****

**Programowanie Obiektowe**

System E-recept

Prowadzący: Autor:

mgr inż. Ewa Żesławska Dawid Jakieła

W65540

Kierunek: 3 IID/2021, grupa GP01

Rzeszów, r.a. 2022/2023

Spis treści

[**1.** **Cele projektu** 3](#_Toc126310974)

[**2.** **Opis techniczny projektu** 3](#_Toc126310975)

[**3.** **Harmonogram realizacji projektu** 4](#_Toc126310976)

[**4.** **Prezentacja warstwy użytkowej projektu** 4](#_Toc126310977)

[**5.** **Testy jednostkowe** 6](#_Toc126310978)

[**6.** **Repozytorium, system kontroli wersji** 7](#_Toc126310979)

[**7.** **Podsumowanie** 7](#_Toc126310980)

[**8.** **Literatura** 8](#_Toc126310981)

1. **Cele projektu**

Głównym celem projektu jest aplikacja, która umożliwiać będzie użytkownikowi szybką rejestrację wizyty do systemu, jej zmianę oraz anulowanie. Dodatkowo po wizycie będzie można sprawdzić swoją wystawioną receptę.

**Wymagania funkcjonalne:**

• Dodawanie pacjenta do systemu

• Możliwość dodawania, usuwania i edycji rekordów z poziomu pacjenta.

• Wyświetlanie danych odnoście e-recepty

**Wymagania niefunkcjonalne:**

• Aplikacja jest czytelna i prosta w obsłudze

• Aplikacja jest tworzona w języku Java

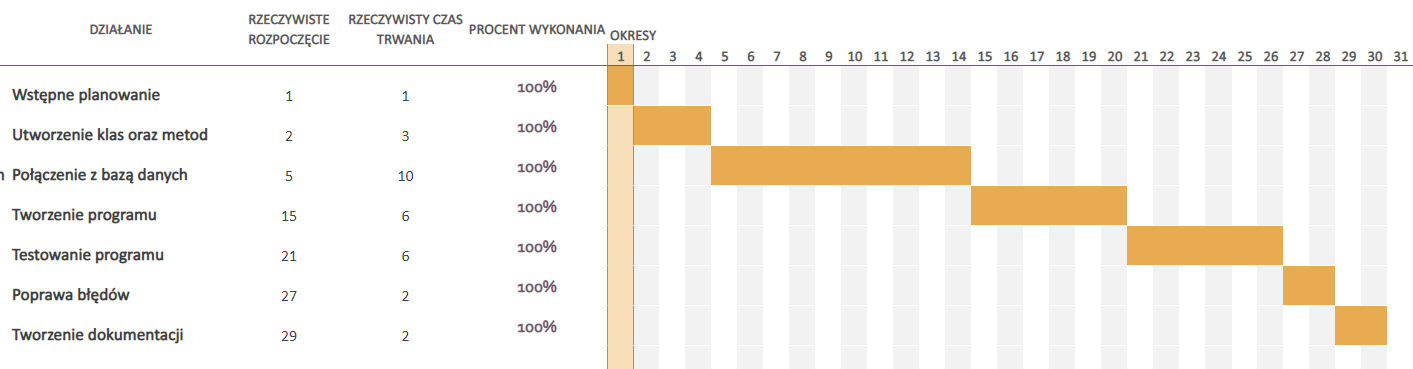
• Aplikacja nawiązuje połączenie z bazą danych i używa rekordów w niej zapisanych.

1. **Opis techniczny projektu**

* Środowisko programistyczne Javy: Java JDK 19
* Środowisko programistyczne: IntelliJ IDEA
* Baza danych : MySQLLite
* Urządzenie z system operacyjnym Windows 10

1. **Harmonogram realizacji projektu**

Poniżej zamieszono harmonogram realizacji projektu.



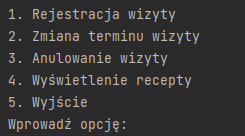
*Rysunek 1. Diagram Gantta*

1. **Prezentacja warstwy użytkowej projektu**

Po uruchomieniu programu, użytkownik(pacjent) może wybrać jedną z opcji z listy menu:

1. Rejestracja wizyty
2. Zmiana terminu wizyty
3. Anulowanie wizyty
4. Wyświetlenie wizyty
5. Wyjście

W zależności od wybranej opcji, użytkownik proszony jest o podanie kolejnych informacji potrzebnych do działania programu.

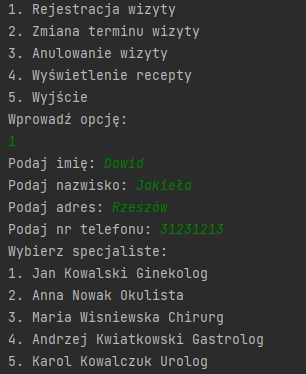


Rysunek 2.Główne Menu aplikacji

Biblioteki Javy wykorzystane do warstwy użytkowej:

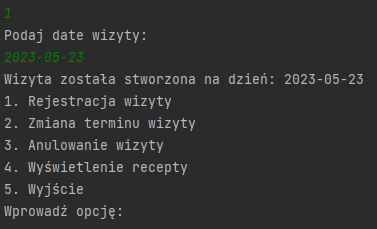
* Java.sql.rowset.CashedRowSet
* Java.sql.rowset.RowSetFactory
* Java.sql.rowset.RowSetProvider
* Java.sql.SQLException
* Java.until.ArrayList
* Java.until.List
* Java.until.Scanner
* Java.sql.\*
* Java.until.\*  
  Służą one do połączenia się z bazą danych i wykonywania odpowiednich operacji na danych.

Zacznę więc od rejestracji nowego pacjenta, po wybraniu opcji program prosi nas o podanie swoich danych. Gdy to zrobimy przechodzimy do kolejnego kroku, jakim jest wybranie specjalisty do którego chcemy się umówić na wizytę.



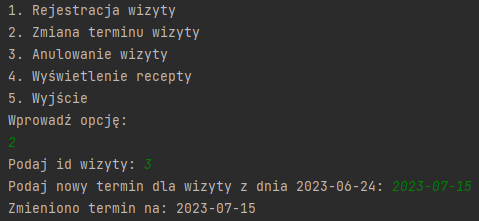
Rysunek 3.Rejestracja pacjenta oraz wizyty

Po wybraniu specjalności lekarza, program prosi nas o podanie daty wizyty, aby mógł ją zapisać w bazie danych. Po wypełnieniu daty, przenosi nas do menu głównego.



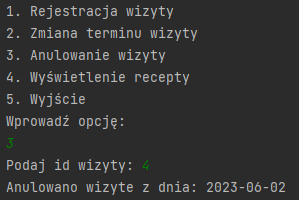
Rysunek 4.Dalszy przebieg rejestracji pacjenta na wizytę

Aplikacja umożliwia nam zmianę daty swojej wizyty. Proces ten wymaga jedynie podania swojego ID\_wizyty.



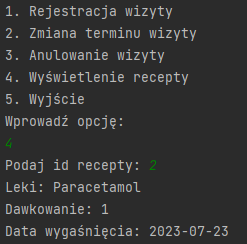
Rysunek 5.Zmiana terminu wizyty

Aplikacja umożliwia również anulowanie zapisanej wizyty. Proces ten nie jest jeszcze w pełni ulepszony, wymaga na razie jak w przykładzie powyżej podania ID\_wizyty, która ma zostać usunięta.



Rysunek 5.Anulowanie wizyty

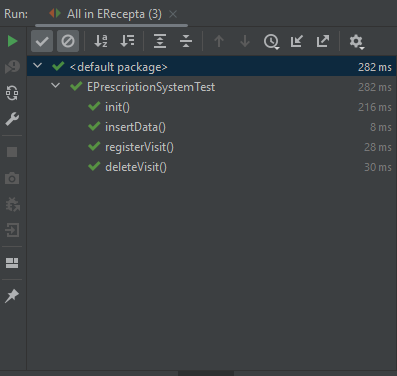
Ostatnim punktem aplikacji jest jej najważniejsza część, wyświetlenie recepty wypisanej przez lekarza. Wszystko opiera się tutaj jak na razie na ID\_wizyty.



Rysunek 6.E-recepta

1. **Testy jednostkowe**

Testy jednostkowe zostały przeprowadzone w technologii JUnit5. Testy polegały na między innymi sprawdzeniu czy baza danych działa prawidłowo i czy prawidłowo wczytuje dane. Dodatkowo została sprawdzona rejestracja użytkownika, dodanie wizyty oraz jej usunięcie. Wszystkie testy zostały wykonane pomyślnie i potwierdziły poprawność działania systemu.

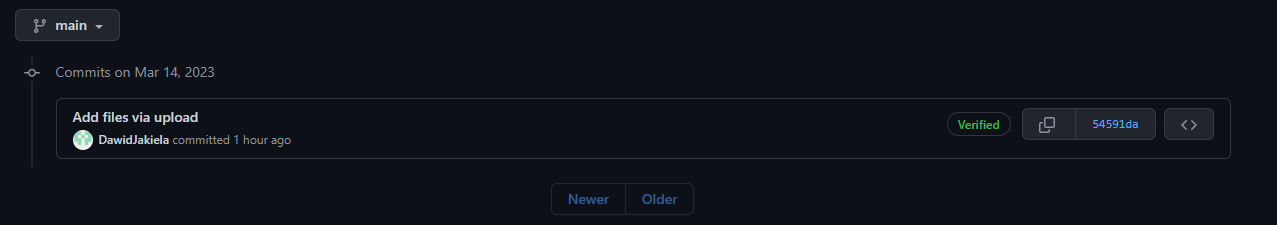


*Rysunek 7.Testy jednostkowe*

1. **Repozytorium, system kontroli wersji**

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji JUnit. Na Rysunku 8 przedstawiono zrzut ekranu pokazujący historię commitów. Do kodu projektu stworzono dokumentację poprzez system komentarzy dokumentującym. Dokumentacja oraz projekt został umieszczony w repozytorium dostępnym pod adresem:

<https://github.com/DawidJakiela/E-recepta>



*Rysunek 8.Commit*

1. **Podsumowanie**

Projekt pomimo działania, nadal potrzebuje wielu dalszych usprawnień. Został on stworzony w oparciu widzę zdobytą na zajęciach oraz własną.

Zawiera wszelkie podstawowe informacje oraz usprawnienia, aby użytkownik potrafił się  
w nim odnaleźć.

1. **Literatura**
2. <https://www.sqlitetutorial.net>
3. <https://www.samouczekprogramisty.pl/testy-jednostkowe-z-junit5/>
4. <https://stackoverflow.com/questions/8002756/sqlite-composite-key-2-foreign-keys-link-table>
5. <https://www.javatpoint.com/string-array-in-java>
6. <https://www.geeksforgeeks.org/how-to-convert-all-linkedhashmap-key-value-pairs-to-list-in-java/>
7. <https://stackoverflow.com/questions/26511532/using-junit-to-test-writing-and-reading-to-sqlite>
8. <https://stackoverflow.com/questions/3693626/how-to-run-test-methods-in-specific-order-in-junit4>
9. <https://www.jetbrains.com/help/idea/junit.html>
10. <https://stackoverflow.com/questions/53030842/why-resultsets-index-in-java-starts-with-1-instead-of-zero>
11. <https://stackoverflow.com/questions/6409371/java-resultset-get-column-name-based-on-index>
12. <https://docs.oracle.com/javase/6/docs/api/javax/sql/rowset/CachedRowSet.html>