмної інженерії. Київ КПІ ім. Ігоря Сікорського. 2020 - 63 с.
5. Леффингуэлл Д. Принципы работы с требованиями. Унифицированный подход. / Д. Леффингуэлл, Д. Уидриг. - М.: Издательский дом «Вильямс», 2002. – 450с.
6. Геометрія з нуля: шкільна програма [Електронний ресурс]: – Режим доступу: https://www.test-uz.ru/video_online.php?cat=geom_7-11. – Загл. з екрану: дата звернення 13.11.2021.
7. Грофф, Д. Энциклопедия SQL / Д. Грофф. – СПб. : Питер, 2004. – 896 с.
8. Коннолли, Т. Базы данных: Проектирование, реализация, сопровождение. Теория и практика / Т. Коннолли. – М. : Издательский дом «Вильямс», 2002. – 1120 с.
9. Загальні відомості про програмування на Java [Електронний ресурс]: – Режим доступу: <https://www.opengl.org.ru/teachpro-web-dizain/obshchie-svedeniya->

o-programirovanii-na-java--vvedenie.html. – Загл. з екрану: дата звернення 15.11.2021.

10. Призначення та застосування JavaScript [Електронний ресурс]: – Режим доступу: <https://dereksiz.org/leksiya-1-naznachenie-i-primenenie-javascript-obshie-svedeniya.html>. – Загл. з екрану: дата звернення 16.11.2021.

11. SQLite - Короткий посібник [Електронний ресурс]: – Режим доступу: <https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-sqlite/sqlite-kratkoe-rukovodstvo>. – Загл. з екрану: дата звернення 16.11.2021.

12. Майерс Г. Надежность программного обеспечения. - М.: Мир, 1980. - 360 с.

13. Соснин Э. А., Пойзнер Б. Н. Методология эксперимента. Учебное пособие. - Инфра-М, 2017. - 164 с.

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```

package my_diploma;

import java.util.ArrayList;

public class GeomQuestion {
    public static final int ONE_OF_MANY=0;

    int id, type;
    float points;
    String content;
    ArrayList<String> variants;
    ArrayList<Integer> answers;
    ArrayList<String> values;
    String alternative;

    public GeomQuestion(int id, int type, String content,
ArrayList<String> variants,
        ArrayList<String> values, ArrayList<Integer>
answers, String alternative, float points) {
        this.id = id;
        this.type = type;
        this.content = content;
        if(variants!=null)
            this.variants = new ArrayList<String>(variants);
        if(answers!=null)
            this.answers = new ArrayList<Integer>(answers);
        if(values!=null)
            this.values = new ArrayList<String>(values);
        this.alternative = alternative;
        this.points = points;
    }

    public float check(ArrayList<Integer> ans, String altern)
{
        float res;

        if(answers.get(0).intValue()==ans.get(0).intValue())
            return points;
            else
                return 0;
    }

    public int getType() {
        return type;
    }

    public String getContent() {
        return content;
    }

    public ArrayList<String> getVariants() {
        return variants;
    }

    public ArrayList<String> getValues() {
        return values;
    }
}

```

```

public float getPoints() {
    return points;
}

@Override
public boolean equals(Object obj) {
    if(((GeomQuestion)obj).id==this.id)
        return true;
    return false;
}
}

public class Plot extends JPanel implements
MouseWheelListener, MouseMotionListener, MouseListener {

    final float MAX_PPP = 2, MIN_PPP = 0.01f;
    public final static int CONTINUOUS = 0;
    public final static int DISCRETE = 1;

    int render_text;
    int xtype, ytype;
    float dx, dy;
    float ppp;
    Dimension center;
    Function fun;
    boolean first;
    boolean isDrag;
    int lx, ly;

    public Plot(Function f, int X_TYPE, float DX, int Y_TYPE,
float DY) {
        super();
        render_text=50;
        ppp = 0.1f;
        xtype = X_TYPE;
        ytype = Y_TYPE;
        if(X_TYPE==CONTINUOUS)
            dx = ppp;
        else
            dx = DX;
        if(Y_TYPE==CONTINUOUS)
            dy = ppp;
        else
            dy = DY;
        center = new Dimension();
        addMouseWheelListener(this);
        addMouseListener(this);
        addMouseMotionListener(this);
        fun = f;
        first = true;
        isDrag = false;

        setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR))
;
    }
    public void setDx(float dx) {
        if(xtype==DISCRETE)
            this.dx = dx;
    }
    public void setDy(float dy) {
        if(ytype==DISCRETE)
            this.dy = dy;
    }
}

```

```

@Override
public void paintGeom(Geomics g) {
    String str;
    if(first) {
        center.width = getSize().width/2;
        center.height = getSize().height/2;
        first = false;
    }
    g.setColor(Color.white);
    g.fillRect(0, 0, getSize().width,
getSize().height);
    if(xtype==CONTINUOUS) {
        for (int i = render_text; center.width + i <=
getSize().width; i += render_text) {
            g.setColor(Color.gray);
            g.drawLine(center.width + i, 0,
center.width + i,
                getSize().height);
            str = String.format("%.2f", i * ppp);
            g.setColor(Color.black);
            g.drawString(str, center.width + i -
str.length() / 2 * 7,
                center.height + 15);
        }
        for (int i = render_text; center.width - i >=
0; i += render_text) {
            g.setColor(Color.gray);
            g.drawLine(center.width - i, 0,
center.width - i,
                getSize().height);
            g.setColor(Color.black);
            str = String.format("%.2f", -i * ppp);
            ;
            g.drawString(str, center.width - i -
str.length() / 2 * 7,
                center.height + 15);
        }
    }
    if(ytype==CONTINUOUS) {
        for (int i = render_text; center.height + i <=
getSize().height; i += render_text) {
            g.setColor(Color.gray);
            g.drawLine(0, center.height + i,
getSize().width, center.height
                + i);
            str = String.format("%.2f", -i * ppp);
            g.setColor(Color.black);
            g.drawString(str, center.width + 5,
center.height + i + 5);
        }
        for (int i = render_text; center.height - i >
0; i += render_text) {
            g.setColor(Color.gray);
            g.drawLine(0, center.height - i,
getSize().width, center.height
                - i);
            g.setColor(Color.black);
            str = String.format("%.2f", i * ppp);
            ;
            g.drawString(str, center.width + 5,
center.height - i + 5);
        }
    } else {

```

```

        for (float y = Math.round((-center.height*ppp-
dy)/dy)*dy; y<=(getSize().height-center.height)*ppp+dy; y+=dy)
{
            g.setColor(Color.gray);
            g.drawLine(0,
center.height+(int)(y/ppp), getSize().width,
                center.height+(int)(y/ppp));
            str = String.format("%.2f", y);
            g.setColor(Color.black);
            g.drawString(str, center.width+5,
                center.height+(int)(y/ppp) +
15);
        }
    }
    g.setColor(Color.black);
    g.drawLine(0, center.height, getSize().width,
center.height);
    g.drawString("x", getSize().width-7, center.height
- 7);
    g.drawLine(center.width, 0, center.width,
getSize().height);
    if(fun==null)
        return;
    Float ly=null, y, lx=null;
    float t;
    for(float x = Math.round((-center.width*ppp-
dx)/dx)*dx; x<=(getSize().width-center.width)*ppp+dx; x+=dx) {
        y = fun.get(x);
        if(y!=null && ly!=null) {
            if(ytype==DISCRETE) {
                y = Math.round(y/dy)*dy;
            }
            g.setColor(Color.red);
            g.drawLine(center.width+(int)(lx/ppp),
center.height-(int)(ly/ppp),
                center.width+(int)(x/ppp),
center.height-(int)(y/ppp));
        }
        if(xtype==DISCRETE) {
            g.setColor(Color.gray);
            g.drawLine(center.width+(int)(x/ppp), 0,
center.width+(int)(x/ppp),
                getSize().height);
            str = String.format("%.2f", x);
            g.setColor(Color.black);
            g.drawString(str,
center.width+(int)(x/ppp) - str.length() / 2 * 7,
                center.height + 15);
        }
        ly = y;
        lx = x;
    }
}
@Override
public void mouseWheelMoved(MouseWheelEvent e) {
    ppp += e.getWheelRotation()*MIN_PPP;
    if(ppp<MIN_PPP)
        ppp = MIN_PPP;
    if(ppp>MAX_PPP)
        ppp = MAX_PPP;
    if(xtype==CONTINUOUS)
        dx = ppp;
    repaint();
}

```

```

    }
    @Override
    public void mouseDragged(MouseEvent e) {
        center.width += e.getX()-lx;
        center.height += e.getY() - ly;
        lx = e.getX();
        ly = e.getY();
        repaint();
    }
    @Override
    public void mouseMoved(MouseEvent e) {
    }
    @Override
    public void mouseClicked(MouseEvent e) {
    }
    @Override
    public void mouseEntered(MouseEvent e) {
    }
    @Override
    public void mouseExited(MouseEvent e) {
    }
    @Override
    public void mousePressed(MouseEvent e) {
        lx = e.getX();
        ly = e.getY();
    }
    @Override
    public void mouseReleased(MouseEvent e) {
    }
}
interface Function {
    public Float get(float x);
}

public class UnevenDiscrete extends JPanel implements
    MouseWheelListener, MouseMotionListener, MouseListener {

    final float MAX_PPP = 2, MIN_PPP = 0.01f;

    int render_text = 50;
    float ppp;
    Dimension center;
    Function fun;
    F fff=null;
    boolean first;
    boolean isDrag;
    int lx, ly;

    public UnevenDiscrete(Function f) {
        fun = f;
        ppp = 0.1f;
        center = new Dimension();
        addMouseWheelListener(this);
        addMouseListener(this);
        addMouseMotionListener(this);
        fun = f;
        first = true;
        isDrag = false;

        setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR))
;

```

```

    }

    public boolean change(String str) {
        try {
            fff = new F(str);
        } catch (IOException e) {
            return false;
        }
        repaint();
        return true;
    }

    @Override
    public void paint(Graphics g) {
        String str;
        g.setColor(Color.white);
        g.fillRect(0, 0, getSize().width,
getSize().height);
        if(first) {
            center.width = getSize().width/2;
            center.height = getSize().height/2;
            first = false;
        }
        Float ly=null, y, lx=null;
        for (int i = render_text; center.height + i <=
getSize().height; i += render_text) {
            g.setColor(Color.gray);
            g.drawLine(0, center.height + i,
getSize().width, center.height
+ i);
            str = String.format("%.2f", -i * ppp);
            g.setColor(Color.black);
            g.drawString(str, center.width + 5,
center.height + i + 5);
        }
        for (int i = render_text; center.height - i > 0; i
+= render_text) {
            g.setColor(Color.gray);
            g.drawLine(0, center.height - i,
getSize().width, center.height
- i);
            g.setColor(Color.black);
            str = String.format("%.2f", i * ppp);
            ;
            g.drawString(str, center.width + 5,
center.height - i + 5);
        }
        int i=0;
        ly = fun.get(0);
        lx = 0f;
        for(float x = 0; x<=(getSize().width-
center.width)*ppp+2*(i+1); x+=fff.exec(i), i++) {
            y = fun.get(x);
            if(y!=null && ly!=null) {
                g.setColor(Color.black);
                g.drawLine(center.width+(int) (lx/ppp),
center.height-(int) (ly/ppp),
center.width+(int) (x/ppp),
center.height-(int) (y/ppp));
            }
            g.setColor(Color.gray);
            g.drawLine(center.width+(int) (x/ppp), 0,
center.width+(int) (x/ppp),

```

```

        getSize().height);
        str = String.format("%.2f", x);
        g.setColor(Color.black);
        g.drawString(str,
center.width+(int) (x/ppp) - str.length() / 2 * 7,
        center.height + 15);

        ly = y;
        lx = x;
    }
    i = 0;
    ly = fun.get(0);
    lx = 0f;
    for(float x = 0; x>=-center.width*ppp-2*(i+1); x-
=fff.exec(i), i++) {
        y = fun.get(x);
        if(y!=null && ly!=null) {
            g.setColor(Color.red);
            g.drawLine(center.width+(int) (lx/ppp),
center.height-(int) (ly/ppp),
            center.width+(int) (x/ppp),
center.height-(int) (y/ppp));
        }
        g.setColor(Color.gray);
        g.drawLine(center.width+(int) (x/ppp), 0,
center.width+(int) (x/ppp),
            getSize().height);
        str = String.format("%.2f", x);
        g.setColor(Color.black);
        g.drawString(str,
center.width+(int) (x/ppp) - str.length() / 2 * 7,
            center.height + 15);

        ly = y;
        lx = x;
    }
    g.setColor(Color.black);
    g.drawLine(0, center.height, getSize().width,
center.height);
    g.drawString("x", getSize().width-7, center.height
- 7);

}
@Override
public void mouseClicked(MouseEvent arg0) {
}

@Override
public void mouseEntered(MouseEvent arg0) {
}

@Override
public void mouseExited(MouseEvent arg0) {
}

@Override
public void mousePressed(MouseEvent e) {
    lx = e.getX();
    ly = e.getY();
}

@Override
public void mouseReleased(MouseEvent arg0) {
}

```

```

@Override
public void mouseDragged(MouseEvent e) {
    center.width += e.getX()-lx;
    center.height += e.getY() - ly;
    lx = e.getX();
    ly = e.getY();
    repaint();
}

@Override
public void mouseMoved(MouseEvent arg0) {
}

@Override
public void mouseWheelMoved(MouseWheelEvent e) {
    ppp += e.getWheelRotation()*MIN_PPP;
    if(ppp<MIN_PPP)
        ppp = MIN_PPP;
    if(ppp>MAX_PPP)
        ppp = MAX_PPP;
    repaint();
}

public float exec(float n) {
    Stack<Float> stack = new Stack<Float>();
    String number = "";
    for(int i=0; i<res.length(); i++) {
        char c = res.charAt(i);
        switch(c) {
            case '1':

stack.push(stack.pop()+stack.pop());
                break;
            case '2':

stack.push(stack.pop()*stack.pop());
                break;
            case '3':
                stack.push((float)
Math.cos(stack.pop()));
                break;
            case '4':
                stack.push(n);
                break;
            case ' ':
                if(number.length()!=0) {

stack.push(Float.valueOf(number));
                    number = "";
                }
                break;
            default:
                number+=c;
        }
    }
    return Math.abs(stack.pop());
}

String encode(String src, int base) {

```



```

String res="";
char c;
int index;
for(int i=0; i<src.length(); i++) {
    c = src.charAt(i);
    index = -1;
    for(int j=0; j<alph.length; j++) {
        if(c==alph[j]) {
            index = j;
            break;
        }
    }
    if(index!=-1)
        return "c"+c;
    res+=base(index, base)+" ";
}
return res;
}

String base(int num, int base) {
String res="";
while(num>=base) {
    res = num % base + res;
    num = num/base;
}
res = num + res;
return res;
}

@Override
public void mouseClicked(MouseEvent e) {
    if(e.getComponent().getName().equals("back")) {
        main.update(main.start);
        return;
    }
    String str = encode(text.getText(), 10);
    if(str.indexOf("c")==0) {
        JOptionPane.showMessageDialog(null,
"Недопустимый символ \""+str.substring(1)+"\"!");
        return;
    }
}

}

public class Draw extends JPanel{
private double scale = 15;
private int border = 20;
private double a, b;
private double xmin, xmax;
int type;
double xstep = 1;
double ystep = 1;

public double getScale () {
return this.scale;
}

public void setScale (double scale) {
this.scale = scale;
}
}

```

```

this.repaint();
    }

public double getStep () {
return (double) 1 / this.getScale();
    }

public int toRasterX (double logicX) {
return (int)Math.round((logicX * this.getScale()) +
this.getWidth()/2);
    }

public int toRasterY (double logicY) {
return (int)Math.round(-(logicY * this.getScale()) +
this.getHeight()/2);
    }

public double getLogicXmin () {
return -(this.getWidth()/2) * this.getStep();
    }

public double getLogicXmax () {
return (this.getWidth()/2) * this.getStep();
    }

public double getLogicYmin () {
return -(this.getHeight()/2) * this.getStep();
    }

public double getLogicYmax () {
return (this.getHeight()/2) * this.getStep();
    }

double closestValue(double number, double step)
{
doubleleft_value = ((int)(number/step))*step;
    left_value += step;
}
return left_value;
}

import * as React from "react";
import { TestQuestionElementBase, ReactTestElement } from
"./reactquestion_element";
import { QuestionRadiogroupModel } from
"./question_radiogroup";import { ItemValue } from
"./itemvalue";
import { TestQuestionCommentItem } from
"./reactquestion_comment";import { ReactQuestionFactory } from
"./reactquestion_factory"; import { Base } from "../base";

export class TestQuestionRadiogroup extends TestQuestionElementBase {
    constructor(props: any) {
        super(props);
    }
    protected get question(): QuestionRadiogroupModel {
        return this.questionBase as
            QuestionRadiogroupModel;
    }
}

```

```

protected renderElement(): JSX.Element {
  var cssClasses =
  this.question.cssClasses; var
  clearButton = null;
  if
    (this.question.canShowClearButton)
    {clearButton = (
      <div>
        <input
          type="button"
          className={this.question.cssClasses.clearButton}
          onClick={() => this.question.clearValue()}
          value={this.question.clearButtonCaption}
        />
      </div>
    )};
  }
  return (
    <fieldset
      className={cssClasses.ro
        ot}
      ref={(fieldset) => (this.control =
        fieldset)} role="radiogroup"
    >
      <legend aria-label={this.question.locTitle.renderedHtml} />
      {this.question.hasColumns
        ? this.getColumns(cssClasses)
        : this.getItems(cssClasses)}
      {clearButton}
    </fieldset>
  );
}
protected getColumns(cssClasses:
  any) {var value =
  this.getStateValue();
  return this.question.columns.map((column: any, ci: number)
    => {var items = column.map((item: any, ii: number) =>
      this.renderItem(item, value, cssClasses, "" + ci + ii)
    )};
    return (
      <div key={"column" + ci} className={this.question.getColumnClass()}>
        {items}
      </div>
    );
  });
}
protected getItems(cssClasses: any):
  Array<any> {var items = [];
  var value = this.getStateValue();
  for (var i = 0; i < this.question.visibleChoices.length; i++) {var item =
  this.question.visibleChoices[i]; items.push(this.renderItem(item, value,
  cssClasses, "" + i));
  }
  return items;
}
protected get textStyle(): any {
  return { marginLeft: "3px", display: "inline", position: "static" };
}
private
  renderItem(
    item:
    ItemValue,
    value: any,

```

```

    cssClasses:
    any, index:
    string
  ): JSX.Element {
    var key = "item" +
    index; return (
      <TestQuestionRadioItem
        key={key}
        question={this.question}
        cssClasses={cssClasses}
        isDisplayMode={this.isDisplayMode}
        item={item}
        textStyle={this.textStyle}
        index={index}
        isChecked={value === item.value}
      />
    );
  }
  private getStateValue(): any {
    return !this.question.isEmpty() ? this.question.renderedValue : "";
  }
}

export class TestQuestionRadioItem extends ReactTestElement {
  constructor(props: any) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
  }
  protected getStateElement():
  Base {return this.item;
  }
  protected get question():
  QuestionRadiogroupModel {return
  this.props.question;
  }
  protected get item():
  ItemValue {return
  this.props.item;
  }
  protected get textStyle():
  any {return
  this.props.textStyle;
  }
  protected get index():
  number {return
  this.props.index;
  }
  protected get isChecked():
  boolean {return
  this.props.isChecked;
  }
  private get hideCaption(): boolean {
    return this.props.hideCaption ===
    true;
  }
  public shouldComponentUpdate():
  boolean {return (
    !this.question.customWidget || !this.question.customWidgetData.isNeedRender
    ||
    !!this.question.customWidget.widgetJson.isDefaultRender ||
    !!this.question.customWidget.widgetJson.render
  )
}

```

```

    );
  }
  handleOnChange(event: any) {
    this.question.renderedValue =
      this.item.value;
  }
  protected canRender(): boolean {
    return !!this.question && !!this.item;
  }
  protected renderElement():
    JSX.Element {var otherItem =
      this.isChecked && this.item.value === this.question.otherItem.value
        ? this.renderOther(this.cssClasses)
        : null;

  var id = this.question.inputId + "_" +
    this.index; var itemText = !this.hideCaption
      ? this.renderLocString(this.item.locText, this.textStyle)
      : "";
  var itemClass =
    this.question.getItemClass(this.item); var labelClass
    = this.question.getLabelClass(this.item); var locText:
    any = this.item.locText;
  var controlLabelClass = this.question.getControlLabelClass(this.item);

  return (
    <div className={itemClass}>
      <label className={labelClass}>
        <input
          className={this.cssClasses.itemControl}
          id={id}
          type="radio"
          name={this.question.name + "_" +
            this.question.id} checked={this.isChecked}
          value={this.item.value}
          disabled={this.isDisplayMode ||
            !this.item.isEnabled}
          onChange={this.handleOnChange}
          aria-
            required={this.question.isRequired}
            aria-label={locText.renderedHtml}
            aria-invalid={this.question.errors.length >
              0} aria-describedby={
              this.question.errors.length > 0
                ? this.question.id + "_errors"
                : null
            }
          role="radio"
        />
        <span className={this.cssClasses.materialDecorator}>
          <svg
            className={this.cssClasses.itemDecorator}
            viewBox="-12 -12 24 24"
          >
            <circle r="6" cx="0" cy="0" />
          </svg>
        </span>
        <span className="check" />
        <span className={controlLabelClass} title={locText["koRenderedHtml"]} >
          {itemText}
        </span>
      </label>
    </div>
  );

```

```

        {otherItem}
      </div>
    );
  }
  protected renderOther(cssClasses: any):
    JSX.Element {return (
      <div className="form-group">
        <TestQuestionCommentItem
          question={this.question}
          otherCss={cssClasses.other}
          cssClasses={cssClasses}
          isDisplayMode={this.isDisplayMode}
        />
      </div>
    );
  }
}

ReactQuestionFactory.Instance.registerQuestion("radiogroup", (props)
=> {return React.createElement(TestQuestionRadiogroup, props);
});

import * as React from "react";
import { ReactTestModel } from "../reactTestmodel";
import { TestPage } from "../page";
import { TestNavigation } from "../reactTestNavigation";
import { TestError, Base } from "../base";
import { Question } from "../question";
import { ITestCreator } from "../reactquestion";
import { ReactQuestionFactory } from
"./reactquestion_factory";import { TestCss } from
"../defaultCss/cssstandard";
import { TestTimerPanel } from "../reacttimerpanel";
import { TestElementBase, TestLocString } from "../reactquestion_element";
import { PageModel } from "../page";
import { StylesManager } from
"../stylesmanager";import { Helpers } from
"../helpers";
import { ReactElementFactory } from "../element-factory";

export class Test extends TestElementBase implements ITestCreator {
  private previousJSON = {};
  private rootRef:
  React.RefObject<HTMLDivElement>;public static
  get cssType(): string {
    return TestCss.currentType;
  }
  public static set cssType(value:
  string) {
    StylesManager.applyTheme(value);
  }
  protected Test: ReactTestModel;
  private isCurrentPageChanged: boolean = false;
  private onCurrentPageChangedHandler = (sender: any, options: any): any
=> {this.isCurrentPageChanged = true;
};

  constructor(props:
  any) {super(props);
  this.handleTryAgainClick =
  this.handleTryAgainClick.bind(this);this.createTest(props);

```

```

    this.updateTest(props, {});
    //set the first page
    var dummy =
    this.Test.currentPage;
    this.rootRef = React.createRef();
  }
  protected getStateElement():
    Base {return this.Test;
  }shouldComponentUpdate(nextProps: any, nextState: any) {if
  (this.isModelJSONChanged(nextProps)) {
    this.createTest(nextProps);
    this.updateTest(nextProps,
    {});
  }
  return true;
}
  componentDidMount(prevProps: any, prevState:
  any) {super.componentDidUpdate(prevProps,
  prevState); if (this.isCurrentPageChanged) {
    this.isCurrentPageChanged = false;
    this.Test.scrollToTopOnPageChange();
  }
  this.updateTest(this.props, prevProps);
}
  componentDidMount() {
    super.componentDidMount
    ();
    var el = this.rootRef.current;
    if (el && this.Test) this.Test.doAfterRenderTest(el);if
    (this.Test) {
      this.Test.startTimerFromUI();
    }
  }
  componentWillUnmount() {
    super.componentWillUnmount
    (); if (this.Test) {
      this.Test.stopTimer();
      this.Test.onCurrentPageChanged.remove(this.onCurrentPageChangedHandler);
    }
  }
  doRender():
  JSX.Element {
    var renderResult;
    if (this.Test.state == "completed") {
      renderResult = this.renderCompleted();
    } else if (this.Test.state == "completedbefore")
      {renderResult = this.renderCompletedBefore();
    } else if (this.Test.state == "loading")
      {renderResult = this.renderLoading();
    } else if (this.Test.state == "starting")
      {renderResult = this.renderStartPage();
    } else {
      renderResult = this.renderTest();
    }
  }
  var header = this.renderHeader();
  var onSubmit = function (event:
  React.FormEvent<HTMLFormElement>) {event.preventDefault();
  };
  var customHeader = <div className="sv_custom_header"
  />;if (this.Test.hasLogo) {
    customHeader = null;
  }
  return (

```

```

    <div ref={this.rootRef} className={this.css.root}>
      <form onSubmit={onSubmit}>
        {customHeader}
        <div className={this.css.container}>
          {header}
          {renderResult}
        </div>
      </form>
    </div>
  );
}

protected renderElement(): JSX.Element {return this.doRender();
}
public get css(): any
{ return
  this.Test.css;
}
public set css(value:
  any) {this.Test.css =
  value;
}
handleTryAgainClick(event:
  any) {
  this.Test.doComplete();
}
protected renderCompleted(): JSX.Element {
  if (!this.Test.showCompletedPage) return
  null;var completedState = null;
  if (this.Test.completedState)
    {var tryAgainButton = null;
    if (this.Test.completedState == "error") {
      var btnText =
        this.Test.getLocString("saveAgainButton");
      tryAgainButton = (
        <input
          type={"button"}
          value={btnText}
          className={this.css.saveData.saveAgainButton}
          onClick={this.handleTryAgainClick}
        />
      );
    }
    var css =
    this.css.saveData[this.Test.completedState];
    completedState = (
      <div className={this.css.saveData.root}>
        <div className={css}>
          <span>{this.Test.completedStateText}</span>
          {tryAgainButton}
        </div>
      </div>
    );
  }
  var htmlValue = { _html: this.Test.processedCompletedHtml };
  return (
    <div>
      <div
        dangerouslySetInnerHTML={htmlValue}
        className={[this.css.body, this.css.completedPage].join("
        ") }

```



```

        />
        {completedState}
    </div>
    );
}
protected renderCompletedBefore(): JSX.Element {
    var htmlValue = { _html: this.Test.processedCompletedBeforeHtml };
    return (
        <div dangerouslySetInnerHTML={htmlValue} className={this.css.body} />
    );
}
protected renderLoading(): JSX.Element {
    var htmlValue = { _html: this.Test.processedLoadingHtml };
    return (
        <div dangerouslySetInnerHTML={htmlValue} className={this.css.body} />
    );
}
protected renderStartPage(): JSX.Element
{
    var startedPage =
        this.Test.startedPage
? this.renderPage(this.Test.startedPage)
    : null;
    var pageId = this.Test.startedPage ? this.Test.startedPage.id : "";
    return (
        <div>
            <div id={pageId} className={this.css.body}>
                {this.renderNavigation("top")}
                {startedPage}
                {this.renderNavigation("bottom")}
            </div>
        </div>
    );
}
protected renderTest(): JSX.Element {
    var currentPage = this.Test.currentPage
        ? this.renderPage(this.Test.currentPage)
        : null;
    var pageId = this.Test.currentPage ? this.Test.currentPage.id : "";
    var topProgress = this.Test.isShowProgressBarOnTop
        ? this.renderProgress(true)
        : null;
    var bottomProgress = this.Test.isShowProgressBarOnBottom
        ? this.renderProgress(false)
        : null;
    if (!currentPage) {
        currentPage = this.renderEmptyTest();
    }
    return (
        <div
            id={pageId}
            className={!currentPage ? this.css.bodyEmpty : this.css.body}
        >
            {topProgress}
            {this.renderTimerPanel("top")}
            {this.renderNavigation("top")}
            {currentPage}
            {this.renderTimerPanel("bottom")}
            {bottomProgress}
            {this.renderNavigation("bottom")}
        </div>
    );
}
}

```

```

protected renderTitle():
  JSX.Element {let title = null;
  let description = null;
  if (this.Test.title && this.Test.showTitle) {
    title = TestElementBase.renderLocString(this.Test.locTitle);
    description = TestElementBase.renderLocString(
      this.Test.locDescription
    );
  }
  return title ? (
    <div className={this.css.headerText} style={{
maxWidth:this.Test.titleMaxWidth }}>
      <h3 className={this.css.title}>{title}</h3>
      <h5 className={this.css.description}>{description}</h5>
    </div>
  ) : null;
}
protected renderHeader(): JSX.Element {
  if ((this.Test.title && this.Test.showTitle) || this.Test.hasLogo) {let
    title = this.renderTitle();
    let style: any = { objectFit: this.Test.logoFit };

    let imageBefore = null;let imageAfter = [];
    if (this.Test.isLogoBefore)
      {imageBefore = (
        <div className={this.Test.logoClassNames}>
          <img
            className={this.Test.css.logoImage}
            src={this.Test.locLogo.renderedHtml}
            width={
              this.Test.logoWidth ? this.Test.logoWidth + "px" : undefined
            }
            height={
              this.Test.logoHeight
                ? this.Test.logoHeight + "px"
                : undefined
            }
            style={style}
          />
        </div>
      );
    if (this.Test.isLogoAfter)
      {imageAfter.push(
        <div className={this.Test.logoClassNames}>
          <img
            className={this.Test.css.logoImage}
            src={this.Test.locLogo.renderedHtml}
            width={
              this.Test.logoWidth ? this.Test.logoWidth + "px" : undefined
            }
            height={
              this.Test.logoHeight
                ? this.Test.logoHeight + "px"
                : undefined
            }
            style={style}
          />
        </div>
      );
    imageAfter.push(<div className="sv-logo--right-tail"></div>);

```

```

    }
    return (
      <div className={this.css.header}>
        {imageBefore}
        {title}
        {imageAfter}
      </div>
    );
  }
  return null;
}
protected renderTimerPanel(location: string) {
  if (this.Test.showTimerPanel !== location) return
  null;return <TestTimerPanel Test={this.Test} />;
}
protected renderPage(page: PageModel):
  JSX.Element {return (
    <TestPage
      Test={this.Test}
      page={page}
      css={this.css}
      creator={this}
    />
  );
}
protected renderProgress(isTop: boolean):
  JSX.Element {return
  ReactDOMFactory.Instance.createElement(
    "Test-progress-" + this.Test.progressBarType.toLowerCase(),
    { Test: this.Test, css: this.css, isTop: isTop }
  );
}
protected renderNavigation(navPosition: string):
  JSX.Element {if (
    this.Test.isNavigationButtonsShowing !== "both" &&
    (this.Test.isNavigationButtonsShowing === "none" ||
      this.Test.isNavigationButtonsShowing !== navPosition)
  ) {
    return null;
  }
  return <TestNavigation Test={this.Test} css={this.css} />;
}
protected renderEmptyTest(): JSX.Element {
  return <span>{this.Test.emptyTestText}</span>;
}
protected createTest(newProps: any)
  {if (!newProps) newProps = {};}
  this.previousJSON = {};
  if (newProps) {
    if (newProps.model) {
      this.Test =
        newProps.model;
    } else {
      if (newProps.json) {
        this.previousJSON =
          newProps.json;
        this.Test = new ReactTestModel(newProps.json);
      }
    }
  }
  } else {
    this.Test = new ReactTestModel();
  }
}

```

```

    if (!!newProps.css) {
        this.Test.mergeCss(newProps.css, this.css);
    }
    this.setTestEvents();
}
private isModelJSONChanged(newProps: any):
    boolean {if (!!newProps["model"]) {
        return this.Test !== newProps["model"];
    }
    if (!!newProps["json"]) {
        return !Helpers.isTwoValueEquals(newProps["json"], this.previousJSON);
    }
    return false;
}
protected updateTest(newProps: any, oldProps?: any)
    {if (!newProps) return;
    oldProps = oldProps ||
    {}}; for (var key in
    newProps) {
        if (key == "model" || key == "children" || key ==
            "json") {continue;
        }
        if (key == "css") {
            this.Test.mergeValues(newProps.css, this.Test.getCss());
            this.Test["updateElementCss"]();
            continue;
        }
        if (newProps[key] === oldProps[key]) continue;
        if (key.indexOf("on") == 0 && this.Test[key] && this.Test[key].add) {if
            (!!oldProps[key]) {
                this.Test[key].remove(oldProps[key]);
            }
            this.Test[key].add(newProps[key]);
        } else {
            this.Test[key] = newProps[key];
        }
    }
}
protected setTestEvents()
    {var self = this;

    this.Test.renderCallback = function ()
        {var counter =
            !!self.state && !!self.state.modelChanged ? self.state.modelChanged :
            0;self.setState({ modelChanged: counter + 1 });
        };
    this.Test.onPartialSend.add((sender) =>
        {if (!!self.state) {
            self.setState(self.state);
        }
    });
    this.Test.onCurrentPageChanged.add(this.onCurrentPageChangedHandler);
}

//ITestCreator
public createQuestionElement(question: Question):
    JSX.Element {return
    ReactQuestionFactory.Instance.createQuestion(
        question.isDefaultRendering() ?
question.getTemplate() :question.getComponentName(),
    {
        question: question,

```

```

        isDisplayMode:
        question.isReadOnly, creator:
        this,
    }
    );
}
public
  renderError(
    key: string,
    error:
    TestError,
    cssClasses: any
  ):
  JSX.Element
  {return (
    <div key={key}>
      <span className={cssClasses.error.icon} aria-hidden="true" />
      <span className={cssClasses.error.item}>
        <TestLocString locStr={error.locText} />
      </span>
    </div>
  )};
}
public questionTitleLocation(): string {
  return this.Test.questionTitleLocation;
}
public questionErrorLocation(): string {
  return this.Test.questionErrorLocation;
}
}

const jwt =
require('jsonwebtoken') const
bcrypt = require('bcrypt')
const loginRouter = require('express').Router()
const User = require('../models/user')

loginRouter.post('/', async (request, response)
=> {const body = request.body

const user = await User.findOne({ username: body.username
})const passwordCorrect = user === null
? false
: await bcrypt.compare(body.password, user.passwordHash)

if (!(user && passwordCorrect)) {
  return
  response.status(401).json({
    error: 'invalid username or password'
  })
}

const userForToken = {
  username:
  user.username, id:
  user._id,
}

const token = jwt.sign(userForToken,
process.env.SECRET) response

```

```

        .status(200)
        .send({ token, username: user.username, name: user.name })
    })

module.exports = loginRouter

import { checkHttpStatus, parseJSON } from '../utils';
import { LOGIN_USER_REQUEST, LOGIN_USER_FAILURE, LOGIN_USER_SUCCESS,
LOGOUT_USER, FETCH_PROTECTED_DATA_REQUEST, RECEIVE_PROTECTED_DATA } from
'../constants';
import { pushState } from 'redux-
router';import jwtDecode from 'jwt-
decode';

export function
  loginUserSuccess(token) {
    localStorage.setItem('token',
token); return {
      type: LOGIN_USER_SUCCESS,
      payload: {
        token:
        token
      }
    }
  }

export function
  loginUserFailure(error) {
    localStorage.removeItem('token');
    return {
      type: LOGIN_USER_FAILURE,
      payload: {
        status: error.response.status,
        statusText: error.response.statusText
      }
    }
  }

export function
  loginUserRequest() {return {
    type: LOGIN_USER_REQUEST
  }
}

export function logout() { localStorage.removeItem('token'); return {
  type: LOGOUT_USER
}
}

export function
  logoutAndRedirect() {return
(dispatch, state) => {
    dispatch(logout());
    dispatch(pushState(null,
'/login'));
  }
}

export function loginUser(email, password,
  redirect="/") {return function(dispatch) {
  dispatch(loginUserRequest());

```

```

return
  fetch('http://localhost:3000/auth/getToken/', {
    method: 'post',
    credentials:
      'include', headers: {
        'Accept': 'application/json',
        'Content-Type':
          'application/json'
      },
    body: JSON.stringify({email: email, password: password})
  })
  .then(checkHttpStatus)
  .then(parseJSON)
  .then(response
    => {try {
      let decoded = jwtDecode(response.token);
      dispatch(loginUserSuccess(response.token));
      dispatch(pushState(null, redirect));
    } catch (e) {
      dispatch(loginUserFailure({
        response: {
          status: 403,
          statusText: 'Invalid token'
        }
      }));
    }
  })
  .catch(error => {
    dispatch(loginUserFailure(error));
  })
}

export function
  receiveProtectedData(data) {return {
    type: RECEIVE_PROTECTED_DATA,
    payload: {
      data: data
    }
  }
}

export function
  fetchProtectedDataRequest() {return {
    type: FETCH_PROTECTED_DATA_REQUEST
  }
}

export function fetchProtectedData(token) {

  return (dispatch, state) => {
    dispatch(fetchProtectedDataRequest());
    return
      fetch('http://localhost:3000/getData/
        ', {credentials: 'include',
          headers: {
            'Authorization': `Bearer ${token}`
          }
        })
      .then(checkHttpStatus)
      .then(parseJSON)
      .then(response => {
        dispatch(receiveProtectedData(response.data));
      });
  };
}

```

```

    })
    .catch(error => {
      if(error.response.status === 401)
      {
        dispatch(loginUserFailure(error));
        dispatch(pushState(null, '/login'));
      }
    })
  }
}

import * as React from "react";
import { TestQuestionElementBase, ReactTestElement } from
"./reactquestion_element";
import { QuestionRadiogroupModel } from
"../question_radiogroup";import { ItemValue } from
"../itemvalue";
import { TestQuestionCommentItem } from
"./reactquestion_comment";import { ReactQuestionFactory } from
"./reactquestion_factory"; import { Base } from "../base";

export class TestQuestionRadiogroup extends TestQuestionElementBase {
  constructor(props: any) {
    super(props);
  }
  protected get question(): QuestionRadiogroupModel {
    return this.questionBase as
    QuestionRadiogroupModel;
  }
  protected renderElement(): JSX.Element {
    var cssClasses =
    this.question.cssClasses;var
    clearButton = null;
    if
    (this.question.canShowClearButton)
    {clearButton = (
      <div>
        <input
          type="button"
          className={this.question.cssClasses.clearButton}
          onClick={() => this.question.clearValue()}
          value={this.question.clearButtonCaption}
        />
      </div>
    );
    }
    return (
      <fieldset
        className={cssClasses.root}
        ref={(fieldset) => (this.control =
        fieldset)}role="radiogroup"
      >
        <legend aria-label={this.question.locTitle.renderedHtml} />
        {this.question.hasColumns
          ? this.getColumns(cssClasses)
          : this.getItems(cssClasses)}
        {clearButton}
      </fieldset>
    );
  }
}

```



```

protected getColumns(cssClasses:
any) {var value =
this.getStateValue();
return this.question.columns.map((column: any, ci: number)
=> {var items = column.map((item: any, ii: number) =>
this.renderItem(item, value, cssClasses, "" + ci + ii)
});
return (
<div key={"column" + ci} className={this.question.getColumnClass()}>
{items}
</div>
);
});
}
protected getItems(cssClasses: any):
Array<any> {var items = [];
var value = this.getStateValue();
for (var i = 0; i < this.question.visibleChoices.length;
i++) {var item = this.question.visibleChoices[i];
items.push(this.renderItem(item, value, cssClasses, "" +
i));
}
return items;
}
protected get textStyle(): any {
return { marginLeft: "3px", display: "inline", position: "static" };
}
private
renderItem(
item:
ItemValue,
value: any,
cssClasses:
any, index:
string
): JSX.Element {
var key = "item" +
index;return (
<TestQuestionRadioItem key={key} question={this.question}
cssClasses={cssClasses}
isDisplayMode={this.isDisplayMode}
item={item}
textStyle={this.textStyle}
index={index}
isChecked={value === item.value}
/>
);
}
private getStateValue(): any {
return !this.question.isEmpty() ? this.question.renderedValue : "";
}
}

export class TestQuestionRadioItem extends ReactTestElement {
constructor(props: any) {
super(props);
this.handleChange = this.handleChange.bind(this);
}
protected getStateElement():
Base {return this.item;
}
}

```

```

protected get question():
  QuestionRadiogroupModel {return
  this.props.question;
}
protected get item():
  ItemValue {return
  this.props.item;
}
protected get textStyle():
  any {return
  this.props.textStyle;
}
protected get index():
  number {return
  this.props.index;
}
protected get isChecked():
  boolean {return
  this.props.isChecked;
}
private get hideCaption(): boolean {
  return this.props.hideCaption ===
  true;
}
public shouldComponentUpdate():
  boolean {return (
  !this.question.customWidget ||
  !!this.question.customWidgetData.isNeedRender ||
  !!this.question.customWidget.widgetJson.isDefaultRender ||
  !!this.question.customWidget.widgetJson.render
  );
}
handleOnChange(event: any) {
  this.question.renderedValue =
  this.item.value;
}
protected canRender(): boolean {
  return !!this.question && !!this.item;
}
protected renderElement():
  JSX.Element {var otherItem =
  this.isChecked && this.item.value === this.question.otherItem.value
  ? this.renderOther(this.cssClasses)
  : null;

  var id = this.question.inputId + "_" +
  this.index; var itemText = !this.hideCaption
  ? this.renderLocString(this.item.locText, this.textStyle)
  : "";
  var itemClass =
  this.question.getItemClass(this.item); var labelClass
  = this.question.getLabelClass(this.item); var locText:
  any = this.item.locText;
  var controlLabelClass = this.question.getControlLabelClass(this.item);

  return (
    <div className={itemClass}>
      <label className={labelClass}>
        <input
          className={this.cssClasses.itemControl}
          id={id}
          type="radio"

```

```

        name={this.question.name + "_" +
        this.question.id}checked={this.isChecked}
        value={this.item.value}
        disabled={this.isDisplayMode ||
        !this.item.isEnabled}
        onChange={this.handleOnChange}
        aria-
        required={this.question.isRequired}
        aria-label={locText.renderedHtml}
        aria-invalid={this.question.errors.length > 0}
        aria-describedby={ this.question.errors.length > 0
            ? this.question.id + "_errors"
            : null
        }
        role="radio"
    />
    <span className={this.cssClasses.materialDecorator}>
        <svg
            className={this.cssClasses.itemDecorator
            } viewBox="-12 -12 24 24"
            >
            <circle r="6" cx="0" cy="0" />
        </svg>
    </span>
    <span className="check" />
    <span className={controlLabelClass} title={locText["koRenderedHtml"]} >
        {itemText}
    </span>
</label>
{otherItem}
</div>
);
}
protected renderOther(cssClasses: any):
    JSX.Element {return (
        <div className="form-group">
            <TestQuestionCommentItem
                question={this.question}
                otherCss={cssClasses.other}
                cssClasses={cssClasses}
                isDisplayMode={this.isDisplayMode}
            />
        </div>
    )
    ;
}
}

ReactQuestionFactory.Instance.registerQuestion("radiogroup", (props)
=> {return React.createElement(TestQuestionRadiogroup, props);
});

ла названий классов.");};

```