

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynieryjnych
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

Aplikacja Żabka finder

Autor:
Szymon Ząbczyk
Rafał Grzegorzek

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2023

Spis treści

1. Ogólne określenie wymagań	3
2. Określenie wymagań szczegółowych	4
3. Projektowanie	5
3.1. Android studio	5
3.2. Github i Github Desktop	7
4. Implementacja	9
4.1. Pliki layout'u	9
4.1.1. activity_start	9
5. Testowanie	10
6. Podręcznik użytkownika	11
Literatura	12
Spis rysunków	13
Spis tabel	14
Spis listingów	15

1. Ogólne określenie wymagań

- Aplikacja powinna korzystać z aktualnej lokalizacji użytkownika, a za pomocą Google Maps wyświetlać najbliższe sklepy spożywcze Żabka.



Rys. 1.1. Google maps

- Użytkownik powinien mieć możliwość wyświetlania sklepów spożywczych Żabka na mapach wraz z ich odległością od aktualnej lokalizacji.
- Aplikacja powinna umożliwiać użytkownikom przeglądanie szczegółów sklepów, takich jak godziny otwarcia, dostępne produkty i oceny klientów, po stronie Google Maps.
- Aplikacja umożliwia skanowanie kodów QR do sprawdzania najnowszych promocji i cen w sklepach Żabka.
- Klienci powinni mieć możliwość zaplanowania trasy do wybranego sklepu spożywczego za pomocą Google Maps.
- Użytkownicy powinni mieć możliwość zapisywania ulubionych sklepów spożywczych i otrzymywania powiadomień o promocjach lub specjalnych ofertach.

2. Określenie wymagań szczegółowych

- Analiza wymagań: Przeprowadzenie szczegółowej analizy wymagań klienta i przetworzenie ich na konkretny plan projektu.
- Projektowanie interfejsu użytkownika: Zaplanowanie interfejsu użytkownika, uwzględniając ergonomię, intuicyjność i estetykę, aby zapewnić jak najlepsze doświadczenie dla użytkownika.
- Konfiguracja środowiska : Utworzenie projektu Android Studio w odpowiednim IDE, takim jak Visual Studio, skonfigurowanie narzędzi i środowiska pracy.



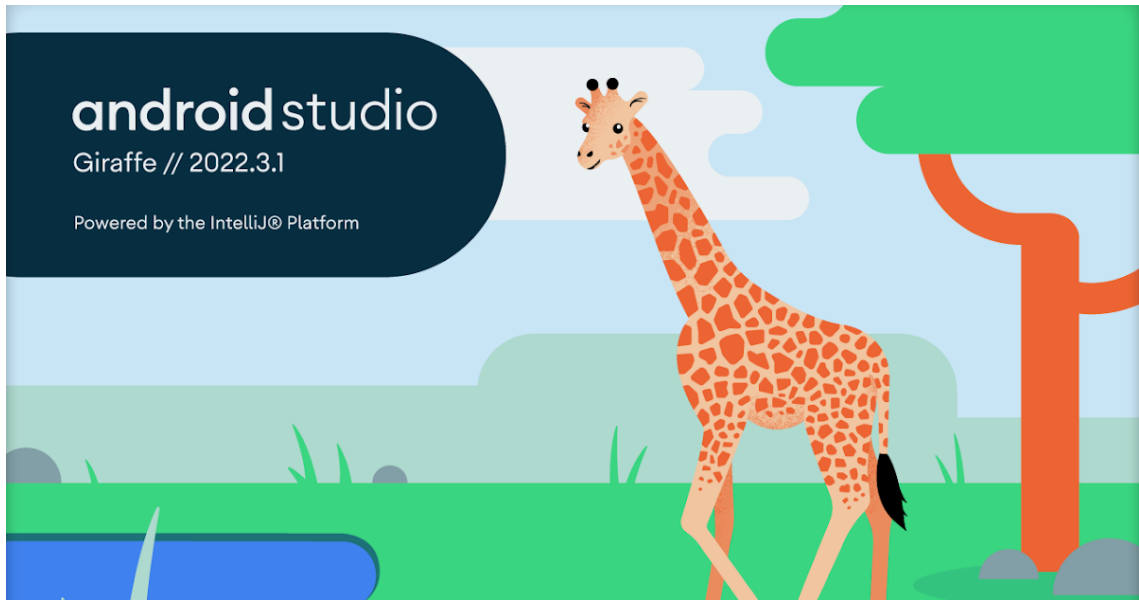
Rys. 2.1. Android Studio

- Integracja z aparatem: Wykorzystanie funkcji aparatu do skanowania kodów QR produktów, oraz przekierowywanie do stron WWW.
- Integracja z usługami map: Wykorzystanie usług map, takich jak Google Maps API lub Mapy Apple, w celu wyświetlania lokalizacji sklepów spożywczych oraz planowania tras do nich.
- Testowanie: Regularne testowanie aplikacji w celu zapewnienia, że wszystkie funkcje działają zgodnie z oczekiwaniami i nie ma błędów.
- Optymalizacja: Ulepszanie wydajności i responsywności aplikacji poprzez optymalizację kodu i zasobów, aby zapewnić płynne działanie.

3. Projektowanie

3.1. Android studio

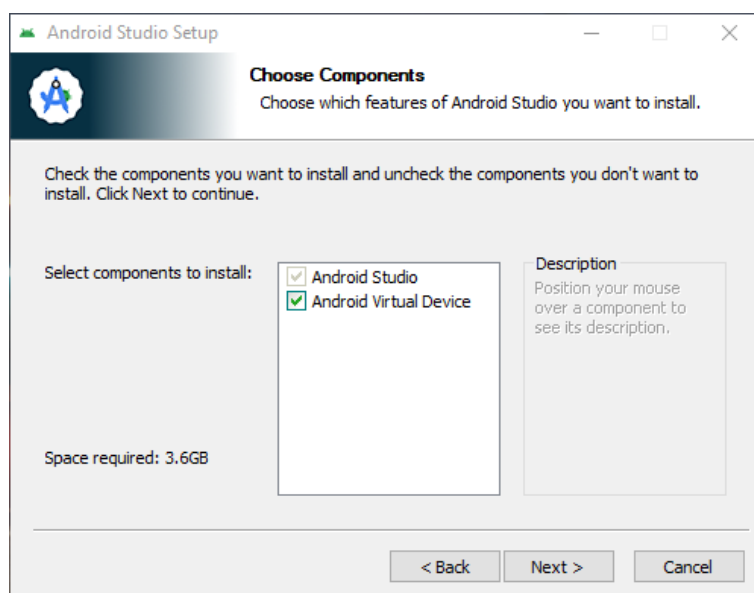
Podczas projektowania aplikacji mobilnej, użyliśmy przydatnego narzędzia jakim jest, Android Studio[1] version - Giraffe — 2022.3.1.



Rys. 3.1. Android Studio Giraffe

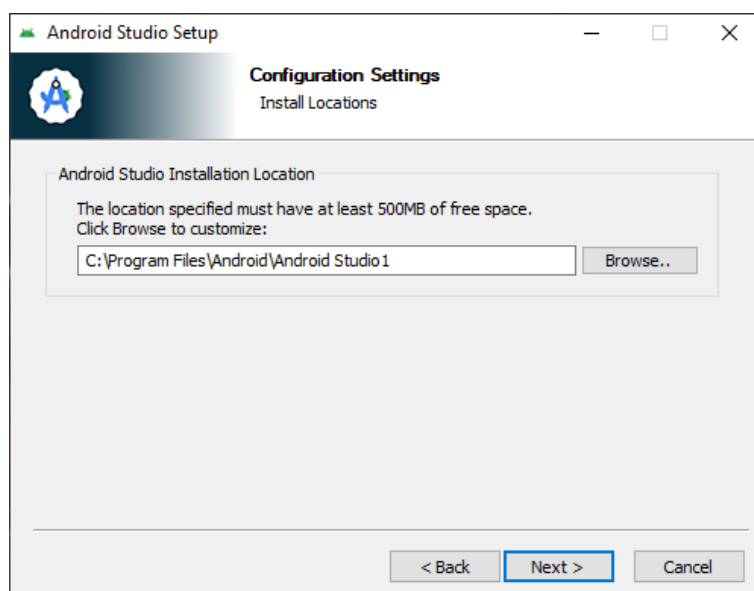
Będąc na stronie pobieramy aplikację przyciskiem "Download Android Studio Hedgehog". *(Android Studio to aplikacja ciągle się rozwijająca, więc możliwe jest że po wejściu na stronę Android Studio, przycisk pobierania pobierze nam aktualnie najnowszą wersję oprogramowania. Jeżeli chcielibyśmy wybrać konkretną wersję Android Studio, np: Giraffe/2022.3.1, musimy poszukać w wersjach archiwalnych. Tutaj znajduje się link do strony)[2].*

Po przeczytaniu umowy licencyjnej, zaznaczamy przycisk i klikamy Download Android Studio. Po wskazaniu miejsca do instalacji, rozpocznie się proces pobierania.



Rys. 3.2. Android Studio Instalacja

W oknie instalatora, jako pierwsze ukazały się nam komponenty, które chcemy zainstalować wraz z narzędziem. W tym przypadku jesteśmy pytani czy chcemy aby obok aplikacji utworzyła się również domyślna wirtualna maszyna.



Rys. 3.3. Android Studio Instalacja

Następnie wybieramy miejsce na dysku (przynajmniej 500MB wolnego miejsca), gdzie znajdować się będą nasze aplikacje. Klikamy "next", później "Install" i oczekujemy zakończenia procesu instalacyjnego.

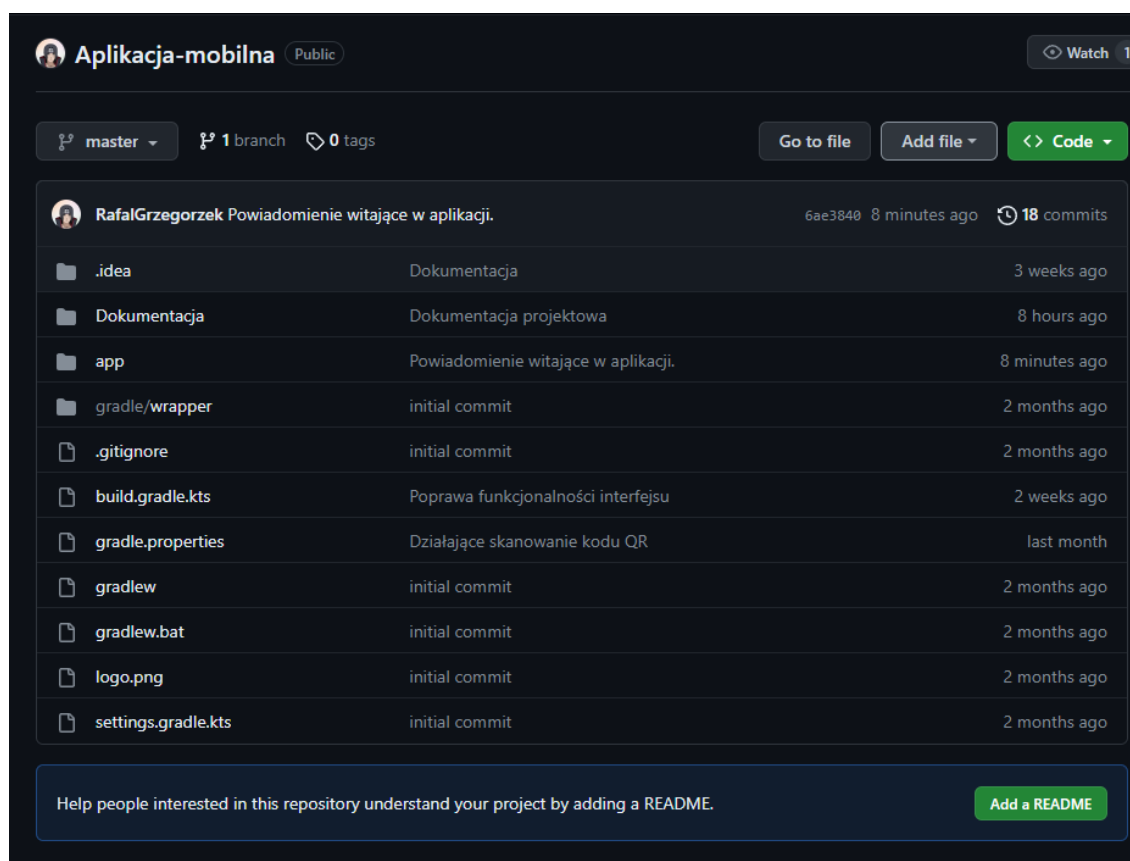
3.2. Github i Github Desktop

Do zarządzania projektem, niezbędną okazała się usługa internetowa do kontroli wersji GitHub.



Rys. 3.4. GitHub

GitHub bardzo ułatwia kooperację i pracę w zespołach. Dzięki jego rozbudowanym funkcjom w prosty sposób jesteśmy w stanie pracować z ludźmi nie martwiąc się o zgubienie lub zniszczenie kodu, ponieważ jest on bezpiecznie przechowywany w chmurze i tylko czeka na dopisywanie i udoskonalanie. Na samym początku tworzymy repozytorium - czyli wirtualny folder w którym będą przechowywane wszystkie commitowane przez nas pliki. Do nowo utworzonego repozytorium, dołączamy pliki aplikacji.



Rys. 3.5. GitHub repozytorium

Pliki możemy załączyć przez stronę, naciskając przycisk "Add file".



Rys. 3.6. GitHub Desktop

Innym sposobem jest pobranie aplikacji GitHub Desktop, która pozwala na dodawanie do repozytorium plików prosto z naszego urządzenia.

4. Implementacja

4.1. Pliki layout'u

Layout w naszej aplikacji podzielony został na 3 pliki.xml:

- activity_start.xml
- activity_main.xml
- fragment_about.xml

4.1.1. activity_start

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="
   http://schemas.android.com/apk/res/android"
3 xmlns:app="http://schemas.android.com/apk/res-auto"
4 xmlns:tools="http://schemas.android.com/tools"
5 android:layout_width="match_parent"
6 android:layout_height="match_parent"
7 android:background="@drawable/background3">
8
9
10 <Button
11 android:id="@+id/buttonstart"
12 android:layout_width="297dp"
13 android:layout_height="101dp"
14 android:layout_marginStart="57dp"
15 android:layout_marginTop="334dp"
16 android:layout_marginEnd="57dp"
17 android:layout_marginBottom="296dp"
18 android:backgroundTint="#0C390F"
19 android:fontFamily="serif-monospace"
20 android:text="Start"
21 android:textSize="48sp"
22 app:layout_constraintBottom_toBottomOf="parent"
23 app:layout_constraintEnd_toEndOf="parent"
24 app:layout_constraintStart_toStartOf="parent"
25 app:layout_constraintTop_toTopOf="parent" />
26
27 </androidx.constraintlayout.widget.ConstraintLayout>
```

Listing 1. Activity_start

5. Testowanie

6. Podręcznik użytkownika

Bibliografia

- [1] *Strona internetowa Android Studio*. URL: <https://developer.android.com/studio> (term. wiz. 11.12.2023).
- [2] *Strona internetowa Android Studio-archiwum*. URL: <https://developer.android.com/studio/archive> (term. wiz. 11.12.2023).

Spis rysunków

1.1. Google maps	3
2.1. Android Studio	4
3.1. Android Studio Giraffe	5
3.2. Android Studio Instalacja	6
3.3. Android Studio Instalacja	6
3.4. GitHub	7
3.5. GitHub repozytorium	8
3.6. GitHub Desktop	8

Spis tabel

Spis listingów

1.	Activity_start	9
----	--------------------------	---