**ЗМІСТ**

Перелік позначень та скорочень5

Вступ6

1 Аналіз предметної області онлайн системи бюджетного обліку 7

1.1 Опис предметної області онлайн системи бюджетного обліку 7

1.2 Аналіз аналогів 8

1.3 Технічне завдання 9

2 Розробка бази даних 12

2.1 Розробка системи бізнес-правил 12

2.2 Розробка схеми бази даних 13

2.3 Реалізація бази даних 14

3 Розробка програмного забезпечення 17

3.1 Опис засобів розробки прикладного програмного забезпечення17

3.2 Розробка прикладного програмного забезпечення17

4 Використання розробленого програмного забезпечення для автоматизації системи бюджетного обліку 20

4.1 Послідовність дій при розгортанні розробленого застосунку на сервері20

4.2 Робота користувача з програмним забезпеченням 21

Висновки25

Список джерел інформації26

Додаток А Інформація, введена в таблиці бази даних 28

**ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

БД – база даних

СУБД – система управління базами даних

ПЗ – програмне забезпечення

SQL – Structured query language

IT – information technologies

UML – Unified Modeling Language

**ВСТУП**

На сьогоднішній день проблема розуміння фінансової грамотності стоїть у багатьох людей, сімей, компаній на всій планеті, що погано впливає на їх капітал. Вони не зможуть відповісти навіть на звичайне питання про те, скільки вони в середньому витрачають на місяць. Розуміння звідки надходять та куди витрачаються гроші – основний крок до фінансового благополуччя особи або групи осіб. Наприклад, аналіз витрат дозволить легше виставити пріоритети куди слід витрачати гроші. Таким чином розробка різноманітних систем бюджетного обліку є дуже актуальною.

Метою курсової роботи є автоматизація процесу ведення обліку розходів і доходів. Об’єкт дослідження – онлайн система бюджетного обліку.

У ході виконання курсової роботи необхідно проаналізувати предметну область і розробити програмне забезпечення для вирішення питання автоматизації процесу бюджетного обліку, що буде орієнтоване на кінцевих користувачів без особливої підготовки.

**1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІДЕЯ РЕАЛІЗАЦІЇ ОНЛАЙН СИСТЕМИ БЮДЖЕТНОГО ОБЛІКУ**

**1.1 Опис онлайн системи бюджетного обліку**

Онлайн система бюджетного обліку - застосунок, що допомагає користувачу вести та переглядати історію власних, або групових платежів, що дозволяє більш ефективно аналізувати і планувати свій бюджет. Усі записи в системі бюджетного обліку мають такі характеристики, як опис, дата й час створення запису, кількість отриманих чи витрачених коштів. Пройшовши реєстрацію у програмному застосунку, користувач стає частиною (user) бази даних і має можливість віртуально вести історію власного, або сімейного бюджету. Застосунок пропонує свої можливості попередньо запропонувавши увійти, або пройти реєстрацію, інакше користуватись застосунком не можна.

Система дозволяє змінювати, доповнювати, фільтрувати і переглядати інформації про витрати та доходи, накладати обмеження доступу до історії коштів, зберігати списки гаманців з різними валютами.

Також ця система може відображати графіки витрат та доходів й інші статистичні моделі.

Усі вище описані функції буде зручно візуалізувати у вигляді діаграми варіантів використання[1], що представлена на рисунку 1.5.

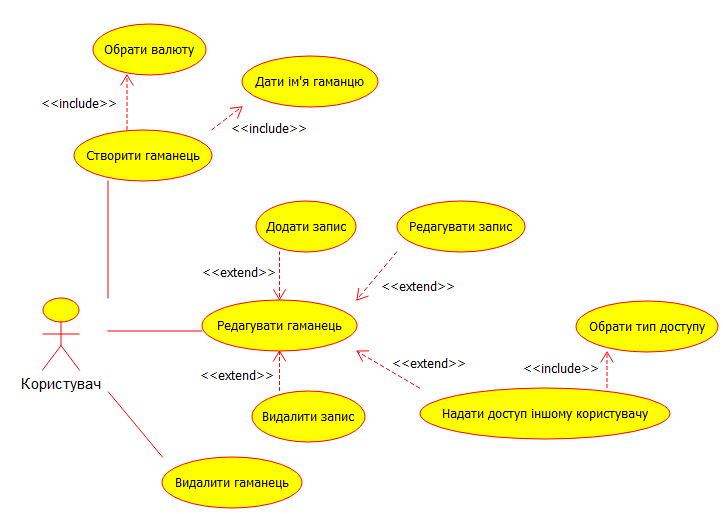


Рисунок 1.5 – Основні можливості користувачів програмного забезпечення представлені у вигляді діаграми варіантів використання

**1.2 Аналіз аналогів**

Шукаючи аналогічні проєкти даної предметної області, вийвилось що їх не так багато, як і їх функціонал.

Наприклад «Домашні фінанси» – система для обліку особистих і сімейних фінансів, ведення домашньої бухгалтерії. На рисунку 1.6 наведено скриншот головної сторінки аккаунта на сайті «Домашні фінанси».

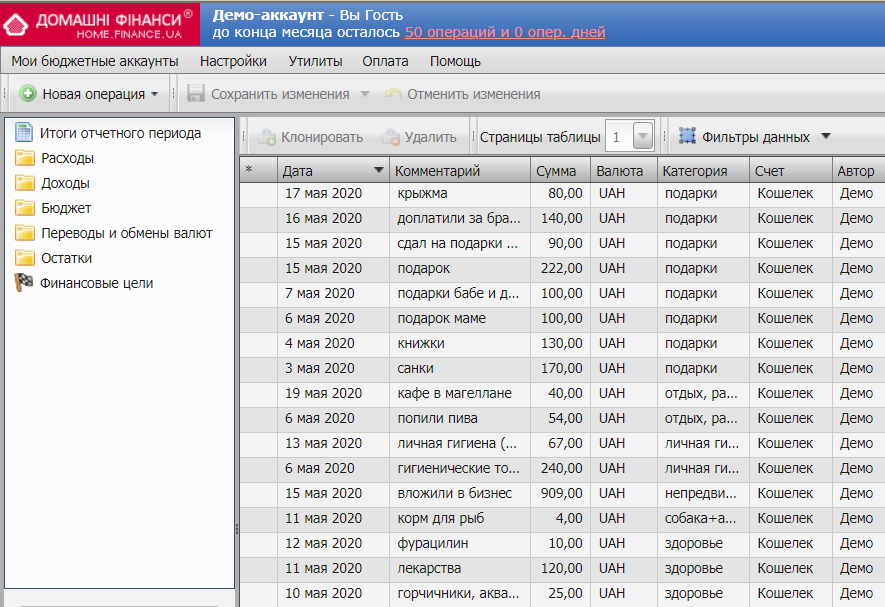


Рисунок 1.6 – Скриншот головної сторінки аккаунта на сайті «Домашні фінанси»

Для зареєстрованих користувачів надається можливість створювати нові операції, в свою чергу до яких можна додавати таблиці та діаграми. Також є можливість ставити фінансові цілі. Нажаль цей проєкт з першого травня 2020 року призупиняється, а з наступного року видаляється назавжди. Ця ситуація дала ідею введення платного статусу акаунтів у майбутніх версіях власного проєкта.

**1.3 Технічне завдання**

Призначення продукту

Програмне забезпечення призначене для процесу бюджетного обліку власного, або групового капіталу.

Перспектива продукту

Розроблене програмне забезпечення може бути порівняно з будь якими невеликими та безкоштовними онлайн системами бюджетного обліку, але з унікальною можливістю давати різний тип доступу іншим користувачам, та має великий потенціал у разі продовження роботи над ним.

Функції продукту

* 1. Реєстрація на сайті.
  2. Вхід у програму під своїм логіном.
  3. Перегляд списку доступних гаманців.
  4. Створення гаманця.
  5. Робота зі списком записів вибраного гаманця.
  6. Надати доступ до гаманця іншому користувачу.
  7. Видалення гаманця.

Функціональні вимоги наведені у таблиці 1.1

Таблиця 1.1 – Функціональні вимоги

| Функція | Опис функції |
| --- | --- |
| Реєстрація на сайті | Система повинна дозволяти новим користувачам реєструватися у системі. Реєстрація не має бути доступною для вже зареєстрованих користувачів. Для реєстрації необхідні: прізвище, ім’я, електронна адреса, логін і пароль. |
| Вхід у програму під своїм логіном | Система повинна дозволяти вже зареєстрованим користувачам входити в программу. Для входу необхідні логін і пароль. Якщо введений неправильний пароль для відповідного логіна або логін якого не має в базі даних, вхід в систему неможливий. |
| Перегляд списку доступних гаманців | Система повинна виводити список доступних гаманців. Мають бути виведені такі характеристики, як назва та валюта. |
| Створення гаманця | Система має давати можливість оформити гаманець. При оформленні гаманця має бути вказана його валюта. |
| Робота зі списком записів вибраного гаманця | Система має давати можливість користувачам переглядати, доповнювати, редагувати, видаляти записи вибраного гаманця відповідно до рівня доступу користувача. |
| Надати доступ до гаманця іншому користувачу | Система має давати можливість користувачу надавати певний рівень доступу до гаманця іншим користувачам. |
| Видалення гаманця | Система має давати можливість користувачу видалити гаманець. |

Характеристики користувачів

Клієнт не має мати спеціальних навичок, щоб користуватися даною системою, оскільки вся робота проходть в программному забезпеченні. Клієнтська частина має інтуітивно зрозумілий інтерфейс, який не стане перешкодою для більшості користувачів комп’ютерів.

Нефункціональні вимоги

1. Робота в сучасних браузерах Internet Explorer 5.5 і вище, Mozilla Firefox7.01 і вище, Safari 5.1.7 і вище, Google Chrome 13.0.772.0.
2. Клієнтські вимоги для Windows: Windows XP з пакетом 2 або пізнішої версії; Windows Vista; Windows 7; Windows 8; Windows 10.
3. Клієнтські вимоги для Mac: Mac OS X 10.6 або більш пізня версія.
4. Клієнтські вимоги для Linux: Ubuntu 10.04+; Debian 6 +.
5. Наявність процесора для Windows: Intel Pentium 4 або пізнішої версії.
6. Наявність процесора для Mac: Intel.
7. Наявність процесора для Linux: Intel Pentium 3 або пізнішої версії.
8. Вільне місце на диску для Windows: 100 МБ.
9. Вільне місце на диску для Mac: 100 МБ.
10. Вільне місце на диску для Linux: 100 МБ.
11. Клієнтські вимоги до ОЗУ для Windows: 128 МБ.
12. Клієнтські вимоги до ОЗУ для Mac: 128 МБ.
13. Клієнтські вимоги до ОЗУ для Linux: 128 МБ.
14. Робота в сучасних браузерах Internet Explorer 5.5 і вище, Mozilla Firefox7.01 і вище, Safari 5.1.7 і вище, Google Chrome 13.0.772.0.
15. Всі інтернет-сторінки, які генеруються системою, повинні повністю завантажуватися щонайбільше за 10 секунд зі швидкістю з’єднання 5Мб/с.
16. Система має витримувати до 200 користувачів одночасно.

Обмеження

Для стабільної роботи сервера, на якому буде використовуватися програмне забезпечення, необхідно встановити php-7.4.6 і СУБД, що працює з версіями mysql не нижче 5.0.

**2 РОЗРОБКА БАЗИ ДАНИХ**

**2.1 Розробка системи бізнес-правил**

В результаті дослідження та аналізу предметної області було сформовано наступну систему бізнес-правил.

Факти.

1. Користувач має мати унікальний код та логін, пароль, електронну адресу, прізвище та ім’я.
2. Гаманець повинне мати унікальний код, валюту, код власника та назву.
3. Запис має такі характеристики як унікальний код, сума, тип операції, дату створення та опис.

Обмеження.

1. Користувач бачить тільки доступні йому гаманці.
2. Користувач може робити тільки ті дії, що відповідають його рівню доступу.
3. Сума в записі – дійсне позитивне значення.

Активатори операцій.

1. При наданні доступу іншим користувачам, перевіряється власний доступ до гаманця. Якщо користувач надає більший рівень доступу ніж має, оформлення надання доступу – недоступне.

**2.2 Розробка схеми бази даних у Erwin Data Modeler**

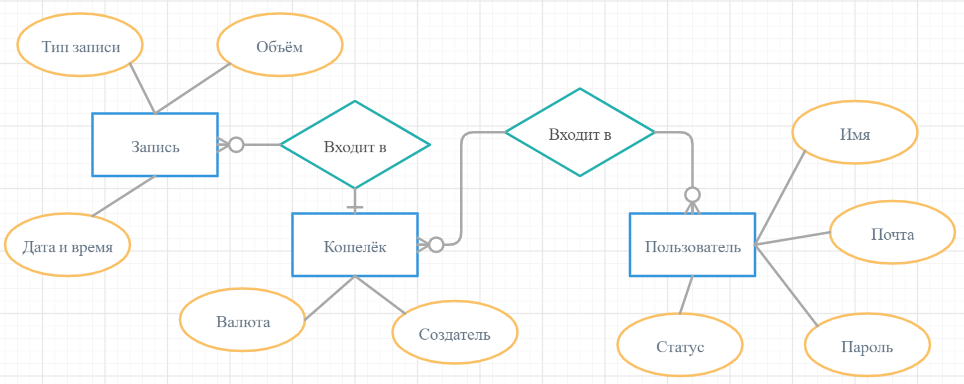


Рисунок 2.1 – Концептуальна схема бази даних

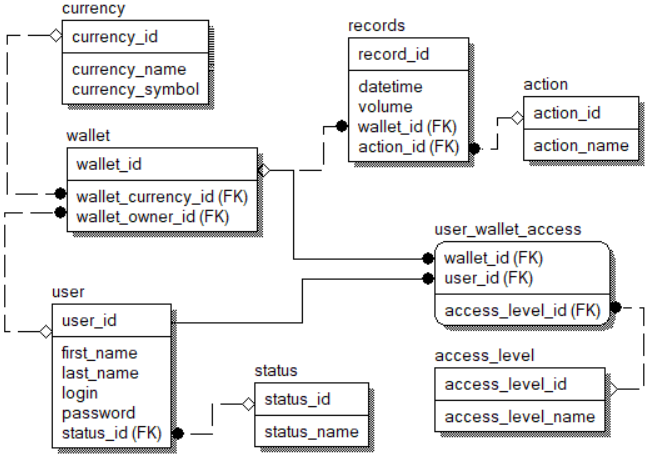


Рисунок 2.2 – Логічна схема бази даних створена у Erwin Data Modeler

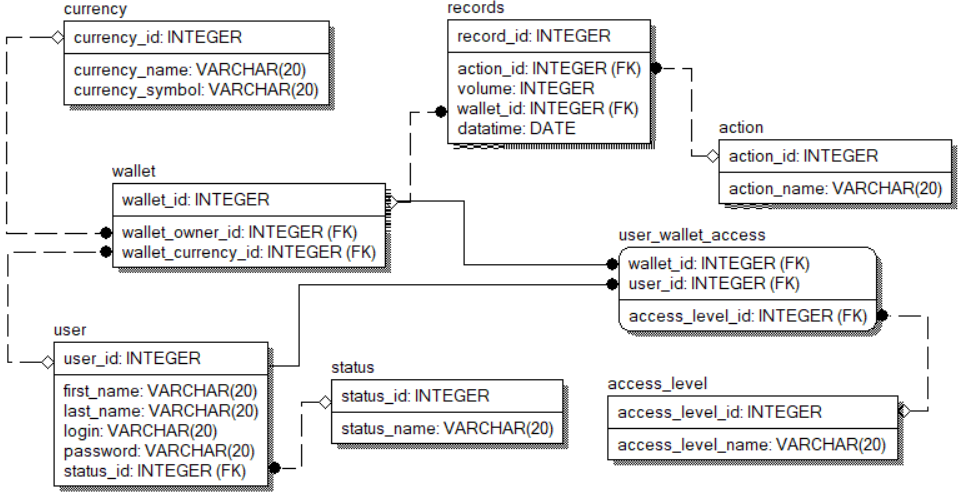
****

Рисунок 2.3 – Фізична модель даних

**2.3 Реалізація бази даних**

На основі моделі даних створено базу даних[2], з урахуванням вимог і обмежень бізнес-правил, а також вимог, зумовленими обраною предметною областю, а саме вимоги доступу користувачів програми до гаманців. База даних була створена в СУБД MySQL[3]. Ця серверна система здатна ефективно функціонувати у взаємодії з інтернет-сайтами та веб-додатками. Крім універсальності і поширеності, СУБД MySQL володіє цілим комплексом важливих переваг перед іншими системами. Зокрема слід відзначити такі якості як: простота у використанні, великий функціонал, безпека, можливість масштабування, швидкість.

Структура створеної бази даних зображена на рисунку 2.4.

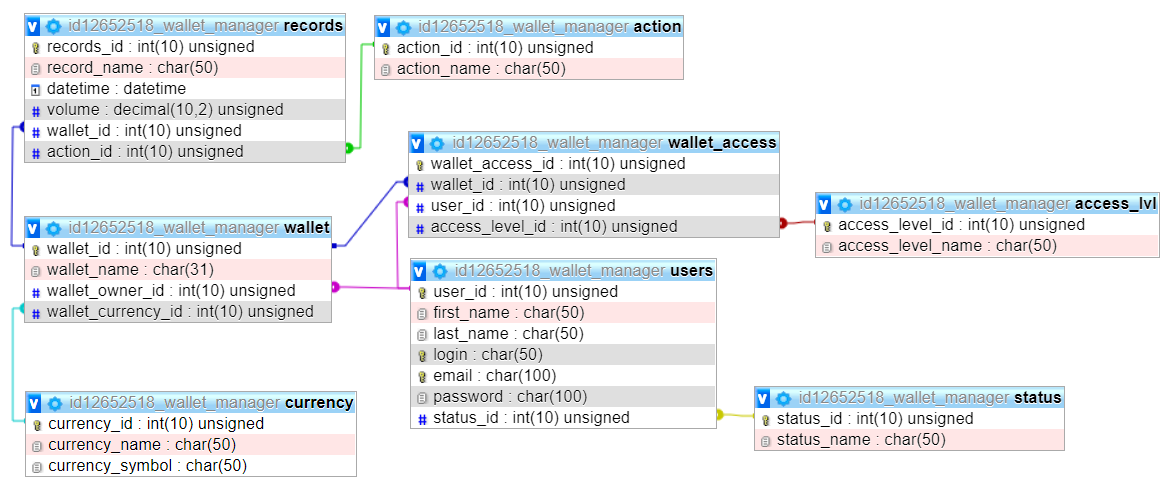


Рисунок 2.4 – Зображення створеної бази даних.

Структури основних створених таблиць, що описані в моделі бази даних зображені на рисунках 2.5 – 2.11.



Рисунок 2.5 – Зображення структури створеної таблиці records.

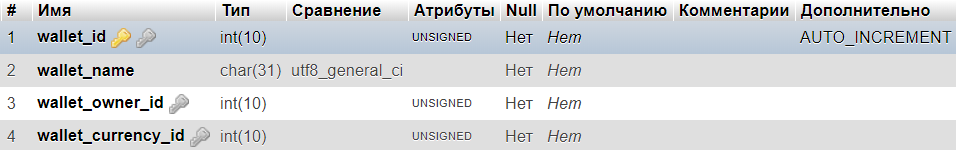


Рисунок 2.6 – Зображення структури створеної таблиці wallet.



Рисунок 2.7 – Зображення структури створеної таблиці users.



Рисунок 2.8 – Зображення структури створеної таблиці wallet\_access.

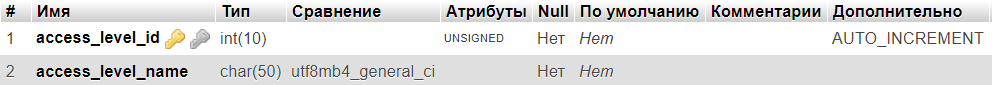


Рисунок 2.9 – Зображення структури створеної таблиці access\_lvl.

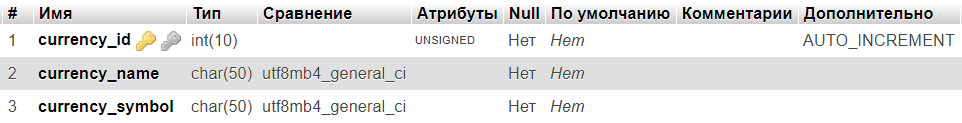


Рисунок 2.10 – Зображення структури створеної таблиці currency.

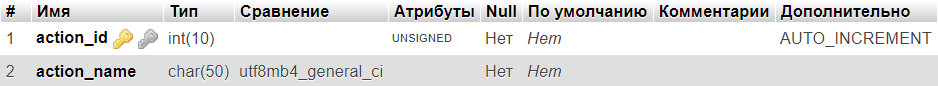


Рисунок 2.11 – Зображення структури створеної таблиці action.

**3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**3.1 Опис стеку технологій і засобів розробки прикладного програмного забезпечення**

Оскільки ПЗ будуть використовувати багато людей, то необхідна СУБД, що підтримує багатопотоковість. Це буде реляційна БД[4], якою можна керувати за допомогою мови SQL. Оскільки ПЗ, що створюється не фінансуєтеся кимось, то відповідно СУБД повинна бути безкоштовною. Оскільки всім цим параметрам відповідає безкоштовний хостинг webhost000[5].

Розробка ПЗ буде вестися за допомогою Python[6], що представляє собою мову програмування, яка перша приходить на думку при обробці великих даних.

**3.2 Розробка прикладного програмного забезпечення**

Основні функції програмного забезпечення можна представити за допомогою use case діаграми, що зображена на рисунку 3.1.

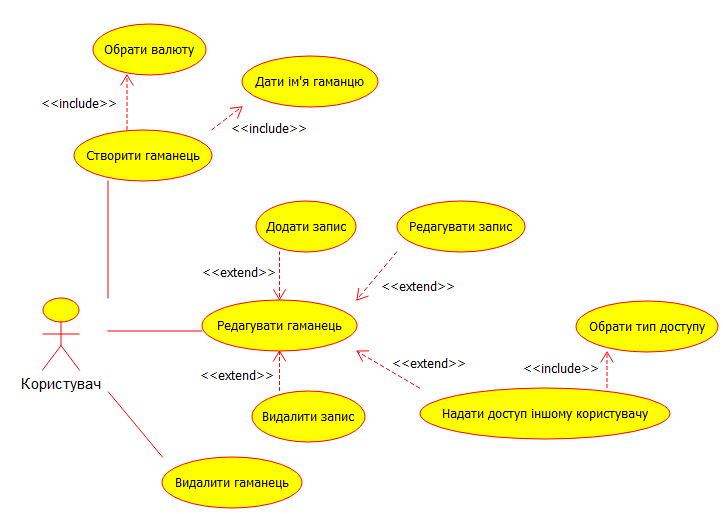


Рисунок 3.1 – Use case діаграма основних функцій ПЗ

В таблицях 3.1, 3.2 наведені основні специфікації для головної функціональної вимогу до програмного забезбечення – оформлення замовлень клієнтом.

Таблиця 3.1 – Головний розділ сценарію виконання варіанта використання «Створення гаманця»

|  |  |
| --- | --- |
| Варіант використання | Створення гаманця. |
| Актори | Зареєстрований користувач. |
| Мета | Створити гаманець. |
| Короткий опис | Користувач створює гаманець з певною валютою та назвою. |
| Тип | Базовий. |

Таблиця 3.2 Типовий хід подій сценарію виконання варіанта використання «Редагування гаманця»

|  |  |
| --- | --- |
| Дія актора | Відгук системи |
| Користувач редагує записи гаманця  Користувач надає доступ іншому користувачу  Виключення 1: користувач не має доступу до дії, яку намагається виконати | Система перевіряє доступність дії для користувача |

Потік повідомлень між структурними елементами системи і основні асоціації між ними під час редагування записів користувачем можна представити у вигляді діаграми кооперації[7], що приведена на рисунку 3.2.

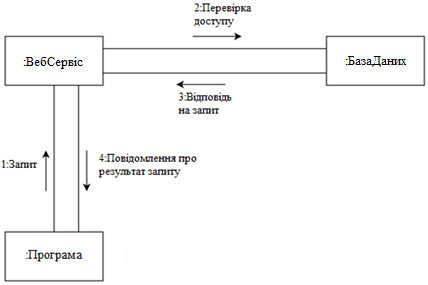


Рисунок 3.2 – Потік повідомлень між структурними елементами системи і основні асоціації між ними під час редагування записів користувачем

# 4 ВИКОРИСТАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ СИСТЕМИ БЮДЖЕТНОГО ОБЛІКУ

**4.1 Послідовність дій при розгортанні розробленого застосунку на сервері**

На сервері має бути встановлена СУБД, що працює з останніми версіями MySQL. Це може бути XAMPP Control Panel, Mysql Workbench, або вебхостинг із уже готовою реалізацією. Створення і заповнення бази даних буде описано за допомогою командного рядка оскільки він буде присутній у кожній СУБД. У обраній СУБД необхідно створити базу даних “wallet\_manager” за допомогою команди “create database wallet\_manager;”. Далі слід послідовно скопіювати у командний рядок текст з файлу localhost.sql.

Також на сервері має бути встановлений PHP з офіціальної сторінки цього продукту[8]. На сервер необхідно перенести файл public\_html, що є папкою зі сторінкою реєстрації та веб оброблювачем запитів програми користувача. Після цього можна відкрити браузер і перевірити працездатність системи, ввівши в адресний рядок http://localhost:8080. Користувачу даного програмного забезпечення слід завантажити виконуючий файл «Money Manager.exe», що є графічним інтерфейсом для взаємодії користувача з веб-сервісом для роботи з бюджетом гаманців у вигляді записів.

**4.2 Робота користувача з програмним забезпеченням**

Вперше відкривши застосунок, користувач може увійти у свій обліковий запис, використовуючи логін і пароль або пройти реєстрацію.

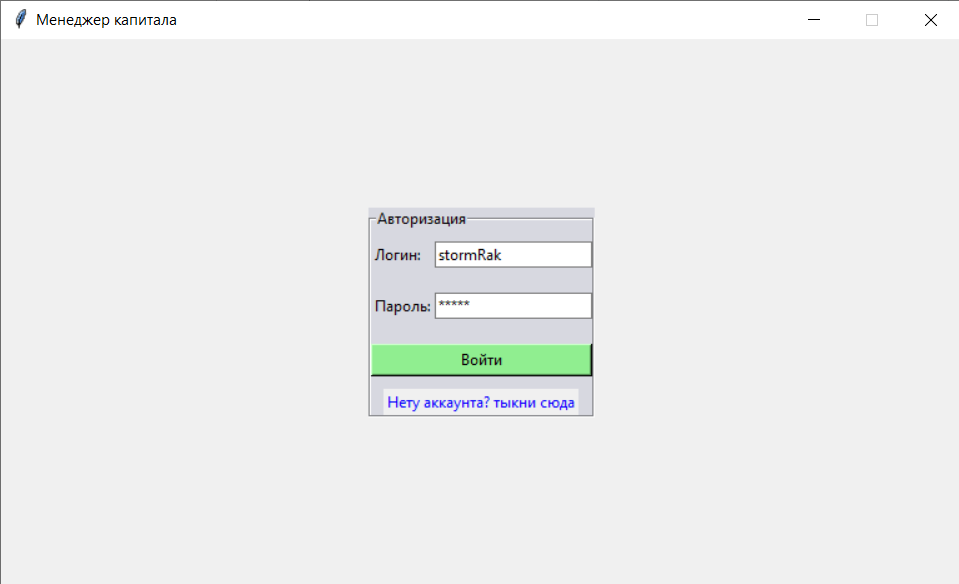
****

Рисунок 4.1 – Форма входу користувача у систему

Новий користувач може пройти реєстрацію. Для цього він має ввести ім’я, фамілію, логін, електрона адреса та пароль. Логін та електрона адреса мають бути унікальними, відсутність пароля або логіна не допускається. Якщо один з цих критеріїв не задовольняється, користувач побачить відповідне повідомлення.

Форма реєстрації відкривається при натисканні на відповідну гіперпосилання в програмі.

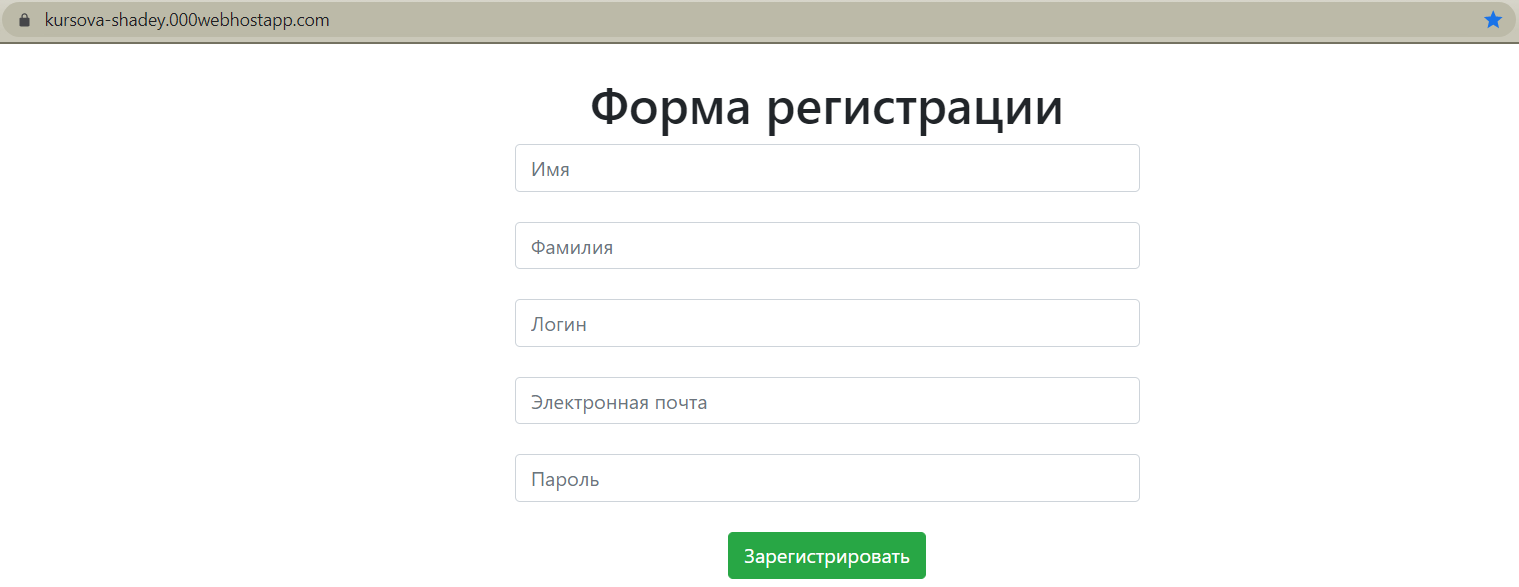
****

Рисунок 4.2 – Форма реєстрації користувача

Увійшовши у систему користувач має змогу переглянути список гаманців чи створити новий. Якщо у користувача не було гаманців, то він автоматично перейде на сторінку створення.

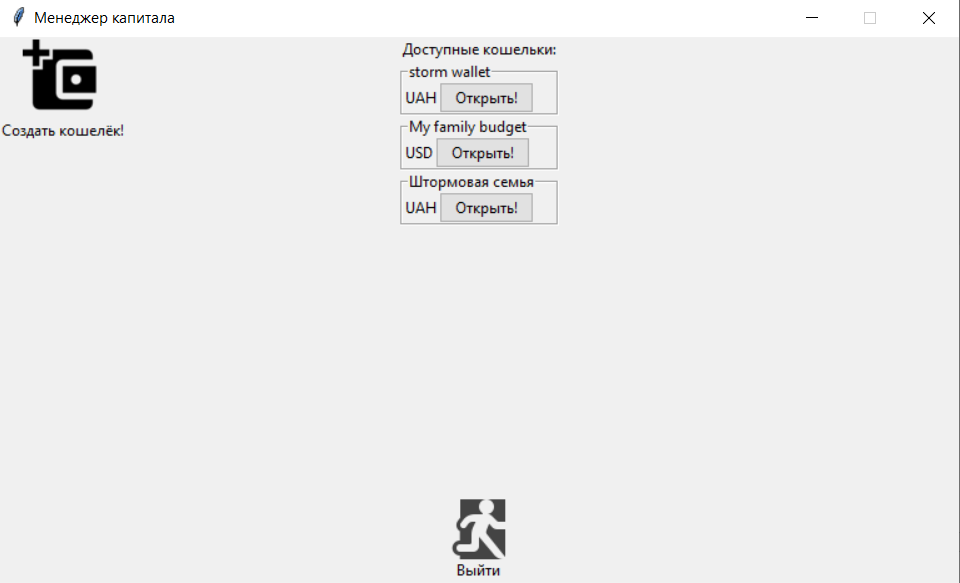


Рисунок 4.3 – Список гаманців, доступних користувачу

Відкривши гаманець, користувач бачить стан балансу, назва гаманця, рівень доступу, список записів та можливі операції для роботи з ними й гаманцем.

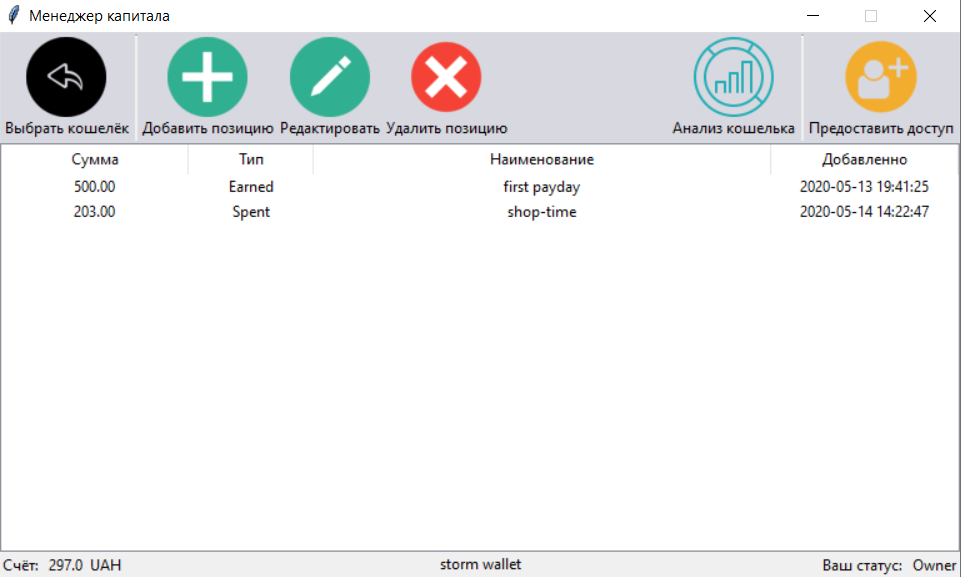


Рисунок 4.4 – Форма створення запису

Натиснувши на відповідну кнопку користувач має змогу створити новий, редагувати існуючий, або видалити запис, при наявності у нього доступу до певної операції.

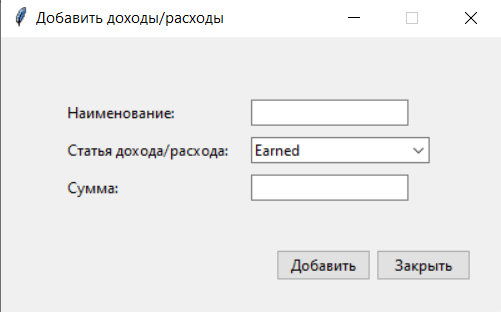


Рисунок 4.5 – Форма створення запису

Якщо користувач не введе некоректних даних, а саме, значення меншого за нуля, то запис буде додано після натискання відповідної кнопки. У разі введення некоректних даних буде видано відповідне повідомлення.

Надати доступ іншому користувачу можна натиснувши на кнопку в правому верхньому кутку, та ввівши хоча б частину логіна користувача, якого потрібно наділити правами доступу.

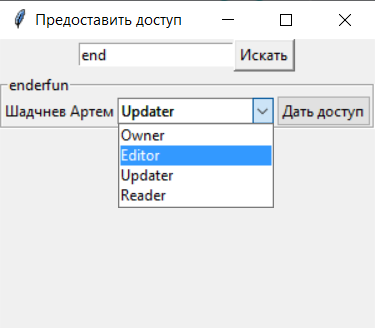


Рисунок 4.6 – Вигляд меню надання доступу

Вийти з облікового запису можна за допомогою кнопки знизу в меню вибору гаманців.

**ВИСНОВКИ**

Розроблене в ході курсової роботи програмне забезпечення реалізує вирішення задачі автоматизації онлайн системи ведення бюджетного обліку. Воно може використовуватись в багатьох галузях людського життя, де приймають участь гроші. Функціонал програми дозволяє працювати над введенням власного чи групового бюджету з дотриманням захисту даних від зловмисників. Безпека створюється завдяки зберіганню хешированного пароля, розподілення рівнів доступу нових користувачів, та обмежень ПЗ, завдяки яким воно лише відправляє запити на веб-сервіс, отримує відповіді та не має реалізації взаємодії саме зі СУБД, щоб у разі декомпіляції програми, неможливо було отримати дані для доступу до СУБД.

Програма може розширювати функціонал в майбутньому, особливо сама програма, куди можна додати різні варіанти діаграм по існуючим коштам, також планується отримати ліцензію, для комерційної вигоди продукту – планується введення платних статусів для користувачів та гаманців, що дасть більш розширений функціонал.

Таким чином дана робота може вважатися живим проєктом, через те що його супроводження продовжиться, як унікального продукту на ринку обраної предметної області – онлайн система автоматизації ведення бюджетного обліку.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Буч Гради. Введение в UML от создателей языка / Гради Буч , Джеймс Рамбо , Ивар Якобсон. - М.: ДМК Пресс, 2015. - 496 c.
2. Генри Хансен. Базы данных: разработка и управление / Хансен Генри, Хансен Джеймс. - М.: Бином, 2015. - 704 c.
3. Getting Started with MySQL //

<https://dev.mysql.com/doc/mysql-getting-started/en/> , 14.05.2020.

1. Ульман Дж. Основы систем баз данных - М.: Финансы и статистика, 2017. - 292 c.
2. Веб хостинг webhost000 //

https://www.000webhost.com/ , 14.05.2020.

1. Головна сторінка Python //

<https://www.python.org/> , 14.05.2020.

1. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку - М.: Вильямс, 2013. - 736 c.
2. Головна сторінка PHP //

<https://www.php.net/> , 14.05.2020.

ДОДАТОК А

Дані введені до таблиць бази даних

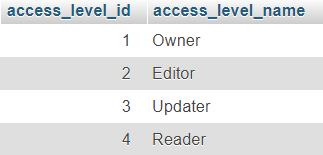


Рисунок A.1 – Дані, введені в таблицю «access\_lvl»

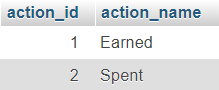


Рисунок A.2 – Дані, введені в таблицю «action»



Рисунок A.3 – Дані, введені в таблицю «currency»



Рисунок A.3 – Дані, введені в таблицю «status»