

## 1. Temat

Projekt obejmuje działanie przychodni lekarskiej w niewielkiej miejscowości, w celu usprawnienia jej pracy i pozbycia się błędów związanych z naturą ludzką takie jak niemożliwość dodzwonienia się do rejestracji czy też długi czas oczekiwania na termin wizyty.

Głównym celem projektu jest przygotowanie systemu, który wspomogę działanie przychodni, w szczególności w aspekcie obsługi rejestrowania pacjentów, komunikacji lekarz –pacjent (potwierdzanie wizyt pacjentów lub ich odwoływanie, przypominanie o nadchodzących wizytach, szybki dostęp do wystawionych recept) jak i lekarz-lekarz (prowadzenie elektronicznej historii medycznej pacjenta), prowadzenia bazy wizyt.

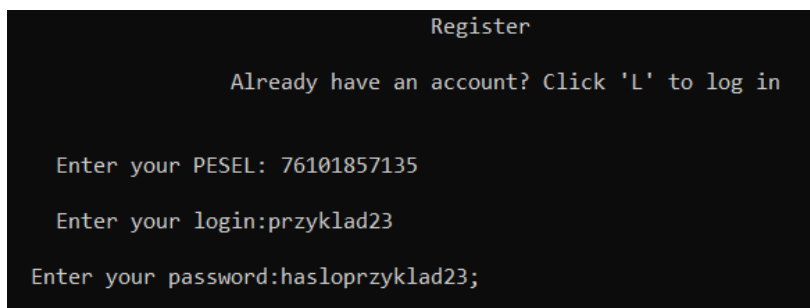
System opiera się na bazie danych złożonych z plików tekstowych zawierających dane na temat lekarzy, grafików, wizyt, historii medycznej pacjentów i danych samych pacjentów. Zadaniem systemu jest nie tylko przetwarzanie danych jak i ich ochrona ze względu na wrażliwość dysponowanych danych.

## 2. Specyfikacja zewnętrzna

Użytkownik komunikuje się z programem poprzez ręczne wprowadzenie danych za pomocą klawiatury. System jest w formie aplikacji konsolowej. System jest dedykowany dla dwóch typów użytkowników dla pacjenta oraz lekarza, każdy z nich ma dostępną inną funkcjonalność. Pacjent może umawiać się na wizyty, przeglądać swoją historię medyczną, jest również informowany o zbliżającym się terminie kolejnej wizyty, wystawieniu recepty lub zmiany statusu umówionej wizyty. Lekarz może układać i edytować swój grafik (dodawać różne wydarzenia oprócz wizyt pacjentów), akceptować lub odwoływać wizyty pacjentów, wystawiać recepty oraz przeglądać historię medyczną pacjenta. Poza funkcjonalnością związaną z typem użytkownika, każdy użytkownik może zmienić swoje hasło i dane (poza peselem).

### 2.1 Rejestracja

Po udanych uruchomieniu projektu użytkownik na ekranie widzi menu rejestracji (Rys.1).



```

Register

Already have an account? Click 'L' to log in

Enter your PESEL: 76101857135

Enter your login:przyklad23

Enter your password:hasloprzyklad23;
  
```

*Rys.1 Menu rejestracji*

Jeśli użytkownik posiada już konto w systemie w celu zalogowania się do swojego konta musi kliknąć klawisz „L”. Jeśli celem użytkownika jest rejestracja to w odpowiedzi na pytanie czy ma już konto powinien kliknąć inny klawisz klawiatury niż „L”, a następnie wprowadzić z klawiatury kolejno PESEL, login oraz hasło. Po poprawnym wpisaniu wymaganych informacji, użytkownik w celu dokończenia tworzenia konta i uzyskania dostępu do pełnej funkcjonalności systemu musi wprowadzić następujące informacje (Rys.2). Jeśli tego nie zrobi lub nie wypełni wszystkich informacji, przy kolejnym logowaniu będzie musiał je wypełnić na nowo (Tylko te z Rys.2).

```

In order to create an account successfully, enter answers to those questions

Name:Mariusz

Surname:Adamczewski

Date of birth:18.10.76

City of birth:Brzeg Dolny

Sex ('F' for female 'M' for male):M_

```

*Rys.2 Informacje kończące tworzenie konta*

Po poprawnym wpisaniu informacji użytkownik został pomyślnie zarejestrowany.

## 2.2 Logowanie

W celu zalogowania się na swoje konto użytkownik powinien podać kolejno swój login i hasło (Rys.3)

```

Log into Your Account

Enter your login:

Enter your password:

```

*Rys.3 Logowanie*

Po poprawnym wpisaniu informacji użytkownik został pomyślnie zalogowany i widzi menu główne swojego konta (Rys.4 dotyczy użytkownika w roli pacjenta, a Rys.5 dotyczy użytkownika w roli lekarza).

```

Ewa Springer
-----
Upcoming appointments:
  +(27.06.22 at 11:30) GP appointment (issued)
  +(01.07.22 at 15:00) Dermatologist appointment (confirmed)

  ~To get more information about appointments click 'Y'
-----

PRESCRIPTION ALERT! You have 1 perscription/s available
  ~To get more information about this click 'P'
-----
  ~To book doctor appointment click 'B'
-----

  ~To get your medical history record click 'R'
-----
  ~LOG out click 'X'
-----
  ~To EXIT click 'E'
-----
  ~Choice:_

```

*Rys.4 Menu główne kont pacjenta*

Pierwsza część menu pacjenta zawiera informacje o aktualnościach związanych z jego kontem (status jego wizyt, informacje o jego wizytach). Pacjent ma możliwość sprawdzenia szczegółów tych informacji oraz ich edycji (np. odwołanie, zmienienie terminu wizyty), jest również informowany o wystawieniu recepty. Ma możliwość umówienia się na wizytę oraz sprawdzenia swojej elektronicznej historii medycznej (Rys.8).

Weekly agenda:

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
20.06.22	21.06.22	22.06.22	23.06.22	24.06.22	25.06.22	26.06.22
09:00-12:20 skin biopsies	08:00-09:00 Cryotherapy		12:00-17:00 remv skin cancr	12:00-14:40 Meeting in Warsaw	08:30-09:00 Patient appoint.	08:00-08:30 Patient appoint.
16:00-18:00 Staff meeting	09:00-10:00 Sclerotherapy		17:00-19:00 Cryosurgery			08:30-09:00 Patient appoint.
	10:00-11:15 break :0					
	13:00-14:40 Dermabrasion					
	14:30-15:00 Patient appoint.					

~To get more information about this week click 'W'

~To display another week click 'D'

~To add appointment description (perscription/notes) click 'P'

APPOINTMENT ALERT! 2 appointment request/s have been issued  
~To get more information about this click 'A'

~To schedule an event click 'S'

~To LOG OUT click 'X'

~To EXIT click 'E'

~Choice: x\_

Rys.5 Menu główne konta lekarza

Lekarz przez menu główne konta lekarza ma możliwość edycji swojego grafiku (umawiania wizyt, odwoływania ich, dodawania innych zajęć do grafiku), potwierdzania wizyt. Po wybraniu klawisza 'W' uzyskuje szczegółowe informacje dotyczące swojego grafiku, ma wtedy możliwość edycji historii medycznej pacjenta lub przepisanie recepty.

### 2.3 Nadchodzące wizyty

Po zalogowaniu się na swoje konto, w menu głównym pacjenta wyświetlane jest przypomnienie o nadchodzących wizytach (w skrócie); użytkownik jest informowany co powinien zrobić by wyświetlić szczegóły tych wizyt.

Upcoming appointments:
+(27.06.22 at 11:30) GP appointment (issued)
+(01.07.22 at 15:00) Dermatologist appointment (confirmed)
~To get more information about appointments click 'Y'

Rys.6 Przypomnienie o nadchodzących wizytach w menu głównym

Po wciśnięciu klawisza 'Y' pacjent przegląda szczegóły nadchodzących wizyt (Rys.7).

```

APPOINTMENT DETAILS: GP appointment
-DATE:27.06.22  TIME:11:30-12:00
  PATIENT'S INFO:
    -NAME:Ewa Springer
    -PESEL:89051746686
DOCTORS'S INFO:Balbina Sobczak
  -notes:
  -medication:
  -Appointment status: issued

APPOINTMENT DETAILS: Dermatologist appointment
-DATE:01.07.22  TIME:15:00-15:30
  PATIENT'S INFO:
    -NAME:Ewa Springer
    -PESEL:89051746686
DOCTORS'S INFO:Lara Jankowska
  -notes:
  -medication:
  -Appointment status: confirmed

Press any key to continue . . . █

```

*Rys.7. Szczegóły nadchodzących wizyt*

## 2.4 Przypomnienie o recepcie

Po zalogowaniu się na swoje konto, w menu głównym pacjenta wyświetlana jest informacja o ilości wystawionych recept; użytkownik jest informowany co powinien zrobić by wyświetlić szczegóły tych recept.

```

-----
PRESCRIPTION ALERT! You have 1 perscription/s available
~To get more information about this click 'P'
-----

```

*Rys.8. Przypomnienie o wystawionych receptach w menu głównym*

Po wciśnięciu klawisza 'P' pacjent przegląda wystawione recepty (Rys.9).

```

AVAILABLE PERSSCRIPTIONS:

Perscription: loratadine 1/day
EVENT DETAILS: Patient appoint.
-DATE:20.06.22  TIME:09:30-10:00
  -notes:Sun Allergy, For 2 weeks avoid sun exposure

Press any key to continue . . . █

```

*Rys.9. Szczegóły wystawionych recept*

## 2.5 Umawianie wizyt

Po wciśnięciu klawisza 'B' w menu głównym konta pacjenta widoczne na ekranie jest menu do umawiania wizyt (Rys.6). Pacjent może umówić się na wizytę poprzez wpisanie daty planowanej wizyty oraz specjalności lekarza (wybierane poprzez wpisanie odpowiedniej liczby stojącej przy specjalności), do którego chce się umówić jeśli nie jest dla niego istotne do którego lekarza zostanie umówiony. Możliwe jest też umawianie wizyty gdy głównym kryterium wybrania terminu wizyty jest wizyta z konkretnym lekarzem. Po wybraniu kryterium wyboru wyświetlane są dostępne terminy w okresie tygodnia. Pacjent wybiera termin wizyty, jeśli żaden z przedstawionych terminów pacjentowi nie odpowiadają, to pacjent może wyświetlić kolejne dostępne terminy (które pokrywają kolejne dwa tygodnie). Wizytę musi potwierdzić lekarz (który ma obowiązek potwierdzania wizyt) aby była wpisana do jego grafiku.

```

BOOKING
-----

1. General Practitioner
  2. Dentist
    3. Dermatologist
      4. Gynecologist
        5. Ophthamologist
          6. General Surgeon
            7. Urologist
              8. Allergist
                9. Cardiologist
                  10. Psychiatrist
                    11. Clinical nutritionist
                      12. Oncologist
                        13. Pediatritian
                          14. Physiotherapist

Choose speciality/ expertise by entering adequate number: 6
-----

Enter appointment's date: 23.11.22

-----

1. 08:00-08:30 dr Krauze
2. 09:00-09:30 dr Krauze
3. 10:00-10:30 dr Krauze
4. 10:00-10:30 dr Goldberg
5. 12:30-13:00 dr Krauze
6. 15:00-15:30 dr Goldberg

Choose available appointment by entering adequate number: 3
-----

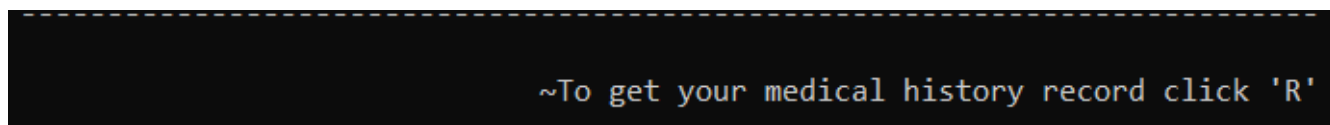
Your doctor will confirm your booking in a few days

```

Rys.10 Umawianie wizyt

## 2.6 Historia medyczna

Po wciśnięciu klawisza 'R' wyświetlana jest historia medyczna pacjenta oraz jest ona zapisywana w pliku tekstowym pod nazwą taką jakie jest imię i nazwisko pacjenta (Rys.11).



*Rys.11 Widok menu- historia medyczna pacjenta*

Po wciśnięciu klawisza 'R' pacjent przegląda swoją historię medyczną (Rys.12).

```

Upcoming appointments::

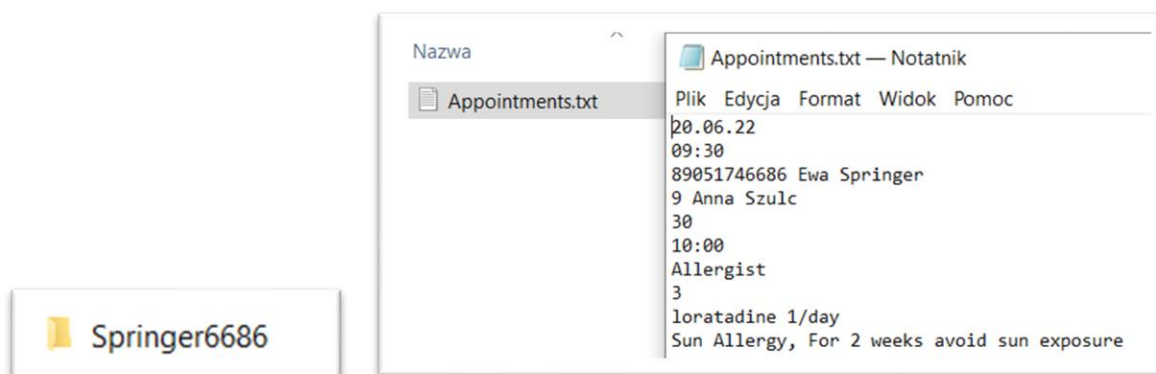
APPOINTMENT DETAILS: GP appointment
-DATE:27.06.22  TIME:11:30-12:00
PATIENT'S INFO:
-NAME:Ewa Springer
-PESEL:89051746686
DOCTORS'S INFO:Balbina Sobczak
-notes:
-medication:
-Appointment status: issued

APPOINTMENT DETAILS: Dermatologist appointment
-DATE:01.07.22  TIME:15:00-15:30
PATIENT'S INFO:
-NAME:Ewa Springer
-PESEL:89051746686
DOCTORS'S INFO:Lara Jankowska
-notes:
-medication:
-Appointment status: confirmed

~You'll find your medical history file in Springer6686 directory
Press any key to continue . . . █
  
```

*Rys.12 Historia medyczna pacjenta*

Wyświetlane są wszystkie wizyty, bez względu na status. Użytkownik informowany jest również, w jakim katalogu znajdzie plik ze swoją historią medyczną (Rys.13).



*Rys.13 Utworzony folder pacjenta, oraz jego zawartość (fragment pliku Appointments.txt)*

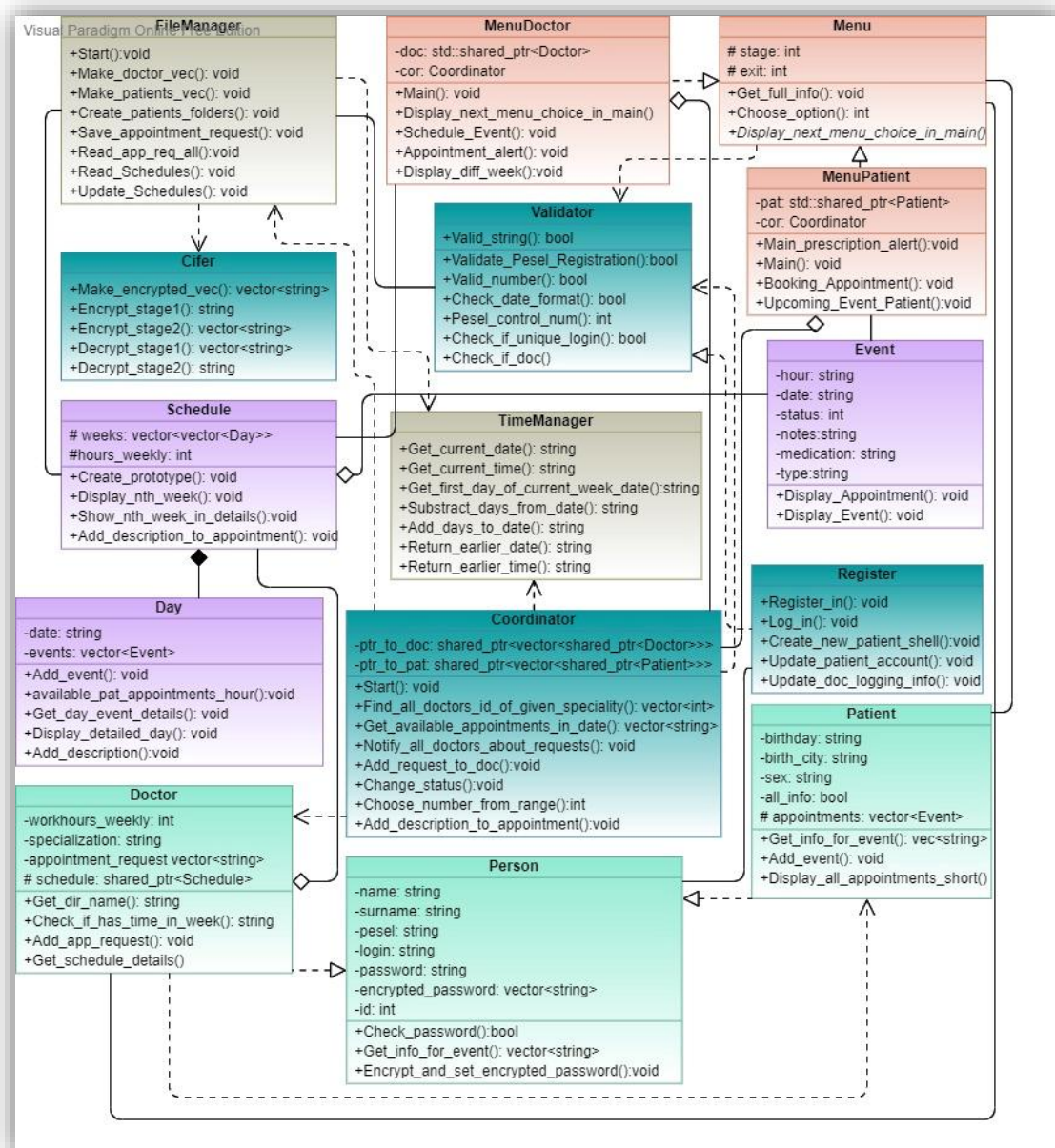
## 4. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikacje z użytkownikiem) od logiki programu. Każdy obiekt w programie służy jako model abstrakcyjnego "wykonawcy", który może wykonywać pracę, opisywać i zmieniać swój stan, oraz komunikować się z innymi obiektami w programie. Do stworzeniu modelu bazy danych wykorzystano pliki tekstowe. Przechowują one informacje kto jest lekarzem, zapisywana jest do niej historia wizyt, grafiki lekarzy oraz historie medyczne pacjentów. Model systemu jest przedstawiony za pomocą diagramu klas przedstawionym na Rys.14.

### 4.1 Klasy

Program jest programem konsolowym, w celu komunikacji z użytkownikiem została stworzona statyczna klasa **Menu**, która jest odpowiedzialna za wyświetlanie odpowiednich treści (menu). Zawiera ona statyczne, wirtualne metody wyświetlające różne menu, po niej dziedziczą dwie klasy **Menu\_Patient** i **Menu\_Doc**. Klasa **Menu** współpracuje ze statyczną klasą **Register**, która jest odpowiedzialna za „zbieranie” informacji potrzebnych do utworzenia nowego konta lub do zalogowania się na istniejące już konto. Klasa **Register** wykorzystuje metody statycznej klasy **Validator**, która sprawdza poprawność podawanych informacji (np. czy w polu, w którym powinien być wpisany PESEL nie zostało wpisane hasło). Klasa **Validator** pełni ważną funkcję w ochronie przechowywanych danych poprzez kodowanie wrażliwych danych takich jak hasło. Kolejnymi utworzonymi klasami jest klasa abstrakcyjna **Person** oraz klasy od niej pochodne klasy **Patient** oraz **Doctor**. Klasy **Patient** i **Doctor** są poszerzone o metody związane z funkcjonalnością związaną z typem użytkownika, oraz pola z (klasa **Patient**) danymi potrzebnymi do wystawienia recepty i kontener zawierający wskaźniki z wizytami danego pacjenta oraz (klasa **Doctor**) kontener zawierający zaplanowane wydarzenia w każdym dniu (grafik), ilość godzin do wypracowania w tygodniu oraz specjalizację lekarską. Każda z pochodnych klas ma dostęp do wirtualnych metod klasy **Person** umożliwiających odwołanie wizyty, zmianę terminu wizyty czy zmianę hasła. Wspomniane klasy wykorzystują klasę **Event**, której obiekty stanowią reprezentację wizyt; wydarzeń. Klasą odpowiedzialną za zarządzanie plikami; odczytywanie, aktualizowanie danych oraz tworzenie jest klasa statyczna **FileManager** ściśle współpracująca z klasami **Validator**, **Register**, **Person** i jej klasami potomnymi oraz z klasą **Cifer**. Klasa **Cifer** przechowuje metody związane z szyfrowaniem oraz odszyfrowaniem haseł. Klasa statyczna **TimeManager** jest odpowiedzialna za operacje na łańcuchach znaków przedstawiające datę i godzinę, takie jak dodawanie czy odejmowanie. Współpracuje z klasami **Coordinator** oraz **FileManager**. Klasa **Schedule** spełnia rolę kalendarza. Silnie współpracuje z klasą **Doctor**, oraz składa się z obiektów klasy **Day**. Obiekty klasy **Day** zawierają dane o wybranym dniu w kalendarzu oraz mogą wykonywać na nich operacje. Klasa **Coordinator** umożliwia komunikację pomiędzy lekarzami i pacjentami; wspomaga zarządzaniem wizyt; Współpracuje z większością klas.





Rys.14 Diagram klas

## 4.2 Biblioteki

W realizacji programu wykorzystywane były biblioteki `std::filesystem`, która ułatwiała zarządzanie ścieżkami dostępu do katalogów i plików, `std::ranges` do kodowania i dekodowania haseł oraz `std::regex` do sprawdzania mocy haseł jak i odpowiedniego formatu danych wpisywanych przez użytkownika (np. format daty).

W celu skrócenia czasu kompilacji zastosowano **moduły**.



```
static void Fill_docs_with_logging_info(std::vector< std::shared_ptr<Doctor>>& docs)
{
    fs::path data_input("Data");
    if (fs::exists(data_input))
    {
        std::ofstream ofStream;
        std::ifstream ifStream;
        std::stringstream ss;
        std::string line;
        for (auto& file : fs::directory_iterator(data_input))
        {
            if (file.path().filename() == "logging_doc.txt")
```

*Rys.15 Przykład wykorzystania biblioteki std::filesystem (fragment metody)*

```
static std::vector<std::string> Decrypt_stage1(std::vector<std::string>&vec, int key)
{
    //subtracting_key
    auto keying = std::views::transform(vec, [&key](std::string encrypted)
    {
        int dig = stoi(encrypted);
        dig -= Get_last_digit(key);
        std::string str = std::to_string(dig);
        return str;
    });
    std::vector<std::string> v(keying.begin(), keying.end());

    return v;
}
```

*Rys.16 Przykład wykorzystania biblioteki std::ranges*

```
static bool Check_date_format(std::string& str)
{
    if (str.size() != 8)
    {
        return 0;
    }
    else
    {
        //date format accepted: DD-MM-YY or DD.MM.YY
        std::regex date_format("([0-2]{1}\\d{1})|(3{1}[0-1]{1}))(-|\\.)(0{1}\\d{1})|(1{1}[0-2]{1}))(-|\\.)(\\d{2})");
        return Validator::Find_Matches(str, date_format);
    }
}
```

*Rys.17 Przykład wykorzystania biblioteki std::regex*

### 4.3 Szyfrowanie hasel

Wszystkie wprowadzane hasła były szyfrowane i zapisywane w pliku zawierającym informacje o danych do logowania.

loging\_doc.txt — Notatnik

Plik	Edycja	Format	Widok	Pomoc
94040974865		Sobierobie	721131041135051525360	233322222
89122185943				
81033026223		Janeczka	70117114105119565649	23333222
78102578127		ATINKRAt	7369869384108105118	22222333
93101642321		Wonderland	1191221051026868	333322
87042864186		MelkaSerdelka	1101031211121121131065665	33333322
97041278662		lasik	1161041071291125859606149	3333322222
64103193924		emipraca	1181191271051155859656571	3333322222
84102419247		asteroid	10612412511012312011410974119119106	333333332333
71042483861		ninka_the_cardiologist	10410197114116115495051525341	332333222222
86020287629				
60001000000				

users\_pat.txt — Notatnik

Plik	Edycja	Format	Widok	Pomoc
86060685537		Warnerek86	811061101131081021085755	233333322
89061369121		Ewkaaa	791071091101031211099951535056	233333322222
80112349723		Nowak23	100101102535452535466	333222222
88071723477		Kuba_do_lek	7012110210954556767	23332222
05300565668		Emiss	8210211410280102127102113102535868	2333233333222
90041414221		Kwiatuszek2	11011711011757587697	33332222
76101868733		login18	72107104116108114566370	233333222
76100774538		Boba22	12112710912212412957585971	3333332222
96062962734		Kvejuqiz	811261091261071101221266663	2333333322
96120633385		Agjae	871141219712110110897485442	23323332222
89051746686		hypernova	115104535452103116	3322233
89051746266		dawidson	1029912110710211711311251	323333332
89051785328		abc	52535455	2222

Rys.18, 19 Fragmenty plików zawierających zaszyfrowane hasła

## 5. Testowanie

Program został przetestowany na różnego rodzaju danych. Danych poprawnych i typowych (np.: poprawne daty, przy wybieraniu możliwych godzin wizyt wybierane były te, z zakresu przedstawianego). Danych poprawnych ale nietypowych (np.: poprawne formaty ale spoza zakresu). Danych niepoprawnych (np.: w miejscu gdzie powinno być wpisane słowo, wpisywana była liczba lub nic nie było wpisywane). Program został sprawdzony pod kątem wycieków pamięci.

Link do githuba: <https://github.com/polsl-aei-pk4/efda8b49-gr11-repo>