



Mobilny Asystent i Organizer
Nawadniania Roślin Domowych

Przedmiot: Projekt systemu informatycznego

Prowadzący:
mgr Jacek Żywcok
Autor:
Jakub Jaworski 5IGRS
18.01.2023r. Katowice

Spis treści

Tematyka projektu	2
Cel projektu	2
Motyw i inspiracja	2
Wstępna wizualizacja projektu	2
Założenia projektowe	3
Ograniczenia projektowe	3
Chronologiczny plan pracy	4
Wymagania funkcjonalne	6
Wymagania niefunkcjonalne	11
Sprzętowe	11
Systemowe	11
Organizacyjne	11
Narzędzia i technologie	12
Opis najważniejszych plików i funkcji	12
Przebieg uruchamiania projektu	14
Przebieg testowania projektu	15
Wnioski z testowania	21
Podsumowanie	22
Podręcznik użytkownika	23
Przeznaczenie i możliwości	23
Podstawowe wymagania	23
Opis instalacji i uruchamiania	23
Menu	24
Znane błędy	25
Spis ilustracji	26
Spis tabel	27
Bibliografia	28

Tematyka projektu

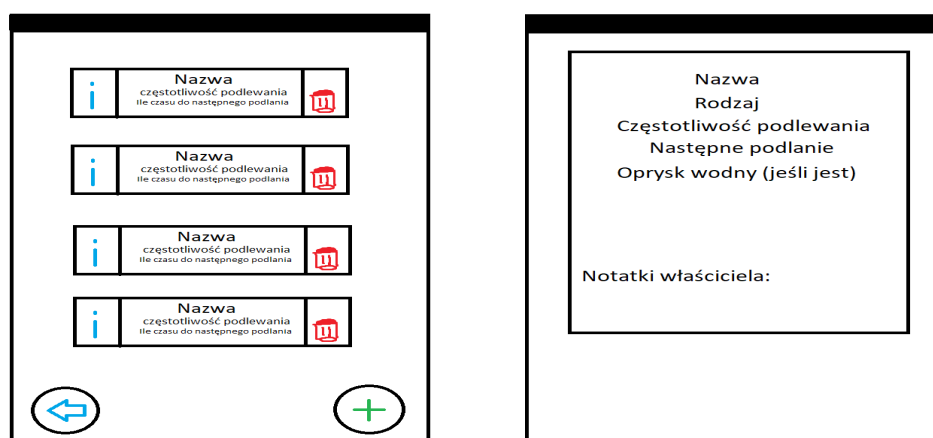
Cel projektu

Celem mojego projektu jest ułatwienie powtarzalnej i pozornie łatwej czynności jaką jest opieka nad domowymi roślinami. Aplikacja powinna przypominać i podawać optymalną ilość wody jaką potrzebuje poszczególne flora nie dopuszczając do przelania lub wysuszenia gleby co skutkuje obumarciem rośliny.

Motyw i inspiracja

Ludzie wyjeżdżając na urlop często zostawiają opiekę nad roślinami zaprzyjaźnionej sąsiadce lub rodzinie. Często ludzie mają problem z właściwym nawadnianiem wszelkiego rodzaju flory która zależnie od gatunku wymaga różnej ilości wody i wilgotności. Ten oto przyziemny problem skłonił mnie do stworzenia aplikacji, która zniweluje lub wyeliminuje te oto trudności.

Wstępna wizualizacja projektu



Ilustracja 1

Założenia projektowe

Projekt zakłada poniższe funkcjonalności:

1. Przechowywanie listy elementów
2. Dodawanie nowej rośliny
3. Edycja danych
4. Usuwanie danych
5. Wysyłanie powiadomień
6. Kompatybilność z każdym urządzeniem mobilnym

Aplikacja ma być przejrzysta i intuicyjna a ponadto powinna działać na urządzeniach z systemem Android 5.0+. Stworzona przy użyciu programu Android Studio w języku Java 18.

Ograniczenia projektowe

- Aplikacja nie będzie cross-platformowa
- Brak wsparcia w trybie landscape i praca tylko na systemach Android 5.0+
- Brak implementacji wersji na tablety
- Brak wsparcia dla ekranów z częstotliwością wyświetlania powyżej 60Hz
- Maksymalna ilość obiektów to 10
- Rozprowadzenie aplikacji tylko przy użyciu Android Debug Bridge

Dodatkowo ograniczeniem jest ilość urządzeń fizycznych na którym mam możliwość sprawdzenia poprawności działania projektu z tego względu projekt będzie sprawdzany na emulowanych urządzeniach z systemem android.

Chronologiczny plan pracy

Projekt będzie powstawał w oparciu o model projektowania kaskadowego ze względu na jednoosobowy zespół pracujący nad projektem i jasną wizję samego projektu.

Projekt będzie opierał się o następujące fazy:



Ilustracja 2

Fazy:

1. Planowanie i analizowanie

- 1.1. Wybór Środowiska i technologii
- 1.2. Poznanie wymagań systemowych i gruntowna analiza
- 1.3. Planowanie działań w procesie tworzenia aplikacji
- 1.4. Planowanie Układu klawiszy i Menu
- 1.5. Założenie efektu finalnego

2. Projektowanie

- 2.1. Projektowanie UI
- 2.2. Projektowanie Menu dodawania i Menu głównego
- 2.3. Projektowanie ikony i logo aplikacji

3. Implementowanie

- 3.1. Praca nad UI projektu, dodanie paneli, fragmentów i przycisków
- 3.2. Stworzenie panelu menu głównego i menu dodania elementu
- 3.3. Praca nad funkcjonalnością menu dodawania rośliny
- 3.4. Dodanie funkcji wyświetlania elementów i menu rośliny
- 3.5. Dodanie funkcji edytowania danych w menu rośliny
- 3.6. Dodanie przykładowych wartości
- 3.7. Stworzenie dokumentacji
- 3.8. Stworzenie instrukcji dla użytkownika

4. Testowanie

- 4.1. Wdrożenie aplikacji na urządzenie mobilne
- 4.2. Testy kompatybilności
- 4.3. Testy obciążeniowe
- 4.4. Testy funkcjonalności aplikacji

Wymagania funkcjonalne

Tabela 1 Funkcjonalność 1

Nr. wymagania	1
Typ wymagania	Funkcjonalne
Przypadek użycia	Dodanie nowej rośliny do pliku.
Opis	Użytkownik wprowadza roślinę do pliku z zapisem.
Uzasadnienie	Dodanie nowej rośliny do pliku jest kluczowym elementem funkcjonowania organizera.

Tabela 2 Funkcjonalność 2

Nr. wymagania	2
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja rośliny w pliku.
Opis	Użytkownik edytuje wprowadzone informacje.
Uzasadnienie	Użytkownik musi mieć możliwość edycji wprowadzonych danych.

Tabela 3 Funkcjonalność 3

Nr. wymagania	3
Typ wymagania	Funkcjonalne
Przypadek użycia	Usunięcie rośliny z pliku.
Opis	Użytkownik usuwa wpis z pliku.
Uzasadnienie	Użytkownik musi mieć możliwość usunięcia wpisu.

Tabela 4 Funkcjonalność 4

Nr. wymagania	4
Typ wymagania	Funkcjonalne
Przypadek użycia	Wybór rodzaju rośliny.
Opis	Użytkownik wybiera rodzaj rośliny.
Uzasadnienie	Użytkownik ma możliwość wyboru rodzaju dzięki czemu aplikacja automatycznie dobierze sugerowaną ilość wody dla rośliny.

Tabela 5 Funkcjonalność 5

Nr. wymagania	5
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja rodzaju rośliny.
Opis	Użytkownik edytuje rodzaj rośliny.
Uzasadnienie	Użytkownik musi mieć opcję edycji rodzaju.

Tabela 6 Funkcjonalność 6

Nr. wymagania	6
Typ wymagania	Funkcjonalne
Przypadek użycia	Dodanie cyklu nawadniania.
Opis	Użytkownik dodaje cykl.
Uzasadnienie	Użytkownik dodaje odstęp czasowy pomiędzy cyklami nawadniania poszczególnej rośliny mierzony w dniach co jest kluczową funkcją aplikacji.

Tabela 7 Funkcjonalność 7

Nr. wymagania	7
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja cyklu nawadniania.
Opis	Użytkownik edytuje cykl.
Uzasadnienie	Użytkownik edytuje odstęp czasowy pomiędzy cyklami nawadniania poszczególnej rośliny.

Tabela 8 Funkcjonalność 8

Nr. wymagania	8
Typ wymagania	Funkcjonalne
Przypadek użycia	Dodanie zraszania do powiadomienia.
Opis	Użytkownik dodaje zraszanie do cyklu.
Uzasadnienie	Użytkownik dodaje opcję zraszania do cyklu poszczególnej rośliny.

Tabela 9 Funkcjonalność 9

Nr. wymagania	9
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja opcji zraszania.
Opis	Użytkownik edytuje opcję zraszania.
Uzasadnienie	Użytkownik musi mieć możliwość edycji opcji zraszania

Tabela 10 Funkcjonalność 10

Nr. wymagania	10
Typ wymagania	Funkcjonalne
Przypadek użycia	Dodanie notatki.
Opis	Użytkownik dodaje notatkę.
Uzasadnienie	Użytkownik musi mieć możliwość spisania porad właściciela dla poszczególnej rośliny.

Tabela 11 Funkcjonalność 11

Nr. wymagania	11
Typ wymagania	Funkcjonalne
Przypadek użycia	Usunięcie notatki.
Opis	Użytkownik usuwa notatkę.
Uzasadnienie	Użytkownik musi mieć możliwość usunięcia porad właściciela dla poszczególnej rośliny

Tabela 12 Funkcjonalność 12

Nr. wymagania	12
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja notatki.
Opis	Użytkownik edytuje notatkę.
Uzasadnienie	Użytkownik musi mieć możliwość edycji porad właściciela dla poszczególnej rośliny.

Tabela 13 Funkcjonalność 13

Nr. wymagania	13
Typ wymagania	Funkcjonalne
Przypadek użycia	Dodanie rodzaju z przykładowym zdjęciem rośliny.
Opis	Użytkownik dodaje rodzaj.
Uzasadnienie	Użytkownik musi mieć możliwość dodania rodzaju.

Tabela 14 Funkcjonalność 14

Nr. wymagania	14
Typ wymagania	Funkcjonalne
Przypadek użycia	Edycja rodzaju rośliny.
Opis	Użytkownik edytuje rodzaj.
Uzasadnienie	Użytkownik musi mieć możliwość edycji rodzaju.

Tabela 15 Funkcjonalność 15

Nr. wymagania	15
Typ wymagania	Funkcjonalne
Przypadek użycia	Podlanie rośliny
Opis	Użytkownik naciska klawisz podlewania rośliny.
Uzasadnienie	Użytkownik musi mieć możliwość zakomunikowania podlewania rośliny w celu resetu cyklu.

Wymagania нефunkcjonalne

Sprzętowe

Aplikacja opisywana w tym dokumencie ma takie same wymagania sprzętowe jak system Android 5.1.1 przez co jest kompatybilna z 90% czyli większością urządzeń na rynku.

- Smartfon lub tablet z ekranem o rozdzielczości 320x480 pikseli lub wyższej
- 65 MB wolnej przestrzeni w głównej pamięci urządzenia
- Architektura procesora x86, x86-64, Arm5, Arm6, Arm7 lub Arm 8
- 452 MB pamięci ram

Systemowe

- System android w wersji co najmniej 5.1.1
- Odblokowany tryb deweloperski android
- Zainstalowany android debug bridge (adb) na komputerze rozpowszechniającym aplikację.

Organizacyjne

Nie stwierdzono

Narzędzia i technologie

W trakcie tworzenia aplikacji wykorzystałem następujące technologie:

- Android Studio 2021.3.1
- Visual Studio Code version 1.74
- Android Debug Bridge (adb)
- Software development kit for android (sdk)

Opis najważniejszych plików i funkcji

AndroidManifest.xml - Plik manifestu opisuje informacje o aplikacji dla narzędzi tworzenia aplikacji na Androida, który jest tworzony automatycznie przez program Android studio.

RoslinaDaneException - Plik zawierający funkcję używaną do wyświetlania powiadomień dotyczących błędów np. błędne dane (zbyt wiele liter).

FirstFragment - Plik zawierający interaktywność i funkcjonalność menu głównego. Zawiera funkcję wdrażającą nawigację w menu, od prostych kliknięć przycisków po bardziej złożone akcje, takie jak paski aplikacji. Pozostałe funkcje to responsywne wyświetlenie wszystkich elementów (roślin) na ekranie i pobranie aktualnej daty w celu obliczenia czasu pozostałego do następnego podlewania.

MainActivity - Plik odpowiedzialny za obsługę UI, nawigacji pomiędzy fragmentami (menu), paską narzędzi i nawigację góra-dół.

RoslinaDane - Plik definiuje własności elementów. Zawiera akcesory get i set.

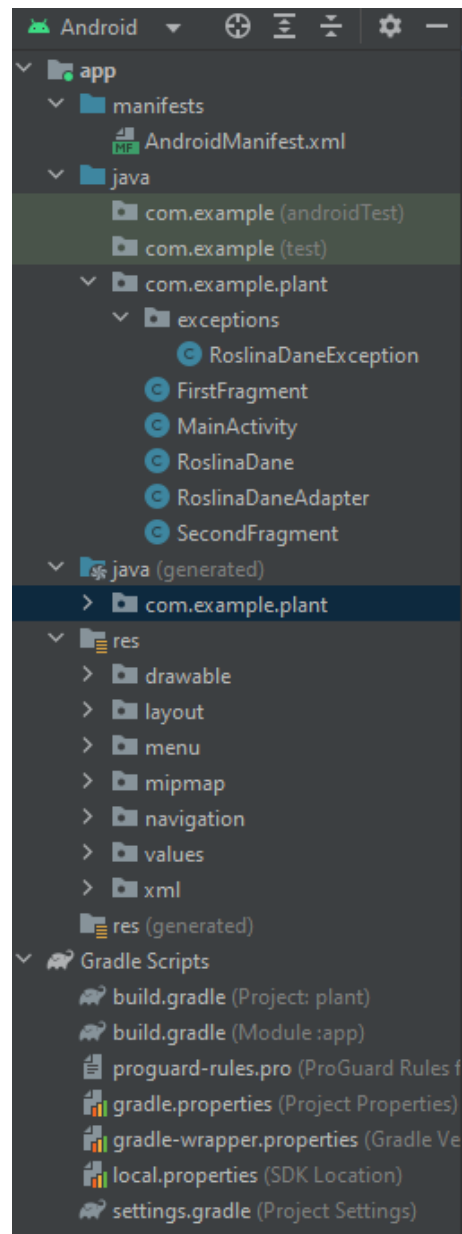
RoslinaDaneAdapter - Jest to plik do obsługi zdarzeń w graficznym interfejsie użytkownika (GUI). W tym przypadku przetwarza dane rośliny na listę wizualną widoczną w menu głównym.

SecondFragment - Plik menu dodawania (dodawanie, podlewanie, edycja, usuwanie)

BuildConfig - Plik z konfiguracją wersji aplikacji

Folder res/ zawiera następujące pliki formatu XML:

1. **Drawable** - Zawiera zdjęcia rodzaju roślin i tła ikony aplikacji
2. **Layout** - Zawiera konfigurację UI aplikacji
3. **Menu** - Zawiera konfigurację UI menu ustawień
4. **Mipmap** - Zawiera konfigurację ikony aplikacji
5. **Navigation** - Zawiera graf przejścia pomiędzy widokami
6. **Values** - Zawiera plik konfiguracyjny użytych kolorów i przycisków



Ilustracja 3.1

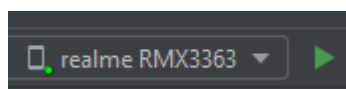
Przebieg uruchamiania projektu

Aby pobrać aplikację na urządzenie mobilne należy uruchomić Android Debug Bridge (adb) i uruchomić wiersz polecenia.

```
C:\Users\kubo-\Desktop\platform-tools>adb pair 192.168.0.73:39677
Enter pairing code: 610735
Successfully paired to 192.168.0.73:39677 [guid=adb-6c1588ff-xl8sh3]
```

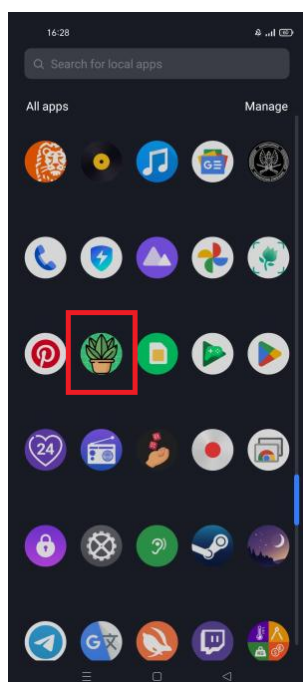
Ilustracja 3.2

Następnie wystarczy podłączyć urządzenia do tej samej sieci Wi-Fi i podać IP urządzenia mobilnego. Aby wysłać aplikację należy w programie Android studio wcisnąć przycisk “run”.



Ilustracja 4

Ostatnim krokiem jest kliknięcie ikony aplikacji na smartfonie aby uruchomić aplikację.



Ilustracja 5

Przebieg testowania projektu

Fazę testowania rozpocząłem od uruchomienia aplikacji na różnych modelach smartfonów:

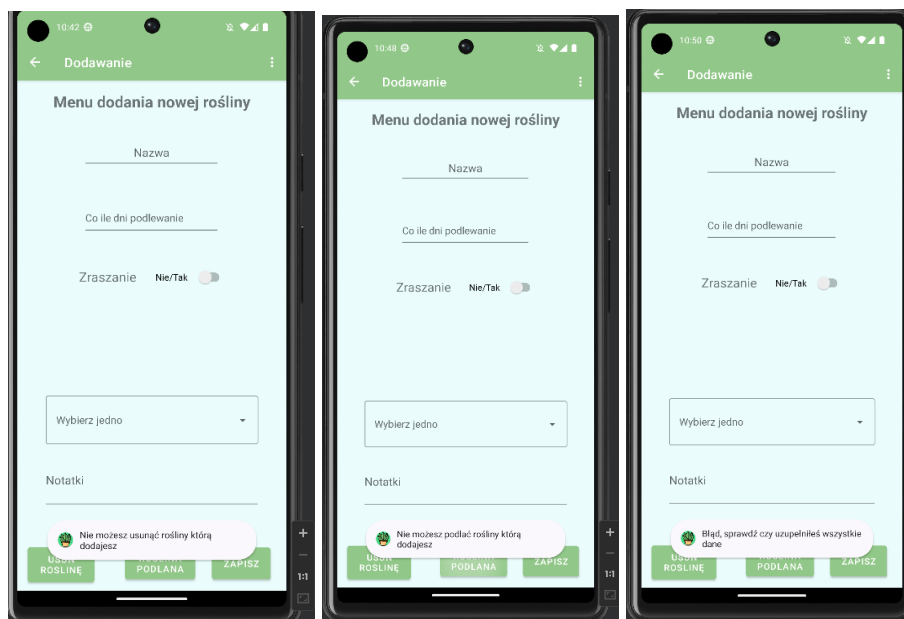
- Pixel 6
- Nexus 4
- Realme GT Master Edition
- Xiaomi Mi A1
- Galaxy Nexus

Następnie użyłem różnych wersji systemu android w celu przetestowania kompatybilności programu, test objął systemy:

- Android 12
- Android 11
- Android 10
- Android Pie
- Android Oreo
- Android Nougat
- Android Marshmallow
- Android Lollipop

Test kompatybilności przebiegł pozytywnie.

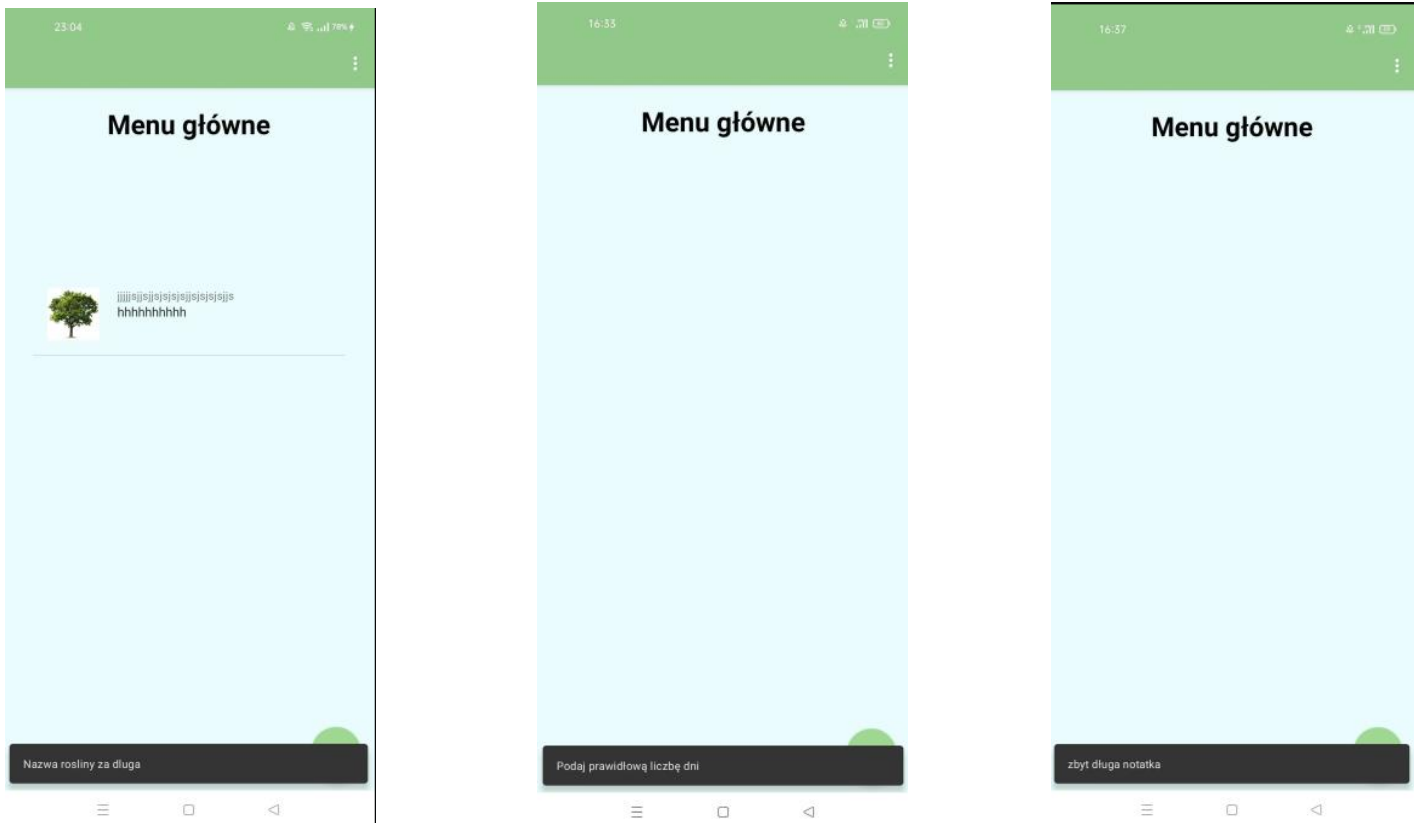
Kolejna faza testowania to testy weryfikacyjne i walidacyjne. Rozpocząłem od testu funkcji usunięcia, podlania i zapisu z brakiem danych o roślinie.



Ilustracja 6

Następnie przetestowałem pola wprowadzania danych:

- Nazwy (string, max 30 liter)
- Dni (integer, max 30 dni)
- Notatki (string, max 300 liter)



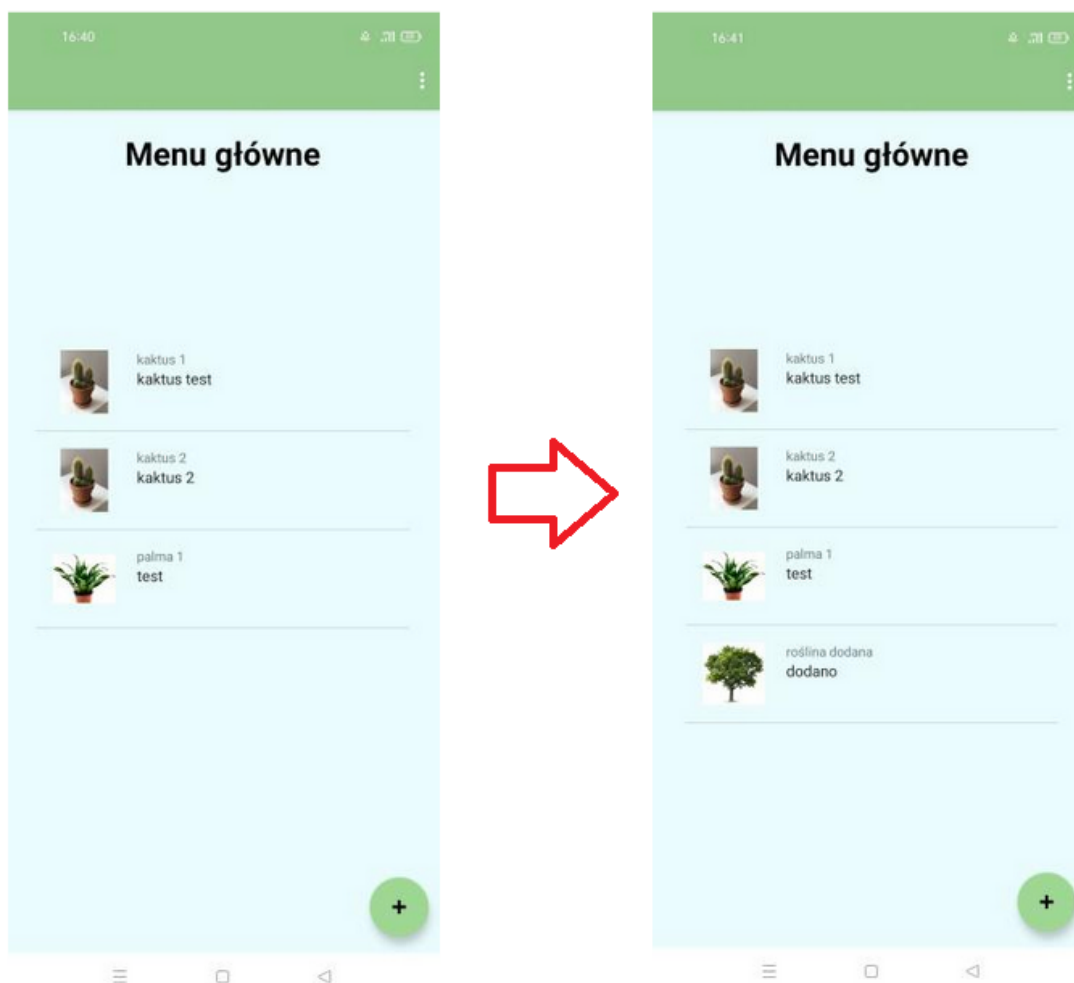
Ilustracja 7

Testy walidacyjne wykazały że wszystkie przypadki wprowadzania danych zakończyły się sukcesem a potencjalny użytkownik nie natrafi na błędy tego typu.

Następnie przeprowadziłem “soak test” i test obciążeniowy. Soak test został zakończony po czterech godzinach z wynikiem pozytywnym. Test obciążeniowy wykazał że aplikacja potrafi utrzymać ponad 1000 elementów bez utraty znacznej wydajności urządzenia.

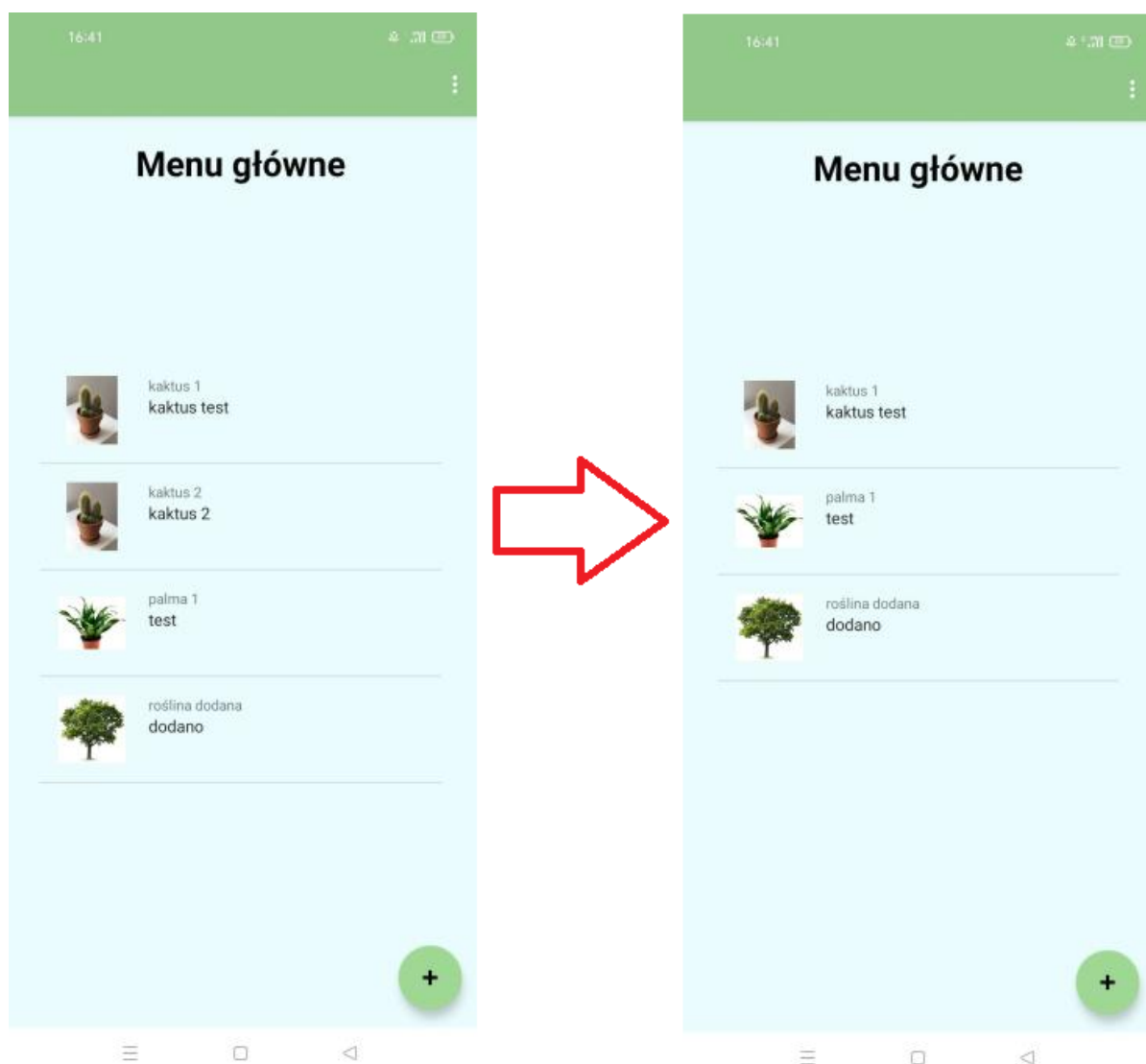
Kolejny test dotyczy usuwania, dodawania i edycji elementów:

1. Test funkcji dodawania



Ilustracja 8

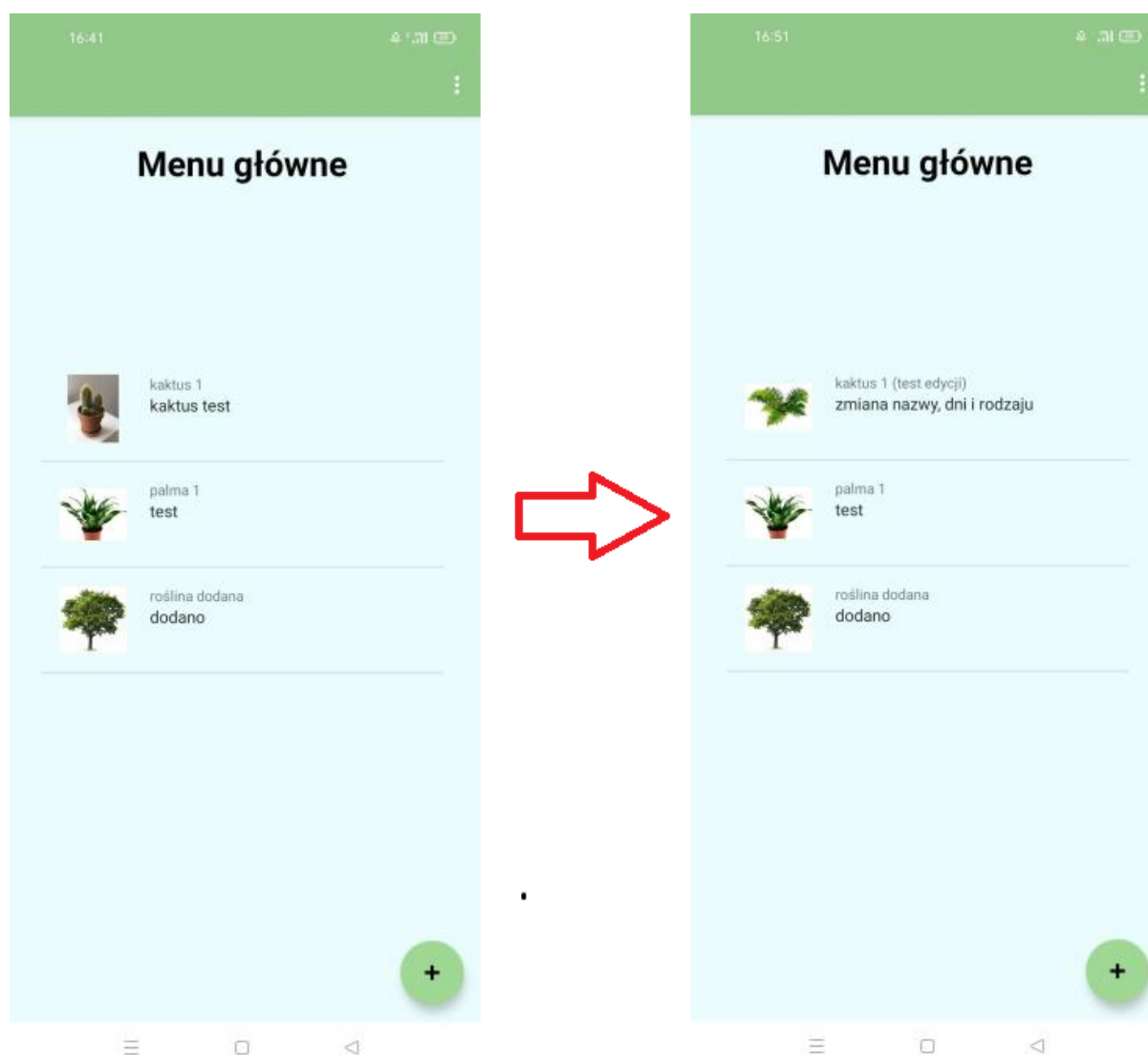
2. Test funkcji usuwania:



Ilustracja 9

Test funkcji usuwania roślin nie wykazał żadnych błędów a działanie odbywało się płynnie.

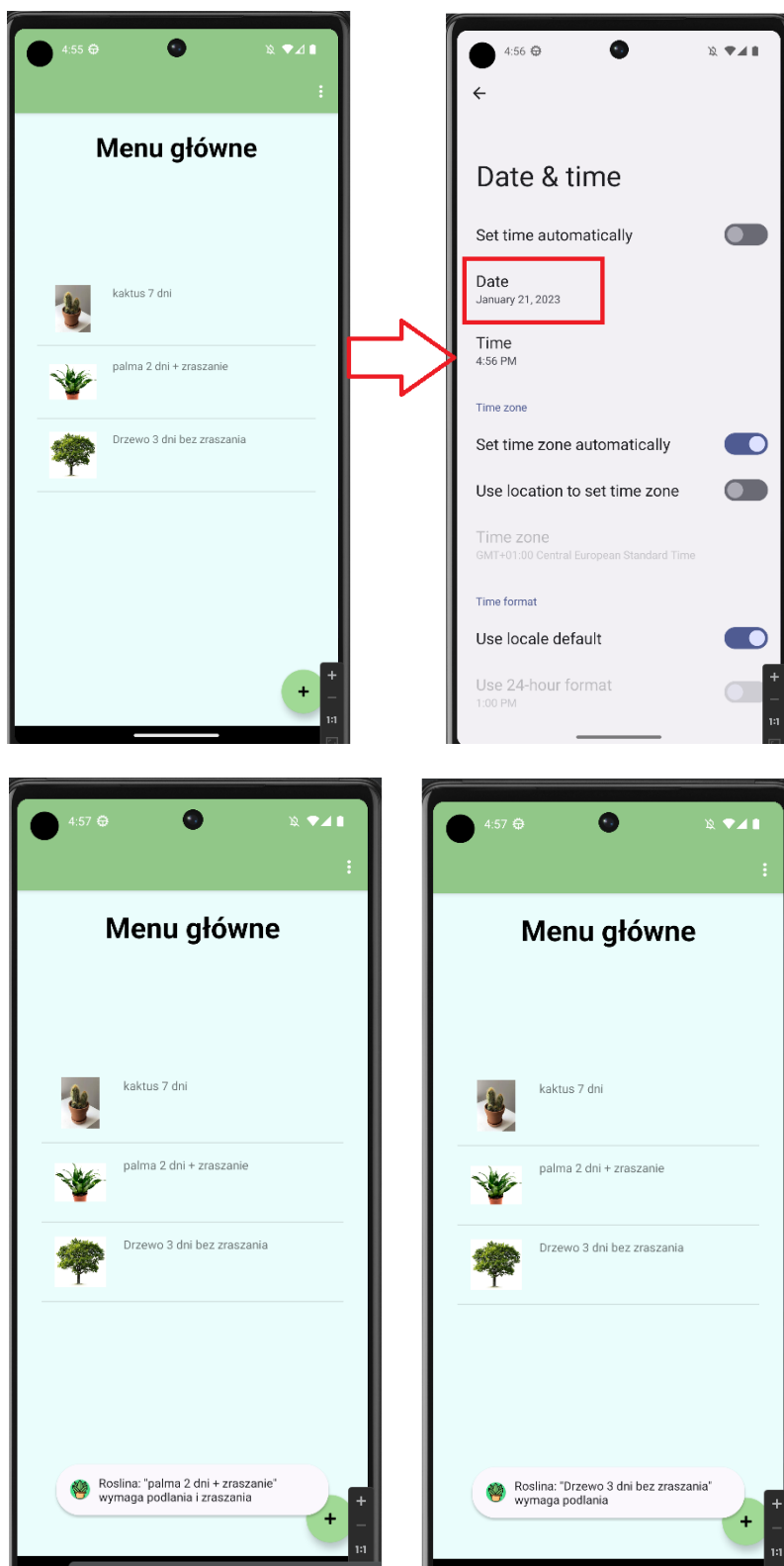
3. Test funkcji edytowania:



Ilustracja 10

Test funkcji edycji został przeprowadzony na każdym rodzaju rośliny i na urządzeniach symulowanych jak i fizycznych, nie natrafiono na żadne błędy.

Ostatnim testem był test powiadomień o podlewaniu i zraszaniu. Posłużyłem się emulatorami aby zmienić datę czego wynikiem powinny być 2 powiadomienia.



Ilustracja 11

Wnioski z testowania

Udało się zrealizować wszystkie zaplanowane testy w około 8 godzin. Aplikacja działa bez większych zarzutów. Jest czytelna, prosta w nawigacji oraz przyjazna dla użytkownika. Zastosowane są przejrzyste kolory i odpowiednie kontrasty.

zostały przeprowadzone testy:

1. Testy funkcjonalne
2. Testy użyteczności z użytkownikiem
3. Soak test
4. Test obciążeniowy
5. Test kompatybilności

Środowisko: Windows 10 64-bit, Android 5-12, Android studio 2022.1.1(Electric Eel)

Podsumowanie użytych narzędzi: zrzuty ekranu za pomocą "snipping tool" windows 10 emulatorze i Android 11 na urządzeniu fizycznym

Elementy aplikacji poddane testom:

1. Nawigacja pomiędzy panelami - działa.
2. Dodawanie elementów - działa.
3. Edycja i usuwanie - działa
4. Wysyłanie powiadomień - działa
5. Stabilność ogólna - działa

Wystąpiły niewielkie problemy ze środowiskiem i pakietem narzędzi sdk, zostały one rozwiązane w krótkim czasie bez znacznych strat czasowych. W przyszłości nie powinny występować błędy dotyczące danych lub funkcji. Zalecam przeprowadzanie okresowych testów kompatybilności na nowych urządzeniach i wersjach systemu Android.

Podsumowanie

Kolejnymi krokami w rozwoju aplikacji będzie rozwinięcie funkcji określenia rodzaju flory dzięki temu program może automatycznie podpowiedzieć użytkownikowi co ile dni powinien roślinę podlewać. Aby projekt osiągnął pełną funkcjonalność musi zostać rozpowszechniony za pomocą Android Market który jest wirtualnym dystrybutorem aplikacji na system android.

Projekt ma przed sobą potencjalnie duże możliwości rozbudowy a także trafia w lukę na rynku aplikacji wspomagających.



Ilustracja 12

Podręcznik użytkownika

Przeznaczenie i możliwości

Aplikacja jest przeznaczona na urządzenia mobilne z systemem Android i pozwala w prosty sposób wprowadzić dowolne rośliny oraz ich edycję. Program ułatwia opiekę nad roślinami i wysyła powiadomienia przypominające o potrzebach flory.

Uruchomienie sprowadza się do jednego kliknięcia a interfejs jest przejrzysty oraz prosty w obsłudze.

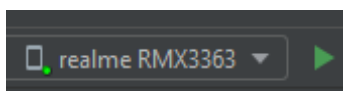
Podstawowe wymagania

Podstawowe wymagania to:

- Urządzenie dotykowe z systemem Android 5.1.1+
- Komputer z dowolnym systemem operacyjnym
- Android Debug Bridge (adb) w celu przesłania aplikacji
- Plik APK
- Android studio lub dowolny program tworzący plik APK

Opis instalacji i uruchamiania

Aby zainstalować aplikację należy wybrać plik APK i podłączyć komputer do urządzenia z systemem android poprzez kabel USB lub połączyć urządzenia przez sieć Wi-Fi. Następnie należy w programie android studio uruchomić instalację projektu.

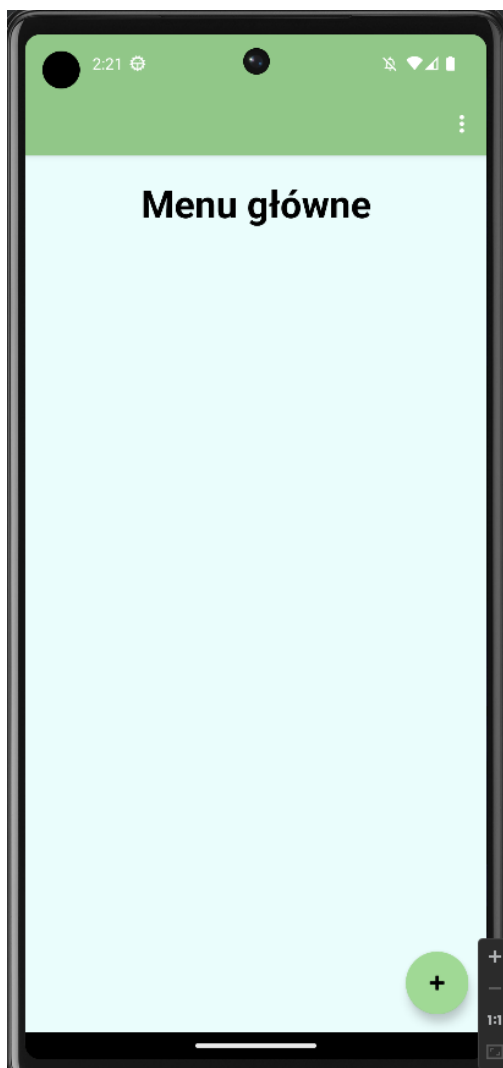


Ilustracja 13

Aby uruchomić aplikację należy kliknąć na ikonę programu.

Menu

Po pierwszym uruchomieniu aplikacji ukazuje się puste menu główne. Aby dodać roślinę należy wcisnąć przycisk dodania “+”.



Ilustracja 14

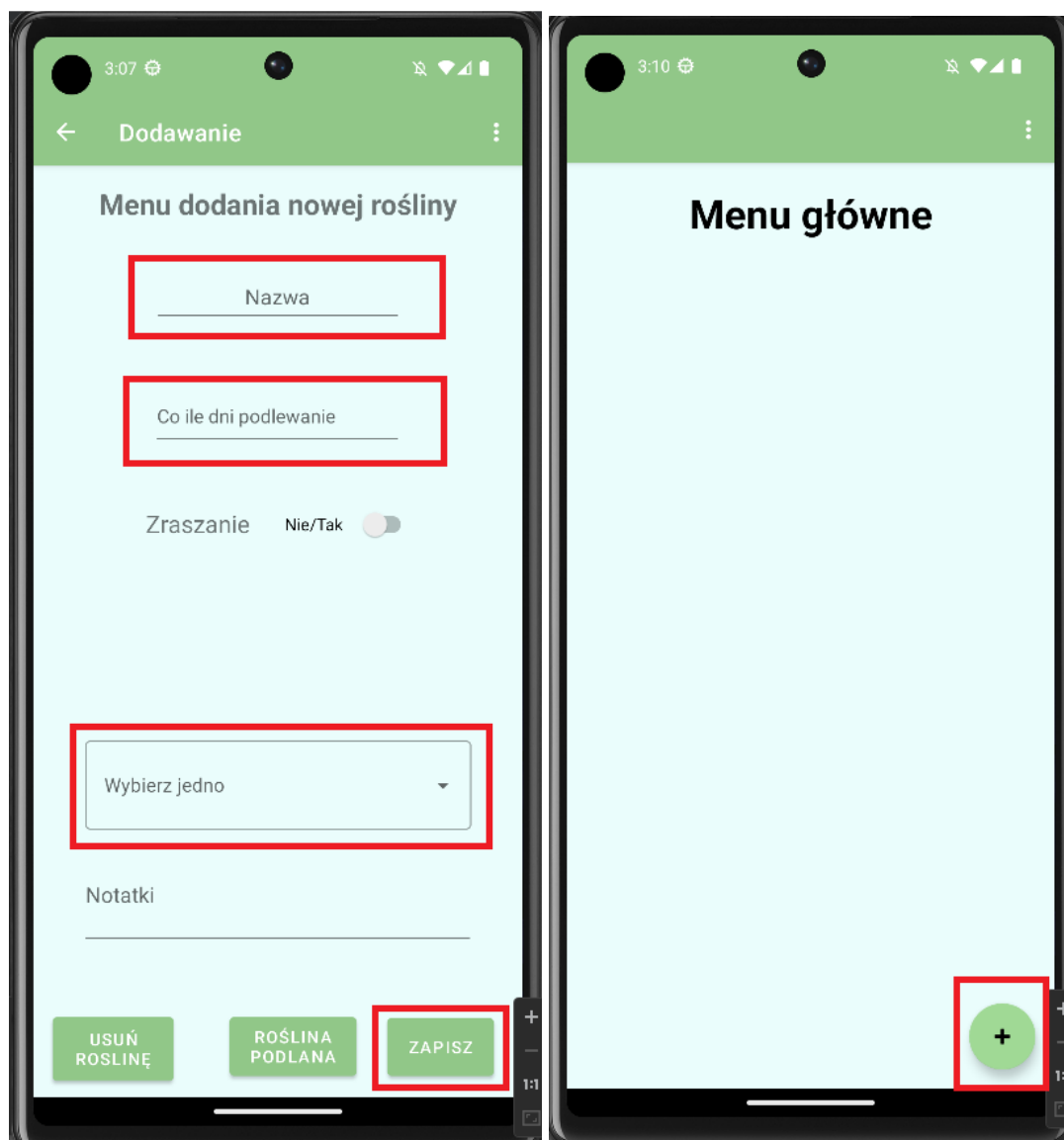
Menu dodania pozwala na dodanie nowej rośliny.

- W pierwszym polu należy wpisać nazwę rośliny.
- Drugie pole wymaga podania liczby dni pomiędzy podlewaniami.
- Trzecie pole to wybór opcji przypomnienia o zraszaniu rośliny.
- Czwarte pole wymaga zaznaczenia rodzaju rośliny.
- Ostatnie pole jest opcjonalne, możliwe jest wpisanie własnych notatek.
- Przycisk “usuń roślinę” usuwa wybraną wcześniej roślinę.
- Przycisk “Roślina podlana” informuje program o podlaniu owej rośliny.
- Przycisk “Zapisz” zapisuje dokonane zmiany lub nową roślinę.

Ilustracja 15

Dodawanie

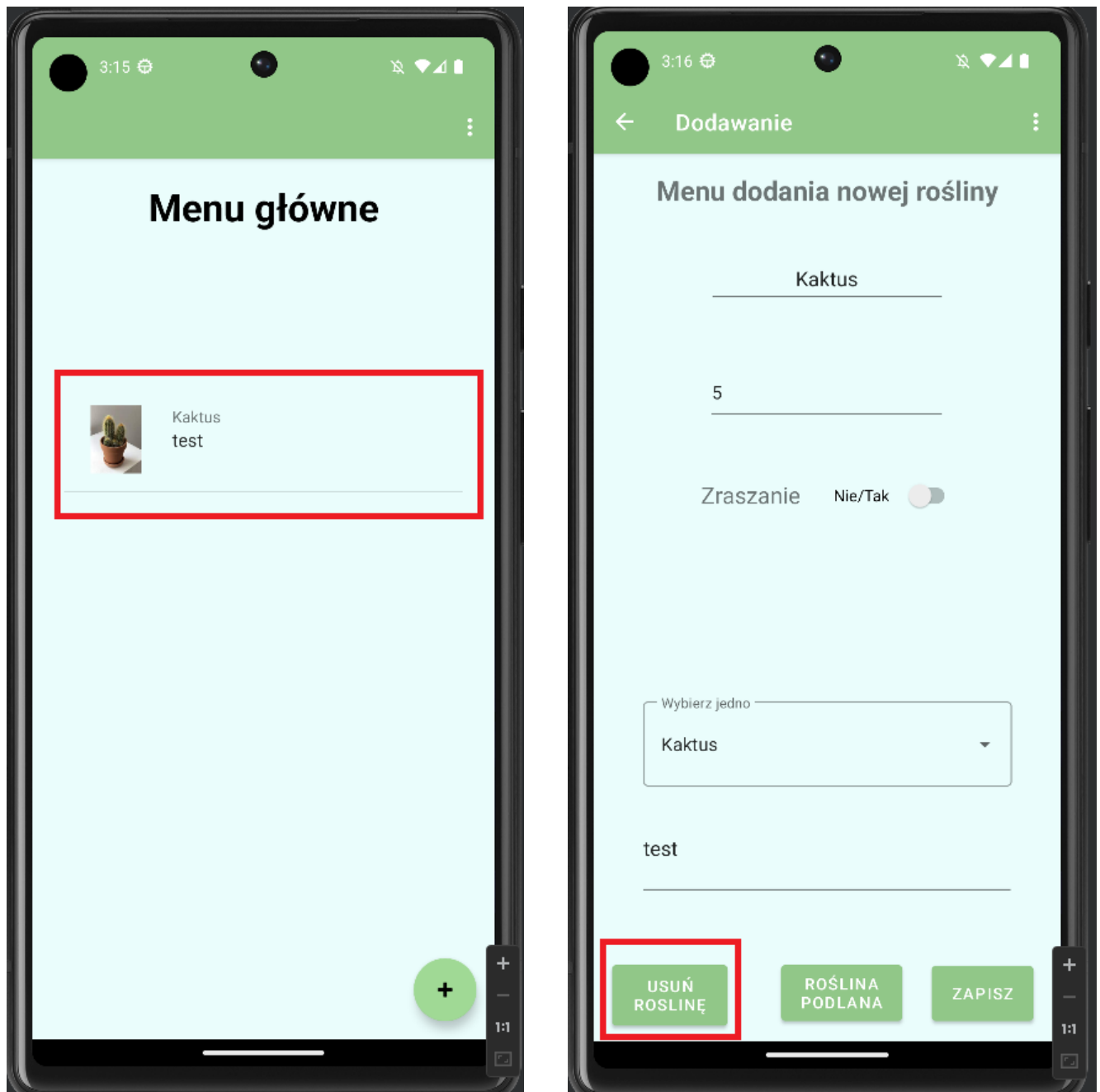
Aby dodać nową roślinę należy najpierw kliknąć przycisk dodania “+” w menu głównym. Następnie wypełnić trzy pola i zatwierdzić wciskając przycisk “Zapisz”.



Ilustracja 16

Usuwanie

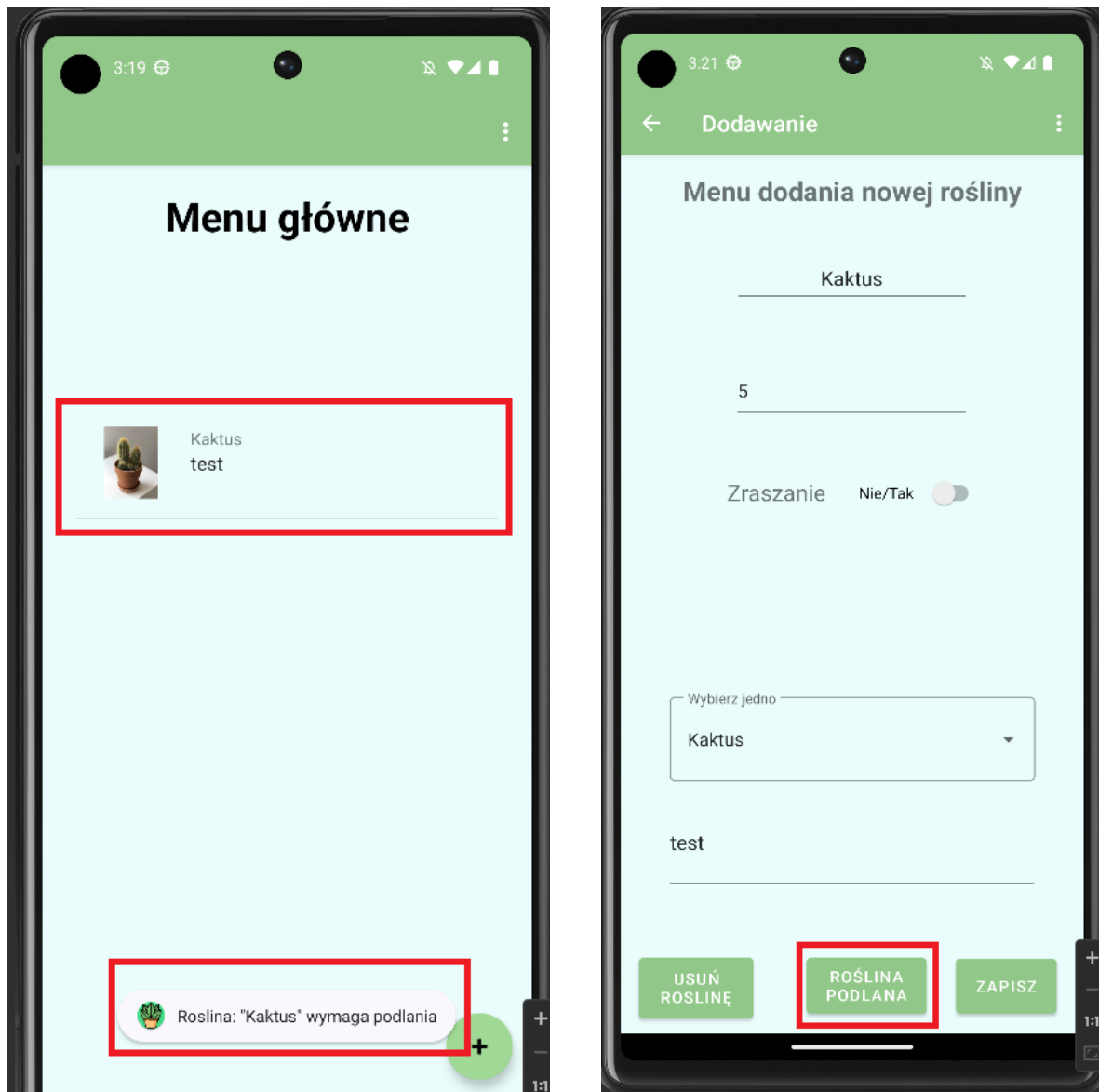
Aby usunąć element należy najpierw kliknąć na daną roślinę w menu głównym. Następnie zatwierdzić wciskając przycisk “Usuń”.



Ilustracja 17

Podlanie

Gdy pojawi się powiadomienie dotyczące wysuszonej rośliny to należy kliknąć na daną roślinę i wcisnąć klawisz "Roślina podlana".



Ilustracja 18

Znane błędy

Aplikacja działa bez większych zarzutów.

Spis ilustracji

[Ilustracja 1 Wizualizacja UI](#)

[Ilustracja 2 Plan pracy](#)

[Ilustracja 3 Połączenie urządzenia z komputerem przez Wi-Fi](#)

[Ilustracja 4 Na zrzucie ekranu widnieje model połączonego urządzenia](#)

[Ilustracja 5 Ikona w menu smartfona](#)

[Ilustracja 6 Test funkcji usunięcia, podlania i dodania z brakiem danych](#)

[Ilustracja 7 Test wprowadzania danych i różnych zmiennych](#)

[Ilustracja 8 Test funkcji dodawania](#)

[Ilustracja 9 Test funkcji usuwania](#)

[Ilustracja 10 Test funkcji edytowania](#)

[Ilustracja 11 Test powiadomień](#)

[Ilustracja 12 Ozdobny rysunek](#)

[Ilustracja 13 Uruchamianie instalacji w Android studio](#)

[Ilustracja 14 Menu Główne](#)

[Ilustracja 15 Menu dodania](#)

[Ilustracja 16 Instrukcja dodania rośliny](#)

[Ilustracja 17 Instrukcja usunięcia rośliny](#)

[Ilustracja 18 Instrukcja podlania rośliny](#)

Spis tabel

[Tabela 1 funkcjonalność 1](#)

[Tabela 2 funkcjonalność 2](#)

[Tabela 3 funkcjonalność 3](#)

[Tabela 4 funkcjonalność 4](#)

[Tabela 5 funkcjonalność 5](#)

[Tabela 6 funkcjonalność 6](#)

[Tabela 7 funkcjonalność 7](#)

[Tabela 8 funkcjonalność 8](#)

[Tabela 9 funkcjonalność 9](#)

[Tabela 10 funkcjonalność 10](#)

[Tabela 11 funkcjonalność 11](#)

[Tabela 12 funkcjonalność 12](#)

[Tabela 13 funkcjonalność 13](#)

Bibliografia

Dokumentacja <https://developer.android.com/docs>

Dokumentacja <https://www.baeldung.com/java-adapter-pattern>

Dokumentacja <https://developer.android.com/studio>

Dokumentacja <https://developer.android.com/studio/command-line/adb>

Darmowe obrazy <https://www.vecteezy.com/free-vector>