

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра комп'ютерних наук

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: : «РОЗРОБКА ЗАСОБІВ ТЕСТОВОГО КОНТРОЛЮ ПРИ  
ВІДДАЛЕНОМУ НАВЧАННІ З ВИКОРИСТАННЯМ ОПТИЧНОГО  
РОЗПІЗНАВАННЯ СИМВОЛІВ»

Виконав: студент 4 курсу, групи КНД–41  
Спеціальності 122 Комп'ютерні науки

Сидоренко М.М.

(прізвище та ініціали)

Керівник Звенигородський О.С.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

## ВСТУП

Тенденція до діджиталізації ярко виражена у сучасному світі, вона притаманна майже всім сферам нашого життя, освіта не виняток. На сьогодні комп'ютери та Інтернет є невід'ємною частиною освітнього процесу. Віддалене навчання є одним з наслідків впливу технологій на навчальний процес. Наразі не обов'язково відвідувати лекції самотужки, потрібен лише доступ до Інтернету та мобільний телефон чи комп'ютер.

Особливо помітна актуальність віддаленого навчання за сьогоднішніх умов, а саме вплив пандемії спричиненої COVID-19. Навчальні заклади вимушені закритись та використовувати альтернативні засоби навчання: лекції по телебаченню чи радіо, відео та аудіо матеріали з Інтернету або віддалене навчання (дистанційне навчання). Таким чином за можна прослуховувати лекції вдома чи в транспорті, а за допомогою засобів дистанційного навчання можливо віддалено поспілкуватись з викладачем або робити практичні завдання.

Існують спеціальні онлайн ресурси для дистанційної освіти, що об'єднують в себе все необхідне для віддаленого навчання: лекції, презентації, текстові/відео/аудіо матеріали, практичні/лабораторні роботи та тестування. Звісно віддалене навчання застосовується не лише в освітніх закладах, існує величезна кількість різних незалежних онлайн курсів на найрізноманітніші теми.

Тут постає питання, як перевірити рівень знань та ступінь засвоєння вивченого матеріалу. Звісно для такого можна використати наприклад практичні завдання, але найчастіше використовуваним засобом для перевірки знань є тестування. Тестування – це найшвидший, простіший та зручніший спосіб перевірити, чого учні навчились. Велике розмаїття типів питань допомагає перевірити знання з різних сторін та можливість використання засвоєного матеріалу у різних ситуаціях.

Розвиток технологій та пристосування їх до потреб у різних галузях можна спостерігати вже сьогодні, а пристосування технологій з однієї галузі до іншої відкриває нові можливості для їх використання. Використання технології оптичного розпізнавання символів для тестування під час віддаленого навчання є одним з таких прикладів. Авжеж, існують засоби для тестування онлайн або на комп'ютері, але ж не всі їх використовують. Або, нажаль, не у всіх є постійний доступ до Інтернету чи персональні комп'ютери. За допомогою подібного засобу тестового контролю потрібно лише фото зробленого тесту для його автоматичної перевірки.

Для розробки подібного проекту потрібно дослідити аналогічні засоби, якщо вони є чи засоби, що застосовується у подібній сфері. Для цього спершу потрібно дослідити подібні наявні засоби, наприклад засоби для тестового контролю, які можливо використовувати під час віддаленого навчання. Виявити, які особливості подібних програм доцільно використати, наприклад, які типи питань можна використати у побідній програмі. Потім необхідно дослідити більш детально саму технологію оптичного розпізнавання символів, бо існує декілька типів подібних технологій наприклад OCR, IWR, ICR, OWR та OMR. Вияснити, які технології можливо використати для спрощення написання та роботи програми. Також необхідно змодельювати роботу програми з декількох сторін, для цього можуть допомогти діаграми нотації IDEF та UML.

Технологія оптичного розпізнавання символів зазвичай використовується для відцифрування документів, а використання її у новій сфері, а саме у сфері освіти досить нове. За допомогою засобу тестового контролю з використанням оптичного розпізнавання символів за умов відданого навчання можна автоматизувати процес перевірки тестів, це пришвидшить та полегшить процес тестового контролю.

# 1 СИСТЕМИ ТЕСТУВАННЯ В УМОВАХ ВІДДАЛЕНОГО НАВЧАННЯ

## 1.1 Вплив віддаленого навчання

Освіта, як одна з найважливіших галузей нашого життя не може стояти на одному місці, а повинна постійно пристосовуватися до нових технологій, впроваджувати інновації та залишатися гнучкою й конкурентоспроможною.

Саме, дистанційна освіта, і є однією з таких інновацій в освіті, особливо, дивлячись на сьогоднішні умови життя. Завдяки такій системі навчання студенти мають можливість вивчати нові матеріали, спілкуватися з іншими студентами, консультуватися з викладачами, а також підтверджувати свій рівень знань у зручний для них час за допомогою домашнього комп'ютера .

На відміну від традиційного способу навчання, завдяки дистанційній освіті всі навчальні заходи виконуються за допомогою спеціально створених систем. З розвитком технологій, такими як смартфони, телебачення та Інтернет, з'явилися нові моделі навчання. Використання сучасних підходів таких, як відео конференції, відеоматеріали, комп'ютерна графіка, презентації, електронні словники чи онлайн тестування стає дуже популярним, і вже не виникає такої потреби невідкладної потреби бути присутнім на лекції фізично. А дистанційне навчання, є формою навчання при якій не має необхідності бути присутнім на заняттях самотужки, достатньо мобільного телефону та доступу до Інтернету. Мінімальні вимоги для успішного дистанційного навчання комп'ютер чи мобільний телефон, також можна придбати веб-камеру для зручності у спілкуванні. Звісно одного обладнання не досить потрібні також і програмні засоби, такі як додатки для відео конференцій наприклад Google Meet чи Zoom або платформи для дистанційного навчання найпоширеніша з яких є Moodle.

Звичайно дистанційна освіта це не завжди вибір, а й інколи необхідність. Не всі учні або студенти мають змогу навчатися через фізичні проблеми або стан хвороби. До них належать:

- Студенти або учні, які проживають у віддалених регіонах або не мають змогу добратися до закладів освіти.
- Інваліди
- Тяжкохворі
- Учні, які часто змінюють місце проживання
- Люди, які хочуть поглибити свої знання навчаючись за кордоном

Звісно «живе» навчання – це добре. Особистий контакт дозволяє викладачам та студентам спілкуватися в традиційній манері, що часто призводить до хорошого взаєморозуміння і взаєминам. Це звичайне явище, і схоже, що подібна форма спілкування набагато простіше. Тож чому дистанційне навчання є таким актуальним? Для цього необхідно розглянути ряд переваг дистанційної освіти:

- 1) Гнучкість – це одна з найголовніших переваг дистанційної освіти. Самостійний вибір необхідних для вивчення матеріалів, тривалості та послідовності їх вивчення;
- 2) Свобода доступу та вибору – ви можете займатись практично скрізь, будь-коли та з будь-яким вчителем;
- 3) Здатність йти в ногу с часом – тобто своєчасне та ефективне оновлення навчальних матеріалів або оновлення чи зміна навчальних курсів. Інформаційні технології постійно оновлюється тому це просто необхідно бути в курсі останніх розробок чи оновлень;
- 4) Навчання у вашому темпі – саме ви вирішуєте з якою швидкістю навчатись, встановлюєте власні пріоритети залежно від ваших обставин та потреб;
- 5) Доступність – незалежно від географічного знаходження та часових обмежень у студента є необмежений доступ до навчальних матеріалів та курсів;

- 6) Мобільність – ефективний зворотній зв'язок між викладачем та студентом є однією з основних вимог для подальшого успіху у навчальному процесі;
- 7) Зменшення витрат – як не дивно також є однією з головних переваг навчання, адже на оплату транспорту йде велика частина наших витрат. Звісно не лише це, завдяки діджиталізації сертифікати з онлайн курсів стають все більш поширенішими, й існують навіть акредитовані університети, що навчають лише онлайн.

Зазвичай у всього є протилежний бік, дистанційна освіта не є виключенням. Саме тому просто необхідно привести недоліки дистанційної освіти:

- 1) Відсутність постійного спілкування між викладачем та студентом – тобто моменти пов'язані з індивідуальним підходом у такому разі виключені. Суттєвим мінусом є також коли поряд немає людини, яка б могла привернути увагу до вивчення тієї чи іншої проблеми;
- 2) При дистанційному навчанні зазвичай використовується лише письмова форма спілкування. Для деяких людей неможливість висловити свої знання чи думки у усній формі може бути суттєвим мінусом;
- 3) Необхідність самодисципліни є обов'язковим при дистанційному навчанні. Адже за умов дистанційної освіти саме студенти влаштовують свій розпорядок дня, а оскільки зазвичай таке навчання відбувається вдома дуже просто відволіктися на інші фактори чи потреби;
- 4) Не завжди навчальні програми та курси є достатньо розробленими через те, що кваліфіковані спеціалісти здатні розробити необхідні навчальні посібники;
- 5) Необхідність постійного доступу до джерел інформації. Як не сумно, але не у всіх, хто хоче навчатися, є доступ до необхідного технічного обладнання, особливо це можна помітити у віддалених регіонах чи країнах з низьким достатком.

Популярність дистанційної освіти обумовлена перевагами, які вона надає студентам. Основним її недоліком є зворотній бік переваги – зручність . Комфорт та свобода, що надає дистанційне навчання вимагає більшої самодисципліни ніж

при традиційній формі навчання. Для когось дистанційна освіта свого роду змагання з собою, перевірка сили волі. Інші досить довго адаптуються до такої форми навчання. Деякі почувають себе набагато комфортніше й одразу ж розуміють, що і як потрібно робити.

Наразі одним з найяскравіших прикладів дистанційного навчання є освітні платформи, що надають можливість людям зі всього світу навчатися у зручній та зручному місці та підвищувати свою кваліфікацію. На сьогодні таких платформ сотні. Основними представниками таких платформ є:

- Google Classroom
- EdApp
- Udemy
- WizIQ
- LinkedIn Learning
- Moodle

У всіх розвинених країнах навчання спрямоване на самоосвіту, здатність та можливість самостійно отримувати необхідну інформацію, виявлення проблем та шукати шляхи їх вирішення, вміти критично аналізувати отримані знання та застосовувати їх для вирішення проблем. Ідеальна система навчання:

- Формує бажання вчитися
- Надає кожному студенту розвиватись індивідуально
- Адаптує засоби для навчання
- Дає кожному студенту можливість навчатись за індивідуальним графіком
- Постійно оцінює результати навчання. [7]

## 1.2 Вплив пандемії COVID-19 на освіту

Дивлячись на ситуацію у світі можливо побачити, що світ все більше пристосовується до цифрових технологій. Тенденція до діджиталізації все більш виражена у світі, її можливо спостерігати у всіх сферах нашого життя будь то наше дозвілля, робота, медицина чи освіта. Сьогодні ми можемо замовити їжу не виходячи з дому, працювати прямо з ліжка, а що б відвідати лікаря не потрібно йти до лікарні. Галузь освіти також пристосовується до нових технологій.

Особливо виражена діджиталізації не лише в освіті, а й в світі, на сьогоднішній день за умов глобальної пандемії викликаної COVID-19. Оскільки весь світ знаходиться під домашнім арештом, ми просто вимушені змінити наші звички та пристосуватися до умов життя під час світового локдауну.

Пандемія COVID-19 не зупинилась на національних кордонах. Це зачепило людей незалежно від національності, доходу чи рівня освіти. Але це не стосується наслідків пандемії, які найбільше вразили най уразливі місця нашого життя. Освіта не є винятком. Пандемія COVID-19 спричинила найбільші в історії порушення системи освіти, вразивши майже 1,6 мільярдів учнів у понад 190 країнах на всіх континентах світу [1]. Хоча держава доклала максимум зусиль для продовження безперервного навчання протягом цього періоду, дітям та студентам довелось покладатись на власні ресурси, щоб продовжувати навчання дистанційно через Інтернет чи телебачення. Ця криза виявила численні нерівності та порушення в нашій системі освіти – від неможливості доступу до Інтернету та комп'ютерів, необхідних для навчання в умовах локдауну до допоміжних засобів та середовища необхідних для зосередження уваги на навчанні. Вчителі також повинні були адаптуватись до нових умов роботи та способів викладання, до яких вони не були підготовлені.

Заклади вищої освіти були більш підготовлені до роботи в умов локдауну. Більша частина закладів вищої освіти мають свої власні засоби для дистанційного навчання. Але це не змінює того факту, що заклади освіти були вимушені вийти



за рамки привичного режиму роботи, зачинити свої двері для студентів, так з'явилась нова норма навчання, як дистанційна освіта.

### **1.3 Розвиток дистанційного навчання**

Дистанційне навчання – це не нова концепція. Деякі джерела відслідковують історію дистанційного навчання ще до 1700-х років. Коли в 1728 році викладач стенографії Калєб Філлїпс купив оголошення в «Boston Gazette», обіцяючи, що студенти "маючи щотижневі розсилки декількох уроків, можуть мати такі ж гарні інструкції, як і ті, що живуть у Бостоні" [2]. Незабаром після цього часу дистанційна освіта практикувалась за допомогою методу, який також відомий як заочна освіта.

Інші стверджують, що початок розвитку дистанційного навчання відбувся в 1840 році, коли англійській педагог, сер Ісаак Пітман, навчав за допомогою листування. Пітман надсилав учням текст на листівках, а учні надсиляли йому свої завдання назад. Заочні курси продовжували нарощуватись, а вже в 1858 році Лондонський університет став першим коледжем, який запропонував ступінь дистанційного навчання [2]. У 1873 році Анна Тікнор в Бостоні, штат Массачусетс, заснувала офіційну заочну освітню програму під назвою «Товариство заохочення домашнього навчання» [2]. А вже у 1888 році у Пенсільванії була заснована найбільша приватна комерційна школа «Міжнародна заочна школа» для забезпечення навчання іммігрантів-шахтарів, які прагнули стати державними інспекторами шахт або бригадирами. У 1894 р. було зараховано 2500 нових студентів, а в 1895 р. – 72 000 студентів. До 1906 року загальна кількість учнів у Міжнародній заочній школі досягло 900 000 [3].

У 1911 році університет Квінсленда в Австралії заснував відділ заочного навчання, який спирався на поштову систему Австралії [2]. Однак складність цих методів полягала в тому, що вони були односторонніми, тобто студенти не мали

можливості взаємодіяти з безпосередньо з викладачами, задавати їм запитання чи спілкуватись з іншими студентами.

Із часом технологічні досягнення відігравали вирішальну роль у дистанційному навчанні. Запровадження радіо дозволило університетам транслювати студентам навчальні курси. В 1922 році Пенсільванський державний коледж став першим коледжем, який транслював курси через радіомережі [2]. Приблизно через десять років університет штату Айова став першим університетом, де телевізор використовувався, як інструмент для дистанційного навчання.

Такі впровадження та інновації тривали і в середині 20 століття. Зазначено, що в 1956 році громадська телевізійна станція Чикаго WTTW у партнерстві з місцевим органами освіти надавав кредитування для телевізійних курсів, так протягом 5 років навчалось понад 5000 студентів. Нью-Йоркський університет і телемережа CBS запускають свої курси «Sunrise Semester», яка також надавала кредитування для телевізійних курсів, а вже в 1962 році газета «New York Times» публікує статтю, в якій вітає домогосподарку з здобуттям ступеню бакалавра з цих самих курсів [4]. Важко повірити, але завдяки ось таким крокам, як телевізійним курси багато хто досяг великих здобутків. Навіть звичайна домогосподарка змогла досягти значних успіхів, маючи змогу вивчити знання університетського рівня, не виходячи з дому.

Технології продовжили розвиватись, а разом с цим і можливості дистанційного навчання. Вже в 1980-х згідно з «Основами дистанційної освіти», була досягнута революційна можливість викладати або навчатись віч-на-віч на відстані завдяки досягненням в електроніці в 80-х. Це пов'язано з відкриттям більш швидких мікросхем там впровадженням Інтернету, що спричинило справжню революцію не лише в галузі освіти [5]. Це дозволило студентам спілкуватись безпосередньо зі своїми викладачами, що надало можливість інтерактивного навчання, на місце пасивного навчання. А в 1984 році Національний технологічний університет, штат Колорадо, створив перший акредитований «віртуальний» університет, який проводив заняття для працівників за допомогою телевізора, за фінансової підтримки таких компаній, як IBM, Motorola та HP. Так навіть великі компанії

почали долучатись до розвитку дистанційного навчання, дозволяючи своїм співробітникам мати долучитись до віртуального навчання.

Надалі вже персональні комп'ютери з доступом до Інтернету прийшли на заміну телевизорам став наступним великим досягненням, що здійснив революцію в галузі дистанційного навчання. У 1989 році Університет Фенікса став першим навчальним закладом, який відкрив повністю онлайн-курс, який пропонував ступень бакалавра та магістра. А у 1996 році Глен Джонс та Бернан Лускін заснували Міжнародний університет Джонса, який став першим акредитованим університетом, що був заснованим на базі Інтернету [2].

Десмонд Кіган писав: «Наприкінці 1990-х дистанційна освіта стала цінною складовою багатьох систем освіти і зарекомендувала себе в областях, де традиційні школи, коледжі та університети мають труднощі із задоволенням попиту» [5]. Завдяки появі мобільних пристроїв, появі відео конференцій та віртуальних класів починаючи з 1990-х років дистанційне навчання почало бурхливий розвиток.

До 2006 року 89% державних та 60% приватних установ США пропонували заняття в Інтернеті[4]. В одній із статей Forbes зазначається: «Онлайн навчання виходить за межі примітивного, модель викладення один до багатьох стала соціальним, спільним, персональним та інтерактивним досвідом, який генерує дві потужні взаємовигідні причини успіху: по-перше, довготермінове бажання вчитись, покращувати своє становище у світі; а по-друге має вирішальне значення – задоволення від участі у навчальному досвіді, який постійно захоплює, приносить користь та створює цінне відчуття соціального зв'язку»[6].

На щастя, дистанційне навчання вийшло за межі одностороннього спілкування. Сьогодні дистанційне навчання називають також онлайн-освітою, воно стало невід'ємною частиною життя будь-якого студента, бо більшість студентів або учнів хоча б раз відвідували або навчалися за онлайн-уроками. На сьогодні ми можемо спостерігати величезну кількість ресурсів для онлайн-освіти, будь-то позакласні курси, різні можливості для спілкування на відстані чи можливість перевірити ваш рівень знань не виходячи з дому.

## 1.4 Засоби тестового контролю

Як вже згадувалось раніше, діджиталізації стала невідворотнім трендом, а персональні комп'ютери та мобільні пристрої стали невід'ємною частиною нашого життя. На час пандемії спричиненої COVID-19, навчальні заклади були вимушені припинити навчання в звичній для нас формі та перейти на дистанційну форму навчання. З'явилась величезна кількість програмних засобів з метою продовжити наше навчання дома такі, як відео конференції, відеоматеріали та презентації, це стало для нас нормою.

Дистанційна форма навчання надала нам багато можливосте та ресурсів для подальшого навчання. Але самого навчання не досить, адже нам постійно потрібно перевіряти рівень знань, щоб дізнатись чи є навчання ефективним, а для викладачів контролювати навчальний процес. Тому не дивно, що наразі програмні продукти для електронного та комп'ютерного тестування стають все більш актуальними у студентському житті.

За своєю об'єктивністю, можливостями та швидкістю комп'ютерне тестування перевершує всі інші форми контролю знань. У сучасній системі освіти комп'ютерне тестування виконує такі основні взаємопов'язані функції: діагностичну, навчальну та виховну. Діагностична функція має на меті виявити рівень підготовки студентів. Навчальна функція тестування спонукає студентів до засвоєння навчального матеріалу. Виховна функція проявляється в частоті та неминучості тестового контролю [8]. Це дисциплінує та спрямовує студентів, допомагає у виявленні та усуненні прогалин у знаннях, а також формує бажання розвивати свої навички.

На сьогодні з розвитком технологій постійно з'являються різні програмні продукти для електронного чи комп'ютерного тестування. Внаслідок постійного збільшення розмаїття таких комплексів програмних засобів постало питання вибору програмного забезпечення або платформи для тестування, яка буде одночасно зручна, як для викладача, так і для студентів.

Тож, хотілося б розглянути декілька систем та онлайн платформ призначених для проведення тестування та визначити їх можливості.

**ADTester** – це пакет програм, призначений лише для проведення тестування. Дозволяє створювати та редагувати тесту будь-якого рівня складності докладав мінімум зусиль. Має максимально простий інтерфейс та велику кількість параметрів, що дозволить легко налаштувати процес сам під ваші потреби. Має два режими для тестування: контроль та навчання. В режимі навчання при виборі неправильної відповіді можливо подивитись підказки по даному питанні. В режимі контролю вже ведеться статистика проходження тесту[9].

Існує п'ять типів запитань:

1. Одиночний вибір;

Передбачає можливість вибору лише одної відповіді з переліку відповідей

2. Множинний вибір;

Передбачає собою можливість обрати декілька відповідей з списку запропонованих. До речі відповідь зараховується лише в тому випадку, якщо було обрано всі правильні відповіді.

3. Ввід відповіді;

Передбачає, що користувач повинен самостійно ввести відповідь

4. Відповідність;

Передбачає, що користувач повинен встановити відповідності між питаннями та запропонованими відповідями.

5. Порядок;

Передбачує, що користувач повинен впорядкувати запропоновані варіанти відповідей по заданим критеріям.

**ClassMarker** – це онлайн платформа, призначена для розробки тестів. Передбачає собою можливість використання для бізнес та начальних тестів. Надає можливості створення опитувань та тестів з обмеженим часом, загальним та обмеженим доступом до тестів, та робити питання які можуть включати в себе зображення, аудіо, відео, документи вбудовані у запитання чи варіанти відповідей. Тести можуть включати в себе власні запитання чи можна обрати

запитання з наданих в банку питань. Є можливості, як платного, так і безкоштовного користування[10].

Існує шість видів запитань:

### 1. Множинний вибір;

Надає можливість обрати одну або декілька правильних відповідей з наданого списку. Приклад представлено на рисунку 1.1.

Multiple Choice

A) Answer 1

B) Answer 2

C) Answer 3

D) Answer 4

Рисунок 1.1 – Множинний вибір представлений на платформі ClassMarker

### 2. Вірно/Невірно

Проста форма питання, що надає лише два варіанти відповідей «Вірно» або «Невірно». Приклад представлено на рисунку 1.2.

True/False

A) True

B) False

Рисунок 1.2 – Питання типу Вірно/Невірно представлено на платформі ClassMarker

### 3. Співставлення;

Питання метою, якого є спів ставити правильні відповіді в залежності від питання. Відповіді надаються у вигляді списку, відповідей може бути більше ніж питань. Приклад представлено на рисунку 1.3.

Match the options below:

Question 1	Select
Question 2	Select
Question 3	Select

Рисунок 1.3 – Питання типу Співставлення представлено на платформі ClassMarker

#### 4. Граматичне;

Завдання полягає в тому, що на екрані відображається певний текст, частіше за все коротке речення. Для того щоб правильно відповісти необхідно записати правильний варіант тексту. Приклад представлено на рисунку 1.4.

Correct the sentence.  
This Grammar type question Incorrect version

This Grammar type answer Correct version

Рисунок 1.4 – Граматичне питання представлено на платформі ClassMarker

#### 5. Вільний ввід;

Необхідно записати відповідь на певне питання власноруч. Приклад представлено на рисунку 1.5.

Free Text type Question  
Your Answer is:

Рисунок 1.5 – Питання типу вільного вводу представлено на платформі ClassMarker

#### 6. Есе;

В якості відповіді на задану тему потрібно написати відповідь у форматі есе. Приклад представлено на рисунку 1.6.

Essay  
Write essay:

This  
IS  
Essay  
type  
answer

Show word count ▾

Рисунок 1.6 – Питання типу Есе представлено на платформі ClassMarker

**EasyTestMaker** – це автоматизований онлайн інструмент для проведення тестів, який допомагає створювати, генерувати та керувати автоматизованими тестами. Під час створення онлайн-тесту дозволяє додавати інструкції, додавати

оцінки до кожного питання та розділяти тест на декілька розділів. Загалом простий у користування інструмент для розробки тестів. EasyTestMaker можливо використовувати безкоштовно для обмеженої кількості до 25 питань, для подальшого додавання питань необхідно придбати абонемент [11]. Особливості даного програмного продукту:

- Різні типи запитань;

Тестова платформа підтримує різні типи запитань;

- Легко підтримує розширені тести;

Платформа легко підтримує варіант питань, якщо вам необхідно створити тести з елементами математичних рівнянь або з використанням наукових символів;

- Безкоштовні шаблони;

В базі шаблонів платформи знаходиться безліч попередньо розроблених тестів, які можна додати в залежності рів потреб користувача;

- Додавання власних питань до шаблонів;

Можливо підвищити ефективність створення тестів, об'єднавши вже існуючі шаблони разом з вашим банком запитань;

- Повне налаштування тестів;

Є можливість змінити майже всі аспекти тестового дизайну, включаючи розмір тексту, шрифти та інше;

- Збір інформації про результати студентів;

Використовуючи прості питання, доступні у вільному доступі платформи викладачі можуть легко переглядати інформацію про студентів;

- Необмежена кількість запитань;

Можливо створити необмежену кількість запитань та тестів

- Розумне оцінювання;

Можливо присвоїти певну оцінку кожному питанню або правильній відповіді. Також можливо відображати оцінку після кожного питання, через певну кількість питань або після завершення тесту;



- Планування та встановлення часу для тестування;

Є можливість запланувати тести та надіслати повідомлення за допомогою SMS, електронної пошти чи посиланням;

- Встановлення обмежень часу для тестування;

Є можливість встановити обмеження на час проходження певного тесту;

- Підтримка мобільних пристроїв;

Всі онлайн тестування оптимізовані для мобільних пристроїв. Це надає можливість проходити тести навіть на вулиці або в транспорті;

- Швидка аналітика;

Генератор тестів надає інтегровані аналітичні інструменти. Ви можете використовувати ці інструменти для відслідковування прогресу тесту для окремих людей або всіх студентів у режимі реального часу;

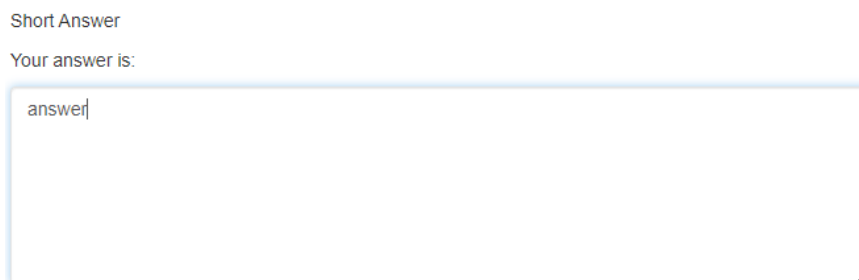
- Анонімні відповіді;

Надає можливість відповідати анонімно. Це допоможе зняти напругу для учнів, які невпевнені у власних силах, та надає можливість проходження тесту декілька разів для покращення розуміння матеріалу.

Дана платформа надає шість видів запитань:

1. Коротка відповідь;

Вимагає на певне питання дати відповідь у формі слова або фрази, що повинна в точності відповідати правильній відповіді. Приклад представлено на рисунку 1.7.



Short Answer

Your answer is:

Рисунок 1.7 – Питання типу Коротка відповідь представлено на платформі

EasyTestMaker

2. Множинний вибір з декількома правильними відповідями;

Надає можливість обрати декілька правильних відповідей з переліку відповідей. Приклад представлено на рисунку 1.8.

Multiple Choice  
Your answer is:  
 1  
 2  
 3  
 4

Рисунок 1.8 – Питання типу Множинний вибір з декількома правильними відповідями представлено на платформі EasyTestMaker

3. Множинний вибір з однією правильною відповіддю;

Надає можливість обрати лише одну правильну відповідь з переліку відповідей. Приклад представлено на рисунку 1.9.

Multiple Choice  
Your answer is:  
 1  
 2  
 3  
 4

Рисунок 1.9 – Питання типу Множинний вибір з однією правильною відповіддю представлено на платформі EasyTestMaker

4. Вірно/Невірно;

Питання на яке є лише два варіанти відповіді Вірно або Невірно. Приклад представлено на рисунку 1.10.

True/False  
1+1=2?  
 True  
 False

Рисунок 1.10 – Питання типу Вірно/Невірно представлено на платформі EasyTestMaker

5. Порядок ранжування;

У такому типі запитання необхідно обрати правильні відповіді до запитань відповіді можливо обрати з списку. Приклад представлено на рисунку 1.11.

5) <input type="text" value="b"/> Question 1	a) Answer 2
6) <input type="text" value="c"/> Question 2	✓ b) Answer 1
7) <input type="text" value=""/> Question 3	✓ c) Answer 4
8) <input type="text" value="d"/> Question 4	✓ d) Answer 3

Рисунок 1.11 – Питання типу Порядок ранжування представлено на платформі EasyTestMaker

#### 6. Заповнення пропусків;

Необхідно заповнити пропуски у тексті, таких пропусків може бути декілька, не залежить від регістру. Приклад представлено на рисунку 1.12.

Fill in the blank

This is  answer

Рисунок 1.12 – Питання типу Заповнення пропусків представлено на платформі EasyTestMaker

**Moodle** – не можна сказати, що Moodle просто платформа для тестового контролю або розробки тестів. Швидше це система управління навчанням, яка може повністю забезпечити всі аспекти для дистанційного навчання. Розроблена ця система Мартіном Дугіамасом в 2002 році, основною метою створення було зробити свій внесок у сприйняття системи електронного навчання та сприяти розвитку онлайн освіти[12].

Наразі можна сказати, що Moodle одна з найбільш поширених, якщо не найбільш поширена платформа для дистанційної та онлайн освіти. Платформа Moodle забезпечує потужний набір інструментів, орієнтований на учнів та середовище для спільного навчання. Moodle дуже простий та гнучкий у використанні, як для викладачів, так і для студентів, сумісна з мобільними пристроями та перекладена на більш ніж 120 мов світ. Крім того, платформа дозволяє створювати свої шаблони та дозволяє інтегрувати зовнішні програми для

розширення можливостей системи, а також має свою велику базу даних шаблонів, які є у вільному доступі для будь-якого користувача.

Повернемося до можливостей тестового контролю платформи Moodle, вона напевно надає найбільший спектр можливих типів запитань для перевірки знань користувачів. Розглянемо ці види запитань:

1) Обчислювальний або просте обчислювання;

Обчислювальний тип запитань потребує від користувача обрахувати числову відповідь з урахуванням конкретних змінних формули. Це такі запитання, як обрахування площі поверхні геометричних фігур чи переведення метричній систем. Числа для такого виду питань можуть генеруватися випадковим чином при кожній спробі чи можливо встановити свої. Також при врахуванні відповіді можливо встановити похибку при якій відповідь буде вважатись правильною. Питання типу простого обчислювання – це найпростіша версія запитання типу обчислювання. Приклад представлено на рисунку 1.13.

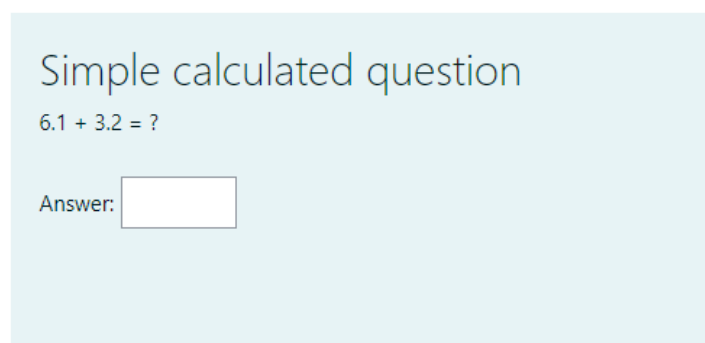


Рисунок 1.13 – Питання типу Обчислювання представлено на платформі Moodle

2) Обчислювальний множинний вибір;

Такий тип запитань вимагає від студентів вибору правильної відповіді з списку наданих відповідей. Відповіді обраховуються на базі випадково сформованого набору значень для змінних у формулі. Як і в обчислювальному типі запитань є можливість власноруч встановити певні змінні чи задати їх випадковим чином. Це питання наприклад для обрахування периметру фігури завдяки декільком випадково сформованих значень. Приклад представлено на рисунку 1.14.

Calculated multichoice

x = Периметр = ?

**При a = 6**

**При b = 6.1**

**При c = 1.9**

a.  $x = 6 / 0.6.1 - 1.9$

b.  $x = 6 + 6.1 + 1.9$

c.  $x = 6 * 6.1 + 1.9$

d.  $x = 6 + 6.1 * 1.9$

e.  $x = 6 + 6.1 / 1.9$

Рисунок 1.14 – Питання типу Обчислювальний множинний вибір представлено на платформі Moodle

### 3) Перетягування зображення;

В питанні такого типу потрібно перетягнути зображення або текст на задані місця на фонове зображення запитання. Перетягнути цільові об'єкти можна лише на задані області. Такі питання можна використати для схем або карт. Приклад представлено на рисунку 1.15.

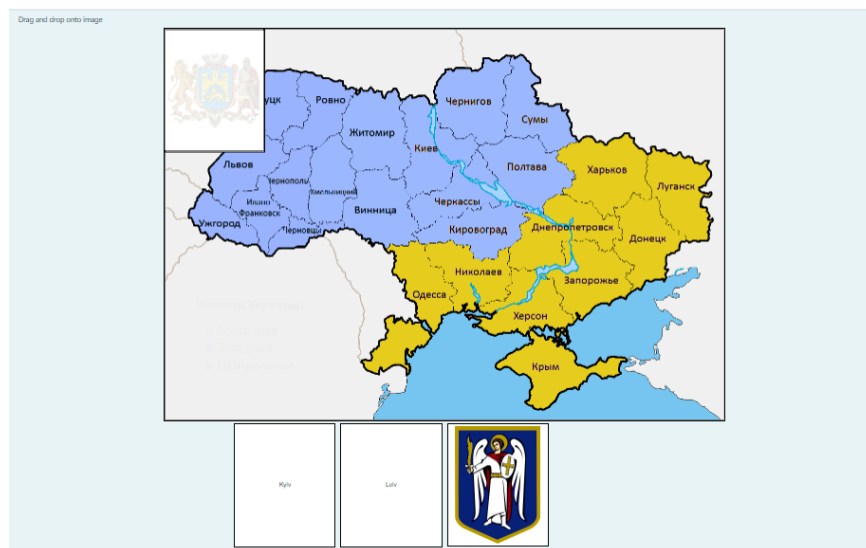


Рисунок 1.15 – Питання типу Перетягування зображення представлено на платформі Moodle

### 4) Перетягування маркерів;

Такий тип запитань дещо схожий з питаннями типу перетягування зображень. Таке питання вимагає від студента перегнути маркер на фонове зображення, відмінністю також є те, що на фоновому зображенні не відображені області для можливих відповідей, тобто студент сам має вирішити де має знаходитись те чи інший маркер. Наприклад запитання типу перетягнути назви країн або міст на карту. Приклад представлено на рисунку 1.16.



Рисунок 1.16 – Питання типу Перетягування маркерів представлено на платформі Moodle

#### 5) Перетягування тексту;

Такий тип запитань потребує перетягнути текстові поля в текст запитань. Використовується для заповнення пустого запитання, щоб зробити речення вірним або відповісти визначеним термінам. Приклад представлено на рисунку 1.17.

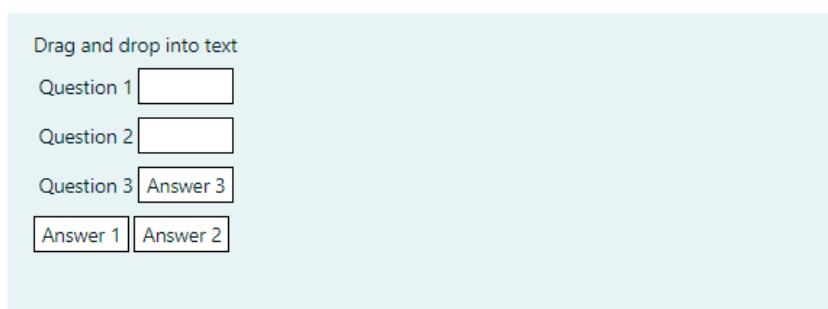


Рисунок 1.17 – Питання типу Перетягування тексту представлено на платформі Moodle

### 6) Вкладене запитання;

Такі питання є дуже гнучкими мають декілька варіантів вкладених запитань таких, як Коротка відповідь, Множинний вибір, Числова відповідь чи Співставлення. Приклад представлено на рисунку 1.18.

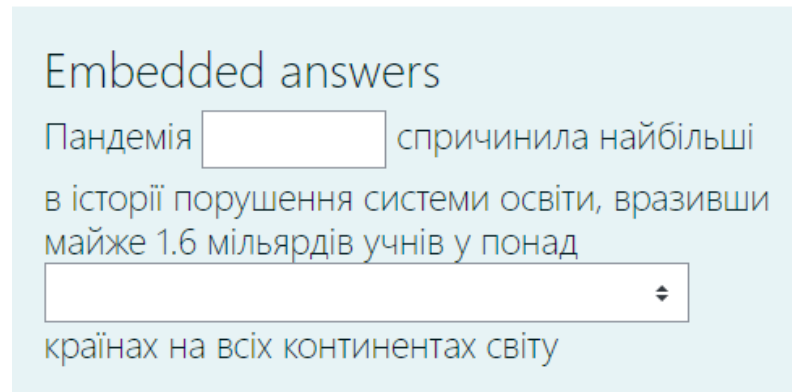


Рисунок 1.18 – Питання типу Вкладене запитання представлено на платформі Moodle

### 7) Есе;

У відповідь на запитання або надану тему, пропонується записати відповідь у вигляді есе. Така відповідь оцінюється вручну творцем тесту. Приклад представлено на рисунку 1.19.

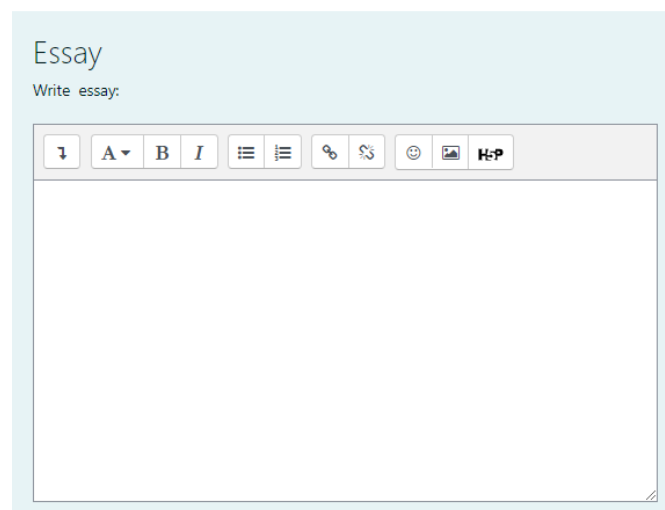


Рисунок 1.19 – Питання типу Есе представлено на платформі Moodle

### 8) Співставлення;

Надається ряд запитань або термінів на які студент має знайти правильні відповіді з варіантів наданих у списку. Кількість відповідей може бути більша, ніж кількість запитань. Приклад представлено на рисунку 1.20.

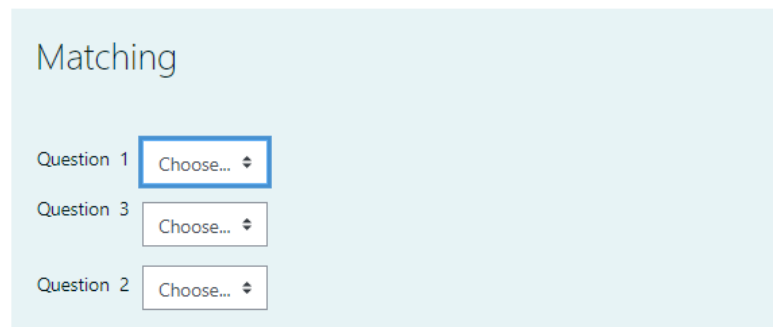


Рисунок 1.20 – Питання типу Співставлення представлено на платформі Moodle

### 9) Множинний вибір;

Це один з типів запитань множинного вибору. Для такого запитання надано список відповідей, можна обрати лише одну відповідь. Приклад представлено на рисунку 1.21.

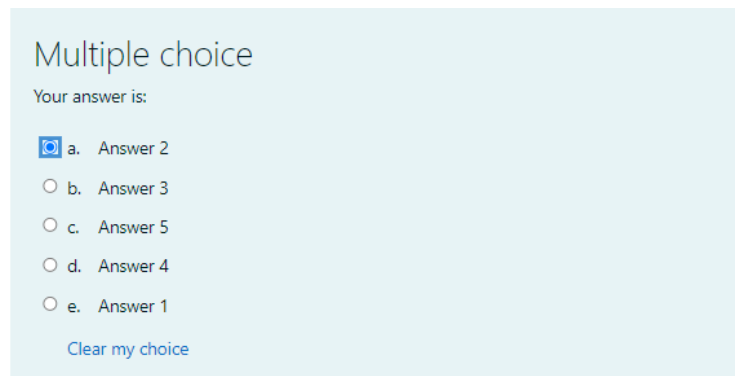


Рисунок 1.21 – Питання типу Множинний вибір представлено на платформі Moodle

### 10) Множинний вибір з декількома правильними відповідями;

Другий тип запитань з типу множинного вибору, як зрозуміло з назви різниця між першим типом в тому, що в такому запитанні є декілька правильних варіантів відповідей. Приклад представлено на рисунку 1.22.

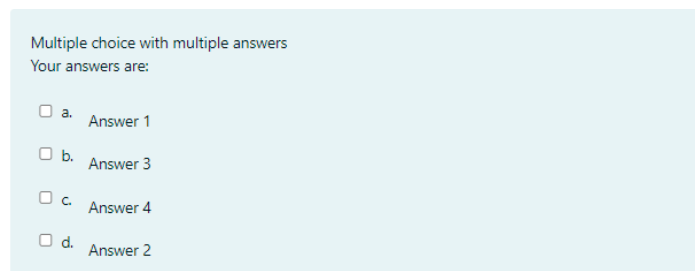
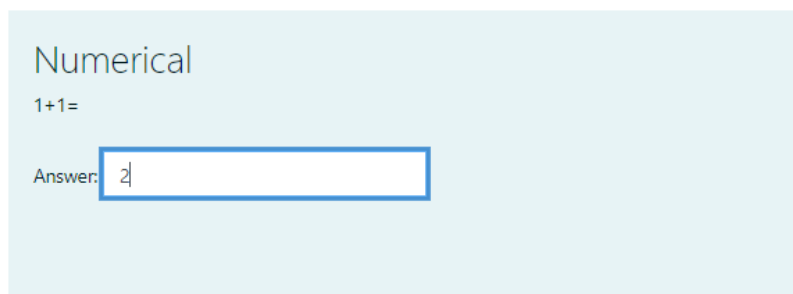


Рисунок 1.22 – Питання типу Множинний вибір з декількома правильними відповідями представлено на платформі Moodle

### 11) Числовий;



З точки зору студента такий тип запитання нічим не відрізняється від запитань типу Коротка відповідь. Але різниця полягає в тому, що відповідь може бути лише у вигляді числа. Для запитань такого типу допустима похибка встановлена творцем тесту, це дозволяє забезпечити діапазон відповідей у певних межах. Приклад представлено на рисунку 1.23.



Numerical

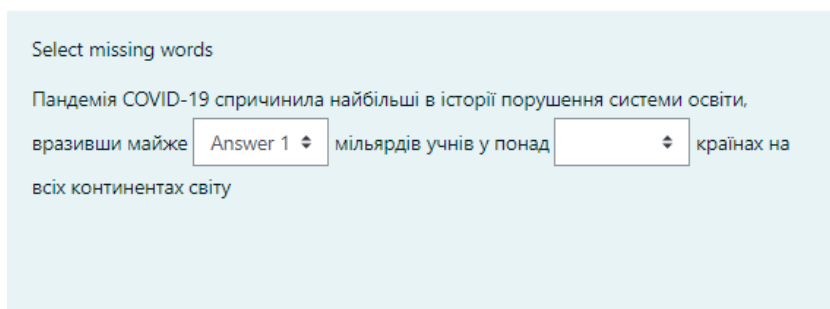
1+1=

Answer:

Рисунок 1.23 – Питання типу Числове представлено на платформі Moodle

12) Вибір відсутніх слів;

Запитання такого типу містять випадваючі списки з можливими варіантами відповідей вбудовані у певний текст. Студент має обрати правильне слово чи фразу з наданих відповідно до тексту. Таких варіантів може бути більше, ніж необхідно. Приклад представлено на рисунку 1.24.



Select missing words

Пандемія COVID-19 спричинила найбільші в історії порушення системи освіти, вразивши майже  мільярдів учнів у понад  країнах на всіх континентах світу

Рисунок 1.24 – Питання типу Вибір відсутніх слів представлено на платформі Moodle

13) Коротка відповідь;

Запитання типу Коротка відповідь потребує від студента набору слова чи фрази в якості відповіді до запитання. Відповіді можуть бути не чутливими до регістру, а також може мати декілька варіантів відповідей на питання. Звісно ваша відповідь має бути точно такою же, як і правильна відповідь, за виключенням регістрів. Приклад представлено на рисунку 1.25.

Short answer  
Your answer is:  
Answer:

Рисунок 1.25 – Питання типу Коротка відповідь представлено на платформі Moodle

14) Вірно/Невірно;

В якості відповіді на запитання дається лише два варіанти Вірно та Невірно. Відповіді потрібно відповідно до контексту запитання. Приклад представлено на рисунку 1.26.

True/False  
1 + 1 = 2?  
Select one:  
 True  
 False

Рисунок 1.26 – Питання типу Вірно/Невірно представлено на платформі Moodle

15) Опис;

Це не тип запитання у звичному вигляді, швидше це спосіб відображення додаткової інформації до тесту чи групи завдань. Приклад представлено на рисунку 1.27.

Description  
Це не тип запитання, швидше це спосіб відображення додаткової інформації до тесту чи групи завдань.

Рисунок 1.27 – Питання типу Опис представлено на платформі Moodle

## 1.5 Мета тестового контролю

Тестування завжди було невід'ємною частинною навчального процесу. Це швидкий та простий процес для виявлення вмінь, навичок та рівню знань учнів. Особливо з зростанням використання технологій в навчальному процесі тестування стало все більш поширеним способом перевірки знань. Процес електронного чи комп'ютерного не потребує особистої перевірки викладачем, що дає можливість швидко дізнатись результат. Звісно електронне тестування також можливо використовувати для самоперевірок, що лише збільшило якість навчання. У процесу тестування є також й інші цілі, ніж перевірки власних здобутків.

Тестування часто викликає негативні емоції у школярів та студентів, бо вони розглядають тестування, як певне «зло». Але ми також маємо розуміти, що це «зло» є вимушеним. Саме завдяки регулярному тестуванню, проведенні невеличких тестів з певною періодичністю чи тестування після великого розділу дозволяє викладачам зрозуміти, як учні сприймають інформацію. Завдяки ось таким регулярним перевіркам вчителі дізнаються потреби учнів та коригують свої навчальні програми, аби учні краще засвоювали матеріал.

Процес тестування також дозволяє нам самим виявити свої слабкості та сильні сторони. Звісно не всі учні розуміють навчальні матеріали однаково, декому певна тема більш проста для розуміння, інша дається більш тяжко. Регулярні перевірки дозволяють вже нам самим коригувати наш навчальний процес, незалежно від навчальної програми.

Можливо це не помітно, але тести вчать нас, як застосовувати здобуті знань до реальних ситуацій. Тестування допомагає ефективніше засвоїти різні типи інформації. Навчання – це більше, ніж заучування формул чи фактів. Учні повинні розуміти, як використовувати вивчену інформацію у реальних життєвих ситуаціях. Саме тестування дозволяє змодельовати для нас певні середовища чи

ситуації, при подальшому вже маючи досвід та знаючи наші помилки ми можемо уникнути подальших помилок.

Також тестування допомагає краще сприймати початковий процес. Навчаючись ми постійно збираємо величезну кількість інформації і не завжди ця інформація є корисною. Наприклад прочитавши книгу або певний розділ ми дізнались великий обсяг інформації, і ми просто не можемо зрозуміти на чому слід сконцентруватись або що є насправді для нас важливим. Вже процес тестування дозволяє нам зосередити нашу увагу на певній найбільш важливій частині інформації, це дозволяє нам краще засвоювати новий матеріал, бо якусь невелику, але важливу частину інформації набагато простіше сприймати, ніж великий обсяг не досить важливої, це також пришвидшує процес навчання.

Звісно не можливо сказати, що тестування це суцільні плюси. Дехто вважає, що тестування забирає у нас дорогоцінний час на інші критичні моменти навчання, як лекції, дискусії чи демонстрації. До певної міри це й справді так, але ж не можемо стверджувати, що такі заходи є більш важливими, ніж тестування. Інші стверджують, що тестування це поверхневий процес, воно дає нам поверхневу повноту знань, ніби учні просто завчають інформацію, але насправді не розуміють вивчений матеріал. Звідси також можна знайти й інший недолік. Ми розуміємо, що тести також складають люди, саме неправильна побудова таких тестів може призвести до помилкового розуміння матеріалу та хибного сприйняття рівня підготовки.

Незважаючи на це тестування залишається найбільш простим та розповсюдженим способом перевірки знань, який надалі буде використовуватись і не лише в галузі освіти.

## 2 МОДЕЛЬ СИСТЕМИ ТЕСТУВАННЯ

### 2.1 Постановка задачі моделювання

Надалі в цьому розділі розглянутьсся задачі для створення засобу тестового контролю при віддаленому навчанні з використанням оптичного розпізнавання символів, моделювання програмного засобу та методи застосовані чи які необхідно знати при розробці даного програмного продукту.

Головним завданням даного програмного засобу є полегшення та пришвидшення перевірки тестів, завдяки автоматизації самого процесу перевірки. Перш ніж, приступити до самого процесу розробки необхідно змоделювати процеси роботи, а також визначити методики, засоби та матеріали, які будуть використовуватись при розробці та безпосередньо при роботі засобу для тестового контролю. Процес моделювання роботи програмного засобу буде детальніше розглянуто окремо в цьому розділі.

Для початку, як зрозуміло з назви проекту головною метою є тестовий контроль. Отже, необхідно визначити тип тесту, який буде перевіряти програмний продукт, детальніше цей момент буде розглянуто у наступному підрозділі.

Сам процес перевірки спроститься до такого стану, що користувач, тобто викладач повинен лише завантажити фото файл зробленого тесту, а на виході отримає результат, тобто оцінку перевіреного тесту. Такий процес становиться можливим завдяки оптичному розпізнанню символів, роз'яснення процесу роботи даної методики буде розглянуто у далі розділі.

Взагалі, наразі існує велика кількість засобів для електронного та комп'ютерного тестування. Але при розробці даного проекту були враховані сьогоденні потреби, а саме перевірка тестів при віддаленому(дистанційному) навчанні. Відсутністю технічних засобів таких, як персональні комп'ютери чи вільний доступ до Інтернету в учнів у віддалених від міст місцях, селах чи селищах, також привернула увагу при розробці особливостей даної системи.

Звичайно не лише це стало причиною розробки подібного проекту, адже велика кількість викладачів у школах чи у закладах вищої освіти розробляють тести побідного типу. Тому такий формат перевірки тестів є досить актуальним.

Тобто програмний продукт повинен мати подальші особливості:

- Можливість виконання тестового завдання без доступу до Інтернету
- Можливість виконати тест на мобільному телефоні
- Легкість під час процесу тестування
- Автоматизація процесу тестування
- Можливість додавання, редагування чи видалення змісту тестів

## **2.2 Вимоги до розробки тесту**

По-перше при розробці тесту потрібно встановити стандарти, як тест буде розроблений. Отже, обраний для тип тестів для даного проекту є друкованим тестом, тобто користувач працює з тестом в додатках, як Блокнот, WordPad, Microsoft Word чи на певних додатках на мобільних пристроях.

У попередньому розділі були наведені приклади питань у різних системах для електронного чи комп'ютерного тестового контролю. Звісно не всі типи запитань можна використати при розробці друкованих тестів, але деякі типи запитань можливо адаптувати до потреб цього проекту. Наприклад типи запитань з зображеннями або перетягуванням звісно у такому роді тестів зробити неможливо, але запитання типу співставлення чи вкладене запитання адаптувати досить легко. Найбільш вдалим типом запитань для друкованих тестів є множинний вибір(з однією чи декількома правильними відповідями), вірно/невірно, коротка відповідь(не важливо числовий чи текстовий тип). З запитаннями типу есе дещо складніше, оскільки перевірка тесту має бути автономною, а питання типу есе вимагають особистої перевірки викладача вони також не можуть бути представлені в стандартах для цього типу тестів.

Іншим обмеженням також є розмір тесту, оптимальним розміром тестового листа є 1 сторінка, а кількість запитань близько 12. Таке обмеження вводиться для покращення якості розпізнання символів при обробці фото файлу тесту програмою. Також при розробці тесту варто відвести окремі місця для таких полів, як назва тесту, ПІБ студента, група студента та при бажанні дата тестування. Тож бажано надсилати 2 типи документу, перший з запитаннями та варіантами відповідей, другий бланк для заповнення відповідей.

Оскільки, даний тип тесту використовує методи розпізнання символів, необхідно чітко позначати типи даних наприклад в місці для вводу ПІБ студента необхідно спочатку рядка вказати «Name:» або перед відповіддю на початку рядку необхідно вказати «Answer:». Такі позначення використовуються для полегшення пошуку та розрізнення інформації. На рисунку 2.1 зображено приклад документу з запитаннями з відповідями на рисунку 2.2 зображено документ типу бланк відповідей, а на рисунку 2.3 варіант документу типу бланк відповідей з мобільного пристрою.

#### Neural Network Test

1. What is called as the degree of adjustment due to learning trial in neural networks?
  - 1) Flexibility rate
  - 2) Learning rate
  - 3) Deviation rate
  - 4) Momentum
2. Which data mining is appropriate for undirected data mining?
  - 1) Association rules
  - 2) Statistical
  - 3) Decision tree
  - 4) Neural network
3. Which neural network model is computationally expensive?
4. Which of the following is similar to genetic algorithm?
  - 1) Association rules
  - 2) Neural network
  - 3) Decision tree
  - 4) Statistical
5. Which neural network topology enables two way connection amongst all processor?
  - 1) Gaussian mixture
  - 2) Feed forward
  - 3) Limited recurrent
  - 4) Fully recurrent
6. Which type of neural network does kohon features maps are?
  - 1) Gaussian mixture
  - 2) Feed forward
  - 3) Limited recurrent
  - 4) Fully recurrent

## Рисунок 2.1 – Документ типу «Запитання з відповідями»

Name: Sydorenko Maksym  
Group: KND-41

### Test: Neural Network Test

1. What is called as the degree of adjustment due to learning trial in neural networks?  
Answer: Learning rate
2. Which data mining is appropriate for undirected data mining?  
Answer: Association rule
3. Which neural network model is computationally expensive?  
Answer: Back propagation
4. Which of the following is similar to genetic algorithm?  
Answer: Neural network
5. Which neural network topology enables two way connection amongst all processor?  
Answer: Fully recurrent
6. Which type of neural network does kohen features maps are?  
Answer: Feed forward
7. Which type does the predictive modeling is an example of?  
Answer: Data Mining process
8. How is the strength of association rules is indicated by support factor and?  
Answer: Confidence factor

## Рисунок 2.2 – Документ типу «Бланк відповідей»

Name : Sydorenko Maksym  
Group: KND-41

### Test: Neural Network Test

Answer: Learning rate  
Answer: Association rule  
Answer: Back propagation  
Answer: Neural network  
Answer: Fully recurrent  
Answer: Feed forward  
Answer: Data Mining process  
Answer: Confidence factor

Рисунок 2.3 – Документ типу «Бланк відповідей» з мобільного пристрою  
Отже, маємо наступні стандарти, які потрібно взяти до уваги при розробці тестів для коректної роботи програмного засобу:

- Розмір тестового листа – 1 сторінка;
- Кількість запитань до 12;
- Бажана довжина відповіді – 1 рядок;



- Бажаний тип запитань:
  - Множинний вибір(з однією чи декількома правильними відповідями);
  - Питання типу коротка відповідь;
  - Вірно/Невірно;
  - Порядкове ранжування;
- Два типи документів для надсилання:
  - Запитання та відповіді;
  - Бланк для відповідей;
- У бланку для відповідей відвести місце для визначення тесту та вводу особистої інформації студента
- Чітко позначати типи даних

### **2.3 IDEF0 модель системи тестування**

IDEF відноситься до групи мов моделювання, що використовуються для реалізацій систем та інженерних рішень. Ряд мов IDEF має широкий спектр застосування, наприклад використовується для функціонального моделювання даних та об'єктно-орієнтованого аналізу.

IDEF (інтегроване визначення) – це графічний засіб моделювання процесів, що застосовується для моделювання процесів та інженерних програмних засобів.

Наразі IDEF включає в себе 16 різних методів(від IDEF0 до IDEF14)[13]. Під час процесу моделювання кожен метод фіксує певний тип даних. Окрім ролі IDEF в аналізі моделей та створенні версій систем, IDEF також корисний при перекладі систем у графічну форму. Для спрощення переходів моделей використовується аналіз розривів.

Одним з найпоширеніших програмних процесів IDEF є використання IDEF для моделювання функцій систем, програмних засобів чи підприємств. Це застосовується для графічного моделювання елементів керування, його функцій з

різними ресурсами, що використовуються в рамках процесів управління, їх процедур та різних взаємодій.

### **Переваги IDEF**

Надалі наведені переваги використання IDEF для моделювання процесів:

- IDEF – це добре зарекомендований стандарт моделювання. Документація знаходиться у вільному доступі та стандартизована;
- IDEF – не залежить від технологій. Використовується майже у всіх можливих сферах будь-то IT чи бізнес;
- IDEF використовує обмежений набір позначень;
- IDEF дуже проста у використанні система моделювання, людина дуже швидко навчиться її розуміти та читати;
- IDEF є офіційною методологією для іменування процесів, діаграм та надання зворотного зв'язку щодо діаграм.

IDEF0 – це метод, призначений для моделювання рішень, дій, діяльності підприємств чи систем, а також для аналізу функціональної перспективи системи.

IDEF0 допомагає користувачу визначити, які функції виконуються, що потрібно для виконання цих функцій, що дана система робить правильно, а що система робить неправильно. Розвиток IDEF0 багато в чому пов'язаний з позначенням SADT (системний аналіз та техніка проектування)[14].

Моделювання діяльності IDEF0 – це техніка для аналізу цілих систем, як сукупності взаємопов'язаних видів діяльності або функцій. Ця функціональна спрямованість важлива – аналізуються функції систем незалежно від об'єктів, які їх виконують. Ідея полягає в тому, що функції в системах можуть виконуватись як частини окремих процесів проектування. Суто функціональна перспектива дозволяє чітко відокремити питання значення від питань реалізації. IDEF0 найкраще підходить, як техніка аналізу та логічного проектування.

Особливістю при читанні діаграми IDEF0 є стрілки та їх положення. При роботі з діаграмою IDEF0 потрібно запам'ятати наступне:

- **Вхід** знаходиться зліва. Вони можуть надходити від краю діаграми(зовнішні входи) чи від іншої функції на діаграмі. Це данні, що визначають певну задачу.
- Елементи **управління** знаходяться зверху. Вони надходити від краю діаграми(зовнішні тригери) чи від іншої функції на діаграмі. Це механізми керування певні інструкції, бібліотеки тощо.
- **Механізми** входять знизу. Використовуються, щоб виконати певну роботу.
- **Виходи** виходять праворуч. Вони можуть підключатися до іншої функції на діаграмі, виходити до краю діаграми(завершувати). Вихід викликає як внутрішні так і зовнішні функції. Виводить результати діяльності діаграми[14].

Надалі на рисунка 2.4-2.7 наведено діаграми IDEF0, яка моделює роботу засобу тестового контролю з використанням оптичного розпізнавання символів (OCR) та її декомпозиції.



Рисунок 2.4 – Діаграма IDEF0

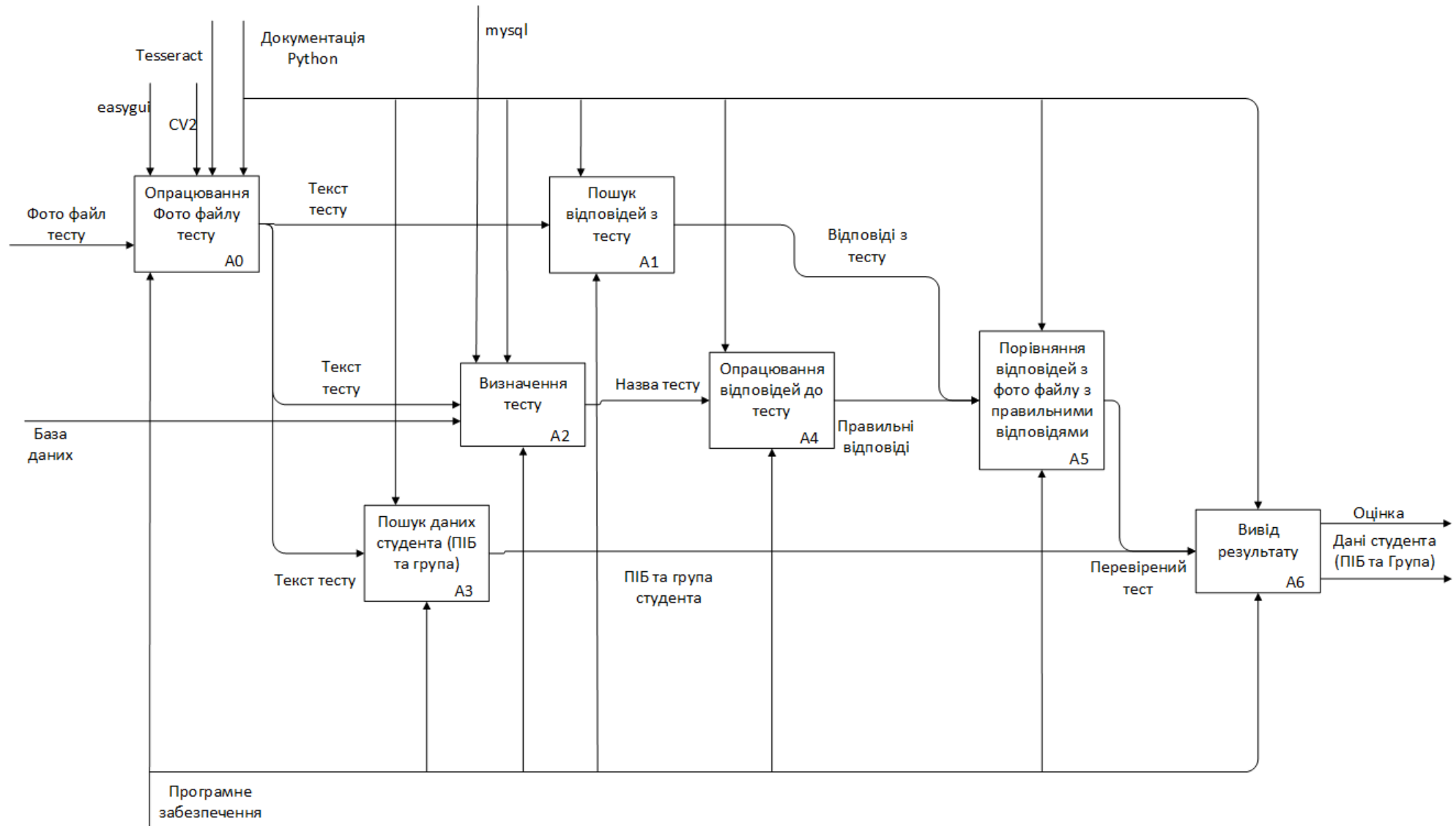


Рисунок 2.5 – Перша декомпозиція діаграми IDEF0

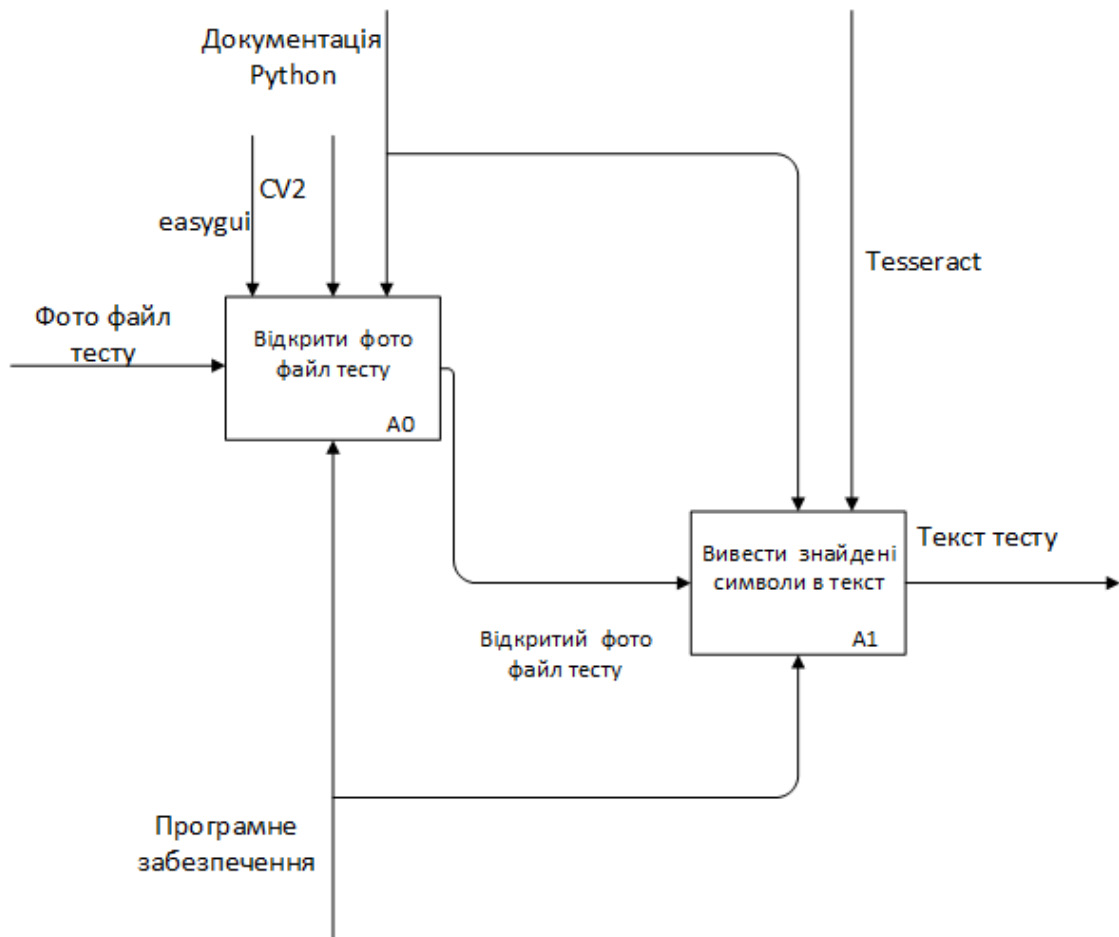


Рисунок 2.6 – Декомпозиція блоку A0 «Обработка фото файла теста» диаграммы IDEF0

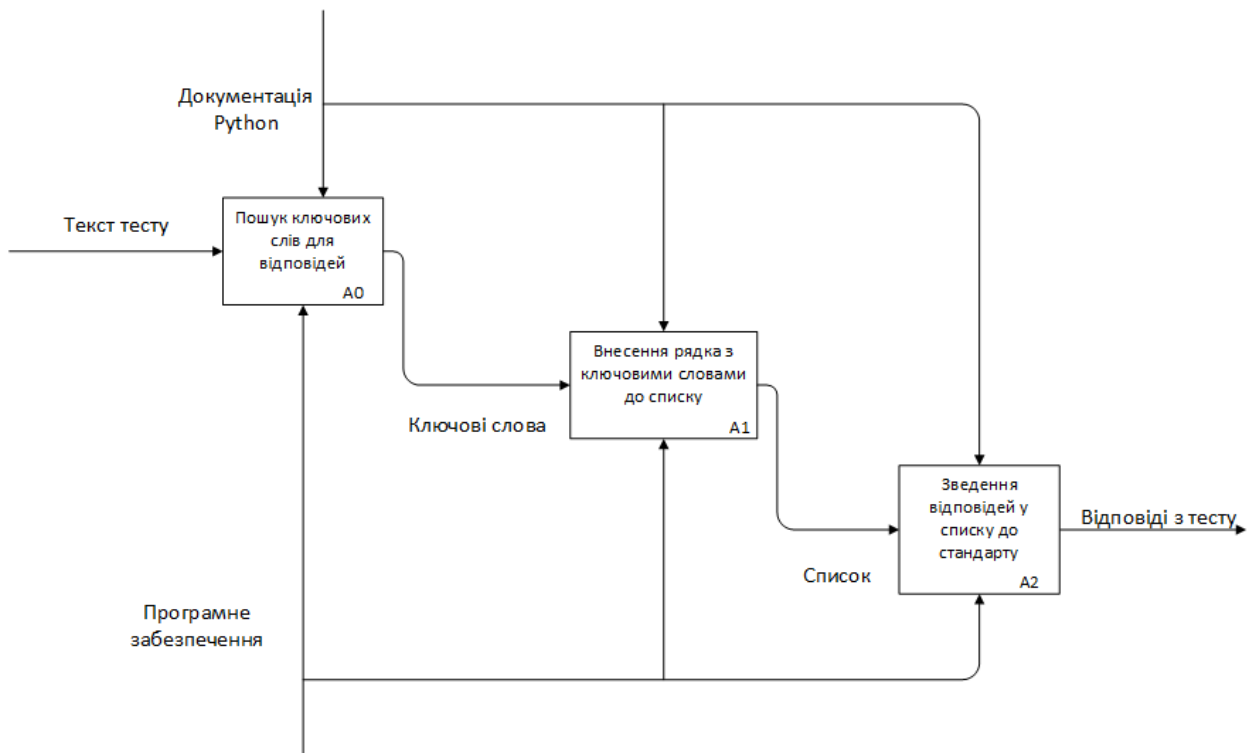


Рисунок 2.7 – Декомпозиція блоку A1 «Поиск ответов из теста» диаграммы IDEF0

**Вхід:**

- Фото файл тесту
- База даних

**Управління:**

- Документація Python
- CV2
- Tesseract
- Easygui
- mysql

**Механізми:**

- Програмне забезпечення

**Вихід:**

- Оцінка
- Дані студента (ПІБ та Група)

**Опис роботи моделі**

Дана діаграма IDEF0 зображує реалістичну версію програмного продукту для тестового контролю за допомогою оптичного розпізнавання символів(OCR). На вході ми маємо фото файл тесту та відповіді до тесту, що містяться у базі даних; на виході вже оцінку з перевіреного тесту та дані про студента, тобто ПІБ та групу. В ролі елементів управління виступають бібліотеки CV2, Tesseract, easygui, mysql та документація Python, а механізмом виступає саме програмне забезпечення. Працює програма так:

Завантажується необхідний фото файл тесту, далі за допомогою оптичного розпізнавання символів з фото файлу тесту знаходить символи та записуються в текст. Відбувається з'єднання з базою даних. Далі за допомогою пошук за ключовими словами знаходяться назва тесту, ПІБ студента, група студента та відповіді з тесту. Відповіді з тесту записуються у список. Знаючи назву тесту необхідно схожу за назвою таблицю у базі даних, попередньо звівши назву тесту

отриману з фото файлу тесту до стандартів(прибравши регістри та пробіли). Знайшовши необхідну таблицю у базі даних з неї вилучаються дані тобто відповіді у список. Потім список з відповідями з тесту та відповідями з бази даних зводяться до одного стандарту(прибравши регістри та пробіли) та перезаписуються. Оновлені списки порівнюються між собою, за правильну відповідь нараховується бал. Наприкінці програма виводить дані студента, тобто ПІБ та групу студента, а також оцінку перевіреного тесту.

### **Опис входів, виходів, механізмів та елементів управління для діаграми**

#### **IDEF0:**

1. Фото файл тесту – фото тесту надіслане студентом.
2. База даних – база даних містить в собі відповіді до тесту. Назва кожної таблиці у базі даних відповідає назві певного тесту. Відповідно кожна таблиця містить відповіді у правильній послідовності.
3. Документація Python – правила написання коду, та необхідна інформація для роботи програми.
4. CV2 – бібліотека Python, необхідна для роботи з фото.
5. Tesseract – бібліотека Python, необхідна для роботи з оптичного розпізнавання символів.
6. Easygui – бібліотека Python, необхідна для роботи з інтерфейсом (в даній роботі допомагає завантажувати фото файл).
7. Mysql – бібліотека Python, необхідна для роботи з базою даних Mysql.
8. Програмне забезпечення – тобто сама програма.
9. Оцінка – оцінка перевіреного тесту.
10. Дані студента (ПІБ та Група) – дані про студента його ПІБ та Група.

## 2.4 Побудова діаграми UML системи тестування

UML (Unified Modeling Language) – це стандартизована мова для моделювання, вона складається з інтегрованого набору діаграм. UML розроблена для допомоги розробникам програмних продуктів та систем, бізнес-моделювань та інших не програмних систем. Ця мова являє собою набір інженерних практик, що довели свою успішність при моделюванні складних та масштабних систем. UML дуже важлива при розробці програмного забезпечення, вона полегшує розуміння функціонування систем. UML в основному використовує графічні позначення, щоб виразити структуру програмних проектів. Використання даних діаграм допомагає команді проекту спілкуватись, досліджувати потенційні проекти та перевіряти дизайн програмних продуктів.

Слід взяти до уваги, що існує велика кількість різних діаграм та схем. Причиною цьому є те, що на розроблену систему можна поглянути з різних точок зору, а в самій розробці програмного продукту бере участь велика кількість фахівців. Наприклад дизайнери, QA, програмісти, керівники проекту чи навіть замовники проекту. Всіх цікавлять різні аспекти продукту чи системи, і кожен вимагає різного рівню деталізації.

Взагалі існує 2 категорії UML які поділяються на 14 різних типів діаграм[15]. Перша категорія це структурні діаграми. Структурні діаграми показують статичну структуру системи та її частин на різних рівнях абстракції та реалізації, а також те, як вони пов'язані між собою. Елементи в структурній діаграмі являють собою різні поняття системи та можуть включати абстрактні, реальні концепції, а також концепції реалізації. Існує декілька типів структурних діаграм, а саме:

- Діаграма класу
- Діаграма профілю
- Діаграма упаковки
- Діаграма об'єкта
- Діаграма компонентів



- Діаграма розгортання
- Складена структурна діаграма

Інша категорія це діаграми поведінки. Діаграми поведінки моделюють динамічну поведінку об'єктів у системах, що також можна представити у вигляді серії змін у системі з часом. Існують наступні типи діаграм поведінки:

- Діаграма діяльності
- Діаграма використання
- Діаграма послідовності
- Діаграма огляду взаємодії
- Діаграма станів
- Часова діаграма
- Схема спілкування

### **Діаграма діяльності**

Діаграма діяльності належить до категорії діаграм поведінки тому, що дана діаграма описує, що має відбуватися у змодельованій системі. Діаграма діяльності не тільки зображує динамічний характер системи, а також використовується для побудови системи за допомогою прямих та зворотних інженерних методів. Але даній діаграмі не відображає потоку повідомлень від однієї діяльності до іншої. Іноді діаграми діяльності розглядаються, як блок-схема. Відрізняються вони тим, що у діаграмі діяльності показуються різні потоки, такі як розгалуження, одиночний, одночасний та паралельний [16].

### **Переваги діаграми діяльності:**

- Дуже просто схематично пояснити кожен крок та етап у системі
- Можливо детально пояснити методи, процеси та функції
- Легко зобразити потоки та бізнес-процеси
- Просто пояснити системні вимоги
- Детально пояснює робочий процес користувача та системи
- Програмні алгоритми

### Компоненти діаграми діяльності:

Перш ніж складати діаграму необхідно знати її базові компоненти, а саме:

- Початковий вузол;

Символізує початок роботи певної системи. Має вигляд пустого кола.

- Дія;

Даний компоненти представляє собою виконання певної задачі користувачем чи програмним продуктом. Найчастіше виглядає, як прямокутник.

- Управління потоками;

Інколи також називають сполучником, показує потік між кроками на діаграмі. Має вигляд стрілки.

- Вузол рішення;

Умовна гілка в потоці, включає в себе лише один вхід і два чи більше виходи. Найчастіше виглядає, як ромб.

- Вузол об'єднання ;

Завершення умовної гілки в потоці. Включає в себе від двох входів та лише один вихід. Також має вигляд ромба.

- Вузол розгалуження;

Використовується коли починається виконання декількох паралельних дій. Включає в себе один вхід та декілька виходів

- Вузол з'єднання;

Використовується коли закінчується виконання декількох паралельних дій. Включає в себе декілька входів та один вихід.

- Кінцевий вузол;

Символізує кінець роботи певної системи. Має вигляд заповненого кола.

Надалі на рисунках 2.8-2.10 буде представлено приклад діаграми діяльності для програмного засобу тестового контролю з використання оптичного розпізнавання символів.

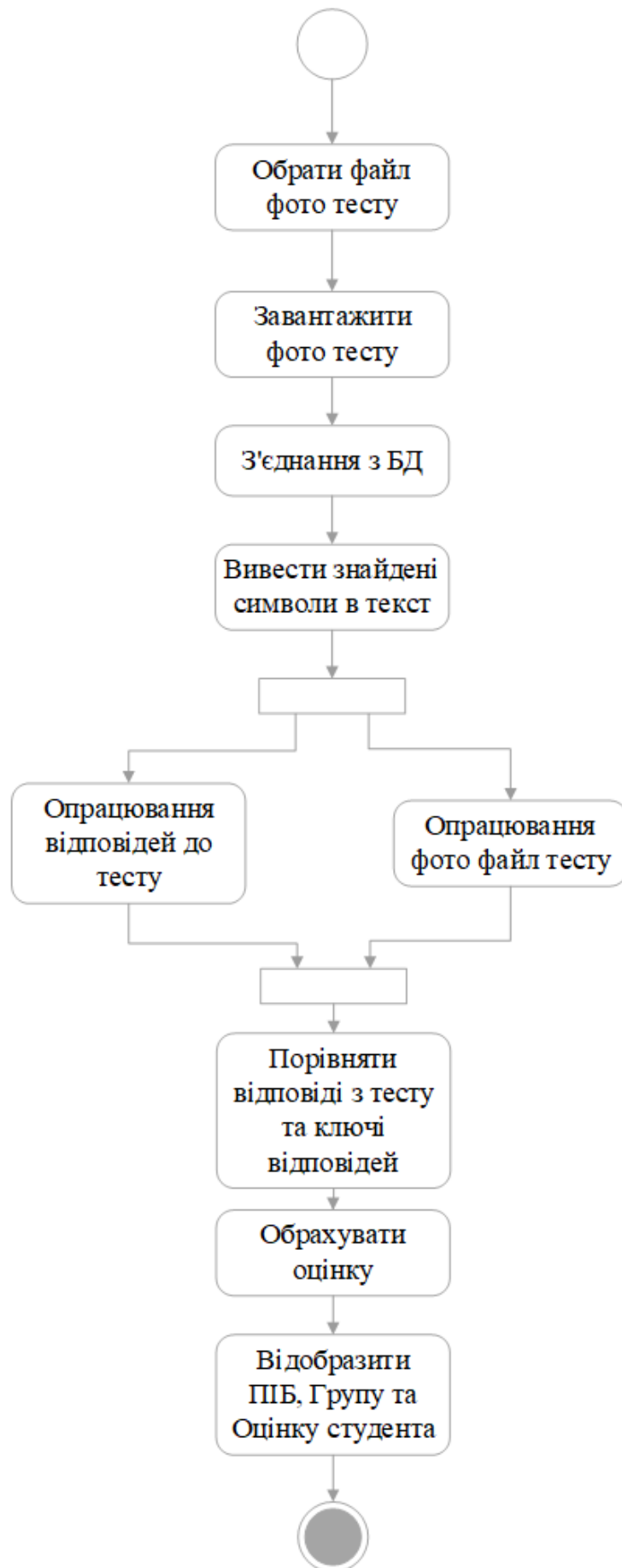


Рисунок 2.8 – Діаграма діяльності UML

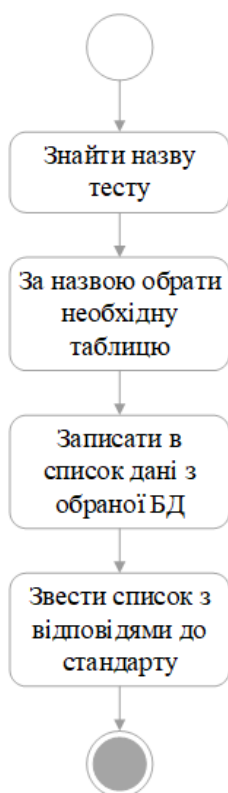


Рисунок 2.9 – Розкриття блоку «Опрацювання відповідей до тесту» діаграма діяльності UML

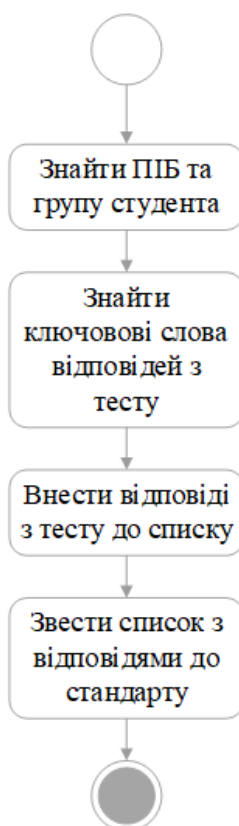


Рисунок 2.10 – Розкриття блоку «Опрацювання відповідей до тесту» діаграма діяльності UML

## 2.5 Оптичне розпізнавання символів (OCR)

Оптичне розпізнавання символів або OCR – це популярна технологія, що допомагає читати друкований чи рукописний текст. Зазвичай технологію оптичного розпізнавання символів використовують для відцифрування документів, завдяки цій технології можна скоротити час необхідний для ручного введення даних та обробки документів.

Існують різні типи оптичного розпізнавання символів(OCR):

- Інтелектуальне розпізнавання слів (IWR) – орієнтований на рукописний текст. Алгоритм роботи полягає в розпізнавання цілого слова, а не відбору окремих символів.
- Інтелектуальне розпізнавання символів (ICR) – орієнтований на рукописний текст. Алгоритм роботи полягає в ідентифікації окремих символів та розвивається за допомогою машинного навчання.
- Оптичне розпізнавання слів (OWR) – орієнтований на друкований(машинописний) текст. Алгоритм роботи полягає в розпізнаванні цілих слів.
- Оптичне розпізнавання символів (OCR) – орієнтований на друкований(машинописний) текст. Алгоритм роботи полягає в розпізнаванні окремих символів..
- Оптичне розпізнавання знаків (OMR) – це технологія збору даних шляхом розпізнавання знаків та шаблонів в документі.[17]

Приклади використовують технології оптичного розпізнавання символів в різних галузях:

- Банківська справа:
  - Перевірка інформації чеків та рахунків написаних від руки
  - Відцифрування заявок, що містять велику кількість документів
- Страхування: автоматизована обробка документів

- Юридичні послуги: оцифрування документів, заяв, квитанцій, заповітів тощо.
- Охорона здоров'я: оцифрування звітів, рентгенівських знімків, історій хвороб, діагностик, лікарняних записів тощо.[18]

Основним принципом оптичного розпізнавання символів є спочатку навчання програмного засобу шаблонам, як вони можуть виникнути та як вони виглядають. У оптичному розпізнаванні символів шаблонами можуть бути букви, цифри чи інші спеціальні символи такі, як коми, знаки запитань, крапки. Навчають програму показуючи їй різні приклади символів усіх видів класів. Беручи за базу такі шаблони програма будує прототипи та опис кожного класу символів. Потім під час розпізнавання порівнює невідомі символи з уже відомими шаблонами, так визначивши найбільш схожий до невідомого символу та отримуємо клас, що найбільш відповідає дійсному символу. Більшість вже працюючих систем розпізнавання символів навчальний процес заздалегідь сформований, однак є системи, які включають в себе засоби для навчання.

## **2.6 Методи оптичного розпізнавання символів**

Робота системи розпізнавання символів потребує декількох кроків. На рисунку 2.11 зображено кроки оптичного розпізнавання символів.



Рисунок 2.11 – Кроки оптичного розпізнавання символів

Першим кроком у процесі розпізнавання є відцифрування необхідного документа за допомогою сканера чи можна зробити фото за допомогою телефону або фотокамери. Коли є цифрова копія документа потрібно визначити границі в яких знаходиться необхідний текст, далі кожен символ виділяється за допомогою сегментації. Потім знайденні символи потрібно попередньо обробити, усунути шуми, щоб полегшити подальші кроки з вилучення та розпізнання символів. Визначається кожен символ завдяки порівнянню знайдених символів з описами класів символів отриманими під час етапу навчання. Наприкінці інформація використовується для відновлення отриманих слів та цифр.

### 2.6.1 Оптичне сканування

Під час процесу сканування відбувається захоплення зображення оригінального документа. Друковані документи зазвичай друкуються двома кольорами чорним та білим. Отже, при розпізнаванні символів загальноприйнятою практикою є перетворення багаторівневого(кольорового) зображення

у дворівневе(чорно-біле) зображення. Такий процес також відомий, як порогове значення, виконується для економії обчислювальних зусиль.

Пороговий процес є дуже важливим, адже результати наступних кроків оптичного розпізнавання символів цілком залежить від якості зображення. Тим не менш, порогове значення виконане на сканері зазвичай дуже просте. Але щоб отримати хороший результат, потрібні більш досконалі методи порогового значення, такі методи залежать від більш досконалих засобів сканування.

### 2.6.2 Сегментація

Сегментація – це процес, що визначає складові зображення. Сегментація необхідна, щоб знайти границі документу. Іншими словами сегментація це ізоляція символів чи слів. Більшість алгоритмів оптичного розпізнавання символів сегментують слова на окремі символи, що визначаються окремо. Зазвичай сегментація проводиться шляхом вилученням кожного визначеного елемента. Це легко зробити, але можуть виникнути проблеми, якщо елементи торкаються одне одного чи вони фрагментовані та складаються з декількох частин.

Основні проблеми сегментації:

#### 1. Вилучення спотворених та фрагментованих символів;

Такі спотворення можуть призвести до того, що декілька символів можуть бути інтерпретовані як один символ або частина символу може бути інтерпретована як один символ. Також такі спотворення можуть виникати через недостатню якість зображення.

#### 2. Розрізнення шуму;

Деякі спеціальні символи можуть бути сприйняти, як спотворення зображення або шум, і навпаки.

#### 3. Сприйняття графічних зображень, як текст;

Це призводить до того, що графічні образи помилково сприйняті, як текст.



#### 4. Сприйняття тексту, як графічних зображень;

У такому випадку текст не буде розпізнано через помилкове сприйняття. Таке буває коли символи розміщені разом, з графічним зображенням [19].

### 2.6.3 Попередня обробка

Зображення отримане під час сканування може містити певний шум. Залежно від методу або засобу сканування чи фотографування символи також можуть бути змазані, нечіткими чи розбитими. Такі дефекти можуть спричинити подальші проблеми, як низький рівень розпізнавання символів, ці дефекти можна усунути за допомогою згладжування.

Згладжування передбачає собою заповнення та розрідження . Заповнення усуває невеликі розриви, діри, пробіли в оцифрованих символах, тоді як розрідження зменшує ширину рядка. Найпоширенішу прийоми згладжування – переміщення вікна по бінарному зображенню символу, застосовуючи певні правила до вмісту вікна.

Окрім згладжування попередня обробка включає в себе нормалізацію. Нормалізація застосовується для отримання символів однакового розміру та нахилу. Щоб мати можливість виправити нахили тексту, спочатку потрібно знайти кут нахилу. Для такого зазвичай використовують перетворення Хафа[20].

### 2.6.4 Вилучення ознак

Вилучення ознак одна з найскладніших проблем розпізнавання символів, метою вилучення ознак є охоплення основних характеристик символів. Один з підходів вилучення ознак полягає у виділенні певних ознак, які характеризують символи, але не враховують найважливіші атрибути. Методи вилучення ознак поділяють на три основні групи:

- Розподіл балів;  
Це зазвичай толерантні до спотворень та варіацій стилю.
- Перетворення рядків;  
Такий метод допомагає зменшити розмірність вектора ознак, і витягнуті символи можна зробити незмінними деформацій, таких як перенесення та обертання.
- Структурний аналіз;  
Під час структурного аналізу виділяються ознаки, що описують геометричні та топологічні структури символу.

Різні ознаки можуть бути оцінені відповідно до їх чутливості до шуму, спотворення та практичності використання. Критерії, що використовуються при такій оцінці, наступні:

- Надійність:
  - 1) Шум;  
Чутливість до нерівностей, пробілів, зазорів тощо
  - 2) Спотворення;  
Чутливість до змін таких, як закруглені кути, неправильні виступи та розширення.
  - 3) Варіації стилю;  
Чутливість до змін стилю використаних під час написання тексту
  - 4) Переклад;  
Чутливість до руху цілого персонажу або його складових
  - 5) Нахил;  
Чутливість до змін у орієнтації символів.
- Практичне використання:
  - 1) Швидкість розпізнавання
  - 2) Складність
  - 3) Незалежність

### 2.6.5 Післяобробка

Результатом розпізнавання документу є набір простих символів. Ці самі символи не достатньої інформації. Щоб отримати необхідну нам інформацію ці символи потрібно пов'язати між собою, тобто визначити, що належить до одного слова, цифри чи рядка. Такий процес також називають групуванням. Групування символів по рядку залежить від розташування символів у документі.

Символи, що знаходяться досить близько між собою групуються разом утворюючи слова чи цифри. Для набору символів відстань між ними є змінною, для деяких шрифтів із фіксованим кроком, групування досить просте, оскільки положення символів відоме. Так як, відстань між словами значно більша, ніж відстань між символами, групування є можливим. Проте виникають проблеми коли текст знаходить під нахилом, особливо у рукописному тексті.

Групування кожного символу не розглядається з боку контексту. Тобто під час розпізнавання тексту, систем, що складаються лише з одного символу недосить. 100% розпізнавання символів не можуть надати навіть найкращі системи та засоби оптичного розпізнавання символів, проте деякі помилки можуть бути виявленні та виправленні. Є декілька основних підходів, першій використовує можливість послідовних символів, що можуть з'явитись разом, певні шаблони. Це можна зробити шляхом використання правил синтаксису слів. Окрім того у різних мовах можна визначити ймовірність появи двох чи більше символів у певній послідовності, що можна використати для виявлення помилок.

Інший підхід використовує словники, це виявилось найбільш ефективним методом визначення та виправлення помилок. Визначивши слово у якому може бути помилка, його можливо перевірити використавши словник. Якщо такого слова не існує, це означає що існує помилка, її можна виправити, змінивши слово на найбільш побідне. На жаль, якщо слово присутнє у словнику, це не доводить, що немає помилки. Недоліком словникового методу є те, що такі запити та порівняння вимагають набагато більше часу.

## 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

### 3.1 Побудова бази даних

База даних є важливою частинною даного програмного продукту, бо в ній будуть зберігатись правильні відповіді до тестів. Також, саме завдяки базі даних у користувачів наявна постійна можливість добавляти чи видаляти відповіді до тестів та можливість редагувати зміст відповідей у разі зміни запитань у тесті. В якості системи управління базами даних було обрано – MySQL, адже вона є найбільш популярною системою подібного типу.

База даних MySQL є реляційним типом баз даних. Взагалі, база даних – це структурована сукупність даних, в ній містяться та упорядковуються дані. Реляційна база даних означає, що дані в такій базі даних зберігаються у вигляді таблиць, а не в одному великому сховищі. Структури баз даних організовані у фізичні файли. Часом MySQL можуть сплутати з SQL – це не одне й теж саме. Як вже зрозуміло MySQL – це структура де зберігаються дані, а SQL – це структурована мова запитів, яка використовується для доступу та управління базами даних. До речі, системи MySQL являє собою програмний засіб з відкритим кодом, тобто кожен може його завантажити та користуватись базою даних MySQL безкоштовно. MySQL був розроблений у 1970-х роках на базі мов C а C++ для швидкої обробки великого обсягу даних [21].

Причини для використання MySQL:

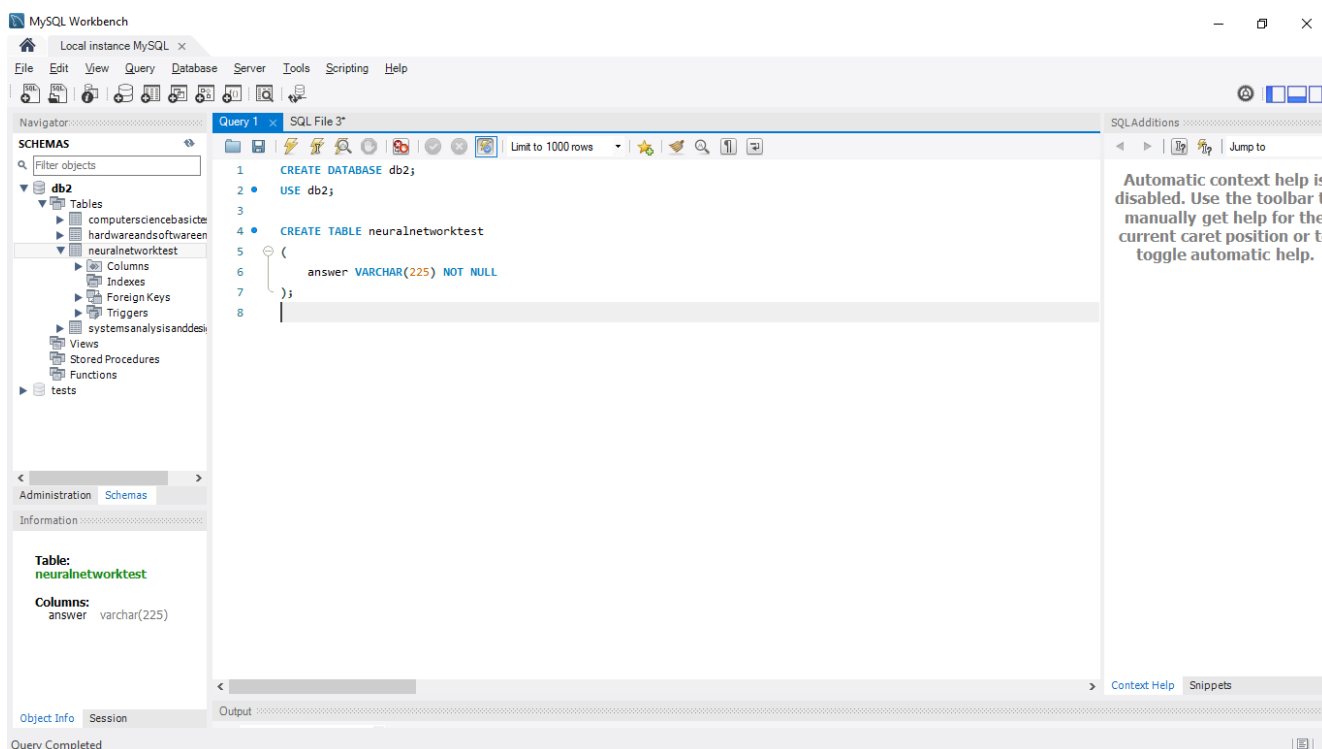
- MySQL підтримує великий спектр мов таких, як Python, Java, C, C++, PHP тощо;
- MySQL простий у використанні, що підходить для малих та великих обсягів; даних;
- MySQL є однією з найбільш безпечний та надійних систем;
- MySQL є дуже гнучкою системою завдяки відкритому коду;

- MySQL є більш економічним у використанні в порівнянні з іншими системами

MySQL підтримує велику кількість графічних інтерфейсів таких, як MySQL WorkBench, SequelPro, DBVosualizer.

### Побудова бази даних у MySQL WorkBench

Для роботи з цим проектом було обрано графічний інтерфейс MySQL WorkBench. MySQL Цей інструмент дозволяє проектувати та моделювати бази даних для реляційної бази даних MySQL. Вона полегшує створення нових моделей даних та редагування уже існуючих баз даних MySQL завдяки вбудованому редактору SQL. MySQL WorkBench має інструменти, які дозволяють розробникам та адміністраторам баз даних візуально створювати фізичні моделі проектування баз даних, які можна легко перевести в бази даних MySQL за допомогою прямої інженерії. MySQL Workbench підтримує створення декількох моделей в одному середовищі. Можна запустити кілька запитів, а результати автоматично відображатимуться на різних вкладках, запити також зберігаються на панелі історії для подальшого пошуку та запуску. На рисунку 3.1 зображено вікно розробки SQL в MySQL WorkBench.



Рисунку 3.1 – Вікно розробки SQL в MySQL WorkBench.

Отже, створюємо нову базу даних з назвою *db2* для цього у вікно розробки необхідно записати наступну команду:

```
CREATE DATABASE db2;
```

Далі необхідно підключитись до створеної бази даних *db2* за допомогою :

```
USE db2;
```

Далі створюємо необхідну таблицю. У випадку з наших проектом кожна таблиця представляє собою окрему базу відповідей до тесту, тому важливо назвати таблицю відповідно до назви тесту, що відповідає йому. У середині дужок пишемо назви колонок та їх тип даних. Наприклад за допомогою наступної команди створимо таблицю *neuralnetworktest*, що відповідає тесту «Neural Network Test», з колонкою *answer* та типом даних *VARCHAR* максимальною довжиною в символів 225, елементи в колонці не можуть бути пустими. Це все можна зробити за допомогою наступної команди:

```
CREATE TABLE neuralnetworktest
(
  answer VARCHAR(225) NOT NULL
);
```

Після вводу даної частини коду його потрібно виконати для цього необхідно натиснути на значок у вигляді жовтої блискавки. На рисунку 3.1 також зображено вже створена база даних *db2* з таблицею *neuralnetworktest*. Далі необхідно заповнити створену таблицю, це можна зробити за допомогою наступної команди:

```
INSERT neuralnetworktest (answer)
VALUES ('Сюди записуємо відповідь до тесту');
```

За одну команду заповнити можна лише один елемент. Звісно цю команду можна повторити декілька разів і виконати їх разом. Щоб передивитись вміст створеної таблиці *neuralnetworktest*, необхідно ввести та виконати наступну команду:

```
select* from neuralnetworktest
```

На рисунку 3.2 представлено заповнену таблицю *neuralnetworktest*.

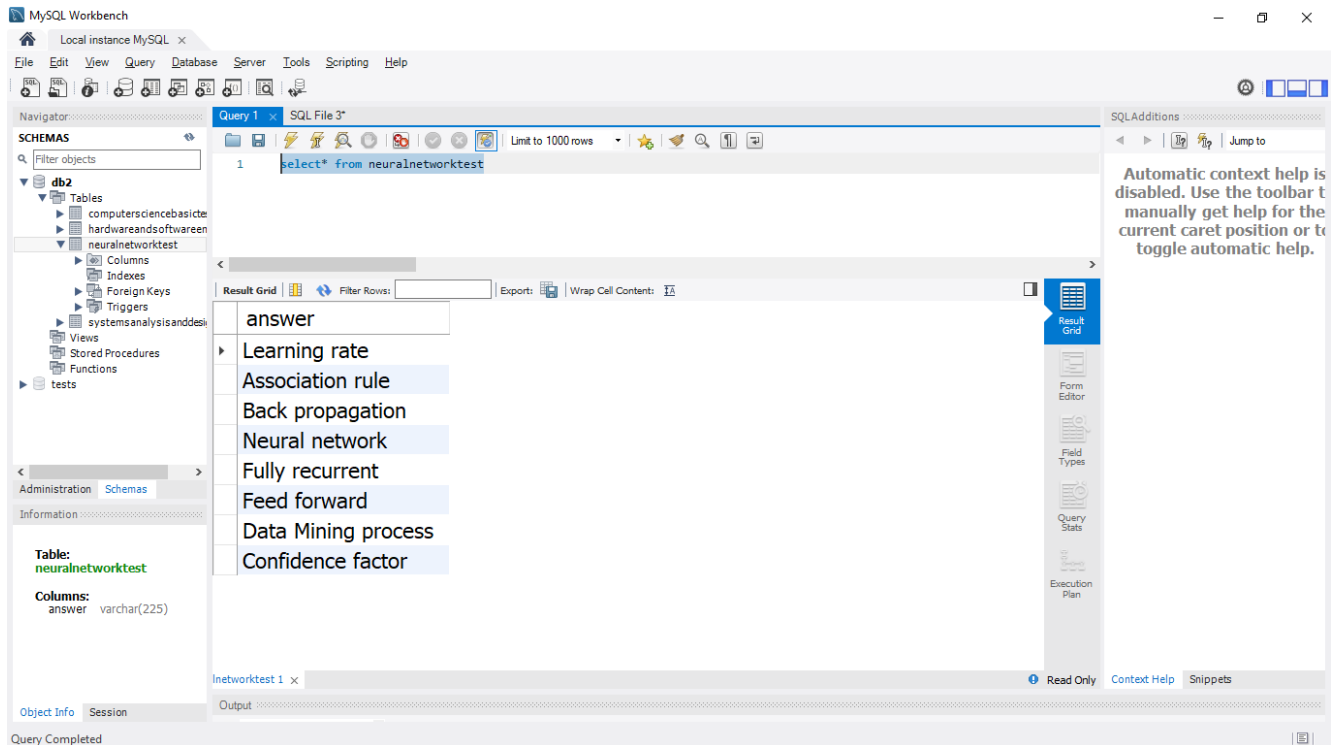


Рисунок 3.2 – Заповнена таблиця neuralnetworktest у MySQL WorkBench

### 3.2 Реалізації програмного продукту

Реалізація програмного засобу відбувається на основі мови програмування Python. Python наразі одна з найпопулярніших та найбільш використовуваних мов програмування у світі та може бути використана для найрізноманітніших типів проектів. Вона включає високо рівневі структури даних, динамічне введення тексту та багато інших функцій, які роблять Python бажаною для розробки додатків. Python також може бути розширена для здійснення системних запитів майже у всі операційні системи та для запуску коду написаного на C та C++. Завдяки своїй простоті та гнучкості, вона може працювати майже на будь-якій системній архітектурі.

Python – дуже універсальна мова, а оскільки вона призначена для загального кодування, може бути використана для різних типів розробок програмного забезпечення. Наприклад Python можна використовувати для розробок веб-сторінок, додатків для мобільних пристроїв, додатків та програмних продуктів

для ПК, обробки великих даних, виконання математичних обчислень та написання системних скриптів. А завдяки своїй простоті написання коду та багатофункціональності, Python також можна використовувати для більш складних проектів, як машинне навчання та розробка штучного інтелекту.

Завдяки своїй простоті Python дуже легка у вивченні, її легко читати та розуміти, що робить цю мову ідеальною мовою для навчання. А завдяки своїй популярності та широті розповсюдження має дуже велику базу різних бібліотек та служб на основі API, що дозволяє вільно розробляти додатки не зважаючи на ціль розробки. Python має багатий функціонал, універсальність та часте використання для професійних розробок програмних продуктів робить її дуже бажаною. Оскільки Python постійно оновлює свій функціонал не потрібно боятись, що вона застаріє.

Python має величезну базу бібліотек, що звісно також були використанні при розробці цього проекту. Основні бібліотеки використанні у цій роботі Tkinter, OpenCV та Tesseract.

Tkinter – це фреймворк, що надає користувачу Python досить просто створити графічний інтерфейс, за допомогою можливостей знайдених у наборі інструментів Tk. Інструменти Tk можна використати для побудови кнопок, меню, полів даних, вікон та тощо. Створені графічні елементи можуть асоціюватись або взаємодіяти з функціями, методами, даними чи іншими інструментами.

OpenCV – це бібліотека використовується для комп'ютерного зору, машинного навчання та обробки зображень. Комп'ютерний зір – це процес за допомогою якого можна зрозуміти, як зображення та відео зберігаються та як можливо витягнути з них дані, в основному використовується для штучного інтелекту. За допомогою OpenCV можна обробляти зображення та відео, що ідентифікувати різні об'єкти чи обличчя.

Tesseract – це механізм для оптичного розпізнавання символів. Його можна використовувати безпосередньо або за допомогою API для вилучення друкованого тексту з зображення. Tesseract підтримує широкий спектр мов та сумісний з багатьма фреймворками. Його можна використовувати з наявним



аналізом макета для розпізнавання символів у документах чи можна використовувати разом із зовнішніми детекторами тексту для розпізнавання символів з зображення. Процес розпізнавання Tesseract залежить від наступних кроків:

- Пошук символів
- Знаходження ліній
- Класифікація символів

Пошук символів здійснюється шляхом упорядкування текстових рядків у крапки, а рядки та області аналізуються на наявності фіксованого тону або пропорційного тексту. Рядки тексту розбиваються на слова по різному залежно від виду інтервалів між символами. Ця процедура працює в декілька заходів. Під час першого проходу Tesseract намагається розпізнати кожен символ по черзі та групує у слова. Кожне слово передається до адаптивного класифікатора у вигляді навчальних даних. Потім адаптивний класифікатор розпізнає текст на зображенні більш точно.

Модернізація Tesseract була зусиллям щодо очищення коду та додавання нової моделі LSTM. Вхідне зображення оброблюється в коробках, рядок за рядком, ці дані надаються в модель LSTM, яка потім надає вихідні дані. LSTM – це мережа довготривалою пам'яті, особливий вид RNN(рекурсивної нейронної мережі). Вона створена, щоб уникнути проблеми довготривалої залежності, це дозволяє запам'ятовувати інформацію на більш довгий проміжок часу. Завдяки LSTM Tesseract досяг кращих показників під час навчання з великою кількістю даних.

### **3.2.1 Реалізація коду програмного продукту**

Програма складається з двох основних функцій `def interface()` та `def main_program()`. Функція `def interface()` виконує функцію початкового вікна за допомогою якого, у користувача є можливість неодноразово запустити програму. На рисунку 3.3 Зображено вікно функції `def interface()`.

**def** inteface():

```

window = Tk()
window.title("TestRecognition")
lbl = Label(window, text="This program is designed to automate \n"
                "the test validation process in terms \n"
                "of distance learning using optical \n"
                "character recognition technology\n",
                font=("The New Times Roman", 15))
lbl.grid(column=0, row=0)
lbl = Label(window, text="To start test recognition push button",
                font=("The New Times Roman", 12))
lbl.grid(column=0, row=1)
btn = Button(window, text="Choose test", width=15, height=3)
btn.grid(column=0, row=2)
btn.config(command=main_program)
window.mainloop()

```

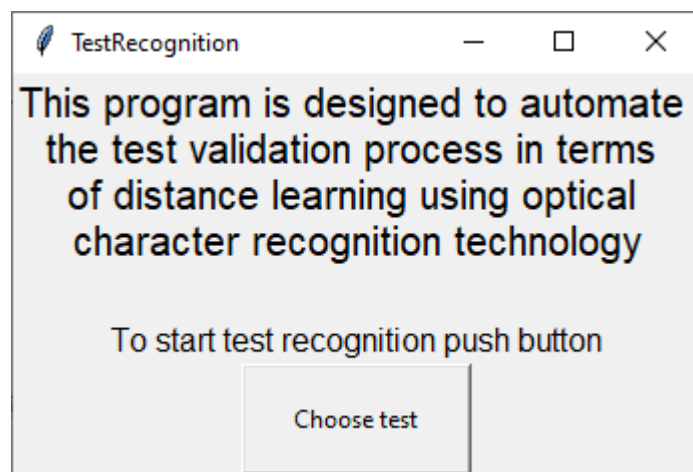


Рисунок 3.3 – Початкове вікно програми «TestRecognition»

Функція `main_program()` є основною функцією, сам в ній проходять основні процеси програми. Містить всі інші необхідні для роботи функції.

**def** main\_program():

```

#Створює оператор для відображення фото файлу тесту
img = ''
img = open(img)
# Розпізнання символів з фото файлу тесту за допомогою
# бібліотеки Tesseract(pytesseract)
text = tess.image_to_string(img)
# Розбиває розпізнаний текст на рядки
splits = text.splitlines()
# Створюються списки для відповідей
test_ans = []
r_answers = []
# З'єднання з базою даних
db = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='Metomorf1234',
    database='db2'
)
#Викликається функція s_name() для пошуку ПІБ студента
name = s_name(splits)
#Викликає функцію s_group() для пошуку групи студента
group = s_group(splits)
#Викликає функцію s_test_name() для пошуку назви тесту
test_name = s_test_name(splits)
#Викликає функцію s_test_ans() для пошуку відповідей з тесту
test_ans = s_test_ans(splits)
#Викликає функцію test_ans_standart()
#для уніфікації формату відповідей з тесту
answers = test_ans_standart(test_ans)

```

```

#Викликає функцію tabnam() для знаходження необхідної таблиці з бази даних
table = tabnam(test_name)
#Викликає функцію tab() для витягу даних(правильних відповідей) з бази даних
r_answers = tab(table)
#Викликає функцію y_mark() для перевірки тесту
mark = y_mark(r_answers)
#Переводить оцінку у формат:
#Правильних відповідей/Загальна кількість
res = ('Mark:', mark, '/', len(answers))
#Викликає функцію nWind() для відображення результатів
nWind()

```

Функція `def_open()` виконує функцію завантаження фото файлу та відкриття її за допомогою бібліотеки `cv2`. У даній випадку бібліотека `easygui` надає можливість обрати будь який файл на комп'ютері.

```

def open(x):
    inp = easygui.fileopenbox()
    x = cv2.imread(inp)
    return x

```

Функція `def s_test_ans()` виконує функцію пошуку відповідей з тесту. Шукає відповіді за ключовим словом `Answer` та записує відповіді у список.

```

def s_test_ans(x):
    test_ans = []
    for line in x:
        if re.search('Answer', line):
            test_ans.append(line)
    return test_ans

```

Функція `def s_name()` виконує функцію пошуку ПІБ студента з тесту. Шукає відповіді за ключовим словом `Name` та записує відповіді у список.

```
def s_name(x):
    name = ''
    for line in x:
        if re.search('Name', line):
            name = line
    return name
```

Функція `def s_group()` виконує функцію пошуку групи студента з тесту. Шукає відповіді за ключовим словом `Group` та записує відповіді у список.

```
def s_group(x):
    group = ''
    for line in x:
        if re.search('Group', line):
            group = line
    return group
```

Функція `def s_test_name()` виконує функцію пошуку назви тесту. Шукає відповіді за ключовим словом `Test` та записує відповіді у список. . Ця функція також зводить формат назви тесту до наступного:

- Видаляє пошукове слово `Test`
- Зводить всі слова до нижнього регістру
- Знищує всі пробіли

Зведення до такого формату необхідне, щоб формат назви тесту відповідав формату назви таблиці(назви тесту), що записана у базі даних.

```
def s_test_name(x):
    test_name = ''
    for line in x:
```

```

if re.search('Test:', line):
    test_name = line

testname = test_name
testname = testname[5:]
testname = testname.replace(' ', '')
testname = testname.lower()

return testname

```

Функція `test_ans_standart()` зводить формат відповідей з тесту у списку до стандарту та перезаписує до іншого списку. Зводить до наступних стандартів:

- Видаляє пошукове слово `Answer`
- Зводить всі слова до нижнього регістру
- Знищує всі пробіли

Така процедура необхідна для того, щоб усунути певну категорію помилок при розпізнанні тесту. А також щоб уникнути помилки при порівнянні з правильними відповідями, бо неправильний регістр чи зайвий пробіл приведуть до того, що відповідь буде вважатись неправильною.

```

def test_ans_standart(x):

```

```

    an = []
    for i in x:
        te = str(i)
        te = te[8:]
        an.append(te)
    m_answers = []
    for i in an:
        n = i
        n = n.replace(' ', '')

```

```

n = n.lower()
m_answers.append(n)
return m_answers

```

Функція `def tabnam()` виконує пошук відповідної таблиці з бази даних. Спочатку у список заносяться всі білиці, що належать даній базі даних. Оскільки витягнення назв таблиці проходить у некоректному форматі, відбувається певне коригування, а саме видаляються всі зайві символи (стандартно 2 в початку та 3 на кінці).

```

def tabnam(x):
    mytable = db.cursor()
    mytable.execute('SHOW TABLES')
    restable = mytable.fetchall()
    res_table = []

    for i in restable:
        te = str(i)
        te = te[2:-3]
        res_table.append(te)

    for i in res_table:
        if x == i:
            table = i
    return table

```

Функція `def tab()` витягує правильні відповіді з обраної таблиці, дані з таблиці заносяться у список. Далі також проходить зведення до певних стандартів:

- Видаляються зайві символи, що з'являються під час витягу даних з таблиці бази даних

- Зводить всі слова до нижнього регістру
- Знищує всі пробіли

Це також робиться, щоб усунути помилки під час порівняння відповідей.

```
def tab(x):
    mycursor = db.cursor()
    mycursor.execute('Select * From ' + x)
    res = mycursor.fetchall()
    r_answers = []

    for i in res:
        te = str(i)
        te = te[2:-3]
        r_answers.append(te)

    m_answers = []

    for i in r_answers:
        n = i
        n = n.replace(' ', '')
        n = n.lower()
        m_answers.append(n)

    return m_answers
```

Функція `y_mark()` виконує функцію перевірки тесту. Ця функція порівнює список відповідей з тесту `answers` та список з правильними відповідями з бази даних `r_answers`. За кожну правильну відповідь нараховується один бал.

```
def y_mark(x):
    mark = 0
    for i in range(len(x)):
```



```

if answers[i] == x[i]:
    mark += 1
return mark

```

Функція `show()` необхідна для відображення фото файлу тесту у початковому вигляді у вигляді вікна. На рисунку 3.4 Зображено приклад цього вікна.

```

def show():

```

```

    cv2.imshow('TestPhoto', img)

```

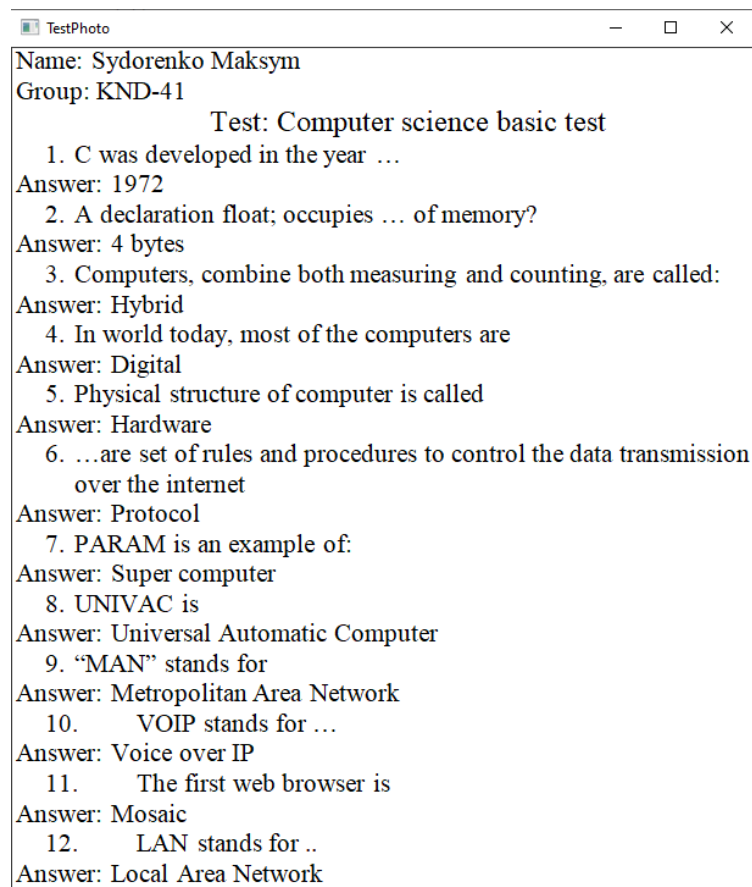


Рисунок 3.4 – Вікно «TestPhoto»

Функція `nWind()` виконує функцію створення другого вікна. В цьому вікні відображається результати тестування(Назва тесту, ПІБ студента, група студента, оцінка), а також є можливість перегляду початкового фото файлу тесту та вікна з результатом розпізнавання тесту. На рисунку 3.5 зображений приклад цього вікна.

```

def nWind():

```

```

    window = Tk()

```

```

    window.title("Result")

```

```

test = Label(window, text=test_name, font=("Arial Bold", 15))
test.grid(column=0, row=1)
sname = Label(window, text=name, font=("Arial Bold", 15))
sname.grid(column=0, row=2)
sgroup = Label(window, text=group, font=("Arial Bold", 15))
sgroup.grid(column=0, row=3)
smark = Label(window, text=res, font=("Arial Bold", 15))
smark.grid(column=0, row=4)
Test_view = Button(window, text="View test", width=10, height=2)
Test_view.grid(column=0, row=5)
Test_view.config(command=show)
Test_text_view = Button(window, text="View test text", width=10, height=2)
Test_text_view.grid(column=0, row=6)
Test_text_view.config(command=tWind)
window.mainloop()

```

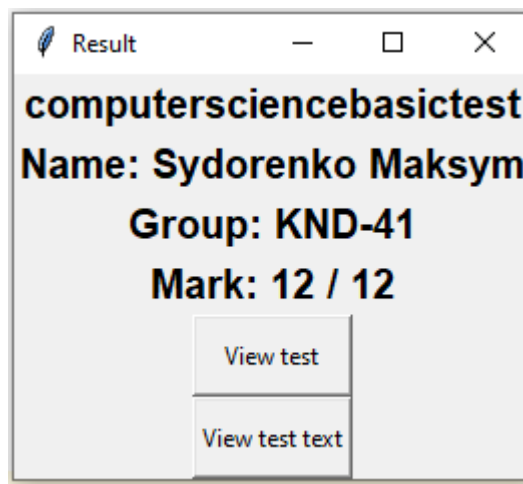


Рисунок 3.5 – Вікно «Result» з результатами перевірки тесту

Функція `tWind()` створює вікно з текстом з фото файлу тесту, розпізнаним за допомогою технології оптичного розпізнавання символів. На рисунку 3.6 зображений приклад цього вікна.

**def** `tWind()`:

```

window2 = Tk()
window2.title("TestInText")

```

```
testtext = Label(window2, text=text, font=('Arial Bold', 12))
testtext.grid(column=0, row=1)
window2.mainloop()
```

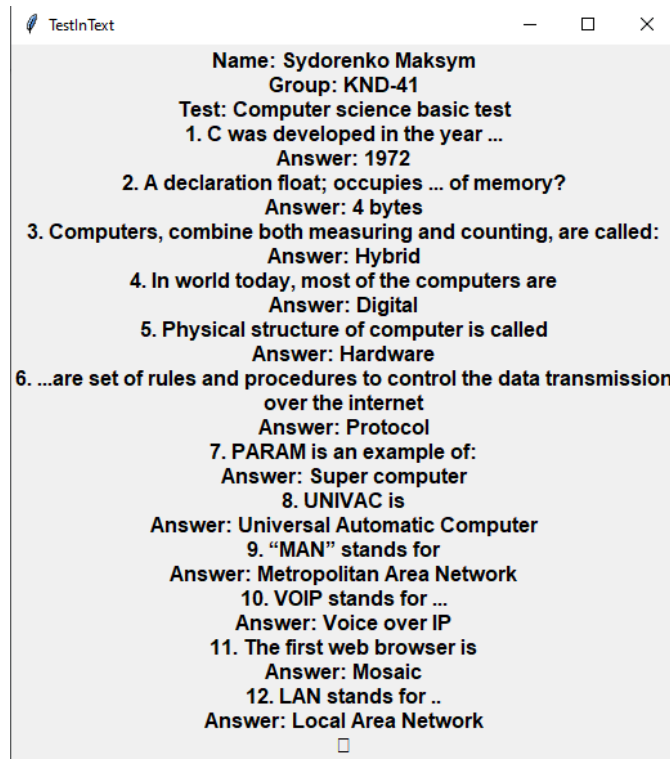


Рисунок 3.6 – Вікно «TestInText» з результатами розпізнавання фото файлу

### 3.2.2 Тестування програмного продукту

Для тестування роботи програного продукту буде використано тест «Neural Network Test», можна побачити на рисунку 3. . Тестування буде проходити за наступних типів тесту:

- Всі відповіді правильні
- Не всі відповіді правильні
- Бланк для відповідей з мобільного пристрою

Всі відповіді правильні;

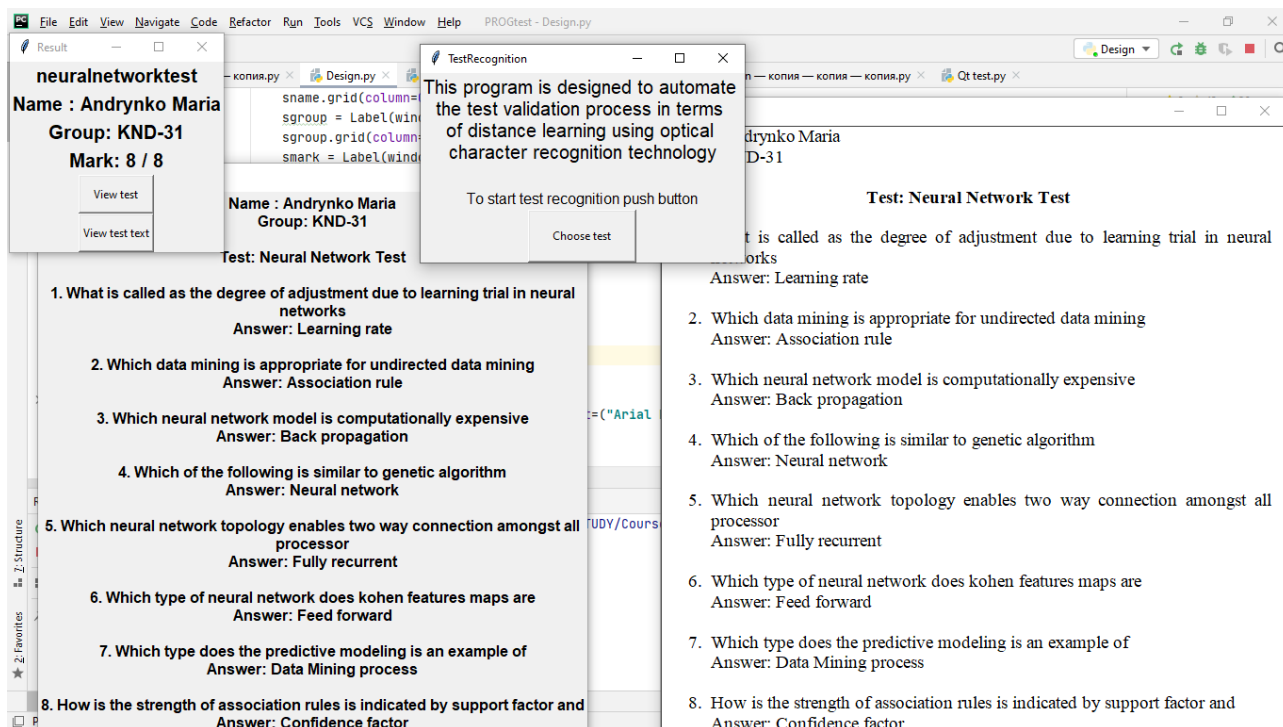


Рисунок 3.7 – Тестування тесту з усіма правильними відповідями

Як видно з рисунку 3.7 у даному випадку всі відповіді в тесту правильні, а результат перевірки тест 8/8, програма правильно розпізнала всі відповіді. Назви тесту, групи студента та ПІБ студента також розпізнано правильно.

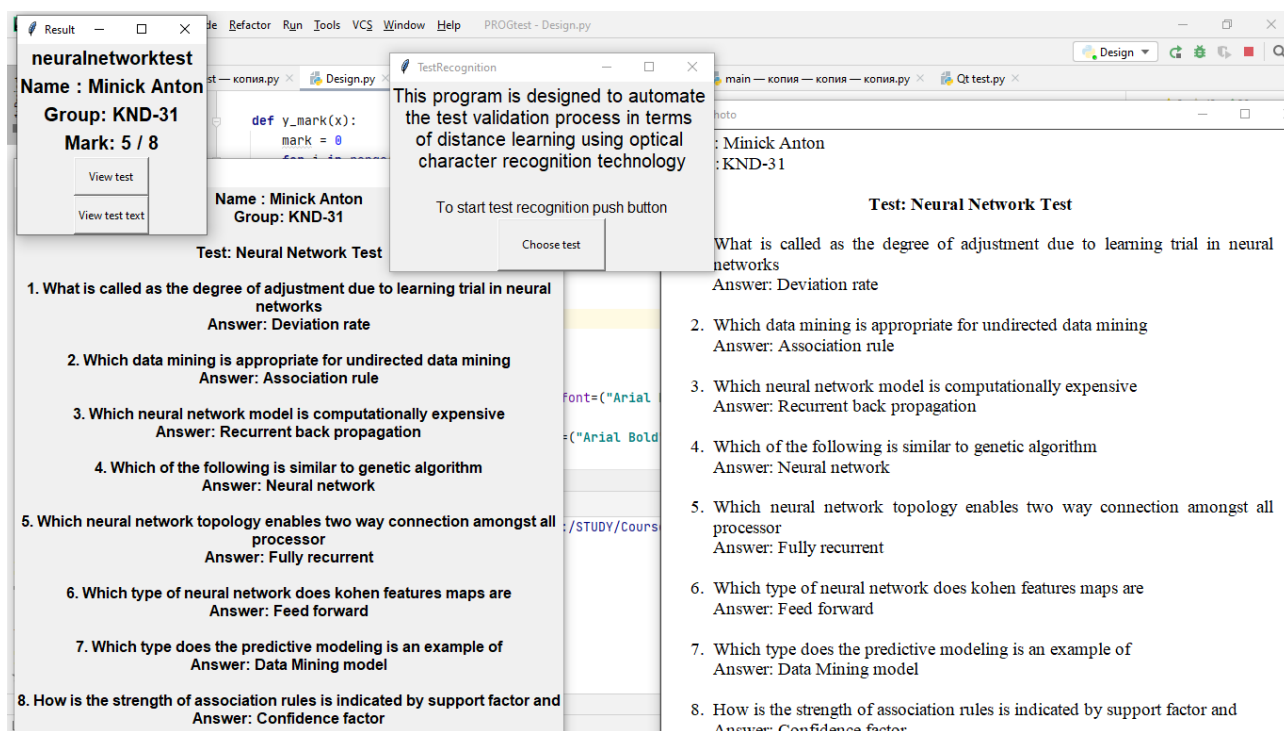


Рисунок 3.8 – Тестування тесту з декількома не правильними відповідями

Як видно з рисунку 3.8 у випадку, де в тесті не всі відповіді правильні, з результату 5/8 видно, що програма правильно розпізнала всі відповіді та відповідно підрахувала оцінку. Назви тесту, групи студента та ПІБ студента також розпізнано правильно.

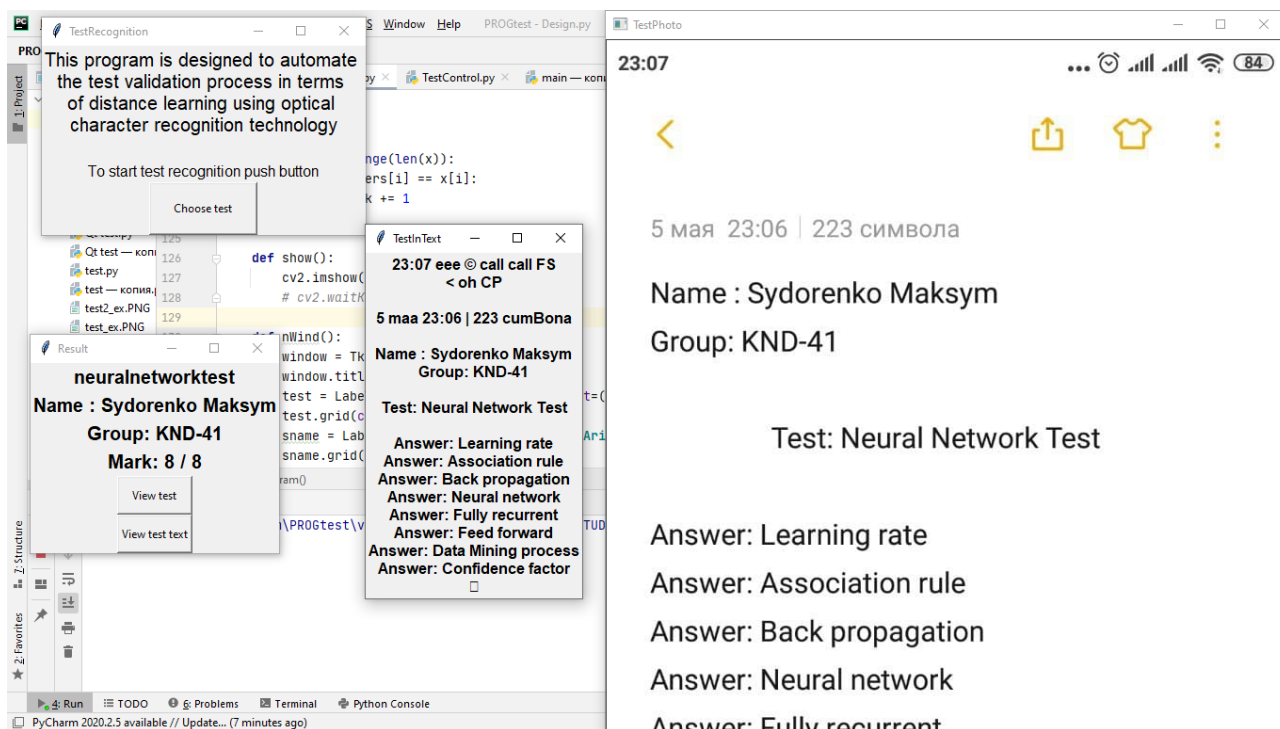


Рисунок 3.9 – Тестування тесту з мобільного пристрою

З рисунку 3.9 у випадку перевірки варіанту з мобільного пристрою програма все правильно розпізнала, знайшла назву тесту ПІБ студента та група студента. Програма успішно пройшла етап тестування без помилок.

## ВИСНОВОК

1. У роботі проаналізовано актуальність віддаленого навчання, особливості засобів для тестового контролю в умовах віддаленого навчання та актуальності самого тестування. Розглянуто наступні засоби тестового контролю: ADTester, ClassMaker, EasyTestMaker та Moodle.

2. За допомогою виявлення особливостей розглянутих засобів тестового контролю, було розроблено вимоги, яким необхідно дотримуватись під час розробки тестів для роботи з програмною для тестового контролю при віддаленому навчання з використанням оптичного розпізнавання символів.

3. Досліджено технологію оптичного розпізнавання символів та її типи: OCR, IWR, ICR, OWR та OMR. Розглянуто методи оптичного розпізнавання символів: оптичне сканування, сегментація, попередня обробка, вилучення ознак, післяобробка.

4. Розроблено можливості діаграм в нотації IDEF0 та діаграми діяльності UML. А також побудовано діаграму в нотації IDEF0 та її декомпозиції та діаграму діяльності UML.

5. Наведені приклади використання баз даних MySQL та процес побудови бази даних необхідної для роботи програми.

6. Наведено та досліджено технології, що необхідні для розробки системи тестування.

7. Наведено деталі розробки засобу тестового контролю при віддаленому навчанні з використанням оптичного розпізнавання символів.

8. Проведено тестування засобу тестового контролю при віддаленому навчанні з використанням оптичного розпізнавання символів.