**Анотація:** в курсовій роботі розроблений сервіс для тренування пам’яті «Memory.pro». Для розробки програмного забезпечення було використано мову програмування Java, бібліотеку JavaFX (з використанням FXML та SASS), JPA фреймворк EclipseLink та СУБД MYSQL.

**Аннотация:** в курсовой работе разработан сервис для тренировки памяти «Memory.pro». Для разработки программного продукта был использован язык программирования Java, библиотека JavaFX (с использование FXML и SASS), JPA фреймворк EclipseLink и СУБД MYSQL.

**Annotation:** a software product for memory training named «Memory.pro» is developed in this coursework. Java programing language, library JavaFX (with FXML and SASS), JPA framework EclipseLink and DBMS MYSQL have been used.

Зміст

[Вступ 4](#_Toc9869164)

[1. Аналіз та опис предметної області 5](#_Toc9869165)

[1.1. Загальний опис предметної області 5](#_Toc9869166)

[1.2. Аналіз вже існуючого програмного забезпечення 5](#_Toc9869167)

[1.3. Дослідження та опис предметної області 6](#_Toc9869168)

[1.4. Вибір засобів розробки 8](#_Toc9869169)

[2. Проектування 9](#_Toc9869170)

[2.1. Ескізний проект 9](#_Toc9869171)

[2.1.1. Схема взаємодії модулів 9](#_Toc9869172)

[2.1.2. Діаграма варіантів використання 10](#_Toc9869177)

[2.2. Технічний проект 11](#_Toc9869178)

[2.2.1. Контекстна діаграми системи 11](#_Toc9869179)

[2.2.2. DFD 1 рівня 11](#_Toc9869180)

[2.2.3. DFD 2 рівня (Статистика) 12](#_Toc9869181)

[2.2.4. Діаграма сутність-зв’язок (ERD) 13](#_Toc9869182)

[2.2.5. Логічна модель системи 14](#_Toc9869183)

[2.2.6. Діаграма класів 15](#_Toc9869184)

[2.2.10 Діаграма послідовностей 18](#_Toc9869185)

[Додаток А. Технічне завдання 20](#_Toc9869186)

[Додаток Б. Інструкція користувача 37](#_Toc9869187)

[Додаток В. Код програмних модулів 45](#_Toc9869188)

# **Вступ**

Пам’ять – невід’ємна частина життя людини. Кожного для ми зустрічаємось з незліченною кількістю інформації, і з кожним роком її кількість тільки зростає. А людині, щоб бути успішною, потрібно постійно вчитися, постійно отримувати все нові і нові знання та навички. Але скільки людина дійсно запам’ятовує? А скільки того, що запам’ятовує вона забуває через день, місяць, рік?

Метою роботи є розробка сервісу для тренування пам’яті «Memory.pro», який призначений допомогти користувачам покращити свою пам’ять. А саме – тренування з запам’ятовування різних типів інформації (таких як числа, слова тощо), ведення статистики тренувань користування та рейтингу користувачів сервісу.

Для досягнення мети необхідно вирішити наступні завдання:

* Провести аналіз предметної області та вже існуючих аналогів програмного забезпечення.
* Розробити проект програмного забезпечення, який містить в собі:

1. Розробку, концептуальної, логічної, фізичної моделі бази даних.
2. Розробку специфікації програмних модулів.
3. Опис алгоритмів програми та розробку інтерфейсу користувача.
4. Опис СУБД та засобів програмної реалізації.
5. **Аналіз та опис предметної області**
   1. **Загальний опис предметної області**

Сервіс «Memory.pro» призначений для тренування та розвитку пам’яті. Основна функція сервісу – тренування з запам’ятовування даних різних типів, таких як цифри, слова, картинки, імена, дати, тощо. При виконанні курсової роботи будуть реалізовані два – цифри та слова.

Під час тренування користувачу необхідно запам’ятати максимальну кількість даних вибраного типу за мінімальний час, після чого відновити їх у тому ж порядку.

Після запам’ятовування формується результат тренування – що користувач запам’ятав правильно, що не правильно, скільки часу він запам’ятовував запропоновану інформацію тощо (детальніше у розділі «результати тренування»

На основі результатів користувачів формується статистика – статистика конкретного користувача та загальна статистика.

* 1. **Аналіз вже існуючого програмного забезпечення**
     1. memoryleague.com

Міжнародний сервіс для змагань із запам'ятовування Online Memory League Championship. Ним користуються багато кращих мнемоністів світу. Має всі базові типи тренувань, досить зручний інтерфейс.

Сервіс повністю англомовний. Є платним ($ 25 в рік, або $ 10 на три місяці). Сервісом можна користуватися безкоштовно, але тільки в демонстраційному режимі.

В першу чергу розрахований на швидкісне запам'ятовування невеликої кількості даних (наприклад, 80 цифр, 30 картинок). Має хорошу систему змагань (через цей сервіс проводиться один з чемпіонатів світу із запам'ятовування).

* + 1. memoryman.ru

Відносно новий сервіс по тренуванню пам'яті. Сервіс російськомовний. Містить всі основні типи тренувань, досить гнучке налаштування показу даних, кількості елементів. Має простий і досить зручний інтерфейс, мінімалістичний дизайн.

На даний момент найбільш популярний в російськомовному співтоваристві мнемоністів. Але має велику кількість помилок і недоробок.

* + 1. Powermemory.ru

Один з перших російськомовних сервісів по тренуванню пам'яті (в контексті мнемоніки). Надійний і стабільний, але містить тренування тільки із запам'ятовування чисел, карт і слів. Слабка можливість настройки тренування. З 2017 року сервіс закритий на оновлення.

* 1. **Дослідження та опис предметної області**

Основним користувачем сервісу є «Користувач» (або «User»). Він характеризується наступними атрибутами:

* Id
* Прізвище
* Ім’я
* Username
* Email
* Дата реєстрації

Користувачем може стати будь-який бажаючий (будь-якого віку, громадянства, національності). У кожного користувача повинен бути особистий кабінет (профіль), в якому він може бачити свої дані, а також може змінювати їх. Користувачі можуть бачити профілі інших користувачів.

Тренування

Тренування - основна функція сервісу. Тренування складається з наступних етапів:

1. Вибір тренування. На цьому етапі користувачу необхідно вибрати необхідний вид тренування і вказати кількість елементів інформації у тренуванні.
2. Запам'ятовування випадкових даних відповідно до типу тренування. Користувачеві показується випадкова послідовність даних зазначеного типу відповідно до встановлених на попередньому етапі параметрам. Користувач послідовно передивляється кожен елемент та запам’ятовує його.
3. Згадування. Після запам'ятовування користувачеві пропонується відтворити дані, які вдалося запам’ятати. Відтворити їх необхідно в тому ж порядку в якому вони були показані під час запам'ятовування.
4. Результати. Після того, як користувач закінчив етап згадування йому показуються результати тренування. Саме результати тренування зберігаються у базі даних і враховуються в особистій і загальній статистиці.

Типи тренувань

1. Числа – мета користувача запам’ятати задану послідовність цифр. У рамках курсової роботу користувачу будуть показуватись по 2 цифри. Таким чином набір можливих даних для запам’ятовування складається с чисел від «00» до «99».
2. Слова – мета користувача запам’ятати задану послідовність слів. Користувачу слова показуються по одному.

Набори даних для тренування у систему вводить адміністратор. У рамках курсової роботи фізично адміністратор реалізований не буде.

Результати тренувань

Результати тренування кожного типу зберігаються в базі даних у відповідних таблицях (для кожного типу тренування окрема таблиця). За результатами будується особиста статистика користувача і загальна статистика. Якщо користувач зробив більше 10% помилок – результат оцінюється в 0 балів.

Всі результати тренувань характеризуються такими атрибутами:

* Id користувача
* Дата тренування
* Час запам’ятовування
* Час згадування
* Оцінка

Результат тренування оцінюється за формулою:

Де correct - кількість правильних відповідей,

total - загальна кількість даних,

time - час запам'ятовування в секундах.

Якщо користувач зробив більше 10% помилок, результат оцінюється у 0 балів.

Статистика

Існує два види статистики – статистика користувача та загальна статистика.

У статистиці користувача відображається список усіх його тренувань за певний період та по кожному виду тренувань.

У загальній статистиці відображаються найкращі результати користувачів по кожному виду тренування.

* 1. **Вибір засобів розробки**

Для розробки програмного забезпечення у цій роботі використано мову програмування Java. Для розробки графічного інтерфейсу користувача використано бібліотеку JavaFX з використанням FXML та CSS (SASS) а також Java Persistence API framework EclipseLink та СУБД MYSQL.

1. **Проектування**
   1. **Ескізний проект** 
      1. **Схема взаємодії модулів**

Взаємодію модулів системи можна представити в вигляді наступної діаграми:



2. 1. 2. **Діаграма варіантів використання**

За допомогою діаграми варіантів використання окреслимо користувачів системи та її функціонал:



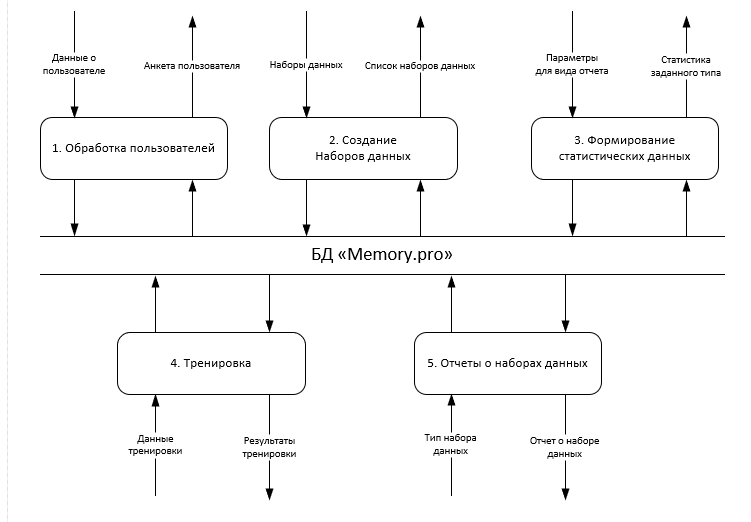
* 1. **Технічний проект**

В результаті опису предметної області та постановки задачі отримано вимоги до БД системи, які можна зобразити такими діаграмами як: DFD, ERD, логічна модель системи.

* + 1. **Контекстна діаграми системи**



* + 1. **DFD 1 рівня**



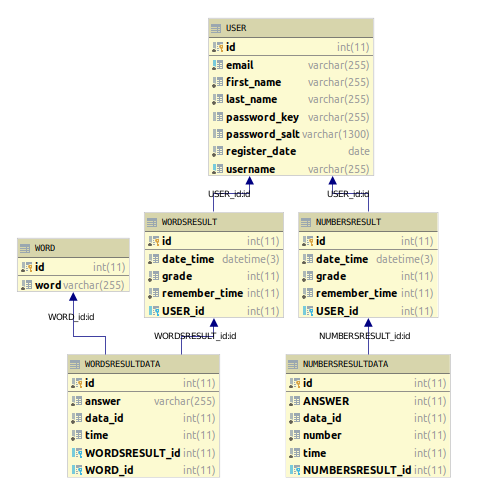
* + 1. **DFD 2 рівня (Статистика)**



* + 1. **Діаграма сутність-зв’язок (ERD)**



* + 1. **Логічна модель системи**



Таблиця User – користувач системи. Усі атрибути зрозумілі з назви. Атрибути password\_key та password\_salt необхідні для авторизації користувача (сам пароль ніде не зберігається, його знає лише користувач). Тобто зберігання паролю користувача основане на механізмі хешування.

Таблиці WordsResult та NumbersResult призначенні для зберігання результатів тренування.

date\_time – час початку тренування

grade – оцінка тренування

remember\_time – час згадування в секундах

Таблиці WordsResultData NumbersResultData зберігають дані тренування. Відповідно слова, та числа. А також час на запам’ятовування кожної одиниці даних у мілісекундах та відповідь користувача (у WordsResultData зберігається як текст, так як, користувач може ввести слово, якого немає в базі даних).

* + 1. **Діаграма класів**

На діаграмі класів зображено взаємодію модулів системи.

У системі усі класи поділяються на такі види:

1. Component – певна візуально та логічно самостійна сутність. Усі компоненті складаються с контролеру (java класу, FXML, та SASS для стилів). Компоненти реалізують лише візуальну складову. Вони не повинні реалізовувати занадто багато логіки, не пов’язаної напряму з відображенням. Назви усіх класів цього виду закінчуються на Component
2. Screen – наслідує від Component. Представляє сутність вікна (сторінки). Screen класи можуть містити у собі компоненти. Один вид компоненту може міститись в різних Screen класах. Назви усіх класів цього виду закінчуються на Screen.
3. Service – класи, що реалізують логічно самостійну сутність. Дані класи реалізують певну задачу (наприклад авторизацію) та надають зручний інтерфейс для роботи з ними. Вони нічого не знають про те, як результати їх роботи будуть представлені користувачу. Дані класи працюють тільки з даними. Назви усіх класів цього виду закінчуються на Service.
4. Entity – описання сутностей для Java Persictence API
5. Dao – класи, що узагальнюють роботу з Entity класами та базою даних в цілому. Тільки в Dao-класах дозволяється працювати з базою даних. Назви усіх класів цього виду закінчуються на Dao.

Кожен вид класів при необхідності може використовувати додаткові класи

Також є два допоміжних класів:

* 1. Common – клас, що дозволяє Screen- та Component-класам спілкуватись між собою, та передавати повідомлення до Bootstrap.
  2. Alerts – надає зручний інтерфейс для виведення повідомлень користувачам (повідомлення про помилки, попередження, інформаційні повідомлення тощо). Даний клас міститься (композиція) у Component і таким чином доступний з усіх Component та Screen класів.

Component класи:

1. HeaderComponent – header, що відображається на усіх сторінках
2. LoginComponent – авторизація користувачів
3. SignupComponent – реєстрація нових користувачів
4. TrainingSetupComponent – вибір параметрів тренування
5. TrainingMemorizeComponent – етап запам’ятовування
6. TrainingRememberComponent – етап згадування
7. ResultComponent – відображає результат тренування за переданими типом тренування на id результату.

Screen класи:

1. Bootstrap – точка входу, головний контролер
2. AuthScreen – сторінка реєстрації та авторизації
3. IndexScreen – головна сторінка
4. ProfileScreen – профіль користувача
5. StatisticScreen – сторінка статистики
6. TrainingScreen – сторінка тренування

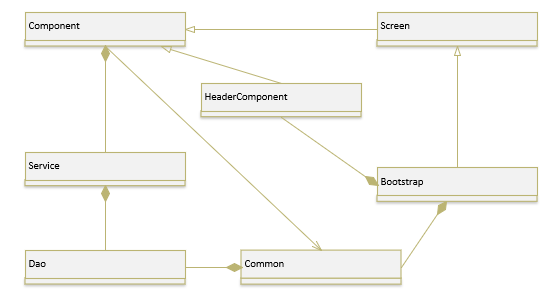
Service класи

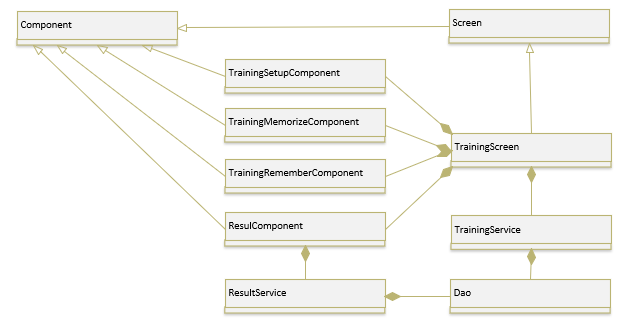
1. AuthService – сервіс авторизації на реєстрації
2. TrainingService – сервіс тренувань (надає дані для тренування, обробляє та зберігає результати.
3. ResultService – сервіс для роботи з результатами тренування (за переданим типом тренування та id результата надає оброблений і готовий для подальшого використання результат тренування (в поверненому результаті вже містяться всі дані такі як – час запам’ятовування, мінімальний та максимальний час на запам’ятовування одного елементу тощо).

Entity та Dao класи (відповідають сутностям на ER діаграмі

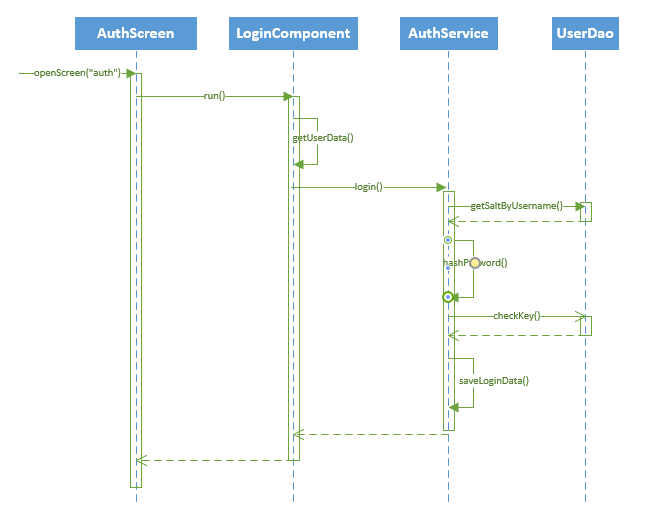
1. User та UserDao
2. NumbersResult та NumbersResultDao
3. NumbersResultData та NumbersResultDataDao
4. WordsResult та WordsResult
5. WordsResultData та WordsResultDataDao
6. Word та WordDao

Так, як сервіс складається з великої кількості класів і при цьому все вони діляться всього на кілька типів, то на діаграмі класів немає сенсу зображати всі класи. Тому буде представлена загальна діаграма класів та діаграма класів відносно TrainingScreen.





* + 1. **Діаграма послідовностей** 
       1. **Діаграма послідовностей процесу «авторизація»**



* + - 1. **Діаграма послідовностей процесу «тренування»**



# **Додаток А. Технічне завдання**

1. Загальні відомості
   1. Найменування системи
      1. Повна назва системи

Сервіс для тренування пам’яті «Memory.pro».

* + 1. Коротка назва

Memory.pro

* 1. Підстави для розробки

Підставою для розробки програмного забезпечення (ПО) є договір між Беркунським Євгенієм Юрійовичем (далі Замовник) та Гашко Дмитро Андрійовичем (далі Виконавець) від 20.05.2019.

* 1. Планові строки початку та закінчення робіт

Початком роботи над програмним забезпеченням слід вважати 21.05.2019, закінчення роботи є 10.06.2019.

* 1. Порядок оформлення та пред’явлення замовнику результатів робіт

Роботи зі створення ПЗ здаються Замовнику в дату закінчення роботи. Виконавець надає замовнику звітні документи, склад яких зазначено у договорі та демонструє результати робіт.

1. Призначення та цілі створення системи
   1. Призначення системи

Сервіс «Memory.pro» призначений для розвитку та тренування пам’яті, аналізу свого прогресу, та прогресу інших учасників.

У рамках проекту будуть реалізовані такі типи тренувань:

1. Запам’ятовування послідовності двозначних чисел
2. Запам’ятовування послідовності слів.
   1. Цілі створення системи

Система створюється з наступними цілями:

1. Надати користувачу інструменти для тренування пам’яті
2. Реалізувати основні види тренувань (числа та слова).
3. Реєстрація/авторизація користувачів.
4. Кабінет користувача
5. Збереження результатів тренувань користувачів
6. Перегляд статистичних даних
7. Короткий опис сервісу

Сервіс «Memory.pro» призначений для тренування та розвитку пам’яті. Основна функція сервісу – тренування з запам’ятовування даних різних типів, таких як цифри, слова, картинки, імена, дати, тощо. При виконанні курсової роботи будуть реалізовані два – цифри та слова.

Під час тренування користувачу необхідно запам’ятати максимальну кількість даних вибраного типу за мінімальний час, після чого відновити їх у тому ж порядку.

Після запам’ятовування формується результат тренування – що користувач запам’ятав правильно, що не правильно, скільки часу він запам’ятовував запропоновану інформацію тощо (детальніше у розділі «результати тренування»

На основі результатів користувачів формується статистика – статистика конкретного користувача та загальна статистика.

1. Вимоги до системи
   1. Вимоги до системи в цілому
      1. Вимоги до структури та функціонування системи

Сервіс повинен мати систему авторизації, де користувач може увійти у систему, або зареєструватись; особистий кабінет користувача; систему тренувань з запам’ятовування чисел та слів. Усі дані користувачів та результатів тренувань повинні зберігатись у базі даних.

* + 1. Вимоги до надійності
       1. Склад показників надійності для системи в цілому

Рівень надійності повинен досягатись узгодженим застосуванням організаційних, організаційно-технічних заходів та програмно-апаратних засобів. Надійність повинна забезпечуватись за рахунок:

1. Застосування технічних засобів, програмного забезпечення, що відповідають класу вирішуваних задач.
2. Своєчасне виконання процесів адміністрування та технічного обслуговування програмно-апаратних засобів.
   * + 1. Перелік аварійних ситуацій, за яких регламентуються вимоги до надійності

Під аварійною ситуацією слід розуміти аварійне завершення роботи складових сервісу: безпосередньо програми, бази даних тощо.

При роботі системи можливі наступні аварійні ситуації, що впливають на надійність роботи системи:

1. Збій у роботі електропостачання серверу БД
2. Перевантаження серверу БД
3. Помилки роботи ПЗ, що не були виявлені про тестуванні та випробуванні системи;
   * 1. Вимоги до ергономіки до технічної естетики

Інтерфейс системи має бути звучним та інтуїтивно зрозумілим. Усі сторінки та компоненти повинні бути виконанні в єдиному стилі. Палітра основних кольорів має складатись не більше ніж з 4 кольорів.

Сервіс обов’язково має бути доступний з клавіатури.

Сервіс має бути на англійській мові.

Усі помилки та виняткові ситуації мають супроводжуватись відповідними повідомленнями.

* + - 1. Вимоги до захисту інформації від несанкціонованого доступу

Забезпечення інформаційне безпеки сервісу має відповідати таким вимогам:

* Для доступу до сервісу користувачу необхідно увійти до системи.
* Якщо користувач ще не зареєстрований у системі, він повинен мати можливість зареєструватись
* Пароль, введений користувачем не зберігається, і навіть не відправляється на сервер бази даних. Авторизація реалізується на основі механізму хешування
* В базі даних має зберігатись ключ та «сіль» (що генерується для кожного користувача) паролю
* Неавторизовані користувачі не повинні мати доступ до сервісу
  1. Вимоги до видів забезпечення
     + 1. Вимоги щодо застосування систем управління баз даних (СУБД)

Всі дані сервісу (дані користувачів, результати тренувань) повинні зберігатись у базі даних на основі СУБД MYSQL.

* + - 1. Вимоги до структури процесу збору, обробки, передачі даних у системі та представленню даних

Процес збору, обробки і передачі даних в системі визначається регламентом процесів збору, перетворення і завантаження даних, що розробляється в етапі «Проектування. Розробка ескізного проекту. Розробка технічного проекту ».

* + - 1. Вимоги до захисту даних від руйнування при аваріях та сбоях електропостачання системи

Необхідно регулярно робити backup даних сервісу

Розробка програмного забезпечення сервісу повинна супроводжуватись веденням GIT-репозиторію, який також має розміщуватись на GitLab.

* + 1. Вимоги до ПЗ

До забезпечення якості ПЗ ставляться такі вимоги:

Сервіс має працювати стабільно, не закриватися при виникненні помилок без попередження користувача.

Сервіс має мати достатню швидкість роботи, щоб користувачі могли вільно ним користуватись

* + 1. Вимоги до організаційного забезпечення

Основними користувачами системи Користувач. До захисту від помилкових дій користувача ставляться такі вимоги:

1. Усі дії, що у разі випадкового виконання можуть завдати шкоди користувачу мають супроводжуватись підтвердженням.
2. для зниження помилкових дій користувачів має бути розроблена інструкція користувача.
   * 1. Вимоги до методичного забезпечення

Порядок роботи з системою викладено у окремому документі – «Інструкції користувача».

1. Склад робіт по створенню системи

Роботи зі створення системи виконуються в три етапи:

* Проектування. Розробка ескізного проекту. Розробка технічного проекту
* Розробка робочої документації.
* Введення в дію.

1. Порядок контролю та прийому системи
   1. Об’єм та види випробувань системи

Система піддається випробуванням наступних видів:

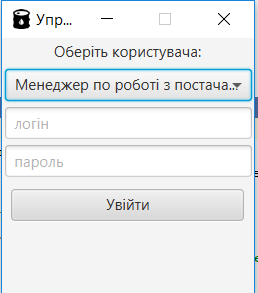
1. Попередні випробування.
2. Дослідна експлуатація.
3. Приймальні випробування.
4. Вимоги до документування

|  |  |
| --- | --- |
| Проектування. Розробка ескізного проекта. Розробка технічного проекта. | Відомість ескізного проекта |
| Пояснювальна записка ескізного проекта |
| Відомість ескізного проекта |
| Пояснювальна записка до технічного проекта |
| Схема функціональної структури |
| Відомість експлуатаціоних документів |
| Паспорт |
| Загальне призначення системи |
| Технологічна інструкція |
| Інструкція користувача |
| Розробка технічної документації. Адаптація програм | Опис технологічного процесу обробки даних |
| Інструкція по формуванню та введенню бази даних |
| Склад вхідних даних |
| Каталог бази даних |
| Програма та методика випробувань |
| Специфікація |
|  | Опис програм |
| Текст програм |
| Введення в дію | Акт прийому в дослідну експлуатацію |
| Протокол випробувань |
| Акт прийому системи в експлуатацію |
| Акт закінчення робіт |
|  |

# **Додаток Б. Інструкція користувача**

Дана інструкція призничена для ознайомлення з функціями ПЗ та полегшення роботи з ним.

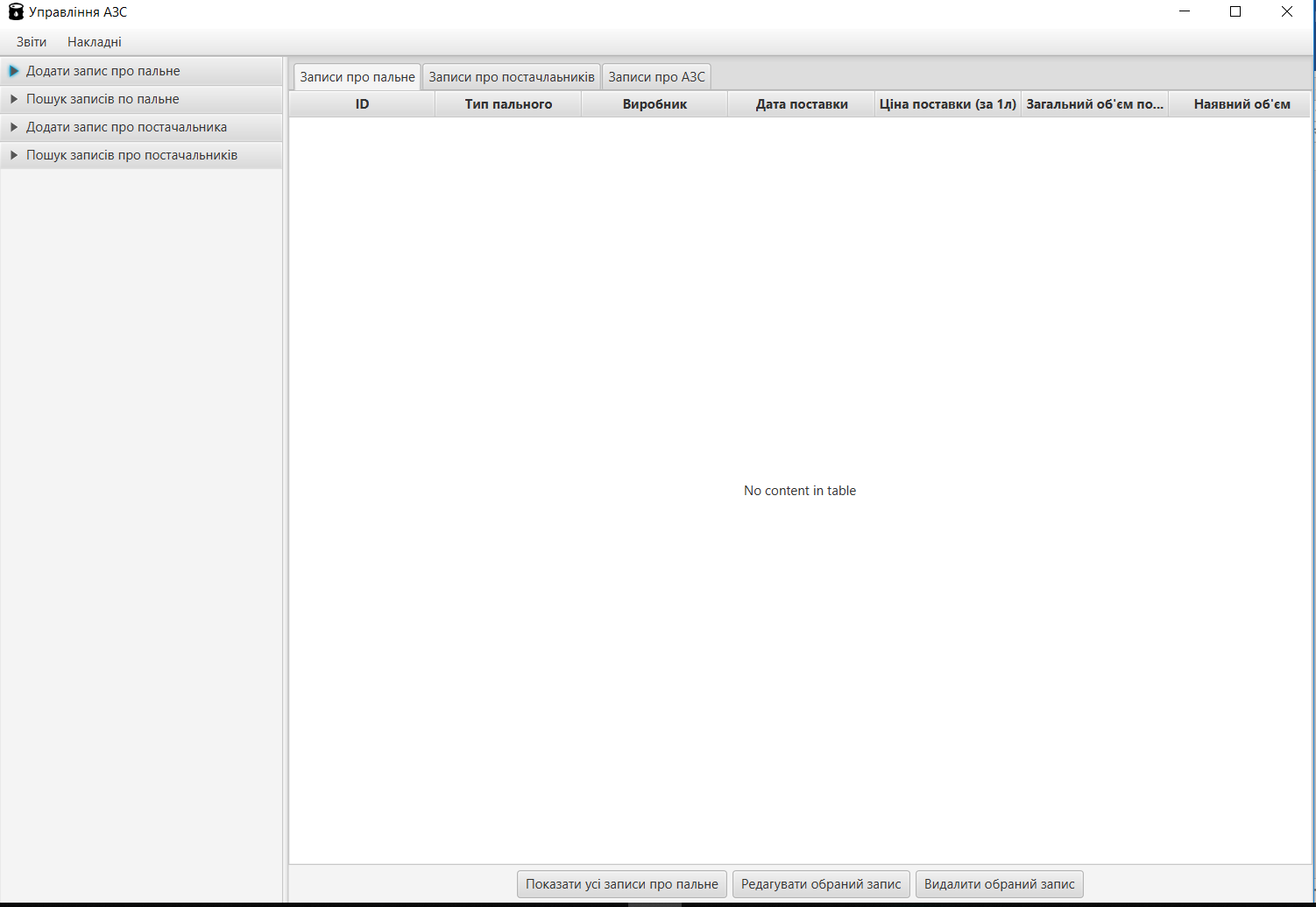
1. Для запуску програми потрібно натиснути на якрлик з написом «Управління АЗС» або відкрити його будь-яким іншим чином.
2. Після запуску з’явиться вікно в якому потрібно буде вибрати користувача (менеджера по роботі з постачальниками або оператора АЗС) та ввести свії логін та пароль. У випадку введення правильного логіну та паролю буде відкрите відповідне АРМ, інакше користувач побачить повідомлення про помилку.



1. Детальна інструкція для менеджера по робота з постачальниками

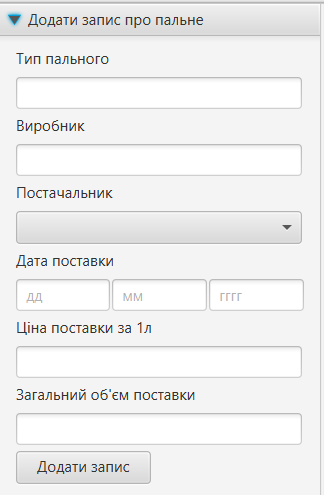
Після відкриття АРМ будуть доступні наступні функції:

* Додати запис про пальне;
* Знайти запис пального;
* Додати запис про постачальника;
* Знайти запис про постачальника;
* Переглянути усі записи про пальне;
* Видалити обраний запис (про пальне);
* Редагувати обраний запис (про пальне);
* Переглянути усі записи про постачальників;
* Видалити обраний запис (про постачальника);
* Редагувати обраний запис (про постачальника);
* Переглянути усі записи про АЗС;
* Видалити обраний запи (про АЗС);
* Редагувати обраний запис про АЗС;
* Створити звіт про постачальників;
* Створити звіт про пальне;
* Створити звіт про оберт (переміщення) пального;
* Створити видаткову накладну.



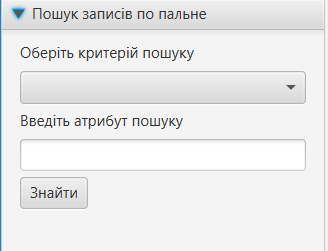
* 1. Додати запис про пальне

Для того, аби додати запис про пальне необхідно натиснути на напис «Додати запис про пальне». Після цього з’являться поля для введення даних про тип пального, виробника пального, постачальника (яко потрібно вибрати зі списку), дату поставки, ціну за 1л, загальний об’єм поставки. Після того, як усі поля будут заповнені необхідно натиснути на кнопку «Додати запис». У випадку заповнення усіх полів правильними даними запис про пальне буде додано до БД системи, інакше буду виведено повідомлення про помилку з поясненям помилки.



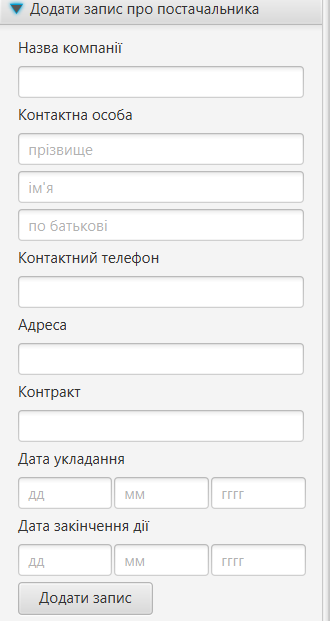
* 1. Знайти запис пального

Для того, аби додати знайти про пальне необхідно натиснути на напис «Пошук записів про пальне». Після цього з’являться випадаюче меню для вибору критерія пошуку поля для введення даних для введеня цього критерію. Після того, як буде обрано критерій пошуку та введено атрибут пошуку у поле необхідно натиснути на кнопку «Знайти». У випадку вибору критерія пошуку з меню та заповнення поля відповідними даними запис про зайндено (або виведено повідомлення, що такого запису немає у БД), інакше буду виведено повідомлення про помилку з поясненям помилки.



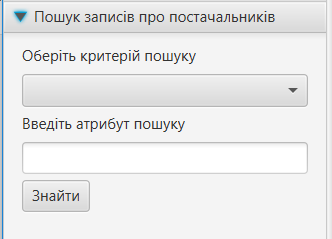
* 1. Додати запис про постачальника

Для того, аби додати запис про пальне необхідно натиснути на напис «Додати запис про постачальника». Після цього з’являться поля для введення даних про назву компанії, ПІБ контактної особи, контактний телефон, адресу, контракт, дата укладання та дату закінчення дії контракту. Після того, як усі поля будут заповнені необхідно натиснути на кнопку «Додати запис». У випадку заповнення усіх полів правильними даними запис про пальне буде додано до БД системи, інакше буду виведено повідомлення про помилку з поясненям помилки.



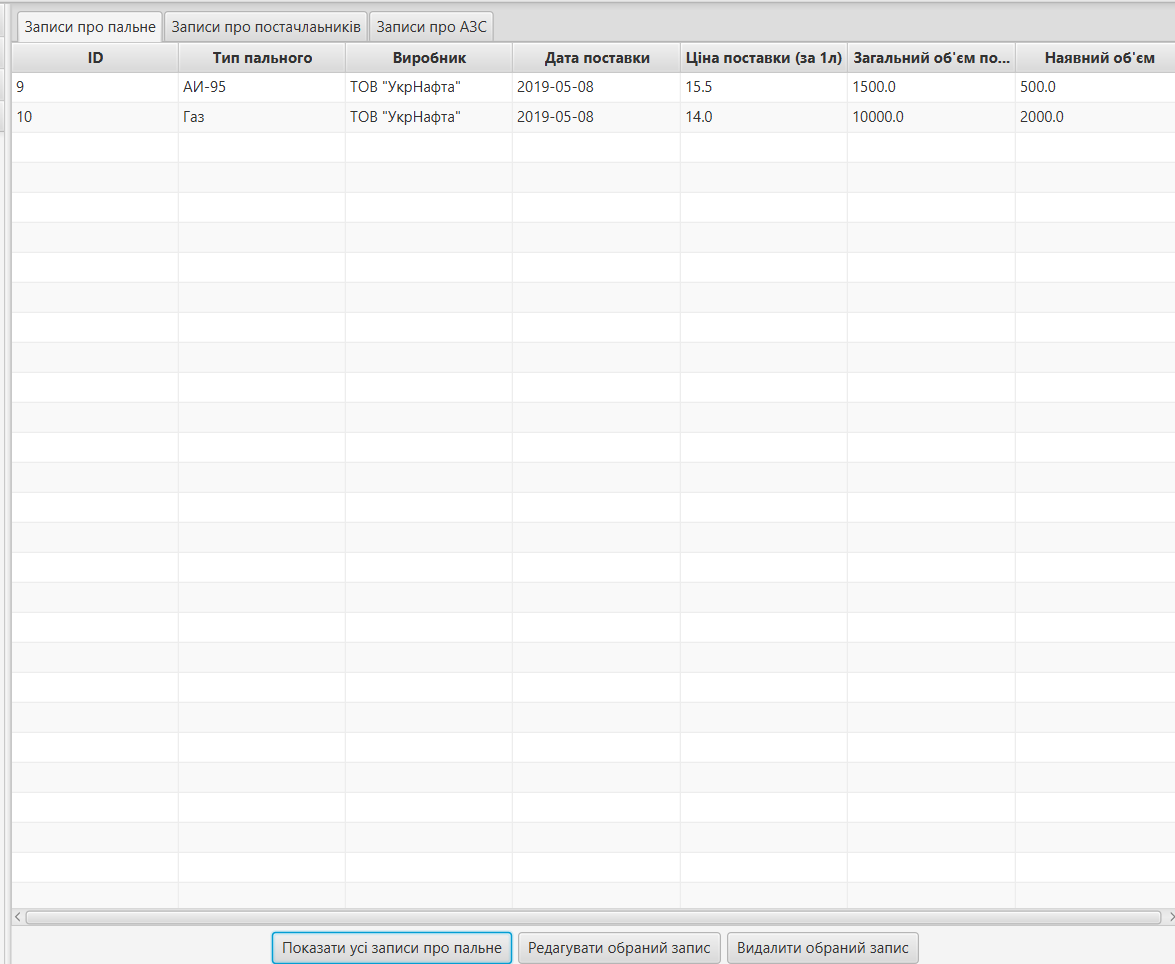
* 1. Пошук записів про постачальника

Для того, аби додати знайти про пальне необхідно натиснути на напис «Пошук записів про постачальників». Після цього з’являться випадаюче меню для вибору критерія пошуку поля для введення даних для введеня цього критерію. Після того, як буде обрано критерій пошуку та введено атрибут пошуку у поле необхідно натиснути на кнопку «Знайти». У випадку вибору критерія пошуку з меню та заповнення поля відповідними даними запис про знайдено(або виведено повідомлення, що такого запису немає у БД), інакше буду виведено повідомлення про помилку з поясненям помилки.



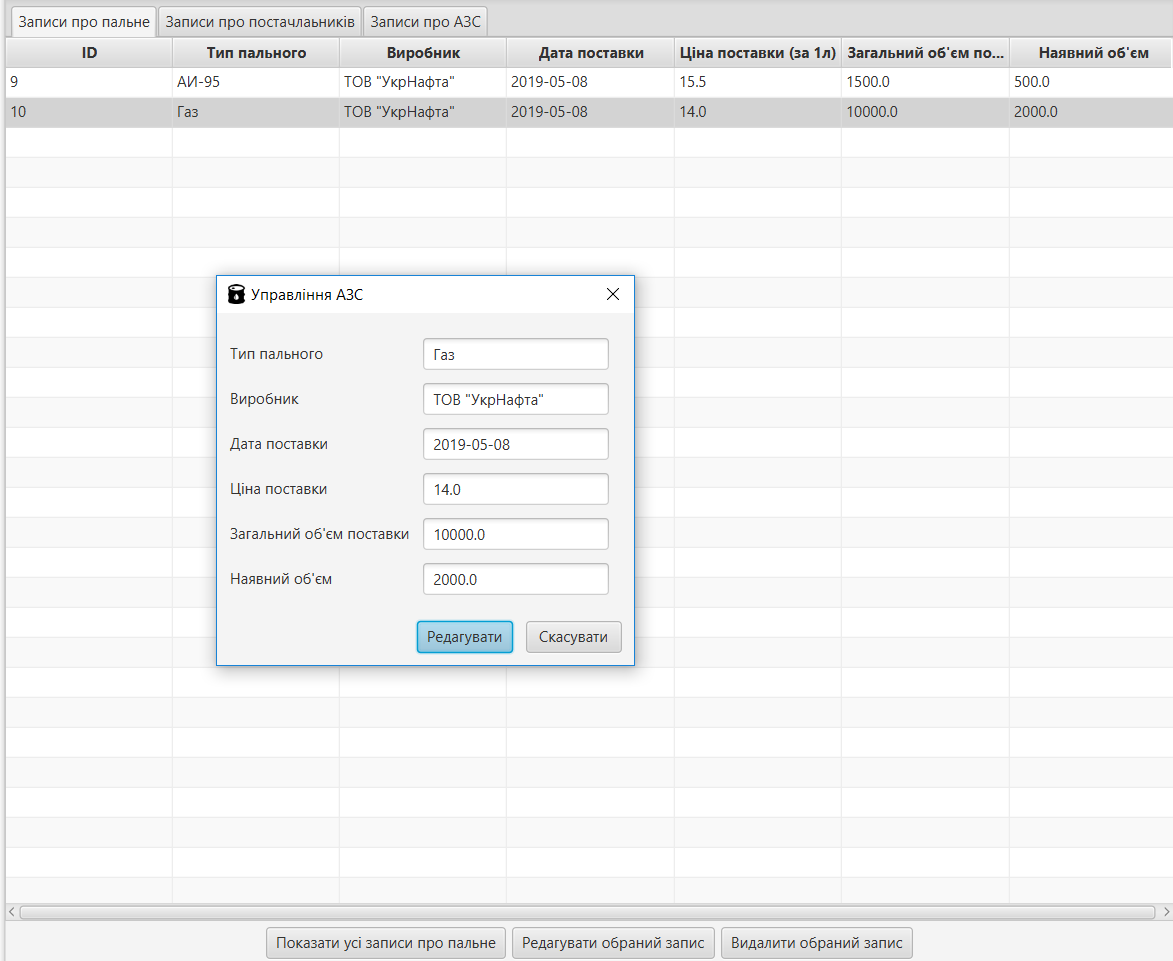
* 1. Переглянути усі записи про пальне

Для того, аби переглянути усі записи про пальне необхідно натиснути на кнопку «Показати усі записи про пальне». У випадку наявності записів у таблицю буде відображено всі записи про пальне, інашке будет виведено повідомлення, що записів у БД не має.



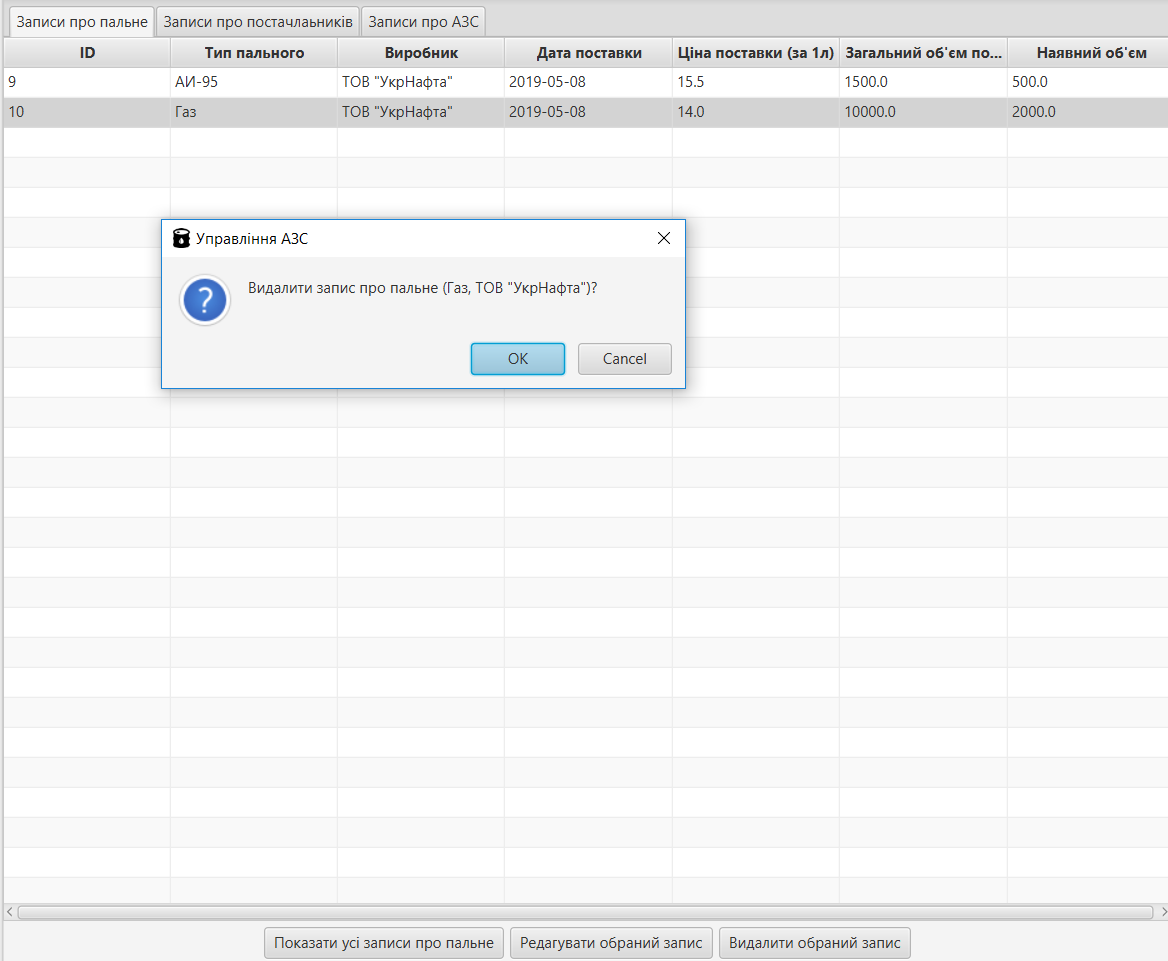
* 1. Редагувати запис про пальне

Для того, аби редагувати запис про пальне треба спочатку знати запис(и), або переглянути усі записи про пальне. Після цього необхідно вибрати (натиснути на запис у таблиці) та натиснути на кнопку «Редагувати обраний запис». Після цього відкриється вікно, у якому будуть доступні для редагування типа пального, виробник, дата поставки, ціна поставки, загальний об’єм пального, наявний об’єм пального. Після того, як буде зкореговано усі необхідні поля потрібно натиснути на кнопку «Редагувати». У випадку заповнення полів даними правильного формату запис про пальне будет зкореговано і це відобразиться у таблиці «Записи про пальне», інашке ви отримаете повідомлення про помилку та вікно редагування закріється. При натисканні кнопки «Скасувати» операція редагування запису буде перевана й жодні зміни до запису внесено не буде.



* 1. Видалити запис про пальне

Для того, аби видалити запис про пальне треба спочатку знати запис(и), або переглянути усі записи про пальне. Після цього необхідно вибрати (натиснути на запис у таблиці) та натиснути на кнопку «Видалити обраний запис». Після цього відкриється вікно, яке попросить підтвердити запит на видалення запису. У випадку натискання кнопки «OK» обраний запис буде видалено, у випадку натискання кнопки «Cancel» запис не буде видалено.

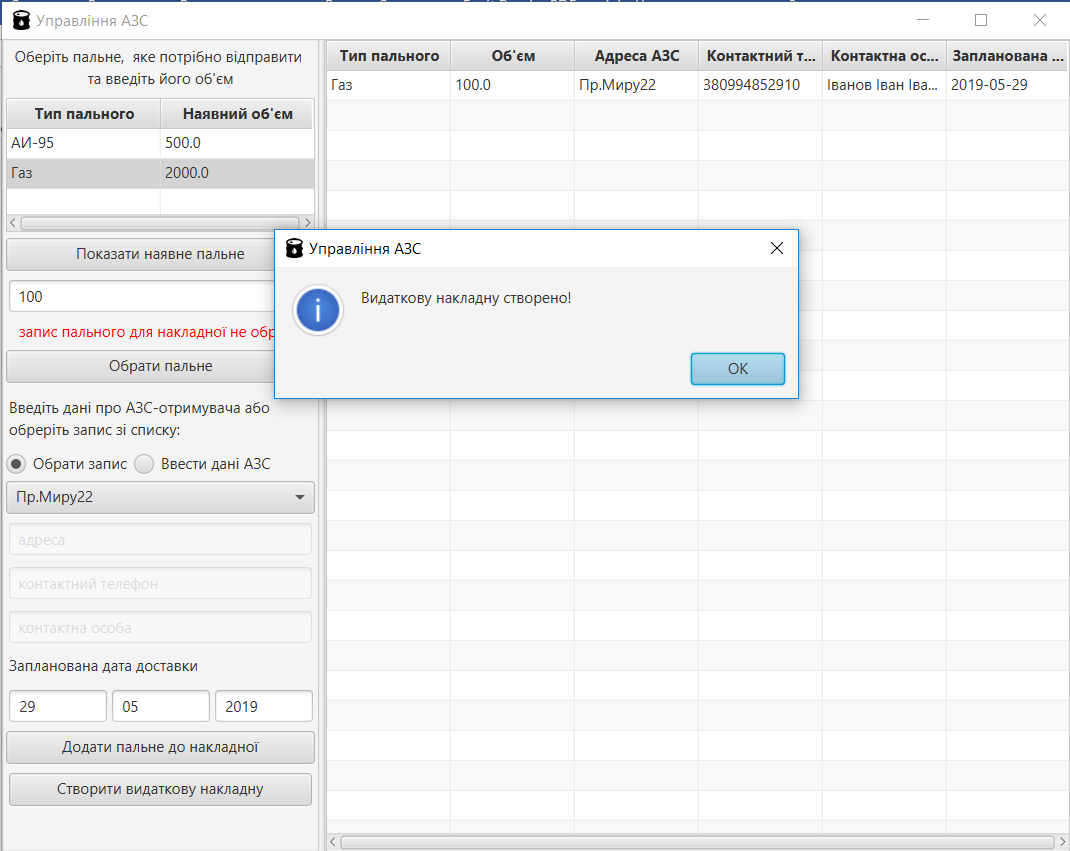


* 1. Створити видаткову накладну

Для того, аби створити видаткову накладну необхідно у верхній панелі натиснути на пункт «Накладні» та вибрати зі списку пункт «Створити видаткову накладну». Після цього відкриється ще одне вікно, у якому будет проходити створення видаткової накладної. Для того, аби створити видаткову накладну спочатку необхідно вибрати одне пальне з доступних. Для цього необхідно натиснути на кнопку «Показати наявне пальне». Ви побачите список усіх типів пального, що є, так їх об’єм. Після цього необхідно ввести в поле для введення об’єм той об’єм, який необхідно відправити та натиснути кнопку «Обрати пальне». У тому випадку, якщо буде обрано об’єм більший, ніж наявний об’єм пального, буде виведено повідомлення про помилку.

Після того, як пальне буде обрано, необхідно обрати АЗС, куда потрібно відправити пальне. Це можна зробити двома шляхами: обрати АЗС з тих, що вже є у БД, або ввести дані про АЗС (адресу, контактний телефон, контактну особу). Обов’язковим є введення запланованої дати доставки пального. Після того, як буде обрано АЗС тим, чи іншим шляхом та введено заплановану дату поставки можна створити запис у видатковії накладній натиснувши кнопку «Додати пальне до накладної». Якщо було введено дані неправильного формату, буде виведено повідомлення про помилку.

Крок з додаванням пального до накладної можна повторювати необхідну кількість разів. Після того, як усе необхідне пальне буде додано до накладної, тобто вона буде сформована, для її створення необхідно натиснути на кнопку «Створити видтакову накладку». У випадку, якщо у накадній не меньше 1 запису, вона будет створена.



# **Додаток В. Код програмних модулів**