

Specyfikacja funkcjonalna

Jakub Wincewicz

Marzec 2016

Spis treści

1	Opis ogólny	2
1.1	Nazwa programu	2
1.2	Poruszany problem	2
1.3	Rozwiązanie problemu	2
1.4	Użytkownik docelowy	2
2	Opis funkcjonalności	3
2.1	Jak korzystać z programu?	3
2.2	Możliwości programu	4
3	Format danych i struktura plików	4
3.1	Przechowywanie danych w programie	4
3.2	Dane wejściowe	4
3.3	Dane wyjściowe	4
4	Scenariusz działania programu	5
5	Testowanie	5

1 Opis ogólny

1.1 Nazwa programu

Split it PRO

1.2 Poruszany problem

W dzisiejszych czasach podczas wyjazdów ze znajomymi stosunkowo często dochodzi do sytuacji gdzie ciężko i niekomfortowo jest zebrać pieniądze na dany wydatek. Za przykład może posłużyć np. prosta sytuacja związana z tankowaniem samochodu. Przyjmijmy, że grupa 5 osób jest w podróży samochodowej już od kilku godzin i część z podróżnych zasnęła. Aby zapłacić uczciwie za zatankowanie należałoby obudzić wszystkich i podzielić należność na każdego, przy czym musielibyśmy założyć, że każdy z podróżnych ma odpowiednią ilość drobnych pieniędzy w gotówce w danej walucie.

Dodatkowym problemem jest ograniczenie wynikające z tego, że za niektóre towary można płacić tylko kartą płatniczą więc bezpośrednie zwracanie należności byłoby utrudnione. Innym przykładem mogą być zakupy spożywcze na wieczornego grilla. Zakładając że wszyscy składają się na jednorazowy grill, jedzenie i coś do picia (jeden paragon) przynajmniej kierowca powinien pozostać trzeźwy, co rzecz jasna wyklucza go z rozliczania drugiego paragonu zawierającego alkohole.

1.3 Rozwiązanie problemu

Aplikacja ma na celu ułatwienie rozliczania wydatków poniesionych podczas podróży grupy znajomych. W aplikacji użytkownicy wpisują każdy wspólny (dla więcej niż jednej osoby). Przy każdym wydatku można zaznaczyć kto zapłacił za dany towar (usługę, etc.) oraz kto jest brany pod uwagę przy rozliczaniu. Wydatki takie jak np. paliwo do samochodu są zazwyczaj dzielone na wszystkich, a bilety do muzeum tylko na zwiedzających.

1.4 Użytkownik docelowy

Użytkownikami docelowymi są osoby w wieku 16-30 lat, zamierzające w najbliższym czasie wyjechać w podróż w grupie więcej niż dwóch osób na minimum trzy dni. Szczególnie dotyczy się to krótkich wyjazdów spowodowanych

wyjazdami okazjonalnymi np. długi weekend majowy.

2 Opis funkcjonalności

2.1 Jak korzystać z programu?

Użytkowanie programu będzie odbywało się w interfejsie graficznych w środowisku android 5.1 - testowany na Nexusa 4.

Po włączeniu aplikacji użytkownik może wybierać wcześniej utworzone lub dodawać nowe podróże - dla każdego kolejnego wyjazdu, albo nowej ekipy wyjazdowej zaleca się tworzenie nowego projektu.

Użytkownik może

- dodawać nowy rachunek
 - kwota
 - data/czas
 - nazwa towaru/usługi/kategorii
 - wybierać osobę, która zapłaciła
 - wybierać osoby, które należy uwzględnić przy rozliczeniu
- dodawać nowych podróżnych
- rozliczać koszty
 - pokazywać aktualny bilans
 - pokazywać proponowany sposób rozliczenia pomiędzy użytkownikami w taki sposób aby ilość wykonanych operacji była jak najmniejsza

Całość danych powinna być przechowywana w chmurze w taki sposób, aby każdy z użytkowników mógł w wygodny sposób dodawać rachunki. Jest to zabezpieczenie na wypadek rozładowania się telefonu przez głównego użytkownika, a także kontrolowanie przez innych czy nigdzie nie nastąpiła pomyłka przy wprowadzaniu danych.

2.2 Możliwości programu

Program powinien umożliwiać:

- wczytywanie do programu początkowej konfiguracji generacji z pliku o wybranym formacie,
- przeprowadzenie zadanej liczby generacji,
- wygenerowanie N obrazów przedstawiających stan wybranych generacji (np. w postaci plików PNG),
- zapisywanie bieżącej generacji do pliku (który może zostać potem wczytany).

3 Format danych i struktura plików

3.1 Przechowywanie danych w programie

Dane będą przechowywane w tablicach dwówymiarowych:

- `int tableA[HEIGHT][WIDTH]` - tablica inicjowana za pierwszym razem przez bitmapę dostarczoną przez użytkownika. W kolejnych stanach generalizacji jest ona wynikiem przejścia przez weryfikację warunków życie/śmierć dla każdej komórki
- `int tableB[HEIGHT][WIDTH]` - ta tablica w pierwszym stanie generalizacji jest zerowana. Dopiero w kolejnych stanach jest wypełniana "1" w miejscach, gdzie powinna pozostać lub urodzić się nowa komórka w zależności od `tableA`.

3.2 Dane wejściowe

Przechowywane w folderze razem z plikiem wykonywalnym, pliki graficzne w formacie *.bmp

3.3 Dane wyjściowe

Przechowywane w folderze z plikiem wykonywalnym, pliki tekstowe w formacie *.txt lub graficzne w formacie *.bmp albo *.png

4 Scenariusz działania programu

1. Plik graficzny będący początkowym stanem generalizacji zostaje przekonwertowany na tablicę i wypełnia `tableA[hieght][width]`
2. Na podstawie pierwszego stanu generalizacji zostają wygenerowane wszystkie następne
3. W zależności od wyboru użytkownika zostaje wygenerowany ostatni lub wszystkie stany generalizacji w pliku graficznym.

5 Testowanie

Program będzie testowany przy użyciu różnych źródłowych map bitowych oraz przy różnych wymiarach. W celu ułatwienia podglądania zmian kolejnych stanów generalizacji poszczególne stany zostaną połączone i zostanie stworzony w nich ruchomy element *.gif