

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ В.Н. КАРАЗІНА  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК

КУРСОВА РОБОТА

з дисципліни «Об'єктно орієнтоване програмування»

Тема «Health Helper (помічник здоров'ю)»

Оцінка \_\_\_\_\_ балів / \_\_\_\_\_

Члени комісії:

\_\_\_\_\_ Поклонський Є. В.

\_\_\_\_\_

Виконав студент 2 курсу

групи КС-22

Моцний Данііл Вячеславович

Перевірив:

доцент

Поклонський Є. В.

## ЗМІСТ

КУРСОВА РОБОТА.....	1
ВСТУП .....	3
РОЗДІЛ 1 КОНЦЕПЦІЯ .....	4
1.1. Загальна концепція продукту .....	4
1.2. Основні можливості та опитування.....	4
РОЗДІЛ 2 АНАЛІЗ .....	5
2.1. Аналіз існуючого програмного забезпечення. ....	5
РОЗДІЛ 3 ПРОЕКТУВАННЯ .....	6
3.1 Вибір мови програмування.....	6
3.2 Розробка.....	8
3.3 Класи UI.....	9
3.4 Класи бізнес-логіки .....	9
РОЗДІЛ 4 ІНСТРУКЦІЯ КОРИСТУВАЧЕВІ.....	10
РОЗДІЛ 5 ТЕСТУВАННЯ .....	16
ВИСНОВКИ.....	17
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	18
ДОДАТКИ.....	19

## ВСТУП

Мета сучасної медичної реформи – забезпечити всім громадянам України рівний доступу до якісних медичних послуг та перебудувати систему охорони здоров'я так, щоб у її центрі був пацієнт. Тому важливо забезпечити зручне спілкування пацієнта і лікаря, надати необхідну лікарську допомогу в будь який час. Таке спілкування можна здійснити за допомогою сучасних інтернет-комунікаційних засобів.

**Актуальність:** Тому розробка додатку який дає можливість пацієнту отримувати консультації від лікаря в зручний для них час є актуальною задачею.

**Мета дослідження:** На основі аналізу існуючого програмного забезпечення, що здійснює взаємодію лікаря і хворого, опитування лікарів зробити систематизацію і узагальнення деяких медичних показників, виявити вимоги, яким має задовольняти програма і розробити такий додаток.

**Об'єкт дослідження:** процес інформаційно-комунікаційного спілкування лікаря і пацієнта.

**Предмет дослідження:** Програмний додаток, що забезпечує спілкування пацієнта з лікарем, надає можливість зробити просту діагностику захворювання.

Для вирішення поставленої задачі було задіяні такі **методи дослідження:** тестове опитування лікарів, щодо інформації, яка їм необхідна для первинної діагностики, прикладного програмування.

Робота має прикладний характер.

**Наукова новизна** роботи полягає в тому що на основі зібраних даних було систематизовано і узагальнено деякі медичні показники, встановлено показники, які необхідні для встановлення первинного діагнозу, що і було віддзеркалено у програмному додатку, за допомогою якого лікар зможе вчасно надати допомогу пацієнту;

## РОЗДІЛ 1 КОНЦЕПЦІЯ

### 1.1. Загальна концепція продукту

В результаті того що виник напрямок електронної медицини, яка на сьогодні є однією з найбільш швидкозростаючих на світовому ринку. Досвід реалізації ініціатив електронної медицини в Україні поки досить мізерний. Тому один з ключових аспектів розробки - створення й заповнення застосунка для сумісної роботи лікаря та пацієнта.

### 1.2. Основні можливості та опитування

З метою виявлення потреб у вдосконаленні засобів комунікації з пацієнтом було опитано лікарів центру первинної медико-санітарної допомоги №11. Лікарі висловились про необхідність створення програми, яка дозволить пацієнту безперервно знаходитися під наглядом лікаря без відвідування лікарні, надсилаючи лікарю найголовніші показники. При цьому пацієнт може отримати необхідну консультацію фахівця і вразі погіршення стану здоров'я - вчасно отримати необхідну пораду лікаря.

## РОЗДІЛ 2 АНАЛІЗ

### 2.1. Аналіз існуючого програмного забезпечення.

Прикладом існуючого подібного програмного забезпечення для взаємодії лікаря і пацієнта є мобільний додаток LEKARIS.com. Продукт розробила українська компанія "Сучасні медичні сервіси", в якій окрім ІТ-фахівців працює велика команда кваліфікованих лікарів. Цей сервіс передбачає і має ресурси для надання платних медичних послуг, що є по суті додатковим заробітком для лікарів. Це може відлякувати пацієнтів і приведе до того, що деяка категорія пацієнтів опиниться поза межею послуги.

На основі вищезазначеного можна зазначити що в Україні відсутні доступні програми для взаємодії лікаря і пацієнта.

Для рішення цієї проблеми необхідно створити:

- на основі аналізу існуючого програмного забезпечення, що здійснює взаємодію лікаря і хворого, опитування лікарів зробити систематизацію і узагальнення деяких медичних показників, виявити вимоги, яким має задовольняти програмний додаток;
- розробити програму «Health Helper», яка допоможе здійснювати спілкування хворого і лікаря, швидко передати лікарю зміни в стані пацієнта.
- додаток повинен мати мінімальні вимоги до апаратного забезпечення.

## РОЗДІЛ 3 ПРОЕКТУВАННЯ

### 3.1 Вибір мови програмування.

Для створення додатку було обрано кросплатформенну мову програмування Java, середовище розробки IntelliJIDEA, платформу для створення візуального інтерфейсу JavaFX 8.0.192, локальну базу даних SQLite, бібліотеку Javafx (Javafx.mail) та Gmail.API.

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

JavaFX — платформа та набір інструментів для створення насичених інтернет-застосунків (англ. Rich Internet Applications, RIA) з можливістю підвантаження медіа та змісту. Вперше продемонстровано Sun Microsystems на Міжнародній конференції Java-розробників JavaOne у травні 2007. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та надавати розвинуті

інтернет-застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

JavaFX складається з JavaFX Script і JavaFX Mobile. Починаючи з випуску JavaFX 2.0 забезпечено можливість створення JavaFX-застосунків, написаних цілком мовою Java. Для розробки застосунків доступний багатий графічний і мультимедійний API, що спрощує створення візуальних програм.

SQLite — полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код SQLite поширюється як суспільне надбання (англ. *public domain*), тобто може використовуватися без обмежень та безоплатно з будь-якою метою.

Клієнтська бібліотека Google *API* для Java - надає функціональність, загальну для всіх API Google, наприклад, транспорт HTTP, обробку помилок, аутентифікацію, розбір JSON, завантаження / завантаження медіа та пакетне завантаження. Бібліотека містить потужну бібліотеку OAuth 2.0 з послідовним інтерфейсом; легкі, ефективні моделі даних XML і JSON, які підтримують будь-яку схему даних; та підтримка буферів протоколів.

IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.<sup>[3]</sup>) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію.

### 3.2 Розробка

Додаток складається з 11 Java-класів, 6-ти під'єднаних fxml-файлів, 3-х «локалізуючих» properties-файлів та файлу локальної бази даних. Для зв'язку з базою даних використовується клас «Conn». В якому імпортується стороння бібліотека SQLite. Для надсилання результату створено клас Send, в якому, для цього імпортовано бібліотеку Javax.mail

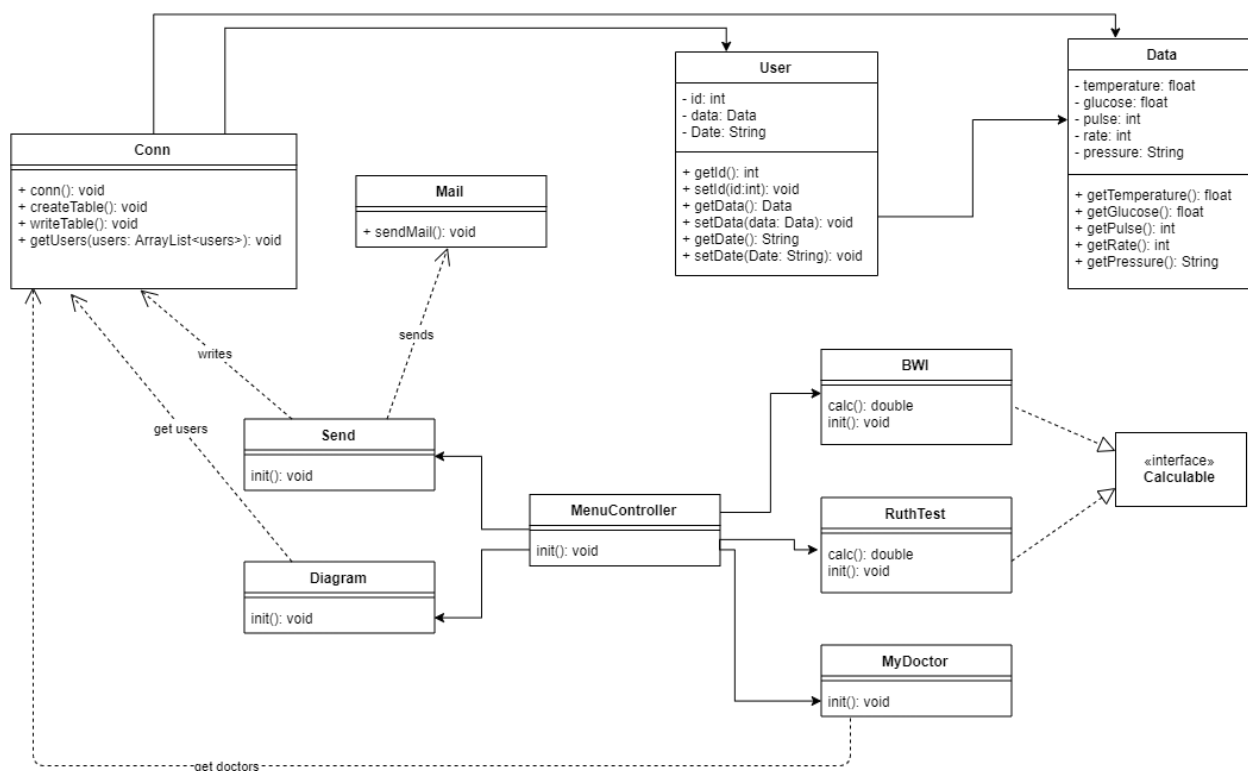


Рис. 3.1, діаграма UML

В Main знаходяться декілька основних методів - налаштування вікна додатка start та метод для створення бази даних і запуску платформи Java FX - main. Для створення бази даних викликаються статичні методи класу Conn (класично застосовані). Інтерфейс вкладки створено в окремому файлі (*main.fxml*) та підключено до відповідного класу-контролера.



### 3.3 Класи UI

Клас MainController – є основник «каркас» додатку з якого виника наслідування інших класів-наслідників.

Клас MyDoctor – пов'язаний з базою даних таким чином: з класу Conn викликається метод ReadDB(1) та заповнює елемент ComboBox.

Один з найважливіших класів додатка це SendMail за допомогою якого відбувається основна функція вкладки *“Відправити результат”* для цього створено відповідні поля та методи . Метод SendFromGmail (2) - статичний та використовується у класі Send (який в свою чергу є конструктором для вкладки *“Відправити результат”*). Інтерфейс вкладки створено в окремому файлі (*send.fxml*) та підключено до відповідного класу-контролера.

### 3.4 Класи бізнес-логіки

Клас DM – це втілення в додаток класичної діаграми Java FX.

Клас BWI – набір методів та полів калькулятора для індексу маси тіла.

Клас RuthTest – набір методів та полів калькулятора для підрахунка проби Руф’є з використанням методу (2).

Для надсилання листа на електронну адресу використовується Gmail.API.

## РОЗДІЛ 4 ІНСТРУКЦІЯ КОРИСТУВАЧЕВІ

Як саме він працює? Все дуже легко. Щоб почати користуватись додатком треба отримати свій унікальний номер за допомогою якого лікар буде Вас ідентифікувати. Далі зайдіть в додаток та оберіть зручну для вас мову, а після цього необхідно вказати унікальний номер у відповідне поле в розділі «відправити результат». Після того як заповнили всі поля, тисніть кнопку «відправити». Це основний принцип поєднання лікаря з пацієнтом. Додаток розподіляється на декілька змістових частин, одна з них використовується для підрахунку медичних тестів, а дві інші ілюструють та надсилають результати. Також було впроваджено набір функцій які дозволяють дослідити результати тестування. Головне меню виглядає таким чином:

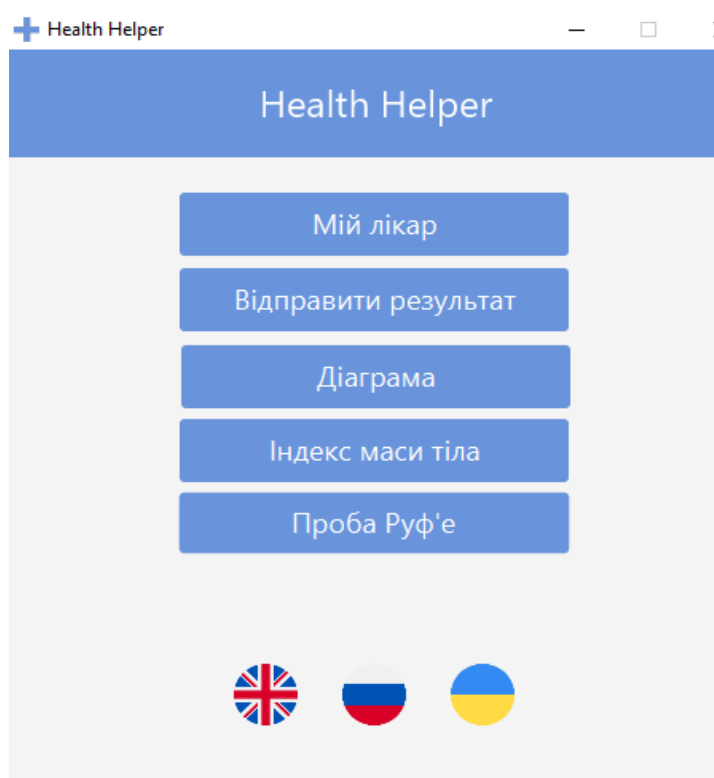
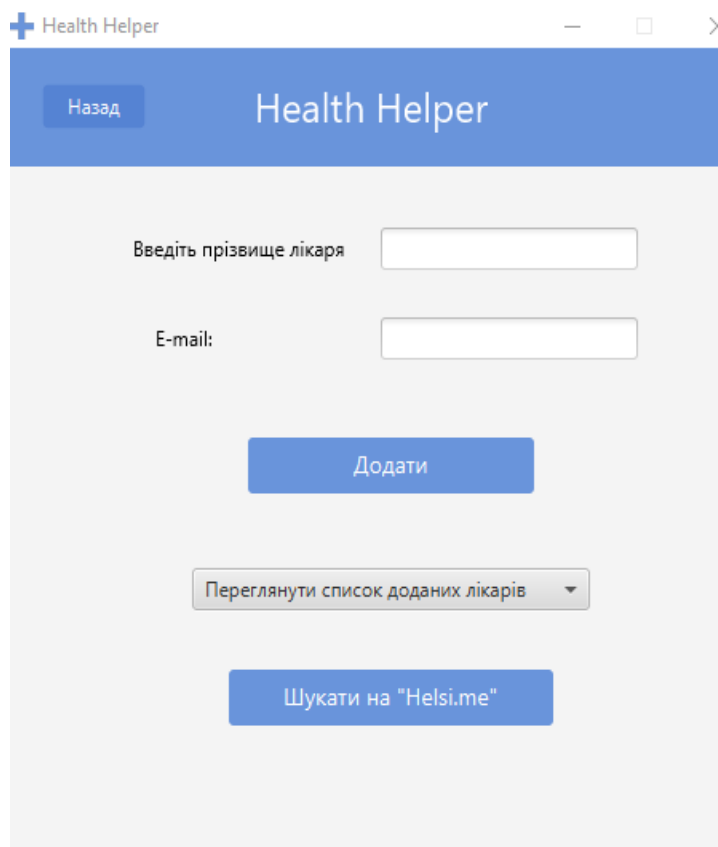


Рис. 4.1, Головне меню

Меню інтуїтивно-зрозуміле для будь-якого користувача. Воно складається з декількох вкладок та кнопок вибору мови.

Мій лікар. Вгорі знаходиться вкладка “Мій лікар”.



The screenshot shows a web application window titled 'Health Helper'. At the top left is a blue header bar with a white plus icon and the text 'Health Helper'. Below the header, on the left, is a blue button labeled 'Назад'. The main content area has a light gray background. It contains two input fields: the first is labeled 'Введіть прізвище лікаря' and the second is labeled 'E-mail:'. Below these fields is a blue button labeled 'Додати'. Underneath the 'Додати' button is a gray button with the text 'Переглянути список доданих лікарів' and a downward arrow. At the bottom is a blue button labeled 'Шукати на "Helsi.me"'. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

Рис. 4.2, Розділ «Мій лікар»

У відведені поля користувач вводить прізвище свого лікаря та e-mail для результатів та спілкування. Після того як він тисне “Додати” в ComboBox-і з’явиться прізвище лікаря, щоб його можна було переглянути на сайті “Helsi.me” та впевнитись в коректності написання, або більш прикладному застосуванні сайту такому як бронювання відвідування лікаря.

The screenshot shows a web application window titled 'Health Helper'. At the top left is a blue header bar with a white plus icon and the text 'Health Helper'. Below the header, on the left, is a blue button labeled 'Назад' (Back). The main content area has a light gray background. It contains several input fields for health data: 'ID', 'Температура' (Temperature), 'Пульс' (Pulse), 'Тиск' (Pressure), 'Частота дихання' (Breathing frequency), and 'Глюкоза крові' (Blood glucose). Each label is followed by a white rectangular input box. Below these fields is a dropdown menu labeled 'Оберіть e-mail' (Select e-mail). At the bottom of the form is a large blue button labeled 'Відправити результат' (Send result).

Рис. 4.3, Розділ «Відправити результат»

Відправити результат. Вкладка “Відправити результат”, основна вкладка цього додатка, де користувач вводить основні показники та надсилає лікарю, обравши раніше вказаний e-mail.

За допомогою класичних показників таких як : пульс, температура, тиск - лікар визначає зміни у стані здоров'я, можливо це не буде точний діагноз але визначити покращення або навпаки погіршення в здоров'ї людини, буде набагато простіше. Таким чином, лікарі мають можливість дистанційно коректувати курс лікування та спостерігати хворим. Як було сказано раніше.

Діаграма. Далі знаходиться вкладка “Діаграма”, вона має важливе значення для моніторингу медичних показників.

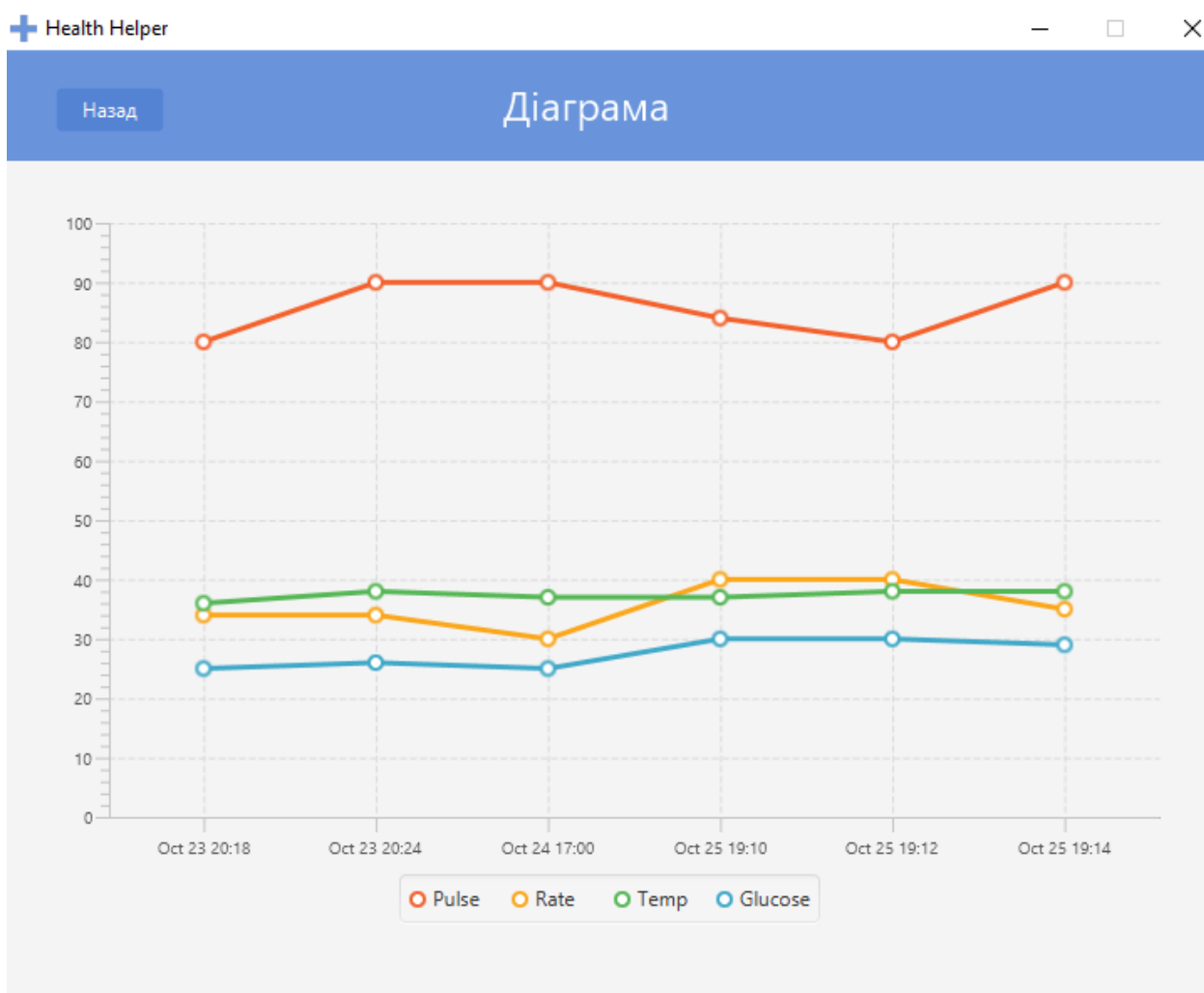


Рис. 4.4, Розділ «Діаграма»

Пульс — а точніше частота пульсу вимірюється в кількості ударів в хвилину. Частота пульсу може бути використана для контролю здоров'я серця і рівня фізичної підготовки взагалі. Звичайно чим пульс нижче, тим краще, але в разі брадикардії це може бути небезпечно. Тривожними симптомами при низькому пульсі є слабкість і непритомність.

Температура тіла — комплексний показник теплового стану організму людини. Нормальна температура тіла людини залежить від точки (місця) вимірювання, часу доби та фізіологічної активності. Температура під пахвами вважається нормальною при вимірюванні ртутним термометром в проміжку 36,5 — 37,2 °C. Ближче до вечора та після прийому їжі вона підвищується.

Кров'яний тиск — тиск, який кров справляє на стінки кровоносних судин, або, інакше кажучи, перевищення тиску рідини в кровоносній системі над атмосферним тиском. Це один з найважливіших параметрів, що характеризує роботу кровоносної системи. Кров'яний тиск вимірюють за допомогою широкої стрічки, яку називають манжетою і вдягають на передпліччя. Повітря напompовують в середину манжети. Кров'яний тиск вимірюють тоді, коли повітря виходить з манжети. Кров'яний тиск на діаграмі вказується середнім арифметичним між верхнім та нижнім числом.

Проба Руф'є. Сторінка Проблема Руф'є — це простий непрямий метод визначення фізичної працездатності, в якому використовуються значення частоти серцевих скорочень (ЧСС) в різні періоди відновлення після відносно невеликих навантажень. Також тут присутня основна функція вкладки “Відправити результат”, працює аналогічно. Для проходження тесту надається інструкція та все необхідне для підрахунку.

Health Helper

Назад Проба Руф'є

Вкажіть вік від 7 до 18 (повних років).

Крок 1. Заміряйте пульс за допомогою секундоміра нижче, протягом 15 секунд в стані спокою після 5-хвилинного відпочинку. Занесіть отриманий результат в поле P1.

Крок 2. Зробіть 30 присідань за 45 секунд. Знову заміряйте пульс за 15 секунд. Занесіть отриманий результат в поле P2.

Крок 3. Хвилину відпочивайте.

Крок 4. Знову заміряємо пульс 15 секунд. Занесіть отриманий результат в поле P3.

Крок 5. Натисніть кнопку розрахунку!

1

15

21

23

22

Рахувати

0 Старт/Стоп Очистити Оберіть e-mail

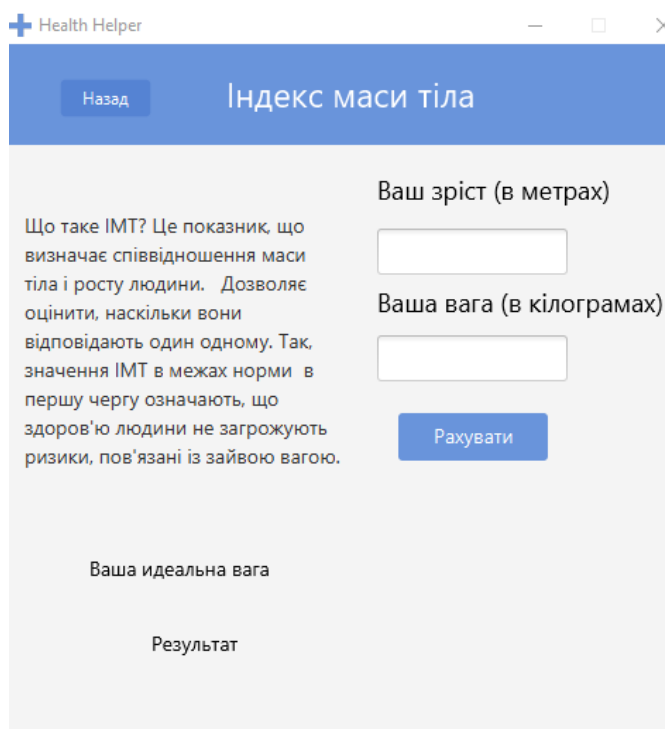
Результат 6.4 Оцінка : Задовільно

Відправити результат

Рис. 4.5, Розділ «Проба Руф'є»

Індекс маси тіла. Індекс маси тіла (ІМТ, англ. BMI, body mass index) – вкладка, що дозволяє оцінити ступінь відповідності маси людини та її зросту, й тим самим оцінити чи є маса недостатньою, нормальною, надмірною (ожирінням).

Показник індексу маси тіла було розроблено бельгійським соціологом і статистиком Адольфом Кетеле (Adolphe Quetelet) 1869 року. За значенням індексу маси популяцію поділяють на кілька категорій, дані про ці категорії і ризик виникнення хронічних та інфекційних захворювань у кожній з них наведені в додатках, табл. 1.



The screenshot shows a web application window titled 'Health Helper' with a blue header bar. Inside the header, there is a 'Назад' (Back) button and the title 'Індекс маси тіла' (Body Mass Index). The main content area is divided into two columns. The left column contains a text block explaining BMI: 'Що таке ІМТ? Це показник, що визначає співвідношення маси тіла і росту людини. Дозволяє оцінити, наскільки вони відповідають один одному. Так, значення ІМТ в межах норми в першу чергу означають, що здоров'ю людини не загрожують ризики, пов'язані із зайвою вагою.' The right column contains two input fields: 'Ваш зріст (в метрах)' (Your height in meters) and 'Ваша вага (в кілограмах)' (Your weight in kilograms), followed by a blue 'Рахувати' (Calculate) button. Below the input fields, there are labels for 'Ваша ідеальна вага' (Your ideal weight) and 'Результат' (Result).

Рис. 4.6, Розділ «ІМТ»

## РОЗДІЛ 5 ТЕСТУВАННЯ

- Для тестування було використано консольні маркери
- Протестовано безпосередньо класи UI
- Протестовано безпосередньо класи бізнес-логіки
- Програмний додаток був протестований сімейними лікарями в умовах комунального некомерційного підприємства Дніпропетровський центр первинної медико-санітарної допомоги №11, що було підтверджено відповідною довідкою (Додаток. Довідка про підтвердження тестування програмного додатку). Були зроблені рекомендації щодо подальшого розвитку додатка. В подальшому планується втілення програмного додатку в сучасні українські лікарні та співпрацю з МОЗ України.



## ВИСНОВКИ

- На основі аналізу існуючого програмного забезпечення, що здійснює взаємодію лікаря і хворого, опитування лікарів зробили систематизацію і узагальнення деяких медичних показників, виявили вимоги, яким має задовольняти програмний додаток.
- Вперше створена платформи для поєднання пацієнта з лікарем первинної ланки сімейної медицини онлайн.
- Розроблений програмний додаток «Health Helper», який має допомогти здійснювати спілкування хворого і лікаря, швидко передати лікарю зміни в стані пацієнта.
- Додаток має мінімальні вимоги до апаратного забезпечення.
- Окреслені напрямки подальшого вдосконалення програмного додатку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. І.С. Зозуля Український медичний часопис науково-практичний медичний журнал ТОВ «МОПІОН» - 2018.  
Дата звернення 28.10.20
2. [helsi.me/reform](https://helsi.me/reform)  
Дата звернення 05.11.20
3. [www.wikipedia.org](https://www.wikipedia.org)  
Дата звернення 11.11.20
4. [developers.google.com/gmail](https://developers.google.com/gmail)  
Дата звернення 19.11.20
5. [www.kmu.gov.ua/ua/diyalnist/reformi/reforma-sistemi-ohoroni-zdorovya](https://www.kmu.gov.ua/ua/diyalnist/reformi/reforma-sistemi-ohoroni-zdorovya)  
Дата звернення 29.11.20

## ДОДАТКИ

### ДОДАТОК А Формати файлів

- .java
- .fxml
- .properties
- .png
- .ico
- .s3db
- .jar

## ДОДАТОК Б Фрагменти коду

```
public class User implements Serializable {
    public User(int id, Data data, String date) {
        this.id = id;
        this.data = data;
        this.date = date;
    }

    private int id;
    private Data data;
    private String date;

    public Data getData() {
        return data;
    }

    public void setData(Data data) {
        this.data = data;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

class Data {
    Data(float temperature, float glucose, int pulse, int rate,
String pressure){
        this.temperature = temperature;
        this.glucose = glucose;
        this.pulse = pulse;
        this.rate = rate;
        this.pressure = pressure;
    }
}
```

```
private float temperature;
private float glucose;
private int pulse;
private int rate;
private String pressure;

public float getTemperature() {
    return temperature;
}

public void setTemperature(float temperature) {
    this.temperature = temperature;
}

public float getGlucose() {
    return glucose;
}

public void setGlucose(float glucose) {
    this.glucose = glucose;
}

public int getPulse() {
    return pulse;
}

public void setPulse(int pulse) {
    this.pulse = pulse;
}

public int getRate() {
    return rate;
}

public void setRate(int rate) {
    this.rate = rate;
}

public String getPressure() {
    return pressure;
}

public void setPressure(String pressure) {
    this.pressure = pressure;
}
}
```

## ДОДАТОК В Таблиці

Таблиця 1, Категорії і ризик виникнення хронічних та інфекційних захворювань

Класифікація маси тіла у дорослих і частота виникнення хронічних неінфекційних захворювань <sup>[1]</sup>				
Класифікація	ІМТ, кг/м <sup>2</sup>	Ймовірність розвитку захворювань		
		Серцево-судинних	Бронхо-легеневих	Ендокринних
Недостатня маса	менше 18,5	Низька	Збільшена	Низька
Норма	18,5—24,9	Низька	Низька	Низька
Надлишкова маса	більше 25,0			
Передожиріння (гладкість)	25,0—29,9	Середня	Низька	Низька
Ожиріння I ступеня	30,0—34,9	Збільшена	Низька	Середня
Ожиріння II ступеня	35,0—39,9	Значно збільшена	Можливо, збільшена	Збільшена
Ожиріння III ступеня	більше 40,0	Істотно збільшена	Збільшена	Значно або істотно збільшена