

28 styczeń 2022

Share Your Expenses - Projekt.

Mateusz Szymański

305899

Spis treści

1	Wstęp	2
2	Opis projektu	2
3	Wymagania dodatkowe	2
3.1	Umiędzynarodowienie (Internationalization)	3
3.2	Proces logowania	3
3.3	Aparat i pamięć urządzenia	3
3.4	Animacje	4
3.5	Wsparcie dla iOS i Android	4
3.6	Złożone formularze z walidacją	4
4	Firebase	5
5	Instrukcja	7
5.1	Przykładowi użytkownicy	7
5.2	Przykłady ekranów aplikacji	8

1 Wstęp

Niniejszy dokument jest końcową dokumentacją projektu wykonanego w celu zaliczenia przedmiotu "Programing of Mobile Applications in Flutter Technology" na kierunku Informatyka i Systemy Informacyjne na wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej w roku akademickim 2021/2022. Projekt jest aplikacją mobilną napisaną przy pomocy technologii i narzędzi Flutter.

2 Opis projektu

Celem aplikacji jest pomoc osobom takim jak rodziny, przyjaciele i współlokatorzy w lepszym kontrolowaniu wydatków w grupie osób. Użytkownik może tworzyć tzw. "grupy wydatków" i dodawać do nich inne osoby za pomocą nazwy użytkownika. Tworząc grupę musimy podać jej tytuł, opis, wybrać jej kategorię oraz ustalić walutę w jakiej będą podawane kwoty. W obrębie jednej grupy wydatków, każdy członek ma możliwość dodania własne poniesione koszty. Dodając nowy wydatek do grupy, użytkownik musi podać jego tytuł, opis, kwotę, datę oraz może zrobić zdjęcie zakupionej rzeczy lub paragonu. Może też załączyć zdjęcie z pamięci urządzenia. Użytkownicy w każdej chwili mogą zobaczyć wydatki całej grupy i sprawdzić kto jest ich właścicielem. Mają również możliwość podglądu zdjęcia paragonu lub przedmiotu należącego do wydatku.

Aby móc używać aplikacji użytkownik musi założyć konto. W tym celu powinien podać poprawny e-mail, bezpieczne hasło oraz nazwę użytkownika, która musi być unikalna i będzie widoczna dla innych użytkowników. Po założeniu konta, klient ma możliwość zalogowania się za pomocą adresu email oraz hasła podanego przy rejestracji. Aplikacja udostępnia również możliwość resetu hasła jeśli nastąpi taka potrzeba. Po zalogowaniu użytkownik może pełnoprawnie korzystać z aplikacji, tworzy i przegląda grupy oraz wydatki. Może również przejść do widoku profilu i potwierdzić swój adres email lub wylogować się z aplikacji.

3 Wymagania dodatkowe

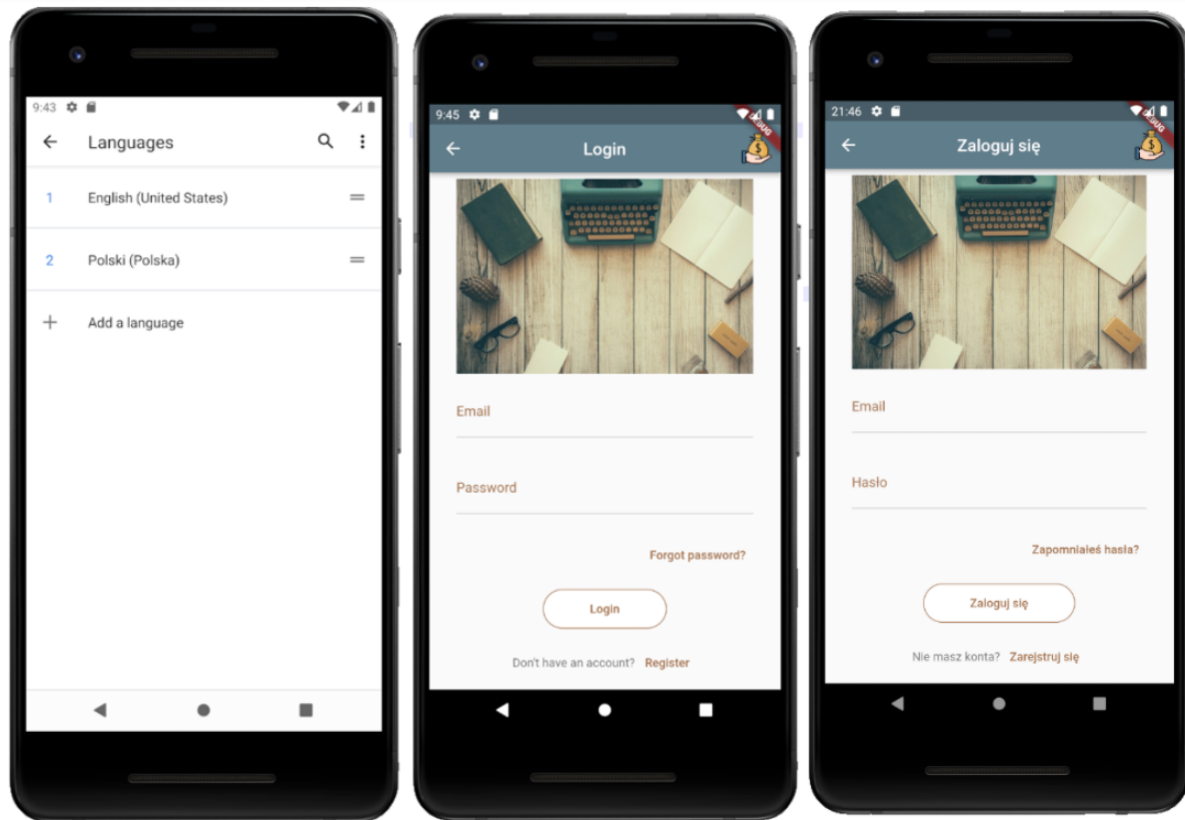
Poza implementacją podstawowych wymagań projektowych ocenianych na 50pkt, zdecydowałem się zaimplementować poniższe opcjonalne wymagania, są to:

- Umiejętność umiędzynarodowienie - 5pkt,
- Proces Logowania - 10pkt,
- Aparat i pamięć urządzenia - 10pkt,
- Animacje - 10pkt,
- Wsparcie dla iOS i Android - 5pkt,
- Złożone formularze z walidacją - 10pkt

Składają się one na kolejne 50 pkt. Każda wymieniona funkcjonalność została opisana poniżej.

3.1 Umieźdzynarodowienie (Internationalization)

Aplikacja udostępnia obsługę dwóch języków. Język angielski będącego domyślnym językiem oraz język polski. Język aplikacji jest zależny od języku używanego w systemie. Jeśli w systemie wybierzemy język polski aplikacja będzie po polsku w przeciwnym wypadku po angielsku. Do implementacji umieźdzynarodowienia zostały wykorzystane biblioteki takie jak: **json-annotation** oraz **intl**.



Rysunek 1: Ekran logowanie w języku polskim i angielskim.

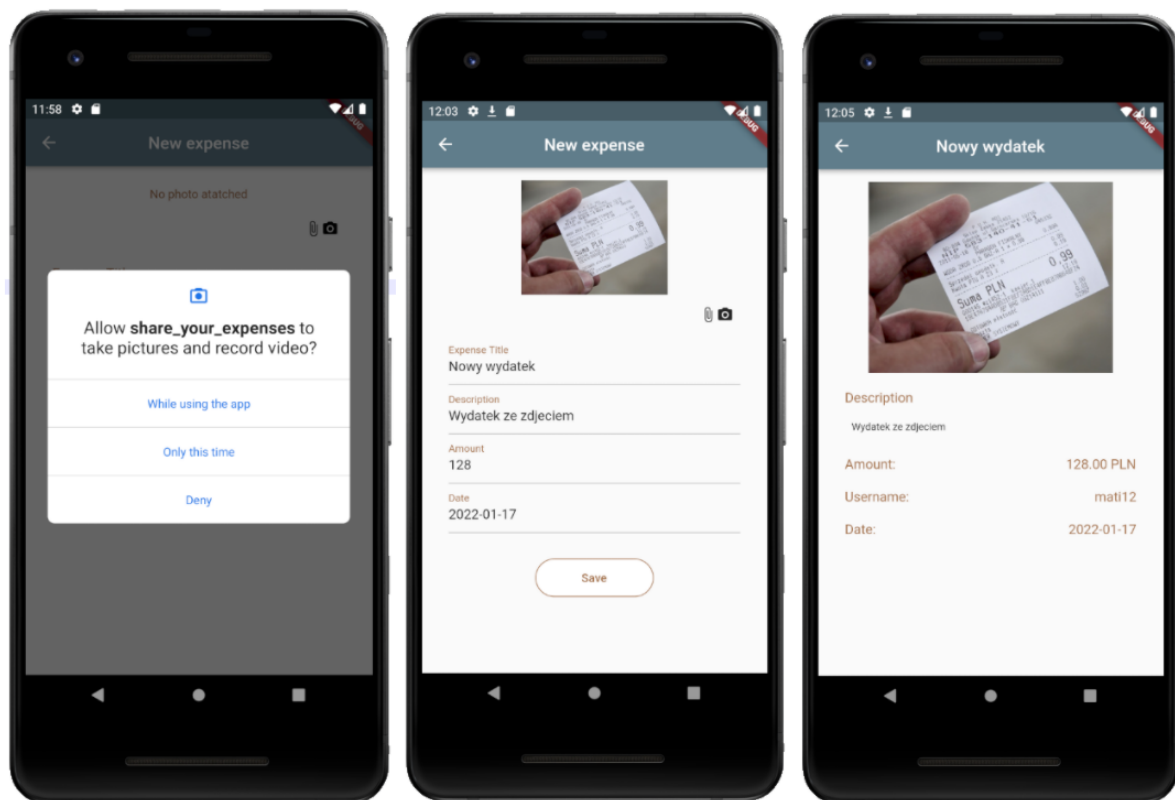
3.2 Proces logowania

Proces logowanie i rejestracji jest zaimplementowany z wykorzystaniem Firebase. Udostępniona jest opcja logowania za pomocą adresu email oraz hasła. Do implementacji zostały wykorzystane biblioteki **firebase-auth** oraz **firebase-core**. Na rysunku nr. 1 możemy zobaczyć ekran logowania. W projekcie komunikacja z serwisem uwierzytelniania zaimplementowana jest w **AuthService**.

3.3 Aparat i pamięć urządzenia

Podczas dodawania nowego wydatku użytkownik posiada możliwość załączenia zdjęcia związanego z wydatkiem. W tym celu została zapewniona możliwość zrobienia zdjęcia lub załadowania go z pamięci. Dodatkowo przed użyciem kamery lub pamięci urządzenia sprawdzane są przyznane uprawnienia i w razie potrzeby wyświetlany

jest dialog proszący o uprawnienia. Następnie zdjęcia są przesyłane do Storage w Firebase. Przy wyświetlaniu szczegółów wydatku jest ono ładowane. Do implementacji obsługi kamery i pamięci urządzenia wykorzystane zostały biblioteki: **image-picker** i **permission-handler**.



Rysunek 2: Ekran dodawania wydatku oraz ekran szczegółów danego wydatku.

3.4 Animacje

W aplikacji zaimplementowane zostały również pojedyncze animacje. Między innymi przy wyświetlaniu listy grup użytkownika lub na ekranie wydatków danej grupy. Aplikacja zawiera również indykator ładowania przy uruchamianiu aplikacji lub wykonywaniu operacji wymagającej czasu.

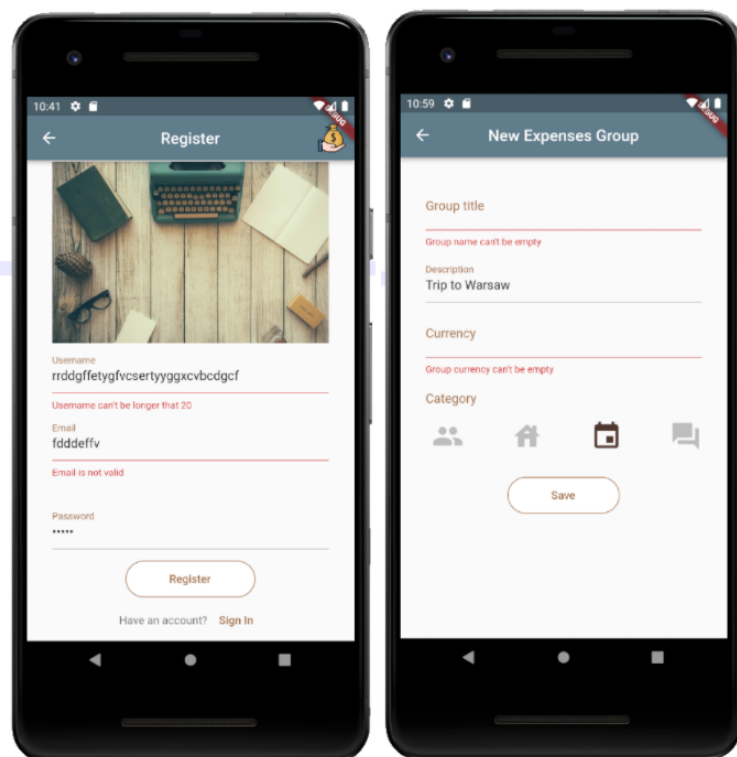
3.5 Wsparcie dla iOS i Android

Aplikacja może zostać uruchomiona zarówno na iOS jak i systemie Android. Wszystkie wykorzystane biblioteki wspierają oba te systemy. Również Firebase został skonfigurowany w aplikacji, tak by funkcjonować na obu systemach.

3.6 Złożone formularze z walidacją

W projekcie każde wprowadzanie danych zostało zaimplementowane przy użyciu formularza z walidacją. Są to kolejno formularze logowania, rejestracji, odnawiania

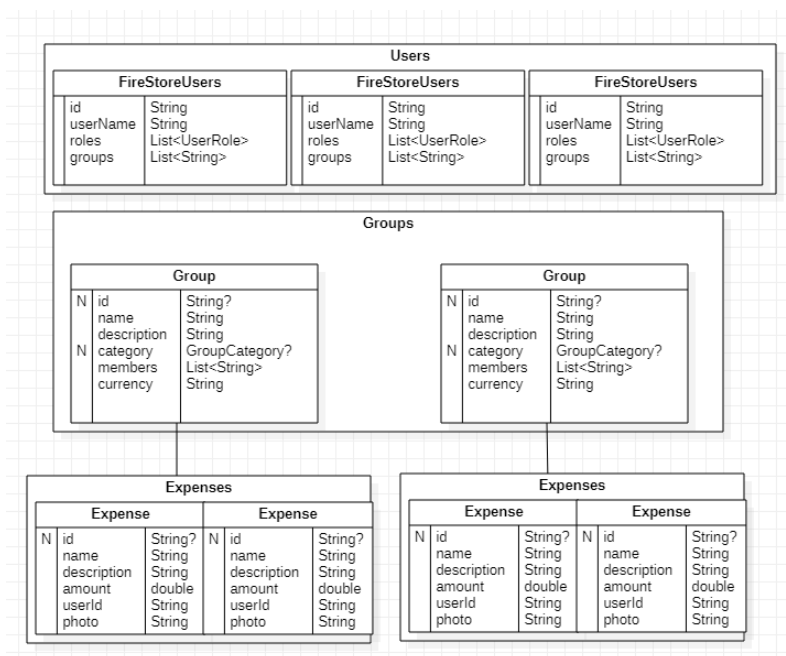
hasła, dodawanie użytkownika do grupy, tworzenie grupy oraz dodawanie nowego wydatku. Każde pole zostało wyposażone w odpowiednią walidację testując przykładowo poprawność podanego adresu email lub długość i poprawność wprowadzanej nazwy użytkownika. Wszystkie funkcje służące do walidacji zawarte są w pliku **validators.dart**.



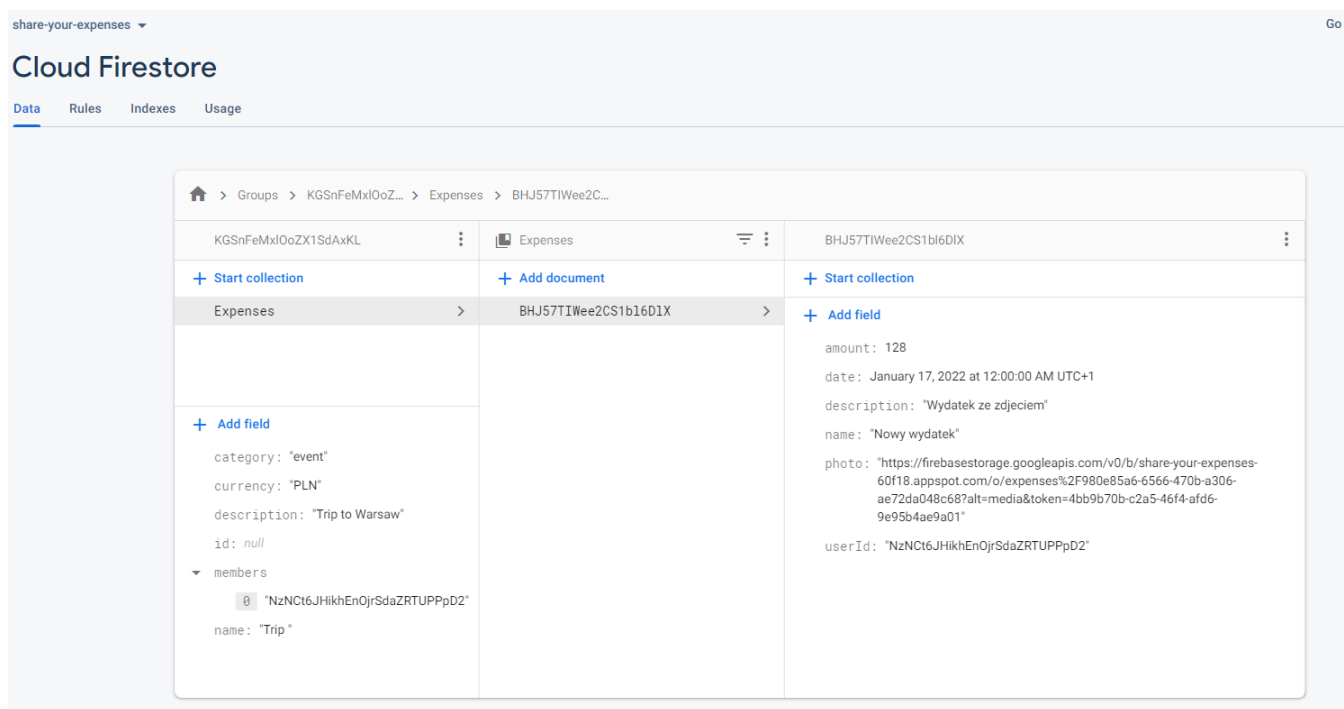
Rysunek 3: Przykłady ekranów gdzie wykorzystano formularze z walidacją.

4 Firebase

Poprawne działanie aplikacji wymaga połączenia z platformą firebase. Aplikacja wykorzystuje uwierzytelnianie (opisany w sekcji logowania) za pomocą adresu email oraz hasła. Dodatkowo korzysta z tzw. Storage w celu przechowywania zdjęć wydatków. Skonfigurowany jest również Firestore database. Występują w nim kolekcje takie jak użytkownicy gdzie dokumenty zawierają podstawowe informacje o użytkowniku i jego przynależności do grup. Możemy wyróżnić również kolekcje tzw. grup wydatków. Dokumenty takiej kolekcji zawierają podstawowe informacje o grupie a także o jej członkach. Dokument grupy zawiera również kolekcje wydatków. Każdy wydatek poza podstawowymi informacjami zawiera również opcjonalne pole będące ścieżką do zdjęcia.



Rysunek 4: Diagram używanych kolekcji (modeli).



Rysunek 5: Zdjęcie Cloud Firestore pokazujące wydatek grupy.

5 Instrukcja

Kod źródłowy aplikacji jest dostępny w folderze wraz z czytany dokumentem jak i również na Githubie(link). W pakiecie jest również zawarte skompilowane apk dla androida, które można łatwo uruchomić przechodząc do folderu Build. Dodatkowo apk można również pobrać z dysku(link).

Istnieje również możliwość skompilowania kodu samemu w tym celu należy posiadać poprawnie zainstalowany zestaw narzędzi Flutter. Następnie pobrać wymagane paczki i zbudować i uruchomić aplikację korzystając przykładowo z Visual Studio Code.

5.1 Przykładowi użytkownicy

Oczywiście z aplikacji można korzystać rejestrując się samemu, jednak w celach prezentacyjnych udostępnione są dwa tzw. konta pokazowe, które należą już do grup i mają w nich wydatki wraz z zdjęciami.

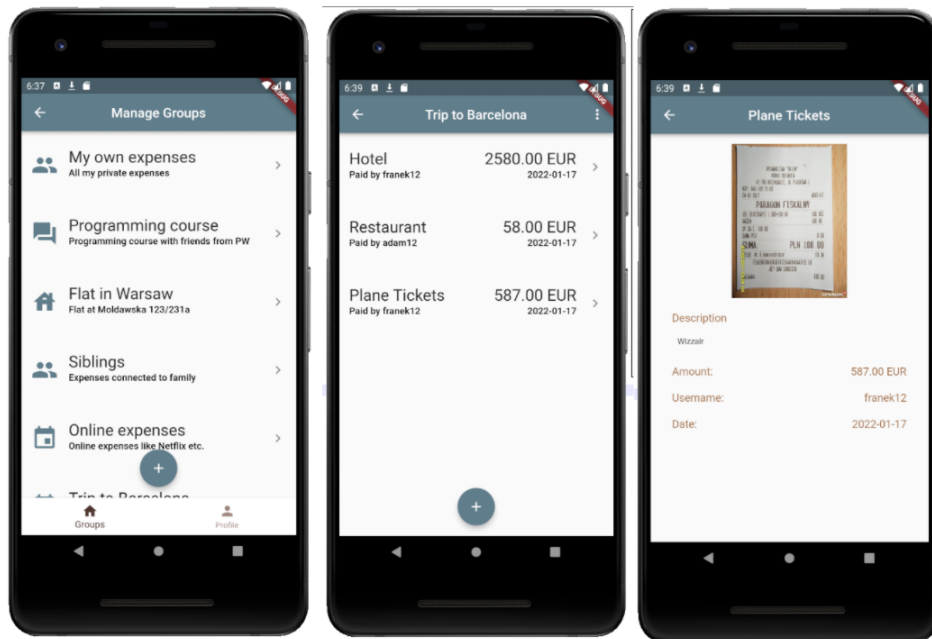
- | |
|---|
| <ul style="list-style-type: none">• Nazwa użytkownika: franek12• Adres email: testowy.franek@email.com• Hasło: hasloFranka12 |
|---|

Tablica 1: Użytkownik nr. 1 - Franek

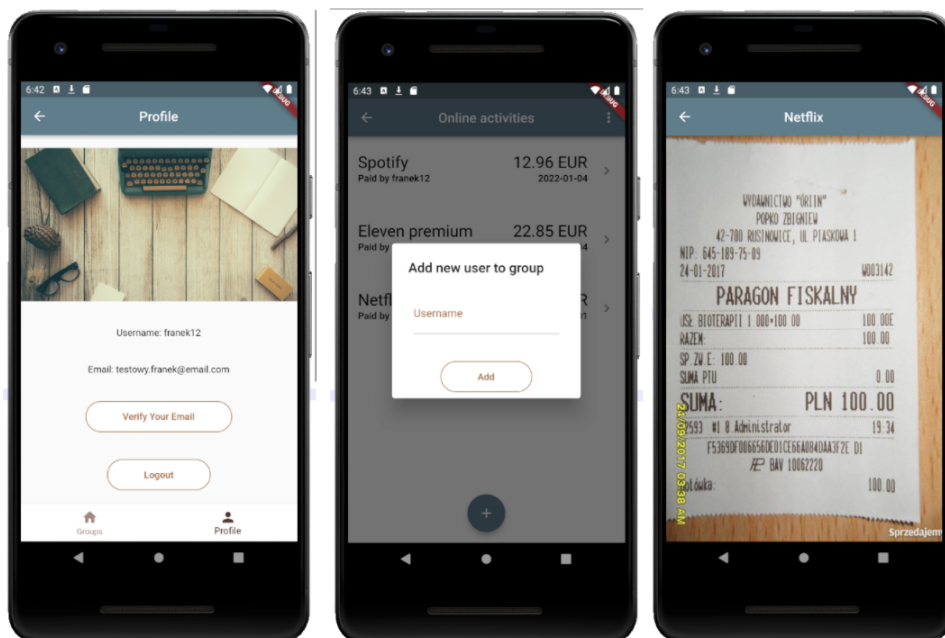
- | |
|--|
| <ul style="list-style-type: none">• Nazwa użytkownika: adam12• Adres email: testowy.adam@email.com• Hasło: hasloAdama12 |
|--|

Tablica 2: Użytkownik nr. 2 - Adam

5.2 Przykłady ekranów aplikacji



Rysunek 6: Przykładowe ekrany dla użytkownika franek12.



Rysunek 7: Przykładowe ekrany dla użytkownika franek12.