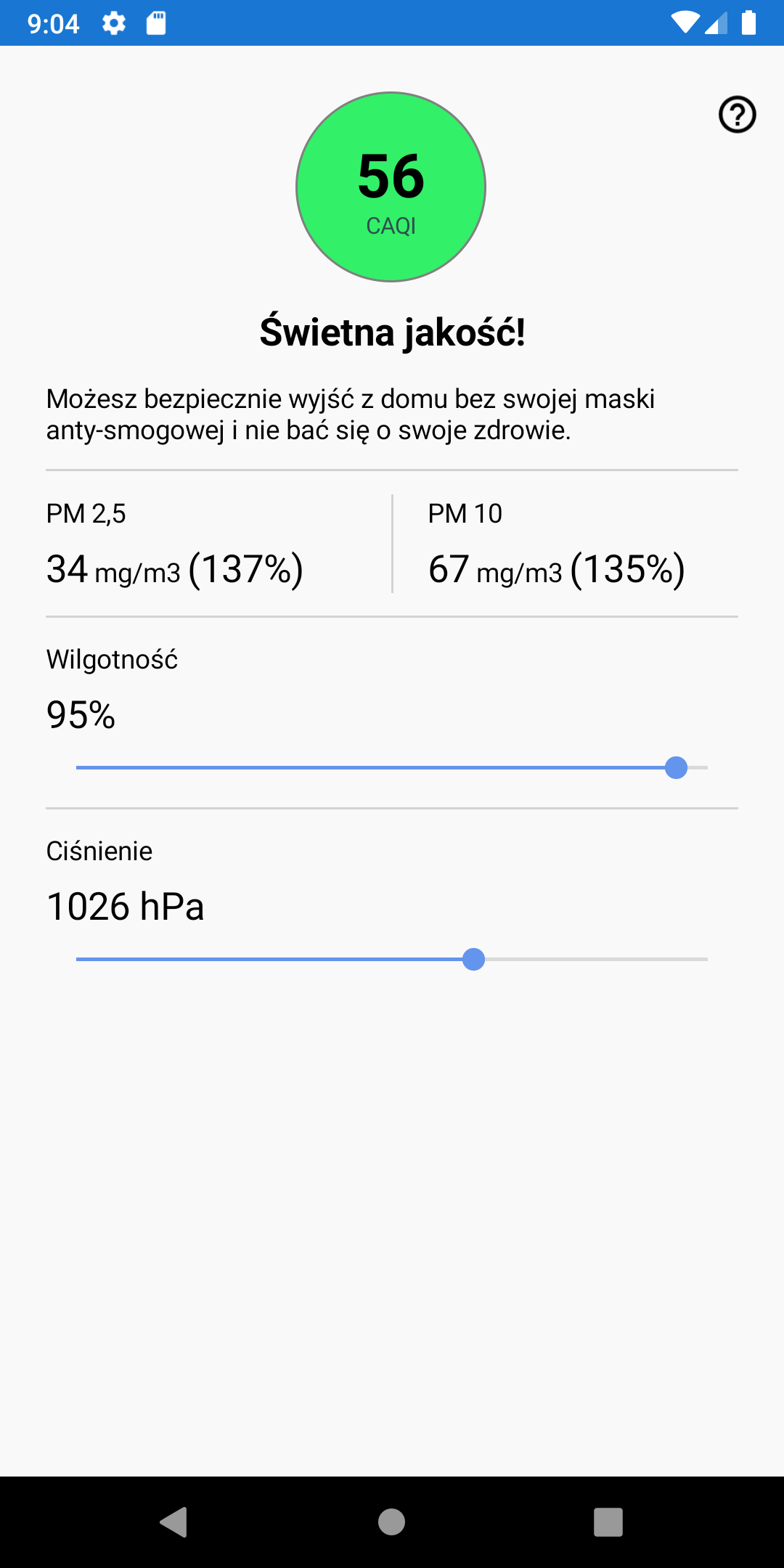
# Laboratorium 1

Zagadnienia

* Warunki zaliczenia
* Wprowadzenie do Xamarin.Forms
* Stworzenie pierwszego projektu i omówienie struktury
* Stworzenie strony z podstawowymi kontrolkami
* Podstawy gita i GitHuba

Zadania

1. Stworzenie pierwszego projektu Xamarin.Forms
   1. W Visual Studio 2019 kliknij „Create a new project” i wyszukaj “Xamarin.Forms”
   2. Wybierz “Mobile App (Xamarin.Forms)” I kliknij Next
   3. Wpisz nazwę projektu I kliknij Create
   4. W kolejnym okienku wybierz szablon „Blank” i zaznacz platformy Android oraz iOS. Kliknij Ok.
   5. Wybierz projekt Androida jako startowy i uruchom go w emulatorze.
2. Przejrzyj nowy projekt i zapoznaj się z jego elementami
   1. Projekt „AirMonitor”:
      1. App.xaml – plik z zasobami globalnymi na poziomie aplikacji
      2. App.xaml.cs – punkt wejściowy aplikacji Xamarin.Forms, ustawiamy mu stronę startową, możemy też zareagować na zdarzenia cyklu życia aplikacji
      3. MainPage.xaml – Przykładowa główna strona aplikacji (plik definiowania wyglądu)
      4. MainPage.xaml.cs - Przykładowa główna strona aplikacji (plik z logiką strony)
   2. Projekt „AirMonitor.Android”:
      1. Properties/AndroidManifest.xml – podstawowy plik konfiguracyjny aplikacji Android, dostępny również w formie graficznej z poziomu właściwości projektu
      2. Assets – folder przeznaczony do zasobów i mediów, np. pliki dźwiękowe, czy tekstowe.
      3. Resources – folder, w którym umieszczamy większość zasobów aplikacji, jak ikony, teksty, style, wygląd stron, itd.
      4. MainActivity.cs – Aktywność, punkt wejściowy aplikacji Android
   3. Projekt „AirMonitor.iOS”:
      1. Asset Catalogs – miejsce, gdzie dodajemy zasoby aplikacji, jak np. ikony
      2. Resources – folder na zasoby, które nie pasują do Asset Catalog, jak launch screen, czcionki, czy pliki dźwiękowe lub konfiguracyjne
      3. Resources/LaunchSreen.storyboard – launch screen/splash screen aplikacji, pierwszy ekran, który użytkownik widzi, po kliknięciu w ikonę aplikacji
      4. AppDelegate.cs – klasa przetwarzająca zdarzenia systemowe
      5. Entitlements.plist – plik, w którym definiujemy dodatkowe usługi i funkcjonalności aplikacji, np. korzystanie z iCloud, powiadomień push, HomeKit
      6. Info.plist – plik z ustawieniami aplikacji, np. wersja, nazwa
      7. Main.cs – punkt wejściowy aplikacji iOS
3. Zapoznaj się z narzędziami kontroli wersji w Visual Studio.
   1. Dodaj projekt do kontroli wersji (git):
      1. File -> Add to Source Control
   2. Dostosuj opcje gita:
      1. W oknie Team Explorer wybierz Settings -> Repository Settings i wprowadź swoje dane (imię i email), które będą powiązane z commitami.
   3. Pierwszy commit:
      1. Zrób pierwszą zmianę w kodzie (np. Zmień tekst w MainPage.xml)
      2. W Team Explorer wybierz Changes i przejrzyj zmiany.
      3. Zmiany mogą być w dwóch stanach przed commitem: unstaged i staged. Zmiany w stanie unstaged nie zostaną zacommitowane, musimy więc przenieść je to stanu staged. Zrobimy to plusem na prawo od napisu Changes lub w menu PPM.
      4. Każdy commit musi mieć wiadomość. Wpisujemy wiadomość w polu na górze.
      5. Klikamy Commit Staged.
   4. Pamiętajmy, żeby robić commity regularnie, np. po każdej większej zmianie, czy skoczonej funkcjonalności (lub jeśli jest ona duża to również w trakcie).
   5. Pushowanie zmian:
      1. Commity są u nas lokalnie na komputerze. Musimy wypushować zmiany na serwer, żeby inni mogli je pobierać.
      2. Załóż konto na GitHubie: <https://github.com>
      3. Stwórz tam repozytorium (publiczne)
      4. Skopiuj adres repozytorium w formacie: <https://github.com/[nazwa_uzytkownika]/[nazwa_repozytorium].git>)
      5. W Visual Studio w Team Explorer -> Settings -> Repository Settings -> Remotes dodaj adres z nazwą “origin” (bez cudzysłowów)
      6. W Team Explorer wybierz Sync i w sekcji Outgoing commits kliknij Push
      7. Będziesz musiał podać login i hasło do GitHuba. Po tym kroku twoje commity będą dostępne z portalu.
   6. Projekt z zajęć i własny powinny być na bieżąco aktualizowane na GitHubie.
4. Stworzenie strony szczegółów z podstawowymi kontrolkami
   1. Wygląd strony jak na zrzucie poniżej
   2. Użyte kontrolki: ContentPage, ScrollView, StackLayout, Grid, Frame, Label, Span, Image, BoxView, Slider
   3. Podpowiedzi:
      1. Przetestuj aplikację na urządzeniu z mały ekranem/rozdzielczością. Czy zawartość można przewinąć tak, żeby zobaczyć to co nie mieści się na ekranie?
      2. Całość ułożona jest od góry do dołu w liniach z odstępami. Który layout ma takie zachowanie?
      3. Jeden z layoutów ma właściwość do zaokrąglenia rogów - możesz go użyć do widoku z zielonym kółkiem.
      4. Label może wyświetlać sforamtowany tekst (np. część pogrubiona lub z innym rozmiarem czcionki). Użyj tej właściwości zamiast tworzyć kilka Labeli obok siebie.
      5. W Sliderze dla ciśnienia będziesz musiał ustawić wartość minimalną np. na 900 i maksymalną np. na 1100. Czy kolejność tych właściwości ma znaczenie?
      6. Widoki dla PM 2,5 oraz PM 10 mają równą szerokość i pionową linię między nimi. Jakiego layoutu użyjesz? Musisz zdefiniować 3 kolumny.
      7. Ikonka “?” po prawej stronie to przycisk - niech po naciśnięciu wyświetla popup z przykładowym wyjaśnieniem czym jest CAQI. Możesz tu użyć wbudowanej metody DisplayAlert.
      8. Podwidoki dla PM 2,5, PM 10 i kolejnych wyglądają tak samo - mają tytuł i pewną zawartość - stwórz dla nich nową kontrolkę (w oddzielnym pliku) i zaembedu w widoku głównym. W kontrolce użyj BindableProperty w dwóch właściwościach - jedna do zbindowania tytułu, druga do zbindowania dowolnej zawartości (możesz to zostawić na koniec).



Przydatne materiały:

* Kontrolki (strony, layouty, widoki): <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/user-interface/controls/>
* Formatowanie tekstu w Label: <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/user-interface/text/label#formatted-text>
* Własne kontrolki: <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/user-interface/layouts/contentview>
* Podstawy XAML: <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/xaml/xaml-basics/>

Dodatkowe materiały:

* Cykl życia aplikacji Xamarin.Forms: <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/app-fundamentals/app-lifecycle>
* Android Resources: <https://docs.microsoft.com/pl-pl/xamarin/android/app-fundamentals/resources-in-android/?tabs=windows>
* Android Manifest w Xamarinie: <https://docs.microsoft.com/pl-pl/xamarin/android/platform/android-manifest>