Міністерство освіти і науки України

Донецький національний університет імені Василя Стуса

Фізико-технічний факультет

Кафедра комп’ютерних технологій

КУРСОВА РОБОТА

на тему: Десктопний додаток «Baymax»

з курсу «Організація баз даних та знань»

Студентки 3 курсу, групи Б16\_д/122В

спеціальності: 6.05.01.01 Комп’ютерні науки

Боднажевської Б.П.

Керівник: к. т. н., доцент кафедри КТ Єпік М.О.

Національна шкала \_\_\_\_\_\_\_\_

Кількість балів\_\_\_\_\_\_\_\_

Оцінка ECTS\_\_\_\_ \_\_\_\_

м. Вінниця – 2019 рік

**Лист оцінок курсової роботи студента 3-го курсу** Боднажевської Б. П.

|  |  |  |  |
| --- | --- | --- | --- |
| КРИТЕРІЇ ОЦІНКИ | макс.  оцінка | оцінка керівн. | оцінка комісії |
| **оцінка етапу підготовки роботи** | **6 б.** | **6 б.** |  |
| дотримання графіка консультацій з керівником (+ пошта) | 2 | 2 |  |
| дотримання термінів практичної реалізації та підготовки звіту | 2 | 2 |  |
| самостійність і ініціативність | 2 | 2 |  |
| **оцінка за звіт - відповідність вимогам до звіту:** | **60 б.:** | **60 б.:** |  |
| ТИТУЛЬНИЙ ЛИСТ | 0.5 | 0.5 |  |
| ЛИСТ ОЦІНОК | 0.5 | 0.5 |  |
| ЛИСТ ЗАВДАННЯ | 0.5 | 0.5 |  |
| РЕФЕРАТ | 1 | 1 |  |
| ЛИСТ ЗМІСТ | 0.5 | 0.5 |  |
| ВСТУП | 1 | 1 |  |
| ПОСТАНОВКА ЗАДАЧІ | 2 | 2 |  |
| ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАДАЧІ | 2 | 2 |  |
| ПРОЕКТУВАННЯ ДОДАТКУ «*назва роботи*» (опис моделей вирішення задачі - концептуальної, логічної, фізичної) | ***9***  (3/3/3) | ***9***  (3/3/3) |  |
| ПРОЕКТУВАННЯ БАЗИ ДАНИХ «назва бази даних» (**обов'язковий навч. елемент** ‑ опис моделей БД - концептуальної, логічної, фізичної) | ***15***  (5/5/5) | ***15***  (5/5/5) |  |
| **ОПИС РЕАЛІЗАЦІЇ РІШЕННЯ** | ***15 б.*** | ***15 б.*** |  |
| Опис засобів реалізації додатку (використані платформи, фреймворки, бібліотеки та ін.) | 2 | 2 |  |
| Опис даних додатку (вхідних, вихідних, проміжних) та їх представлення в додатку | 1 | 1 |  |
| Реалізація роботи з базою даних (**обов'язковий навч. елемент)** | 6 | 6 |  |
| Опис можливостей для еволюції додатку; опис засобів адміністрування; опис інтерфейсу користувача (*залежно від спеціалізації*) | 1 | 1 |  |
| Опис контрольних прикладів | 3 | 3 |  |
| Опис вимог до ПЗ та АЗ | 1 | 1 |  |
| Опис інструкцій користувачеві | 1 | 1 |  |
| ВИСНОВКИ | 1 | 1 |  |
| СПИСОК ПОСИЛАНЬ | 1 | 1 |  |
| ДОДАТКИ | 1 | 1 |  |
| обсяг роботи(не менше 30 і не більше 50) | 2 | 2 |  |
| оформлення(тексту, малюнків, таблиць, формул, додатків, посилань) | 8 | 8 |  |
| **оцінка за доповідь** | **14 б.** | **14 б.** |  |
| структура доповіді, оформлення слайдів | 5 | 5 |  |
| знання матеріалу роботи і рівень відповіді на питання | 5 | 5 |  |
| дотримання регламенту, активність | 4 | 4 |  |
| **оцінка за практичну реалізацію** | **20 б.** | **20 б.** |  |
| працююча програма (наявність проекту у керівника) | 7 | 7 |  |
| відповідність практичної реалізації постановці задачі | 7 | 7 |  |
| наочність процесу демонстрації *(наявність презентації або демо-версії програми)* | 6 | 6 |  |
| **ВСЬОГО** | **100 б.** | **100 б.** |  |

|  |  |
| --- | --- |
| Керівник: Комісія: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (П.І.Б. підпис) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (П.І.Б. підпис) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (П.І.Б. підпис) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (П.І.Б. підпис) |

Міністерство освіти і науки України

Донецький національний університет імені Василя Стуса

Фізико-технічний факультет

Кафедра комп`ютерних технологій

Затверджую

в.о. завідувача кафедри КТ,

к.т.н. Меркулова К.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“ \_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

Завдання

|  |  |
| --- | --- |
| На курсову роботу студента (ки) 3-го курсу бакалаврату | Боднажевської Богдани Павлівни |
| Тема курсової роботи | Десктопний додаток «Baymax» |
| Постановка задачі в короткій формі | Виконати огляд предметної області та описати розширене технічне завдання.  Вибрати засоби і методи створення настільного додатку та програмно реалізувати їх.  Спроектувати БД  Розробити програмний засіб з графічним інтерфейсом на основі вже спроектованої БД  Провести комп’ютерні експерименти. Підготувати технічну документацію. |
| Рекомендовані дані (2-3 адреси сайтів, матеріали яких рекомендує керівник курсової роботи) | 1. Організація баз даних і знань[Електронний ресурс]. Режим доступу –   <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxjeXJpbGthZG9tc2t5fGd4OjQ3MDY3ZjA0ZWE0ZmE3NzQ>   1. Java-Програмування [Електронний ресурс]. Режим доступу –   <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxjeXJpbGthZG9tc2t5fGd4OjQ3MDY3ZjA0ZWE0ZmE3NzQ> |

|  |  |  |
| --- | --- | --- |
| Дата отримання  версії звіту керівником | Етап виконання роботи | Виконання (зазначає керівник) |
| 04.03.2019 | Проектування програми |  |
| 29.03.2019 | Реалізація програми |  |
| 30.04.2019 | Підготовка та редагування звіту за зауваженнями керівника (за змістом) |  |
| 03.05.2019 | Редагування звіту (з оформлення) |  |

Дата видачі завдання \_\_\_04.03.2019\_\_\_

Керівник (П.І.Б) \_\_ Єпік М.О.\_\_

РЕФЕРАТ

Тема курсової роботи – «Десктопний додаток «Baymax»

Предметна область – контроль візитів та діагнозів.

Мета роботи – спроектувати базу даних у предметній області «контроль візитів та діагнозів», на основі якої розробити зручний у користувані десктопний додаток для оптимізації контролю за здоров’ям.

Об’єкт вивчення – систематизування медичного журналу користувача, враховуючи особисті дані та статистику цього ж журналу за якийсь період часу.

Предмет вивчення – База Даних MYSQL.

Методи дослідження:

* Аналіз – накопичення, вивчення та аналіз специфіки: професії лікар, роботи лікарні, діагнозу.
* Синтез – на основі отриманих даних(зазначених вище) створення та заповнення БД з урахуванням специфіки професії лікаря(можливість наявності декількох), співвідношенням між професійною специфікою лікаря та сферою обслуговування певної лікарні, виявленням зв’язку між лікарем та діагнозом, враховуючи особливості класифікації захворювання та специфікою роботи лікаря.
* Моделювання – враховуючи всі особливості даної області дослідження розподілення накопичених даних між основними сутностями (лікар, лікарня та діагноз) та реалізація за допомогою об’єктів.

Необхідне обладнання – ПК, IntelliJ IDEA, MySQL Workbench.

Короткий опис функцій програми – створення особистої картки користувача у якій будуть розміщені особисті дані (вік, ріст, вага, тощо); створення візиту, основними характеристика якого є – дата, час, лікар та лікарня; додавання, призначеного лікарем, діагнозу та його лікування.

Розвиток об’єкту дослідження – для подальшого розвитку додатку в даній області є можливість аналіз специфіки певного препарату, його сумісність із поставленим діагнозом, а також із професійної специфікою лікаря.

ключові слова: База Даних, Додаток, Моделювання, Предметна область, Проектування, Діаграма, Система, Функціонал.

ЗМІСТ

[ВСТУП 12](#_Toc8597825)

[1. ПОСТАНОВКА ЗАДАЧІ 13](#_Toc8597826)

[1.1 Назва продукту 13](#_Toc8597827)

[1.2 Призначення і предметна область 13](#_Toc8597828)

[1.3 Цільова аудиторія 13](#_Toc8597829)

[1.4 Вимоги до функціональних характеристик 13](#_Toc8597830)

[1.5 Вимоги до інформаційних структур та методів розв’язання 13](#_Toc8597831)

[1.6 Вимоги до програмних засобів, які використовуються програмою 14](#_Toc8597832)

[1.7 Графічний інтерфейс користувача 14](#_Toc8597833)

[2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАДАЧІ 16](#_Toc8597834)

[2.1 Загальний опис предметної області 16](#_Toc8597835)

[2.1 Огляд підходів до вирішення задач 16](#_Toc8597836)

[3 ПРОЕКТУВАННЯ ДОДАТКУ «BAYMAX» 18](#_Toc8597837)

[3.1 Опис концептуальної моделі задачі 18](#_Toc8597839)

[3.2 Опис логічної моделі задачі 19](#_Toc8597850)

[3.3 Опис фізичної моделі задачі 21](#_Toc8597893)

[4 ПРОЕКТУВАННЯ БАЗИ ДАНИХ 22](#_Toc8597896)

[4.1 Опис концептуальної моделі 22](#_Toc8597897)

[4.1.1 Вимоги до БД з боку користувачів 22](#_Toc8597898)

[4.1.2 Визначення сутностей предметної області та їх аналіз 22](#_Toc8597899)

[4.1.3 Опис моделі проектованої системи у вигляді діаграми потоків даних 23](#_Toc8597900)

[4.1.4 Опис моделі даних у вигляді ER-діаграми 23](#_Toc8597901)

[4.1.5 Опис обмежень 24](#_Toc8597903)

[4.2 Опис логічної моделі 26](#_Toc8597907)

[4.2.1 Опис набору відношень 27](#_Toc8597908)

[4.2.2 Уточнена ER-діаграма 27](#_Toc8597909)

[4.2.3 Нормалізація до 3НФ 28](#_Toc8597910)

[4.3 Опис фізичної моделі 29](#_Toc8597911)

[4.3.1 Вибір СУБД 30](#_Toc8597912)

[4.3.2 Опис таблиць 30](#_Toc8597913)

[4.3.3 Опис типових SQL-запитів 35](#_Toc8597914)

[4.3.4 Обсяг пам’яті 35](#_Toc8597915)

[4.3.5 Налаштування сервера 35](#_Toc8597916)

[5 ОПИС РЕАЛІЗАЦІЇ РІШЕННЯ 37](#_Toc8597917)

[5.1 Опис засобів реалізації додатку 37](#_Toc8597918)

[5.2 Опис даних в додатку 38](#_Toc8597919)

[5.2.1 Вхідні дані 38](#_Toc8597920)

[5.2.2 Вихідні дані 38](#_Toc8597921)

[5.2.3 Проміжні дані 38](#_Toc8597922)

[5.3 Реалізація роботи з БД 39](#_Toc8597923)

[5.3.1 Короткий опис використаних стандартних класів, компонентів або функцій по роботі з БД 39](#_Toc8597924)

[5.3.2 Опис налаштувань підключення до БД 39](#_Toc8597925)

[5.3.3 Опис представлення даних в БД 39](#_Toc8597926)

[5.3.4 Опис виконання запитів 40](#_Toc8597927)

[5.4 Еволюція додатку 41](#_Toc8597928)

[5.5 Опис контрольних прикладів 41](#_Toc8597929)

[5.6 Опис вимог до ПЗ та АЗ 48](#_Toc8597948)

[5.7 Опис інструкції користувачеві 48](#_Toc8597949)

[ВИСНОВКИ 50](#_Toc8597950)

[СПИСОК ПОСИЛАНЬ 51](#_Toc8597951)

[ДОДАТКИ 52](#_Toc8597952)

[Додаток А 52](#_Toc8597953)

[Додаток Б 52](#_Toc8597965)

[Додаток Д 54](#_Toc8597983)

ВСТУП

В сучасному світі людство оперує безмежною кількістю інформації, яку зберігати та сортувати без допоміжних засобів просто неможливо. Тому актуальність інформаційних систем сьогодні дуже висока. Вони допомагають упорядковувати, знаходити та порівнювати певну структуру даних.

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем.

Даний додаток створений для оптимізації контролю за здоров’ям. Його основна задача – відстежування візитів, діагнозів і захворювань, також це дає можливість приймати активну участь в керуванні вашим здоров’ям та його покращенням.

Оскільки Baymax створений для особистого користування, то користувач особисто обирає інформацію, яка буде розміщена в додатку.

Потрібно розробити: середовище розробки, вибір мови програмування, дослідження предметної області, виявлення основних сутностей та зв’язків, визначення технологій, оптимізація коду.

Основні результати роботи - база даних у предметній області «Контроль візитів та діагнозів», прототип десктопного додатку для реалізації зв’язку між користувачем та створеною БД

При вирішені даної задачі буде використано декілька об’єктів(лікар, лікарня, користувач). Головним об’єктом є користувач, тому графічний інтерфейс повинен бути пристосований саме для зручного користування.

1. ПОСТАНОВКА ЗАДАЧІ
   1. Назва продукту

Десктопний додаток «Baymax»

* 1. Призначення і предметна область

Програма слугує для оптимізації контролю користувачем свого здоров’я. Предметна область «Контроль візитів і діагнозів». Основна задача додатку – відстежування візитів, діагнозів і захворювань.

* 1. Цільова аудиторія

Здоров’я важливою складовою людського життя, тому програма призначена для використання людьми різних вікових категорій. Програма передбачає початковий рівень комп'ютерної підготовки користувача.

* 1. Вимоги до функціональних характеристик

Програма забезпечує можливість виконання перерахованих нижче функцій: автентифікація, реєстрація, створення візиту, редагування візиту, сортування візитів, пошук візиту (враховуючи такі параметри: лікар, лікарня, дата візиту), видалення візиту, перегляд інформації про лікаря та лікарню, додавання до програми встановленого лікарем діагнозу та призначеного лікування, редагування лікування, видалення захворювання, сортування діагнозів, пошук діагнозів або лікування, перегляд діагнозів та візитів

* 1. Вимоги до інформаційних структур та методів розв’язання
* Простий та зрозумілий інтерфейс
* Користувач має доступ лише до необхідної інформації і функцій
* Дані зберігаються лише у БД
* Додавання, отримання, змінення та видалення даних відбувається через SQL запити
  1. Вимоги до програмних засобів, які використовуються програмою
* Мова високого рівня – Java
* Мова запитів – SQL
* Платформа для з’єднання з базою даних – JDBC
* Патерн MVC
* Патерн DAO
  1. Графічний інтерфейс користувача

Даний програмний продукт містить декілька панель:

* Головне вікно
* Автентифікації та реєстрації

Данна панель містить поля для входу в систему певного, вже зареєстрованого, користувача або ж реєстрація нового користувача додатком

* Заповнення характеристик основної інформації нового користувача

Панель може бути доступна лише при реєстрації нового користувача. Тут розміщенні поля особистої інформації та характеристика користувача, після їх заповнення данні вносяться у БД

* Основна інформація

Панель слугує для відображення інформації та характеристики користувача і містить елементи керування основними сутностями. Відображенням даної панелі керує користувач, він вирішує чи буде вона відображатись на екрані

* Редагування характеристик та основної інформації

На панелі розміщена особиста інформація та характеристика певного користувача, щоб отримати доступ до цієї панелі необхідно додатково пройти автентифікацію, це дає змогу змінювати дані безпечним чином

* Візити

Панель апелює даними, які відносяться до сутності візиту, а також дозволяє їх змінювати, видаляти, а також додавати нові

* Діагнози

Панель апелює даними, які відносяться до сутності візиту, а також дозволяє їх змінювати, видаляти, а також додавати нові

2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАДАЧІ

* 1. Загальний опис предметної області

База даних є сукупність даних на машинних носіях, які використовуються при функціонуванні системи обробки інформації, організовані по визначеним правилам, які передбачають загальні принципи описування збереження і маніпулювання ними, а також які незалежні від прикладних програм. В основі організації бази даних є модель даних, яка визначає правила, у відповідності з якими структуруються дані. За допомогою моделі представляється велика кількість даних і описуються взаємно зв’язки між ними. Найбільш поширені такі моделі даних: ієрархічна, мережева, реляційна.

Реляційна модель даних являє собою набір двомірних плоских таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив є плоскою двомірною таблицею.

Сьогодні через завантаженість люди забувають про прийом ліків чи не пам’ятають який лікар їх призначив, або ж просто не приходять на прийом через те що не можуть запам’ятати усе. Цей додаток допоможе систематизувати усі ці дані, а також буде нагадувати про ліки чи візит до лікаря.

2.1 Огляд підходів до вирішення задач

Для реалізації додатка використана логічна зв'язка MySQL + Java + JDBC + MVC + DAO.

MySQL – це система управління реляційними базами даних, яка дозволяє отримувати, додавати та обробляти дані. Будь яка база даних в MySQL надається у вигляді набору реляційних таблиць і набору між ними, а доступ до даних здійснюється за допомогою SQL-запитів.

Java – об’єктно-орієнтована мова програмування, сукупність технологій створення додатків.

JDBC – стандартний АРІ для незалежного з’єднання мови програмування Java з різними базами даних.

MVC – конструкційний шаблон проекту, у якому котролер апелює даними між об’єктами, не зберігаючи їх у зміні оптимізуючи пам’ять, оскільки це впливає на якість, безпеку і швидкість роботи додатка.

DAO – шаблон проектування, основною задачею якого є структурування коду.

3 ПРОЕКТУВАННЯ ДОДАТКУ «BAYMAX»

На даному етапі потрібно сформулювати основні задачі, які має виконувати додаток.

3.1 Опис концептуальної моделі задачі

Мета роботи додатка – систематизувати медичний журнал.

Користувач має доступ до такого функціоналу:

Автентифікація, реєстрація, індивідуальний логін, пароль, особисті дані (номер, еmail, ім’я, прізвище, по батькові), характеристика (ріс, вага, вік), створення візиту (дата, час, лікарня, лікар, коментар (за бажанням)), редагування візиту, пошук візиту (враховуючи такі параметри: лікар, лікарня, дата), видалення візиту, перегляд інформації про лікаря та лікарню, встановлення діагнозу (з наступними характеристиками: лікар, дата, лікарня, план лікування (медичний засіб, дозування, період прийому)), перегляд історії візитів та діагнозів.

Але оскільки усі дані взаємодіють, то простий користувач не має доступу певного функціоналу:

* Реєстрація лікаря
* Реєстрація лікарні
* Змінення даних про лікаря чи лікарню

Дане рішення було прийнято для полегшення користувачеві роботи із додатком, а також для більш точної журналізації даних, оскільки користувач маю такий функціонал міг би зареєструвати одного лікаря двічі.

Для вирішення поставленого завдання будується програмний додаток, який буде працювати із даними, які збережені у базі даних MySQL та даними, які вводить користувач. Ці дані будуть зберігатись також у базі даних MySQL. База даних буде попередньо спроектована та частково наповнена інформацією про клієнтів, продукцію та замовлення.

Для кращого розуміння функціональності додатку з боку користувача наведено «use case» діаграму на наступному рисунку (рисунок 1).

Зображення, що містить текст, карта

Автоматично згенерований опис

Рисунок – діаграма «use case»

3.2 Опис логічної моделі задачі

Основною задачею даного додатку є відстежування візитів, діагнозів і захворювань.

Для розробки додатка «Baymax» потрібно створити базу даних в предметній області «Контроль візитів і діагнозів». При проектуванні бази даних використовується мова запитів MySQL, а реалізація програми з використанням мови високого рівня – Java.

Основним об’єктом даного додатку є користувач. Тому візит, діагноз і лікування пов’язані і є залежними. Для оптимізації пам’яті при видалені певного користувача видаляються також пов’язані із ним об’єкти.

Також в даному додатку існують об’єкти, на які зміна чи видалення певного користувача не впливає : лікар і лікарня. Це є наслідком того, що дані об’єкти доступні усім користувачам.

Програма складається з єдиного модуля, що містять в собі пакети з класами:

* Basic – в даному пакеті містяться 2 основних класи
* Controller – головний контролер, який запускає додаток.
* Graphic – вікно додатку, у яке динамічно загружаються панелі.
* User – в даному пакеті розміщені класи, які відповідають за об’єкт «user»
* Autentification
* Controller\_user
* Create\_user
* Database\_autentification
* Database\_registration
* Database\_user
* Menu
* User
* Visit – в даному об’єкті розміщені класи, які відповідають за реалізацію об’єкта «visit»
* Controller\_visit
* Create\_visit
* Create\_visit\_grafic
* Data\_procesing
* Database\_create\_visit
* Database\_visit
* Delete\_visit\_tatabase
* Edit\_visit
* Edit\_visit\_database
* Edit\_visit\_grafic
* Information\_database
* Search\_visit
* Show\_all\_visit\_grafic
* Show\_one\_visit
* Visit
* Visit\_grafic
* Desiase - в даному об’єкті розміщені класи, які відповідають за реалізацію об’єкта «desiase»
* Controller\_desiase
* Database\_desiase
* Desiase
* Desiase\_grafic
* Plan
* Show\_all\_desiase
* Plan

3.3 Опис фізичної моделі задачі

Оскільки даний програмний продукт розроблений для особистого користування і користувач сам обирає обсяг інформації, яка буде зберігатись у додатку, то 80% даних у БД вносить користувач.

Для зручного використання програми користувач при перегляді даних отримує дані, які з’єднуються складними SQL-запитами(inner join). Додаток розроблений для взаємодії з користувачем тому усі вхідні дані користувач може переглядати, тому вони також є і вихідними. Орім даних, які вносить користувач є також дані, які користувач може лише переглядати. Цими даними є об’єкти: лікар і лікарня. Таке рішення було прийняте для зручності і легкості у користуванні.

4 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

4.1 Опис концептуальної моделі

Проектування бази даних полягає в побудові комплексу взаємозв'язаних моделей даних. Концептуальне моделювання дозволяє врахувати логічне уявлення структури даних у базі даних. Вірно розроблена модель бази даних має підтримувати усі явлення користувачів. Концептуальне моделювання є основою подальшого проектування бази даних та додатку для її обробки.

4.1.1 Вимоги до БД з боку користувачів

* Можливість зберігання великого обсягу даних
* Можливість входу в додаток за особистим логіном і паролем

4.1.2 Визначення сутностей предметної області та їх аналіз

База даних складається з 5 сутностей: користувач, візит, лікар, захворювання, план лікування.

Користувач ідентифікується по унікальному логіну і паролю для доступу до інформації та особистих даних.

Візит, окрім основної інформації, має логічний зв’язок із сутністю лікар. Цей зв’язок є необхідним для структурованості даних і зручності користувача.

Захворювання пов’язане із лікарем оскільки саме від визначає діагноз тому цей зв’язок є необхідним

План лікування містить в собі інформацію про певний лікувальний засіб, особливості та період його прийому. І оскільки зв’язок між планом і захворюванням н до 1 то логічний зв’язок між планом і лікарем є необхідним.

Лікар містить необхідну для користувача інформацію, а також його кваліфікацію .

4.1.3 Опис моделі проектованої системи у вигляді діаграми потоків даних

Опис моделі проектованої системи у вигляді діаграми потоків без декомпозиції поданий на наступному рисунку (рисунок 2).

Зображення, що містить текст, карта

Автоматично згенерований опис

Рисунок 2 – Діаграма потоків

4.1.4 Опис моделі даних у вигляді ER-діаграми

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок – ER-діаграма

На рисунку (рисунок 3) показана модель даних у вигляді ER-діаграми

4.1.5 Опис обмежень

До бази даних є ряд обмежень:

* У користувача може бути лише одна характеристика та інформація.
* За один візит користувач може піти лишу у одну лікарню.
* Визначити діагноз може лише один лікар.

У даній БД обмеження унікальності мають такі поля:

* user\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що user\_id – це унікальний особистий id користувача, саме через id з’єднуються такі дані: особиста інформація, характеристика, візит та діагноз.

* u\_login

тип даних – Varchar; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що u\_login – це поле за допомогою якого визначається id для доступу до інших даних.

* information\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що information\_id – це поле за допомогою якого визначається набір даних, а саме особистої інформації.

* i\_user\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що на одного користувача можливий тільки один запис.

* ch\_user\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що на одного користувача можливий тільки один запис.

* classification\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що classification\_id – це поле за допомогою якого визначається набір даних, а саме класифікацію доктора.

* visit\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що visit\_id – це поле за допомогою якого визначається набір даних, а саме дані про візит.

* clinick\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що clinick\_id – це поле за допомогою якого визначається набір даних, а саме дані про клінику.

* doctor\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що doctor\_id – це поле за допомогою якого визначається набір даних, а саме дані про доктора.

* dessiase\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що dessiase\_id – це поле за допомогою якого визначається набір даних, а саме дані про діагноз.

* treatment\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що treatment\_id – це поле за допомогою якого визначається набір даних, а саме дані про лікування.

* plan\_id

тип даних – Bigint; мінімальна довжина символів – 1; максимальна довжина символів – 20; причина унікальності даного поля полягає в тому, що plan\_id – це поле за допомогою якого визначається набір даних, а саме дані про лікування.

4.2 Опис логічної моделі

Логічне проектування бази даних – це процес перетворення концептуальної моделі в логічну модель з урахуванням особливостей обраної СУБД. Основним завданням логічного проектування є розробка логічної схеми, орієнтованої на вибрану СУБД.

4.2.1 Опис набору відношень

Первинним ключем у всіх сутностях буде виступати їх номер – id.

Так як за вимогами реляційної моделі не можна використовувати на пряму зв’язки «багато до багатьох», то між такими таблицями потрібно створити, так звані, проміжні таблиці. Такі таблиці будуть мати у собі посилання на таблиці, між якими зв’язок «багато до багатьох» і таким чином зв’язок між проміжною таблицею та пов’язаною із нею буде «один до багатьох».

У сутності «Користувач» ключем виступає його id. Користувач може мати лише один лист характеристик та один – інформаціїї, виходячи з цього необхідно додати зовнішні ключі: на таблицю «characteristic» - c\_user\_id, на таблицю «information» - i\_user\_id.

У візиті може бути зареєстровано декілька лікарів, тому необхіно створити проміжну таблицю із посиланями таблиці «visit» та «doctor» - «visit\_doctor».

У одній лікарні може працювати декілька лікарів, а також один лікар може працювати у декількох лікарнях. Виходячи з цього потрібно створити проміжну таблицю «clinic\_doctor» з посиланням на таблиці «clinic» та «doctor».

У одного лікаря може бути декілька професійних кваліфікацій, тому необхідно створити проміжну таблицю «doctor\_classification», яка поєднує таблиці «doctor» та «classification».

4.2.2 Уточнена ER-діаграма

ER-діаграма показана на рисунку (рисунок 3)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 3 - ER-діаграма

4.2.3 Нормалізація до 3НФ

Зображення, що містить знімок екрана

Автоматично згенерований описНа рисунку (рисунок 4) показана БД

Рисунок 4 -База Даних

База даних, яка показана на рисунку (рисунок 5) не нормалізована до НФ3, так як має в собі зв’язки багато до багатьох.

Саме тому була проведена нормалізація.

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 5 – Нормалізована БД

Після нормалізації БД має такий вигляд (рисунок 6) та складається з 14 таблиць. Так як в один візит додається одна лікарня, та є можливість за один візит піти до декількох лікарів була створена проміжна таблиця visit\_doctor.

Один лікар може працювати в декількох лікарнях та мати декілька специфікацій професії, тому додані проміжні таблиці clinic\_doctor та doctor\_classification.

За таким же принципом додана таблиця treatment\_plan.

4.3 Опис фізичної моделі

4.3.1 Вибір СУБД

Для обробки даних розробленої бази даних необхідно розробити додаток у MYSL.

Плюси:

* декларативність
* простота коду запитів
* наявність стандарту

4.3.2 Опис таблиць

Умовно таблиці, що складають всю БД можна поділити на два типи: таблиці-словники, та таблиці з інформацією, що може часто змінюватися, або доповнюватись. Таблиці-словники найчастіше містять в собі статичну інформацію, яка не змінюється протягом довгого періоду часу. Найпростішим прикладом такої таблиці в даній базі даних будуть співвідношення «ключ – розшифрування».

4.3.2.1 Таблиця «user»

Дана таблиця відображає одну з основних сутностей. У ній зберігаються усі користувачі додатка.(рисунок 6). Логін перевіряється на унікальність.

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 6- таблиця «users»

4.3.2.2 Таблиця «characteristic»

Дана таблиця доповнює одну з основних сутностей, а саме «user» і містить характеристику кожного з користувачів(рисунок 7)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 7 – таблиця «срфкфсеукшіешс»

4.3.2.3Таблиця «information»

Дана таблиця доповнює одну з основних сутностей, а саме «user» і містить характеристику кожного з користувачів(рисунок 8)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 8 – таблиця «information»

4.3.2.4 Таблиця «clinic»

Дана таблиця містить дані про кожну з лікарень, які внесені до бази даних. Користаувач не має доступу до редагування, додавання чи видалення цієї інформації, натомість він може її переглядати та з’єднувати з візитом(рисунок 9)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 9 – таблиця «clinic»

4.3.2.5 Таблиця «doctor»

Дана таблиця містить дані про лікарів, які внесені до БД. Користувач не має доступу до редагування, додавання чи видалення цієї інформації, натомість він може її переглядати та з’єднувати з візитом, діагнозом чи планом лікування.(рисунок 10)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 10 – таблиця «doctor»

4.3.2.6 Таблиця «clssification»

Дана таблиця доповнює одну з основних сутностей, а саме «лікар». В ній міститься класифікація лікарів. Користувач може бачити цю інформацію при перегляді об’єкту «лікар».(рисунок 11)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 41 – таблиця «classification»

4.3.2.7 Таблиця «desiase»

Дана таблиця відображає одну з основних сутностей, а саме «діагноз».(рисунок 12)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 12 – таблиця «desiase»

4.3.2.8 Таблиця «treatment»

Дана таблиця доповнює одну з основних сутностей, а саме «діагноз». Саме тут зберігаються дані, які визначають який лікар призначив лікування(рисунок 13)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 53 – таблиця «treatment»

4.3.2.9 Таблиця «plan»

Дана таблиця відображає одну з основних сутностей, а саме «план лікування». Вона має зв’язок із такими таблицею «treatment» і при видалені даних із таблиці «treatment» дані у таблиці «plan» будуть відповідно видалені (рисунок 14)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 14 – таблиця «plan»

4.3.2.10 Таблиця «visit»

Дана таблиця відображає одну з основних сутностей, а саме «візит» (рисунок 15)

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 156 – таблиця «visit»

4.3.3 Опис типових SQL-запитів

* select - вибрати
* insert – внести дані
* update – редагувати
* delite – видалити запис
* drop – видалити таблицю або запис

4.3.4 Обсяг пам’яті

На даному етапі БД займає 1 МБ. Але оскільки додаток є десктопним, то для швидкої роботи необхідний об’єм пам’яті складає 2МБ.

4.3.5 Налаштування сервера

При р оботі додатку з БД використовується JDBC.

public Integer user\_id;

public String userName = "root";

public String password = "b5p639bj8";

public String connectionUrl ="jdbc:mysql://localhost:3306/baymax\_2? verifyServerCertificate=false&useSSL=false&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connection = DriverManager.getConnection(connectionUrl, userName, password)

5 ОПИС РЕАЛІЗАЦІЇ РІШЕННЯ

5.1 Опис засобів реалізації додатку

Для реалізації додатка використана логічна зв'язка MySQL + Java + JDBC + MVC + DAO.

MySQL – це система управління реляційними базами даних, яка дозволяє отримувати, додавати та обробляти дані. Будь яка база даних в MySQL надається у вигляді набору реляційних таблиць і набору між ними, а доступ до даних здійснюється за допомогою SQL-запитів.

Java – об’єктно-орієнтована мова програмування, сукупність технологій створення додатків.

JDBC – стандартний АРІ для незалежного з’єднання мови програмування Java з різними базами даних.

MVC – конструкційний шаблон проекту, у якому котролер апелює даними між об’єктами, не зберігаючи їх у зміні оптимізуючи пам’ять, оскільки це впливає на якість, безпеку і швидкість роботи додатка.

DAO – шаблон проектування, основною задачею якого є структурування коду.

До проекту підключені такі бібліотеки:

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.sql.\*;

5.2 Опис даних в додатку

5.2.1 Вхідні дані

Оскільки даний програмний продукт розроблений для особистого користування і користувач особисто обирає обсяг інформації, яка буде зберігатись у додатку, то 80% даних у БД вносить користувач.

Перелік вхідних даних та їх характеристика: логін ( varchar); пароль (varchar); ім’я (varchar); прізвище (varchar); по батькові (varchar); номер (int); імейл (varchar); вік (int); ріст (int); вага (int); дата (data); час (time); замітка (varchar); назва (varchar); опис (varchar); дата (data).

5.2.2 Вихідні дані

Для зручного використання програми користувач при перегляді даних отримує дані, які з’єднуються складними SQL-запитами(inner join). Додаток розроблений для взаємодії з користувачем тому усі вхідні дані користувач може переглядати, тому вони також є і вихідними. Орім даних, які вносить користувач є також дані, які користувач може лише переглядати. Цими даними є об’єкти: лікар і лікарня. Таке рішення було прийняте для зручності і легкості у користуванні. Перелік даних, які є лише вихідними: ПІБ лікаря (varchar); імейл лікаря (varchar); номер лікаря (int); назва лікарні (varchar); адреса лікарні (varchar); номер лікарні (int).

5.2.3 Проміжні дані

Так як основним шаблоном при реалізації додатку є MVC, то у додатку не використовуються проміжні дані. Оскільки контролер апелює не просто даними, а об’єктами, які після виконання ряду певних методів обновляються

5.3 Реалізація роботи з БД

5.3.1 Короткий опис використаних стандартних класів, компонентів або функцій по роботі з БД

При роботі з базою даних був використаний стандартний фреймворк JDBC.

Він є стандартним при роботі у зв’язці MYSQL + Java.

Основні сутності:

* Connection;
* Statement;
* PreparedStatement;
* CallableStatement;
* ResultSet;
* Driver;
* DatabaseMetaData

5.3.2 Опис налаштувань підключення до БД

Для підключення БД потрібно створити об’єкт Connection.

Connection connection = DriverManager.getConnection(connectionUrl, userName, password)

connectionUrl – посилання на певну базу даних

username – ім’я користувача

password – пароль

5.3.3 Опис представлення даних в БД

Даний додаток слугує для слідкування за здоров’ям користувача і тому саме він додає інформацію про себе (після реєстрації), а вже потім заповнює необхідні дані.

У програму дані з БД отримуються у вигляді списків об’єктів, які додаток запрошує через запити.

Усі дані із програми додаються в БД через відповідні їм запити у типі даних, які відповідають необхідним.

Тип DATE отримується після перетворення типів int на STRING та приведення до необхідного вигляду (використовується клас «Data\_parse»)

Для зручного використання програми користувач при перегляді даних отримує дані, які з’єднуються складними SQL-запитами(inner join). Додаток розроблений для взаємодії з користувачем тому усі вхідні дані користувач може переглядати, тому вони також є і вихідними. Орім даних, які вносить користувач є також дані, які користувач може лише переглядати. Цими даними є об’єкти: лікар і лікарня. Таке рішення було прийняте для зручності і легкості у користуванні. Перелік даних, які є лише вихідними: ПІБ лікаря (varchar); імейл лікаря (varchar); номер лікаря (int); назва лікарні (varchar); адреса лікарні (varchar); номер лікарні (int).

5.3.4 Опис виконання запитів

Нижче наведено декілька прикладів запитів:

Знайти id усіх лікарів, які працюють у певній лікарні:

select clinic\_doctor\_id from clinic\_doctor inner join clinic on clinic\_doctor.c\_d\_clinic\_id = clinic.clinic\_id where c\_name = '"+cl+"'"

Вивести інформацію про лікарню та лікаря, які були додані в даний візит:

select clinic.c\_name, doctor.d\_name, doctor.d\_surname, doctor.d\_midlename, visit.v\_data, visit.v\_time, visit.visit\_id from doctor inner join visit\_doctor on doctor.doctor\_id = visit\_doctor.v\_d\_doctor\_id inner join visit on visit\_doctor.v\_d\_visit\_id = visit.visit\_id inner join clinic on clinic.clinic\_id = visit.v\_clinick\_id where visit.v\_user\_id = 5 group by visit.v\_data desc"

Вивести всю інформацію про користувача (окрім його id, логіна та пароля):

select i\_name, i\_surname, i\_number, i\_email, ch\_age, ch\_grow, ch\_weight from information innner join users on i\_user\_id = users.user\_id inner join characteristic on characteristic.ch\_user\_id = users.user\_id where i\_user\_id = '" + user\_id + "'"

Внести новий запис в БД про користувача:

INSERT INTO information (i\_name, i\_surname, i\_number, i\_email, i\_user\_id) values (?, ?, ?, ?, ?)

В останньому із представлених вище запитів для заповнення даних безпосередньо у базу даних використовується метод PreparedStatement.

Метод preparedStatement\_inf.execute() виконує дві фукції:

* Зберігає внесені дані
* Закриває сесію

5.4 Еволюція додатку

Даний додаток містить досить обширний функціонал.

В подальшому планується його розширення:

* Розробити систему спільного доступу «Сім’я»
* Розробити чат для спілкування між пацієнтами та пацієнтом та лікарем
* Розробити систему оцінювання лікаря, яка буде додаватись до його особистої інформації
* Додати функції сповіщення про вживання ліків у певний момент
* Додати функцію сповіщення про візит за певний період часу, який визначає користувач

5.5 Опис контрольних прикладів

Оскільки основою структури даної програми є MVC, тому головне вікно є пустим, а в залежності від вибору користувача в головне вікно імпортується та видаляються потрібні панелі.

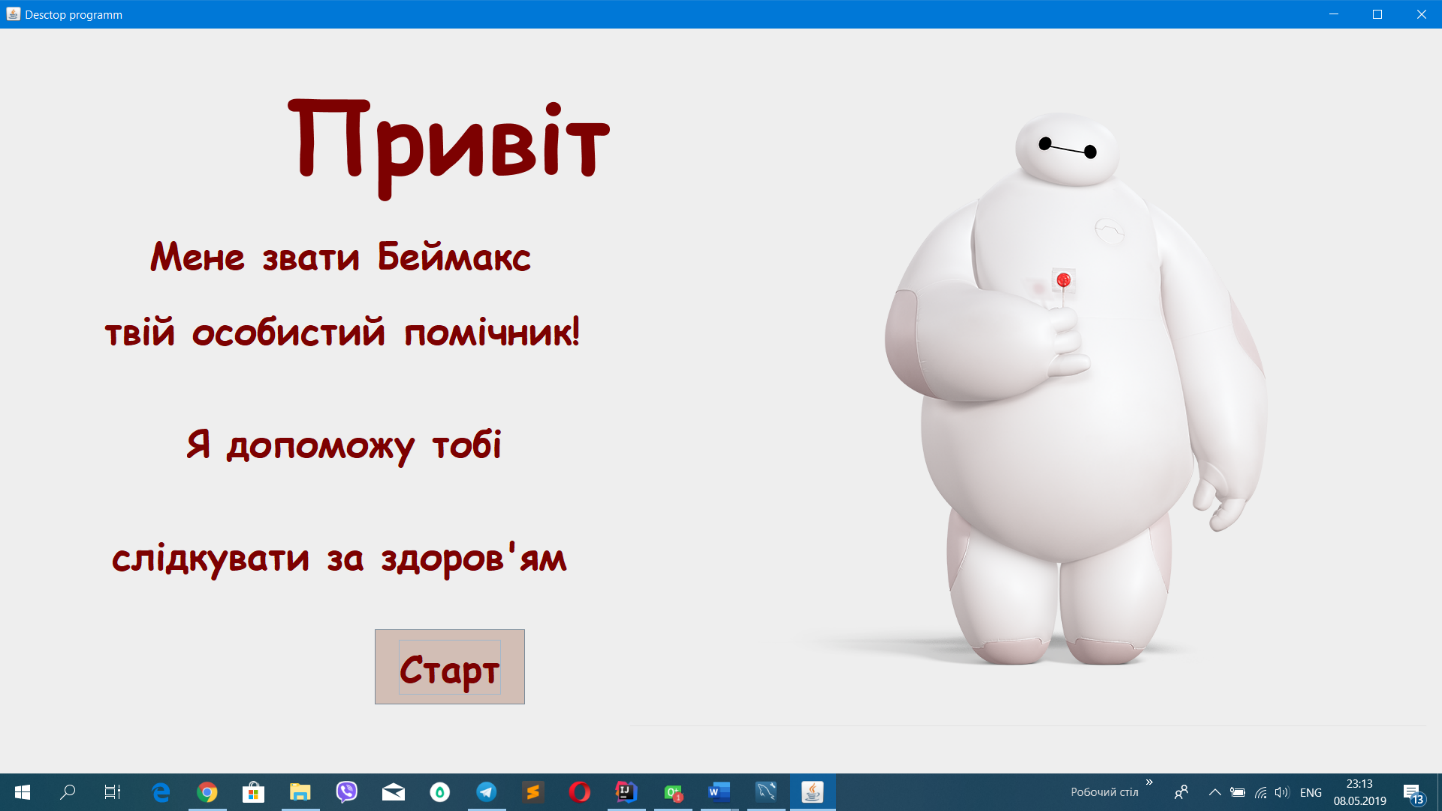
На рисунку (рисунок 16) показана панель яка загружається на початку програми.

Рисунок 16 – головне вікно

На рисунку 17 показана панель автентифікації користувача. Ідентифікатором унікальності користувача є логін.

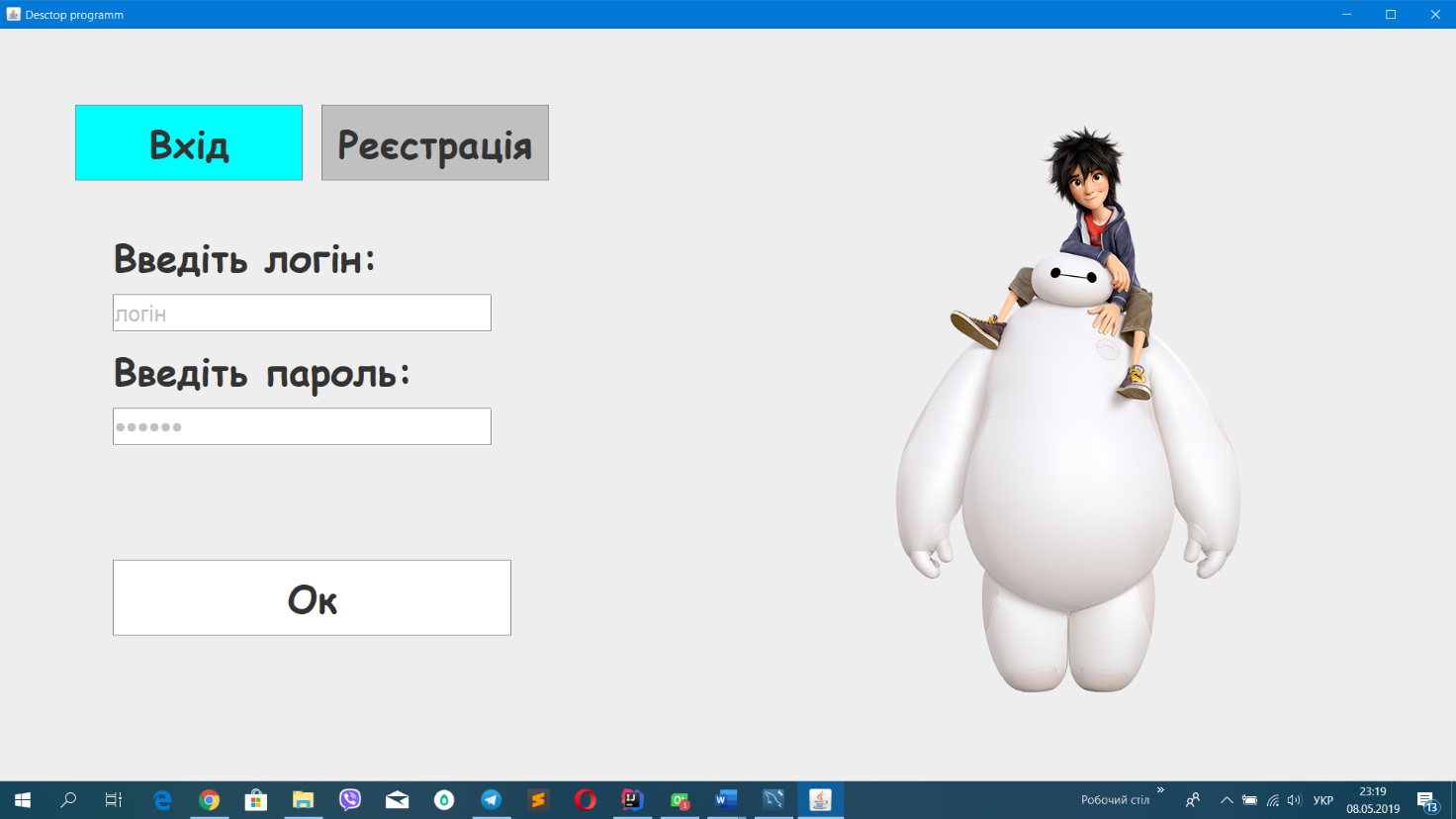


Рисунок 178 - автентифікація

На рисунку (рисунок 18) показана панель реєстрації нового користувача. Після закінчення вводу, дані обробляються і після перевірки на їх коректність заносяться в базу даних.

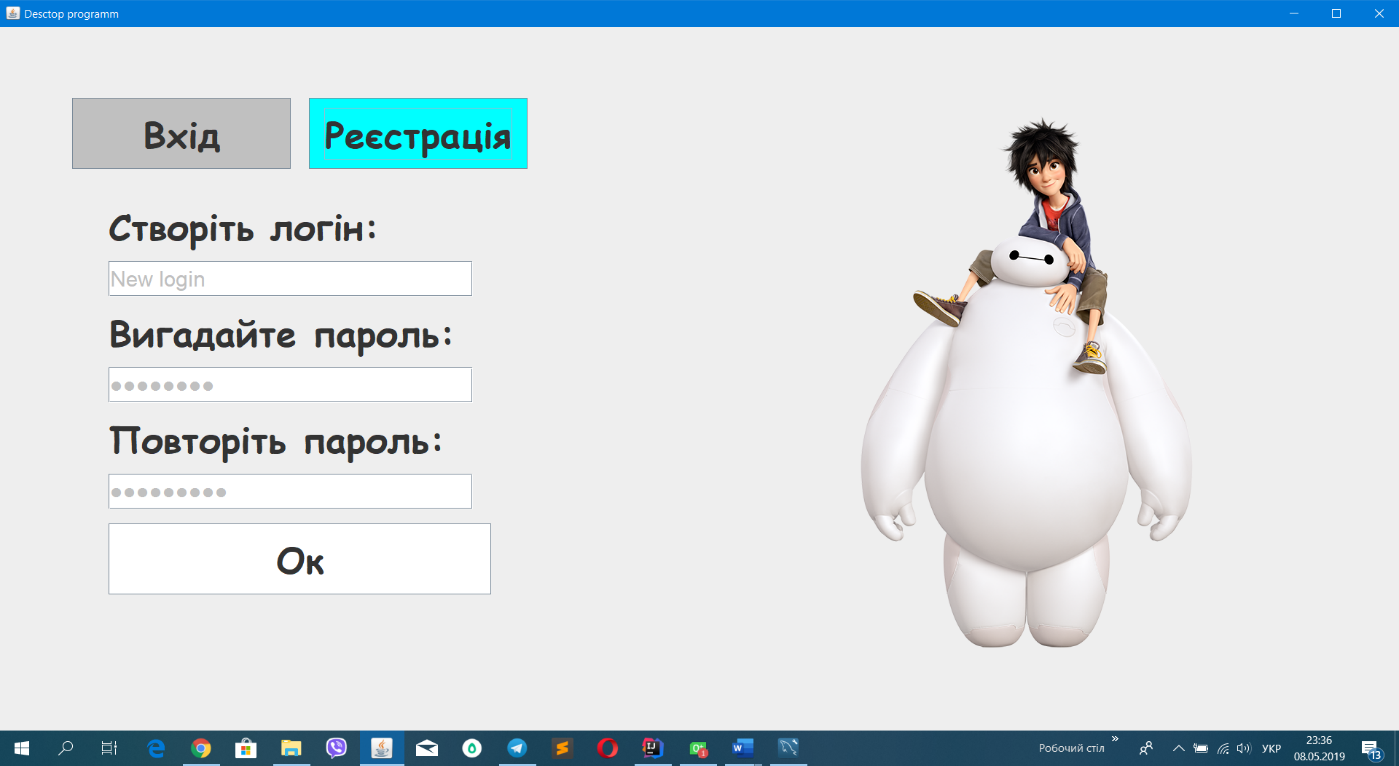


Рисунок 18 - реєстрація

На рисунках (рисунок 19 та рисунок 20) показані панелі для заповнення даних про нового користувача додатку. Між панелями є динамічний перехід, а для зручного користування дані для заповнення відомостей про користувача знімаються із полів вводу по натиску на кнопку «Готово».

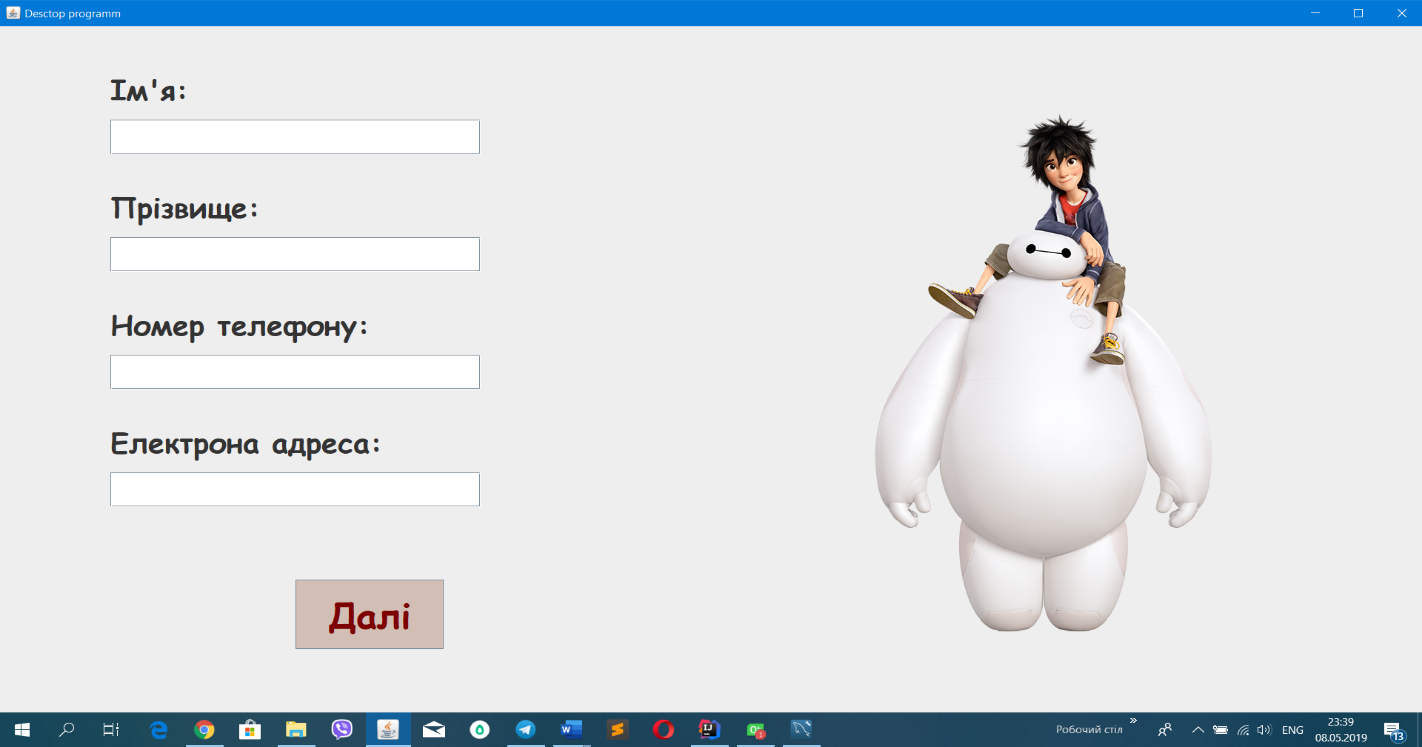
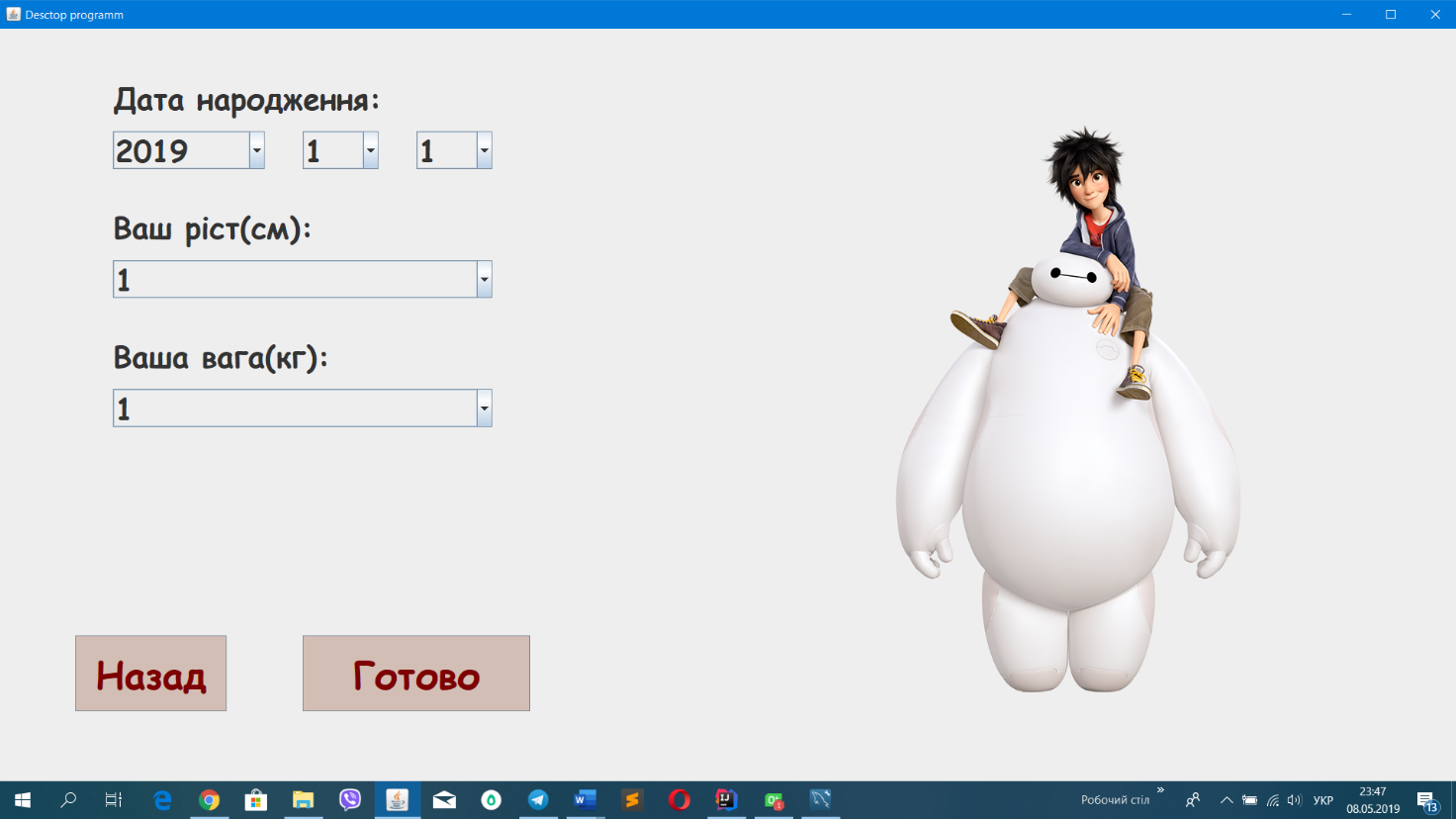


Рисунок 209 - заповнення даних 1

Рисунок 1910 – заповнення даних 1

На рисунку (рисунок 21) показана панель на якій розміщена характеристика користувача, який даний момент часу є активним користувачем. Також на даній панелі розміщені основні елементи керуванням програми. Керувати показом даної панелі можна за допомогою вертикальної кнопки чорного кольору.

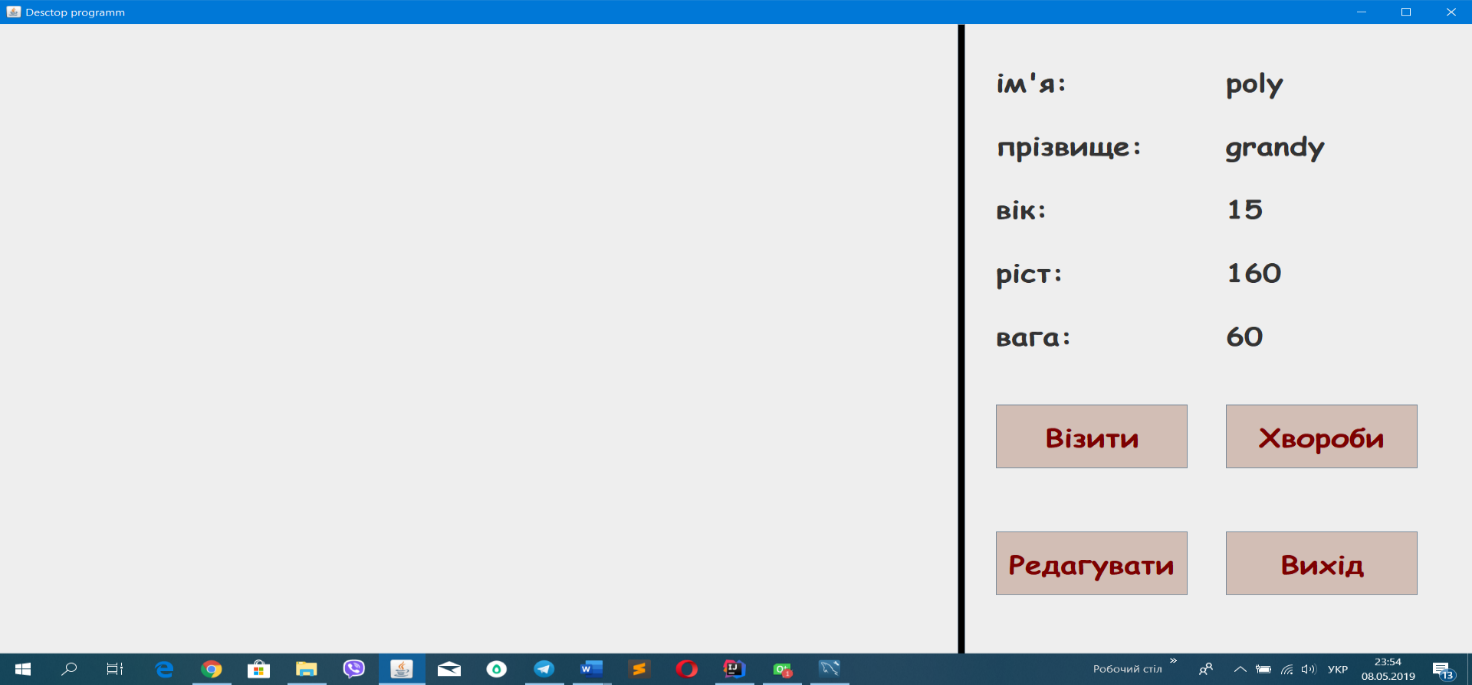
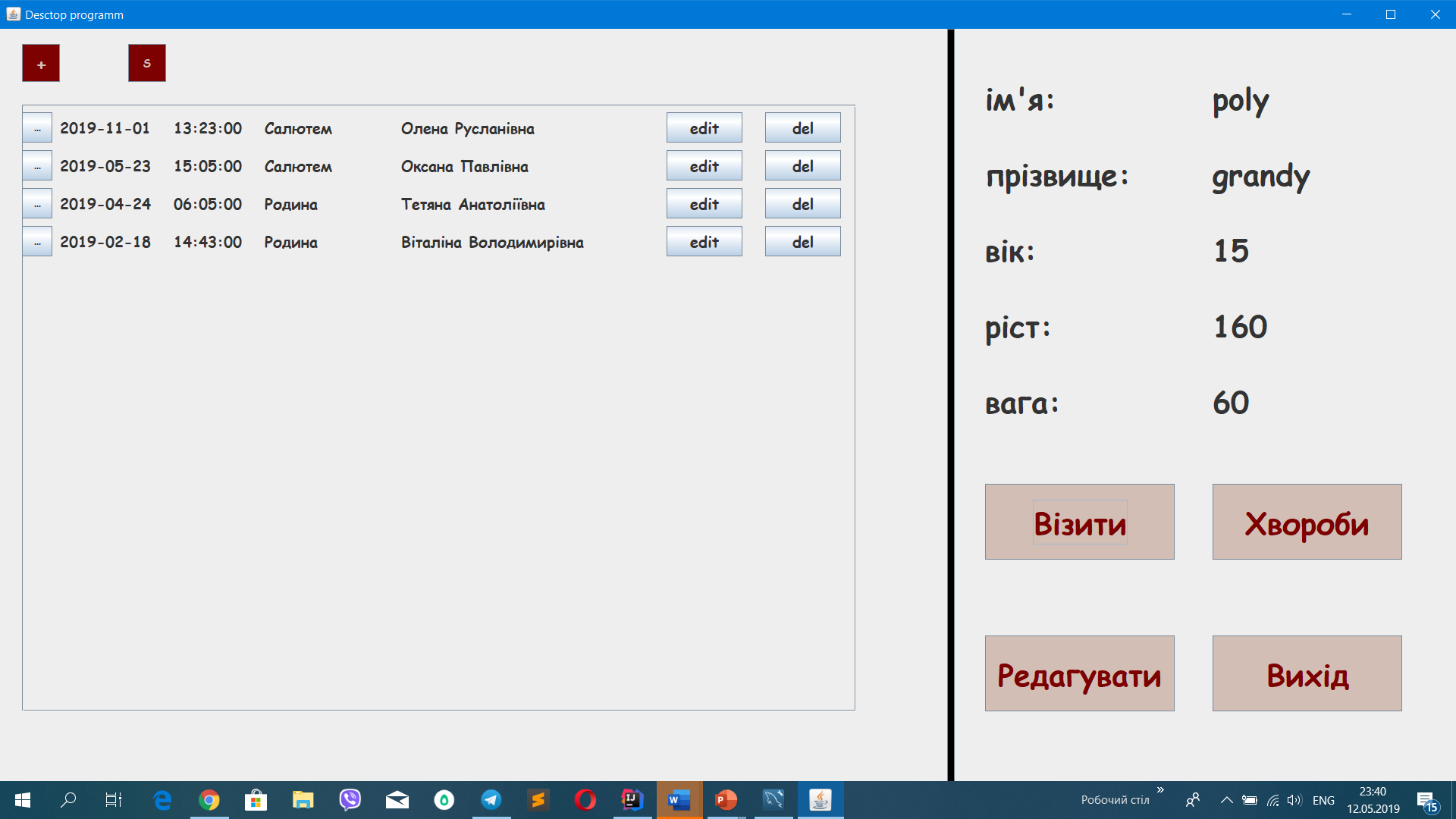


Рисунок 22 – перегляд списку візитів

Рисунок 2112 – інформаційна панель

На рисунку (рисунок 22) показана панель відображення усіх візитів певного користувача (активного). Візити автоматично сортуються по даті запланованого візиту.

На рисунку (рисунок 23) показана панель створення нового візиту. Для зручності заповнення при виборі лікарні з’являється додаткова інформація про адресу лікарні, а також у полі вибору «лікар» відбувається фільтрація усіх лікарів, користувач може обрати лише того, який працює в уже обраній ним лікарні, і також з’являється інформація про класифікацію вибраного лікаря.

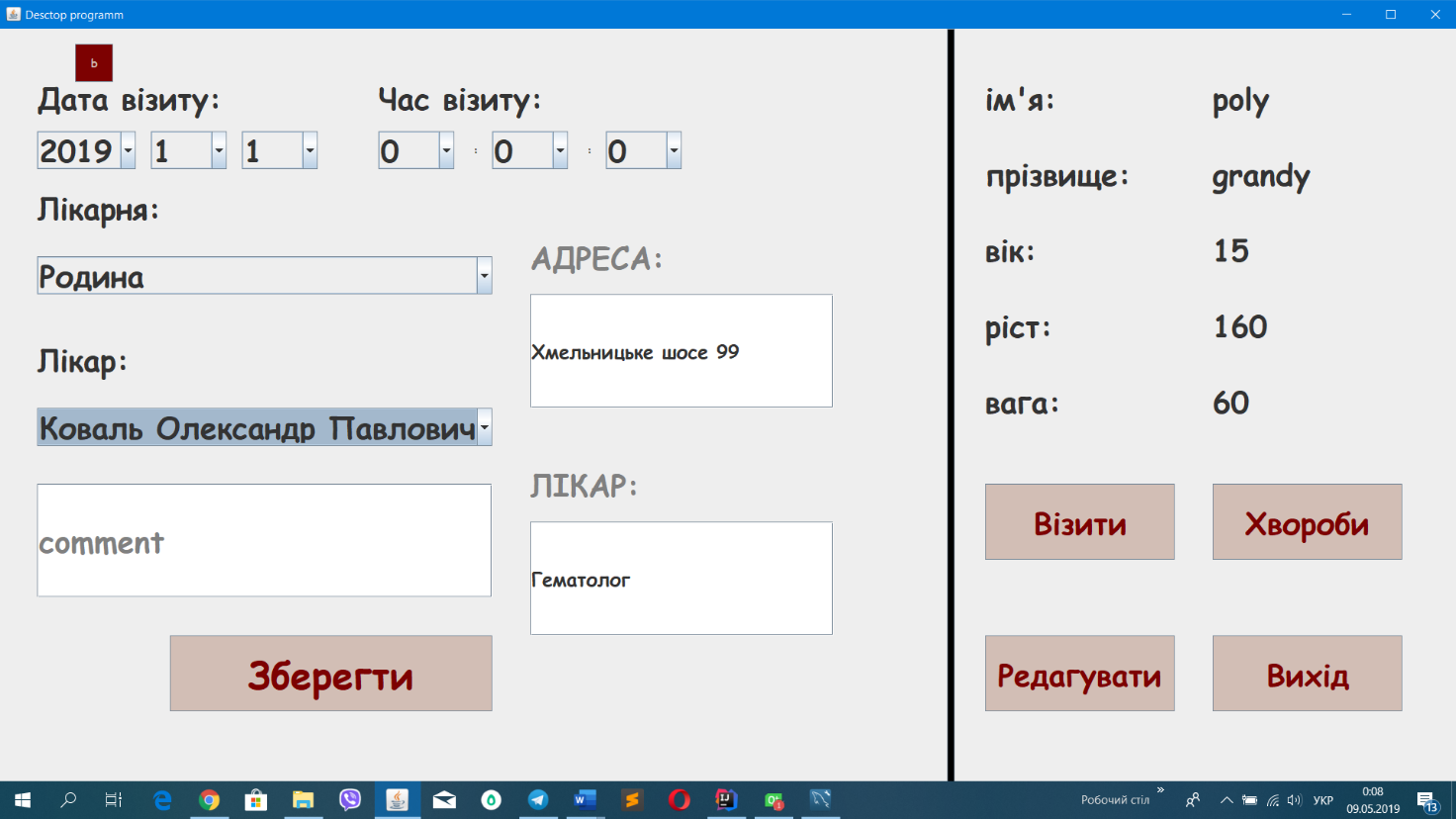


Рисунок 23 – створення візиту

На рисунку (рисунок 25) показані елементи пошуку візиту. Здійснювати пошук можна за вибором по трьом критеріям: лікар, лікарня та дата. На рисунку (рисунок 26) показані результати пошуку.

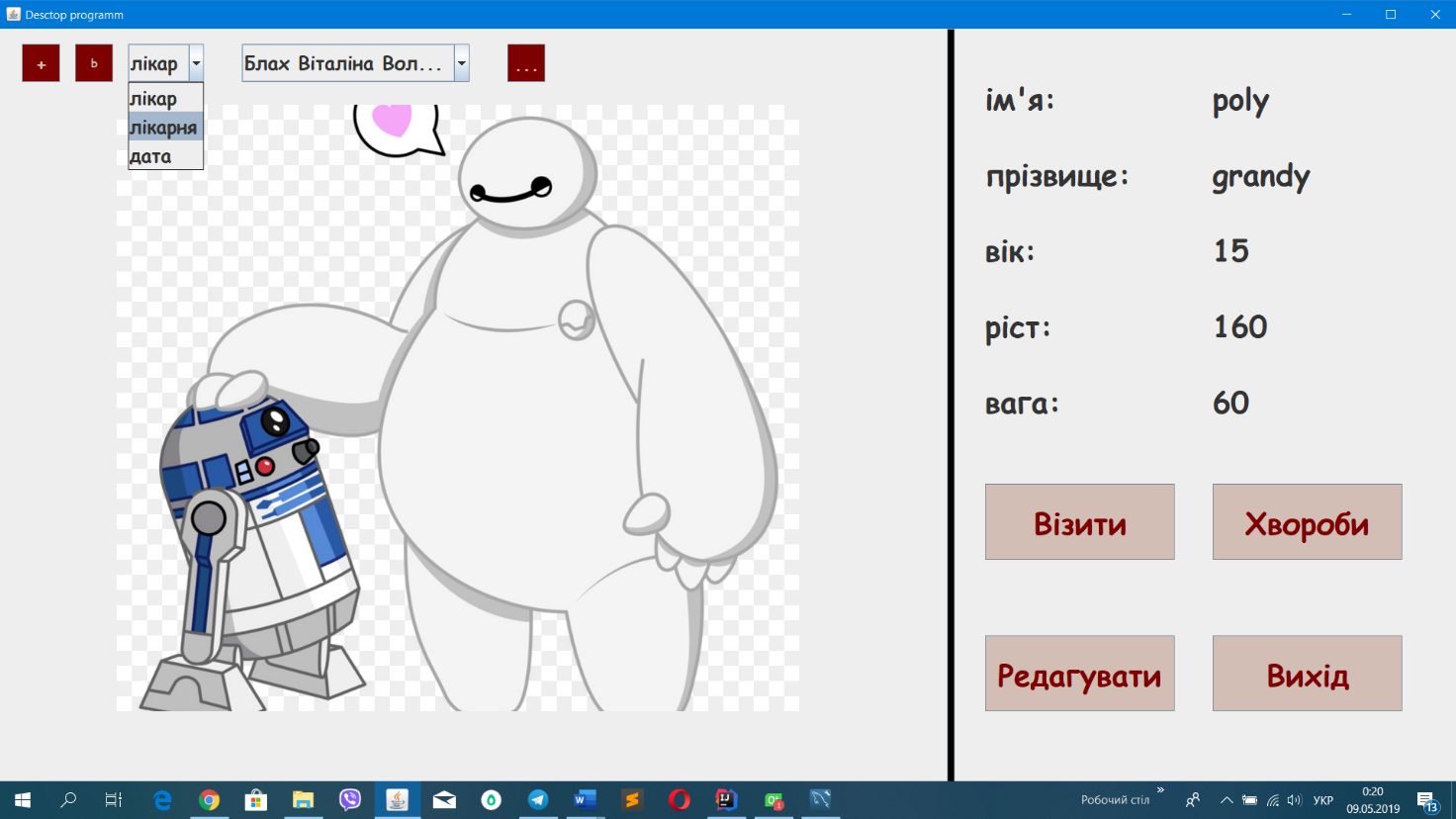


Рисунок 25 – пошук візиту

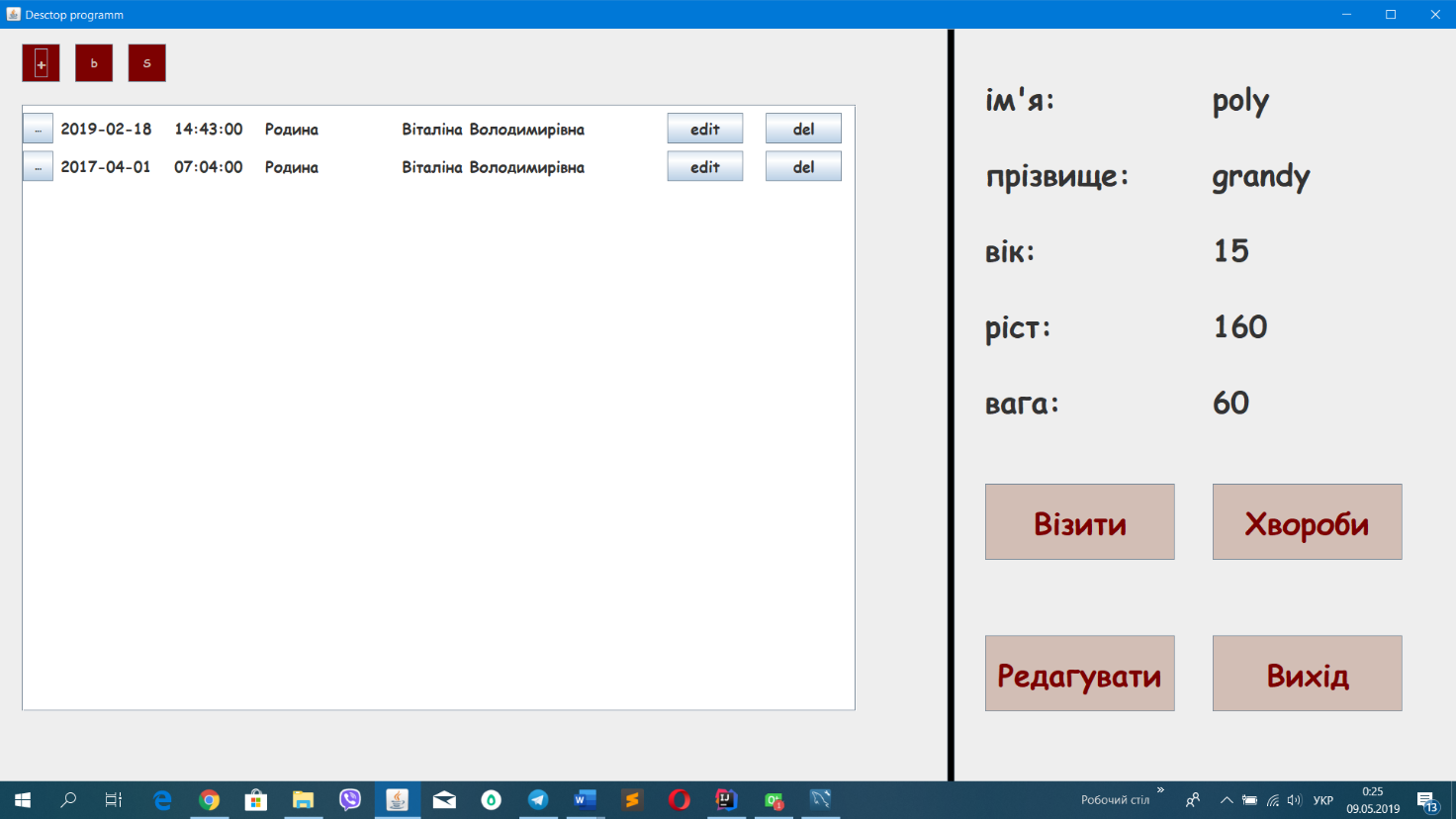


Рисунок 26 – результати пошуку

На рисунку (рисунок 27) показана панель редагування візиту усі внесені зміни будуть збережені.

Зображення, що містить знімок екрана

Автоматично згенерований опис

Рисунок 27 – редагування візиту

5.6 Опис вимог до ПЗ та АЗ

Програма повинна працювати на операційній системі Windows 10.

До складу технічних засобів повинен входити IВМ-сумісний персональний комп'ютер (ПЕОМ), що включає в себе:

* процесор, не менше intel core 2 duo;
* оперативну пам'ять, об'ємом не менше 1 Gb;
* маніпулятор типу "миша";
* Пристрій для введення – клавіатура.

Також для роботи програми повинні бути встановлені IntelliJ IDEA ULTIMATE 2019.1 та сервер MySQL.

5.7 Опис інструкції користувачеві

Start – при натисканні відкривається вікно автентифікації та реєстрації

Автентифікація – для підтвердження потрібно ввести логін і пароль

Реєстрація – щоб зареєструвати нового користувача потрібно:

* Створити логін, якого ще немає в БД
* Створити пароль (кількість символів не менше 6)
* Повторити створений пароль

Наступним етапом реєстрації є введення особистих даних та характеристик: ПІБ, номер, імейл, ріст, вага, вік.

Панель інформації та основних елементів користування знаходиться у правій частині екрану. Керувати відображенням цієї панелі можна за допомогою чорної вертикальної кнопки

Щоб перейти до перегляду, редагуванню, додаванню чи видаленню візитів потрібно натиснути кнопку «Візит», яка розміщена на панелі інформації.

Щоб перейти до перегляду, редагуванню, додаванню чи видаленню діагнозів потрібно натиснути кнопку «Діагноз», яка розміщена на панелі інформації.

Home – вихід у головне меню

Exit – вихід із свого акаунту.

Після переходу до «Візиту»:

«+» - створення нового візиту

«b» - повернутися до попереднього меню

«…» - перегляд певного візиту

«del» - видалення певного візиту

«edit» - редагування певного візиту

«s» - пошук візитів за певним критерієм

«зберегти» - зберігає будь які внесені зміни

ВИСНОВКИ

Проектування бази даних полягає в побудові комплексу взаємозв'язаних моделей даних. У процесі роботи спроектовано та розроблено базу даних у предметній області «контроль візитів та діагнозів». Вірно розроблена модель має підтримувати усі явлення користувачів. Дана база даних складається з 14 таблиць та 5 основних сутностей. Головною складністю є проектування, адже якщо допустити помилку саме при проектуванні це може мати великі наслідки, адже якщо її виявити підчас створення або ж користування, то виправлення допущеної помилки може привести до перепланування та переробки бази даних.

На основі спроектованої та створеної бази даних реалізований прототип додатку «Baymax». Програма слугує для оптимізації контролю користувачем свого здоров’я. Предметна область «Контроль візитів і діагнозів». Основні функції: автентифікація та реєстрація користувача, створення, видалення, перегляд, редагування візитів.

В подальшому планується поглиблене вивчення технологій, аналіз предметної області, покращити інтерфейс користувача, збільшити функціонал та можливості користувача.

СПИСОК ПОСИЛАНЬ

1. Лекції з дисциплін «Організація баз даних» [Електронний ресурс]. Режим доступу до лекцій: https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxjeXJpbGthZG9tc2t5fGd4OjQ3MDY3ZjA0ZWE0ZmE3NzQ
2. «Java програмування» [Електронний ресурс].Режим доступу: https://sites.google.com/site/cyrilkadomsky/coursesJava
3. Java. Реалізація шаблону DAO [Електронний ресурс]. Режим доступу: https://www.dokwork.ru/2014/02/daotalk.html
4. Підключення до бази даних [Електронний ресурс]. Режим доступу: https://metanit.com/java/database/2.2.php
5. Створення бази даних [Електронний ресурс]. Режим доступу: http://javastudy.ru/webapp/create-mysql-connect-intellij-idea/
6. MySQL [Електронний ресурс]. Режим доступу: https://www.mysql.com
7. Використання JDBC в Java [Електронний ресурс]. Режим доступу –

http://www.quizful.net/post/using-jdbc.

1. Керівництво по MySql [Електронний ресурс]. Режим доступу - http://www.mysql.ru/docs/mysql-man-4.0-ru/

ДОДАТКИ

Додаток А

Настроювання серверу і підключення до бази даних

public String userName = "root";

public String password = "b5p639bj8";

public String connectionUrl ="jdbc:mysql://localhost:3306/baymax\_2? verifyServerCertificate=false&useSSL=false&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";

try {

Class.forName("com.mysql.cj.jdbc.Driver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

try (Connection connection = DriverManager.getConnection(connectionUrl, userName, password);

Statement statement = connection.createStatement())

Додаток Б

Представлення об’єктів (візит)

public user() {};

public user(int id\_u, String surname\_u, String name\_u, int number\_u, String email, int grow\_u, Integer year\_u, Integer mount\_u, Integer day\_u, int weight\_u, String age\_u\_2) {

this.id\_u = id\_u;

this.surname\_u = surname\_u;

this.name\_u = name\_u;

this.number\_u = number\_u;

this.email = email;

this.grow\_u = grow\_u;

this.year\_u = year\_u;

this.mount\_u = mount\_u;

this.day\_u = day\_u;

this.weight\_u = weight\_u;

this.age\_u\_2 = age\_u\_2;

}

Додаток В

Приклад запитів

Знайти id усіх лікарів, які працюють у певній лікарні:

select clinic\_doctor\_id from clinic\_doctor inner join clinic on clinic\_doctor.c\_d\_clinic\_id = clinic.clinic\_id where c\_name = '"+cl+"'"

Вивести інформацію про лікарню та лікаря, які були додані в даний візит:

select clinic.c\_name, doctor.d\_name, doctor.d\_surname, doctor.d\_midlename, visit.v\_data, visit.v\_time, visit.visit\_id from doctor inner join visit\_doctor on doctor.doctor\_id = visit\_doctor.v\_d\_doctor\_id inner join visit on visit\_doctor.v\_d\_visit\_id = visit.visit\_id inner join clinic on clinic.clinic\_id = visit.v\_clinick\_id where visit.v\_user\_id = 5 group by visit.v\_data desc"

Вивести всю інформацію про користувача (окрім його id, логіна та пароля):

select i\_name, i\_surname, i\_number, i\_email, ch\_age, ch\_grow, ch\_weight from information innner join users on i\_user\_id = users.user\_id inner join characteristic on characteristic.ch\_user\_id = users.user\_id where i\_user\_id = '" + user\_id + "'"

Внести новий запис в БД про користувача:

INSERT INTO information (i\_name, i\_surname, i\_number, i\_email, i\_user\_id) values (?, ?, ?, ?, ?)

Додаток Д

Контролер

У даному додатку показаний один із контролерів – user

public class controler\_user {

database\_autentification database\_autentification = new database\_autentification();

database\_registration database\_registration = new database\_registration();

public user user;

private int user\_id = 5;

public int autentification(String login, String password)throws SQLException, ClassNotFoundException{

user\_id = database\_autentification.autentification(login,password);

user = database\_registration.get\_user(user\_id);

return user\_id;

}

public void registration(String login, String password)throws SQLException, ClassNotFoundException{

database\_registration.registration(login, password);

this.user\_id = database\_autentification.autentification(login,password);

}

public void registration\_data(String name,String surname,int number,String email,Integer year,Integer mount,Integer day,Integer grow,Integer weight)throws SQLException, ClassNotFoundException{

user = create\_user.start(name, surname, number, email, year, mount, day, grow, weight, this.user\_id);

}

}