МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»

ІМЕНІ ІГОРЯ СІКОРСЬКОГО

ІНСТИТУТ СПЕЦІАЛЬНОГО ЗВ`ЯЗКУ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ЗВІТ

з дисципліни «Технологія Розробки Програмного Забезречення»

Автоматична Система Управління Медпункту

Виконав: курсант С - 94 групи

Кілочицький І.В.

Перевірив: доцет кафедри № 5

Соколов Володимир Володимирович

Київ 2021 р.

**Автоматизована система управління «Медпункт»**

**Програма та методика приймальних випробувань**

**1562.001.ПМ**

|  |  |
| --- | --- |
|  | "УГОДЖЕНО"  Комерційний директор  ВАТ «Medical CK5»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соколов В. В.  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_ 2021 р. |

|  |  |
| --- | --- |
| "УГОДЖЕНО"  І.О. начальника відділу  впровадження  ВАТ «Medical CK5»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соколов В. В.  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_ 2021 р. | "УГОДЖЕНО"  Начальник відділу  по роботі з  ВАТ «ЕК «Medical CK5»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соколов В. В.  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_ 2021 р. |

Київ 2021

**ЗМІСТ**

[Об'єкт випробувань 5](#_Toc92186259)

[1.1 Повне найменування системи 5](#_Toc92186260)

[1.2 Комплектність випробувальної системи 5](#_Toc92186261)

[Мета випробувань 5](#_Toc92186262)

[Загальні положення 5](#_Toc92186263)

[3.1 Перелік керівних документів для проведення випробувань 5](#_Toc92186264)

[3.2 Місце та тривалість випробувань 5](#_Toc92186265)

[3.3 Організації, що беруть участь у випробуваннях 5](#_Toc92186266)

[3.4 Перелік документів, що пред'являються на випробування 5](#_Toc92186267)

[3.5 Попередні перевірки 5](#_Toc92186268)

[3.6 Перевірка функціонування системи у режимі централізованої обробки даних 6](#_Toc92186269)

[3.7 Перевірка виконання вимог функціонального призначення системи 6](#_Toc92186270)

[3.8 Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5» 6](#_Toc92186271)

[Умови, порядок проведення та режими випробувань 7](#_Toc92186272)

[4.1 Умови проведення випробувань 7](#_Toc92186273)

[4.1.1 Попередні перевірки 7](#_Toc92186274)

[4.1.2 Перевірка функціонування системи у режимі централізованої обробки даних 7](#_Toc92186275)

[4.1.3 Перевірка виконання вимог функціонального призначення системи 8](#_Toc92186276)

[4.1.4 Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5» 8](#_Toc92186277)

[4.2 Порядок реєстрації результатів випробувань та виявлених недоліків 8](#_Toc92186278)

[4.3 Умови початку та завершення окремих етапів випробувань 9](#_Toc92186279)

[4.3.1 Перевірка функціонування системи у режимі централізованої обробки даних 9](#_Toc92186280)

[4.3.2 Перевірка виконання вимог функціонального призначення системи 9](#_Toc92186281)

[4.3.3 Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5» 9](#_Toc92186282)

[4.4 Обмеження за умов проведення випробувань 9](#_Toc92186283)

[4.5 Заходи, що забезпечують безпеку та безаварійність проведення випробувань 10](#_Toc92186284)

[4.6 Порядок взаємодії організацій, що беруть участь у випробуваннях 10](#_Toc92186285)

[4.7 Вимоги до персоналу, що проводить випробування, та порядок його допуску до випробувань 10](#_Toc92186286)

[4.8 Перелік робіт, що проводяться після завершення випробувань, вимоги до них, обсяг та порядок проведення 11](#_Toc92186287)

[Матеріально-технічне забезпечення випробувань 11](#_Toc92186288)

[Метрологічне забезпечення випробувань 11](#_Toc92186289)

[Звітність 11](#_Toc92186290)

[Попередні умови для перевірки виконання вимог функціонального призначення системи 18](#_Toc92186291)

[8.1 Перевірка заповнення бази даних 18](#_Toc92186292)

[Опис методик тестування 18](#_Toc92186293)

[9.1 Перевірка функціонування системи у режимі централізованої обробки даних 18](#_Toc92186294)

[9.2 Перевірка відповідності функціональності системи вимогам ТЗ 18](#_Toc92186295)

[Додаток Б. Тестові приклади 19](#_Toc92186296)

# Об'єкт випробувань

## Повне найменування системи

* + 1. Автоматизована система управління Медпункт
    2. Позначення - "Медпункт", система.

## Комплектність випробувальної системи

* + 1. На випробування виставляється Система версії 2.0 у складі:
* серверна частина системи;
* клієнтська частина системи;
* копію бази даних Замовника;
* експлуатаційна документація.

# Мета випробувань

2.1. Метою проведення приймальних випробувань Система є:

* перевірка функціонування системи у режимі централізованої обробки даних;
* перевірка виконання вимог функціонального призначення системи;
* формування переліку додаткових та змінюваних функцій системи, що підлягають реалізації для задоволення першочергових вимог ВАТ ЕК «Medical CK5».

# Загальні положення

## Перелік керівних документів для проведення випробувань

* + 1. Приймальні випробування Система проводяться на підставі Наказу Голови правління – генерального директора ВАТ ЕК «Medical CK5» щодо проведення приймальних випробувань від «\_\_» квітня 2021 р. № \_\_\_\_\_.

## Місце та тривалість випробувань

* + 1. Місце випробувань: м. Київ, центральний Магазин ВАТ ЕК " Medical CK5 ",
    2. Час випробувань: 20-23 грудня 2021р.

## Організації, що беруть участь у випробуваннях

* + 1. У випробуваннях беруть участь такі організації:
* ВАТ ЕК «Medical CK5» – замовник, організація, що експлуатує;
* ТОВ «S&S» – виконавець, організація що супроводжує.

## Перелік документів, що пред'являються на випробування

* + 1. На випробування пред'являється така документація:
* загальний опис системи;
* керівництво адміністратора системи;
* посібник користувача системи;
* технологічна інструкція;
* програма та методика приймальних випробувань системи (даний документ).
  + 1. Документація постачається:
* в електронному вигляді (на оптичному носії);Обсяг випробувань

## Попередні перевірки

* + 1. Попередні перевірки охоплюють різні аспекти системи та її застосування та включають оцінки наступних показників:
* комплектність та якість експлуатаційної документації на систему;
* час та обсяг пам'яті, необхідні для виконання резервного копіювання-відновлення даних системи;
* адекватність розподілу ролей користувачів системи штатної структури та кваліфікації обслуговуючого персоналу системи.
  + 1. Перелік конкретних перевірок (розв'язуваних задач) для оцінки заданих показників з посиланнями на відповідні методики (розділи методик) випробувань наведено у таблиці 1.

Таблиця 1.

|  |  |  |
| --- | --- | --- |
| № групи перевірок | Назва групи перевірки | Методика перевірки |
| 1 | Перевірка комплектності та якості експлуатаційної документації | Виконується шляхом візуальної перевірки |
| 2 | Визначення часу та обсягу пам'яті резервного копіювання-відновлення даних | Виконується адміністратором системи на етапі підготовки до випробувань |
| 3 | Оцінка адекватності ролей користувачів | Виконується адміністратором системи під час встановлення системи |

## Перевірка функціонування системи у режимі централізованої обробки даних

* + 1. Перевірка функціонування системи у режимі централізованої обробки даних включає оцінку наступних показників функціонування системи:
* час реакції системи (продуктивність роботи оператора) проти децентралізованим режимом під час введення даних у систему, наприклад: купівля товару, формування кошику, додавання даних до бази даних.
* швидкість виконання масових розрахункових операцій (розрахунок загальної вартості за вибраним критерієм).
  + 1. Перелік конкретних перевірок (розв'язуваних задач) для оцінки заданих показників з посиланнями на відповідні методики (розділи методик) випробувань наведено у таблиці 2.

Таблиця 2.

|  |  |  |
| --- | --- | --- |
| № групи перевірок | Назва групи перевірки | Методика виконання |
| 1 | Оцінка часу реакції системи | п.2.1 додатка А |
| 2 | Швидкість виконання масових розрахункових операцій | п.2.1 додатка А |

## Перевірка виконання вимог функціонального призначення системи

* + 1. Перелік конкретних перевірок (розв'язуваних задач) для оцінки заданих показників з посиланнями на відповідні методики (розділи методик) випробувань наведено у таблиці 3.

Таблиця 3.

|  |  |  |  |
| --- | --- | --- | --- |
| № групи перевірок | Назва групи перевірки | № пункту методики (Додаток А) | Примітки |
| 1 | Функція «Авторизація» | 2.2.1 |  |
| 2 | Функція «Введення та виведення даних про документи» | 2.2.2 |  |
| 3 | Функція «Вивід списку створених облікових записів**»** | 2.2.3 |  |
| 4 | Функція «Створення облікової картки» | 2.2.4 |  |
| 5 | Функція «Роботи з обліковою карткою пацієнта» | 2.2.5 |  |
| 6 | Функція «Документи у обліковій картці» | 2.2.6 |  |
| 7 | Функція «Вихід з облікового запису» | 2.2.7 |  |
| 8 | Функція «Зарахування на облік та списання медичних препаратів» | 2.2.8 |  |

## Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5»

* + 1. Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5» включає такі перевірки:
* на відповідність документу "Концептуальний проект системи автоматизованого обліку споживання електричної енергії побутовими абонентами";
* на відповідність технології виконання бізнес-процесів, ухвалених у ВАТ ЕК «Medical CK5».
  + 1. Перелік конкретних перевірок (розв'язуваних задач) з посиланнями на відповідні методики (розділи методик) випробувань наведено у таблиці 4.

Таблиця 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № групи перевірок | № розділу Концепції | Назва групи перевірки | № пункту методики (Додаток А) | Примітки |
| 1 | 1 | Історія життєвого циклу ліків | 2.3 |  |
| 2 | 2 | Облік поставок ліків | 2.3 |  |
| 3 | 3 | Облік видачі ліків | 2.3 |  |
| 4 | 4 | Отримання звіту | 2.3 |  |
| 5 | 5 | Історія хвороби | 2.3 |  |

# Умови, порядок проведення та режими випробувань

## Умови проведення випробувань

### Попередні перевірки

* + - 1. Попередні перевірки виконуються на етапі підготовки системи до випробувань. Деякі показники оцінюються за результатами дослідної експлуатації або під час підготовки та виконання інших перевірок.

### Перевірка функціонування системи у режимі централізованої обробки даних

* + - 1. Перевірка функціонування системи у режимі централізованої обробки даних проводиться на копії реальної БД Медпункт, встановленої на центральний сервер
      2. На підготовчому етапі виконується таке:
* Налаштувати системну автентифікацію на сервері 1 так, щоб віддалені користувачі могли підключатися до системи;
* З використанням системних утиліт необхідно сформувати копію БД Медпункт (на знімний носій, компакт-диск або ноутбук) для перенесення на центральний сервер;
* Перевірити працездатність Система на центральному сервері 1 та в разі потреби встановити та налаштувати систему та системне оточення відповідно до керівництва адміністратора;
* Завантажити отриману копію БД Медпункт на центральний сервер 1 та перевірити коректність завантаження шляхом виконання процедури шляхом формування звітів;
* Перевірити вхід до системи з віддаленого клієнта Медпункт із зазначенням імені користувача та пароля, попередньо виданих адміністратором.
  + - 1. У процесі випробувань штатний персонал Медпункт виконує поточну роботу із введення реальних даних у БД, що дозволить перевірити працездатність централізованого режиму роботи системи при повсякденному штатному навантаженні.
      2. На одному-двох робочих місцях виконуються перевірки, наведені в таблиці 2 з виміром часу виконання операцій, що дозволить перевірити працездатність централізованого режиму роботи системи за підвищеного навантаження.
      3. Після закінчення перевірок провадиться повернення системи в децентралізований режим роботи Медпункт із збереженням (перенесенням у МЕДПУНКТ) змін, зроблених у БД на центральному сервері 1 у процесі введення реальних даних з віддалених клієнтських робочих місць МЕДПУНКТ:
* Після закінчення випробувань необхідно припинити роботу з БД системи;
* З використанням системних утиліт необхідно сформувати копію БД (змін) МЕДПУНКТУ із центрального сервера 1 (на знімний носій, компакт-диск або ноутбук) для перенесення її на локальний сервер МЕДПУНКТ;
* Відновити конфігурацію клієнтської частини всіх робочих станцій МЕДПУНКТУ (у файлі конфігурації змінити адресу сервера) так, щоб вони зверталися до локального сервера додатків МЕДПУНКТУ;
* Заблокувати облікові картки клієнтів МЕДПУНКТУ на сервері 1 так, щоб віддалені користувачі не могли підключатися до системи;
* Завантажити отриману копію БД на локальний сервер МЕДПУНКТУ та перевірити коректність завантаження шляхом виконання процедури перевірки БД Система з локального робочого місця та формування звітів;
* Перевірити вхід до системи з локального клієнта МЕДПУНКТУ.

### Перевірка виконання вимог функціонального призначення системи

* + - 1. Перевірка виконання вимог функціонального призначення системи проводиться в локальному режимі обленерго на копії реальної БД Центрального МЕДПУНКТУ, що встановлюється на центральному сервері 2.
      2. Окремі перевірки можуть виконуватись віддалено з клієнтських робочих місць Центрального МЕДПУНКТУ у централізованому режимі на сервері 2.
      3. На етапі підготовки необхідно:
* Перевірити працездатність Система на центральному сервері 2 та в разі потреби встановити та налаштувати систему та системне оточення відповідно до керівництва адміністратора;
* Завантажити копію БД МЕДПУНКТУ на центральний сервер 2 та перевірити коректність завантаження шляхом виконання процедури перевірки БД Система із клієнтського робочого місця;
* При необхідності виконати підключення 1-2 віддалених робочих місць Центрального МЕДПУНКТУ до центрального сервера 2;
  + - 1. У ході випробувань необхідно на одному-двох робочих місцях виконати перевірки, наведені у таблиці 3.
      2. По закінченні виконання перевірок повторно сформувати перелічені в пункті 5.2.3.3 і порівняти результати з очікуваними.
      3. Наведений режим наскрізних перевірок дозволяє комплексно оцінити правильність роботи розрахункової частини системи.

### Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5»

* + - 1. Перевірка відповідності функціональності системи приватним вимогам ВАТ ЕК «Medical CK5» проводиться на наявність реалізованої функціональності Система шляхом виконання перевірок, наведених у таблиці 4.
      2. Перевірки виконуються тієї ж конфігурації системи, як і перевірка виконання вимог функціонального призначення системи.
      3. За результатами перевірок кожної з них виставляється оцінка за шкалою, наведеної у таблиці 5.

Таблиця 5. Шкала оцінок.

|  |  |
| --- | --- |
| **Оцінка** | **Опис** |
| 5 | Функціональність реалізована, працює згідно специфікації |
| 4 | Функціональність реалізована, працює згідно зі специфікацією, є незначні зауваження |
| 3 | Функціональність реалізована, працює згідно зі специфікацією, є значні зауваження |
| 2 | Функціональність реалізована частково, є значні зауваження, використання важко. |
| 1 | Функціональність реалізована частково, є значні зауваження, використання не можливе |
| 0 | Функціональність не реалізована |

## Порядок реєстрації результатів випробувань та виявлених недоліків

* + 1. За дорученням голови комісії або за погодженням членів комісії для реєстрації результатів випробувань та виявлених недоліків призначаються секретарі з числа членів комісії, які беруть участь у проведенні певної групи перевірок у територіально рознесених місцях у межах підкомісій.
    2. Секретарі проводять робочі протоколи випробувань. По закінченні всіх перевірок виходячи з робочих протоколів формуються загальні протоколи випробувань.
    3. Висновки комісії та узагальнені результати випробувань фіксуються у Акті проведення приймальних випробувань, до якого додаються протоколи випробувань.

## Умови початку та завершення окремих етапів випробувань

### Перевірка функціонування системи у режимі централізованої обробки даних

* + - 1. Переведення Центрального МЕДПУНКТУ у централізований режим обробки даних має проводитися у позаробочий час у проміжок між двома робочими днями.
      2. Перевірки починаються під час виконання наступних умов:
* успішне перенесення БД на центральний сервер 1;
* успішна перевірка входу в систему клієнта МЕДПУНКТУ каналами зв'язку на центральний сервер 1;
* повне завершення підготовчого етапу на початок робочого дня.
  + - 1. Перевірки завершуються (припиняються) під час виконання однієї з наступних умов:
* виконано всі перевірки, передбачені цією програмою;
* закінчився час випробувань;
* відмова технічних засобів, відмова каналів зв'язку та інші форс-мажорні обставини.
  + - 1. Повернення Центрального МЕДПУНКТУ з централізованого режиму обробки даних до децентралізованого має здійснюватися у позаробочий час у проміжок між двома робочими днями.

### Перевірка виконання вимог функціонального призначення системи

* + - 1. Перевірка виконання вимог функціонального призначення системи проводиться незалежно від перевірки функціонування системи в режимі централізованої обробки даних.
      2. Перевірки починаються під час виконання наступних умов:
* успішне перенесення копії БД Центрального МЕДПУНКТУ на центральний сервер 2;
* повне завершення попереднього етапу.
  + - 1. Перевірки завершуються (припиняються) під час виконання однієї з наступних умов:
* виконано всі перевірки, передбачені цією програмою;
* закінчився час випробувань;
* відмова технічних засобів, відмова каналів зв'язку та інші форс-мажорні обставини.
  + - 1. Перенесення змін копії БД Центрального МЕДПУНКТУ, проведених під час випробувань, у реальну БД Центрального МЕДПУНКТУ не провадиться.

### Перевірка відповідності функціональності системи вимогам ВАТ ЕК «Medical CK5»

* + - 1. Перевірка відповідності функціональності системи до приватних вимог ВАТ ЕК «Medical CK5» проводиться після завершення перевірки виконання вимог функціонального призначення системи, т.к. їх результати взаємодоповнювані.

## Обмеження за умов проведення випробувань

* + 1. Перевірка функціонування системи у режимі централізованої обробки даних проводиться на реальних даних за наявності зв'язку між МЕДПУНКТУ та центральним Медпунктом обленерго. Усі зміни, проведені під час перевірок, мають бути перенесені до локальної БД після закінчення випробувань. При неможливості роботи МЕДПУНКТУ у централізованому режимі він має бути переведений у децентралізований режим із відновленням працездатності системи.
    2. Перевірка виконання вимог функціонального призначення системи проводиться у тестовому режимі копії реальної БД МЕДПУНКТУ.
    3. Перевірка відповідності функціональності системи приватним вимогам ВАТ ЕК «Medical CK5» проводиться тестовому режимі на тій самій копії реальної БД МЕДПУНКТУ, що й перевірка виконання вимог функціонального призначення системи. Зміни не повинні переноситися до реальної БД МЕДПУНКТУ
    4. Збої та відмови в роботі активного та пасивного мережевого обладнання, канало-утворюючої апаратури, системного та прикладного ПЗ, серверного та клієнтського апаратного забезпечення, а також позаштатні стани (наявність процесів або явищ, що аномально споживають системні ресурси), які можуть позначитися на результатах тестування документуються у графі зауважень протоколи проведення випробувань, а результати перевірки системи – анулюються. Після усунення збоїв, відмов або причин нештатного стану системи перевірки виконуються повторно.
    5. Збої у роботі технічних засобів та помилки персоналу не враховуються при оцінці результатів випробувань.
    6. Замовник надає фізичний доступ представнику Виконавця на майданчики для проведення випробувань з метою виконання необхідних робіт. Представник Виконавця має право призупинити проведення випробувань у разі фіксації проблем у роботі мережевої та серверної апаратури, системного та прикладного ПЗ, які впливають або можуть впливати на результати проведення випробувань, про що повідомляє Замовника та представників випробувальної комісії письмово із зазначенням характеру проблеми, що виникла. Випробування призупиняються під час усунення зафіксованої проблеми.
    7. Окремі елементи (етапи) випробувань можуть виконуватись Замовником самостійно, без участі представників Виконавця за взаємною згодою сторін.

## Заходи, що забезпечують безпеку та безаварійність проведення випробувань

* + 1. Необхідний контроль за роботою програмних та технічних засобів, а також каналів зв'язку за суворого дотримання вимог техніки безпеки під час проведення випробувань.
    2. Для усунення ризику втрати даних при перевірці функціонування системи в режимі централізованої обробки даних необхідно щоденне створення резервних копій БД, що перевіряється на центральному сервері в необхідній кількості екземплярів.

## Порядок взаємодії організацій, що беруть участь у випробуваннях

* + 1. Підготовчі роботи з проведення випробувань виконує експлуатуюча організація з участю супроводжуючої організації.
    2. Забезпечення функціонування технічних та програмних засобів під час випробувань провадить експлуатуюча організація.
    3. Перевірки відповідно до цієї програми виконує штатний персонал користувачів МЕДПУНКТУ та фахівці експлуатуючої організації у присутності супроводжуючої організації.
    4. Супроводжувальна організація забезпечує присутність своїх фахівців під час підготовки та проведення випробувань, а також для усунення аварійних ситуацій, пов'язаних із роботою Система.
    5. Взаємодія організацій проводиться у робочому порядку з оформленням відповідних протоколів та актів у разі потреби.
    6. Підготовка протоколів та актів за результатами випробувань проводиться комісією замовника.
    7. Обговорення та затвердження результатів випробувань проводиться на розширеному засіданні комісії у присутності всіх організацій, що беруть участь у випробуваннях.

## Вимоги до персоналу, що проводить випробування, та порядок його допуску до випробувань

* + 1. До персоналу, який проводить випробування, пред'являються такі вимоги:
* персонал має бути штатним;
* персонал повинен бути підготовлений та навчений роботі з АСУ Медпункт;
* персонал повинен бути ознайомлений з експлуатаційною документацією на випробувану систему в частині, що стосується;
* персонал повинен бути ознайомлений з цією програмою та методикою випробувань, наказом на проведення випробувань, технікою безпеки під час проведення робіт;
* персонал повинен мати досвід роботи з Система щонайменше два місяці.
  + 1. Порядок допуску персоналу до випробувань встановлюється наказом проведення випробувань чи іншими розпорядчими документами.

## Перелік робіт, що проводяться після завершення випробувань, вимоги до них, обсяг та порядок проведення

* + 1. По закінченні перевірки функціонування системи в режимі централізованої обробки даних провадиться повернення системи в децентралізований режим роботи МЕДПУНКТУ із збереженням (перенесенням в МЕДПУНКТУ) змін, зроблених у БД на центральному сервері 1 у процесі введення реальних даних з віддалених клієнтських робочих місць МЕДПУНКТУ такому порядку:
* після закінчення випробувань необхідно припинити роботу з БД системи;
* з використанням системних утиліт необхідно сформувати копію БД (змін) МЕДПУНКТУ із центрального сервера 1 (на знімний носій, компакт-диск або ноутбук) для перенесення її на локальний сервер МЕДПУНКТУ;
* відновити конфігурацію клієнтської частини всіх робочих станцій МЕДПУНКТУ (у файлі конфігурації змінити адресу сервера) те щоб вони зверталися до локальному серверу додатків МЕДПУНКТУ;
* заблокувати облікові записи клієнтів МЕДПУНКТУ на сервері 1 так, щоб віддалені користувачі не могли підключатися до системи;
* завантажити отриману копію БД на локальний сервер МЕДПУНКТУ та перевірити коректність завантаження шляхом виконання процедури перевірки БД Система з локального робочого місця, а також формування звітів;
* перевірити вхід до системи з локального клієнта МЕДПУНКТУ.

# Матеріально-технічне забезпечення випробувань

* 1. Для проведення випробувань експлуатуюча організація повинна надати у користування свої штатні технічні, програмні, телекомунікаційні та інші засоби, включаючи:
* локальна обчислювальна мережа (ЛОМ) та необхідну кількість робочих місць;
* ЛОМ Центрального МЕДПУНКТУ;
* канали зв'язку ЛОМ;
* приміщення для проведення випробувань та проведення нарад.

# Метрологічне забезпечення випробувань

* 1. Для вимірювання часових інтервалів можуть використовуватися механічний або електронний секундомір (годинник), а також системний годинник.
  2. Для вимірювання обсягів пам'яті можуть використовуватись штатні засоби операційних систем.

# Звітність

* 1. У процесі тестування та в результаті їх завершення мають бути оформлені такі документи:
* протоколи виконання перевірок (проведення випробувань);
* журнал випробувань (за потреби);
* акт про зупинення випробувань (при необхідності);
* акт про припинення випробувань (за потреби).
  1. Обсяг проведених випробувань та отримані результати записуються до протоколу випробувань. Результати робіт та контрольні виміри засвідчуються підписами сторін-учасників членів комісії безпосередньо на місці випробувань. Усі протоколи затверджуються Головою комісією.
  2. За результатами випробувань складається акт проведення приймальних випробувань системи. Акт підписується членами комісії та затверджується керівниками Виконавця та Замовника протягом 10 днів з моменту закінчення випробувань. При необхідності складається та затверджується акт технічного стану системи після випробувань.

Програма

SQLiteConnection DataBase;

SQLiteCommand CmdDataBase;

SQLiteDataReader ReaderDataBase;

public static string TIME = "DATE('now')";

public List<DBTable> Tables { get; private set; }

public DBTable activeTable { get; set; }

public Accounts LogAcc { get; private set; }

public Doctors LogDoctor { get; private set; }

public ASUDBController()

{

DataBase = new("Data Source = asu.db; Version = 3;");

try

{

DataBase.Open();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

System.Environment.Exit(0);

}

CmdDataBase = new();

Tables = new();

CmdDataBase.Connection = DataBase;

\_Update();

}

public void UpdateData(DBTable table, string varibleUpdate, object valueUpdate, string whereVarible, object whereValue)

{

CmdDataBase.CommandText = table.cmdUpdateDB(varibleUpdate, valueUpdate, whereVarible, whereValue);

try

{

CmdDataBase.ExecuteNonQuery();

\_Update();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

return;

}

}

public void AddData(DBTable table)

{

CmdDataBase.CommandText = table.cmdAddDB();

try

{

CmdDataBase.ExecuteNonQuery();

\_Update();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

return;

}

}

public DBTable GetTable(string nameT, string varible, object value)

{

foreach(var table in Tables)

{

if(table.GetName() == nameT)

{

if(table.HaveData(varible, value))

{

return table;

}

}

}

return null;

}

public void DellData(DBTable table, object value)

{

if (value is int valueINT)

CmdDataBase.CommandText = table.cmdDellDB(valueINT);

else if (value is string valueSTRING)

CmdDataBase.CommandText = table.cmdDellDB(valueSTRING);

if (CmdDataBase.CommandText != DBTable.ErrorExceptrionSTR)

CmdDataBase.ExecuteNonQuery();

\_Update();

}

public void DBClose()

{

if (ReaderDataBase != null)

{

if (!ReaderDataBase.IsClosed)

{

ReaderDataBase.Close();

}

}

DataBase.Close();

}

public void Update()

{

\_Update();

}

private void \_Update()

{

if (Tables.Count > 0)

Tables.Clear();

List<string> tables = new();

CmdDataBase.CommandText = @"SELECT name

FROM sqlite\_schema

WHERE type ='table' AND name NOT LIKE 'sqlite\_%';";

ReaderDataBase = CmdDataBase.ExecuteReader();

while (ReaderDataBase.Read())

{

tables.Add(ReaderDataBase.GetString(0));

}

ReaderDataBase.Close();

foreach (var table in tables)

{

CmdDataBase.CommandText = $"SELECT \* FROM {table}";

ReaderDataBase = CmdDataBase.ExecuteReader();

while (ReaderDataBase.Read())

{

if (table == "AccActions")

{

Tables.Add

(

new AccActions

{

ID = ReaderDataBase.GetInt32(0),

CreatedBy = ReaderDataBase.GetString(1),

CreatedAt = ReaderDataBase.GetString(2),

Action = ReaderDataBase.GetString(3)

}

);

}

else if (table == "Accounts")

{

Tables.Add

(

new Accounts

{

ID = ReaderDataBase.GetInt32(0),

Username = ReaderDataBase.GetString(1),

Password = ReaderDataBase.GetString(2),

CreatedBy = ReaderDataBase.GetInt32(3),

RankID = ReaderDataBase.GetInt32(4)

}

);

}

else if (table == "Clients")

{

try

{

Tables.Add

(

new Clients

{

ID = ReaderDataBase.GetInt32(0),

Username = ReaderDataBase.GetString(1),

Sex = ReaderDataBase.GetString(2),

DateOfBirth = ReaderDataBase.GetString(3),

IDRhFactor = ReaderDataBase.GetInt32(4),

Weight = ReaderDataBase.GetString(5),

Growth = ReaderDataBase.GetInt32(6),

NumberPhone = ReaderDataBase.GetString(7),

CreatedAt = ReaderDataBase.GetString(8),

DocID = ReaderDataBase.GetInt32(9),

MedStory = ReaderDataBase.GetString(10)

}

);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else if (table == "DayOfWeek")

{

Tables.Add

(

new DayOfWeek

{

ID = ReaderDataBase.GetInt32(0),

Day = ReaderDataBase.GetString(1)

}

);

}

else if (table == "Doctors")

{

if(ReaderDataBase.GetInt32(0) != 100)

{

Tables.Add

(

new Doctors

{

ID = ReaderDataBase.GetInt32(0),

Username = ReaderDataBase.GetString(1),

Sex = ReaderDataBase.GetString(2),

DateOfBirth = ReaderDataBase.GetString(3),

IDPassport = ReaderDataBase.GetInt32(4),

CreatedAt = ReaderDataBase.GetString(5),

CreatedBy = ReaderDataBase.GetInt32(6),

IDSpec = ReaderDataBase.GetInt32(7)

}

);

}

}

else if (table == "DrugsDef")

{

Tables.Add

(

new DrugsDef

{

ID= ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1),

Amount = ReaderDataBase.GetInt32(2),

Category = ReaderDataBase.GetString(3),

Desc = ReaderDataBase.GetString(4)

}

);

}

else if (table == "GroupPerms")

{

Tables.Add

(

new GroupPerms

{

GroupID = ReaderDataBase.GetInt32(0),

PermID = ReaderDataBase.GetInt32(1)

}

);

}

else if (table == "Perms")

{

Tables.Add

(

new Perms

{

ID = ReaderDataBase.GetInt32(0),

CreatedBy= ReaderDataBase.GetString(1),

CreatedAt= ReaderDataBase.GetString(2),

Key = ReaderDataBase.GetString(3),

Desc= ReaderDataBase.GetString(4)

}

);

}

else if (table == "PhysDef")

{

Tables.Add

(

new PhysDef

{

ID = ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1),

CreatedAt = ReaderDataBase.GetString(2),

Category= ReaderDataBase.GetString(3),

Desc = ReaderDataBase.GetString(4),

}

);

}

else if (table == "Plan")

{

Tables.Add

(

new Plan

{

ID = ReaderDataBase.GetInt32(0),

Username = ReaderDataBase.GetString(1),

PlanSheduler = ReaderDataBase.GetString(2),

CreatedBy = ReaderDataBase.GetString(3),

CreatedAt = ReaderDataBase.GetString(4),

Desc = ReaderDataBase.GetString(5)

}

);

}

else if (table == "PlanningCl")

{

Tables.Add

(

new PlanningCl

{

ClientID = ReaderDataBase.GetInt32(0),

PlanID = ReaderDataBase.GetInt32(1)

}

);

}

else if (table == "ProcedClients")

{

Tables.Add

(

new ProcedClients

{

ProcedureID = ReaderDataBase.GetInt32(0),

ClientID = ReaderDataBase.GetInt32(1) ,

DocID = ReaderDataBase.GetInt32(2),

Time = ReaderDataBase.GetString(3)

}

);

}

else if (table == "Procedures")

{

Tables.Add

(

new Procedures

{

ID = ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1)

}

);

}

else if (table == "RankGroups")

{

Tables.Add

(

new RankGroups

{

ID = ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1)

}

);

}

else if (table == "RhFactor")

{

Tables.Add

(

new RhFactor

{

ID = ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1)

}

);

}

else if (table == "Shedule")

{

Tables.Add

(

new Shedule

{

ID = ReaderDataBase.GetInt32(0),

Time = ReaderDataBase.GetString(1)

}

);

}

else if (table == "SheduleDoctors")

{

Tables.Add

(

new SheduleDoctors

{

DocID = ReaderDataBase.GetInt32(0),

TimeID = ReaderDataBase.GetInt32(1),

DayID = ReaderDataBase.GetInt32(2)

}

);

}

else if (table == "Sheduler")

{

Tables.Add

(

new Sheduler

{

ID = ReaderDataBase.GetInt32(0),

ClientID = ReaderDataBase.GetInt32(1),

DoctorID = ReaderDataBase.GetInt32(2),

DateTime = ReaderDataBase.GetString(3),

DrugID = ReaderDataBase.GetInt32(4),

ActionID = ReaderDataBase.GetInt32(5)

}

);

}

else if (table == "Specialization")

{

Tables.Add

(

new Specialization

{

ID = ReaderDataBase.GetInt32(0),

Name = ReaderDataBase.GetString(1)

}

);

}

}

ReaderDataBase.Close();

}

}

Додаток А. Методики випробувань

# Попередні умови для перевірки виконання вимог функціонального призначення системи

## Перевірка заповнення бази даних

Перед початком роботи впевнитися в тому, що всі таблиці заповнені.

# Опис методик тестування

## Перевірка функціонування системи у режимі централізованої обробки даних

|  |  |  |  |
| --- | --- | --- | --- |
| № перевірки | Назва перевірки | Методика виконання | Результат виконання |
| 1 | Заповнення списка лікарів | Заповнення всіх складових панелі «Лікарі» | Вивід заданих рандомними значеннями |
| 2 | Заповнення списка хворих | Заповнення всіх складових панелі «Хворі» | Вивід заданих рандомними значеннями |
| 3 | Заповнення пункта “медина картка” | Заповнення всіх складових панелі «Медична картка» | Вивід заданих рандомними значеннями |

## Перевірка відповідності функціональності системи вимогам ТЗ

Перевіряється факт наявності функціональності, наведеної в пп 2.3 ТЗ на створення «Системи».

Деякі пункти можуть бути перевірені на коректність реалізації за згодою членів комісії.

|  |  |
| --- | --- |
| **Перелік пунктів ТЗ** | **Перевіряються обов’язково** |
| Вхід адміністратора |  |
| Вхід користувача | + |
| Створення нової медичної картки |  |
| Пошук лікаря |  |
| Додання інформації щодо огляду | + |
| Пошук ліків |  |
| Заповнення медичної картки |  |
| Вибір наступної дати прийому |  |
| Відстежування лікування | + |

# Додаток Б. Тестові приклади

**1.1 Користувачі програми**

У проекті присутні дві категорії користувачів, які мають різні функції, – це адміністратор і користувач, при цьому проект зорієнтований більшою мірою на адміністратора.

Належність користувача або адміністратора визначається вибором відповідної програми.

Користувачем у проекті вважається людина, яка користується послугами медичного пункту. Користувач персоніфікується при вході у програму, зазначаючи свої прізвище-ім’я, свій логін та пароль, обирає лікаря, а також має право на перегляд своєї медичної картки.

Адміністратор авторизується при вході у програму, вводячи набір ім’я-пароль з вже існуючих даних. Адміністратор підтримує списки які існують у ПЗ: має право на додавання нових, редагує та видаляє існуючі, має право на запис виданих ліків лікарям, створює квитанції щодо лікування. Сортує лікарів та хворих за графіком прийому свого медичного персоналу, може робити пошук серед хворих за критеріями імені, прізвища та дати народження.

Питання такі, як захист паролів та безпека даних, у даній роботі не розглядаються.

**1.2 Функції програми**

Програма надає користувачу наступні можливості:

1. Вхід у програму;
2. Запис до лікаря;
3. Вибір лікаря;
4. Перегляд медичної картки;
5. Вихід із програми.

Програма надає адміністратору наступні можливості:

1. Вхід у програму;
2. Облік ліків;
3. Організація роботи персоналу;
4. Робота із записами до лікарів;
5. Доступ до медичних картов;
6. Перегляд персональної інформації працівників та користувачів;
7. Перегляд та організація графіків прийому;
8. Вихід із програми.

Загальні функції:

1. Завантаження даних з постійної пам’яті в оперативну;
2. Збереження поточного стану даних у постійній пам’яті;
3. Генерація тестових даних.

**1.3 Функції адміністратора**

*Функція «Вхід у програму»*

Адміністратор входить у програму за допомогою одного з наборів ім’я – пароль. Імена та паролі адміністраторів зберігається в даних програми. Усі інші функції адміністратора стають доступними одразу після правильного введення імені та пароля.

Основний сценарій

1. Після запуску програми відкривається вікно входу адміністратора (рис. 1.1).
2. Адміністратор вводить ім’я і пароль і натискає на кнопку «Увійти».
3. Введені дані перевіряються, і відкривається головне вікно адміністратора (рис. 1.2).
4. Якщо дані були введені некоректно, на формі вводу з’являється повідомлення про це, і користувач намагається увійти знову. Кількість спроб необмежена.



Рисунок 1.1 – Вікно входу адміністратора/користувача

*Функція «Лікарі»*

Для того щоб праце здібність була на вищому рівні, повинна бути функція для заповнення лікарів. Тому адміністратор повинен їх заповнити в першу чергу.

Вона складається з заповнення персональних даних лікарів та додавання їх до реєстру. У запис реєстрації входить: прізвище, ім’я, по-батькові, дата народження, посада, спеціалізація, стаж, графік роботи, паспорт, прийнятий на роботу, ким доданий в базу, к-ть прийомів в місяць, робочих годин в місяці, фото лікаря.

Сценарій «Заповнення лікаря»

1. На спеціалізованій панелі, адміністратор переходить на вкладку «Доктори» (рис. 1.3).
2. Натискає на кнопку у вигляді знаку «Плюс».
3. Заповнює спеціальні панелі.
4. Після заповнення даних, автоматично зберігаються та додаються до реєстру.
5. Після зроблених операцій є можливість перегляду даних про лікаря.

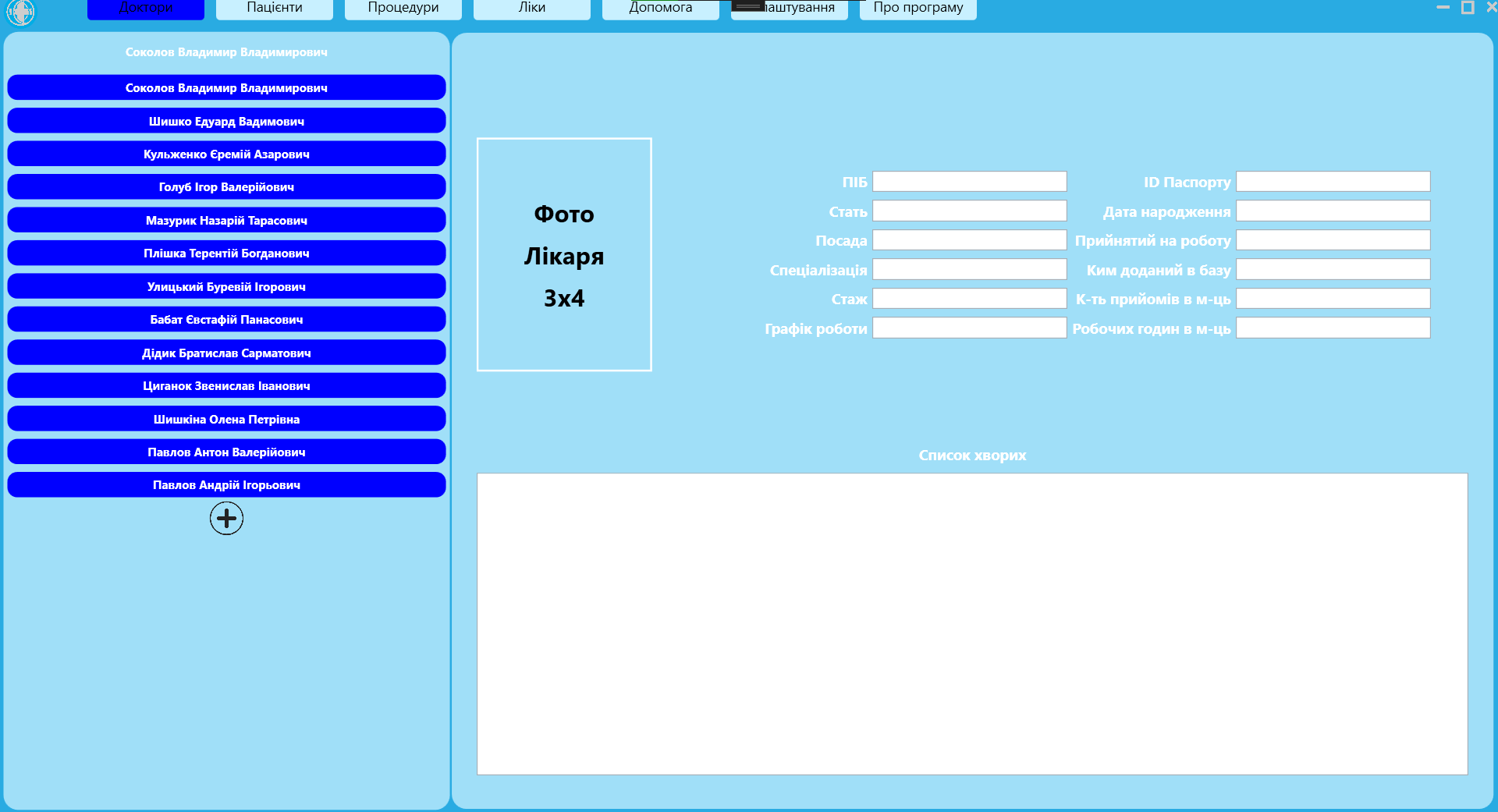


Рисунок 1.2 – Вкладка “Доктори”

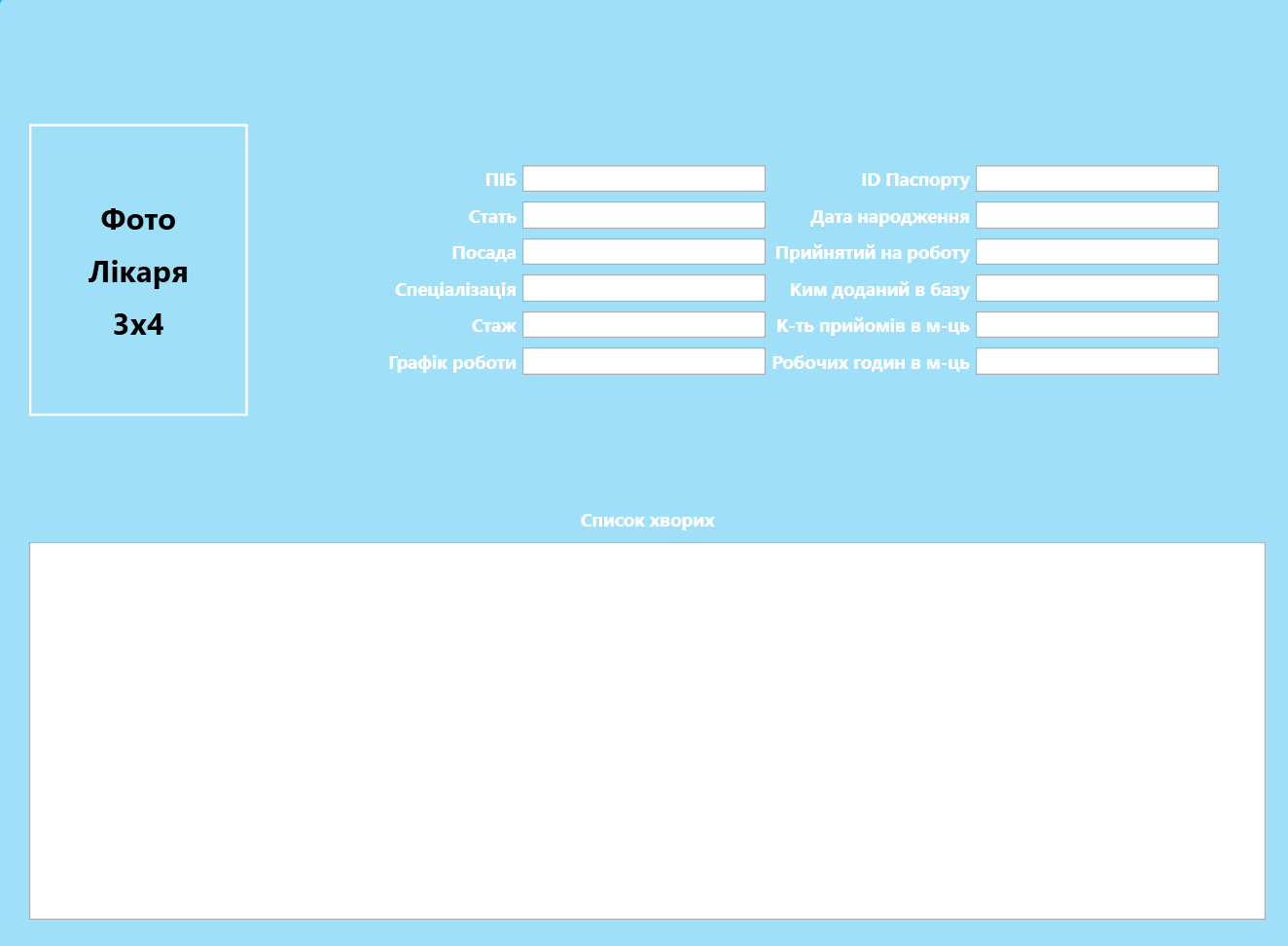


Рисунок 1.3 – Вкладка інформації про лікаря

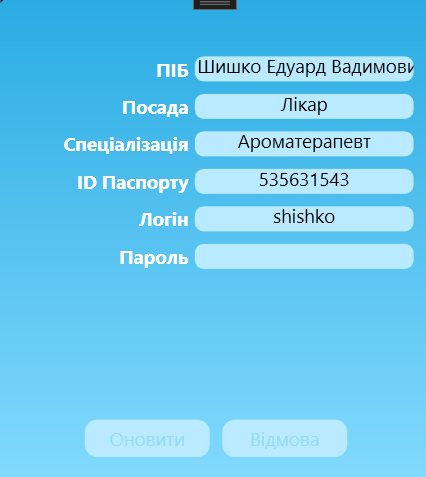


Рисунок 1.4 – Вікно оновлення даних про лікаря

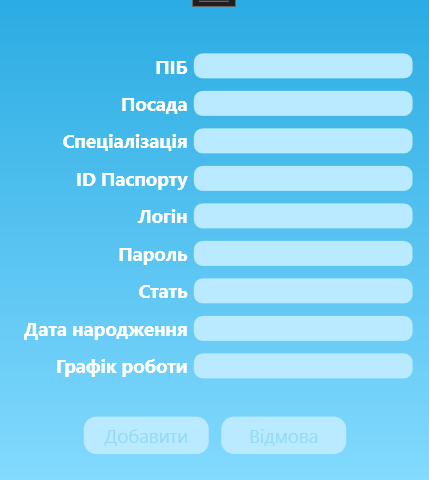


Рисунок 1.5 – Вікно додавання даних про лікаря

*Функція «Організація списків пацієнтів»*

Окрім заповнення лікарів, потрібно заповнити списки пацієнтів. Вони мають право на звичайних доступ до системи. Заповняються в них майже такі ж самі дані як і у лікарів, але додаються інші: група крові, номер телефону, П.І.Б. , стать, дата народження, зріст, вага, стан лікування, лікар.

Сценарії заповнення інформації про пацієнта, робить сам пацієнт. Він не відрізняється нічим, але має обмежений функціонал можливостей.

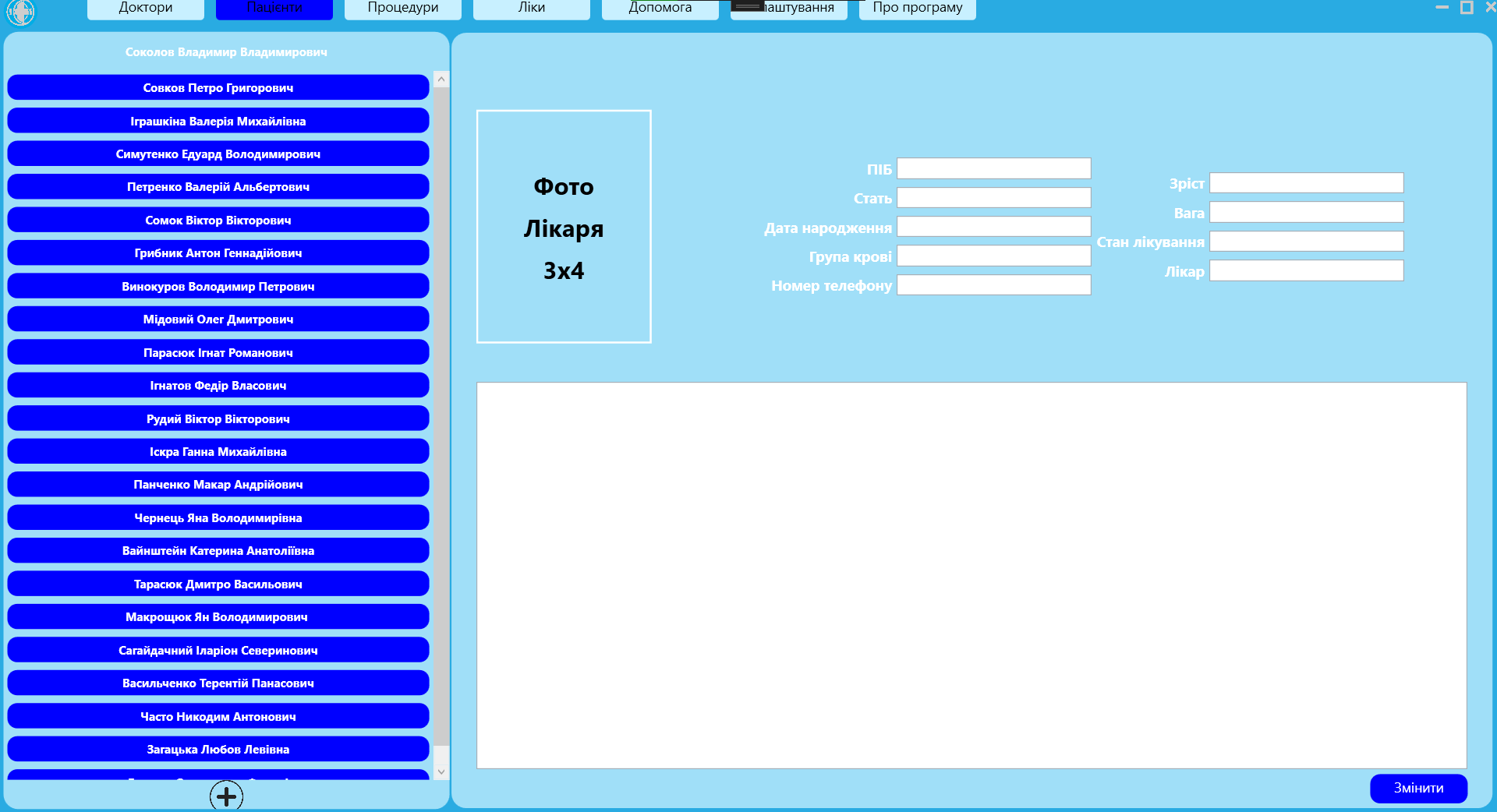


Рисунок 2.1 – Вкладка “Про пацієнта”

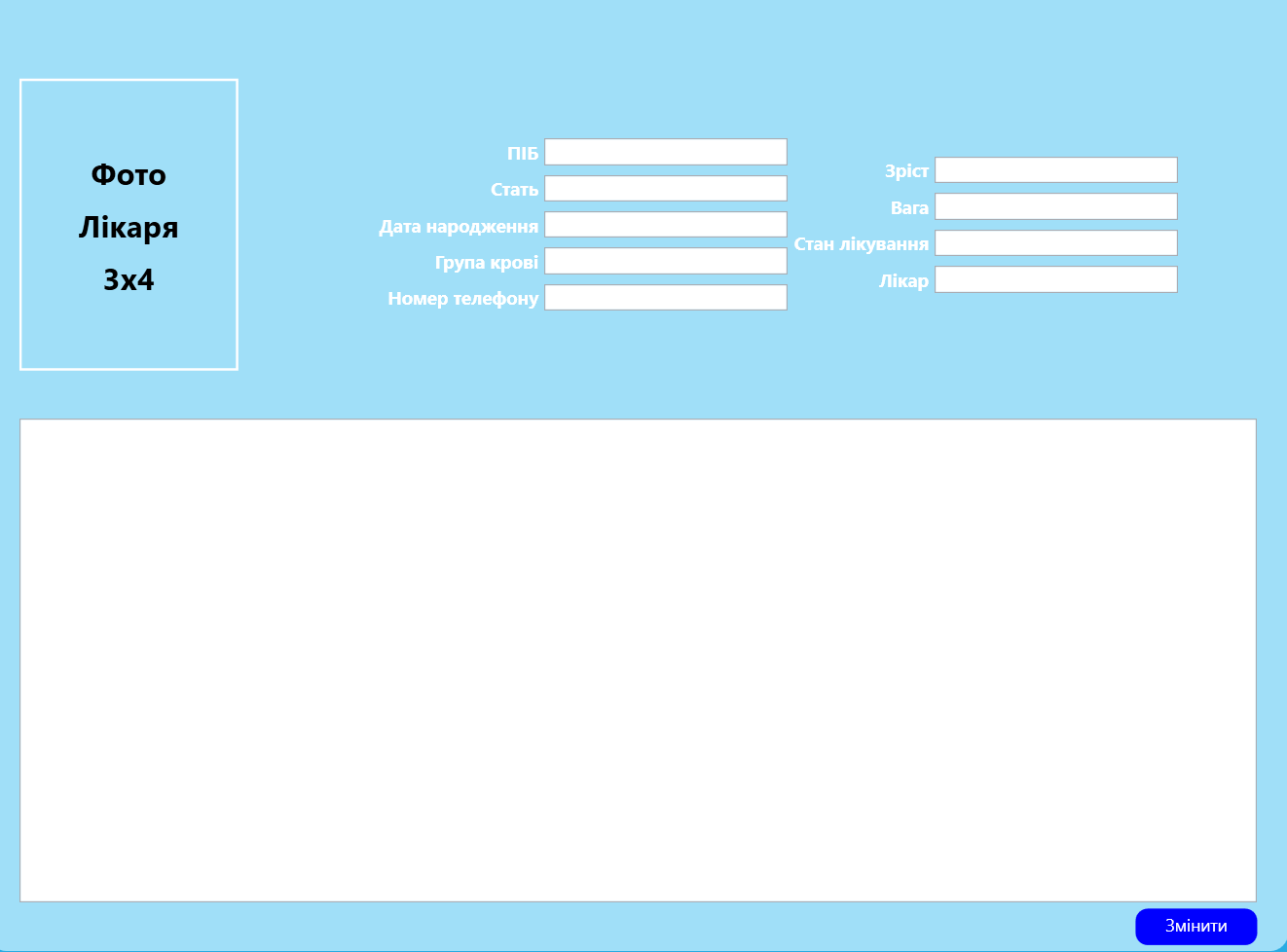


Рисунок 2.2 – Вкладка “Про пацієнта”

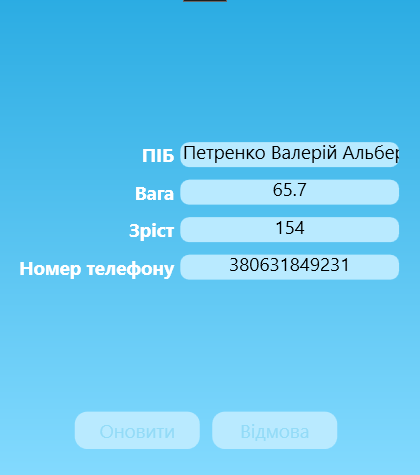


Рисунок 2.3 – Вікно оновлення інформації

*Функція «Процедури»*

Дана функція реалізує запис пацієнтів лікарями на процедури які їм прописані.

Сценарій «Запис на процедури»

1. На вкладці «Процедури» лікар обирає необхідну процедуру хворому.
2. Заповнює дані щодо проводження: назва процедури, хто проводить, пацієнт, час.
3. Якщо відбуваються якісь форс мажорні обставини, є функція редагування.

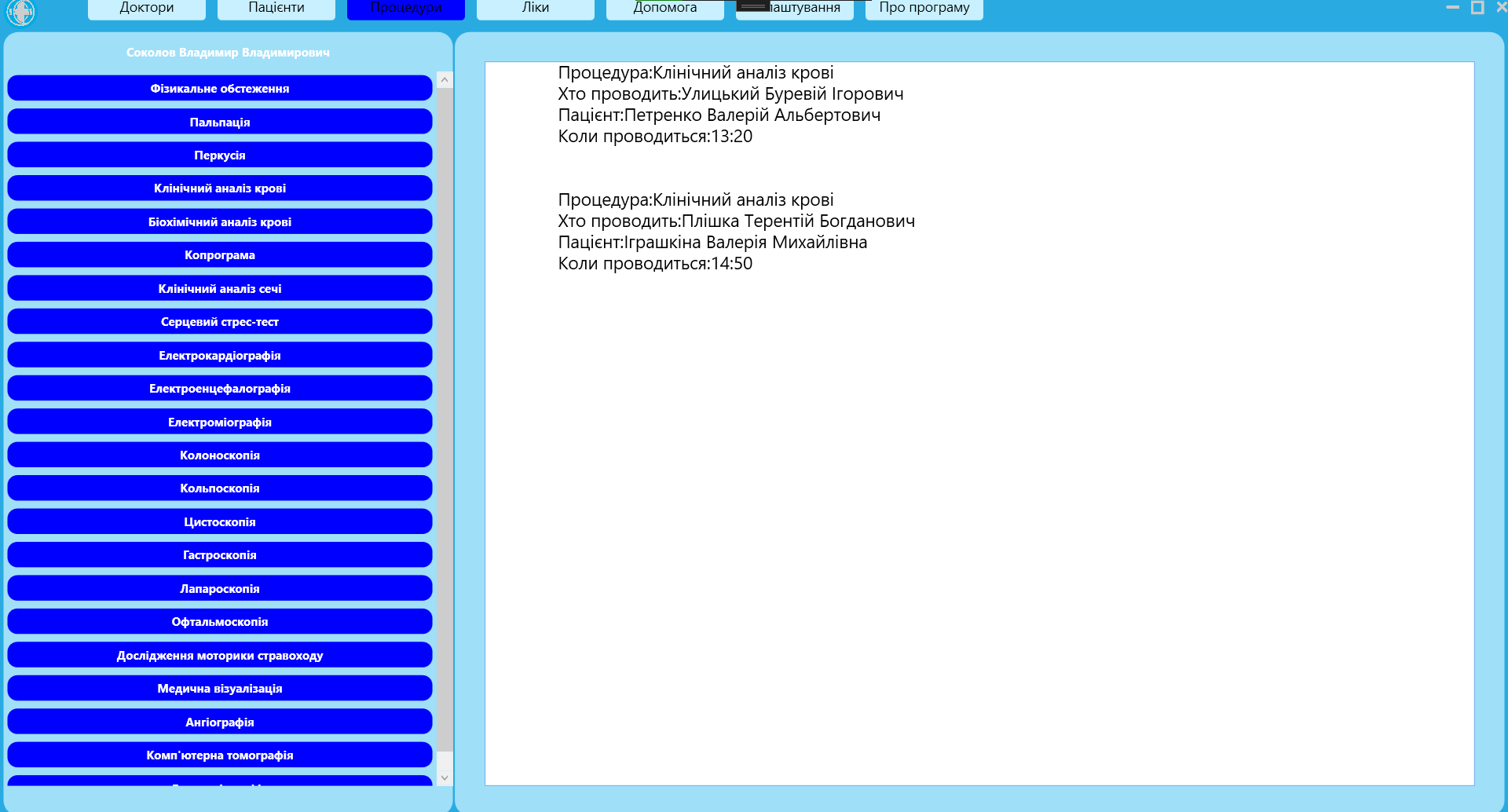


Рисунок 3.1 – Вкладка “Процедури”

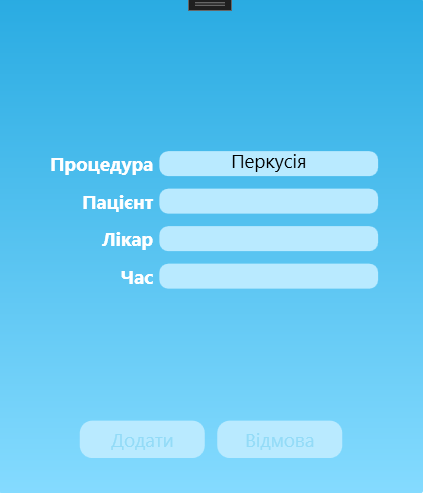


Рисунок 3.2 – Вікно додавання процедури

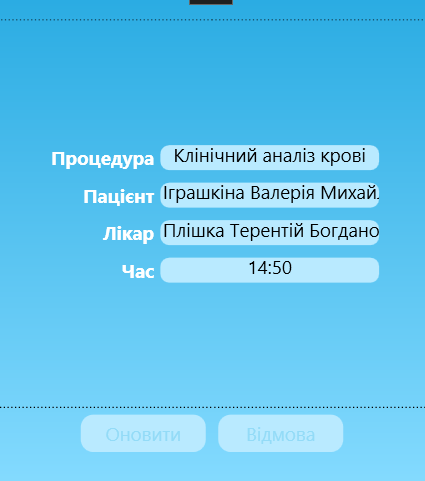


Рисунок 3.2 – Вікно оновлення процедури

*Функція «Про програму»*

Функція показує для всіх ліцензію даної автономної системи управління. Кожен з користувачів може ознайомитись з нею.

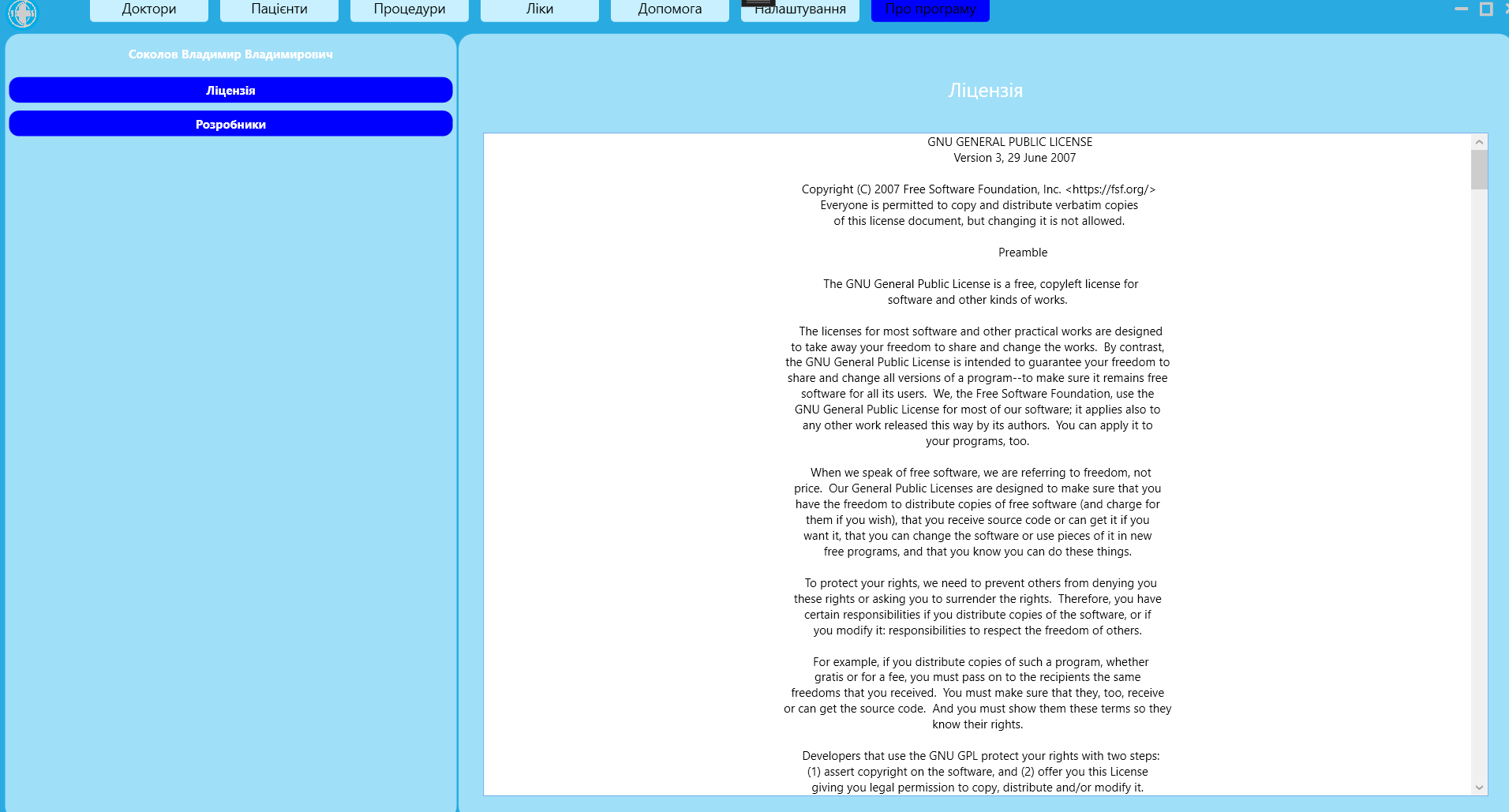


Рисунок 4.1 – “Про програму”

*Функція «Вихід із програми»*

У процесі роботи адміністратор/користувач вносить зміни до даних, які зберігаються у спільному для адміністратора та користувача файлі даних медичного пункту. Відповідно при авторизації нового користувача та реєстрації ним відбувається автоматичне зберігання змін у цьому файлі. При виході з програми повідомлень про додаткове збереження немає, вона завершує свою роботу.