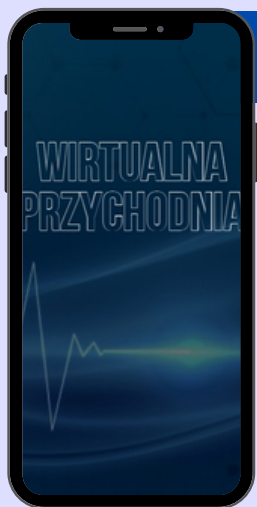


WIRTUALNA PRZYCHODNIA

Aplikacja mobilna



TECHNOLOGIE



Android
Studio



Kotlin



Firebase

T W Ó R C A

Aplikacja została stworzona przez Justynę Drozdowską, absolwentkę Katolickiego Uniwersytetu Lubelskiego na kierunku informatyka.

C E L

Głównym celem, którym kierowano się podczas tworzenia projektu było stworzenie prostej w obsłudze aplikacji mobilnej, która ułatwi użytkownikom dostęp do informacji medycznych. Użytkownikami, do których skierowana jest ta aplikacja są zarówno osoby starsze, jak i młodsze. Z tego powodu bardzo ważne jest stworzenie intuicyjnego wyglądu, który wyeliminuje ryzyko problemów w odnalezieniu potrzebnych funkcjonalności.

Z A L E T Y A P L I K A C J I

- Aplikacja jest prosta w obsłudze oraz posiada czytelną czcionkę.
- Niebieska kolorystyka aplikacji nawiązuje do medycyny, natomiast często ludzie niechętnie wybierają się do lekarza oraz boją się wizyt, dlatego motywem przewodnim jest tło ombre, które ma na celu uspokoić pacjentów.
- Jedna aplikacja posiada wiele funkcjonalności, dzięki czemu nie trzeba pobierać kilku oddzielnych aplikacji.

K L U C Z O W E F U N K C J E

- Spersonalizowane konto pacjenta, które umożliwia korzystanie z wszystkich funkcji aplikacji.
- Baza danych z medycznymi zagadnieniami.
- Forum, które zapewnia dostęp do kontaktu z innymi użytkownikami.
- Zdalna rejestracja na wizyty.

WIRTUALNA PRZYZCHODNIA

Kroki tworzenia aplikacji

1. WSTĘPNE ZAŁOŻENIA

W pierwszej kolejności opracowana została wstępna architektura oraz wygląd aplikacji. Wybrane zostały technologie, które będą wykorzystywane do tworzenia projektu.

2. WYMAGANIA

Opracowanie oraz sprecyzowanie wymagań funkcjonalnych oraz нефункциональных.

3. BAZA DANYCH

Stworzenie struktury bazy danych.

4. TWORZENIE APLIKACJI

Budowanie modułów w aplikacji w kolejności:

- Ekran powitalny
- Rejestracja oraz logowanie
- Menu
- Profil użytkownika
- Informacje
- Forum
- Rejestracja online

5. TESTOWANIE

Testowanie aplikacji, sprawdzanie działania poszczególnych modułów.

6. DOKUMENTACJA

Podczas wszystkich kroków powstawała dokumentacja, zapisywane były ważne informacje dotyczące funkcjonalności.

WIRTUALNA PRZYCHODNIA

Analiza wymagań

W Y M A G A N I A F U N K C J O N A L N E

- Możliwość założenia konta oraz zarządzania nim.
 - Możliwość dostępu do spersonalizowanego konta po zalogowaniu się.
 - Wykrywanie błędnych danych oraz czytelne komunikaty, na podstawie których użytkownik dowie się, dlaczego wprowadzone przez niego informacje są niepoprawne.
 - Dostęp do forum, możliwość czytania i dodawania własnych tematów oraz komentarzy.
 - Dostęp do dokumentacji oraz ogólnodostępnej wiedzy medycznej.
 - Dostęp do rejestracji wizyt.
-

W Y M A G A N I A N I E F U N K C J O N A L N E

- Aplikacja dostępna na platformie mobilnej Android.
 - Aplikacja dostępna na telefony posiadające minimalne SDK API 19: system Android 4.4 lub nowsze.
 - Baza danych Firebase, w której będą przechowywane podstawowe dane, które są potrzebne do działania aplikacji, preferencje, komentarze i tematy z forum. Do działania bazy Firebase jest potrzebne połączenie z internetem.
 - Spójna, spokojna kolorystyka oraz czytelna czcionka.
-

WIRTUALNA PRZYZCHODNIA

Baza danych



Firebase

BAZA DANYCH FIREBASE

Firebase to platforma, której głównym celem jest wspomaganie programistów przy tworzeniu aplikacji. Platforma Firebase posiada darmowy plan dla użytkowników indywidualnych, natomiast są w nim ograniczenia dzienne wykonywanych akcji takich jak zapytania oraz zapisywanie dokumentów. W przypadku klientów korporacyjnych możliwe jest wykupienie pakietów, które będą umożliwiały korzystanie z bazy danych większemu gronu użytkowników. Zaletą Google Firebase jest bezproblemowe uruchomienie nawet na starszych urządzeniach, baza nie posiada wygórowanych wymagań sprzętowych oraz systemowych. Podłączenie bazy Firebase do autorskich aplikacji jest bardzo proste. Ogromną zaletą Firebase jest rozbudowana dokumentacja, z której mogą korzystać wszyscy programiści.

MODUŁY FIREBASE

Platforma Firebase posiada wiele funkcji, które są przydatne do budowania autorskich projektów. Wymienienie ich oraz opisanie nie jest konieczne, ponieważ istnieje rozbudowana dokumentacja <https://firebase.google.com/docs>. Najbardziej ciekawe funkcje to między innymi:

- **Authentication.** Prawie w każdej aplikacji jest konieczność tworzenia kont oraz uwierzytelniania użytkowników. Firebase udostępnia usługę, która weryfikuje tożsamość użytkowników przez e-mail i hasło, numer telefonu oraz konta w większych serwisach takich jak Google, Facebook, Apple, GitHub, Twitter, Microsoft.
- **Firestore Database.** Baza danych, która umożliwia tworzenie skalowalnej struktury, a dane organizowane są w kolekcje.
- **Realtime Database.** Baza danych czasu rzeczywistego, w której wszystkie dane magazynowane są jako obiekt JSON. Nie występują tutaj rekordy oraz tabele.
- **Machine Learning.** Baza danych Firebase zapewnia możliwość wykorzystania uczenia maszynowego w aplikacji. W kategorii uczenie maszynowe są dostępne następujące funkcje: rozpoznawanie tekstu z obrazków, identyfikacja obiektów z obrazków, wykrywanie twarzy ze zdjęć, wykrywanie i śledzenie obiektów z obrazów czy filmów, skanowanie kodów kreskowych, wykrywanie języka tekstu oraz tłumaczenie tekstu, inteligentne odpowiedzi tekstowe i identyfikacja znanych lokalizacji na obrazkach.

WIRTUALNA PRZYCHODNIA

Przykładowe struktury danych



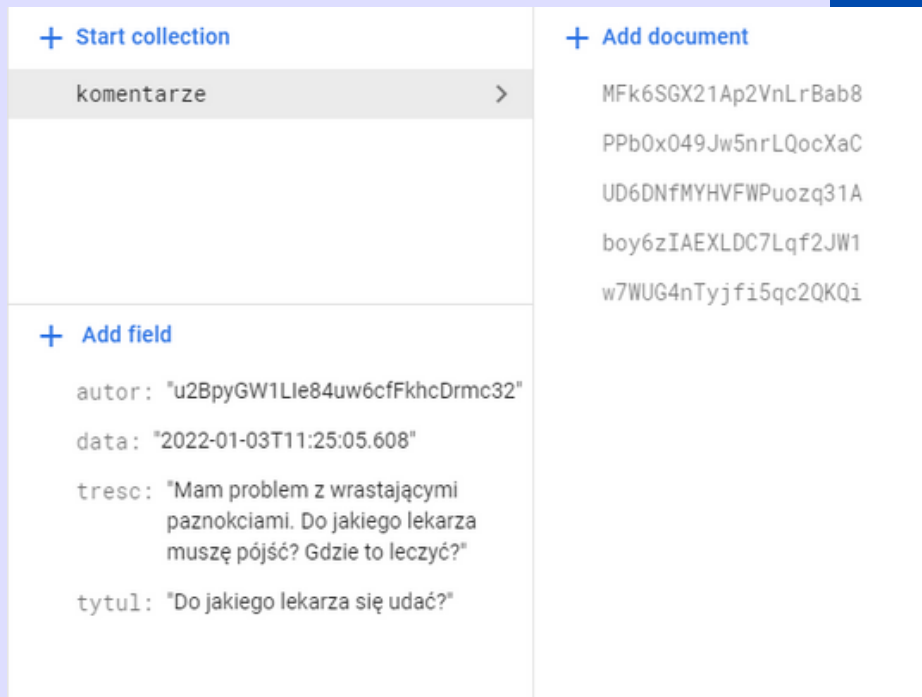
Rysunek 1. Baza danych dla zapisów

REALTIME DATABASE

Baza danych czasu rzeczywistego wykorzystywana jest głównie do aktywności informacjami, zapisy oraz do przechowywania danych o użytkownikach. Na rysunku 1 przedstawiona została struktura bazy danych, która wykorzystywana jest do zapisów online. Kluczami w zapisach w bazie danych są poszczególne daty, w których prowadzone są przyjęcia pacjentów. Każda data zawiera pola z informacją o rozpoczęciu wizyt danego dnia i zakończeniu przyjęć. Każda data ma kolejny podrzędny klucz wizyty, w których zapisane są zajęte już godziny przez innych pacjentów. W bazie danych zapisywana jest godzina oraz uid użytkownika, który umówił się na wizytę.

FIRESTORE DATABASE

Moduł Firestore Database wykorzystywany jest do tworzenia aktywności forum. Struktura bazy danych została przedstawiona na rysunku 2. Każdy temat to inny dokument, który zawiera autora, datę dodania, treść oraz tytuł. W każdym dokumencie tematu jest kolekcja komentarze. Ta kolekcja składa się z dokumentów komentarzy, które zawierają informacje o autorze, datę oraz treść. Jako autor dodawany jest unikalny identyfikator uid, żeby dane takie jak imię i nazwisko nie powtarzały się wiele razy w bazie danych. Za pomocą samego uid z każdej aktywności mogą zostać wczytane potrzebne dane użytkownika.



Rysunek 2. Baza danych dla forum

WIRTUALNA PRZYCHODNIA

Bezpieczeństwo aplikacji

Kwestia bezpieczeństwa aplikacji towarzyszyła podczas całego procesu powstawania poszczególnych funkcjonalności. W tym celu została zainicjowana baza danych Firebase, która umożliwia bezpieczne przechowywanie danych. Do aplikacji mają dostęp tylko uwierzytelnieni użytkownicy, którzy muszą zarejestrować swoje konto. Przy rejestracji użytkowników wykorzystywany jest moduł Authentication, który obsługuje rejestrację oraz logowanie za pomocą różnych metod. Wszystkie dane podawane podczas rejestracji są zapisywane w bezpieczny sposób przy wykorzystaniu odpowiednich funkcji w bazie Firebase. Dodatkowym zabezpieczeniem procesu dodawania nowych użytkowników do bazy jest odpowiednia walidacja w kodzie. Użytkownik musi podać poprawne dane, żeby pomyślnie zakończyć proces rejestracji.

Przy rejestracji wymagane jest podanie hasła, które zawiera minimum:

- od 9 do 32 znaków
- 1 literę wielką
- 1 literę małą
- 1 cyfrę
- 1 znak specjalny spośród następujących: @\$%*?#^()

Do wyznaczania, czy hasło jest bezpieczne służy wyznaczanie entropii. Wzór na entropię, który ma za zadanie wyznaczyć liczbę bitów przypadających na jeden znak to:

$$H(X) = \log_2 N$$

gdzie N oznacza liczbę wszystkich znaków w alfabecie.

Na początku trzeba obliczyć N, czyli zsumować wszystkie znaki, które mogą wystąpić w hasle. Alfabet polski składa się z 32 znaków, natomiast trzeba rozróżnić małe oraz wielkie litery. Do hasła może być użytych 10 cyfr oraz 9 znaków specjalnych. $N = 32 + 32 + 10 + 9 = 83$

Można zastosować wzór oraz podstawić do niego $N = 83$.

$$H(X) = \log_2 83 \approx 6,4$$

Oznacza to, że na jeden znak w hasle przypada około 6,4 bitów. Następnie należy obliczyć, ile bitów znajduje się w hasle składającym się z 9 znaków. Wzór na ilość informacji w hasle to:

$$I(X_k) = k * H(X)$$

gdzie k to liczba znaków w hasle.

W aplikacji Wirtualna Przychodnia użytkownik powinien podać hasło, które zawiera minimum 9 znaków, dlatego $k = 9$.

$$I(X_k) = 9 * 6,4 \approx 57,6 \text{ bitów}$$

Powszechnie przyjęte jest, że hasło, które zawiera powyżej 56 bitów jest uznawane za hasło bezpieczne. Powyższe przykłady pozwalają wyciągnąć wniosek, że zastosowanie hasła od 9 do 32 znaków, które składa się z różnorodnych znaków, jest bezpieczne dla użytkownika.

WIRTUALNA PRZYCHODNIA

Rejestracja

REJESTRACJA

Użytkownik po włączeniu aplikacji widzi ekran powitalny, który zawiera najważniejsze informacje takie jak nazwa aplikacji, krótki opis oraz przycisk START. Przycisk przenosi użytkownika do modułu logowania i rejestracji. Jeżeli użytkownik jest już zalogowany, przycisk przeniesie go do menu z głównymi opcjami programu. Użytkownik musi posiadać konto, żeby skorzystać z aplikacji Wirtualna Przychodnia. W tym celu musi dokonać rejestracji poprzez specjalny formularz (rysunek 3). W formularzu użytkownik musi podać swoje dane takie jak: imię, nazwisko, nazwę użytkownika, adres e-mail, numer telefonu oraz hasło.

The screenshot shows a registration form titled "WirtualnaPrzychodnia". It contains several input fields for user data: "Imię", "Nazwisko", "E-mail", "Numer telefonu", "Nazwa użytkownika", "Hasło", and "Powtórz hasło". Below the fields are two buttons: "ZAREJESTRUJ SIĘ" and "MASZ JUŻ KONTO? PRZEJDŹ DO LOGOWANIA".

Rysunek 3. Rejestracja

This screenshot shows the registration form with sample data entered: "Justyna", "Winiarczyk", "juswin@ex.pl", "123123123", and "juswin". The password field is masked with dots. A red exclamation mark icon indicates an error. A tooltip message states: "Hasło musi mieć od 9 do 32 znaków, w tym 1 literę wielką, 1 literę małą, 1 cyfrę i 1 znak specjalny spośród następujących: [@\$%*?#^()]". The "MASZ JUŻ KONTO? PRZEJDŹ DO LOGOWANIA" button is visible at the bottom.

WIRTUALNA PRZYZCHODNIA

Rejestracja

WALIDACJA WPROWADZANYCH DANYCH

Dla dobrego działania modułu rejestracji, została stworzona odpowiednia walidacja wprowadzanych danych:

- pole „**Imię**” dopuszcza tylko imiona wpisane wielką literą. Nie dopuszcza innych znaków niż litery. Walidacja przewiduje, że użytkownik może mieć podwójne imię pisane z łącznikiem;
- pole „**Nazwisko**” dopuszcza tylko nazwiska wpisane wielką literą. Wprowadzenie innych znaków niż litery będzie błędem. Użytkownik może wprowadzić również dwuczłonowe nazwisko pisane z łącznikiem;
- pole „**Nazwa użytkownika**” dopuszcza dowolny ciąg znaków. Jedynym warunkiem jest unikalność w aplikacji. Jeżeli użytkownik wprowadzi nazwę, która występuje już w bazie danych, zostanie wyświetlony błąd;
- pole „**E-mail**” przyjmuje tylko poprawne adresy email, które zawierają wielkie i małe litery, cyfry, jeden znak „@” oraz domenę;
- pole „**Numer telefonu**” sprawdza, czy w podanym ciągu występują same cyfry. Sprawdzana zostaje również liczba wprowadzonych cyfr, która powinna wynosić 9;
- pole „**Hasło**” zgodnie z powszechnymi zasadami tworzenia bezpiecznych haseł, dopuszcza wprowadzenie hasła, które zawiera od 9 do 32 znaków. Powinno być najbardziej zróżnicowane, dlatego powinno zawierać minimum: jedną małą literę, jedną wielką literę, jedną cyfrę oraz jeden znak specjalny spośród następujących: @\$%*?#^();
- pole „**Powtórz hasło**” wymaga wprowadzenia identycznego hasła, które zostało podane w pierwszym polu z hasłem;
- dodatkowo żadne z wyżej wymienionych pól nie może być puste.

```
/* Sprawdzanie czy pole nazwisko nie jest puste */

if (TextUtils.isEmpty(nazwisko!!.text.toString().trim())) {
    nazwisko!!.setError(getString(R.string.rejestracja_puste_nazwisko))
    nazwisko!!.requestFocus()
    return@OnClickListener
}
/* Sprawdzanie czy pole nazwisko nie zawiera żadnych błędów
 * Zgodnie z zasadami nazwiska mogą być maksymalnie dwuczłonowe
 * oraz pisane łącznikiem
 * */
if (!"[A-ZĄĆĘŁŃÓŚŻ]{1}[a-
ząćęłńóśż]+"
.toRegex().matches(nazwisko!!.text.toString().trim())
&& (!"[A-ZĄĆĘŁŃÓŚŻ]{1}[a-ząćęłńóśż]+[-]{1}[A-
ZĄĆĘŁŃÓŚŻ]{1}[a-
ząćęłńóśż]+"
.toRegex().matches(nazwisko!!.text.toString().trim())
&& (!"[A-ZĄĆĘŁŃÓŚŻ]{1}[a-ząćęłńóśż]+[ ]{1}[A-
ZĄĆĘŁŃÓŚŻ]{1}[a-
ząćęłńóśż]+"
.toRegex().matches(nazwisko!!.text.toString().trim())) {
    nazwisko!!.setError(getString(R.string.rejestracja_bledne_nazwisko))
    nazwisko!!.requestFocus()
    return@OnClickListener
}
```

Rysunek 4. Przykładowy kod do walidacji

WIRTUALNA PRZYCHODNIA

Menu i profil użytkownika



Rysunek 5. Menu

M E N U

Moduł menu zawiera spis wszystkich funkcjonalności aplikacji Wirtualna Przychodnia oraz dwa intuicyjne przyciski profil i wyloguj się. Przycisk wyloguj się służy do wylogowania z obecnie zalogowanego konta użytkownika. Przycisk profil przenosi użytkownika do aktywności, która zawiera dane podane przy logowaniu (rysunek 6).

Imię	Jan
Nazwisko	Kowalski
Numer telefonu	123123111
ZAPISZ	
POWRÓT DO PROFILU	
Hasło	
Powtórz hasło	
Wprowadź obecne hasło	
ZMIEN HASŁO	

P R O F I L U Ż Y T K O W N I K A

Użytkownik w tym module może edytować swoje dane. Edycja profilu została podzielona na dwie części. Jedna część zawiera edycję danych, natomiast druga część to zmiana hasła. Użytkownik w pierwszej sekcji może edytować imię, nazwisko oraz numer telefonu. Do zmiany hasła użytkownik musi wprowadzić nowe hasło, następnie powtórzyć nowe hasło oraz wprowadzić obecne hasło. W przypadku jeżeli nowe hasło nie spełnia wymagań, wyświetla się odpowiedni komunikat. Walidacja dla hasła jest dokładnie taka sama jak przy rejestracji nowego użytkownika. Tak samo dzieje się w przypadku, gdy nowe hasło oraz jego powtórzenie się od siebie różnią. Następnie sprawdzana jest poprawność obecnego hasła. Wprowadzenie obecnego hasła jest konieczne ze względów bezpieczeństwa użytkownika.

Imię	Jan
Nazwisko	Kowalski
Nazwa użytkownika	JanKowalski
Numer telefonu	123123111
E-mail	jan.kowalski@gmail.com
EDYTUJ SWOJE DANE	

Rysunek 6. Profil użytkownika

WIRTUALNA PRZYCHODNIA

Forum i komentarze



Rysunek 7. Widok aktywności forum

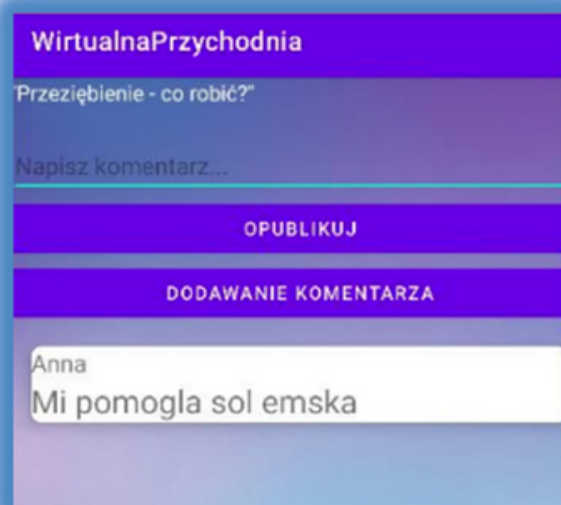
F O R U M

Po wybraniu przycisku forum, który znajduje się w aktywności menu, następuje otworenie listy, która zawiera tematy z bazy danych. Każdy temat wypisany jest na innym obiekcie, który tworzony jest dynamicznie. Taki mechanizm jest możliwy dzięki zastosowaniu RecyclerView, do jego działania potrzebny jest adapter oraz odpowiednie funkcje.

Na liście wypisywane są tematy oraz nazwy użytkowników, którzy dodali wątek na forum. Na samej górze aktywności znajduje się przycisk dodawanie tematu, który jak sama nazwa wskazuje, przenosi użytkownika do aktywności, w której może skonfigurować swój wątek. W każdy temat można kliknąć, po tej czynności następuje otworenie sekcji komentarzy do konkretnego tematu.

K O M E N T A R Z E

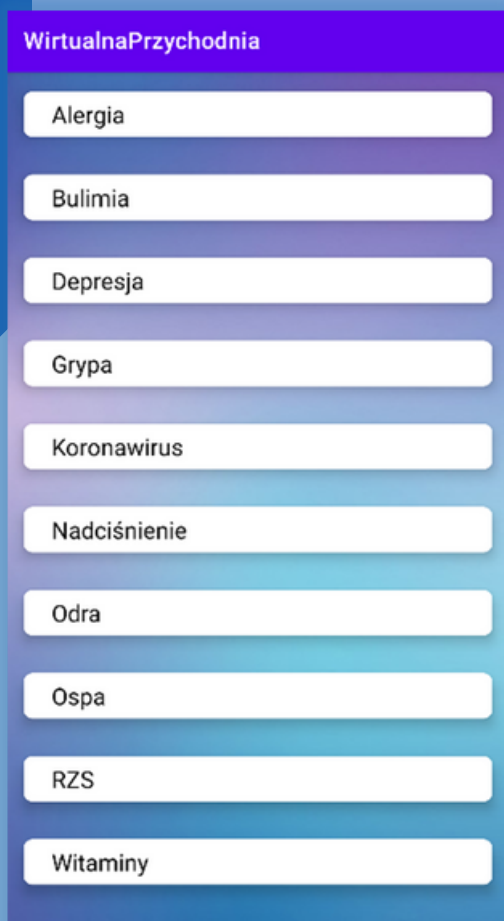
W aktywności z komentarzami wypisane są dane takie jak tytuł oraz dokładny opis wątku. Pod tymi danymi znajduje się przycisk dodawanie komentarza, a pod spodem znaczną część widoku zajmuje dynamiczna lista z komentarzami. Lista komentarzy tworzona jest w podobny sposób do listy tematów, również wykorzystywany jest RecyclerView oraz adapter. Jeżeli użytkownik chce dodać odpowiedź, to musi kliknąć wspomniany wcześniej przycisk, który rozwinie layout, w którym znajduje się pole do wpisywania tekstu komentarza oraz nowy przycisk opublikuj. Zastosowany jest taki mechanizm, żeby sekcja dodawania komentarzy nie zajmowała na ekranach części widoku użytkownikom, którzy czytają forum.



Rysunek 8. Widok sekcji komentarzy

WIRTUALNA PRZYCHODNIA

Informacje



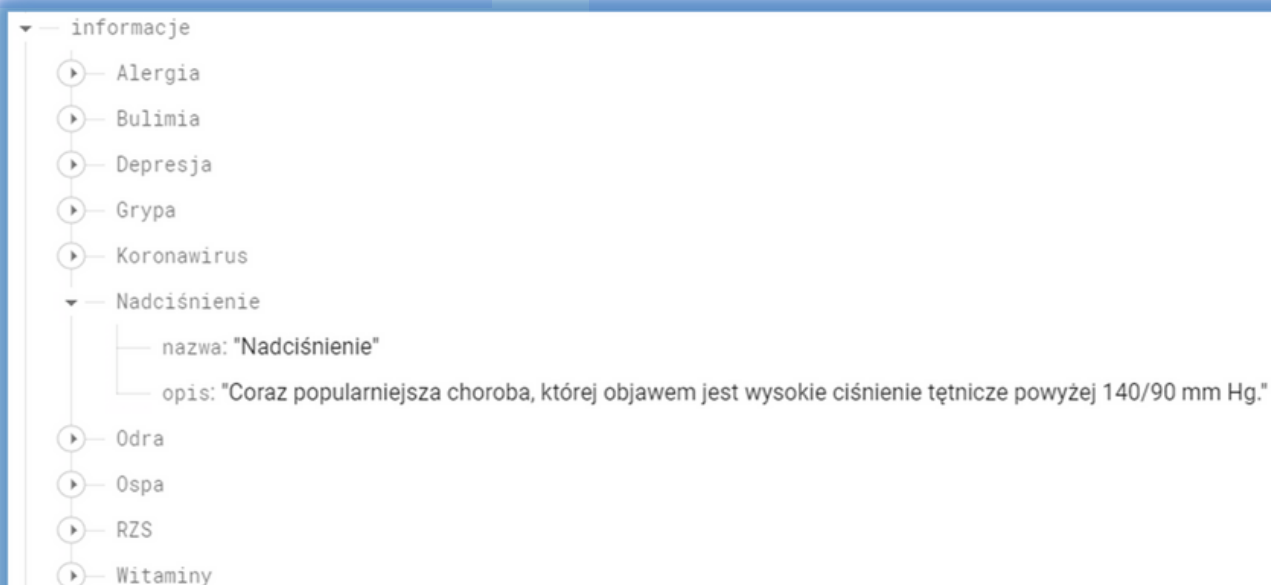
Rysunek 9. Widok informacji

INFORMACJE

Kolejna główna aktywność to informacje. Po wejściu w aktywność użytkownikowi wyświetla się spis różnych zagadnień w tematyce zdrowotnej.

Każde wypisane zagadnienie można kliknąć, żeby zobaczyć szczegóły. W tym celu tworzy się nowa aktywność, w której zostaje wyświetlone zagadnienie oraz jego krótkie omówienie, żeby w jak najprostszy sposób przybliżyć znaczenie tego hasła. Hasła wyświetlane są w kolejności alfabetycznej.

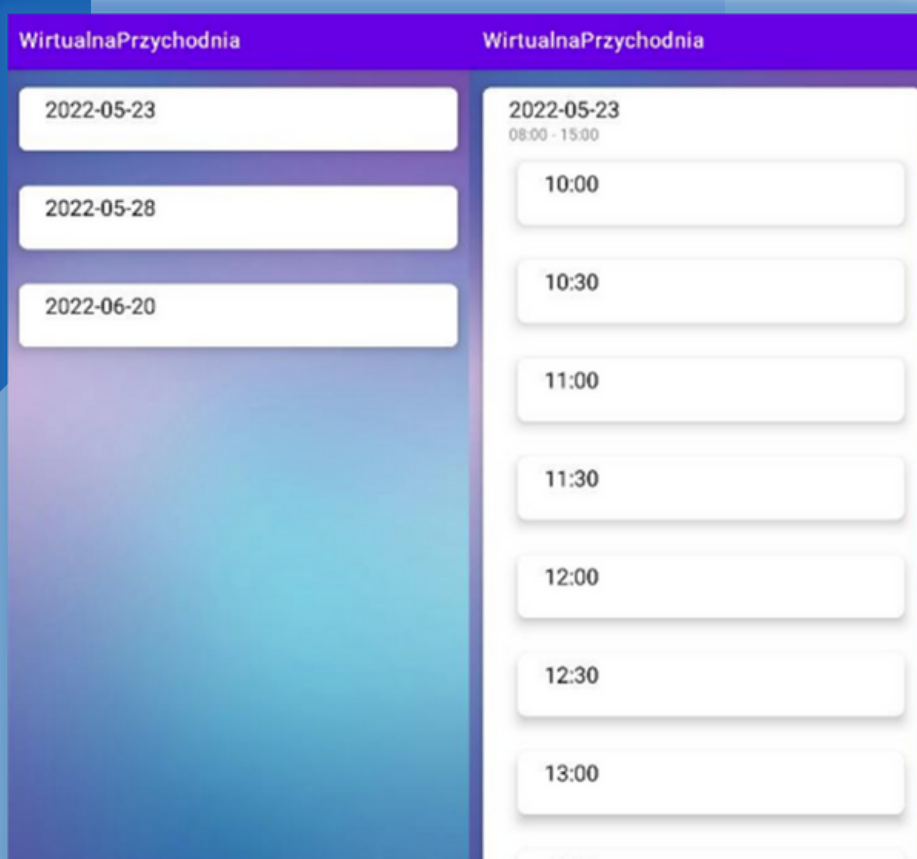
Informacje przechowywane są za pomocą Firebase Realtime Database. Informacje wypisywane są w aplikacji przy użyciu adapterów oraz dynamicznej listy – RecyclerView, która w zależności od ilości danych w bazie danych, tworzy odpowiednio elementy.



Rysunek 10. Informacje w bazie Firebase

WIRTUALNA PRZYCHODNIA

Zapisy online



Rysunek 11. Widok zapisów online

ZAPISY

Zapisy online służą do zapisów pacjentów na wizytę na wybrany dzień oraz wybraną godzinę. Po wejściu przez użytkownika do tej aktywności wyświetla się dynamiczna lista stworzona przez zastosowanie RecyclerView oraz adaptera. Stworzone zostały funkcje, które w odpowiedni sposób odczytują daty przyjęć z bazy danych, następnie odczytuje dane start oraz stop, czyli godziny przyjęć w konkretnym dniu. Głównym założeniem przy tym mechanizmie jest wypisanie dat z najbliższych 30 dni, w które prowadzone są przyjęcia pacjentów. Pacjent po przeglądaniu dat może kliknąć w wybrany termin, żeby rozwinąć listę z wolnymi godzinami w tym dniu. Widok zapisów jest przedstawiony na rysunku 11.

WIRTUALNA PRZYCHODNIA

Zapisy online

```
var database2 = FirebaseDatabase.getInstance().getReference( path: "zapisy")

potwierdzenie.setOnClickListener { it: View!
    database2.child(data.toString()).child( pathString: "wizyty").child(godzina).addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {

            if (snapshot.exists()) {
                val informacja = snapshot.getValue(Godziny::class.java)
                // textdane.setText(informacja?.pacjent)
                komunikat.setText("Przepraszamy. Podana godzina została właśnie zajęta. Wróć do poprzedniej strony")
            } else {
                database2!!.child(data.toString()).child( pathString: "wizyty").child(godzina).child( pathString: "pacjent").setValue(uid)
                database2!!.child(data.toString()).child( pathString: "wizyty").child(godzina).child( pathString: "godzina").setValue(godzina)

                komunikat.visibility = View.GONE
                komunikat2.setText("Wizyta w dniu " + data + " o godzinie " + godzina + " została potwierdzona.")
                powrotmenu.visibility=View.VISIBLE
            }
        }
    })
}
```

Rysunek 12. Kod odpowiadający za weryfikację wolnych terminów

DZIAŁANIE KODU

Lista z godzinami tworzona jest w dynamiczny sposób dla każdej daty. W tym celu stworzone zostały dwie listy. Wypełniane są poprzez pętle while oraz for. Jedna tablica zawiera wszystkie godziny pomiędzy godziną rozpoczęcia oraz zakończenia przyjęć, które są w odpowiednio zadeklarowanym odstępie czasu. Druga tablica zawiera tylko zajęte już godziny, które wczytywane są z bazy. Następnie wykorzystuję wbudowaną funkcję list, minus – która od listy wszystkich godzin odejmuje listę zajętych godzin. W ten sposób tworzona jest trzecia lista, która zawiera listę wolnych godzin.

Po kliknięciu na wybraną datę rozwinię się lista z godzinami, które są wolne tego dnia. Jeżeli użytkownik będzie przeglądał dzisiejszą datę, to odpowiednia funkcja odpowiada za ukrywanie godzin, które już minęły, aby nie można było się zapisać. Po wybraniu pasującej daty oraz godziny, użytkownik przechodzi do podsumowania wizyty. Wyświetlone zostają jego dane, musi je zweryfikować. Jeżeli dane się nie zgadzają ze stanem faktycznym, musi wrócić do menu oraz je edytować. Jeżeli wyświetlone dane są poprawne, to może kliknąć zapisz się na wizytę. Zostanie wyświetlony komunikat, że wizyta konkretnego dnia o wybranej porze została potwierdzona. Wyświetli się także przycisk powrotu do menu. W przypadku, gdy inny użytkownik również wybierze tą samą godzinę, odpowiednia funkcja (rysunek 12) przy kliknięciu sprawdza, czy wybrany termin jest dalej wolny. Jeżeli okaże się, że termin został zarezerwowany przez kogoś innego, wyświetlony jest czytelny komunikat, że podana godzina została właśnie zajęta. Użytkownik powinien wtedy wrócić do listy i wybrać inną godzinę.

WIRTUALNA PRZYCHODNIA

Podsumowanie

ZREALIZOWANIE CELU

Głównym celem projektu Wirtualna Przychodnia było stworzenie aplikacji mobilnej, z której będą mogły korzystać osoby młodsze oraz osoby starsze, które często nie są zaawansowane w obsłudze aplikacji oraz urządzeń mobilnych. W tym celu wygląd aplikacji został specjalnie dostosowany do zróżnicowanego grona odbiorców. Czcionki są zwykłe, czytelne, wszystkie przyciski oraz opcje są łatwe do zauważenia oraz dobrze opisane. Założony cel został zrealizowany.

KORZYŚCI Z PROJEKTU

- Udoskonalenie umiejętności programowania w Kotlinie.
- Usystematyzowanie wiedzy na temat bazy danych Firebase oraz metod pobierania danych z bazy.
- Nauka wyszukiwania oraz rozwiązywania błędów w środowisku programistycznym.
- Korzystanie z Trello, narzędzia służącego do organizacji pracy.

WNIOSKI

- Zastosowanie informatyki w medycynie oraz ochronie zdrowia usprawnia wiele zadań oraz wnosi wiele pozytywnych aspektów do tej dziedziny naukowej. Głównymi zaletami jest szybsze, dokładniejsze oraz bardziej efektywne gromadzenie danych, które mogą być wykorzystywane do leczenia pacjentów, a także do wykonywania badań nad chorobami.
- Dokładne poddanie analizie zastosowania technologii informatycznych w medycynie i służbie zdrowia pozwala wyciągnąć wnioski, że w ciągu najbliższych lat, prace nad wdrażaniem nowych rozwiązań bazujących na informatyce będą prowadzone na jeszcze większym poziomie niż do tej pory.