

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

КЛЮЙКО Анатолій Васильович

**Онлайн словник з іноземної мови/ Online
dictionary on foreing languages**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-42
А. В. Ключко

Науковий керівник:
викладач КРЕПИЧ С.Я.

Бакалаврську дипломну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

РЕЗЮМЕ

Дипломна робота містить 103 сторінки, 41 таблицю, 63 рисунків, список використаних джерел із 20 найменувань та 2 додатків.

Метою дипломної роботи є дослідження та детальний аналіз уже існуючих інтернет-сервісів вивчення іноземної мови і розробці, на основі проведеного аналізу власного інтернет-сервісу, який виключатиме недоліки з розглянутих аналогів.

Об'єктом дослідження є процес вивчення іноземної мови.

Предмет досліджень – застосування сучасних технологій для розробки on-line словника з іноземної мови.

Методи розробки базуються на мові програмування PHP, бази даних MySQL.

Одержані результати полягають в розробці програмної системи «On-line словник з іноземної мови».

Ключові слова: іноземна мова, вивчення слів, тренування, база даних, діаграма класів, реляційна модель бази даних, тестування.

SUMMARY

This thesis contains 103 pages, 41 tables, 63 figures, list of sources with 20 titles and 2 applications.

The aim of the thesis is detailed research and analysis of existing Internet services for foreign language learning and development, based on the analysis of its own Internet service, which exclude the shortcomings of the considered unique.

The object of research is the process of learning a foreign language.

The subject of research is the use of modern technology to develop online foreign language dictionary

The methods of development in programming language PHP, database MySQL.

The results are in development of a software system «On-line Dictionary of foreign languages.»

Keywords: foreign language, learning words, training, database, diagram class, relational database model, testing.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 Аналіз предметної області.....	12
1.1 Коротка характеристика об'єкту управління	12
1.2 Опис предметної області.....	15
1.3 Огляд і аналіз існуючих аналогів	18
1.4 Специфікація вимог до системи.....	26
Висновки до першого розділу	53
РОЗДІЛ 2 ПРОЕКТУВАННЯ WEB-СИСТЕМИ.....	55
2.1 Розроблення програмної архітектури системи	55
2.2. Проектування структури бази даних	59
Висновки до другого розділу.....	63
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	64
3.1. Програмна реалізація проекту.....	64
3.1.1. Загальний опис програмного забезпечення.....	64
3.1.2. Реалізація функцій системи.....	66
3.2. Програмна реалізація бази даних.....	71
Висновки до третього розділу	75
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ.....	76
4.1. Тестування	76
4.1.1. Функціональне тестування.....	76
4.1.2. Тестування безпеки.....	76
4.2 Розгортання програмного продукту.....	78
4.3. Інструкція користувача	79
Висновок до четвертого розділу.....	84
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86
ДОДАТОК А	88
ДОДАТОК Б.....	99

ВСТУП

Актуальність дипломної роботи полягає в тому, що на сьогоднішній день люди витрачають багато часу та грошей на вивчення іноземних мов. Серед причин, які визначають прагнення вивчення мови є: складання іспиту, кар'єрний ріст, закордонні подорожі, спілкування з людьми по інший бік океану, читання новин. Знання іноземної мови часто зустрічається обов'язковим параметром для отримання посади на певні вакансії. Вивчення іноземної мови тренує пам'ять, адже чим більше людина вчиться, тим швидше з часом засвоює інформацію. Іноземна мова відкриває нові можливості для людей.

До популярних методів вивчення іноземної мови можна віднести:

- відвідування приватних занять(репетитор);
- перегляд фільмів;
- читання книг;
- слухання аудіоматеріалів;
- спілкування з людьми, які вільно володіють іноземною мовою тощо.

Кожний із розглянутих методів має ряд переваг у часі, витратах та ефективності. Найкращим варіантом є поєднання всіх цих методів.

Багато сайтів, що допомагають вивчити іноземну мову є частково або повністю платними, розраховані на обмежену кількість днів використання. Не завжди мають зручний та зрозумілий інтерфейс користувача.

Тому розробка веб-сайту, який би надавав можливість вивчення іноземної мови і був би доступний, зрозумілим будь-якому користувачу є актуальною задачею.

Мета роботи полягає в дослідженні та детальному аналізі уже існуючих інтернет-сервісів вивчення іноземної мови і розробці, на основі проведеного

аналізу власного інтернет-сервісу, який виключатиме недоліки з розглянутих аналогів.

При вивченні мови виникають потреби розподіляти слова, що відносяться до деяких категорій (словників). Може виникнути ситуація, коли користувач вивчає декілька іноземних мов одночасно. Для цього йому необхідно використовувати різні словники.

Веб-сайт відповідно до своїх функцій надає користувачам змогу доступу до цих словників та вивчення іноземних слів ще за допомогою проходження тренувань.

Особливістю цього програмного продукту є те, що користувачі можуть використовувати словники не лише для вивчення іноземної мови, а й для інших своїх потреб (наприклад, створення словників антонімів, синонімів, тощо).

Отже, спираючись на мету, можна виділи такі основні задачі, які ставляться до розроблюваного продукту:

1. дослідити існуючі аналоги вивчення іноземної мови, виділити переваги і недоліки кожного з них;
2. на основі аналізу розробити функціональні і не функціональні вимоги;
3. спроектувати базу даних;
4. обрати технологію для створення програмного продукту;
5. реалізація функціональних і не функціональних вимог.

Об'єкт дослідження – процес вивчення іноземної мови.

Предмет досліджень – застосування сучасних технологій для розробки on-line словника з іноземної мови.

Під час створення системи використовуватимемо наступні технології необхідні для розробки інтернет-сервісів:

PHP – скриптова мова, що використовується для створення веб-сторінок на стороні сервера [1].

JavaScript – мова програмування, що використовується для інтерактивності, створення динамічних сторінок.

Sql – мова, яка дає можливість користувачу взаємодіяти із базою даних [8].

Практична цінність розроблюваного інтернет-сервісу полягає в зручності та доступності всім користувачам без обмежень вивчати іноземну мову, без оплати, з необмеженою кількістю словників та слів в них.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Коротка характеристика об'єкту управління

Найбільш поширеними мовами в Україні є : українська, російська, молдовська, угорська, польська, циганська, румунська, болгарська. Серед людей, які спілкуються із людьми за кордоном розповсюдженою є англійська мова.

У Європі школяри вивчають такі мови: англійська, французька, німецька, іспанська, російська. Це становить 95% іноземних мов, що вивчають в Європі.

Вивчення англійської мови переважає у країнах Старого Світу, а також дуже стрімко почалось у країнах Східної Європи. Англійську мову 90% учнів старших класів вивчають як другу мову.

Європейські країни значною мірою штучно підштовхують школярів до вивчення англійської мови. Понад 60% курсів, оплачуваних по лінії спільних програм Євросоюзу, викладаються англійською мовою. Тому влітку британські та ірландські лінгвістичні школи бувають сильно завантажені італійськими, іспанськими і французькими групами школярів, які приїжджають вивчати англійську по лінії програм, що фінансуються Євросоюзом.

Іншими використовуваними мовами цих програм є французька (17%), німецька та іспанська (обидві по 5%).

Німецька та французька вивчаються в європейських середніх школах приблизно в однаковій пропорції. Згідно з історичного поділу на германські та романські групи мов німецька більш популярна серед жителів півночі (Скандинавія, Нідерланди), а французька - серед жителів півдня (Італія, Іспанія).

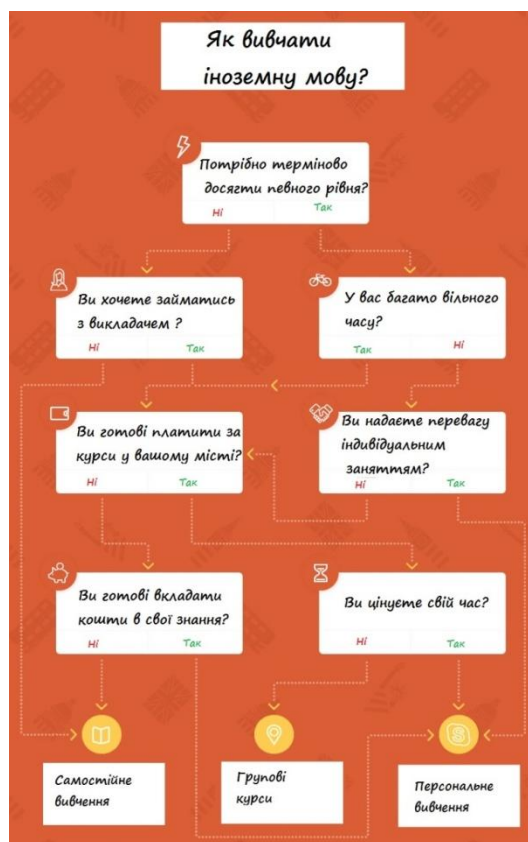
За даними опитування, проведеного Європейською Комісією, як другу мову європейці відзначили англійську (41%), французьку (19%), німецьку (10%), іспанську (7%) і італійську (3%).

Трохи менше половини (47%) серед опитаних зазначили, що вони не знають жодної іншої мови, крім своєї рідної.

На питання, які дві мови, крім своєї рідної європейці знаходять найбільш корисними для вивчення, 75% відзначили англійську, 40% - французьку, 23% - німецьку, 18% - іспанську.

Знання англійської та іспанської мов найбільш часто зазначалося студентами та менеджерами. А французька, як не дивно, згадувався як краща мова до вивчення робітниками і домогосподарками. Німецьку хотіли б вивчати студенти й підприємці, що ведуть власний бізнес.

Є три шляхи вивчення іноземної мови, а саме: самостійне вивчення, групові курси, персональні заняття. Для того, щоб визначити який метод вивчення іноземної мови більш підходить вам, можна скористатись алгоритмом, що зображений на рисунку 1.1.



Рисунк 1.1 – Алгоритм визначення вивчення іноземної мови

Розглянемо переваги і недоліки кожного з методів.

Групові курси.

Переваги:

- підвищення комунікативних навиків;
- доступна ціна;
- можливість вчитись на помилках одногрупників;
- нові знайомства;
- розвиток командних навиків роботи;
- вивчати матеріал за допомогою групового діалогу.

Недоліки:

- більш сміливі люди можуть заважати спілкуватись з викладачем;
- сором'язливим людям важко висловлювати свої думки перед аудиторією;
- термін навчання залежить від рівня знань мови інших людей.

Персональні заняття.

Переваги:

- вам будуть приділяти 100% уваги;
- індивідуальний підхід;
- викладач зможе допомогти позбутись певних прогалин у знаннях;
- різноманітність завдань;

Недоліки:

- вартість;
- визначена тривалість уроку;
- графік навчання залежить від викладача.

Самостійне вивчення.

Переваги:

- безкоштовно;
- самостійний вибір темпу, інтенсивності, графіку навчання;

- самостійний вибір навчання (читання книг, прослуховування аудіоматеріалів, перегляд фільмів, тощо).

Недоліки:

- лінощі;
- багато речей, що відволікають(телевізор, комп'ютерні ігри, розмови);
- не великий мотив;
- відсутність контролю;
- швидкість навчання.

Для певної категорії людей, зокрема школярів, студентів, робітників тощо, які не мають змоги приділяти багато часу на відвідування групових курсів, або мають обмежені грошові ресурси, найкращим варіантом є вивчення іноземної мови самостійним способом. Саме тому актуальною є розробка системи «Онлайн словник з іноземної мови», яка дозволить людині самостійно обирати зручний для неї час на вивчення іноземної мови або її удосконалення

1.2. Опис предметної області.

В життєвому циклі онлайн словника існують такі бізнес процеси як створення словника, додавання нових слів в словник, проходження тренування. Для того, щоб виконати ці можливості, користувач повинен бути авторизованим, що включає процес реєстрації. При виникненні труднощів можна написати повідомлення адміністратору.

Розглянемо детальніше усі процеси. Основним процесом є проходження тренувань, адже це основний спосіб вивчення нових слів. Для того, щоб проходити тестування потрібно мати певний набір слів у словнику. Тому почнемо огляд із бізнес процесу створення словника. В словник додаються слова оригінальною мовою та переклад.

Характеристики бізнес-процесу «Створення словника» наведено в таблиці 1.1.

Таблиця 1.1

Характеристика бізнес-процесу «Створення словника»

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Створення словника
Основні учасники	Авторизований користувач
Вхідна подія	Потреба у новому словнику
Вхідні документи	Немає
Вихідна подія	Новий словник
Вихідні документи	Відсутні

Оскільки створення словника є необхідне для наповнення його словами, тому наступним важливим бізнес процесом є додавання слів в словник.

Характеристики бізнес-процесу «Додавання слів у словник» наведено в таблиці 1.2.

Таблиця 1.2

Характеристика бізнес-процесу «Додавання слів в словник»

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Додавання слів в словник
Основні учасники	Авторизований користувач
Вхідна подія	Потреба у нових словах
Вхідні документи	Немає
Вихідна подія	Нове слово в словнику
Вихідні документи	Відсутні

Найважливішим процесом є проходження тренування. Існують тренування різних видів: «Оригінал - переклад», «Переклад - оригінал»,

«Конструктор», «Випадкові слова». Кожне тренування впливає на покращення вивчення нових слів. Найдоцільніше їх комбінувати.

Характеристики бізнес-процесу «Тренування» наведено в таблиці 1.3.

Таблиця 1.3

Характеристика бізнес-процесу «Тренування»

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Тренування
Основні учасники	Авторизований користувач
Вхідна подія	Потреба у вивченні нових слів
Вхідні документи	Немає
Вихідна подія	Запам'ятовування слів
Вихідні документи	Відсутні

Оскільки всі ці бізнес-процеси не можуть бути задіяні без процесу реєстрації користувача, тому розглянемо це детальніше. Користувач сайту, який має бажання мати власні словники, повинен бути зареєстрований.

Характеристика цього бізнес-процесу представлена в таблиці 1.4.

Таблиця 1.4

Характеристика бізнес-процесу «Реєстрації користувача»

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Реєстрація користувача
Основні учасники	Звичайний користувач
Вхідна подія	Потреба у використанні сервісу
Вхідні документи	Немає
Вихідна подія	Новий користувач
Вихідні документи	Відсутні

1.3. Огляд і аналіз існуючих аналогів.

Першим етапом при проектуванні системи «Онлайн словник з іноземної мови» необхідно дослідити системи аналогічні до проєктованої для виділення основних цікавих характеристик та недоліків які можливо усунути.

Розглянемо три веб-ресурси вивчення іноземної мови, а саме: Busuu, LinguaLeo, WordSteps.

LinguaLeo – ресурс для вивчення іноземної мови, що базується на ігровій схемі. Зараз цей ресурс налічує більше 11 мільйонів користувачів, з яких більше 50% із Європи.

Вивчення іноземної мови полягає у піклуванні про левенятка Лео. Чим більше слів ти вивчаєш – тим більше його годуєш. Тим більший він росте.

Lingualeo.

На рисунку 1.2 проілюстрована екранна форма тренування типу «Переклад – слово».

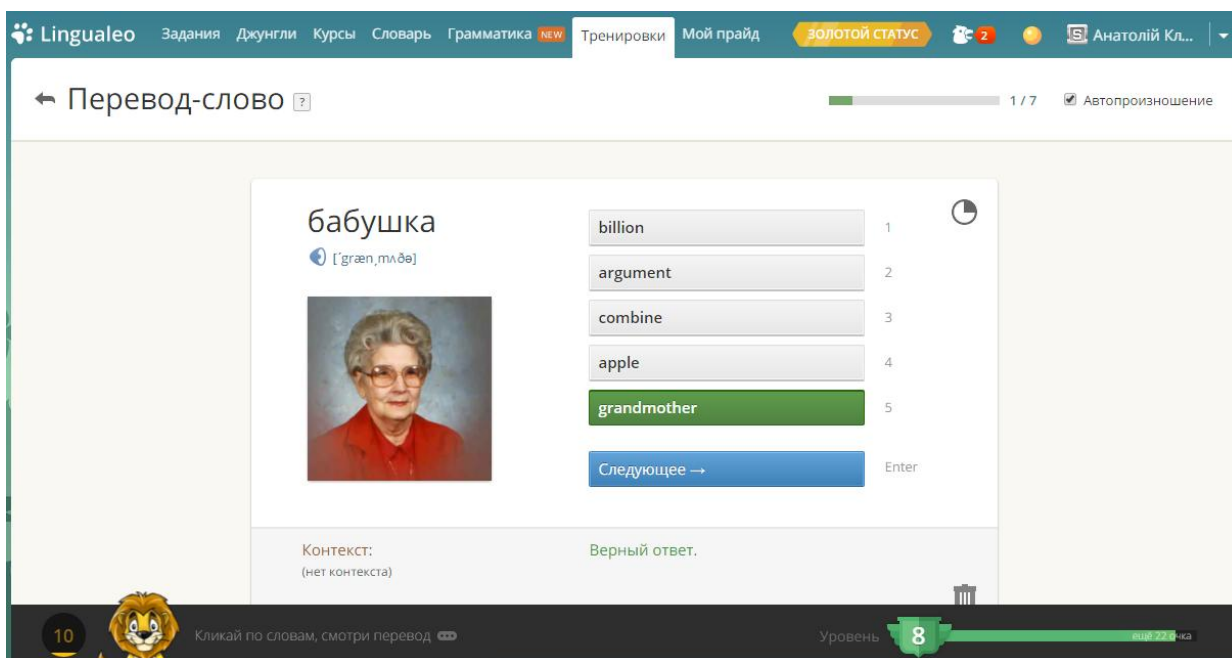


Рисунок 1. 2 – «Переклад – слово»

На представленій екранній формі наведений пункт тренування «Переклад - Слово», де користувачу пропонується переклад слова. Із різних варіантів

потрібно підібрати оригінал слова. Якщо користувач відповідає вірно, то про це повідомляється зеленим кольором, інакше – червоним.

На рисунку 1.3 проілюстрована екранна форма тренування типу «Слово – переклад».

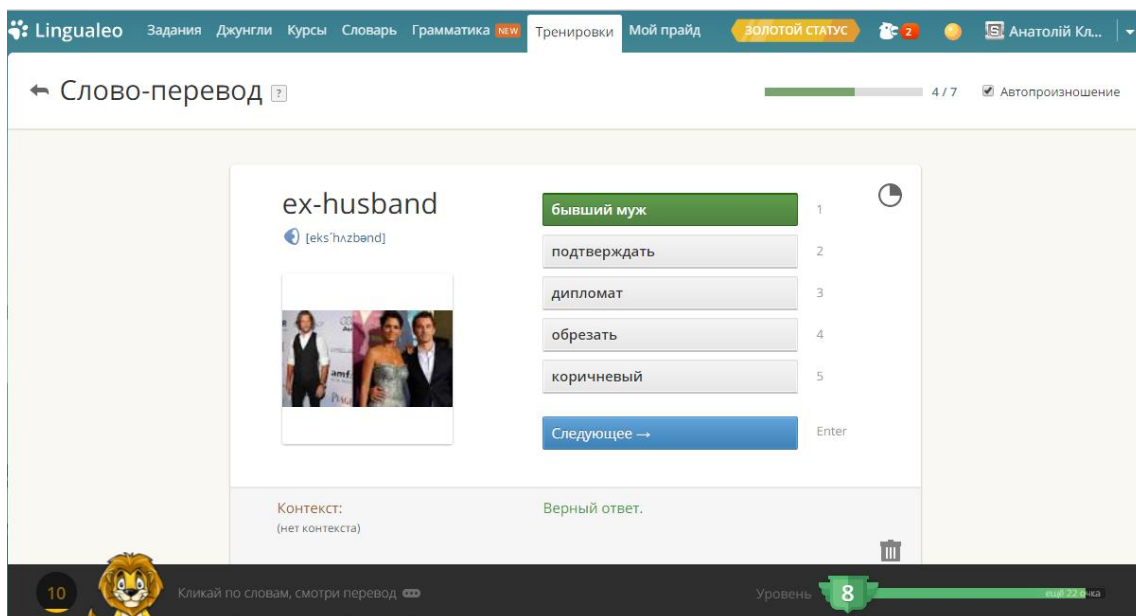


Рисунок 1.3 – «Слово – переклад»

На представлений екранній формі користувачу пропонується оригінал слова, із різних варіантів потрібно підібрати переклад слова. Якщо користувач відповідає вірно, то про це повідомляється зеленим кольором, інакше – червоним.

Ці два тренування є оберненими один до одного, дуже легкі у вивченні нових слів, проте потребують постійного використання для досягнення високої ефективності.

Корисне тренування, що дозволяє запам'ятати слова не візуально, а як вони пишуться. В цьому тренуванні потрібно скласти слова із випадково перемішаних літер. Є два варіанти, як це зробити: натисканням лівої кнопки мишки на відповідну літеру, або написання слова у комірці за допомогою клавіатури.

На рисунку 1.4 проілюстрована екранна форма тренування типу «Конструктор», а на рисунку 1.5 проілюстрована екранна форма тренування типу «Лео-спринт».

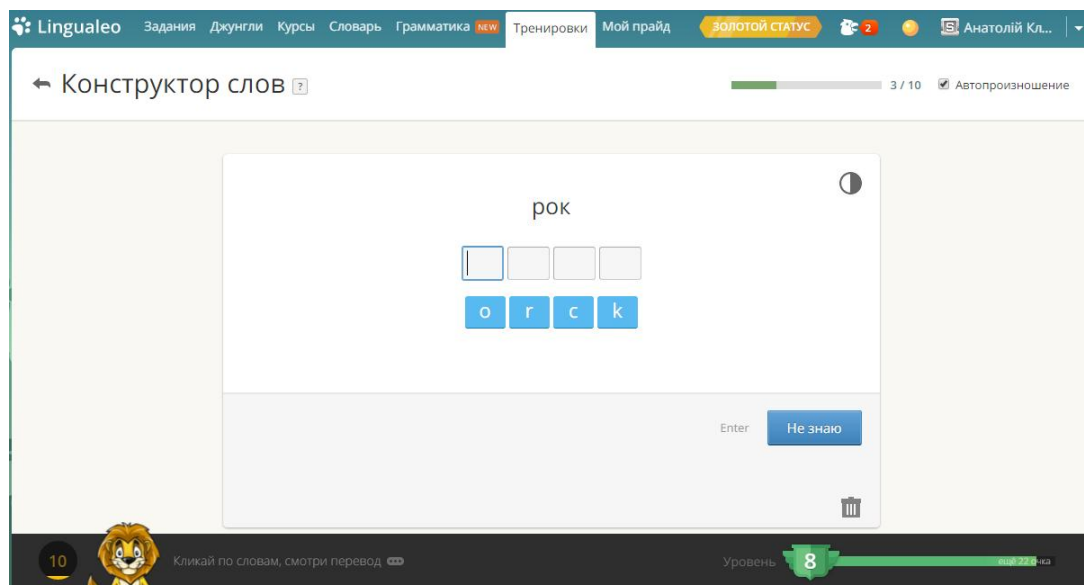


Рисунок 1.4 – «Конструктор»

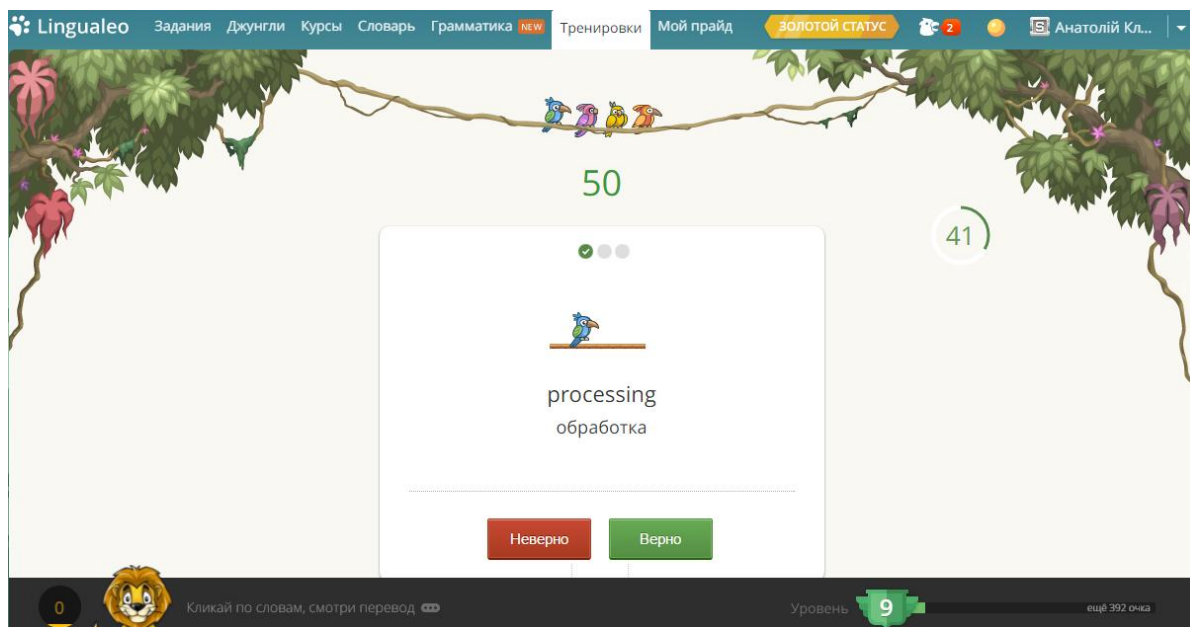


Рисунок 1.5 – «Лео-спринт»

Перш за все хочеться сказати, що недоліком цього тренування є обмеженість використання. Лише 3 рази протягом дня ним можна

скористатись, якщо не купувати «Золотий статус». Ідея тренування полягає у тому, що протягом одної хвилини з'являється слово оригінальною мовою і два переклади цього слова, з яких один є вірний. Потрібно вибрати правильний переклад, при цьому нараховуються бали. Протягом хвилини потрібно набрати якнайбільше балів. Ця вправа дозволяє збільшити швидкість відтворення слів.

Busuu.

Функціональністю, яка представляє цей ресурс вивчення іноземної мови з хорошої сторони, є проходження уроків по рівнях: A1, A2, B1, B2. Кожен користувач має змогу повторити чи вивчити для себе новий рівень. Кожен рівень має свій список уроків. Список уроків для рівня A1 проілюстровано на рисунку 1.6.

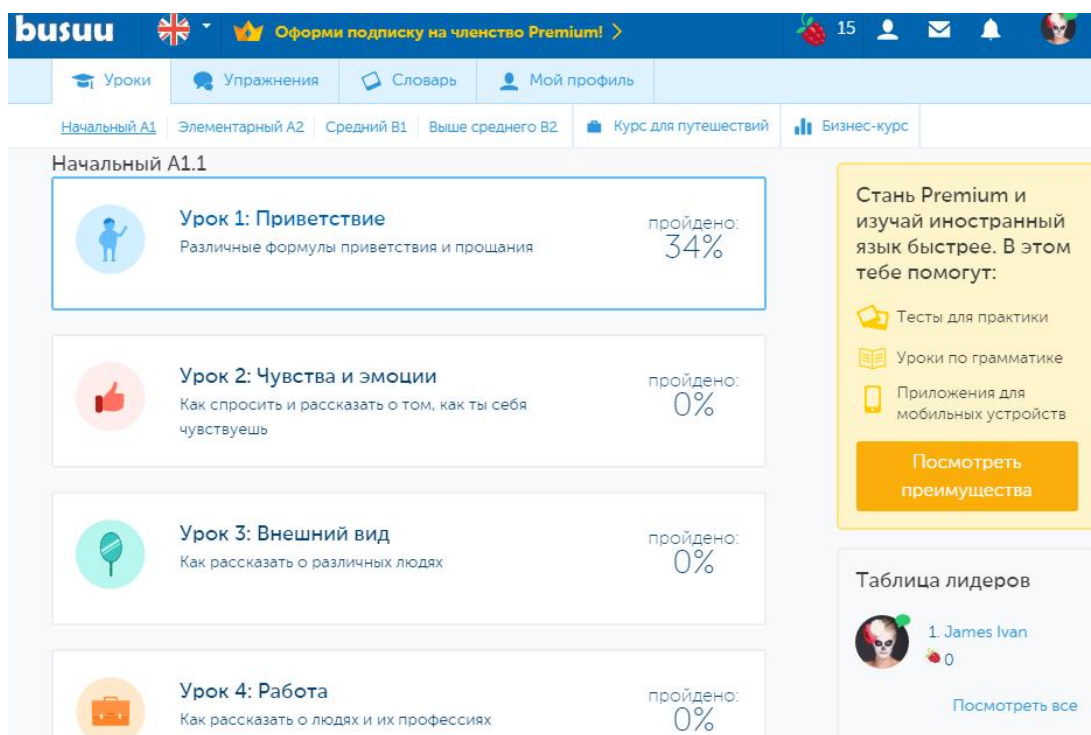


Рисунок 1.6 – Уроки рівня A1.

Корисні для себе слова з уроку можна додавати відразу в свій словник. Проходження уроку являє собою голосовий та візуальний супровід слова

оригінальною мовою, що дозволяє використовувати різні рівні пам'яті людини, для запам'ятовування.

Ще одною цікавою можливістю є наявність «живого» спілкування з людьми. Можна відкрити чат з іншими користувачами та спільно вивчати мову чи просто поспілкуватись з людиною, яка мешкає по інший бік океану.

Приклад проходження уроку рівня А1 проілюстровано на рисунку 1.7, а на рисунку 1.8 проілюстровано як виглядає вікно спілкування.

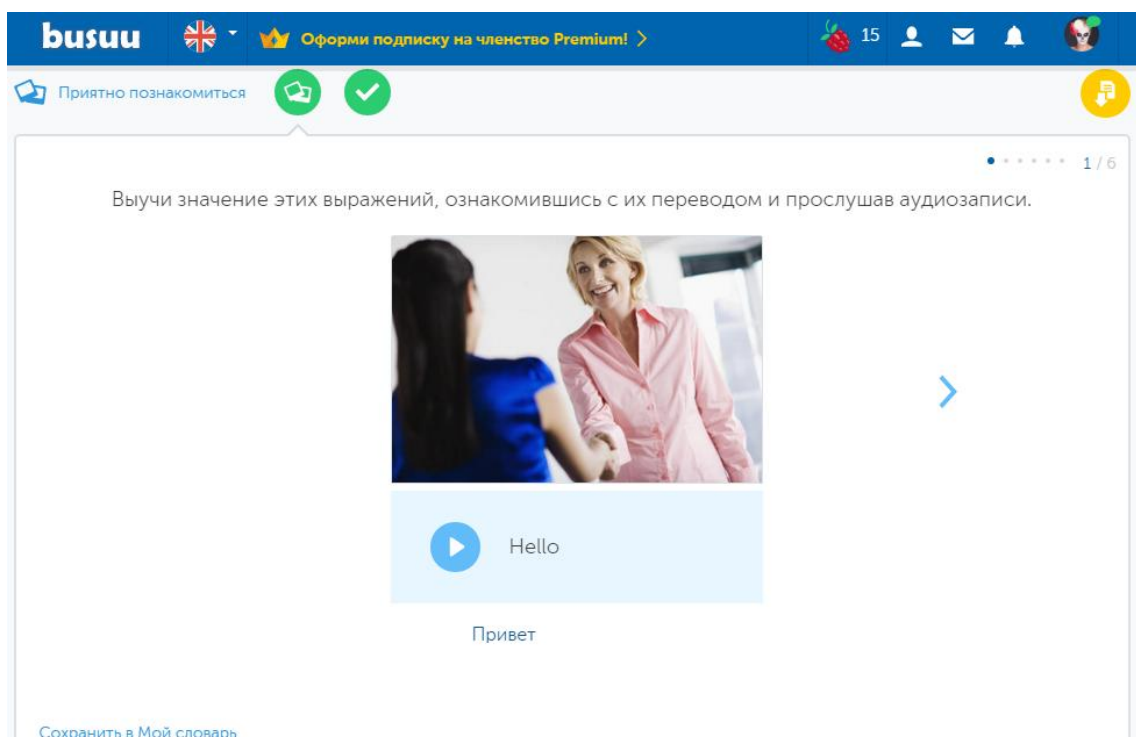


Рисунок 1.7 – Проходження уроку рівня А1.

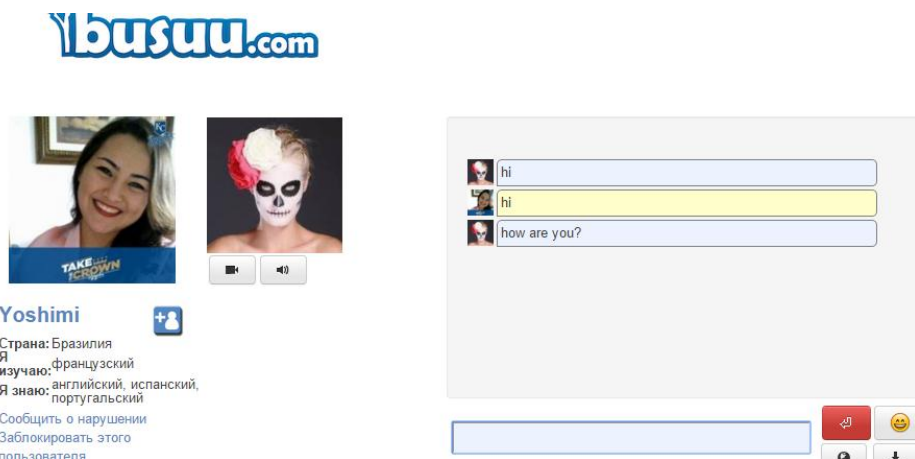


Рисунок 1.8 – Чат

Недолік цього ресурсу в тому, що пропонує платні курси для подорожей та бізнесу. Ці курси груповані по темах. На рисунку 1.9 проілюстровано список доступних бізнес-курсів.

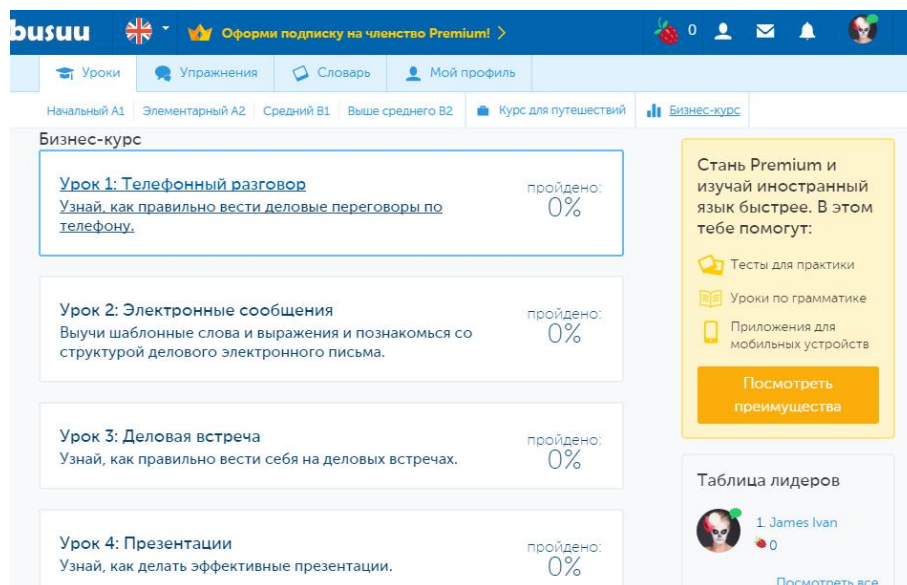


Рисунок 1.9 – Бізнес-курси

WordSteps.

Корисною особливістю цього ресурсу є те, що користувач сам визначає вивчив він слово чи ні. Процес проходження тренування вивчення слова проілюстровано на рисунку 1.10.

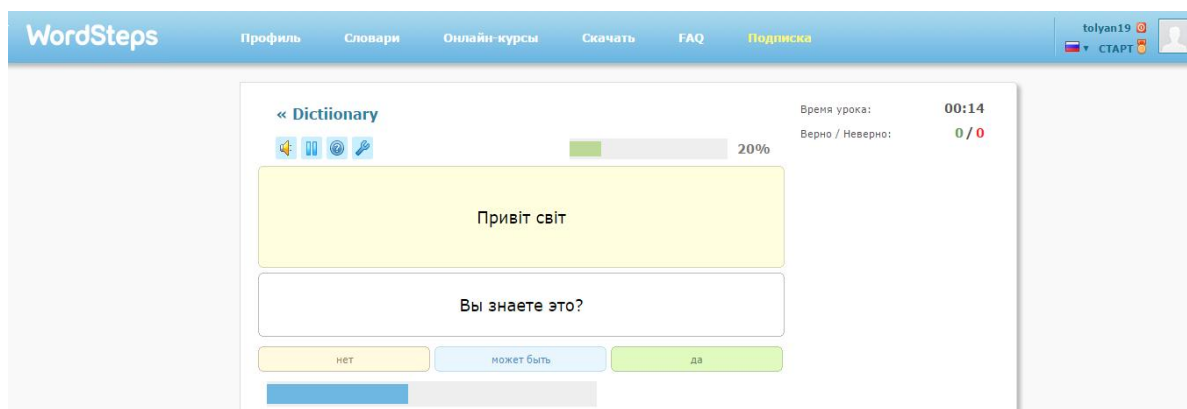


Рисунок 1.10 – Процес вивчення слова

Перед початком тренуванням, користувач має можливість вибрати словники, які до нього будуть входити. Під час тренування користувачу з'являються слова оригінальною мовою та переклад. Після цього з'являється лише переклад слова. Користувач повинен вибрати один із запропонованих варіантів: не знає, можливо знає, знає. Коли користувач визначився із кожним словом, йому пропонується правильно підібрати до перекладу слова його оригінал(див. рис. 1.11).

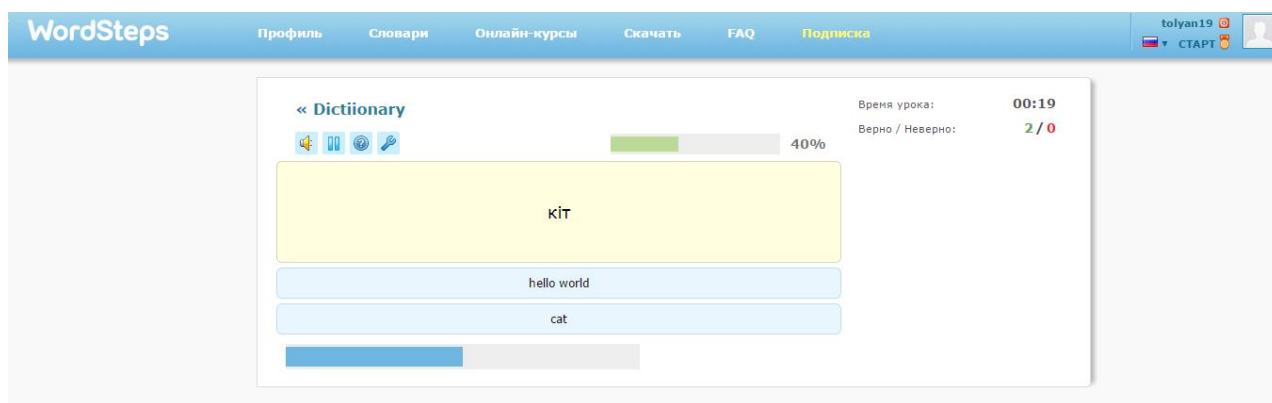


Рисунок 1.11 – Процес проходження тренування.

Цей ресурс вивчення іноземної мови надає можливість переглядати матеріали (аудіо, відео, текст) за різними категоріями (фільми, жарти, мистецтво, хобі і таке інше). При чому можна вибрати складність матеріалу та мови (оригінал-переклад), які нам потрібні. На рисунку 1.12 проілюстровано матеріали по темі «Об'єкти і властивості»

Недоліком цього ресурсу є те, що він надає платні курси (як *lingualeo*), список курсів наведено на рисунку 1.13, ціни вказані в порівняльній таблиці 1.5. Також WordSteps має досить незручний інтерфейс. Коли перший раз користуєшся сайтом, часто виникають питання: як створити словник? Як пройти тестування? А що я можу ще зробити на сайті?

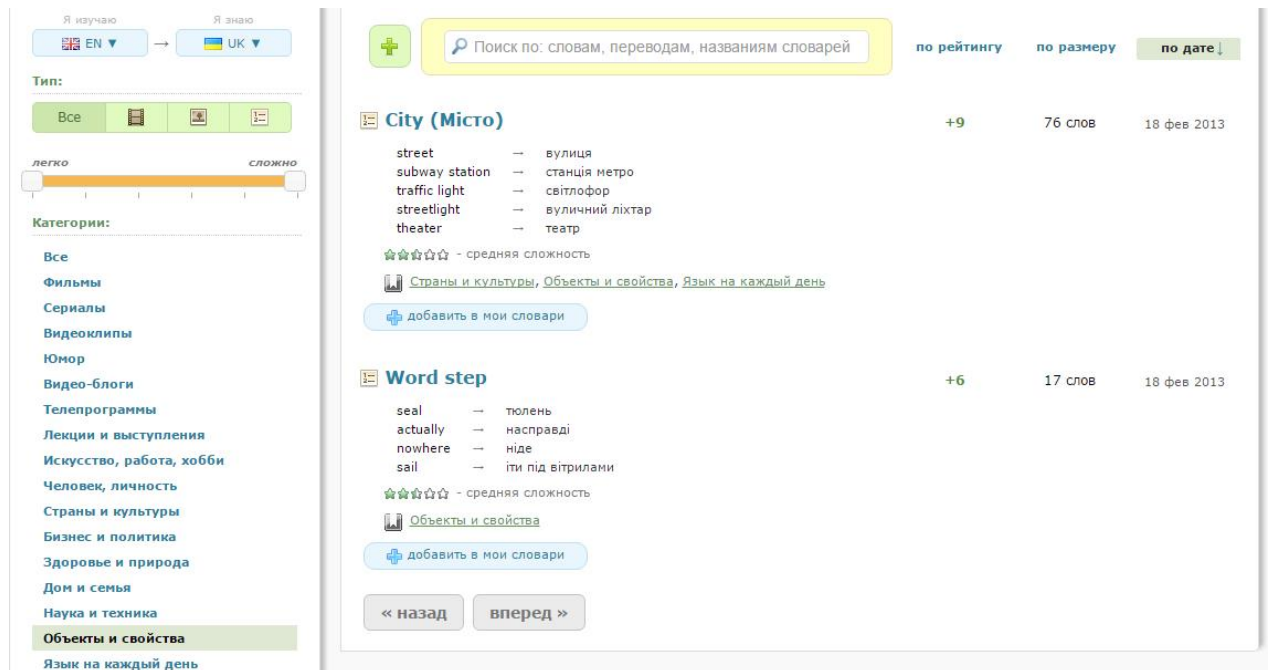


Рисунок 1.12 – «Об’єкти і властивості»

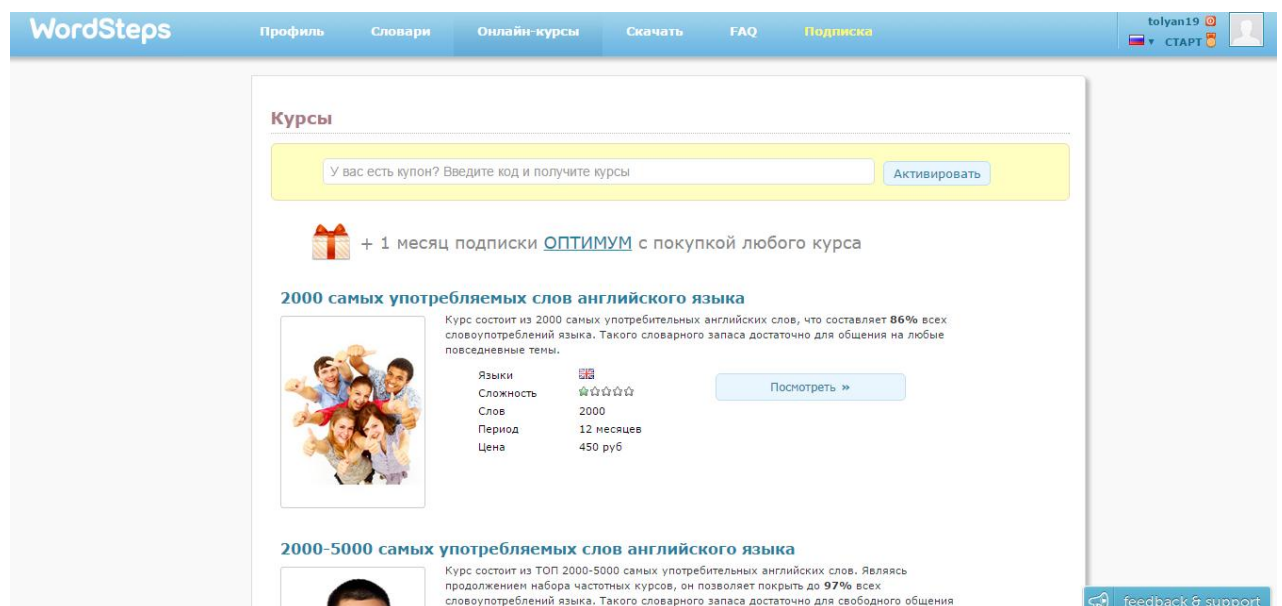


Рисунок 1.13 – Платні «онлайн-курси».

Зробивши аналіз існуючих аналогів можна скласти наступну таблицю 1.5.

Таблиця 1.5

Порівняльна таблиця

Ресурс	lingualeo.com	busuu.com	wordsteps.com
--------	---------------	-----------	---------------

Продовження таблиці 1.5

Мобільний додаток	існує	існує	існує
Інтерфейс користувача	Зручний, інтуїтивно зрозумілий.	Добрий, інтуїтивно зрозумілий.	Початківцю важко зрозумілий
Допомога користувачу	Пасивна*, готова база відповідей	Пасивна, готова база відповідей	Пасивна
Вартість продукту	330 грн / рік	1 міс / €15, 6 міс / €10, 12 міс / €8.83 24 міс / €	«Старт» безкоштовно, «Оптимум» 90руб/міс, «Максимум» 150руб/міс
Кількість вивчення мов	1	12	31
Мультимовність	підтримується	підтримується	підтримується

*Пасивна допомога – допомога у режимі офлайн, відсутність онлайн консультанта.

1.4. Специфікація вимог до системи.

Специфікація вимог для програмної системи - це повний опис поведінки системи що розробляється. Вона включає множину прецедентів які описують всі взаємодії, які користувачі мають з програмним забезпеченням. Прецеденти також відомі як функціональні вимоги. На додачу до прецедентів також включає нефункціональні вимоги. Нефункціональні вимоги є вимогами які накладають обмеження на проект, чи реалізацію. Специфікація вимог до системи включає: глосарій проекту, опис варіантів використання. У таблиці 1.6 подано глосарій.

Глосарій

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
Веб – система	Клієнт – серверний програмний продукт. Клієнтом є браузер, сервером – веб-сервер.
Авторизація	Надання особі певних прав на виконання дій. Також перевірка цих прав при спробі виконати дії.
Словник	Список слів, що відносяться до словника.
Слово	Складається із двох частин: оригінал – слово оригінальною мовою, переклад – переклад слова.
Матеріал(Відеоматеріал)	Посилання на відео ресурс.
Тренування	Процес, за допомогою якого відбувається вивчення іноземної мови.
Додатки	Словники, матеріали створенні адміністратором.
2. Користувачі системи	
Користувач	Людина, яка користується системою.
Не авторизований користувач	Користувач, який не пройшов авторизацію.
Авторизований користувач	Користувач, який пройшов авторизацію.
Адміністратор	Користувач, який керує системою, наділений особливими правами доступу.

Розкадровка варіантів використання.

Оскільки в системі присутні певні «Актори» (дійові особи, які виконують певну функцію), тому кожен з цих акторів може виконувати різні дії. Для прикладу «хто-що» робить прийнято використовувати діаграму варіантів використання. Діаграма варіантів використання проілюстрована на

рисунок 1.14.

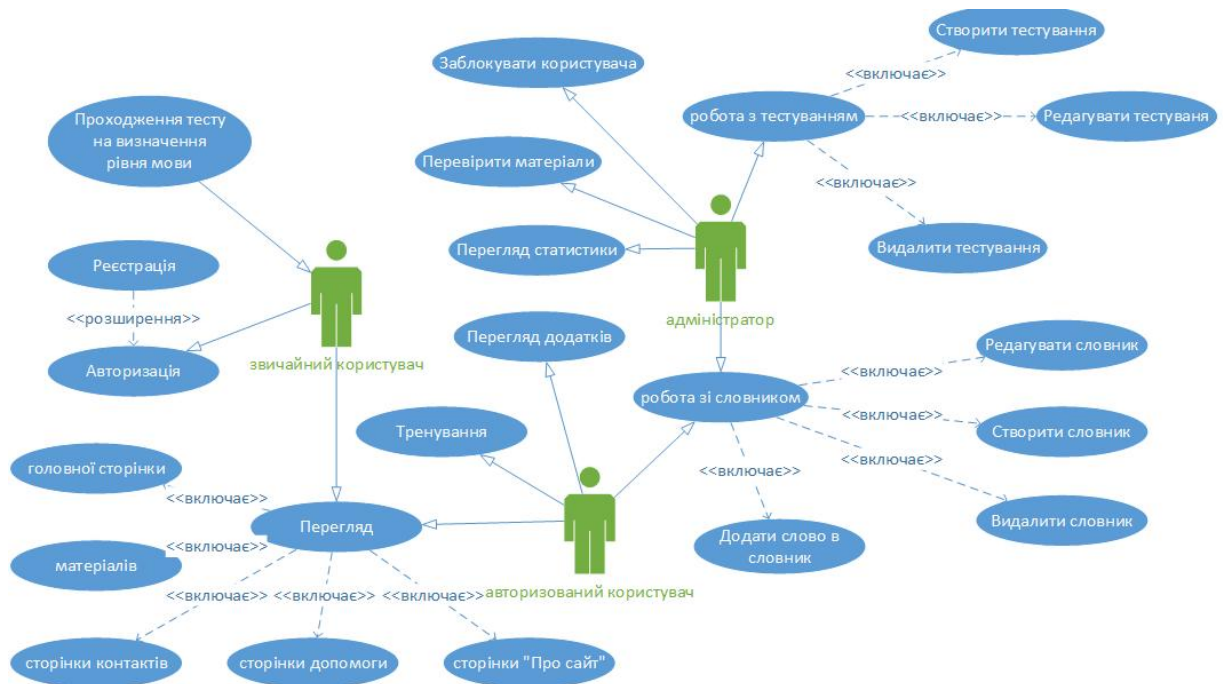


Рисунок 1.14 – Діаграма варіантів використання.

Прототипи функцій для звичайного та авторизованого користувача:

Прототип функції «Авторизація» проілюстровано на рисунку 1.15, а її варіант використання наведено у таблиці 1.7. Ця функція дозволяє не авторизованому користувачу авторизуватись в системі. Для цього необхідно ввести email та пароль, які були вказані при реєстрації, та натиснути кнопку «Увійти».

Форма авторизації

Email
Пароль
<i>Увійти</i>

Рисунок 1.15 – Прототип функції «Авторизація»

Варіант використання «Авторизація»

Контекст використання	Авторизація
Дійові особи	Не авторизований користувач, адміністратор
Передумова	Користувач зареєстрований і знаходиться на сторінці авторизації
Тригер	Кнопка «Увійти»
Сценарій	1.Заповнити поля у формі даними реєстрації 2.Підтвердити натиснувши на відповідну кнопку
Пост-умова	Користувач авторизований

Прототип функції «Реєстрація» проілюстровано на рисунку 1.16, а її варіант використання наведено у таблиці 1.8. Ця функція дозволяє користувачу створити свій акаунт. Для цього необхідно ввести свій email адрес, ім'я, із випадючого списку вибрати мову та її рівень знань, ввести пароль та натиснути кнопку «Зареєструватись».

Форма реєстрації

email

ім'я

Мова ▼

Рівень ▼
Дізнатись свій рівень можете тут

Пароль

Повторіть пароль

Зареєструватись

Рисунок 1.16 –Прототип функції «Реєстрація»

Варіант використання «Реєстрація»

Контекст використання	Реєстрація
Дійові особи	Не зареєстрований користувач
Передумова	Відкрита сторінка реєстрації

Тригер	Кнопка «Зареєструватись»
Сценарій	1.Заповнити поля форми реєстрації 2.Підтвердити натиснувши на відповідну кнопку
Пост-умова	Користувач зареєстрований

Кожен користувач сайту може дізнатись свій рівень знань з певної мови. Для цього йому необхідно перейти на сторінку визначення рівня мови та пройти там тестування. Це тестування складається із 21 запитання. Після відповідей на всі запитання та натиснення кнопки «Відправити» користувачу буде показано його результат. Прототип функції «Визначення рівня мови» проілюстровано на рисунку 1.17, а її варіант використання наведено у таблиці 1.9.

Питання 1:

Варіант1
Варіант2
Варіант3
Варіант4

Питання 2:

Варіант1
Варіант2
Варіант3
Варіант4

Питання3 :

Варіант1
Варіант2
Варіант3
Варіант4

Рисунок 1.17 – Прототип функції «Визначення рівня мови»

Таблиця 1.9

Варіант використання «Визначення рівня мови»

Контекст використання	Визначення рівня мови
Дійові особи	Звичайний або авторизований користувач
Передумова	Відкрита сторінка визначення рівня мови

Тригер	Кнопка «Відправити»
Сценарій	1.Відповісти на всі питання 2.Підтвердити натиснувши на відповідну кнопку
Пост-умова	Результат тесту

Кожен користувач системи може відкрити сторінку «Про сайт». На цій сторінці розміщується інформація про сайт, а саме: як ним користуватись, для кого був розроблений, інша важлива інформація для користувачів. Прототип функції «Перегляд сторінки “Про сайт” » проілюстровано на рисунку 1.18, а її варіант використання наведено у таблиці 1.10.

Рисунку 1.18 показує прототип веб-сторінки «Про сайт». Верхня частина містить навігаційне меню з пунктами: «Логотип», «Допомога», «Матеріали», «Про сайт», «Контакти» та «Логотип». Основна частина сторінки розділена на три колонки. Ліва колонка містить два відеоматеріали та список термінів: «cat - кіт», «dog - собака», «animal - тварина», «black - чорний». Центральна колонка містить текстовий опис сайту. Права колонка містить форму авторизації з полями «Email», «Пароль» та кнопкою «Увійти», а також форму реєстрації з полями «email», «Ім'я», «Мова», «Рівень», паролем, повторенням пароля та кнопкою «Зареєструватись».

Рисунок 1.18 – Прототип функції «Перегляд сторінки “Про сайт”»

Таблиця 1.10

Варіант використання «Перегляд сторінки “Про сайт” »

Контекст використання	«Перегляд сторінки “Про сайт” »
Дійові особи	Будь – який користувач

Передумова	Немає
Тригер	Запит сторінки «Про сайт»
Сценарій	Відкрити сторінку «Про сайт»
Пост-умова	Завантажена сторінка

Кожен користувач системи може відкрити сторінку «Допомога». На цій сторінці користувач може написати лист адміністратору чи розробнику сайту із своїм запитанням, пропозицією чи будь якою іншою причиною. Для цього користувачу необхідно ввести текст повідомлення, вказати свій email адрес, на який буде відправлена відповідь та натиснути кнопку «Відправити». Прототип функції «Перегляд сторінки “Допомога”» проілюстровано на рисунку 1.19, а її варіант використання наведено у таблиці 1.11.

Логотип	Допомога	Матеріали	Про сайт	Контакти	Логотип
Відеоматеріал	Текст...			Форма авторизації	
<i>cat - кіт</i> <i>dog - собака</i> <i>animal - тварина</i> <i>black - чорний</i>	Email			<input type="text" value="Email"/> <input type="text" value="Пароль"/> <input type="button" value="Увійти"/>	
Відеоматеріал	Відправити			Форма реєстрації	
				<input type="text" value="email"/> <input type="text" value="Ім'я"/> <input type="text" value="Мова"/> ▼ <input type="text" value="Рівень"/> ▼ Дізнатись свій рівень можете тут <input type="text" value="Пароль"/> <input type="text" value="Повторіть пароль"/> <input type="button" value="Зареєструватись"/>	

Рисунок 1.19. Прототип функції «Перегляд сторінки “Допомога”»

Варіант використання «Перегляд сторінки «Допомога» »

Контекст використання	Перегляд сторінки «Допомога»
Дійові особи	Будь – який користувач
Передумова	Немає
Тригер	Запит сторінки «Допомога»
Сценарій	Відкрити сторінку «Допомога»
Пост-умова	Завантажена сторінка

Кожен користувач системи може відкрити сторінку «Контакти». На цій сторінці сайту розміщені контакти, через які можна зв'язатись із розробником чи адміністратором сайту. Прототип функції «Перегляд сторінки «Контакти»» проілюстровано на рисунку 1.20, а її варіант використання наведено у таблиці 1.12.

Логотип	Допомога	Матеріали	Про сайт	Контакти	Логотип
<p>Відеоматеріал</p> <p>cat - кіт dog - собака animal - тварина black - чорний</p> <p>Відеоматеріал</p>	<p>Телефон: email:</p>			<p>Форма авторизації</p> <p>Email <input type="text"/></p> <p>Пароль <input type="password"/></p> <p><input type="button" value="Увійти"/></p> <p>Форма реєстрації</p> <p>email <input type="text"/></p> <p>Ім'я <input type="text"/></p> <p>Мова <input type="text"/></p> <p>Рівень <input type="text"/></p> <p>Дізнатись свій рівень можете тут</p> <p>Пароль <input type="password"/></p> <p>Повторіть пароль <input type="password"/></p> <p><input type="button" value="Зареєструватись"/></p>	

Рисунок 1.20 – Прототип функції «Перегляд сторінки «Контакти»»

Варіант використання «Перегляд сторінки «Контакти» »

Контекст використання	Перегляд сторінки «Контакти»
Дійові особи	Будь – який користувач
Передумова	Немає
Тригер	Запит сторінки «Контакти»
Сценарій	Відкрити сторінку «Контакти»
Пост-умова	Завантажена сторінка

Користувачі сайту можуть користуватися матеріалами, які пропонують інші користувачі. Під матеріалом розуміється посилання на відеоресурс для вивчення іноземної мови. Дані відеоресурси (матеріали) перевіряються адміністратором. Щоб запропонувати свій матеріал необхідно відкрити сторінку із матеріалами, у відповідне поле ввести посилання на відеоресурс та натиснути кнопку «Відправити». Всі запропоновані матеріали користувачами з'являються у розділі матеріалів після підтвердження їх адміністратором. Прототип функції «Додавання матеріалів» проілюстровано на рисунку 1.21, а її варіант використання подано у таблиці 1.13.

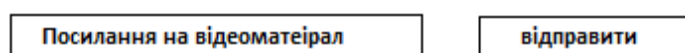


Рисунок 1.21 – Прототип функції «Додавання матеріалів»

Варіант використання «Додавання матеріалів »

Контекст використання	Додавання матеріалів
Дійові особи	Будь – який користувач
Передумова	Відкрита сторінка матеріалів
Тригер	Кнопка «Відправити».

Сценарій	1.Написати свій матеріал 2.Підтвердити, натиснувши на відповідну кнопку
Пост-умова	Результат додавання матеріалу

Кожен користувач системи може відкрити сторінку «Матеріали». На цій сторінці сайту розміщені матеріали, запропоновані іншими користувачами. Якщо користувач вважає, що певний відеоматеріал не відповідає тематиці сайту, то він може написати лист, вказавши посилання на відеоматеріал та причину невідповідності. Прототип функції «Перегляд матеріалів» проілюстровано на рисунку 1.22, а її варіант використання наведено у таблиці 1.14.



Рисунок 1.22 – Прототип функції «Перегляд матеріалів»

Таблиця 1.14

Варіант використання «Перегляд матеріалів»

Контекст використання	Перегляд матеріалів
Дійові особи	Будь – який користувач
Передумова	Немає
Тригер	Запит сторінки «Матеріали»

Сценарій	Відкрити сторінку «Матеріали»
Пост-умова	Завантажена сторінка

Кожен користувач системи може відкрити головну сторінку. На цій сторінці сайту розміщено основну інформацію про сайт, останні новини. Прототип функції «Перегляд головної сторінки» проілюстровано на рисунку 1.23, а її варіант використання наведено у таблиці 1.15.

Рисунок 1.23 – Прототип функції «Перегляд головної сторінки»

Таблиця 1.15

Варіант використання «Перегляд головної сторінки»

Контекст використання	Перегляд головної сторінки
Дійові особи	Будь – який користувач
Передумова	Немає
Тригер	Запит до головної сторінки
Сценарій	Відкрити головну сторінку
Пост-умова	Завантажена сторінка

Авторизований користувач може створювати для себе довільну кількість словників, в яких зберігати слова. Для того, щоб створити словник, необхідно відкрити сторінку створення словника, ввести його назву, натиснути кнопку «Додати». Прототип функції «Створити словник» проілюстровано на рисунку 24, а її варіант використання наведено у таблиці 1.16.

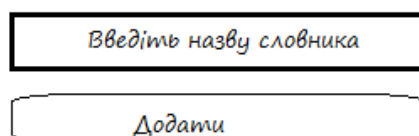


Рисунок 1.24 – Прототип функції «Створити словник»

Таблиця 1.16

Варіант використання «Створити словник»

Контекст використання	Створити словник
Дійові особи	Авторизований користувач
Передумова	Авторизація, відкрита сторінка додавання словника
Тригер	Кнопка «Додати».
Сценарій	1.Ввести назву словника. 2.Натиснути кнопку «Додати»
Пост-умова	Результат створення словника

Авторизований користувач зберігає свої слова в словниках. Щоб додати слово в словник, перш за все потрібно створити словник, якщо немає жодного словника. Після цього відкрити сторінку додавання слів в словник, вибрати із випадуючого списку словник, в який буде додане слово, ввести оригінал слова та його переклад, натиснути кнопку «Додати». У кожному словнику може зберігатись довільна кількість слів. Прототип функції «Додати слово в словник» проілюстровано на рисунку 1.25, а її варіант використання наведено у таблиці 1.17.

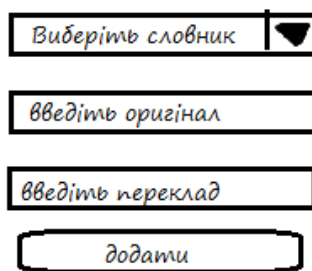


Рисунок 1.25 – Прототип функції «Додати слово в словник»

Таблиця 1.17

Варіант використання «Додати слово в словник»

Контекст використання	Додати слово в словник
Дійові особи	Авторизований користувач
Передумова	Авторизація, відкрита сторінка додавання слова в словник
Тригер	Кнопка «Додати».
Сценарій	1.Вибрати словник. 2.Ввести оригінал слова. 3.Ввести переклад слова 4.Підтвердити, натиснувши на відповідну кнопку
Пост-умова	Слово додано в словник.

Буває потреба внести зміни у вже існуючий словник. Наприклад, змінити переклад слова. Для цього існує функція «Редагувати словник». Щоб редагувати словник, потрібно відкрити відповідну сторінку та зробити наступні кроки: із випадаючого списку вибрати потрібний словник, внести зміни в словник, зберегти результат. Прототип функції «Редагувати словник» проілюстровано на рисунках 1.26 – 1.27, а її варіант використання наведено у таблиці 1.18.

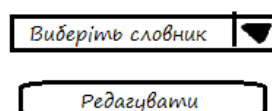


Рисунок 1.26 – Прототип функції «Редагувати словник»

оригінал	Переклад
cat	кіт
dog	собака

зберегти

Рисунок 1.27 – Прототип функції «Редагувати словник»

Таблиця 1.18

Варіант використання «Редагувати словник»

Контекст використання	Редагувати словник
Дійові особи	Авторизований користувач
Передумова	Авторизація, відкрита сторінка редагування словника
Тригер	Кнопка «Редагувати» та кнопка «Зберегти»
Сценарій	1.Вибрати словник. 2.Натиснути кнопку «Редагувати». 3.Внести зміни в словник. 4.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Редагований словник.

Авторизований користувач може видалити свої раніше створені словники. У цьому випадку разом із словником видаляються всі слова, які до нього належали. Щоб видалити словник, необхідно відкрити відповідну сторінку, вибрати із випадаючого списку словник для видалення і натиснути кнопку «Видалити». Прототип функції «Видалити словник» проілюстровано на рисунку 1.28, а її варіант використання наведено у таблиці 1.19.

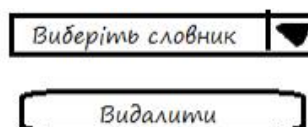


Рисунок 1.28 – Прототип функції «Видалити словник»

Варіант використання «Видалити словник»

Контекст використання	Видалити словник
Дійові особи	Авторизований користувач
Передумова	Авторизація, відкрита сторінка видалення словника
Тригер	Кнопка «Видалити».
Сценарій	1.Вибрати словник 2.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Словник видалений.

Для вивчення слів з іноземної мови авторизований користувач може проходити тренування наступних видів: «Оригінал – переклад», «Переклад – оригінал», «Конструктор», «Випадкові слова». Для цього необхідно перейти на відповідну сторінку тренування. Прототип функції «Тренування» проілюстровано на рисунку 1.29, а її варіант використання наведено у таблиці 1.20.

Тестування	Заробити балів
Оригінал - переклад	
Перекла - оригінал	
Конструктор	
Випадкові слова	

Рисунок 1.29 – Прототип функції «Тренування»

Варіант використання «Тренування»

Контекст використання	Тренування
Дійові особи	Авторизований користувач
Передумова	Авторизація, відкрита сторінка тренування
Тригер	Одне із списку тренувань.

Сценарій	Вибрати тип тренування
Пост-умова	Відкрите тренування.

Функція «Перегляд додатків» дозволяє користувачу додавати до свої словників слова із його рівнем знань. Ці спеціальні словники створює адміністратор. Вони дозволяють користувачу зменшити час на додавання потрібних слів, адже певний список словників буде доступний відразу після реєстрації. Прототип функції «Перегляд додатків» проілюстровано на рисунку 1.30, а її варіант використання наведено у таблиці 1.21.

<i>Тварини Додати собі</i>	<i>Кольори Додати собі</i>
<i>cat-кіт</i>	<i>Red-червоний</i>
<i>dog-пес</i>	<i>green-зелений</i>
	<i>black-чорний</i>

Рисунок 1.30 – Прототип функції «Перегляд додатків»

Таблиця 1.21

Варіант використання «Перегляд додатків»

Контекст використання	Перегляд додатків.
Дійові особи	Авторизований користувач.
Передумова	Авторизація.
Тригер	Запит сторінки «Перегляд додатків».
Сценарій	Відкрити сторінку «Перегляд додатків».
Пост-умова	Завантажена сторінка.

Прототипи функцій та варіанти використання для адміністратора:

Адміністратор може створювати довільну кількість словників, назви яких повинні бути унікальні. Для цього йому необхідно ввести назву словника, вибрати мову та рівень, для яких буде створено словник та натиснути кнопку

«Зберегти». Прототип функції «Створити словник» проілюстровано на рисунку 1.31, а її варіант використання наведено у таблиці 1.22.

Прототип функції «Створити словник» складається з трьох елементів керування, розташованих вертикально, та кнопки «ЗБЕРЕГТИ» знизу:

- Перший елемент: текстовий поле з підписом «назва словника».
- Другий елемент: комбодомовий список з підписом «Виберіть мову» та стрілкою вниз.
- Третій елемент: комбодомовий список з підписом «Виберіть рівень» та стрілкою вниз.
- Четвертий елемент: кнопка з підписом «ЗБЕРЕГТИ».

Рисунок 1.31 – Прототип функції «Створити словник»

Таблиця 1.22

Варіант використання «Створити словник»

Контекст використання	Створити словник
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка створення словника.
Тригер	Кнопка «Зберегти».
Сценарій	1.Ввести назву словника. 2.Вибрати мову. 3.Вибрати рівень. 4.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Створений словник.

Якщо адміністратору необхідно внести зміни у вже існуючі словники, то він може скористатись функцією «Редагувати словник». Для того, щоб редагувати словник, необхідно його вибрати у випадяючому списку та натиснути кнопку «Редагувати». Після цього з’являється сторінка, на якій можна вносити зміни. Прототип функції «Редагувати словник» проілюстровано на рисунках 1.32 – 1.33, а її варіант використання наведено у таблиці 1.23.

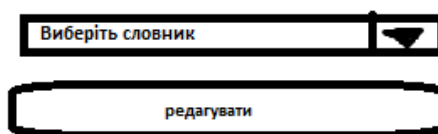


Рисунок 1.32 – Прототип функції «Редагувати словник»

оригінал	Переклад
cat	кіт
dog	собака

зберегти

Рисунок 1.33 – Прототип функції «Редагувати словник»

Таблиця 1.23

Варіант використання «Редагувати словник»

Контекст використання	Редагувати словник
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка редагування словника.
Тригер	Кнопка «Редагувати».
Сценарій	1.Вибрати словник. 2.Натиснути на кнопку «Редагувати». 3.Внести зміни в словник. 4.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Редагований словник.

Для того, щоб адміністратору видалити вже існуючий словник, йому необхідно зайти на сторінку видалення словників, вибрати із випадаючого списку потрібний словник та натиснути кнопку «Видалити». Прототип функції «Видалити словник» проілюстровано на рисунку 1.34, а її варіант використання наведено у таблиці 1.24.

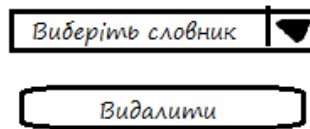


Рисунок 1.34 – Прототип функції «Видалити словник»

Таблиця 1.24

Варіант використання «Видалити словник»

Контекст використання	Видалити словник
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка видалення словника
Тригер	Кнопка «Видалити».
Сценарій	1.Вибрати словник. 2.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Словник видалений.

Адміністратор може додавати в словники необмежену кількість слів. Для цього йому необхідно перш за все створити словник, якщо його не існує. Наступним кроком потрібно вибрати цей словник, вказати оригінал та переклад слова і натиснути на кнопку «Зберегти». Прототип функції «Додати слово в словник» проілюстровано на рисунку 1.35, а її варіант використання наведено у таблиці 1.25.



Рисунок 1.35 – Прототип функції «Додати слово в словник»

Варіант використання «Додати слово в словник»

Контекст використання	Додати слово в словник
Дійові особи	Адміністратор.
Передумова	Авторизація, створений словник, відкрита сторінка додавання слова в словник.
Тригер	Кнопка «Зберегти».
Сценарій	1.Вибрати словник. 2.Ввести оригінал. 3.Ввести переглад. 4.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Слово додалось в словник.

Будь які користувачі можуть пропонувати свої відеоматеріали. Після перевірки адміністратором на їх коректність, приймається рішення: про додання їх або відхилення. Для того, щоб додати – матеріали, адміністратору необхідно зайти на сторінку матеріалів, позначити її, відмітивши позначкою та натиснути кнопку «Підтвердити». Для відміни всіх заявок натискається кнопка «Видалити всі заявки». Прототип функції «Перевірити матеріал» проілюстровано на рисунку 1.36, а її варіант використання наведено у таблиці 1.26.

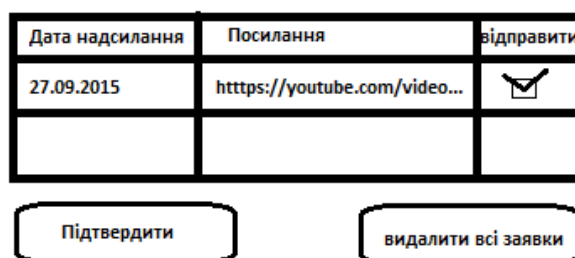


Рисунок 1.36 – Прототип функції «Перевірити матеріал»

Варіант використання «Перевірити матеріал»

Контекст використання	Перевірити матеріал.
Дійові особи	Адміністратор.
Передумова	Авторизація, відкрита сторінка перевірки матеріалів.
Тригер	Кнопки «Підтвердити» та «Видалити всі заявки».
Сценарій	1.Відмітити матеріали. 2.Підтвердити натиснувши на відповідну кнопку.
Пост-умова	Матеріали підтверджено.

Прототип функції «Перегляд статистики» проілюстровано на рисунку 1.37, а її варіант використання наведено у таблиці 1.27. У цьому розділі сайту виводиться статистика сайту. А саме: кількість користувачів у системі, кількість існуючих словників, кількість слів, кількість відеоматеріалів, тощо.

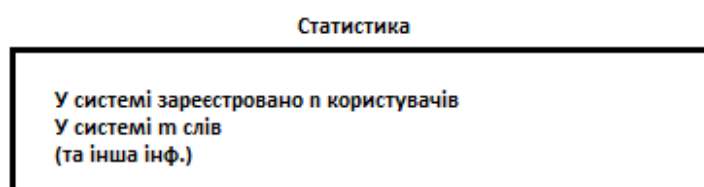


Рисунок 1.37 – Прототип функції «Перегляд статистики»

Варіант використання «Перегляд статистики»

Контекст використання	Перегляд статистики
Дійові особи	Адміністратор.
Передумова	Авторизація.
Тригер	Запит сторінки «Статистика».
Сценарій	Відкрити сторінку «Статистика».
Пост-умова	Завантажена сторінка.

З певних причин буває потрібно заборонити користувачу доступ до сайту. Для цієї можливості адміністратор може скористатись функцією «Заблокувати користувача». Для цього необхідно відкрити відповідну сторінку, відмітити із списку користувача(або декілька користувачів), якого необхідно заблокувати та натиснути кнопку «Заблокувати». Прототип функції «Заблокувати користувача» проілюстровано на рисунку 1.38, а її варіант використання наведено у таблиці 1.28.

Ім.я	Email	Позначити
hunter	hunter@gmail.com	<input checked="" type="checkbox"/>

Заблокувати

Рисунку 1.38 – Прототип функції «Заблокувати користувача»

Таблиця 1.28

Варіант використання «Заблокувати користувача»

Контекст використання	Заблокувати користувача
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка блокування користувачів
Тригер	Кнопка «Заблокувати».
Сценарій	1. Відмітити користувачів. 2. Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Користувач заблокований.

Створення тестів відбувається для того, щоб користувачі могли пройти тестування на визначення рівня мови. Щоб додати тест, необхідно перш за все вказати назву мови, до якої він буде належати. Наступним кроком потрібно ввести питання та відповіді до них, з яких останній варіант вказати як

правильний та натиснути кнопку «Зберегти». Кожна мова має один тест, який складається із 21-го питання. Прототип функції «Створити тест» проілюстровано на рисунку 1.39, а її варіант використання наведено у таблиці 1.29.

Мова:

Додайте тест:

1. Питання:

A

B

C

Правильна відповідь

D

21 Питання:

A

B

C

Правильна відповідь

D

Рисунок 1.39 – Прототип функції «Створити тест»

Таблиця 1.29

Варіант використання «Створити тест»

Контекст використання	Створити тест
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка створення тесту
Тригер	Кнопка «Зберегти».
Сценарій	1.Ввести мову, яку визначатиме тест 2.Ввести питання та відповіді тесту. 2.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Результат створення тесту.

Для того, щоб зробити зміни в тесті на визначення рівня мови, необхідно вибрати мову, до якої належить тестування та натиснути на кнопку «Редагувати

тест». Після цього з'явиться сторінка, на якій можна вносити зміни в тест. Прототип функції «Редагувати тест» проілюстровано на рисунках 1.40 – 1.41, а варіант її використання наведено у таблиці 1.30.

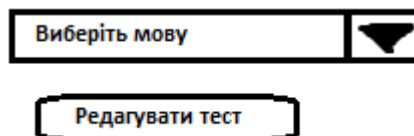


Рисунок 1.40 – Прототип функції «Редагувати тест»

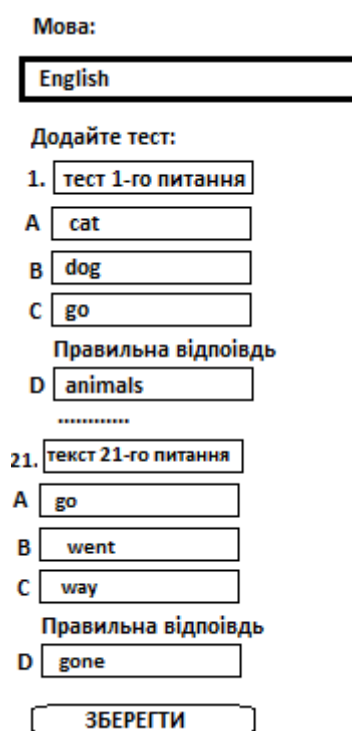


Рисунок 1.41 – Прототип функції «Редагувати тест»

Таблиця 1.30

Варіант використання «Редагувати тест»

Контекст використання	Редагувати тест.
Дійові особи	Адміністратор.
Передумова	Авторизація, відкрита сторінка редагування тесту.
Тригер	Кнопка «Редагувати тест».

Сценарій	1.Вибрати мову. 2.Внести зміни в тест. 3.Підтвердити, натиснувши на відповідну кнопку.
Пост-умова	Змінений тест.

Оскільки кожен тест на визначення рівня мови, належить до конкретної мови, то щоб його видалити, необхідно вибрати мову, до якої він належить. Прототип цієї функції «Видалити тест» проілюстровано на рисунку 1.42, а варіант її використання наведено у таблиці 1.31.



Рисунок 1.42 – Прототип функції «Видалити тест»

Таблиця 1.31

Варіант використання «Видалити тест»

Контекст використання	Видалити тест
Дійові особи	Адміністратор
Передумова	Авторизація, відкрита сторінка видалення тесту
Тригер	Кнопка «Видалити».
Сценарій	1.Вибрати мову. 2.Підтвердити, натиснувши на відповідну кнопку
Пост-умова	Тест видалений.

Специфікацію функціональних та не функціональних вимог наведено у таблицях 1.37 та 1.38.

Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1	Перегляд головної сторінки	Обов'язкове	Низька	Користувач, Адміністратор
2	Перегляд сторінки «Контакти»	Обов'язкове	Низька	Користувач, Адміністратор
3	Перегляд сторінки «Допомога»	Обов'язкове	Низька	Користувач, Адміністратор
4	Перегляд сторінки «Про сайт»	Обов'язкове	Низька	Користувач, Адміністратор
5	Реєстрація	Обов'язкове	Середня	Користувач, Адміністратор
6	Авторизація	Обов'язкове	Середня	Користувач, Адміністратор
7	Визначення рівня знань мови	Обов'язкове	Середня	Користувач, Адміністратор
8	Перегляд матеріалів	Обов'язкове	Низька	Користувач, Адміністратор
9	Додавання матеріалів	Обов'язкове	Середня	Користувач
10	Тренування	Обов'язкове	Висока	Користувач
11	Перегляд додатків	Обов'язкове	Середня	Користувач
12	Створити словник	Обов'язкове	Середня	Користувач

Продовження таблиці 1.37

13	Додати слово в словник	Обов'язкове	Середня	Користувач
14	Редагувати словник	Обов'язкове	Висока	Користувач
15	Видалити словник	Обов'язкове	Низька	Користувач
16	Створити словник(адміністратор)	Обов'язкове	Середня	Адміністратор
17	Редагувати словник(адміністратор)	Обов'язкове	Висока	Адміністратор
18	Видалити словник(адміністратор)	Обов'язкове	Середня	Адміністратор
19	Додати слово в словник(адміністратор)	Обов'язкове	Середня	Адміністратор
20	Перевірити матеріал	Обов'язкове	Середня	Адміністратор
21	Перегляд статистики	Обов'язкове	Низька	Адміністратор
22	Заблокувати користувача	Обов'язкове	Низька	Адміністратор
23	Створити тестування	Обов'язкове	Висока	Адміністратор
24	Редагувати тестування	Обов'язкове	Висока	Адміністратор
25	Видалити тестування	Обов'язкове	Середня	Адміністратор

Значення функціональних вимог:

При використанні деяких функціональних вимог необхідно буде пройти авторизацію користувачу сайту і адміністратору. Функції, які доступні лише після авторизації описані у пункті прототипи функцій.

Специфікація не функціональних вимог

№	Назва вимоги	Характеристики
1	Застосовність	
1.1	Час навчання користувачів	Не більше 20 хв.
1.2	Час відгуку	Не більше 30 с.
2	Надійність	
2.1	Середній час безвідмовної роботи	7 днів.
	Максимальна норма помилок або дефектів	10.
3	Робочі характеристики	
3.1	Швидкодія для транзакції(середнє значення)	1-2с.
3.2	Місткість	1000 користувачів.
4	Проектні обмеження	
4.1	Мова програмування	PHP
4.2	Framework	Laravel
5	Документація	Наявність коротких, зрозумілих інструкцій користувачу
6	Інтерфейс користувача	Веб-застосунок

Значення не функціональних вимог:

Цей проект буде реалізовано за допомогою мови програмування PHP з використанням бази даних MySQL.

Висновки до першого розділу:

1. Результатом даного розділу є специфікація вимог до програмного продукту.
2. Було проведено порівняння трьох ресурсів для вивчення іноземної мови: Lingualeo, Bussu, WordSteps.

3. Розглянуто усі недоліки і переваги цих ресурсів, які були враховані в подальшій розробці.
4. Сформовано всі функціональні та не функціональні вимоги до розроблюваного програмного продукту.

РОЗДІЛ 2 ПРОЕКТУВАННЯ WEB-СИСТЕМИ

2.1 Розроблення програмної архітектури системи

Для вирішення бізнес проблеми, була вибрана клієнт-серверна архітектура. Вона найбільш придатна для задач саме такого типу, коли потрібно забезпечити доступ до системи багатьом користувачам. Найголовнішим є те, що клієнт-серверна архітектура надає можливість віддаленого доступу.

Проект, який побудований на клієнт-серверній архітектурі, повинен складатися з клієнта, сервера та каналу зв'язку [10].

Основне завдання сервера полягає у наданні інформації клієнтам, що звертаються до них. Клієнти безпосередньо звертаються до серверів через канал зв'язку. Клієнтом є кожен користувач. Клієнти не залежні один від одного. Сервер же ж обробляє запити від усіх користувачів, тобто від всіх клієнтів. Кожен клієнт не знає, про існування інших клієнтів. На рисунку 2.1 проілюстровано структуру клієнт-серверної архітектури [2].

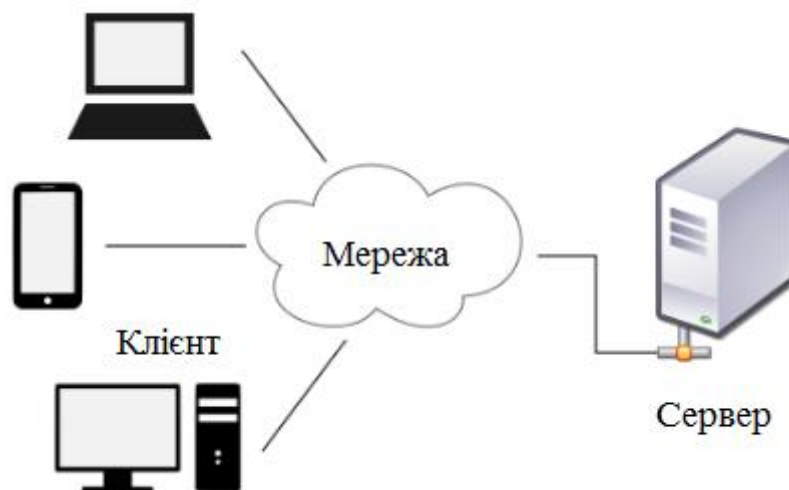


Рисунок 2.1 – Клієнт-серверна архітектура

У нашому випадку клієнтом є браузер користувача. В ньому буде відображатись результат відповіді сервера. Користувач через браузер буде

робити запити відповідно до своїх потреб. Обробка запитів відбувається на стороні сервера. Тут відбувається вся бізнес-логіка програмної системи, результати якої віддаються клієнтові [11].

На рисунку 2.2 проілюстровано IDEF0 модель для роботи адміністратора, розглянуто функцію перевірки матеріалів. А на рисунку 2.3 проілюстровано DFD модель цього ж процесу. Це дозволяє нам графічно описати бізнес-процеси.

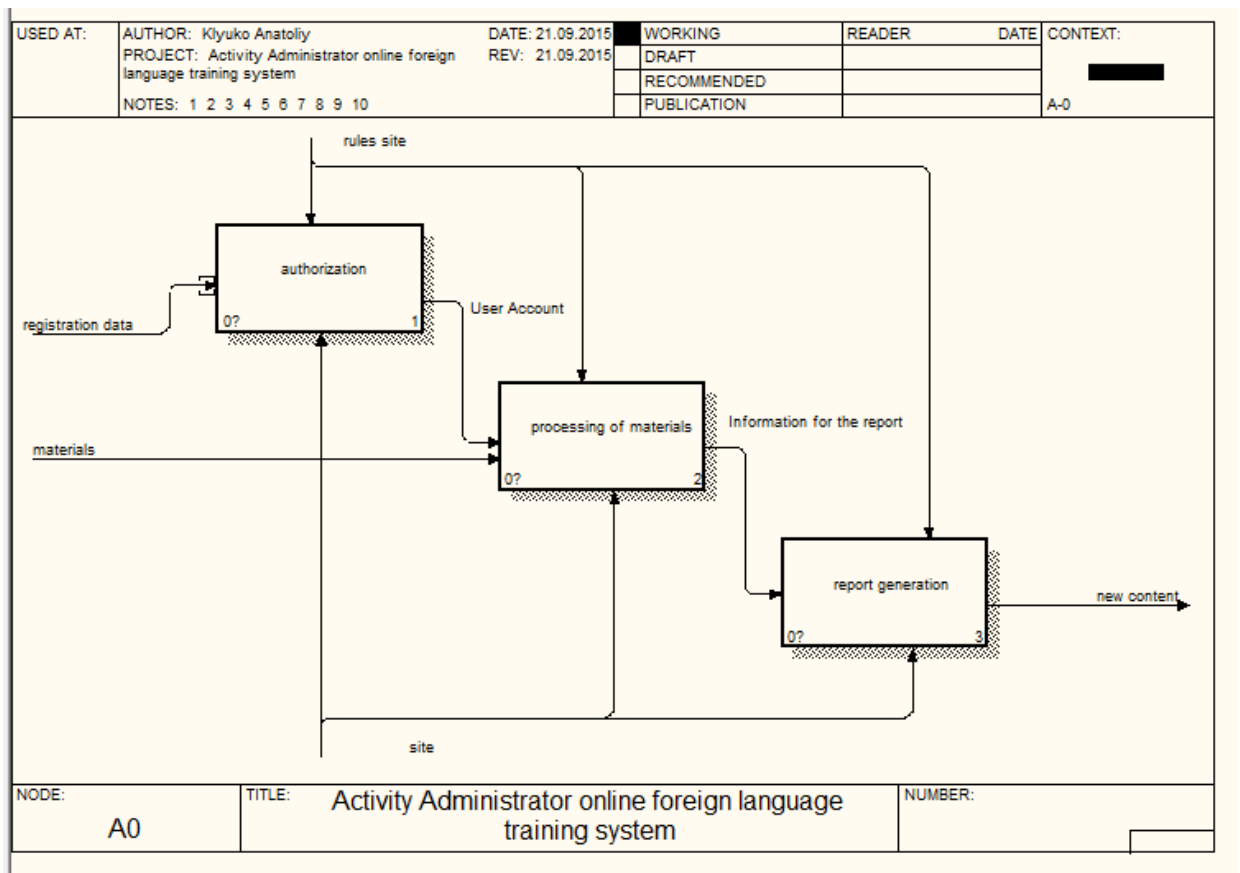


Рисунок 2.2 – Модель IDEF0

DFD описує:

- функції обробки інформації (роботи);
- зовнішні посилання, які забезпечують інтерфейс із зовнішніми об'єктами, що знаходяться за межею модельованої системи;
- таблиці для зберігання документів (сховища даних) [15].

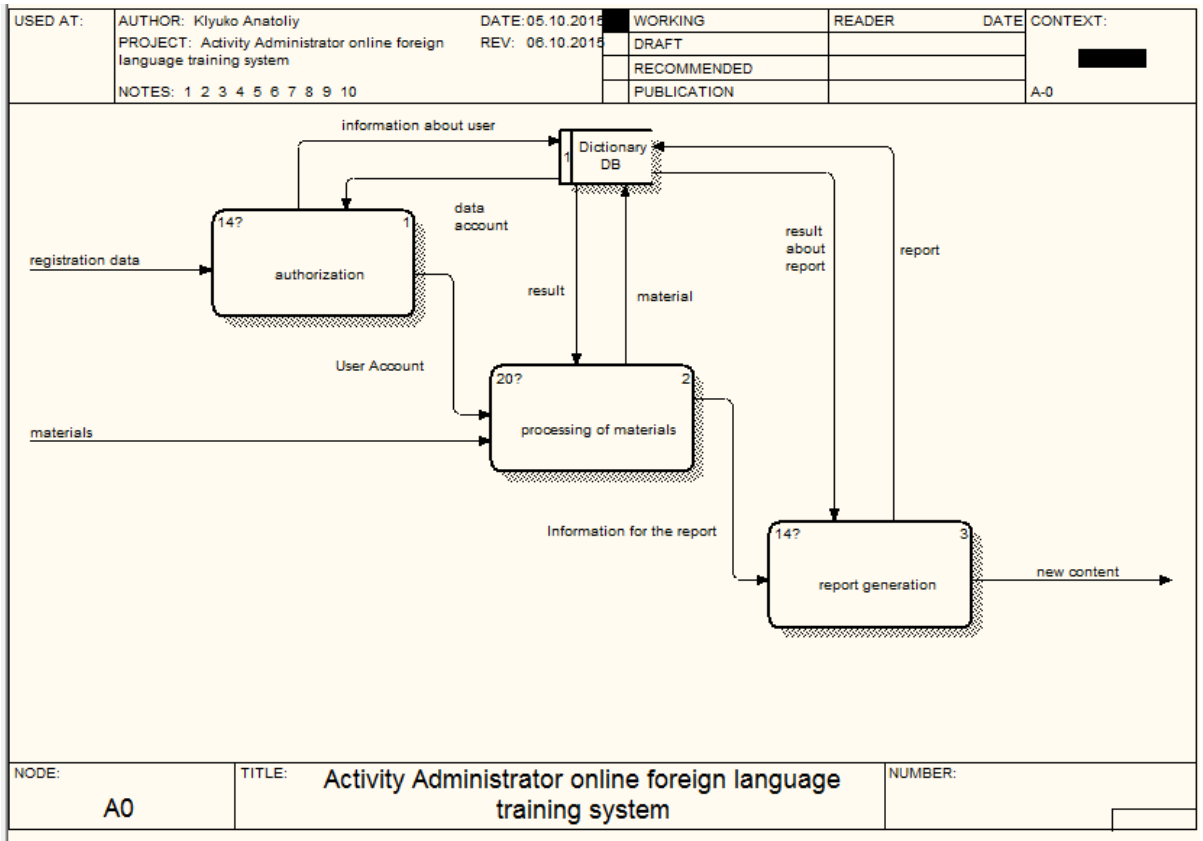


Рисунок 2.3 – Модель DFD

На рисунку 2.4 проілюстровано коротку діаграму станів системи [7].

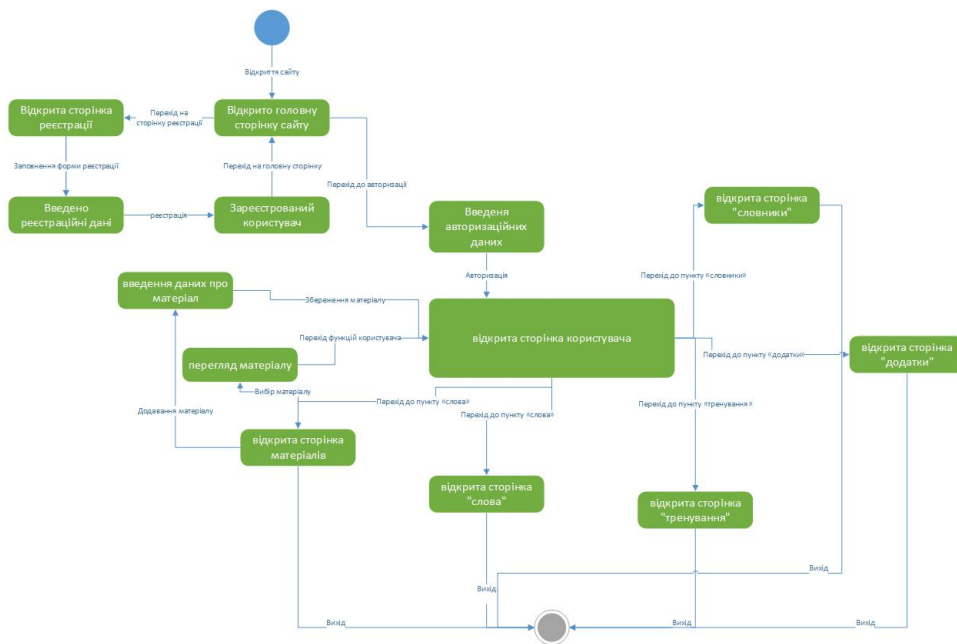


Рисунок 2.4 – Діаграма станів системи

Робота починається із завантаження головної сторінки сайту. Після чого потрібно авторизуватись або зареєструватись. Якщо користувача немає в системі, потрібно зареєструватись. Для цього потрібно заповнити реєстраційну форму. Якщо користувач раніше вже був зареєстрований, потрібно пройти процес авторизації. Для цього необхідно заповнити авторизаційну форму з даними, що були вказані при реєстрації. Після чого завантажиться сторінка користувача, на якій можна зробити наступні дії: відкрити сторінку з матеріалами, словниками, словами, додатками, тренуваннями. Процес тренування можна проілюструвати наступною діаграмою послідовностей, рисунок 2.5.

Користувач авторизується, заходить на сторінку тренувань, відповідає на запитання, які йому надходять. Програмна система ж надсилає користувачу запитання та перевіряє його відповіді та надсилає результат.

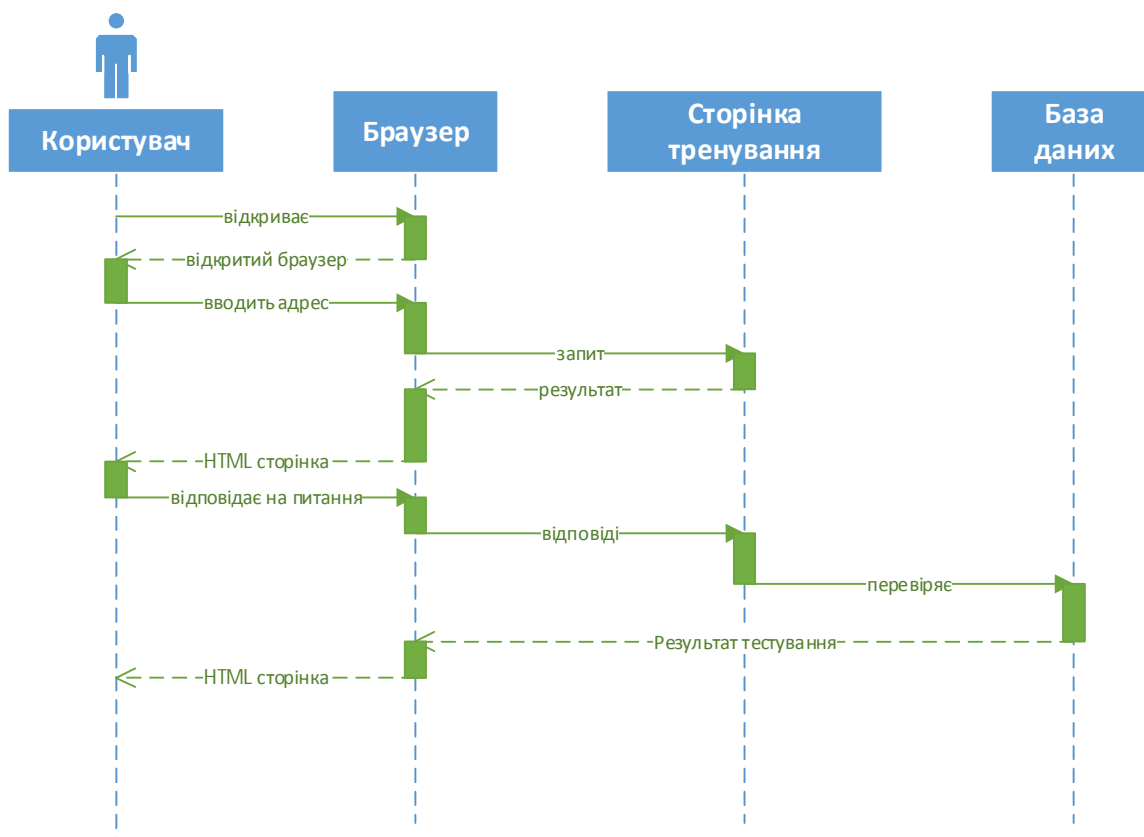


Рисунок 2.5 – Процес тренування

При об'єктно-орієнтованому підході часто використовують UML-діаграму класів [4]. Вона показує, які класи будуть створені і яка між ними взаємодія. Uml- діаграма проекту представлена на рисунку 2.6.

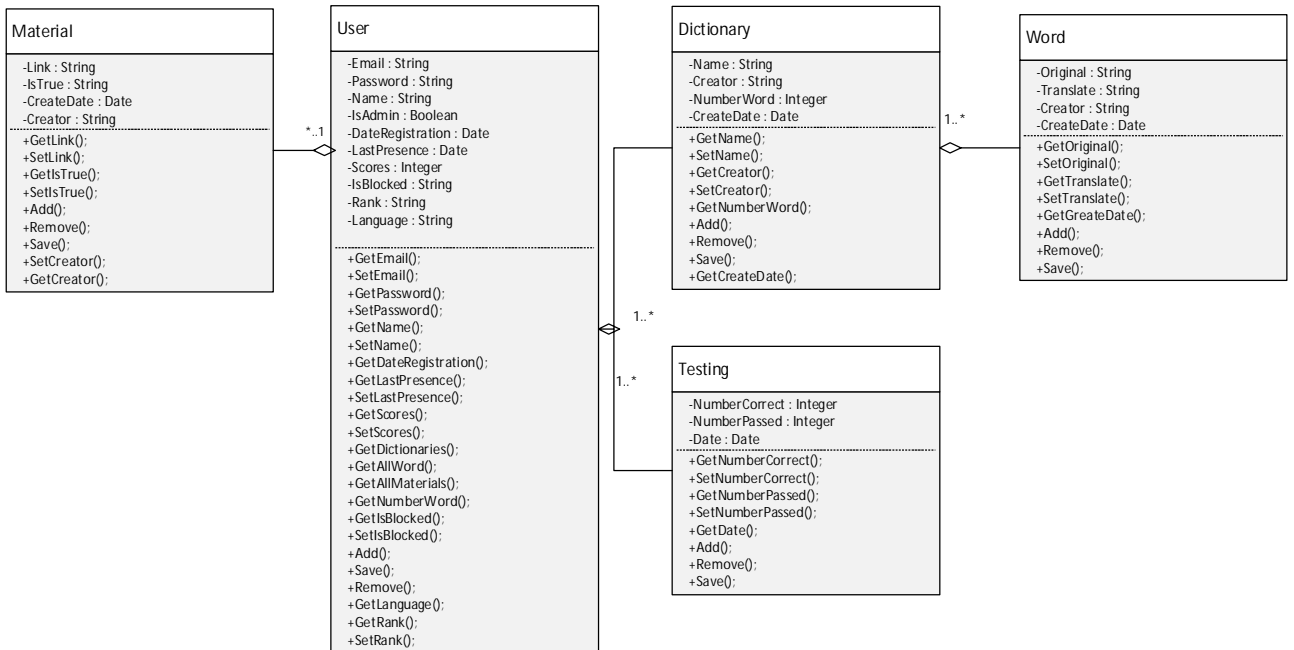


Рисунок 2.6 - UML-діаграма класів

Кожен користувач має довільну кількість словників, які в себе включають довільну кількість слів. Також користувач може додавати матеріали, скільки йому буде необхідно.

2.2. Проектування структури бази даних

При створенні web-додатку потрібно розробити базу даних, в якій буде зберігатися потрібна інформація. На сайті кожен може реєструватися, тому потрібно забезпечити зберігання даних кожного користувача (email, пароль, ім'я звання, мова). Також про користувача потрібно ще зберігати його дату реєстрації, останню активність на сайті, його кількість зароблених балів, чи не заблокований він адміністратором.

Оскільки кожен користувач має свої словники, тому потрібно зберігати інформацію про них. Словник складається із назви та набору слів. Слово в свою

чергу складається з оригінальної мови та перекладу. Таблицю ідентифікаторів наведено у таблиці 2.1.

Таблиця 2.1

Таблиця ідентифікаторів

Таблиця	Властивість	Тип	По замовчуванню
Users	id	int	-
	name	varchar	-
	email	varchar	-
	password	varchar	-
	rank	varchar	-
	isBlocked	tinyint	false
	lastPresence	datetime	-
	scores	int	0
	language_id	int	-
Materials	id	int	-
	user_id		null
	url	varchar	-
	date	datetime	-
Language	id	int	-
	name	varchar	-
Question	id	int	-
	ask	varchar	-
	correct	varchar	-
	one	varchar	-
	two	varchar	-
	three	varchar	-
Dictionary	id	int	
	name	varchar	

Word	id	int	-
	original	varchar	-
	translate	varchar	-
	createDate	datetime	-
Admin	id	int	-
	name	varchar	-
	password	varchar	-

Для швидшого виконання певних операцій над даними (наприклад пошук) використовуються індекси. Індокси слід створювати у відношеннях на полях, які найчастіше використовуватимуться та на полях, що є зовнішніми ключами – для пришвидшення вибірки з таблиці. Також не варто створювати занадто багато індексів [18].

Індокси для зовнішніх ключів та для полів пошуку зображено у таблицях 2.2 – 2.3 відповідно.

Таблиця 2.2

Індокси зовнішніх ключів

Відношення	Поле для індекса
Materials	user_id
Users	language_id
Word_dictionary	word_id
	dictionary_id
User_dictionary	user_id
	dictionary_id
Language_question	language_id
	question_id

Індекси полів для пошуку

Назва індекса	Відношення	Назва полів
users_email_password	Users	email, password
word_original_translate	Word	original, translate
dictionary_name	Dictionary	name
question_ask	Question	ask

Кожен користувач може запропонувати свій матеріал, тому для цього йому необхідно вказати лише посилання на нього. Потрібно зберігати також дату додання матеріалу та користувача, який його додав. На рисунку 2.7 проілюстрована ERD діаграма [14].

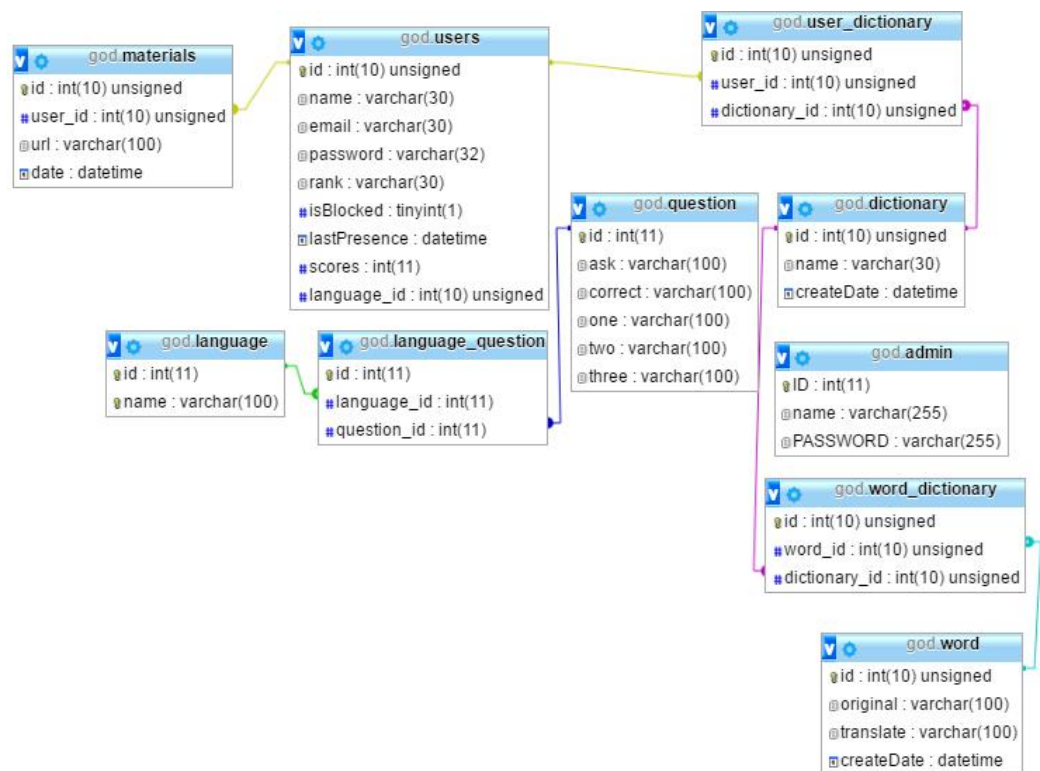


Рисунок 2.7 – ERD діаграма

Для проходження визначення рівня мови, необхідно відповісти на 21 запитання. Ці запитання зберігатимуться у відношенні Question, яке міститиме запитання, одну правильну та три не правильні відповіді.

Відношення `Word_dictionary`, `User_dictionary`, `Language_question` є допоміжними. Вони використовуються для зв'язку двох інших таблиць багато до багатьох.

Висновки до другого розділу:

1. Визначено архітектуру програмної системи.
2. Побудовано діаграму станів.
3. Побудовано послідовностей.
4. Побудовано UML-діаграму класів.
5. Було спроектовано структура базу даних, побудовано ERD діаграму.
6. Був написаний DDL код для створення таблиць бази даних.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Програмна реалізація проекту

3.1.1. Загальний опис програмного забезпечення.

Розроблювана система повинна представляти собою веб-сайт, на основі клієн-серверної архітектури. Уся бізнес логіка системи виконуватиметься на серверній частині, також тут зберігаються усі дані. Клієнт взаємодіє із системою через будь який браузер.

Оскільки у системі є користувачі різних ролей адміністратор, звичайний користувач і авторизований користувач, то на сайті повинна бути реалізована система авторизації. Кожен користувач має певні права доступу до системи.

Так звичайний користувач не може виконувати функції адміністратора системи, додавати словники, слова. Це може робити лише авторизований користувач [5].

Для звичайного користувача доступні тільки статичні сторінки сайту на яких він може переглядати загальну інформацію про сайт, залишити матеріал, написати звернення.

Для того, щоб створити словник, користувачу необхідно перейти на відповідну сторінку сайту, заповнити форму створення словника, яка включає назву. Для того, щоб додати слово в словник, необхідно зайти на відповідну сторінку, заповнити інформацію про слово(переклад та оригінал слова).

Для авторизованого користувача також є можливість, крім можливостей звичайного користувач, ще пройти тренування. Для цього необхідно вибрати словник, з якого воно буде проходити. Тренування «Оригінал – переклад» - зміст його полягає у тому, що користувачу видаються слова оригінальною мовою, йому необхідно вказати переклад слова, який він вказував при додаванні слова.

Тренування «Переклад – оригінал» - зміст його полягає у тому, що користувачу видається переклад слова, йому необхідно вказати оригінал слова, який він вказував при додаванні слова.

Тренування «Конструктор» - зміст його полягає у тому, що користувачу видаються літери оригінального слова та переклад, йому необхідно із цих літер скласти слово, яке він вказував при додаванні.

Тренування «Випадкові слова» - зміст його полягає у тому, що користувачу видаються випадкові слова із словників інших користувачів, необхідно вказати переклад слова, який вказали інші користувачі.

Адміністратор сайту має такі можливості як блокування користувачів. Для цього йому необхідно перейти на відповідну сторінку після авторизації та вибрати користувачів, яким необхідно заблокувати доступ до сайту. Також адміністратор може створювати тестування для визначення рівня мови. Для цього йому також потрібно перейти на відповідну сторінку та заповнити форму тестування, що складається із 21 питання, вказати до кожного питання три не правильні та одну правильну відповідь та вибрати мову, до якої буде прив'язане це тестування. Також може створити мову, для цього необхідно вказати назву мови.

Обґрунтування вибору мови програмування.

Для реалізації програмного продукту обрано мову програмування PHP. Це мова програмування для створення динамічних веб-сторінок. Вона є об'єктно-орієнтованою мовою. PHP має широкі можливості для роботи з різними СУБД, наприклад MySQL. Вона є з відкритим вихідним кодом та спеціально створювалась для створення веб-систем. Оскільки синтаксис мови схожий до синтаксису мов C++, Java тому це буде велике полегшення у роботі для людей, які вже знайомі з цими мовами [16]. Ця мова програмування на сьогоднішній день входить в стандартний перелік компаній, що надають послуги хостингу, а також входить в набір LAMP – стандартний набір для створення веб-сайтів.

Оскільки ця мова програмування орієнтована на веб-сервіси, то її використання дуже широке. Користувачі з різними пристроями можуть мати доступ до систем, написаних цією мовою: планшет, телефон, комп'ютер, чи інший пристрій [20].

3.1.2. Реалізація функцій системи.

Оскільки в системі присутні різні ролі користувачів, було створено наступну структуру: в загальному корені зберігатимуться файли, які можуть переглядати або звичайні користувачі, або авторизовані. У папці admin будуть зберігатись файли, до яких може мати доступ лише адміністратор. У папці lib зберігатимуться додаткові файли(набори різних потрібних бібліотек). Наприклад, бібліотека з валідністю даних. Також тут зберігається файл конфігурації з базою даних. В ньому записано ім'я користувача, пароль, назва бази даних. При потребі зміни бази даних, необхідно змінити дані в цьому файлі [6].

Нижче наведений код файлу:

```
function ConnectDB()  
{  
    $db = mysql_connect('localhost','god','123456');  
    mysql_select_db('god', $db);  
    mysql_set_charset('utf8');  
    mysql_query("SET NAMES 'utf8'");  
}
```

Тут також вказано кодування, яке буде використовуватись для роботи з базою даних.

Розмітка сторінок побудована за допомогою Bootstrap. Це фреймворк, який складається з інструментів, що допомагають оформити веб-сторінку. Він використовує сучасні досягнення в HTML та CSS, є одним з найпопулярніших фреймворків. Тому користувачам бажано користуватись сучасними версіями браузерів, так, як у старих версіях браузерів не все може відобразитись коректно.

Bootstrap підключений за допомогою наступного тегу:

```
<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
rel="stylesheet">
```

Механізм авторизації.

Авторизація була поділена на два типи. Для звичайних користувачів та для адміністратора. Для того, щоб авторизуватись звичайному користувачу, йому необхідно на будь якій сторінці сайту пройти авторизацію. Щодо адміністратора, то він може авторизуватись лише на сторінці, що знаходиться за адресою /admin. У свою чергу, звичайний користувач не знаючи цієї сторінки ніколи не зможе підібрати пароль адміністратора. Це своєрідний захист [3].

Для авторизації необхідно вказати свою електронну адресу та пароль. Нижче подано php скрипт, який відповідає за авторизацію.

```
<?php
    session_start();
    require_once '/lib/connect.php';
    ConnectDB();
    if(isset($_POST['submit']))
    {
        $email = strtolower(htmlspecialchars(strip_tags(trim($_POST['email']))
    ));
        $password =
    strtolower(htmlspecialchars(strip_tags(trim($_POST['password'])) ));
        $n_password = md5($password);
        $query = mysql_query("SELECT id, password FROM users WHERE email =
    '$email' ");
        $rLogin = mysql_fetch_array($query);

        if($rLogin['password'] == $n_password)
        {
            $_SESSION['UserId'] = $rLogin['id'];
            header("Location: index.php");
        }
        else
            echo " <p style=\"color:#F20C2F\">Не вірний пароль або логін!<p>
                <a href=\"login.php\">Назад</a>";
    }
?>
```

Вся інформація про користувача зберігається в сесії. Сесії – це механізм зберігання інформації про користувача на стороні сервера. Особливістю є час існування сесії. Вона може бути завершена за допомогою виклику спеціальної функції або, наприклад, закриттям браузер(не вкладки).

Приклад вмісту файлу, що відповідає за знищення інформації про користувача із сесії:

```

<?php
session_start();
unset($_SESSION['UserId']);
unset($_SESSION['UserName']);
unset($_SESSION['UserLanguage']);
unset($_SESSION['UserRank']);
session_destroy();
header("Location: index.php");
?>

```

Механізм додавання нових матеріалів.

Для того щоб додати новий матеріал, необхідно перейти на відповідну сторінку, вказати посилання на матеріал. Якщо матеріал додає авторизований користувач, зберігається його ідентифікатор. Нижче подано код, який відповідає за додавання нових матеріалів.

```

<?php
if(isset($_POST['submit']))
{
echo $_POST['url'] . "<br>";
$url = parse_url_if_valid($_POST['url']);
if (!$url) {
echo "Не вірне посилання, перевірте його будь ласка!";
} else {
if(isset($_SESSION['UserId']))
{
$UserId = $_SESSION['UserId'];
$insert = mysql_query("INSERT INTO materials(`user_id`,`url`)
VALUES('".$_SESSION['UserId']."', '".$_POST['url']."'");
if($insert){
echo "Ваше посилання: " . $url . " - було запропоноване! Дякуємо!";
}
else
{
echo "Не вірне посилання, перевірте його будь ласка ще раз!";
}
}
else
{
$insert = mysql_query("INSERT INTO materials(`url`)
VALUES('".$_POST['url']."'");
if($insert){
echo "Ваше посилання: " . $url . " - було запропоноване! Дякуємо!";
}
else
{
echo "Не вірне посилання, перевірте його будь ласка ще раз!";
}
}
}
}
?>

```

Як було сказано раніше, інформацію про користувача, який додає матеріал дізнаємось із сесії. Якщо користувач не авторизований, то це поле залишаємо порожнім.

Створення нового тестування адміністратором.

Для того щоб створити нове тестування, адміністратору необхідно перейти на відповідну сторінку та заповнити відповідну форму. Php скрипт, що відповідає за додавання нового тестування подано нижче:

```
<?php

if(isset($_POST['submit']))
{
    $language = htmlspecialchars(strip_tags(trim($_POST['language'])));
    for($i = 1; $i < 22; $i++){
        $question[] = htmlspecialchars(strip_tags(trim($_POST['q' . $i ])));
        $right[] = htmlspecialchars(strip_tags(trim($_POST['r' . $i ])));
        $a[] = htmlspecialchars(strip_tags(trim($_POST['a' . $i ])));
        $b[] = htmlspecialchars(strip_tags(trim($_POST['b' . $i ])));
        $c[] = htmlspecialchars(strip_tags(trim($_POST['c' . $i ])));
    }
    $number = mysql_query("SELECT name FROM language WHERE name = '$language' ");
    $num_rows = mysql_num_rows($number);
    if($num_rows <= 0)
    {
        $insert = mysql_query("INSERT INTO language(`name`) VALUES('$language.')");
        if($insert)
        {
            $lastId = mysql_query("SELECT id FROM language ORDER BY id DESC LIMIT 1");
            $resLastId = mysql_fetch_array($lastId);
            $lastIdLanguage = $resLastId['id'];
            for($i = 0; $i < 21; $i++){
                $insertQuestion = mysql_query("INSERT INTO
                question(`ask`,`correct`,`one`,`two`,`three`)
                VALUES('".$question[$i]."', '".$right[$i]."', '".$a[$i]."', '".$b[$i]."', '".$c[$i].
                "')");
                if($insertQuestion){
                    $lastId = mysql_query("SELECT id FROM question ORDER BY id DESC LIMIT 1");
                    $resLastId = mysql_fetch_array($lastId);
                    $lastIdQuestions = $resLastId['id'];
                    $insertLanQuest = mysql_query("INSERT INTO
                    language_question(`language_id`,`question_id`)
                    VALUES('$lastIdLanguage.', '$lastIdQuestions.')");
                }
            }
        }
        echo " <p style=\"color:#5DDD2C\">Мову: $language додано. Тестування створено!
        <p>";
    }
    else
    {
        echo " <p style=\"color:#F20C2F\">Така мова вже зареєстрована 1!<p>";
    }
    else
    {
        echo " <p style=\"color:#F20C2F\">Така мова вже зареєстрована!<p>";
    }
}
?>
```

Якщо ж необхідно внести зміни до вже створеного тестування, то це робиться за допомогою наступного коду:

```

<?php
if(isset($_POST['save_testing']))
{
$language = htmlspecialchars(strip_tags(trim($_POST['language'])));
echo $language;
$counter = 0;
for($i = 1; $i < 22; $i++){
$question[] = htmlspecialchars(strip_tags(trim($_POST['q' . $i ])));
$right[] = htmlspecialchars(strip_tags(trim($_POST['r' . $i ])));
$a[] = htmlspecialchars(strip_tags(trim($_POST['a' . $i ])));
$b[] = htmlspecialchars(strip_tags(trim($_POST['b' . $i ])));
$c[] = htmlspecialchars(strip_tags(trim($_POST['c' . $i ])));
$qid[] = htmlspecialchars(strip_tags(trim($_POST['qid' . $i ])));
$res = mysql_query ("UPDATE question SET
question.ask = '$question[$counter]',
question.correct = '$right[$counter]',
question.one = '$a[$counter]',
question.two = '$b[$counter]',
question.three = '$c[$counter]'

WHERE
question.id = '$qid[$counter]'
" );
if($res){
echo " <p>Питання $i збережено!<p>";
}
else
{
echo "Питання $i НЕ ЗБЕРЕЖЕНО ";
die('Invalid query: ' . mysql_error());
}
$counter++;
}
echo " <p><a href=\"index.php\">На головну</a></p>";
}
else
{
header("Location: testing.php");
}
?>

```

Кожен користувач може створювати собі довільну кількість словників, вказавши лише його назву. Скрипт, за допомогою якого створюється словник подано нижче:

```

<?php
if(isset($_POST['newDictionary']))
{
$name = strtolower(htmlspecialchars(strip_tags(trim($_POST['name']))));
if(strlen($name) > 1) {
$insert = mysql_query("INSERT INTO dictionary(`name`)
VALUES('".$name."')");
$lastId = mysql_insert_id();
$insertUD = mysql_query("INSERT INTO user_dictionary(`user_id`,`dictionary_id`)
VALUES('".$_SESSION['UserId']."', '".$lastId."')");
echo "збережено!";
}
else
{
echo " <p style=\"color:#F20C2F\">Коротка назва!<p>";
}
}
?>

```

В додатковій таблиці також зберігається користувач, якому належить цей словник.

3.2. Програмна реалізація бази даних

Для проекту я обрав базу даних MySQL. MySQL — вільна система керування реляційними базами даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання [9].

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Для того щоб додати новий запис в таблицю необхідно використати наступний синтаксис [12]:

```
INSERT INTO назва_таблиці(властивість`) VALUES('значення')
```

Вибірка здійснюється за допомогою команди:

```
SELECT властивість1, властивість2 FROM таблиця;
```

Для того, щоб створити таблицю, потрібно знати з яких властивостей вона складається, визначити тип кожної властивості. Для того, щоб створити таблицю, необхідно використати команду create. Створити таблицю можна

двома способами. Це у візуальному редакторі або в спеціальній консолі. Недоліком першого способу є незручність та час створення таблиці. В консолі це робиться набагато краще. Перевагою першого способу є те, що не потрібно знати команди та правила створення таблиць, адже для цього потрібно заповнити лише відповідну форму створення таблиці. Недоліком другого способу є те, що потрібно знати синтаксис команд [19]. На рисунку 3.1 проілюстровано візуальний спосіб створення таблиць, а на рисунку 3.2 проілюстровано спосіб створення таблиць в спеціальній консолі.

Рисунок 3.1 – Візуальний редактор

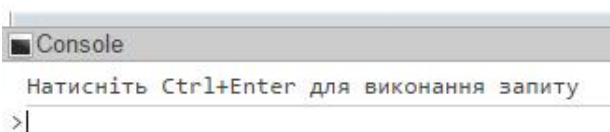


Рисунок 3.2 – Спеціальна консоль

Особисто я надаю перевагу другому способу, адже це набагато швидше, що я є дуже важливим. Також це корисно, адже знати синтаксис мови SQL добре кожному. Також, при потребі можна використати вже раніше заготовлені частини коду.

Для додавання нового користувача в систему використовується запит:

```
$insert = mysql_query("INSERT INTO users(`name`,`email`,`password`,`rank`,`language_id`) VALUES('".$name."', '".$email."', '".$n_password."', '".$rank."', '".$language_id.'");");
```

Запит, що використовується для знаходження ідентифікатора мови за відомою назвою подано нижче:

```
$number = mysql_query("SELECT id FROM language WHERE name = '$language' ");
```

Для додавання нового матеріалу авторизованим користувачем використовується наступний запит:

```
$insert = mysql_query("INSERT INTO materials(`user_id`,`url`) VALUES('".$UserId."', '".$url.'");");
```

Для додавання матеріалу не авторизованим користувачем використовується такий запит:

```
$insert = mysql_query("INSERT INTO materials(`url`) VALUES('".$url.'");");
```

Створення нового словника відбувається за допомогою наступного запиту:

```
$insert = mysql_query("INSERT INTO dictionary(`name`) VALUES('".$name.'");");
$insertUD = mysql_query("INSERT INTO user_dictionary(`user_id`,`dictionary_id`) VALUES('".$_SESSION['UserId']."', '".$lastId.'");");
```

Додавання нових слів у словник відбувається за допомогою запитів, що наведені нижче:

```
$query = mysql_query("SELECT dictionary.id as dic_nameFROM user_dictionary
JOIN dictionary ON user_dictionary.dictionary_id = dictionary.id
JOIN users On user_dictionary.user_id = users.id
where dictionary.name = '$dictionary_name' and users.id = '$user_id'; ");
$insert = mysql_query("INSERT INTO word(`original`,`translate`)
VALUES('".$original."', '".$translate.'");");
$insertWD = mysql_query("INSERT INTO word_dictionary(`word_id`,`dictionary_id`)
VALUES('".$lastId."', '".$dictionary_id.'");");
```

Вихідний код всіх таблиць подано нижче:

```
-- phpMyAdmin SQL Dump
-- version 4.3.11
-- http://www.phpmyadmin.net
--
-- Хост: 127.0.0.1
-- Версія сервера: 5.6.24
-- Версія PHP: 5.6.8

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```

SET time_zone = "+00:00";

CREATE TABLE IF NOT EXISTS `admin` (
  `ID` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `PASSWORD` varchar(255) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `language` (
  `id` int(11) NOT NULL,
  `name` varchar(100) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `language_question` (
  `id` int(11) NOT NULL,
  `language_id` int(11) DEFAULT NULL,
  `question_id` int(11) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=43 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `materials` (
  `id` int(10) unsigned NOT NULL,
  `user_id` int(10) unsigned DEFAULT NULL,
  `url` varchar(100) NOT NULL,
  `date` datetime DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `question` (
  `id` int(11) NOT NULL,
  `ask` varchar(100) DEFAULT NULL,
  `correct` varchar(100) DEFAULT NULL,
  `one` varchar(100) DEFAULT NULL,
  `two` varchar(100) DEFAULT NULL,
  `three` varchar(100) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=43 DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(10) unsigned NOT NULL,
  `name` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL,
  `password` varchar(32) NOT NULL,
  `rank` varchar(30) NOT NULL,
  `isBlocked` tinyint(1) DEFAULT NULL,
  `lastPresence` datetime DEFAULT CURRENT_TIMESTAMP,
  `scores` int(11) DEFAULT NULL,
  `language_id` int(10) unsigned NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

ALTER TABLE `admin`
  ADD PRIMARY KEY (`ID`);

-- Індекс таблиці `language`
ALTER TABLE `language`
  ADD PRIMARY KEY (`id`), ADD UNIQUE KEY `name` (`name`);

-- Індекс таблиці `language_question`
ALTER TABLE `language_question`
  ADD PRIMARY KEY (`id`), ADD KEY `language_id` (`language_id`), ADD KEY `question_id` (`question_id`);

-- Індекс таблиці `materials`
ALTER TABLE `materials`
  ADD PRIMARY KEY (`id`), ADD KEY `user_id` (`user_id`);

```

```

-- Індокси таблиці `question`
ALTER TABLE `question`
  ADD PRIMARY KEY (`id`);

-- Індокси таблиці `users`
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

-- AUTO_INCREMENT для збережених таблиць
-- AUTO_INCREMENT для таблиці `admin`
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
-- AUTO_INCREMENT для таблиці `language`
ALTER TABLE `language`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=3;
-- AUTO_INCREMENT для таблиці `language_question`
ALTER TABLE `language_question`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=43;
-- AUTO_INCREMENT для таблиці `materials`
ALTER TABLE `materials`
  MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=10;
-- AUTO_INCREMENT для таблиці `question`
ALTER TABLE `question`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=43;
-- AUTO_INCREMENT для таблиці `users`
ALTER TABLE `users`
  MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=5;
-- Обмеження зовнішнього ключа збережених таблиць
-- Обмеження зовнішнього ключа таблиці `language_question`
ALTER TABLE `language_question`
ADD CONSTRAINT `language_question_ibfk_1` FOREIGN KEY (`language_id`) REFERENCES
`language` (`id`),
ADD CONSTRAINT `language_question_ibfk_2` FOREIGN KEY (`question_id`) REFERENCES
`question` (`id`);
-- Обмеження зовнішнього ключа таблиці `materials`
ALTER TABLE `materials`
ADD CONSTRAINT `materials_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users`
(`id`);

```

Для розгортання нового проекту його достатньо лише імпортувати через відповідний інтерфейс. Весь код програмної системи подано у додатку А.

Висновки до третього розділу:

1. Розроблена структура програмного продукту.
2. Реалізовано програмний продукт.
3. Спроектвано та реалізовано базу даних.
4. Вивчив та застосував синтаксис мови SQL.

РОЗДІЛ 4

ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1 Тестування

Для перевірки програмного продукту на відповідність сформованих в специфікації вимогам та для виявлення і усунення помилок в програмному продукті було проведено тестування. Даний етап розробки є дуже важливим, адже визначає чи прийнятна система для подальшої експлуатації. На сьогоднішній день існує багато видів тестувань, кожен з яких дозволяє перевірити конкретний аспект програмного продукту.

Враховавши специфіку розроблюваного продукту, було складено та проведено наступні види тестування:

- функціональне тестування;
- тестування безпеки.

4.1.1. Функціональне тестування.

Для розроблюваної системи було проведено функціональне тестування. Це дозволило перевірити чи відповідають вимоги, задокументовані в специфікації вимог, реальним можливостям системи. Функціональне тести охоплюють всі розроблені функції та розроблені з акцентом на тестування найбільш ймовірних типів помилок в програмній системі.

Було проведено 15 тестів. Всі мали успішний результат. Отже функціональне тестування показало, що всі функціональні вимоги були реалізовані добре і відповідають поставленим вимогам. Тобто відповідають на 100%. Текст функціонального тестування наведено в додатку Б.

4.1.2. Тестування безпеки

Тестування безпеки в розроблюваному програмному продукті є дуже важливим. Оскільки в системі присутні різні користувачі (звичайний та

адміністратор) тому необхідно забезпечити максимальний захист кожного з учасників. Для забезпечення захисту було розроблено різні сторінки авторизації для користувачів. Тобто звичайний користувач авторизується на будь якій сторінці сайту. Адміністратору необхідно авторизуватись за посиланням /admin. Звісно, про цю сторінку авторизації користувачі не знають, вони її можуть знайти лише випадково ввівши в адресному рядку браузера. Але не знаючи пароль та логін адміністратора вони не зможуть увійти в систему. Всі паролі користувачів зберігаються в базі даних у зашифрованому вигляді за допомогою функції md5 [13]. Приведені паролі у форму він відображається символом «*», що унеможливорює підглянути пароль іншій особі.

Дуже важливим є процес авторизації та аутентифікації. Після проходження авторизації всі дані про користувача зберігаються в сесії. Сесія – це механізм зберігання інформації про користувача на стороні сервера. Особливістю є час існування сесії. Вона може бути завершена за допомогою виклику спеціальної функції або, наприклад, закриттям браузер (не вкладки).

Приклад вмісту файлу, що відповідає за знищення інформації про користувача із сесії:

```
<?php
session_start();
unset($_SESSION['UserId']);
unset($_SESSION['UserName']);
unset($_SESSION['UserLanguage']);
unset($_SESSION['UserRank']);
session_destroy();
header("Location: index.php");
?>
```

У порівнянні з куками, інформація про користувача зберігається до тих пір, поки триває життєвий цикл самої куки. Тобто навіть після закриття браузера. Це є перевагою у порівнянні із сесією, адже користувачу не потрібно кожного разу вводити авторизаційні дані. Проте це позбавляє безпеки даних користувача.

Щоб заборонити зловмиснику використати різні вразливості, наприклад SQL injections, було використано функції, що інвертують дані ін'єкції як

звичайний текст. Тобто, зловмисник не зможе видалити таблиці чи зробити інших неправомірних дій з даними.

Приклад захисту подано нижче:

```
$email = strtolower(htmlspecialchars(strip_tags(trim($_POST['email']))));
```

4.2 Розгортання програмного продукту

Для того щоб розгорнути програмний продукт необхідно на комп'ютері встановити XAMPP. Це такий набір інструментів, що включає в себе Apache, MySQL, інтерпретатор PHP. XAMPP є кросплатформена збірка. Тому можна інстальювати на різні операційні системи.

Завантажити XAMPP можна за посиланням <https://www.apachefriends.org/ru/index.html>. Після встановлення та запуску XAMPP відкриється вікно налаштувань, яке проілюстроване на рисунку 4.1.

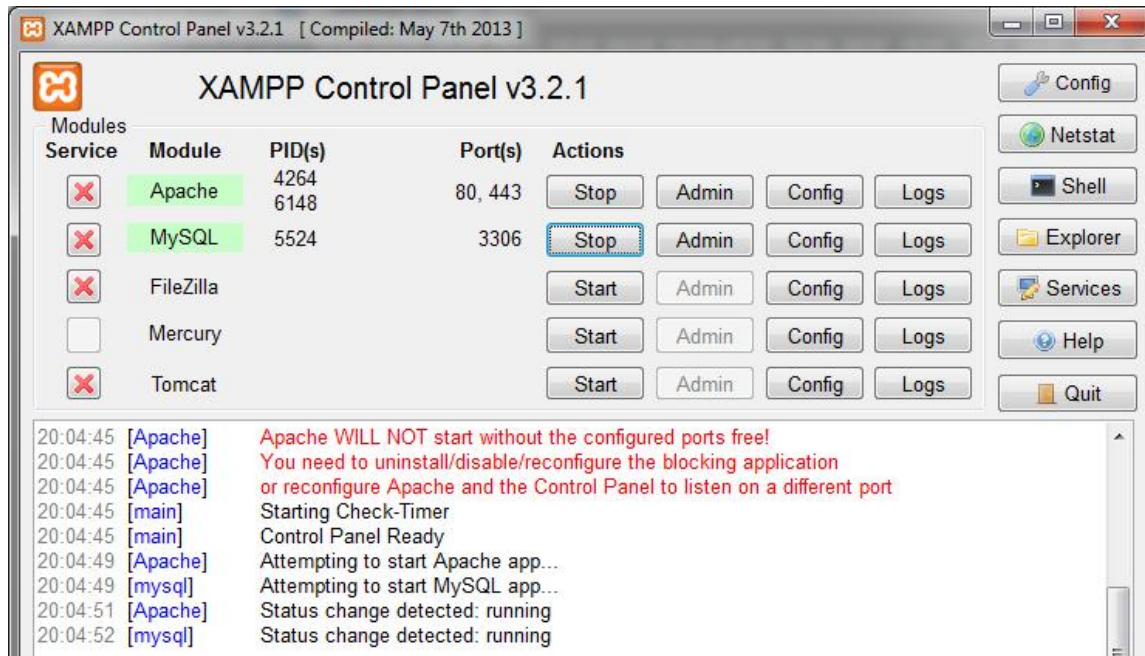


Рисунок 4.1 – Вікно налаштувань XAMPP

Необхідно запуснути сервер Apache та MySQL. Особливістю є те, що Apache використовує 80 порт. Скайп також використовує цей порт. Тому для

того, щоб запустити Apache потрібно закрити скайп, якщо він відкритий. Після запуску Apache знову можна використовувати скайп.

Для того, щоб створити проект, необхідно за адресою C:\xampp\htdocs(я робив на windows 7 x64) створити папку, яка буде використовуватись як домен майбутнього проекту. Для роботи з проектом всі файли надалі необхідно створювати в цій папці. На рисунку 4.2 проілюстровано структуру проекту.

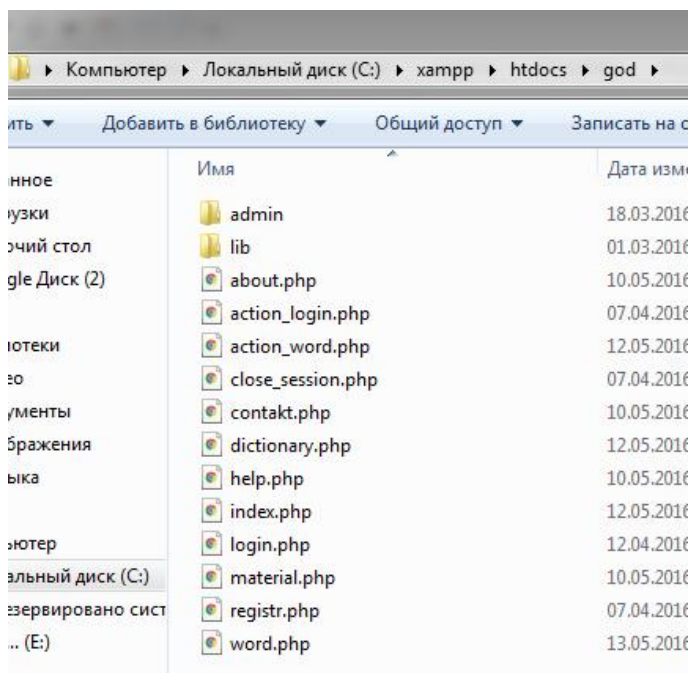


Рисунок 4.2 – Структура проекту

Для доступу до головної сторінки проекту необхідно перейти за посиланням: <http://localhost/god>. Для того, щоб перейти до бази даних, необхідно перейти за посиланням <http://localhost/phpmyadmin>.

4.3 Інструкція користувача

Звичайний користувач може переглядати статичні сторінки сайту: «Контакти», «Допомога», «Про сайт», «Матеріали».

На рисунку 4.3 проілюстровано сторінку «Контакти». На ній відображається інформація для зв'язку з розробником. Для того, щоб

переглянути цю сторінку необхідно перейти за посиланням:
<http://god/contact.php>.

Головна Про сайт Матеріали Допомога Контакти

Меню

Про сайт Матеріали

Допомога Контакти

Контакти

Email: klyukoanatoliy@gmail.com

Моб.тел.: +38 099 3812 832

skype: tolyan---tolya

Рисунок 4.3 – «Контакти»

Якщо у користувача виникають запитання, побажання, рекомендації, то він про це може написати, перейшовши на сторінку допомоги, що знаходиться за адресою: <http://god/help.php>. На рисунку 4.4 проілюстровано форму допомоги.

Ви маєте до нас запитання чи зауваження? Будь ласка, напишіть нам!

Ваша електронна адреса:

На цей email прийде відповідь

Ми не будемо використовувати ваше електронну адресу в інших цілях.

Текст звернення

Submit

Рисунок 4.4 – Форма допомоги

Для того, щоб переглянути раніше додані матеріали або запропонувати свій, необхідно перейти за посиланням: <http://god/material.php> або у розділ «Матеріали». Додавання матеріалу відбувається заповненням форми та натиском на кнопку «Додати». На рисунку 4.5 проілюстровано сторінку «Матеріали».

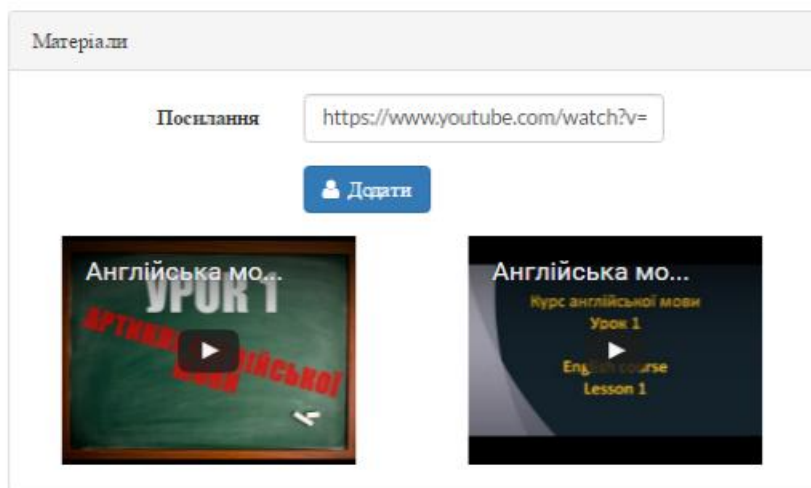


Рисунок 4.5 – Сторінка «Матеріали»

Після проходження процесу реєстрації та авторизації користувач може створювати свої словники. Для цього необхідно перейти у відповідний пункт меню «Словники» або перейти за посиланням: <http://god/dictionary.php>. Щоб додати новий словник, необхідно написати його назву та натиснути відповідну кнопку «Створити». Якщо користувач бажає видалити словник, то йому необхідно вибрати його із випадаючого списку та натиснути відповідну кнопку «Видалити». Слід пам'ятати, що разом із видаленням словника видаляється і його вміст. На рисунку 4.6 проілюстровано сторінку «Словники».

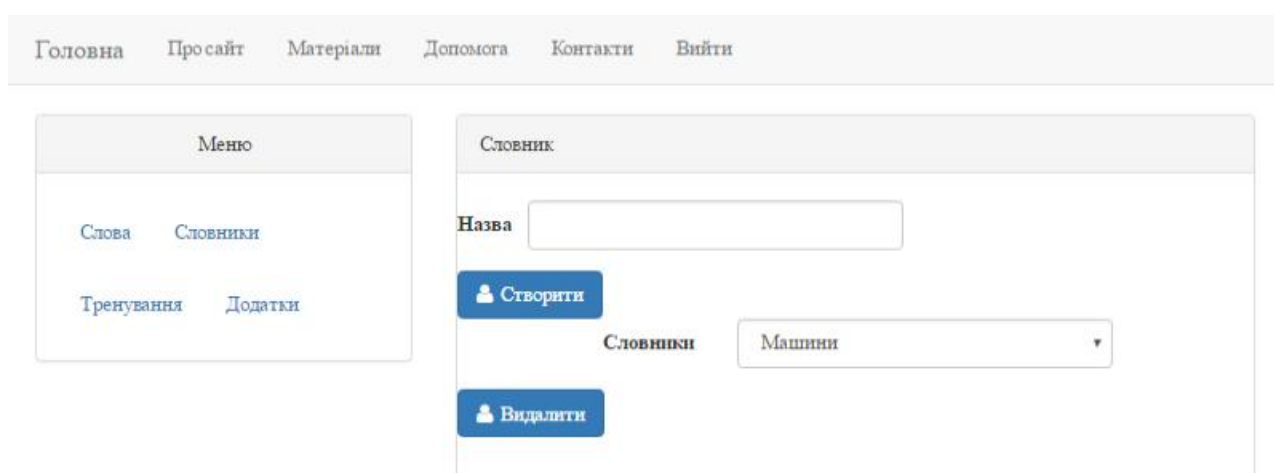
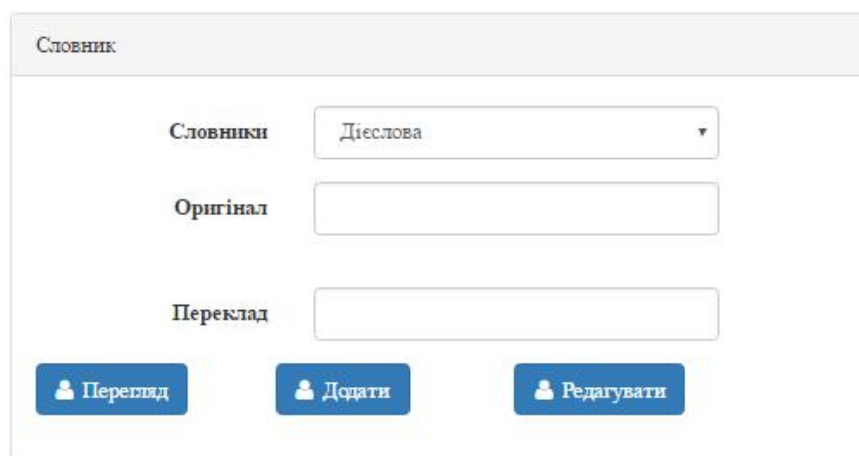


Рисунок 4.6 – Сторінка «Словники»

Для того щоб мати можливість переглядати, додавати, редагувати слова необхідно перейти у пункт меню «Слова» або за посиланням: <http://localhost/god/word.php>. Щоб переглянути свої слова необхідно із випадаючого списку вибрати словник та натиснути відповідну кнопку «Перегляд». Слова додаються у вибраний словник з випадаючого списку. Щоб їх додати у полі «Оригінал» необхідно вказати слово оригінальною мовою, а у полі «Переклад», необхідно написати переклад слова. При натиску на кнопку «Редагувати» відкриється сторінка, на якій можна змінити значення слів словника, який був вибраний у випадаючому списку [17]. На рисунку 4.7 проілюстровано сторінку «Слова».



The image shows a web interface for a dictionary. At the top, there is a header with the word 'Словник'. Below it, there is a section with three main elements: a dropdown menu labeled 'Словники' with 'Дієслова' selected, an input field labeled 'Оригінал', and another input field labeled 'Переклад'. At the bottom of this section, there are three blue buttons with white text and icons: 'Перегляд' (with a magnifying glass icon), 'Додати' (with a plus icon), and 'Редагувати' (with a pencil icon).

Рисунок 4.7 – Сторінка «Слова»

Тренування знаходяться у пункті меню «Тренування». Необхідно вибрати тип тренування та словник. На рисунку 4.7 проілюстровано тренування «Оригінал – переклад». Із обраного словника у випадковому порядку виводяться слова оригінальною мовою, користувачу необхідно згадати і вказати переклад цього слова.

На рисунку 4.9 проілюстровано тренування «Конструктор». Його мета полягає у тому, що користувачу необхідно скласти слово із літер, які у випадковому порядку вивелись.

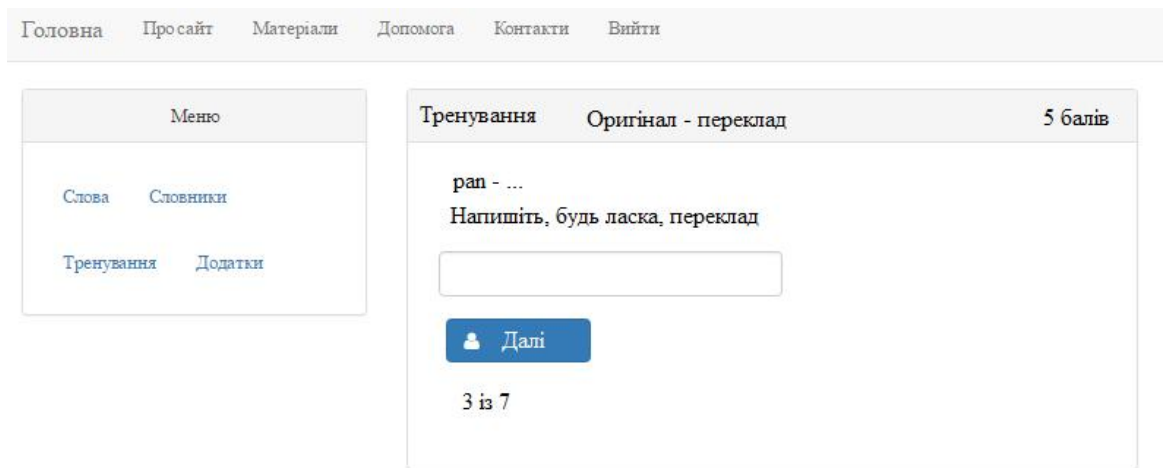


Рисунок 4.8 – Тренування «Оригінал – переклад»

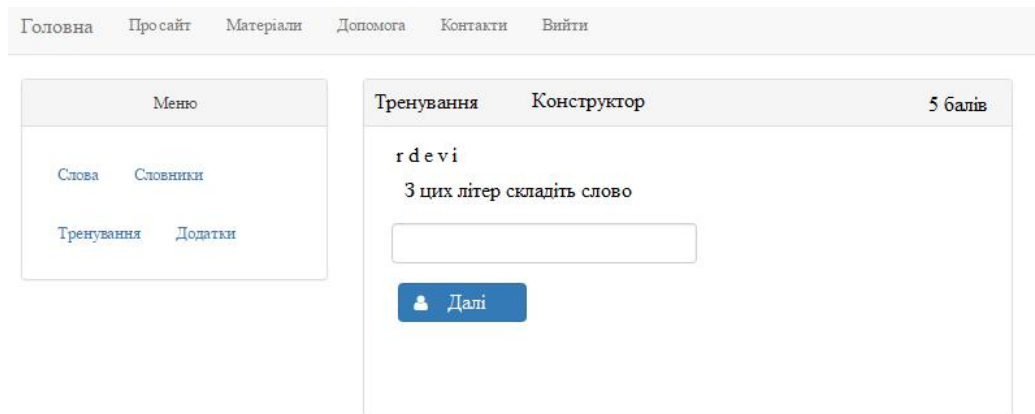


Рисунок 4.9 – Тренування «Конструктор»

На рисунках 4.10 – 4.11 проілюстровано порядок дій для блокування користувача адміністратору. Для початку необхідно перейти у відповідний пункт меню та вибрати користувача, якого необхідно заблокувати (див. рис.4.10).

Id	Name
2	kolvan
3	sergiy
4	ivan
5	vova
6	vasya

Рисунок 4.10 – Вибір користувача

Після цього необхідно відмітити пункт «Обмеження» (див. рис. 4.11) та натиснути кнопку «Змінити». Також за допомогою цього пункту також можна зняти блокування користувача.

status	id	Name	Обмеження
<input checked="" type="checkbox"/>	2	kolyan	<input type="checkbox"/>

Рисунок 4.11 – Блокування користувача

Адміністратор перейшовши у пункт «Тестування» може створити нове тестування. Для цього необхідно йому вказати мову, до якої належатиме тестування, та заповнити форму, що складається із 21 питання. На рисунку 4.12 наведено форму лише для одного запитання.

Мова

Питання 1:

Відповіді:

Правильна

A

B

C

Рисунок 4.12 – Форма створення тестування

Висновки до четвертого розділу:

1. Було проведене функціональне тестування.
2. Описано детально розгортання системи.
3. Описано інструкцію користувача: звичайного, авторизованого, адміністратора.

ВИСНОВКИ

Результатом цієї дипломної роботи було створено новий програмний продукт. Для початку було проведено аналіз існуючих подібних ресурсів. На основі яких була складена специфікація вимог до програмного продукту. Я розглянув всі переваги і недоліки цих ресурсів і врахував сильні і слабкі сторони кожного з них. Після цього я сформував всі функціональні та не функціональні вимоги до майбутнього програмного продукту.

Наступним етапом я визначив архітектуру програмної системи, побудував діаграму станів, послідовностей, UML-діаграму класів. Я використав базу даних MySQL. Спроектував структуру бази даних програмного продукту, та написав DDL код для створення таблиць.

Після цього я реалізував всі функціональні та не функціональні вимоги за допомогою мови програмування PHP. Провів функціональне тестування, а також описав інструкцію користувача: звичайного, авторизованого, адміністратора.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP. [Електронний ресурс] – Режим доступу: вільний. — <http://php.net/>. - (дата звернення 21.05.2016).
2. Winesett Jeffrey Agile Web Application Development PHP5. ISBN 978-1-847199-58-4 изд. Packt Publishing, 2010.
3. Вирт Н. Алгоритмы и структуры данных.– М.: Мир, 1989.
4. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide. — 2-е изд. — М., СПб.: ДМК Пресс, Питер, 2004. — 432 с. — ISBN 5-94074-260-2.
5. Гутманс Э., Баккен С, Ретанс Д. PHP 5. Профессиональное программирование./ Пер. с англ. СПб: Символ- Плюс, 2006. 704 с., ил.
6. Дейт К. Введение в системы баз данных / 6-издание. - Киев: Диалектика, 1998. - 784 с.
7. Джозеф Шмуллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство = Sams Teach Yourself UML in 24 Hours, Complete Starter Kit. — М.: Вильямс, 2005. — 416 с. — ISBN 0-672-32640-X.
8. Документация по MySQL [Електронний ресурс] – режим доступу: вільний. – <http://www.mysql.ru/docs/> - (дата звернення 29.05.2016).
9. Документация по MySQL [Електронний ресурс] – режим доступу: вільний. – <http://www.mysql.ru/docs/> - (дата звернення 20.05.2016).
10. Зандстра М. PHP: Objects, Patterns, and Practice / Мэт Зандстра. – Санкт-Петербург: Вильямс, 2011. – 560 с.
11. Зандстра М., PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition — М.: «Вильямс», 2010. — С. 560. — ISBN 978-5-8459-1689-1.
12. Кириллов В.В. Структуризованный язык запросов (SQL). - СПб.: ИТМО, 1994. - 80 с.

13. Кнут Д. Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms — 3-е изд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4
14. Концептуальное проектирование реляционных баз данных с использованием языка UML [Электронный ресурс] – режим доступа: вільний. – <http://citforum.ru/database/articles/umlbases.shtml/> - (дата звернення 14.04.2016).
15. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2.
16. Ларри Ульман. Ульман Л. Основы программирования на PHP: /Ларри Ульман. Пер. с англ. -М.: ДМК Пресс, 2001. -288 с.: ил. (Самоучитель).
17. Нильсен Я., Перниче К. Веб-дизайн: анализ удобства использования веб-сайтов по движению глаз = Eyetracking Web Usability — М.: «Вильямс», 2010. — С. 480. — ISBN 978-5-8459-1652-5.
18. Оптимизация запросов с помощью индексов [Электронный ресурс] – режим доступа: вільний. – <http://xpoint.ru/know-how/MySQL/Optimizatsiya/Indeksyi/> - (дата звернення 24.04.2016).
19. Справочное руководство по MySQL [Электронный ресурс] – режим доступа: вільний. – http://www.php.su/mysql/manual/?page=MySQL_indexes/ - (дата звернення 22.04.2016).
20. Стивен Хольцнер . PHP в примерах. / Стивен Хольцнер . М.: 000 «Бином-Пресс», 2007 г. Пер. с англ. 352 с

ДОДАТОК А

Лістинг коду

Файл connect.php

```
<?php

function ConnectDB()
{
$db = mysql_connect('localhost','god','123456');
mysql_select_db('god', $db);
mysql_set_charset('utf8');
//mysql_query("SET NAMES 'cp1251'");
mysql_query("SET NAMES 'utf8'");
}

function ValidEmail($str)
{
$str = strtolower($str);
if(preg_match("/^([a-z0-9_\.-]+)@([a-z0-9_\.-]+)\.([a-z\.\.]{2,6})$/", $str))
{
return true;
}
else
{
return false;
}
}
```

Файл validation.php

```
<?php
function checkPassword($password){
$re = "[+,-,/,*]+[0-9]+/u";
$str = $password;
return !preg_replace($re, '', $str);
}
function parse_url_if_valid($url)
{
$arUrl = parse_url($url);
$ret = null;

if (!array_key_exists("scheme", $arUrl)
|| !in_array($arUrl["scheme"], array("http", "https")))
$arUrl["scheme"] = "http";

if (array_key_exists("host", $arUrl) &&
!empty($arUrl["host"]))
$ret = sprintf("%s://%s%s", $arUrl["scheme"],
$arUrl["host"], $arUrl["path"]);
else if (preg_match("/^\w+\.[\w\.\.]+(\./.*)?$/", $arUrl["path"]))
$ret = sprintf("%s://%s", $arUrl["scheme"], $arUrl["path"]);
```

```

    if ($ret && empty($ret["query"]))
    $ret .= sprintf("%s", $arUrl["query"]);
    return $ret;
}

```

Файл action_login.php

```

<?php

session_start();
require_once '/lib/connect.php';
ConnectDB();

if(isset($_POST['submit']))
{
$email = strtolower(htmlspecialchars(strip_tags(
trim($_POST['email'] ))));
$password =
strtolower(htmlspecialchars(strip_tags(trim($_POST['password']
))));
$n_password = md5($password);
$query = mysql_query("SELECT id, password FROM users WHERE email =
'$email' ");
$rLogin = mysql_fetch_array($query);

if($rLogin['password'] == $n_password)
{

$_SESSION['UserId'] = $rLogin['id'];
header("Location: index.php");

}
else
echo " <p style=\"color:#F20C2F\">Не вірний пароль або логін!<p>
<a href=\"login.php\">Назад</a>";

}

?>

```

Файл action_word.php

```

<?php
session_start();
header('Content-Type: text/html; charset=utf-8');
require_once '/lib/connect.php';
ConnectDB();
if(! isset($_SESSION['UserId']))
{
header("Location: index.php");

}
if(isset($_SESSION['UserId']))
{
echo"
<ul class=\"nav navbar-nav\">
<li><a href=\"close_session.php\">Вийти</a></li>

```

```

</ul>";
}
else
{
echo"
<ul class=\"nav navbar-nav navbar-right\">
<li><a href=\"login.php\">Увійти</a></li>
<li><a href=\"registr.php\">Реєстрація</a></li>
</ul>";
}

if(isset($_POST['newWord']))
{
$dictionary_name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['dictionary']))));
$original = strtolower(htmlspecialchars(strip_tags(
trim($_POST['original']))));
$translate = strtolower(htmlspecialchars(strip_tags(
trim($_POST['translate']))));
if(strlen($original) > 1 && strlen($translate) > 1){

$user_id = $_SESSION['UserId'];
print $translate;
echo"<br>";
echo mb_detect_encoding($translate);
echo "<br>";
echo $dictionary_name;
echo "<br>";
echo $original;
echo "<br>";

echo " <p style=\"color:#F20C2F\">Коротке слово!<p>";

}

?>

<?php
if(isset($_POST['view']))
{
$dictionary_name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['dictionary']))));
echo"view " . $dictionary_name;

}
?>

<?php
if(isset($_POST['editWord']))
{
$dictionary_name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['dictionary']))));
echo"edit " . $dictionary_name;
$rLogin = mysql_fetch_array($query);

if($rLogin['password'] == $n_password)
{

```

```
}  
?>
```

Файл close_session.php

```
<?php  
session_start();  
  
unset($_SESSION['UserId']);  
unset($_SESSION['UserName']);  
unset($_SESSION['UserLanguage']);  
unset($_SESSION['UserRank']);  
session_destroy();  
header("Location: index.php");  
?>
```

Файл dictionary.php

```
<?php  
session_start();  
require_once '/lib/connect.php';  
ConnectDB();  
if(! isset($_SESSION['UserId']))  
{  
header("Location: index.php");  
  
}  
  
if(isset($_SESSION['UserId']))  
{  
echo"  
<ul class=\"nav navbar-nav\">  
<li><a href=\"close_session.php\">Вийти</a></li>  
</ul>";  
}  
else  
{  
  
<ul class=\"nav navbar-nav navbar-right\">  
<li><a href=\"login.php\">Увійти</a></li>  
<li><a href=\"registr.php\">Реєстрація</a></li>  
</ul>";  
}if(isset($_SESSION['UserId']))  
{  
echo"  
<ul class=\"nav navbar-nav navbar-right\">  
<li><a href=\"\">Слова</a></li>  
<li><a href=\"dictionary.php\">Словники</a></li>  
<li><a href=\"\">Тренування</a></li>  
<li><a href=\"\">Додатки</a></li>  
</ul>";  
}  
else  
{  
echo"  
<ul class=\"nav navbar-nav navbar-right\">  
<li><a href=\"about.php\">Про сайт</a></li>
```

```

<li><a href=\"material.php\">Матеріали</a></li>
<li><a href=\"help.php\">Допомога</a></li>
<li><a href=\"kontakt.php\">Контакти</a></li>
</ul>";
} if(isset($_POST['newDictionary']))
{
$name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['name']))) );
if(strlen($name) > 1) {
$insert = mysql_query("INSERT INTO dictionary(`name`)
VALUES('".$name."')");
$lastId = mysql_insert_id();

$insertUD = mysql_query("INSERT INTO
user_dictionary(`user_id`,`dictionary_id`)
VALUES('".$_SESSION['UserId'].','.$lastId.'')");
echo "збережено!";
else
{
echo " <p style=\"color:#F20C2F\">Коротка назва!<p>";
}

}
while ($row = mysql_fetch_array($number))
{
$name = $row['name'];
echo "<option value=\"".$name\">$name</option>";
}
}

```

Файл material.php

```

<?php
session_start();
require_once '/lib/connect.php';
ConnectDB();
if(! isset($_SESSION['UserId']))
{
header("Location: index.php");

}

if(isset($_SESSION['UserId']))
{
echo"
<ul class=\"nav navbar-nav\">
<li><a href=\"close_session.php\">Вийти</a></li>
</ul>";
}
if(isset($_POST['submit']))
{
echo $_POST['url'] ."<br>";
$url = parse_url_if_valid($_POST['url']);
if (!$url) {

echo "Не вірне посилання, перевірте його будь ласка!";
} else {

```



```

if(isset($_SESSION['UserId']))
{
$UserId = $_SESSION['UserId'];
$insert = mysql_query("INSERT INTO materials(`user_id`,`url`)
VALUES('".$_.$UserId."', '".$_.$url."')");
if($insert){
echo "Ваше посилання: " . $url ." - було запропоноване! Дякуємо!";
}else
{
echo "Не вірне посилання, перевірте його будь ласка ще раз!";
}
}
else
{
$insert = mysql_query("INSERT INTO materials(`url`)
VALUES('".$_.$url."')");
if($insert){
echo "Ваше посилання: " . $url ." - було запропоноване! Дякуємо!";
}else
{
echo "Не вірне посилання, перевірте його будь ласка ще раз!";
}
}
}
}
?>

```

Файл register.php

```

<?php

if(isset($_POST['submit']))
{
echo $_POST['name'] . "<br>";
echo $_POST['email'] . "<br>";
echo $_POST['password'] . "<br>";
echo $_POST['rank'] . "<br>";
echo $_POST['language'] . "<br>";
echo $_POST['password_confirmation'] . "<BR>";

$name = $_POST['name'];
$email = $_POST['email'];
$rank = $_POST['rank'];
$language = $_POST['language'];
$password = $_POST['password'];
$password_repeat = $_POST['password_repeat'];
echo $password;
echo "<br>";
echo $password_repeat;
if($password == $password_repeat)
{
$number = mysql_query("SELECT id FROM language WHERE name =
'$language' ");
$row = mysql_fetch_array($number);
$language_id = $row['id'];
echo $language_id;
}
}
}
}

```

```

$n_password = md5($password);
$insert = mysql_query("INSERT INTO
users(`name`,`email`,`password`,`rank`,`language_id`)
VALUES('".$name."', '".$email."',
'".$n_password."', '".$rank."', '".$language_id.'')");
if($insert){
echo"ура буде авторизація";
$number = mysql_query("SELECT id FROM users WHERE email = '$email'
");
$row = mysql_fetch_array($number);
$user_id = $row['id'];
$_SESSION['UserId'] = $user_id;
$_SESSION['UserName'] = $name;
$_SESSION['UserLanguage'] = $language;
$_SESSION['UserRank'] = $rank;
header("Location: index.php");
}
else
{
echo"Невідома помилка Реєстрації";
}
}
else
{
echo"Паролі не співпадають!";
}
}
}

```

Файл word.php

```

<?php

if(isset($_POST['newWord']))
{
$dictionary_name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['dictionary']))) );
$original = strtolower(htmlspecialchars(strip_tags(
trim($_POST['original']))) );
$translate = $_POST['translate'];
if(strlen($original) > 1 && strlen($translate) > 1){

$user_id = $_SESSION['UserId'];
echo $translate;

echo "<br>";
echo $dictionary_name;
echo "<br>";
echo $original;
echo "<br>";

$query = mysql_query("SELECT dictionary.id as dic_name
FROM user_dictionary
JOIN dictionary ON user_dictionary.dictionary_id = dictionary.id
JOIN users On user_dictionary.user_id = users.id
where dictionary.name = '$dictionary_name' and users.id =
'$user_id'; ");

```

```

$rLogin = mysql_fetch_array($query);
$dictionary_id = $rLogin['dic_name'];

$insert = mysql_query("INSERT INTO word(`original`,`translate`)
VALUES('".$original."','".$translate."')");
$lastId = mysql_insert_id();

$insertWD = mysql_query("INSERT INTO
word_dictionary(`word_id`,`dictionary_id`)
VALUES('".$lastId."','".$dictionary_id."')");

echo "збережено!";

}
else
echo " <p style=\"color:#F20C2F\">Коротке слово!<p>";

}
?>

```

Файл action_edit_testing.php

```

<?php
session_start();
require_once '../lib/connect.php';
ConnectDB();
if(! isset($_SESSION['SuperId']))
{
header("Location: index.php");
}

if(isset($_POST['save_testing']))
{
$language = htmlspecialchars(strip_tags(trim($_POST['language']
)));
echo $language;
$counter = 0;
for($i = 1; $i < 22; $i++){
$question[] = htmlspecialchars(strip_tags(trim($_POST['q' . $i ]
)));
$right[] = htmlspecialchars(strip_tags(trim($_POST['r' . $i ]
)));
$a[] = htmlspecialchars(strip_tags(trim($_POST['a' . $i ]
)));
$b[] = htmlspecialchars(strip_tags(trim($_POST['b' . $i ]
)));
$c[] = htmlspecialchars(strip_tags(trim($_POST['c' . $i ]
)));
$qid[] = htmlspecialchars(strip_tags(trim($_POST['qid' . $i ]
)));
//echo $qid[$counter] . "<br>";

$res = mysql_query ("UPDATE question SET
question.ask = '$question[$counter]',
question.correct = '$right[$counter]',
question.one = '$a[$counter]',
question.two = '$b[$counter]',
question.three = '$c[$counter]'

```

```

WHERE
question.id = '$qid[$counter]'
" ) ;

if($res){
echo " <p>Питання $i збережено!<p>";
}
else
{
echo "Питання $i НЕ ЗБЕРЕЖЕНО ";
die('Invalid query: ' . mysql_error());
}
$counter++;
}
header("Location: index.php");
echo "<p><a href=\"index.php\">На голову</a></p>";

}
else
{
header("Location: testing.php");
}

```

?>

Файл actions_index.php

```

<?php

session_start();
require_once '../lib/connect.php';
ConnectDB();

if(isset($_POST['submit']))
{
$name = strtolower(htmlspecialchars(strip_tags(
trim($_POST['name']))) );
$password =
strtolower(htmlspecialchars(strip_tags(trim($_POST['password']))) );
$n_password = md5($password);
$query = mysql_query("SELECT password FROM admin WHERE name =
'$name' ");
$rLogin = mysql_fetch_array($query);

if($rLogin['password'] == $n_password)
{

$_SESSION['SuperId'] = $n_password;
header("Location: index.php");

}
else
echo " <p style=\"color:#F20C2F\">Не вірний пароль або логін!<p>";
}

```

?>

Файл edit_testing.php

```
<?php
if(isset($_POST['edit_testing']))
{
    $language = htmlspecialchars(strip_tags(trim($_POST['language']
    )));
    $number = mysql_query("SELECT id FROM language WHERE name =
    '$language' ");
    $row = mysql_fetch_array($number);
    $language_id = $row['id'];
    echo $language_id;

    $number = mysql_query("SELECT language.id as id_lan,
    language_question.question_id as lan_qu_id,
    question.id as qu_id, question.ask as qu_ask, question.correct as
    qu_corr,
    question.one as qu_one, question.two as qu_two, question.three as
    qu_three
    FROM language_question
    JOIN language ON language_question.language_id = '$language_id'
    JOIN question On language_question.question_id = question.id
    ");

    $counter = 0;
    $num = 1;
    while ($row = mysql_fetch_array($number))
    {
        $id_lan = $row['id_lan'];
        $qu_id[] = $row['qu_id'];

        $question[] = $row['qu_ask'];
        $right[] = $row['qu_corr'];
        $a[] = $row['qu_one'];
        $b[] = $row['qu_two'];
        $c[] = $row['qu_three'];
        header("Location: testing.php");
    }
}
```

Файл testing.php

```
<?php

if(isset($_POST['submit']))
{
    $language = htmlspecialchars(strip_tags(trim($_POST['language']
    )));

    for($i = 1; $i < 22; $i++){
        $question[] = htmlspecialchars(strip_tags(trim($_POST['q' . $i ]
        )));
        $right[] = htmlspecialchars(strip_tags(trim($_POST['r' . $i ]
        )));
        $a[] = htmlspecialchars(strip_tags(trim($_POST['a' . $i ]
        )));
        $b[] = htmlspecialchars(strip_tags(trim($_POST['b' . $i ]
        )));
        $c[] = htmlspecialchars(strip_tags(trim($_POST['c' . $i ]
        )));
    }
}
```

```

$number = mysql_query("SELECT name FROM language WHERE name =
'$language' ");
$num_rows = mysql_num_rows($number);

if($num_rows <= 0)
{
$insert = mysql_query("INSERT INTO language(`name`)
VALUES('".$language."')");
if($insert)
{
$lastId = mysql_query("SELECT id FROM language ORDER BY id DESC
LIMIT 1");
$resLastId = mysql_fetch_array($lastId);
$lastIdLanguage = $resLastId['id'];
for($i = 0; $i < 21; $i++){
$insertQuestion = mysql_query("INSERT INTO
question(`ask`,`correct`,`one`,`two`,`three`)
VALUES('".$question[$i]."', '".$right[$i]."', '".$a[$i]."', '".$b[$i].
"', '".$c[$i]."'");
if($insertQuestion){
$lastId = mysql_query("SELECT id FROM question ORDER BY id DESC
LIMIT 1");
$resLastId = mysql_fetch_array($lastId);
$lastIdQuestions = $resLastId['id'];

$insertLanQuest = mysql_query("INSERT INTO
language_question(`language_id`,`question_id`)
VALUES('".$lastIdLanguage."', '".$lastIdQuestions."')");
}
}
echo " <p style=\"color:#5DDD2C\">Мову: $language додано.
Тестування створено! <p>";
}
else
{
echo " <p style=\"color:#F20C2F\">Така мова вже зареєстрована
!<p>";
}

$id++;
$insert = mysql_query("INSERT INTO
projects(`Id`,`Name`,`NumberOfShares`,`NumberOfPrice`,`Comisiya`,`A
bout`,`Date`,`Dividends`,`Status`,`IPO`,`BuyOrders`,`SellOrders`,`R
ecentOrderexecutions`,`Owner`,`PublicShares`,`WebPage`,`CompletelyN
ame`,`Mark`)
VALUES('".$id."', '".$name."', '".$numbershares."', '".$numberprice."'
, '".$comis."', '".$about."', '".$today."', '".$name."', '".$status."',
'".$ipo."', '".$TableBuyOrders."', '".$TableSellOrders."', '".$TableRec
entOrderexecutions."', '".$owner."', '".$publicshares."', '".$webpage.
mysql_query("ALTER TABLE projects ORDER BY Id");
}
else
{
echo " <p style=\"color:#F20C2F\">Така мова вже зареєстрована!<p>";
}
}?>

```