STRONA TYTUŁOWA

Streszczenie

Każdy chociaż raz w życiu potrzebował udowodnić, że nabyty przez niego przedmiot został przez niego faktycznie zakupiony. Paragon w postaci kawałka papieru   
z najważniejszymi informacjami na temat zakupów jest najpopularniejszą formą wydawania dowodu kupna wystawianego przez osobę sprzedającą. ­Powstaje więc konieczność organizacji takich dokumentów. Praca traktuje o tym problemie oraz stara się wyjść naprzeciw trudnościom jakie napotykamy podczas przechowywania tego rodzaju dokumentów.

Celem pracy jest projekt oraz implementacja aplikacji, umożliwiającej efektywnie organizowanie dowodów zakupów, a także ich skanowanie w celu usprawnienia procesu ich ewidencji. ­­­Aplikacja ma umożliwiać także ich eksportowanie, informowanie   
o kończących się terminach gwarancji lub terminach możliwości zwrotu.

Praca zawiera wizję oraz specyfikację wymagań aplikacji mobilnej, która pozwala na organizowanie paragonów. W ramach pracy powstanie projekt i implementacja aplikacji mobilnej. Zostaną wykonane również testy aplikacji oraz zamieszczone zostanie podsumowane otrzymanych efektów wraz ze wskazaniem dalszych kierunków rozwoju.

Abstract

Everyone, at least once in their life, had to prove that the item they own was actually purchased by them. The most popular form of proof of the purchase is a receipt in the form of a piece of paper with the most important information on it. Therefore, it is necessary to organize such documents. This thesis deals with this problem and tries to overcome the difficulties encountered when storing receipts.

The aim of the thesis is to design and implement an application that allows one to organize their receipts effectively, as well as scan them to improve the process of their record. The application is also intended to allow exporting receipts and inform users about ending warranty periods or return dates of the items they purchased.

The thesis contains a vision and a specification of the requirements of the mobile application that improves receipts organizing. As a part of the thesis, the mobile application was designed and implemented. The project and a brief implementation description were included. Also there were application tests results presented and at the end, the results obtained were summarized followed by some further development directions.

Spis treści

[Streszczenie 1](#_Toc26704907)

[Abstract 1](#_Toc26704908)

[Wstęp 1](#_Toc26704909)

[Wprowadzenie do problematyki 1](#_Toc26704910)

[Geneza i motywacja pracy 2](#_Toc26704911)

[Cel i zakres pracy 3](#_Toc26704912)

[Struktura pracy 3](#_Toc26704913)

[1. Sposoby organizacji paragonów 4](#_Toc26704914)

[1.1. Aspekt prawny 4](#_Toc26704915)

[1.2. Sposoby wykorzystania paragonów 5](#_Toc26704916)

[2. Wizja sytemu 6](#_Toc26704917)

[2.1. Wprowadzenie 6](#_Toc26704918)

[2.2. Zidentyfikowanie grupy użytkowników 6](#_Toc26704919)

[2.3. Potrzeby i cechy 7](#_Toc26704920)

[2.4. Inne wymagania produktowe 7](#_Toc26704921)

[3. Rozwiązania konkurencyjne 8](#_Toc26704922)

[3.1. PanParagon 8](#_Toc26704923)

[3.2. Paragon: Paragony i karty lojalnościowe w jednym miejscu 10](#_Toc26704924)

[3.3. Receipt Lens-Expense Tracking & Reporting 12](#_Toc26704925)

[3.4. Receipt Bank 13](#_Toc26704926)

[3.5. Podsumowanie 14](#_Toc26704927)

[4. Specyfikacja i analiza wymagań 15](#_Toc26704928)

[4.1. Słownik pojęć 15](#_Toc26704929)

[4.2. Reguły biznesowe i model domenowy 15](#_Toc26704930)

[4.3. Historyjki użytkowników 17](#_Toc26704931)

[4.4. Diagram Przypadków użycia 19](#_Toc26704932)

[4.5. Skrócony opis przypadków użycia 20](#_Toc26704933)

[4.6. Scenariusze przypadków użycia wraz z prototypami interfejsów 21](#_Toc26704934)

[5. Projekt aplikacji ReceiptCarer 27](#_Toc26704935)

[5.1. Wykorzystane środowiska i narzędzia programistyczne 27](#_Toc26704936)

[5.2. Projekt Bazy Danych 28](#_Toc26704937)

[5.3. Architektura rozwiązania 30](#_Toc26704938)

[5.3.1. Architektura fizyczna 30](#_Toc26704939)

[5.3.2. Architektura logiczna 31](#_Toc26704940)

[5.4. Ekstrahowanie informacji z paragonów 32](#_Toc26704941)

[6. Implementacja 35](#_Toc26704942)

[6.1. Logowanie 35](#_Toc26704943)

[6.2. Rejestracja 36](#_Toc26704944)

[6.3. Lista paragonów 37](#_Toc26704945)

[6.4. Paragon – widok szczegółowy i edycja 38](#_Toc26704946)

[6.5. Eksportowanie paragonów 42](#_Toc26704947)

[6.6. Dodawanie nowych paragonów 42](#_Toc26704948)

[6.7. Lista grup paragonów 45](#_Toc26704949)

[6.8. Dodawanie nowej grupy 45](#_Toc26704950)

[6.9. Edycja grupy 47](#_Toc26704951)

[6.10. Wyszukiwanie paragonów 47](#_Toc26704952)

[7. Testy 49](#_Toc26704953)

[7.1. Testy jednostkowe 49](#_Toc26704954)

[Zakończenie 50](#_Toc26704955)

[1. Podsumowanie pracy 50](#_Toc26704956)

[2. Kierunki dalszego rozwoju 50](#_Toc26704957)

[Bibliografia 50](#_Toc26704958)

[Spis rysunków 51](#_Toc26704959)

[Spis tabel 52](#_Toc26704960)

[Załącznik 52](#_Toc26704961)

Wstęp

Wprowadzenie do problematyki

„Teczka z dokumentami i rachunkami już mi się nie domyka, segregator pęka   
w szwach, a szuflada nie chce się zamknąć. Po drodze zapewne zgubiłam kilka paragonów i dowiem się o tym, kiedy rozkleją mi się buty zimowe, które według sprzedawcy będą pewnie „nie do chodzenia po wodzie”. Albo kiedy będę chciała zwrócić do sklepu jakąś rzecz, którą kupiłam dwa tygodnie temu, bo to mi się zdarza, zwłaszcza jeśli chodzi   
o garderobę.” [1]

Tak opisuje swoją sytuację autor artykułu. Z podobnymi spostrzeżeniami autor pracy spotkał się również podczas przeprowadzania rozmów ze znajomymi i członkami rodziny.

Paragony stanowią codzienność naszego życia. W większości przypadków nie zwracamy na nie uwagi i po prostu wyrzucamy je do śmieci tuż po ich otrzymaniu. Sytuacja się zmienia, gdy nadchodzi moment kupna czegoś wartościowego, szczególnie gdy mowa   
o elektronice. Zważając na fakt, że tego typu urządzenia mają tendencje do psucia się   
w najmniej odpowiednich momentach, przechowywanie dowodów ich zakupu jest niezmiernie ważne, aby nie zamykać sobie możliwości późniejszej reklamacji lub zwrotu nabytego towaru w przypadku pojawiających się usterek.

W przypadku drogich produktów pojawia się kolejny aspekt związany   
z paragonami – gwarancje i zwroty. Te pierwsze trwają zazwyczaj kilka lat. Przypomnienie sobie, czy istnieje jeszcze możliwość oddania przedmiotu na gwarancję może być dla każdego człowieka nie lada problemem.

Małe papierowe druczki stanowią ważny element sprzedaży (czy kupna) towaru   
i są istotne nie tylko dla klientów, ale również dla sprzedawców – bez otrzymania dowodu zakupu, sprzedawca może nam co najwyżej współczuć z powodu posiadania niezadowalających w działaniu, czy jakości produktów. Omawiany aspekt ma duże znaczenie dla bardzo szerokiej grupy wiekowej – kupowanie towarzyszy nam przez całe życie, mając swój początek już w młodości, gdy nabywamy pierwsze drogie sprzęty. W tego rodzaju sytuacjach zazwyczaj dokumentacją dowodu kupna zajmują się dorośli, np. rodzice, ale nie oznacza to, że nie mogłyby się tego podjąć i te młodsze osoby, jednocześnie od najmłodszych lat ucząc się odpowiedzialności, w tym dbania o swoje finanse.

Po wpisaniu frazy „organizowanie paragonów” w popularną wyszukiwarkę internetową Google [1] znaleźć można wiele artykułów na temat tradycyjnych form przechowywania paragonów: zwykłe pudełko, słoiki, przegrody, teczki, a nawet portfele. Niestety warto zauważyć, że paragony wydawane w sklepach są przeważnie wyprodukowane ze słabej jakości materiału, na którym atrament ma tendencję do blaknięcia. Co więcej, są zazwyczaj niewielkich rozmiarów, co nie ułatwia ich utrzymania i zachowywania. Ciekawym jest, że w wynikach wyszukiwania nie występują żadne porady związane z cyfrową formą przechowywania dowodów zakupu, która jest równie dobrą, a właściwie lepszą formą organizacji w przypadku tego rodzaju dokumentów.

Tradycyjne formy organizacji dowodów zakupów nie pozwalają na skuteczną   
i wygodną formę ich organizacji, co z czasem może powodować problemy związane z ich niedoskonałą organizacją.

Geneza i motywacja pracy

Ludzka natura sprawia, że drobne, małe przedmioty są w naszym życiu traktowane jak błahostki. Takimi przedmiotami są np. paragony, które najczęściej lądują w koszu. Może się okazać, że przychodzi potrzeba ich użycia. Brak konsekwencji związanej   
z organizowaniem dowodów zakupów może przełożyć się na straty pieniężne. Autor pracy niejednokrotnie spotkał się z sytuacją, gdy nagle coś, co posiada, zepsuło się lub okazało się mieć usterkę, czy wadę od początku i koniecznym było udać się do miejsca zakupu produktu w celu dokonania jego zwrotu lub zareklamowania go. Niestety, najczęściej gdy robi się zakupy, to odruchowo wyrzuca się paragon do kosza, a potem okazuje się, że np. mleko jest przeterminowane i można, a nawet należałoby iść je zwrócić do sklepu, używając do tego otrzymanego wcześniej paragonu, ale niestety został on już wyrzucony do śmieci.

Warto wspomnieć, że dowód zakupów przydaje się na wielu innych płaszczyznach,   
np. również w celu odsprzedania produktu – dzięki możliwości udowodnienia ceny produktu   
w momencie zakupu oraz wciąż trwającego okresu gwarancji, osoby odsprzedające produkty mogą wynegocjować lepsze ceny sprzedaży.

Podczas rozmów z ludźmi zainteresowanie narzędziami do organizacji paragonów wykazywali także Ci, którzy dowody zakupów muszą zdawać w ramach rozliczania się,   
np. pracownik rozliczający się z pracodawcą z wydatków z delegacji. W takich sytuacjach przetrzymywanie dowodów zakupów w postaci papierowej może być uciążliwe, więc proste narzędzie na to pozwalające, a dodatkowo oferujące możliwość eksportowania byłoby rewelacyjnym usprawnieniem.

Paragony najczęściej są przekazywane nam w formie papierowej. Trzeba także wspomnieć, że istnieją inne formy dowodu zakupu. Jeżeli zakupu dokonano za pomocą karty kredytowej to istnieje możliwość ukazania dowodu zakupu pokazując wyciąg z konta bankowego. W przypadku większości banków jest to jednak trudne. Aplikacje banków pozwalają na wyszukiwanie zrealizowanych za pomocą karty płatniczej przelewów. Nie jest to jednak wygodne, ani nie pozwala na znalezienie bardziej szczegółowych informacji, co może się przełożyć na nieskuteczność w udowodnieniu zakupu.

Niektóre sklepy oferują także możliwość przetrzymywania faktur za zakupy bezpośrednio na swoim koncie po wcześniejszym zarejestrowaniu się na stronie sklepu. Także zakupy przez Internet pozostawiają najczęściej ślad zakupu w formie e-maila wysłanego przez sprzedawcę.

Z powyższego wynika, że istnieje zarówno wiele form wykorzystania paragonów jak   
i wiele form dostępu do nich lub ich odpowiedników. Przeważnie taka kolej rzeczy przekłada się na wysoką skuteczność w załatwianiu spraw związanych z dowodzeniem zakupu. Autor pracy jednak podkreśla, że zarówno bazując na doświadczeniu własnym jak i innych niekoniecznie jest to prawdą. Zbyt trudne lub zawiłe rozwiązania powodują często niechęć do działań. Przykładem może być oddawanie wadliwych lub uszkodzonych przedmiotów na gwarancję. Autor pracy spotkał się z kilkoma przypadkami w swoim życiu gdzie osoba mu znajoma postanowiła nie oddawać przedmiotu, który kwalifikował się pod oddanie na gwarancję, ponieważ po prostu uznawała to za zbyt pochłaniające czas i energię działanie. Takie osoby najczęściej nie przykładają uwagi do organizacji dowodów zakupów. Wiele ludzi nie jest świadomych też, że cyfrowa forma dowodu zakupu, w tym zdjęcie paragonu, jest tak samo dobrym i wystarczającym dowodem, jak papierowy odpowiednik.

Co ciekawe sprzedawcy często pozostają obojętni w kwestii prawnej i wykazują niechęć do przyjmowania innych form dowodów, niż papierowa. Wynika z tego, że świadomość konsumentów jak i sprzedawców ma spore znaczenie w kwestii motywacji do organizacji dowodów zakupów.

Nieważne czy motywacją do przechowywania paragonów w wygodniejszy sposób jest chęć reklamacji, sprzedaży, rozliczania się czy ewidencji dowodów zakupów dla pracodawcy, istnienie takiej potrzeby powoduje chęć jej zaspokojenia. Z tego oraz z wyżej przedstawionych powodów narodził się pomysł zaprojektowania aplikacji będącej skuteczną formą organizacji paragonów.

Cel i zakres pracy

Celem pracy jest projekt oraz implementacja aplikacji, umożliwiającej efektywnie organizowanie dowodów zakupów, a także ich skanowanie w celu usprawnienia procesu ich ewidencji. Aplikacja ma umożliwiać także ich eksportowanie, informowanie   
o kończących się terminach gwarancji lub terminach możliwości zwrotu i prowadzenie prostych statystyk związanych z paragonami.

W ramach pracy powstaną wizja aplikacji, specyfikacja i analiza wymagań, projekt oraz opis implementacji. W pracy znajdzie się także podsumowanie otrzymanych wyników wraz z prezentacją zrealizowanej aplikacji.

Struktura pracy

Praca będzie składać się z 7 rozdziałów, wstępu zakończenia oraz bibliografii.   
W rozdziale 1. przedstawione zostały sposoby organizacji paragonów, poruszono tematykę aspektu prawnego przy udowadnianiu zakupów oraz wyodrębniono profile użytkowników. Rozdział 2. zawiera wizję systemu wraz ze zidentyfikowanymi grupami użytkowników. Określono potrzeby, cechy i inne wymagania produktowe. W rozdziale 3. zostały przeanalizowane i porównane z proponowanym rozwiązaniem rozwiązanie konkurencyjne. Rozdział 4. Przedstawia specyfikację wymagań, zawarto w nim również przypadki użycia oraz przykładowe scenariusze wraz z prototypami interfejsu. Rozdział 5. Zawiera projekt wraz z zastosowanymi technologiami. Rozdział 6. przedstawia implementację i prezentację rozwiązania. W ostatnim rozdziale zostały przeprowadzone testy na zaimplementowanym rozwiązaniu. Następnie jest zakończenie. Pracę kończy spis cytowanej literatury.

# Sposoby organizacji paragonów

Proces dowodzenia zakupów zwykle odbywa się poprzez okazanie dokumentu poświadczającego fakt zakupu jakiegoś produktu lub usługi. Form przechowywania takich dowodów jest bardzo wiele. W przypadku tradycyjnych papierowych paragonów można skorzystać z prostego przechowywania ich w pudełku, w portfelu lub trzymać razem   
z opakowaniem zakupionego przedmiotu. Autor pracy spotkał się z osobami, które po prostu wykonują zdjęcia paragonów i w razie potrzeby wyszukują je na swoim smartfonie. Nie zmienia to jednak faktu, że wraz z czasem liczba paragonów rośnie, a co za tym idzie rośnie także trudność w ich wyszukiwaniu. Taka forma organizacji nie pozwala także na śledzenie wydatków, które są ściśle związane z paragonami.

Istnieje zawsze opcja segregacji paragonów w różnego typu, zrobione własnoręcznie lub zakupione, segregatory. Taka forma organizacji jest skuteczna i pozwala grupować,   
a zatem śledzić wydatki. Możliwe, że przy bardzo skrupulatnej segregacji można zapanować nad bardzo wieloma paragonami. Jest to jednak niezwykle czasochłonne,   
a w życiu istnieją inne bardziej istotne kwestie by poświęcać na paragony zbyt dużo czasu. Trzeba znaleźć więc skuteczniejsze rozwiązanie.

Warto wspomnieć, że w dzisiejszych czasach jest coraz więcej udogodnień ze strony sklepów. Ich polityka często pozwala na uproszczone formy realizacji zwrotów, gwarancji lub prowadzenia samych transakcji poprzez usługi online. Przykładem jest największa platforma transakcyjna online Allegro [2], gdzie został wprowadzony Program Ochrony Kupującego [3], pozwalający zarówno w przypadku kupującego, jak i sprzedającego, na szybkie odzyskanie pieniędzy w przypadku problemów z porozumieniem się pomiędzy stronami. W takim przypadku dokumenty poświadczające zakupy są kolekcjonowane   
i zarządzanie przez witrynę świadczącą usługę, co znacząco ułatwia zwroty, odsprzedaże lub ewentualne odnalezienie dowodu zakupu.

Nawet w przypadku korzystania z transakcji online, gdzie dowody zakupów są przechowywane na serwerach, przy dużej liczbie stron, z których czerpie się usługi, może pojawić się problem śledzenia, gdzie i kiedy wykonano zakup. Pojawia się potrzeba zcentralizowania wszystkich dowodów zakupów w jednym miejscu.

## Aspekt prawny

W tej pracy najczęstszą frazą używaną do określania dowodów zakupów są „paragony”, ze względu na to, że jest to najpopularniejsza nazwa na dowody zakupu.   
W świetle prawa jednak nie jest to jedyna dozwolona forma. Ściślej mówiąc, prawo nie używa tej frazy, tylko używa słów kluczowych „dowód zakupu”:

„Przepisy prawa w kwestii [..] reklamacji nie posługują się pojęciem paragonu, lecz dowodu zakupu, a ten może być rozmaity. Z tego powodu UOKiK [4] wskazuje, że paragon fiskalny jest tylko jednym z dowodów nabycia rzeczy w danym sklepie, a co za tym idzie tylko jedną z podstaw złożenia reklamacji.” [5]

Urząd Ochrony Konkurencji i Konsumentów jasne określa, że reklamacja bez paragonu jest możliwa i to tylko jedna z wielu możliwych dowodów zakupu, jaki konsument może przedstawić sprzedawcy zgłaszając reklamację. Dowodami mogą być także potwierdzenie płatności, wyciąg z rachunku bankowego, e-mail czy zeznanie świadków. [5]. W wypowiedzi na temat reklamacji bez paragonu Krzysztof Lehmann, delegatura UOKiK w Bydgoszczy podkreśla, że jeśli kupiony towar ma wadę, konsument ma prawo do złożenia reklamacji, żądając doprowadzenia towaru do stanu zgodności z umową.

Z powyższego wynika, że dowody zakupów mogą objawiać się w różnej formie. Istnieje więc możliwość przechowywania ich w formie cyfrowej.

## Sposoby wykorzystania paragonów

Dowody zakupów znajdują zastosowanie w wielu różnych sytuacjach. Przeważnie wykorzystywane są one do udowodnienia zakupu w celu oddania go na gwarancję (lub na zwrot, gdy polityka firmy na to pozwala). Paragony w codziennym życiu służą także grupom lub jednostkom do rozliczania się. Na paragonie widnieją przeważnie informacje na temat produktów i ich cen, co może posłużyć do łatwego ich wyodrębnienia, a następnie podziału w grupie. Pracownikom firm, wyjeżdżającym na delegacje, dowody zakupów służą do ostatecznego rozliczania się z delegacji.

Przy odsprzedaży posiadając dowód zakupu zyskujemy na zaufaniu, że faktycznie zakupiliśmy ten przedmiot. Dostajemy także możliwość wynegocjowania większej ceny, gdy dany przedmiot posiada ważną gwarancję (do której potrzebny jest dowód zakupu).   
W takim przypadku dołączenie dowodu zakupu do sprzedawanego przedmiotu jest kluczowym aspektem przy jego wycenie.

Paragony również mają bardzo ważna rolę przy zarządzaniu wydatkami. Ze względu na to, że są najpopularniejszą formą dowodów zakupów, przechowywanie i kategoryzowanie ich pozwala także na prowadzenie analizy wydatków.

# Wizja sytemu

## Wprowadzenie

System służący do organizacji paragonów musi zapewniać kompleksową i, co ważne, wygodną w użytkowaniu obsługę podstawowych czynności związanych   
z przechowywaniem, przeszukiwaniem, eksportowaniem zdjęć dowodów zakupów. Proponowane rozwiązanie powinno dostarczyć możliwość ich dodawania   
i kategoryzowania. Proces ten powinien minimalizować liczbę akcji jakie są potrzebne do dodania paragonu poprzez skanowanie ich i ekstrahowanie podstawowych informacji z nimi związanymi. Aplikacja w pierwszej wersji nie przewiduję możliwości skanowania innych form dowodów zakupów niż paragon wydawany z kasy fiskalnej przez osobę sprzedającą. Pomocnym narzędziem przy organizacji paragonów jest tworzenie spersonalizowanych kategorii. Powinna zatem zaistnieć możliwość tworzenia i zarządzania własnymi kategoriami. Z czasem, gdy liczba paragonów w poszczególnych kategoriach będzie zbyt wielka, by móc łatwo znajdować dane paragony, powinien istnieć sposób ich przeszukiwania. System powinien także zapewniać możliwość informowania   
o wprowadzonych przez użytkownika podczas dodawania paragonu długości trwania gwarancji. Pozwoliłoby to na informowanie użytkownika o nadchodzących kończących się terminach kończących gwarancje. Gdy nadejdzie moment wykorzystania danego paragonu, powinna istnieć możliwość jego eksportowania lub eksportowania całych kategorii. Powinna także istnieć możliwość rejestrowania i logowania się do aplikacji co pozwoli na ich łatwe przenoszenie między urządzeniami mobilnymi.

## Zidentyfikowanie grupy użytkowników

W przyszłym systemie identyfikujemy wiele grup użytkowników ze względu na wykorzystanie przechowywanych dowodów zakupów. Mogą to być osoby chcące przechowywać dowody zakupów by móc je później wykorzystać w celu złożenia gwarancji, mogą to być także osoby które ewidencją dowodów zakupów zajmują się by móc później je przekazać przy odsprzedaży. Mogą to być to także pracownicy firm, którzy zajmują się przekazywaniem dowodów zakupów z delegacji, na której byli do działu księgowości. Ostatecznie wszystkie wyżej wymienione grupy użytkowników posiadają wspólne cechy   
i potrzeby więc w ramach pracy zostanie zidentyfikowana tylko jedna grupa użytkowników: użytkownicy indywidualni. Opisy oraz odpowiedzialności użytkownika zostały zawarte   
w tabeli 1.

*Tabela 1 Opis użytkownika systemu*

|  |  |  |
| --- | --- | --- |
| Nazwa | Opis | Odpowiedzialności |
| Użytkownik indywidualny | Osoba zajmująca się organizacją paragonów na potrzeby własne. | Zbieranie i ewidencjonowanie paragonów.  Pilnowanie nadchodzących terminów kończących gwarancje.  Przekazywanie dowodów zakupów w razie potrzeby. |

## Potrzeby i cechy

Tabela 2. Przedstawia potrzeby użytkowników oraz jakie cechy przyszłego systemu wynikają z nich.

*Tabela 2 Potrzeby i cechy*

|  |  |  |
| --- | --- | --- |
| Potrzeba | Priorytet | Cechy |
| Użytkownicy indywidualni potrzebują wsparcia dla zarządzania paragonami | Wysoki | Definicja (tworzenie, przeglądanie, usuwanie  i edytowanie) paragonów wraz z informacjami  o nich i dodawanie ich do poszczególnych kategorii.  Definicja kategorii.  Możliwość edycji już istniejących kategorii.  Możliwość przeszukiwania paragonów po datach dodania paragonów, liczbie dni pozostałych do zakończenia gwarancji oraz informacjach związanych z dowodami zakupów.  Skanowanie zdjęć paragonów i ekstrahowanie  z nich podstawowych informacji.  Możliwość eksportowania paragonów. |
| Użytkownicy indywidualni potrzebują wsparcia do dodawania wielu paragonów w jak najszybszym czasie | Wysoki | Możliwość dodawania wielu paragonów w ramach jednej operacji dodawania. |
| Użytkownicy indywidualni potrzebują dostawać powiadomień o nadchodzących kończących się terminach gwarancji. | Wysoki | Automatyczne pojawianie się powiadomień  o kończących się terminach gwarancji. |

## Inne wymagania produktowe

Tabela 3. Przedstawia dodatkowe wymagania, jakie musi spełnić przyszły produkt. System powinien być uniwersalny pod względem użytkowania bez względu na grupę użytkowników korzystających z niego. Dodatkowo, powinien być zrealizowany na urządzenia mobilne z wbudowanym aparatem umożlwiającym robienie zdjęć. Aplikacja jest ściśle związana z możliwością wykonywania zdjęć więc istotnym staje się posiadać taką możliwość. Mobilność telefonów komórkowych powinna zapewnić łatwy dostęp do paragonów.

*Tabela 3 Inne wymagania produktowe*

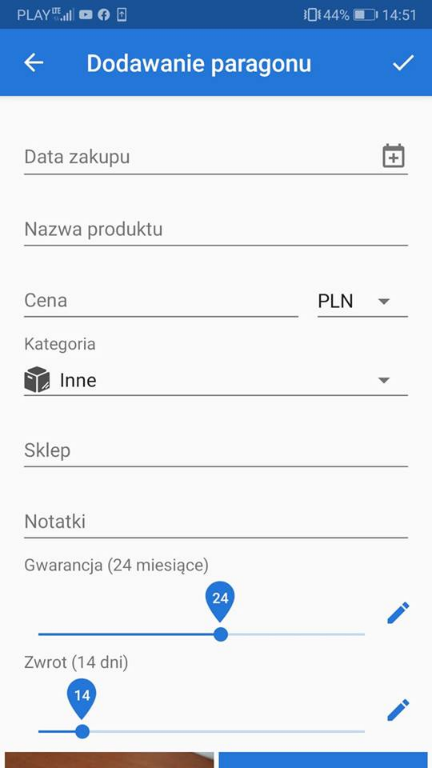
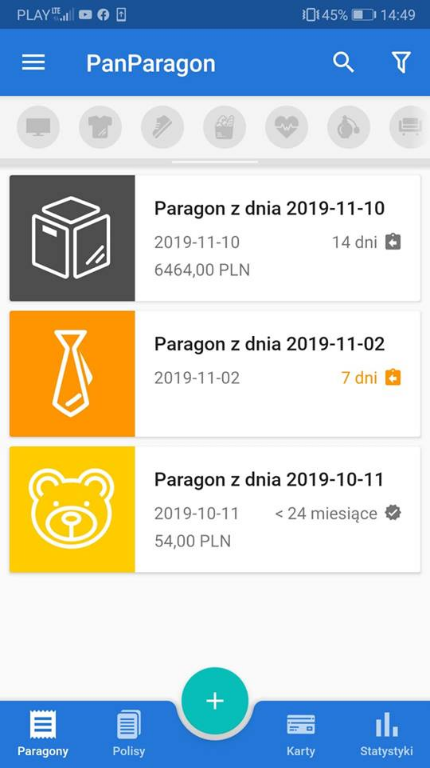
|  |  |  |
| --- | --- | --- |
| Potrzeba | Priorytet | Cechy |
| Kompatybilność z systemami Android w wersji co najmniej 9.0 | Wysoki | Możliwość zainstalowania i użytkowania aplikacji na systemach z androidem w wersji co najmniej 9.0. |
| Przechowywanie danych użytkownik w chmurze | Wysoki | Możliwość rejestrowania i logowania się  w aplikacji oraz wczytywania danych z chmury. |
| Wbudowany aparat w urządzeniu mobilnym | Wysoki | Obowiązek posiadania modułu zajmującego się wykonywaniem zdjęć w telefonie komórkowym na którym znajduje się system użytkownika aplikacji. |

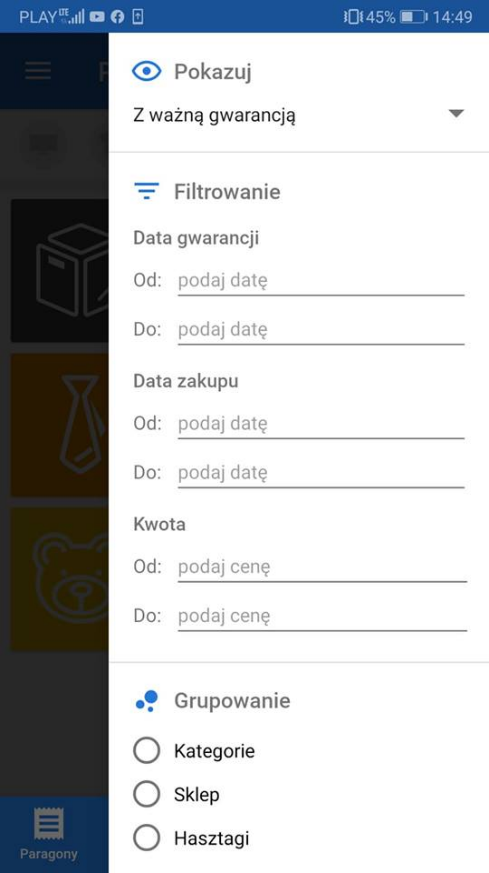
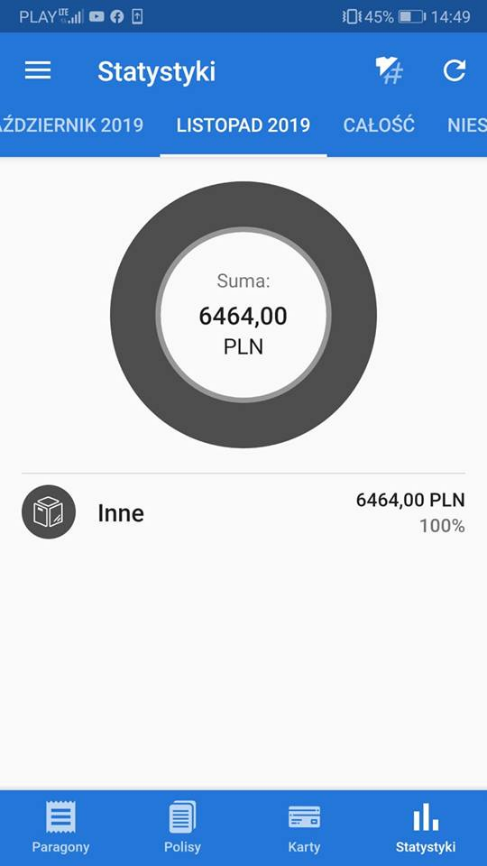
# Rozwiązania konkurencyjne

W tym rozdziale zostaną przedstawione rozwiązania konkurencyjne, które w całości lub po części pozwalają na organizacje paragonów.

## PanParagon

Pierwsza na liście rozwiązań konkurencyjnych jest najbardziej popularna aplikacja mobilna na systemy Android, co za tym idzie największa konkurencja dla projektowanego systemu. PanParagon [6] oferuje przechowywanie (rysunek 3.1), dodawanie (rysunek 3.2)   
i zarządzanie paragonami. Nie posiada jednak możliwości dodawania wielu paragonów naraz. Co więcej jest to aplikacja darmowa, bez reklam, z możliwością wykupienia konta Premium, dostarczającego drobne udogodnienia związane z możliwością prowadzenia statystyk wewnątrz aplikacji. Mimo tego darmowa wersja aplikacji pozwala na skanowanie paragonów i ekstrahowanie danych z nimi związanych poprzez wbudowany w aplikacji moduł skanujący. Aplikacja posiada możliwość prowadzenia statystyk z podziałem na kategorię oraz miesiące (rysunek 3.3). Pozwala także na filtrowanie paragonów (rysunek 3.4). Proces samego filtrowania jest jednak utrudniony i mógłby zostać uproszczony. PanParagon jest zaprojektowany nie tylko pod przechowywanie i kategoryzowanie paragonów. Może służyć także do przechowywania polis oraz kart lojalnościowych, co nie należy do funkcjonalności projektowanego systemu. Nie pozwala jednak na tworzenie własnych kategorii. Brak tej funkcjonalności w aplikacji powoduje ograniczenie w jej użytkowaniu i wielu użytkowników w komentarzach pod aplikacją w Google Play [7] zwraca na to uwagę. PanParagon po zalogowaniu się do aplikacji pozwala na przechowywanie danych w chmurze oraz pozwala na korzystanie z konta Gościa bez konieczności logowania się. Dane są wtedy przechowywane lokalnie i mogą być synchronizowane po zarejestrowaniu się. Aplikacja pozwala na udostępnianie pojedynczych zdjęć paragonów. Nie pozwala na eksport wielu paragonów jednocześnie. PanParagon pozwala na informowanie o kończących się terminach gwarancji poprzez powiadomienia.





*Rysunek 3.4 – PanParagon* [6]*, widok filtrowania paragonów. [Źródło: opracowanie własne]*

*Rysunek 3.3 – PanParagon* [6]*, widok statystyk dla listopada 2019 r. [Źródło: opracowanie własne]*

*Rysunek 3.1 – PanParagon* [6]*. ekran główny, widok paragonów [Źródło: opracowanie własne]*

*Rysunek 3.2 – PanParagon* [6]*, widok dodawania paragonu po wykonaniu wcześniej zdjęcia [Źródło: opracowanie własne]*

## Paragon: Paragony i karty lojalnościowe w jednym miejscu

Kolejnym rozwiązaniem konkurencyjnym jest zdecydowanie mniej zaawansowana od poprzednich rozwiązań konkurencyjnych aplikacja mobilna służąca do przechowywania paragonów (rysunek 3.5) oraz kart lojalnościowych. Jest darmowa i do działania wymaga logowania. Dzięki temu oferuje dostęp do danych na różnych urządzeniach mobilnych. Posiada funkcję wyszukiwania paragonów oraz ich filtrowania (rysunek 3.6). Opcje wyszukiwania i filtrowania są jednak bardzo ubogie i nie pozwalają na bardziej zaawansowane przeszukiwanie danych np.: po datach dodania paragonu lub po datach kończących terminy gwarancji. Pomimo bardzo przejrzystego interfejsu proces dodawania paragonów nie należy do najwygodniejszych i jest czasochłonny (rysunek 3.7). Co więcej aplikacja nie posiada opcji skanowania paragonów i ekstrahowania z nich danych. Największym minusem aplikacji jest brak możliwości kategoryzowania paragonów. Jedyną formą grupowania paragonów jest możliwość wyboru koloru ikony znajdującej się po lewej stronie paragonu w oknie głównego ekranu. Brakuje również możliwości dodawania wielu paragonów naraz. Istnieje możliwość eksportowania zdjęć pojedynczych paragonów na wskazany e-mail. Aplikacja nie oferuje wysyłania więcej niż jednego paragonu. Co więcej aplikacja informuje na temat kończących się terminów gwarancji lecz robi to tylko z poziomu aplikacji (nie wysyła powiadomień po za aplikacją) za pomocą informacji kolorystycznych.



*Rysunek 3.6 – Paragon: Paragony i karty lojalnościowe w jednym miejscu* [12]*, widok filtrowania paragonów [Źródło: opracowanie własne]*

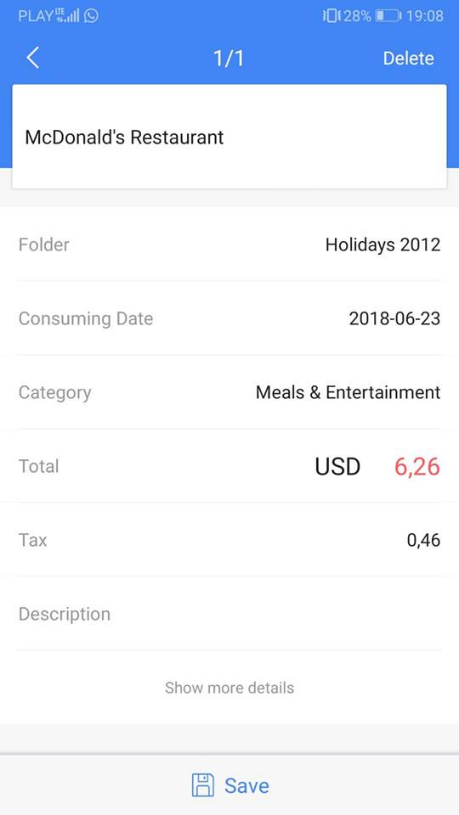
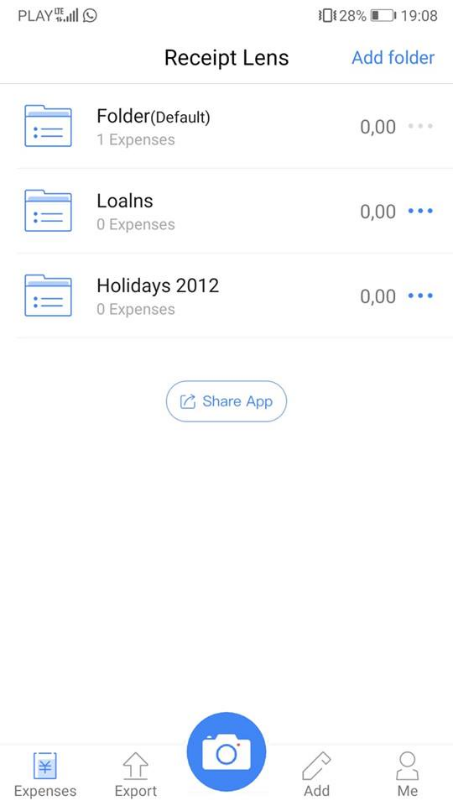
*Rysunek 3.5 – Paragon: Paragony i karty lojalnościowe w jednym miejscu* [12] *, widok zarządzania paragonami [Źródło: opracowanie własne]*



*Rysunek 3.7 – Paragon: Paragony i karty lojalnościowe w jednym miejscu* [12] *, widok dodawania paragonu [Źródło: opracowanie własne]*

## Receipt Lens-Expense Tracking & Reporting

Organizacja paragonów może służyć także do zarządzania wydatkami. Na tym aspekcie skupia się właśnie następne w kolei rozwiązanie konkurencyjne. Jest nią aplikacja mobilna służąca do zarządzania wydatkami. To co wyróżnia ją na tle innych aplikacji jest to, że cała aplikacja skupia się na tworzeniu tak zwanych folderów, do których dodaje się paragony. Główny widok aplikacji zawiera więc foldery wraz z informacją jaka kwota sumarycznie się składa ze wszystkich zgromadzonych w niej rachunków (rysunek 3.8). Interfejs użytkownika jest bardzo czytelny i nowoczesny. Co więcej aplikacja pozwala na skanowanie i dodawanie wielu paragonów naraz. Sam proces dodawania jest ograniczony i trudny   
w użytkowaniu (rysunek 3.9). Jako pierwsza aplikacja na liście rozwiązań konkurencyjnych, oprócz możliwości grupowania paragonów, posiada możliwość eksportowania całych folderów, a co za tym idzie, wielu paragonów naraz. Aplikacja nie oferuje możliwości prowadzenia żadnych statystyk, co jest dziwne, zważywszy na fakt, że jak wskazuje tytuł aplikacji, ma ona służyć do śledzenia i raportowania o wydatkach. Posiada możliwość wyszukiwania paragonów w obrębie grupy ze względu na datę dodania. Ponadto nie posiada możliwości dodawania wielu paragonów naraz. Nie przewiduje także związanych   
z wydatkami i paragonami gwarancji, już nie wspominając o informowaniu użytkowników o kończonych się terminach gwarancji. Aplikacja posiada możliwość tworzenia konta   
i przechowywania danych w chmurze. Można z niej korzystać lokalnie bez logowania się.

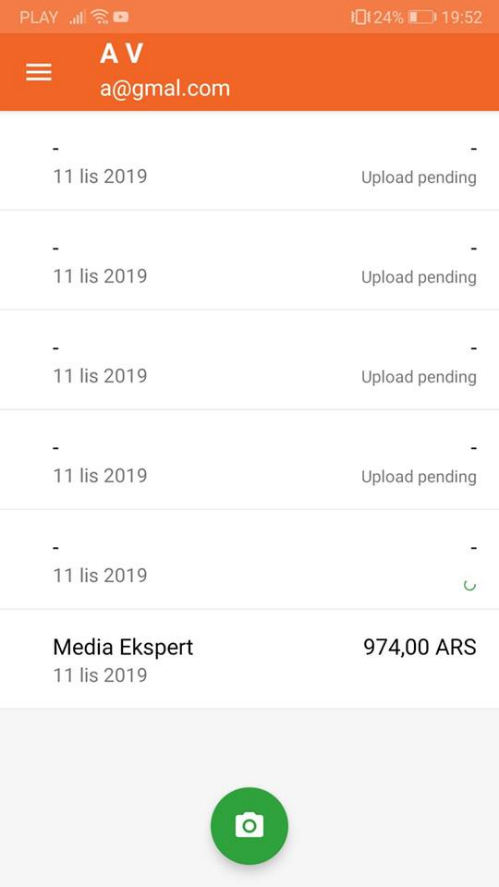
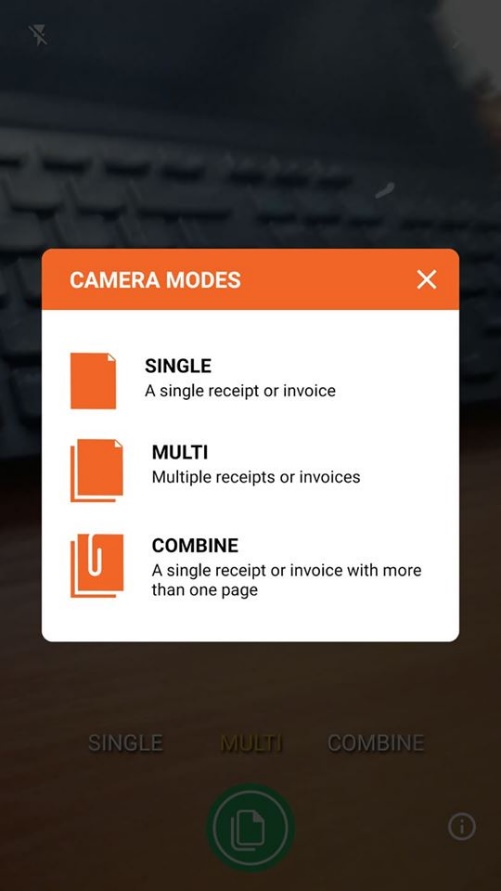


*Rysunek 3.8 - Receipt Lens-Expense Tracking & Reporting* [10]*, widok ekranu głównego z folderami. [Źródło: opracowanie własne]*

*Rysunek 3.9 - Receipt Lens-Expense Tracking & Reporting* [10]*, widok dodawania wydatku do folderu. [Źródło: opracowanie własne]*

## Receipt Bank

Kolejny przedstawiony konkurencyjny system do organizacji paragonów nie należy do najbardziej zaawansowanych aplikacji mobilnych na rynku. Wymaga logowania i nie pozwala na działania bez dostępu do Internetu. Posiada tylko możliwość dodawania paragonów z podstawowymi informacjami. Widok główny aplikacji (rysunek 3.10) posiada tylko paragony i guzik pozwalający na dodanie nowych pozycji na liście. Nie posiada możliwości skanowania paragonów ani nawet ich filtrowania w celu szybkiego ich przeszukiwania. Nie posiada żadnych informacji na temat gromadzonych paragonów takich jak łączna suma, kategoryzowania, czy informacja na temat gwarancji. Jest ograniczona   
w swoim działaniu, a jednak posiada dwie funkcjonalności wyróżniające je na tle konkurencji. Po pierwsze jako jedyne z rozwiązań konkurencyjnych posiada możliwość dodawania pojedynczych paragonów, wielu paragonów naraz lub wielu zdjęć do jednego paragonu (rysunek 3.11). Przy starcie aplikacji, zakładając, że wcześniej się zalogowano do niej, od razu przenosi użytkownika do możliwości dodania paragonów. Jest to aplikacja ukierunkowana na proste i szybkie przechowywanie paragonów. Umożliwia także eksportowanie pojedynczych paragonów. Jest zdecydowanie najszybsza pod tym względem i można wyciągnąć z tego wnioski.



*Rysunek 3.10 – Receipt Bank* [11]*, widok ekranu głównego. [Źródło: opracowanie własne]*

*Rysunek 3.11 – Receipt Bank* [11]*, widok opcji dodawania paragonów. [Źródło: opracowanie własne]*

## Podsumowanie

Podsumowując, istnieje wiele ciekawych rozwiązań pomagających w organizowaniu paragonów. Każde z nich pozwala na inne doświadczenie i skupia się na innym aspekcie związanym z organizacją paragonów. Oferują więc różne rozwiązania, które nie zawsze spełniają potrzeby wszystkich grup użytkowników. Proponowane w pracy rozwiązanie powinno trafić do jak najszerszej grupy użytkowników. W tabeli 4. Przedstawiono zestawienie głównych funkcjonalności oferowanych przez wcześniej przeanalizowane systemy i porównano je z proponowanymi w pracy rozwiązaniem (ostatnia kolumna).

*Tabela 4 Porównanie oferowanych funkcjonalności*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PanParagon | Paragon: Paragony i karty lojalnościowe w jednym miejscu | Receipt Lens-Expense Tracking & Reporting | Receipt Bank | Proponowane rozwiązanie |
| Dodawanie paragonów | Pojedynczo | Pojedynczo | Pojedynczo lub kilka naraz | Pojedynczo lub kilka naraz | Pojedynczo lub kilka naraz |
| Skanowanie paragonów | + |  | + |  | + |
| Kategoryzowanie | +  Ustalone z góry kategorie |  | +  System tworzonych przez użytkownika folderów | +  Ustalone z góry kategorie | +  System tworzenia przez użytkownika grup |
| Przeszukiwanie paragonów | + Zaawansowane filtrowanie |  | +  Tylko po datach dodania i w obrębie jednego folderu |  | + |
| Sortowanie paragonów | + | + |  |  |  |
| Eksportowanie paragonów | +  Tylko pojedyncze paragony | +  Tylko pojedyncze paragony (tylko drogą mailową) | +  Pojedyncze foldery z paragonami lub wiele folderów (tylko drogą mailową) | +  Tylko pojedyncze paragony | +  Pojedyncze paragony (tylko drogą mailową) |
| Przechowywanie informacje na temat gwarancji | + | + |  |  | + |
| Powiadamianie o kończonych się terminach gwarancji | + |  |  |  | + |

# Specyfikacja i analiza wymagań

## Słownik pojęć

Tabela 5. przedstawia słownik kluczowych dla pracy pojęć wykorzystywanych   
w systemie.

*Tabela 5 Słownik pojęć*

|  |  |  |
| --- | --- | --- |
| Termin | Synonim | Definicja |
| Eksport paragonów | Wysyłka paragonów | Oznacza wysłanie zdjęcia lub zdjęć paragonów poza aplikację na dany adres mailowy. |
| Grupa | Grupa paragonów | Grupy służą do grupowania paragonów. Posiadają informację na temat liczby znajdujących się w niej paragonów. |
| Gwarancje |  | Ważne czasowo zobowiązanie producenta lub sprzedawcy do bezpłatnej naprawy lub wymiany zakupionego, wadliwego towaru. Gwarancja jest ważna dopóki posiadany jest dowód zakupu (w aplikacji jest to zdjęcie paragonu). |
| Ikona |  | Obraz o danym kolorze służący do łatwego rozróżniania grup. |
| Paragon | Dowód zakupu, Rachunek, | Unikatowy dokument papierowy, który świadczy dowód zakupu produktu lub usługi. |
| Powiadomienie |  | To czynne poinformowanie użytkownika aplikacji  o sytuacji zaistniałej w systemie. |
| Filtrowanie | Przeszukiwanie | Możliwość wyodrębniania poszczególnych elementów  z pośród listy tych elementów na podstawie unikatowych informacji posiadanych przez elementy. |
| Skanowanie | Skanowanie paragonu | Możliwość ekstrahowania informacji na podstawie zdjęcia paragonu. |
| Termin gwarancji |  | Graniczna data, do której istnieje możliwość oddania zakupionego, wadliwego towaru na gwarancje. |
| Użytkownik |  | Osoba będąca w posiadaniu aplikacji i korzystająca z jej funkcjonalności. |
| Znacznik | Tag | Słowo lub słowa kluczowe przechowujące informację na temat paragonu. |

## Reguły biznesowe i model domenowy

Zdefiniowano reguły biznesowe, zgodnie z którymi wykonano aplikację. Reguły zostały zapisane językiem naturalnym i posiadają identyfikator „REG00x”, gdzie x to numer reguły (czyt. Reguła nr x). Na rysunku 4.1 przedstawiono model domenowy, będący odwzorowaniem systemu zgodnego z regułami biznesowymi.

**Użytkownik**

REG001. Użytkownik może lecz nie musi posiadać wiele paragonów

REG002. Użytkownik musi posiadać co najmniej jedną grupę

REG003. Użytkownik musi posiadać nazwę użytkownika

REG004. Nazwa użytkownika musi być unikalna­

REG005. Użytkownik musi posiadać adres email

REG006. Adres email musi być unikalny

**Paragon**

REG007. Paragon musi należeć tylko do jednego użytkownika

REG008. Paragon musi należeć tylko do jednej grupy

REG009. Paragon musi posiadać datę dodania

REG010. Paragon może posiadać datę kończącą gwarancję

REG011. Data kończąca gwarancję musi być późniejsza niż data dodania

REG012. Paragon musi posiadać nazwę paragonu

REG013. Paragon nie musi posiadać żadnego znacznika

REG014. Paragon musi posiadać co najmniej jedno zdjęcie

REG015. Paragon musi posiadać kwotę pieniężną (znajdującą się na paragonie)

REG016. Suma pieniężna musi być większa lub równa zero

**Znacznik**

REG017. Znacznik musi należeć do co najmniej jednego paragonu

REG018. Znacznik musi posiadać nazwę znacznika

REG019. Nazwa znacznika musi być unikalna

REG020. Nazwa znacznika musi rozpoczynać się od znaku „#”

**Grupa**

REG021. Grupa musi być przypisana do użytkownika

REG022. Do grupy nie musi być przypisany żaden paragon

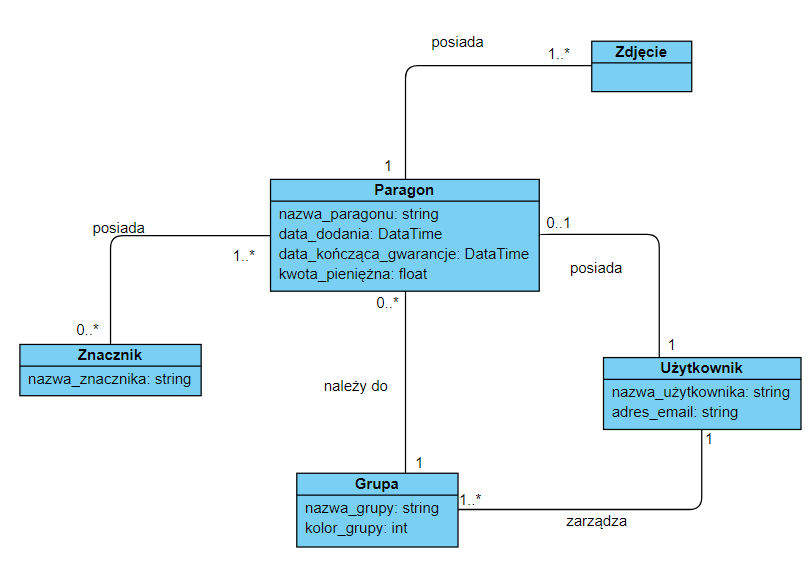
REG023. Grupa musi posiadać nazwę grupy

REG024. Nazwa grupy musi być unikalna

REG025. Grupa musi posiadać kolor grupy

**Zdjęcie**

REG026. Zdjęcie musi być przypisane do paragonu



*Rysunek 4.1 Diagram klas (domenowy)*

*[Źródło: opracowanie własne]*

## Historyjki użytkowników

Po analizie funkcjonalności systemu przedstawiono wymagania systemu w formie historyjek użytkownika. W tabeli 6 przedstawiono macierz śladowania wymagań do cech systemu aby sprawdzić czy wszystkie zebrane wymagania zostały spełnione. W kolumnach zostały przedstawione cechy systemu, a w wierszach historyjki użytkowników.

**Zarządzanie paragonami:**

**HU01.** Jako użytkownik chcę mieć możliwość dodawania paragonów wraz z informacjami ich dotyczącymi, aby móc je organizować w celu późniejszego wykorzystania.

**HU02.** Jako użytkownik chcę mieć możliwość przeglądania listy paragonów, aby móc wybierać i zarządzać interesującymi mnie paragonami.

**HU03.** Jako użytkownik chcę mieć możliwość usuwania wcześniej dodanych paragonów, aby móc porządkować paragony, których organizacją się zajmuje.

**HU04.** Jako użytkownik chcę mieć możliwość edycji dodanych paragonów, aby móc poprawiać istniejące informacje lub uzupełniać je o brakujące.

**HU05.** Jako użytkownik chcę mieć możliwość przypisywania do grupy paragonów, aby móc je sprawniej organizować.

**HU06.** Jako użytkownik chcę mieć możliwość skanowania paragonów, aby móc szybko ekstrahować informacje z nich bez potrzeby tracenia czasu na ręczne ich uzupełnianie.

**Zarządzanie grupami:**

**HU07.** Jako użytkownik chcę mieć możliwość tworzenia grup dla paragonów, aby móc je kategoryzować do łatwiejszej ich organizacji.

**HU08.** Jako użytkownik chcę mieć możliwość przeglądania grup paragonów, aby móc zarządzać nimi oraz przeglądać informacje związane z nimi.

**HU09.** Jako użytkownik chcę mieć możliwość usuwania grup dla paragonów, gdy uznam, że nie są już potrzebne w procesie organizacji paragonów.

**HU10.** Jako użytkownik chcę mieć możliwość edycji grup dla paragonów, aby móc zmieniać informacje na temat grupy w razie potrzeby.

**Przeszukiwanie paragonów:**

**HU11.** Jako użytkownik chcę mieć możliwość filtrowania paragonów, aby móc szybko odnaleźć jeden lub wiele z interesujących mnie z zapisanych wcześniej paragonów.

**Eksportowanie paragonów:**

**HU12.** Jako użytkownik chcę mieć możliwość eksportowania zdjęć paragonów, aby móc je okazać przy składaniu reklamacji.

**Terminy kończące ważność gwarancji:**

**HU13.** Jako użytkownik chcę mieć możliwość bycia informowanym   
o nadchodzących terminach kończących ważność gwarancji, aby móc wiedzieć czy istnieje jeszcze taka możliwość w przypadku wady zakupionego produktu.

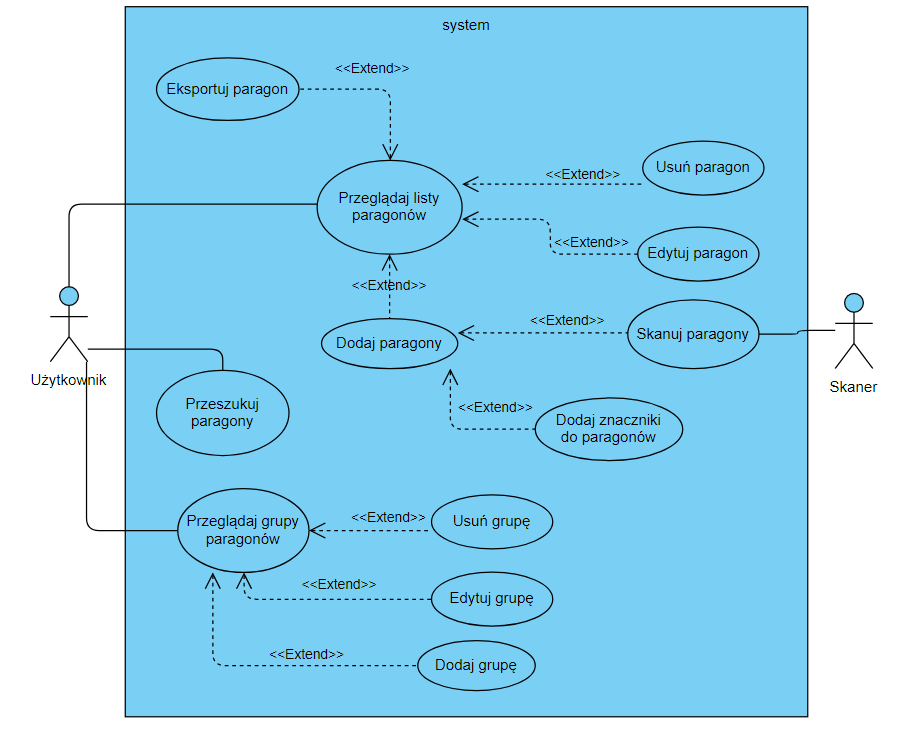
*Tabela 6 Macierz śladowania wymagań*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Skanowanie paragonów | Przeglądanie paragonów | Dodawanie paragonów | Edycja paragonów | Usuwanie paragonów | Kategoryzowanie paragonów | Dodawanie grup paragonów | Przeglądanie grup paragonów | Edycja grup paragonów | Usuwanie grup paragonów | Filtrowanie paragonów | Eksportowanie paragonów | Informacje na temat gwarancji |
| HU01. |  | + |  |  |  |  |  |  |  |  |  |  |  |
| HU02. |  |  | + |  |  |  |  |  |  |  |  |  |  |
| HU03. |  |  |  |  | + |  |  |  |  |  |  |  |  |
| HU04. |  |  |  | + |  |  |  |  |  |  |  |  |  |
| HU05. |  |  |  |  |  | + |  |  |  |  |  |  |  |
| HU06. | + |  |  |  |  |  |  |  |  |  |  |  |  |
| HU07. |  |  |  |  |  |  | + |  |  |  |  |  |  |
| HU08. |  |  |  |  |  |  |  | + |  |  |  |  |  |
| HU09. |  |  |  |  |  |  |  |  |  | + |  |  |  |
| HU10. |  |  |  |  |  |  |  |  | + |  |  |  |  |
| HU11. |  |  |  |  |  |  |  |  |  |  | + |  |  |
| HU12. |  |  |  |  |  |  |  |  |  |  |  | + |  |
| HU13. |  |  |  |  |  |  |  |  |  |  |  |  | + |

­­

## Diagram Przypadków użycia

Aby zamodelować funkcjonalności systemu wykorzystano diagram przypadków użycia. Diagram został zademonstrowany na rysunku 4.2. Do wykonania wskazanych przypadków użycia wymagana jest poprawna autentykacja użytkownika. Na diagramie nie zostały przestawione przypadki użycia związane z logowaniem i rejestracją użytkowników.



*Rysunek 4.2 Diagram przypadków użycia*

*[Źródło: opracowanie własne]*

## Skrócony opis przypadków użycia

Tabela 7. Przedstawia skrócony opis przypadków użycia. Opisy mają za zadanie przybliżać funkcjonalności jakie przyszły system ma oferować w ramach konkretnych akcji jakie ma podejmować użytkownik systemu.

Tabela Skrócone opisy przypadków użycia

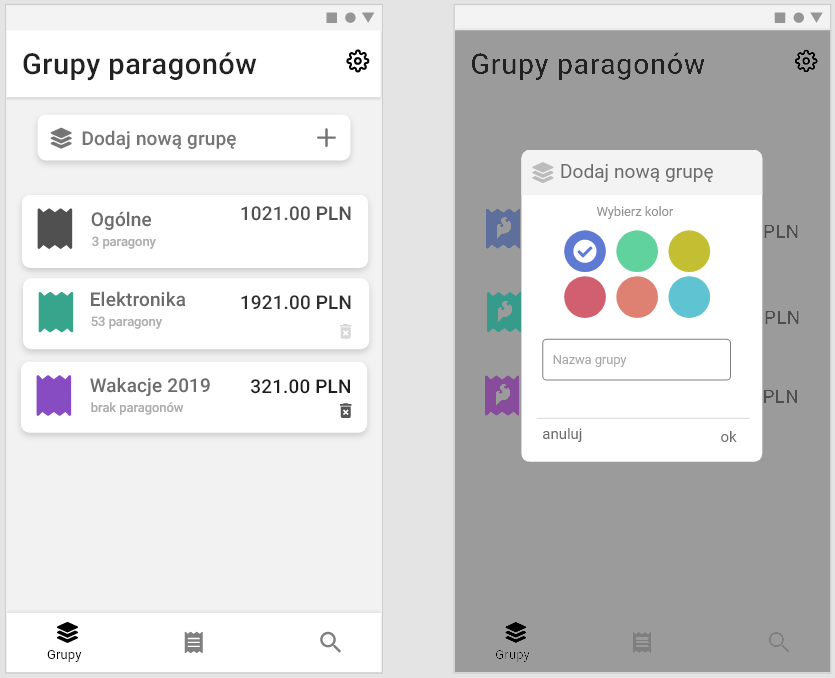
|  |  |
| --- | --- |
| Nazwa przypadku użycia | Skrócony opis |
| Dodaj grupę | System umożliwia dodanie nowej grupy paragonów o podanych parametrach. |
| Dodaj paragony | System umożliwia dodanie nowych paragonów o podanych parametrach. |
| Edytuj grupę | System umożliwia zmianę parametrów istniejącej grupy paragonów. |
| Edytuj paragon | System umożliwia zmianę parametrów istniejącego paragonu |
| Eksportuj paragon | System umożliwia eksport zdjęć związanych z paragonami na podanego wcześniej adres emaila. |
| Przeglądaj grupy paragonów | System wyświetla listę grup paragonów. |
| Przeglądaj listy paragonów | System wyświetla listę paragonów. |
| Przeszukuj paragony | System umożliwia przeszukiwanie z pośród wszystkich paragonów jakie zostały dodane do systemu te, o wskazanych przez użytkownika parametrach. |
| Przypisz paragony do grup | System umożliwia przypisanie danego paragonu do już istniejącej grupy podczas dodawania nowego paragonu |
| Skanuj paragony | System umożliwia na ekstrahowanie niektórych informacji ze zdjęcia paragonu wykonanego. |
| Usuń grupę | System umożliwia usunięcie istniejącej grupy. |
| Usuń paragon | System umożliwia usunięcie istniejącego paragonu. |

## Scenariusze przypadków użycia wraz z prototypami interfejsów

W następującym podrozdziale przedstawiono przykładowe przebiegi przypadków użycia wraz ze scenariuszami alternatywnymi i wyjątkowymi. W tabeli 8. Przedstawiono przebieg scenariusza związanego z oddawaniem nowej grupy. Następnie na rysunkach 4.3 oraz 4.4 przedstawiono kolejno ekrany związane z przedstawionym scenariuszem.

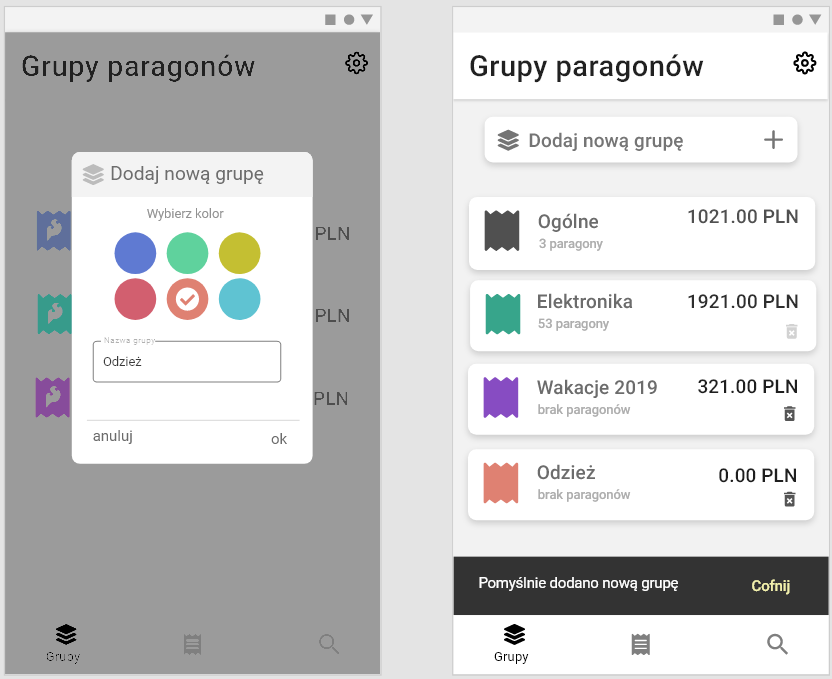
*Tabela 8 Dodaj grupę*

|  |  |
| --- | --- |
| Nazwa przypadku użycia | Dodaj grupę |
| Aktorzy | Użytkownik |
| Warunki wstępne | Użytkownik jest zalogowany, system wyświetla listę grup. [Rysunek 4.3] (PU przeglądaj grupy paragonów) |
| Warunki końcowe | Do systemu zostaje dodana nowa grupa paragonów |
| Scenariusz główny | 1. Użytkownik zleca dodanie grupy paragonów. 2. System wyświetla parametry do wypełnienia dla nowej grupy paragonów (nazwa grupy, kolor grupy). [Rysunek 4.3] 3. Użytkownik wprowadza dane nowej grupy paragonów.[Rysunek 4.4] 4. Użytkownik zatwierdza akcję dodania. 5. System sprawdza poprawność danych (REG021) 6. System zapamiętuje grupę paragonów i wyświetla powiadomienie o jego dodaniu. [Rysunek 4.4] 7. Zakończenie scenariusza. |
| Scenariusz alternatywny 1 | 4a. Użytkownik anuluje akcję dodania.   1. Powrót do punktu 1. Scenariusza głównego. |
| Scenariusz alternatywny 2 | 3a. Użytkownik zatwierdza akcję dodania przed wypełnieniem pól.   1. System wyświetla powiadomienie o braku obowiązkowych danych. 2. Następuje powrót do punktu 2. Scenariusza głównego. |
| Scenariusz wyjątku | 5\* W systemie istnieje grupa paragonów o takiej nazwie (REG024).   1. System wyświetla powiadomienie o tym, że grupa o podanej nazwie już istnieje w systemie. 2. Następuje powrót do punktu 1. Scenariusza głównego. |



*Rysunek 4.3 Dodaj nową grupę paragonów – Lista grup (po lewej).   
Widok dodawania grupy (po prawej)*

[Źródło: opracowanie własne]



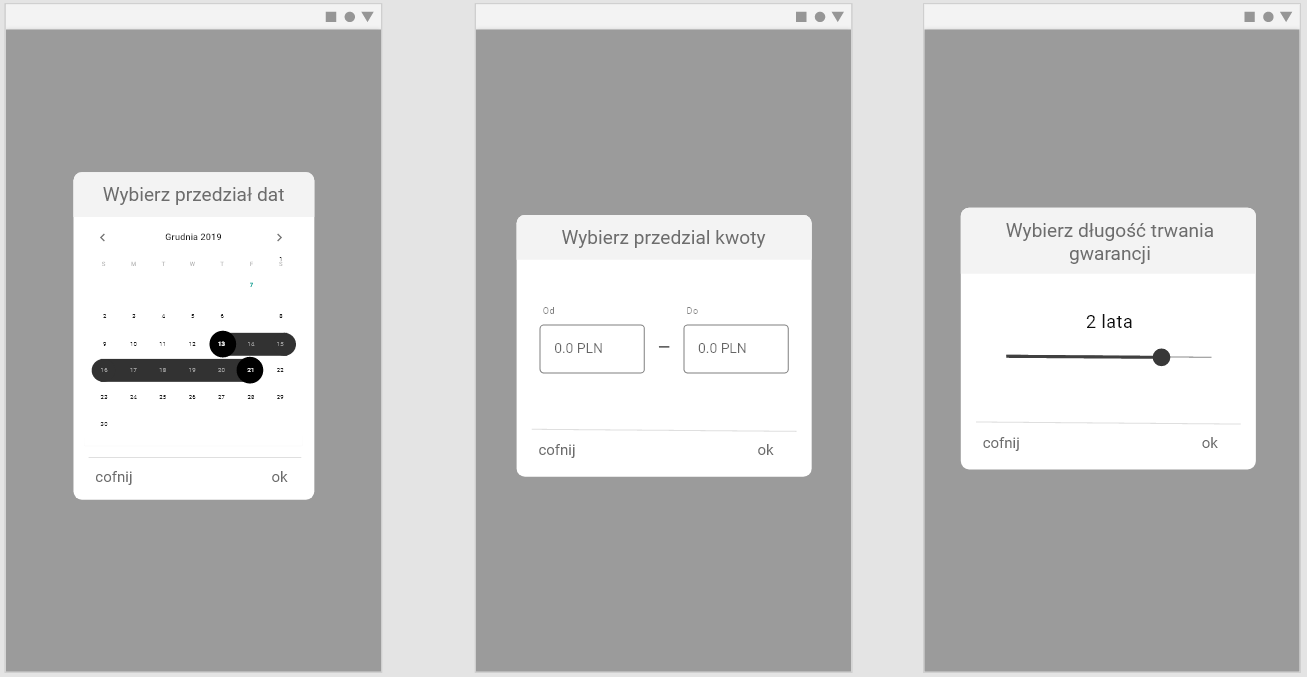
*Rysunek 4.4* *Dodaj nową grupę paragonów – Uzupełnienie parametrów grupy (po lewej).   
Dodanie nowej grupy do listy grup wraz z komunikatem (po prawej).*

[Źródło: opracowanie własne]

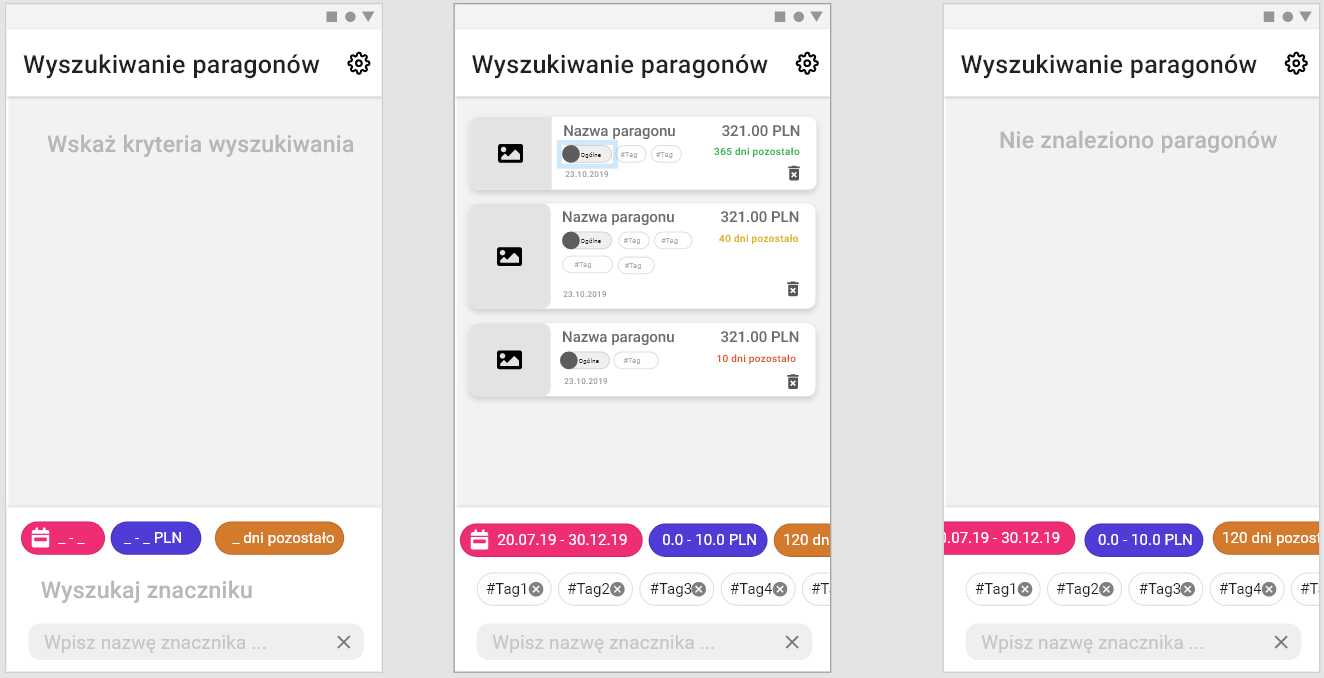
W tabeli 9. Przedstawiono przebieg scenariusza związanego z przeszukiwaniem paragonów. Następnie na rysunkach 4.5 oraz 4.6 przedstawiono kolejno ekrany związane ze przedstawionym scenariuszem.

*Tabela 9 Przeszukuj paragony - Scenariusz*

|  |  |
| --- | --- |
| Nazwa przypadku użycia | Przeszukuj paragony |
| Aktorzy | Użytkownik |
| Warunki wstępne | Użytkownik jest zalogowany. |
| Warunki końcowe | System wyświetla paragony wskazane przez kryteria wyszukiwania. |
| Scenariusz główny | 1. Użytkownik zleca przeszukiwanie paragonów 2. System wyświetla możliwe kryteria wyszukiwania: [Rysunek 4.5]   - daty dodania paragonu  - sumy pieniężne na paragonach,  - gwarancje paragonów,  - znaczniki na paragonach.   1. Użytkownik wybiera kryteria wyszukiwania 2. System wyszukuje paragony według wskazanych przez użytkownika kryteriów wyszukiwania. [Rysunek 4.6] 3. System wyświetla znalezione paragony według kolejności wyszukania. Wyświetlone dane to: nazwa paragonu, znaczniki należące do paragonu, długość trwania gwarancji przypisana do paragonu, kwota widniejąca na paragonie, grupa, do której należy paragon oraz miniaturka paragonu. [Rysunek 4.6] 4. Zakończenie scenariusza |
| Scenariusz alternatywny 1 | 3a. Użytkownik anuluje akcję wyszukiwania paragonów  Powrót do punktu 1. Scenariusza głównego |
| Scenariusz alternatywny 2 | 5a System nie znalazł żadnych paragonów pasujących do wskazanych przez użytkownika kryteriów wyszukiwania   1. System wyświetla komunikat o braku zgadzających się z kryteriami wyszukiwania paragonów [Rysunek 4.6]   Następuje powrót do punktu 2. Scenariusza głównego |



*Rysunek 4.5* Przeszukaj paragony *– kryteria wyszukania. Wybór daty dodania paragonu (po lewo), wybór kwot (środek) oraz wybór długości trwania gwarancji (po prawo)*

[Źródło: opracowanie własne]

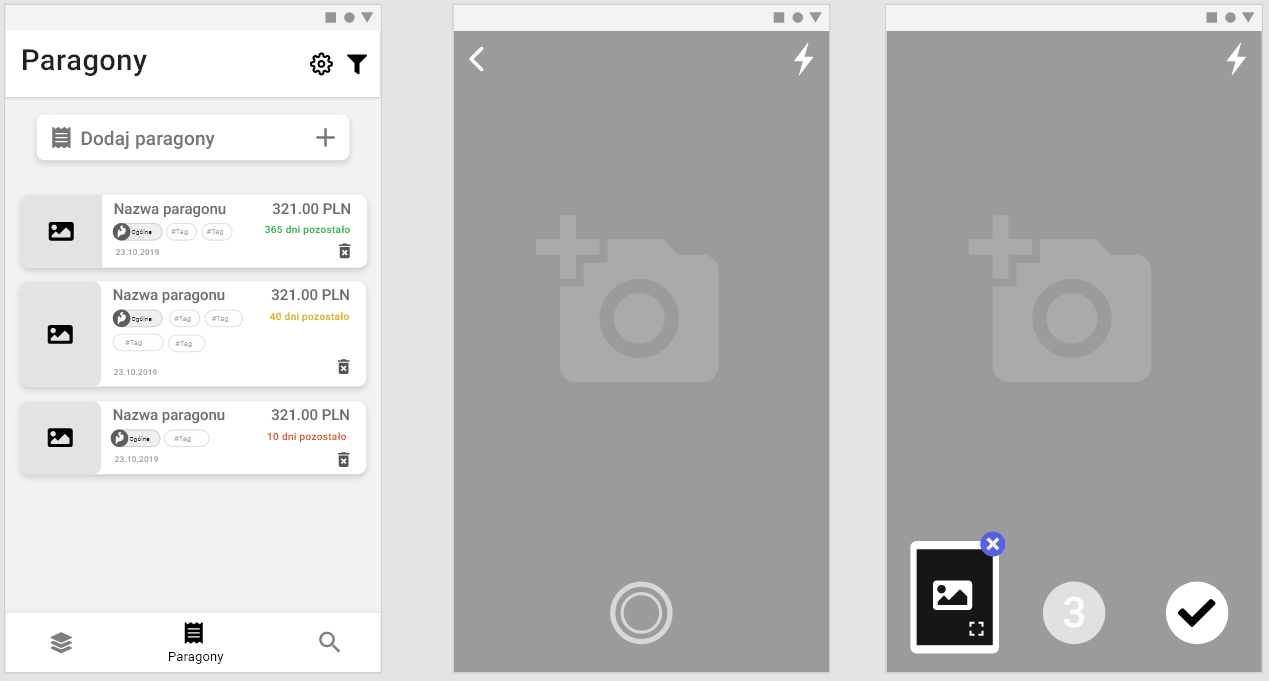
*Rysunek 4.6 Przeszukaj paragony – brak kryteriów (po lewo), wypełnienie kryteriów i wynik wyszukania (środek), komunikat w przypadku braku znalezienia paragonów dla poszczególnych kryteriów.*

[Źródło: opracowanie własne]

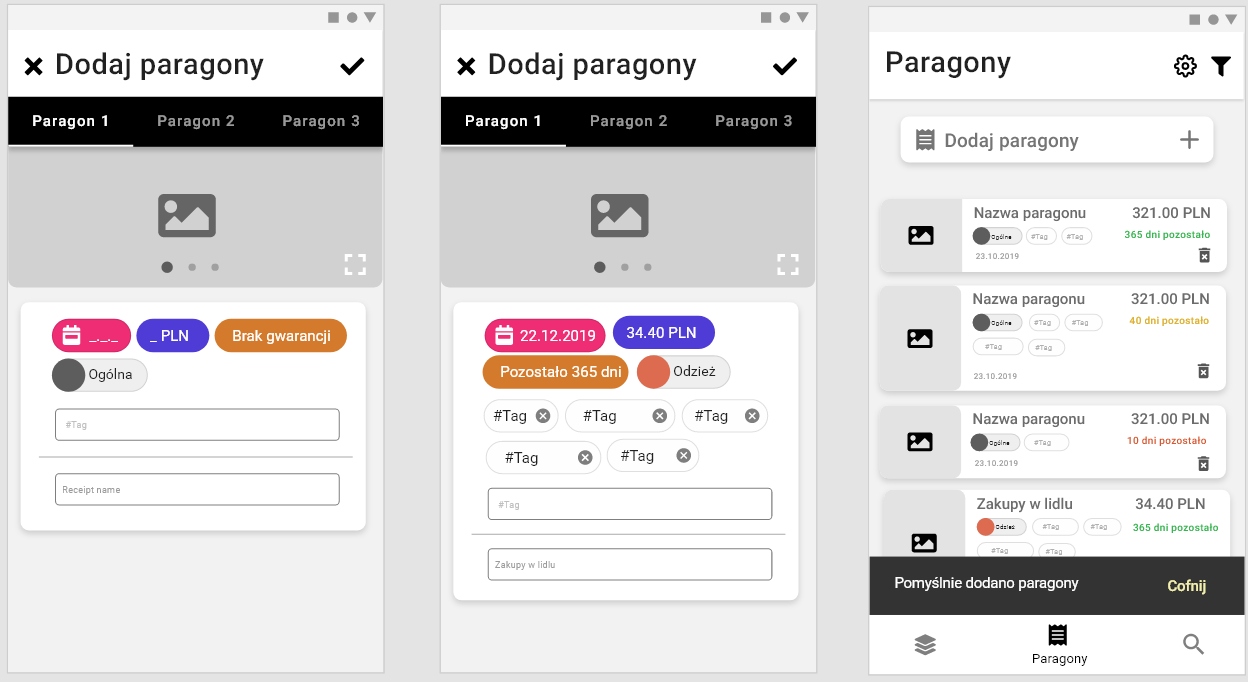
W tabeli 11. Przedstawiono przebieg scenariusza związanego z dodawaniem paragonów. Następnie na rysunkach 4.9, 4.10 oraz 4.11 przedstawiono kolejno ekrany związane ze przedstawionym scenariuszem.

*Tabela 10 Dodanie paragonów - Scenariusz*

|  |  |
| --- | --- |
| Nazwa przypadku użycia | Dodaj paragony |
| Aktorzy | Użytkownik, Skaner |
| Warunki wstępne | Użytkownik jest zalogowany, system wyświetla listę paragonów (PU przeglądanie listy paragonów) |
| Warunki końcowe | Do systemu zostaje dodany jeden lub wiele paragonów |
| Scenariusz główny | 1. Użytkownik wybiera akcję dodania nowych paragonów 2. System wyświetla ekran podglądu widoku kamery 3. Użytkownik zleca wykonanie zdjęcia kamerze. 4. System zapamiętuje zdjęcie. 5. System uaktualnia interfejs o dokonane zdjęcie (wyświetla miniaturkę zdjęcia). Uwaga: użytkownik może dodać wiele paragonów na raz, akcje 3-5 mogą być powtarzane wielokrotnie). 6. Użytkownik zatwierdza zdjęcia paragonów 7. Użytkownik zleca zeskanowanie zdjęcia skanerowi 8. Skaner ekstrahuje dane z paragonów (nazwę sklepu, w którym dokonano zakupu, datę zakupu oraz sumaryczną cenę). 9. System zapamiętuje i dodaje dane do poszczególnych paragonów 10. System wyświetla ekran z parametrami paragonów (zdjęcia paragonów, daty dodania paragonów, informacje na temat gwarancji paragonów, suma pieniężna, nazwy paragonów oraz znaczniki związane z paragonami) 11. Użytkownik wypełnia parametry 12. Użytkownik zatwierdza dodawanie paragonów 13. System zapamiętuje paragony i wyświetla powiadomienie o udanym ich dodaniu 14. Zakończenie scenariusza |
| Scenariusz alternatywny 1 | 6a Użytkownik zleca wykonanie kolejnego zdjęcia   1. Następuje punkt 4. Scenariusza głównego |
| Scenariusz alternatywny 2 | 6b Użytkownik usuwa ostatnio dodane zdjęcie   1. System usuwa ostatnio dodane zdjęcie 2. Powrót do punktu 6. Scenariusza głównego |
| Scenariusz alternatywny 3 | 6c Użytkownik anuluje akcje dodawania paragonów   1. Powrót do punktu 1. Scenariusza głównego |
| Scenariusz wyjątku | 7\* Skaner nie rozpoznaje jednego lub kilku paragonów   1. System wyświetla powiadomienie o nieudanym wyekstrahowaniu jednego lub kilku paragonów 2. Powrót do punktu 10. Scenariusza głównego |



Rysunek . *Dodaj paragony – kolejno krok 0 (z lewej) oraz krok 1 (z prawej)*

 *[Źródło: opracowanie własne]*

*Rysunek 4.8 Dodaj paragony –kolejno krok 2 (z lewej) oraz krok 3 (z prawej)*

[Źródło: opracowanie własne]

# Projekt aplikacji ReceiptCarer

Następujący rozdział poświęcony jest projektowi aplikacji. Zaprezentowane analizy i założenia projektowe mają za zadanie ułatwić implementację oraz wyznaczyć kierunek tworzenia aplikacji mobilnej.

## Wykorzystane środowiska i narzędzia programistyczne

Będąca przedmiotem projektu aplikacja mobilna przeznaczona jest na urządzenia mobilne z systemem Android. Całość napisana jest w języku **Java** [8] w środowisku programistycznym **Android Studio** [9]. Wykorzystano możliwości narzędzie **Firebase** [10]do zaimplementowania systemu logowania się i rejestracji do aplikacji. Baza danych została zrealizowana za pomocą **Firebase Cloud Firestore** [10]oraz **Firebase Cloud Storage**. **Firebase Cloud Firestore** stanowi bazę danych gdzie przechowywane są wszystkie dane potrzebne do prawidłowego funkcjonowania aplikacji. **Firebase Cloud Storage** pozwala na przechowywanie plików graficznych (plików które mają rozmiar większy niż 1 mb) i dostęp do nich. Firebase udostępnia także narzędzie **OCR** [11] **(**do rozpoznawania tekstu z plików graficznych)o nazwie **Firebase Vision**. Umożliwiło ono ekstrahowanie tekstu ze zdjęć paragonów. Za pomocą systemu kontroli wersji **GiT** [12]utrzymano stabilność i skalowalność projektu. Do przygotowania wszelkiego rodzaju diagramów zastosowano **Visual Paradigm Online** [13]. Przy użyciu biblioteki **Picasso** [14]zrealizowano dynamiczne pobieranie oraz ładowanie zdjęć zapisanych w Firebase Cloud Storage do aplikacji.

Poniżej znajdują się bardziej szczegółowe informacje na temat poszczególnych narzędzi użytych przy implementacji aplikacji.

**Android Studio**

Android Studio to zintegrowane środowisko programistyczne (IDE [15]) zaprojektowane do produkcji aplikacji mobilnych na systemy Android [16] od firmy Google [1]. Android studio zostało stworzone przez JetBrains [17] na podstawie innego środowiska – IntelliJ IDEA [18]. Jest to podstawowe narzędzie do programowania natywnego z użyciem systemu Android oferujące wiele funkcjonalności takich jak wsparcie dla języków Java oraz Kotlin, narzędzia do refaktoryzacji i optymalizacji kodu, integracja z systemami kontroli wersji takich jak GiT, wbudowany symulator urządzenia mobilnego z systemem Android. Ze względu na fakt, że jest do dedykowane narzędzie do programowania na system Android oraz, że jest wspierane przez firmę zajmującą tworzeniem tego systemu to środowisko programistyczne zostało użyte do implementacji aplikacji projektowanej w tej pracy dyplomowej.

**GIT**

GiT to rozporoszony system kontroli wersji. Stanowi darmowe i otwarte oprogramowanie pomagające w rozwoju oprogramowania poprzez tworzenie repozytoriów zachowujących poszczególne etapy kodu w trakcie jego rozwijania. Pozwala na bezpieczne tworzenie zmian i zminimalizowanie ryzyka utraty danych. Z tych właśnie powodów jest on używany w pracy.

**Visual Paradigm Online**

Visual Paradigm Online [13] jest odpowiednikiem online programu Visual Paradigm przeznaczonego do pomocy w projektowaniu oprogramowania. Pozwala na tworzenie wszelkiego rodzaju diagramów w tym diagramów UML (takich jak diagram przypadków użycia lub diagram ERD). Narzędzie jest proste w obsłudze i zostało wykorzystane w pracy do tworzenia wszelkiego rodzaju diagramów.

**Firebase – Authentication, Cloud Firestore Database, Vision oraz Cloud Firestore Storage**

Firebase jest mobilną i webową platformą programistyczną udsotępniającą wiele produktów związanych z przetwarzaniem danych. Jedną z udostępnianych usług, z której korzysta aplikacj jest Firebase Cloud Firestore [10]. Jest to baza danych NoSQL, w której dane są przechowywane w postaci dokumentów. Te dokumenty są grupowane w kolekcje i wykorzystywane przy organizacji bazy danych oraz do tworzenia zapytań do takiej bazy danych. Cloud Firestore pozwala na przechowywanie i synchronizację danych w chmurze. Pozwala ona wysyłać jedynie dane do maksymalnego rozmiaru 1 mb. Oznacza to, że nie umożliwia przechowywania plików graficznych. Wykorzystywane jest do tego inne narzędzie oferowane przez Firebase: Firestore Cloud Storage. Pliki przechowywane są za jej pomocą w folderach i pod folderach. Dostęp do nich jest poprzez adresy url.

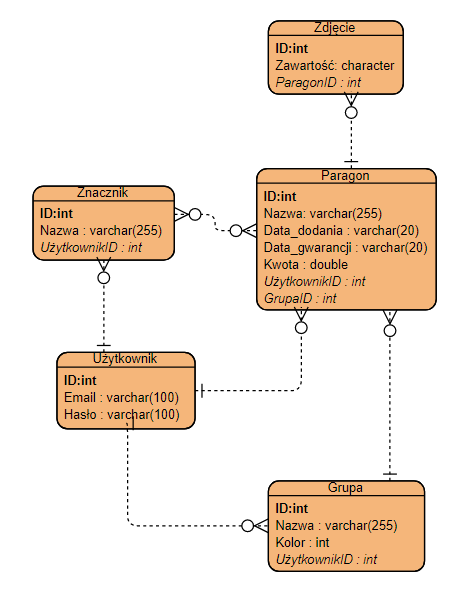
Firebase Authentication jest serwisem oferującym możliwość logowania i rejestracji do systemu. Do skanowania paragonów zostało użyte kolejne narzędzie od Firebase służące do rozpoznawania znaków ze zdjęć o nazwie Firestore Vision. Pozwala na wykorzystanie modułów rozpoznawania tekstu zarówno online jak i offline.

Firebase jest darmowy i pozwala na pełne wykorzystanie wszystkich oferowanych przez siebie narzędzi dopóki aplikacja, która je wykorzystuje jest małych rozmiarów (nie posiada wielu czynnie korzystających z aplikacji użytkowników). Wykorzystanie tylu narzędzi w jednym miejscu jest ogromnym plusem i pozwala na sprawną i szybką implementacje. Z tego powodu Firebase i oferowane przez niego usługi zostały wykorzystane w aplikacji.

## Projekt Bazy Danych

Aplikacja ReceiptCarer wykorzystuję bazę danych NoSQL. W celu uproszczonego zaprezentowania struktury bazy danych, na rysunku 4.20 zamieszczono diagram ERD (charakterystyczny dla relacyjnych baz danych, mimo że omawiana baza jest nierelacyjna). Projekt ten został następnie przetłumaczony na możliwości oferowane przez nierelacyjną bazę danych. Na diagramie ERD encja *Użytkownik* przechowuje podstawowe informacje potrzebne do autentykacji użytkownika w systemie. Encja *Grupa* przechowuje podstawowe informacje na temat grup paragonów, w tym kolor jakim jest reprezentowany w aplikacji. Encja ta jest powiązana z konkretnym użytkownikiem. Encja *Paragon* przechowuje informacje o paragonach. Jest ona związana z konkretnym użytkownikiem i grupą paragonów. Może posiadać wiele znaczników. Daty zawarte w encji *Paragon* są typu varchar(20) (pole tekstowe o stałej długości, może przechowywać od 0 do 20 znaków) ze względy na wykorzystanie w implementacji formatowania tekstu na daty wewnątrz aplikacji. Nie ma więc potrzeby przechowywania ich w bazie danych w innej formie. Encja *Znacznik* przechowuje informację na temat konkretnego znacznika. Jest powiązana z konkretnym użytkownikiem i może posiadać do niej dostęp wiele paragonów. Każdy paragon posiada co najmniej jedno zdjęcie. Zdjęcie jest powiązane z konkretnym paragonem.

Baza danych NoSQL pozwala na tworzenie zagnieżdżonych danych. Oznacza to, że pozwala na to by dany obiekt posiadał kolumnę typu lista i przechowywał wewnątrz siebie wiele elementów danego typu. Pozwala także na mapowanie obiektów bezpośrednio z bazy danych na ten sam typ obiektów w aplikacji. Oznacza to, że definiując obiekt w aplikacji, przesyłany jest bezpośrednio na serwer wraz z jego zawartością i wice versa. Taka technologia pozwala na projektowanie bazy danych w taki sposób by implementowany model danych w aplikacji nie potrzebował być tłumaczony do zapisania w bazie danych. ReceiptCarer wykorzystuje ten mechanizm w następujący sposób. Aplikacja posiada klasy obiektów o nazwach: Paragon (ang: Receipt), Znacznik (ang: Tag), Użytkownik (and: User), Grupa (ang: Group). Obiekt typu Użytkownik posiada identyfikator definiowany podczas rejestracji i na jego podstawie pobierany jest z bazy danych wraz z jego zawartością, którą są trzy listy następujących typów: Paragon, Grupa, Znacznik. Obiekt typu Paragon posiada listę identyfikatorów znaczników, listę linków do zdjęć do niego należących przechowywanych w Firebase Cloud Firestroe oraz identyfikator grupy do której należy. Pobierając użytkownika z bazy danych pobierane są wszystkie najważniejsze elementy potrzebne do funkcjonowania aplikacji. Taka kolej rzeczy bardzo ułatwiła zarządzanie bazą danych. Następnie wewnątrz aplikacji każdy identyfikator służył za klucz obcy rozpoznawany już wewnątrz i w trakcie działania aplikacji. Gdy następuje modyfikacja danych wewnątrz aplikacji, są one synchronizowane z bazą danych Firestore.



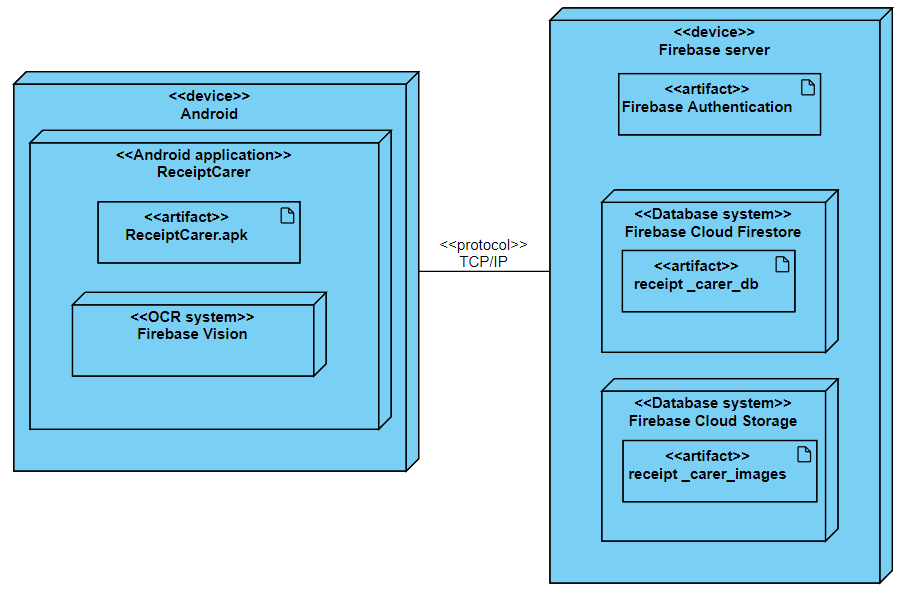
*Rysunek 5.1 Diagram ERD bazy danych*

*[Źródło: opracowanie własne]*

## Architektura rozwiązania

### Architektura fizyczna

Poszczególne moduły aplikacji w formie diagramu rozmieszczenia zostały przedstawione na rysunku 4.10.



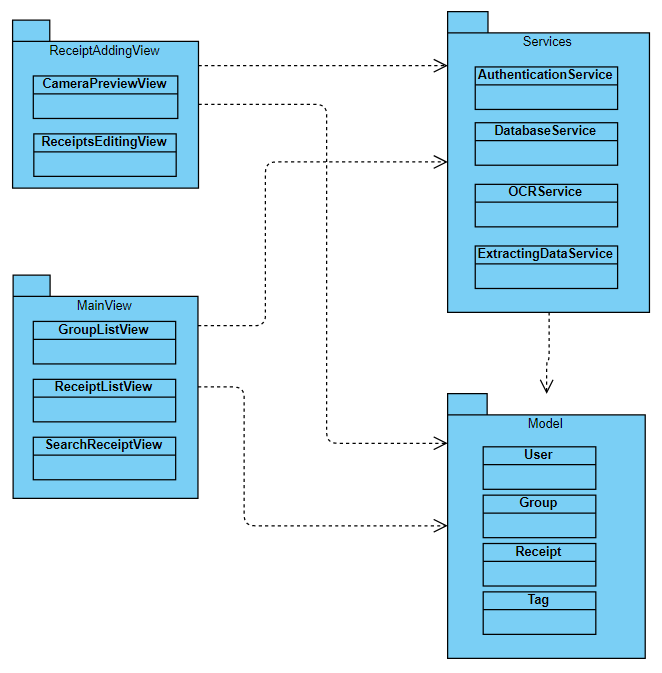
*Rysunek 5.2 Diagram rozmieszczenia*

*[Źródło: opracowanie własne]*

Urządzenie z systemem Android ma zainstalowaną aplikację ReceiptCarer. Aplikacja wykorzystuje system OCR od Firebase do rozpoznawania tekstu ze zdjęć. Jest on wbudowany w aplikacje czyli pozwala to na skanowanie paragonów bez potrzeby łączenia z internetem. Aplikacja łączy się za pomocą protokołu TCP/IP z serwerem Firebase, na którym znajdują się różne usługi, z których korzysta aplikacja. Są to między innymi baza danych Firebase Cloud Firestore oraz Cloud Storage (dla plików graficznych). Aplikacja wykorzystuje połączenie z serwerem także do autentykacji użytkowników w aplikacji.

### Architektura logiczna

Rysunek 4.11 przedstawia diagram pakietów opisujący wzorzec architektoniczny zastosowany w aplikacji.



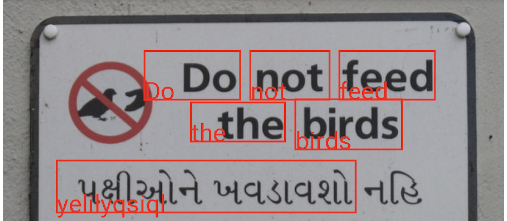
*Rysunek 5.3 Diagram pakietów – wzorzec MVP*

*[Źródło: opracowanie własne]*

Warstwa prezentacji została podzielona na dwa oddzielne pakiety. Pakiet MainView reprezentuje widoki związane z wyświetlaniem list grup i paragonów oraz modułu wyszukiwania. Pakiet ReceiptAddingView natomiast reprezentuje widoki związane z dodawaniem nowych paragonów do aplikacji. Są one zależne od warstwy serwisów oraz reprezentujący warstwę danych pakiet Model. Warstwa serwisu jest także zależna od warstwy danych.

## Ekstrahowanie informacji z paragonów

Aplikacja jako jedno ze swoich założeń ma posiadać możliwość ehstrahowania informacji z paragonów. Rozpoznawanie tekstu przez algorymy OCR od Firebase jest tylko początkiem ekstrahowania danych. Pozwala on na wyodrębnienie tekstu w formie tak zwanych bloków tak jak przedstawia rysunek 5.4. Każdy blok zawiera informacje o tekście który się znajduje w jego granicach oraz o punktach (o współrzędnych x, y względem zdjęcia którego dotyczą) oznaczajacych wierzchołki tych prostokątów. Taki podział pozwala na wykorzystanie przy rozpoznawaniu specyficznych elementów paragonów. Punkty wierzchołków w aplikacji zostały wykorzystane w przypadku rozpoznwania kwoty sumarcznej widniejącej na paragonie.



*Rysunek 5.4 Wynik algorytmu OCR od Firebase w formie bloczków tekstu [Źródło:* [19]*]*

W przypadku ekstrahowania informacji z paragonów samo skanowanie tekstu nie wystarcza. Nastąpiła więc potrzeba wymyślania heurystyk, które pozwalałyby na wyodrębnienie poszczególnych informacji w jak najskuteczniejszy sposób ale zarazem by nie trwało to zbyt długo. W pracy informacjami, które są ekstrahowane z paragonów są: nazwa sklepu, data wydania paragonu, oraz cena na paragonie. Każdy z tych elementów wymagał idendywidualnego podejścia do ekstrahowania, które będą omówione w dalszej części pracy.

Dla zwiększenia szansy znalezienia porównywanych w heurystyce fraz związanych z poszczególnymi danymi z paragonu każda wartość tekstowa jest zamienia na małe litery. Dzięki temu usuwany jest problem związanyz tym, że formalnie w wykorzystanym języku rogramowania jakim jest Java, dwa wyrażenia są traktowane jako różne teksty pomimo ich oczywistego podobieństwa. Przykładem może być słowo „SUMA” oraz „Suma”, które na różnych paragonach mogą się znajdować. Ważnym jest więc sposowanie (jak to się nazywa ☹ formatowanie? )

Dla każdego bloku tekstu wykonywane są poszczególne heurystyki:

**Nazwa sklepu**

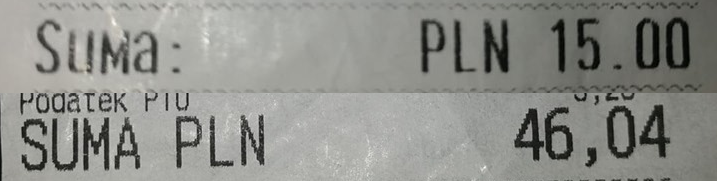
Heurystyka rozpoznwania nazwy sklepu sprawdza czy dany blok zawiera jedną z zapisanych w formie listy nazw sklepów. Na potrzeby implementacji i testów w aplikacji będzie znajdować się siedem nazw sklepów takich jak: „Lidl”, „Żabka”, „Biedronka” itd. Trzeba wziąć ejdnak pod uwagę, że im dłuższa lista skelpów tym więcej przy każdym bloku musi wykonać obliczeń telefon zwiększając tym długośc trwania ekstrahowania danych co jest bardzo nie pożądane w przypadku omawianej aplikacji. Dodatkowo obserwując paragony z listy sklepów można zauważyć, że nie ma potrzeby badania wszystkich bloków pod kątem nazwy sklepu jeśli w pierwszych kulkunastu blokach nie została ona znaleziona. Nazwa sklepu znajduję się przeważnie na początku dowodu zakupu. Z tego powodu przy 20 bloku i nie powodzdeniu w znalezieniu nazwy sklepu heurystyka porzuca jego poszukiwania. Liczba ta została oszacowana na podstawie 15 paragonów z różnych sklepów.

**Data wydania paragonu**

Data na paragonie może być przedstawiwana w wielu formatach dd-MM-yyyy,   
yyyy-dd-MM, yyyy/dd/MM itd. (gdzie „d” oznacza dzień, „M” oznacza miesiąc, „y” oznacza rok). Heurystyka bada czy dany blok zawiera tekst, który pasuje do którejś z podanych form zapisywania dat.

**Cena sumaryczna widniejąca na paragonie**

Największym wyzwaniem okazało się znalezienie sumy sumarycznej na paragonie. Jest tak ze względu na fakt, że różne pragony z różnych sklepów inaczej podchodzą do sprawy zapisu sumy. Oczywistym jest, że rozpoznawanie samego tekstu czy jest liczbą dziesiętną nie jest wystarczające gdyż na paragonie znaduję się wiele tkaich liczb. Trzeba znaleźć więc punkt zaczeniepienia, który posłuży za znalezienie kwoty symarycznej. Dla przykładu na rysnku 5.5 przedstawiono dwie możliwe formy zapisu kwoty sumarycznej na paraognie.



*Rysunek 5.5 Przykładowe formy zapisu kwoty sumarycznej na paragonie [Źródło: opracowanie własne]*

Wspólnym punktem odniesienia w obu przypadkach przedstaiwonych na rysunku jak i każdym paragonie, z którym autor pracy miał styczność jest słowo „PLN”. Problemem okazuję się fakt, że słowa te są w roznych odległościach od wyszukiwanej kwoty sumarycznej co wpływa znacząco to w jaki algorytm tworzy bloki. W zależności od przypadku suma sumaryczna może znajdować się w tym samym bloku co słowo klucz „PLN” lub w sąsiedztwie nawet kilku bloków za lub przed blokiem zawierającym punkt zaczeniepie. Z tego powodu heurystyka w pierwszej kolejności, jeśli zajdize blk który zawiera frazę klucz, przeszukuje ten blok czy nie zawiera liczby dziesiętnej. Kolejnym krokiem w razie nieznalezienia jest zebranie bloków znajdujących się w najbliższym sąsiedztwie bloku frazy klucz. W tym przypadku nie wystarczy tylko przepuszkać wszystkie sąsiednie bloki w poszukiwaniu liczb dzesietynch ponieważ w okolicach frazy może być kilka tkaich. Nie sposó wyłonić tą konkretną liczbe. Pomocnym staje się dodatkowa informacja niosona przez bloki tekstu jaką są punkty wierzchołków prostokątów wyznaczonych przez algorytm rozpoznwania tekstu. Można na ich podstawie sprawdzić czy sąsiednie bloki znajduję się na tej samej linii co blok, zawierający frazę „PLN”. Heurystyka wymylsona na potrzeby apliakcji sprawdza więc czy jeden z wybranych wierzchołków w sposunku do innych bloków jest mniej więcej na tej samej linii. Sprawdzana jest mianowicie wpsółrzędna „y” punktów znajdujących się w lewym górnym rogu rpostokątów bloków. Ze względu na fakt ze bloki nie muszą znajdowąc się dokładnie na jedne linii sprawdzane jest czy znajduję się w położeniu zakładając ze mogą być odsunięte o kilka pixeli od siebie. Wartością która wykazała największą skuteczność bez popełniania błędów było odchylenie 50-sięciu pikseli względem współrzędnej y. Taka wartość pozwala na drobne przekrzywienia zdjęć paragonów.

W przypadku badania tego typu, czyli czy bloki znajduja się na jednej linii, ważnym jest by zdjęcie które jest wykonywane było w miarę prostę (paragon nie był obrócony). Na tym etapie możnaby zaimplementować heurystyki, pozwalające na naprawianie zaburzeń związanych z ułożeniem paragonu. Przekształcały by orginalne zdjęcie paragonu dla lepszej jego czytelności przez heurystyki. Algorytmy wykrozsytywane do ropoznawania zdjęcia doskonale sobie radzą z teog typu zaburzeniami, lecz w przypadku ekstrahowania informacji z paragonów autor pracy zdecydował uprościć implemnetwane heurystyki by nie zwiększać ich złożoności działania. Pozwoliło to na możliwość wprowadzenia większej bazy sklepów lub formatów dat. Aplikacja z ząłożenia ma być szybka i zakłada się, że zdjęcia wykonywane przez uzytkonwika nie będą zncząco zniekształcone.

**Dodatkowa heurystyka**

Istnieje możliwość, że wyszukiwane słowa klucz takie jak porównywane nazwy sklepów z tekstem znajdującym się w blokach mogą być nie rozpoznane przez algorytmy OCR w sposó zgodny z rzeczywistością. Przykąłdem może być nazwa sklepu „Lidl” gdzie częstym przypadkiem rozpoznwania przez Firestore Vision było nie słowo „Lidl” lecz słowo „Lid1”, z zamienioną jedną literką na cyfrę „1”, ze względu na podobieństwie liter. Jest to częsty przypadek i całkowicie zrozumiały. Nastapiła więc konieczność ochrony przez tego typu błędami. Wykrzystywane do tego miara odmienności napisów (skończonych ciągów znaków) o nazwie odległość Levenshteina [20]. Odległością między dwoma napisanmy nazywamy w wprzypadku algorytmu najmnijeszą liczbe działań prostych, przeprowadzających jeden napis w drugi. Do diząłań prostych zaliczamy:

- wstawienie nowego znaku do napisu

- usunięcie znaku z napisu,

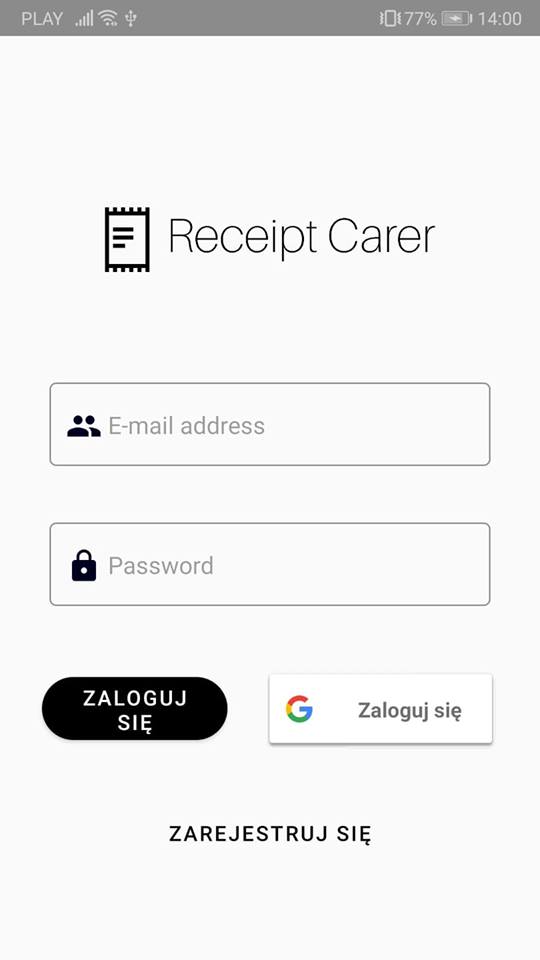
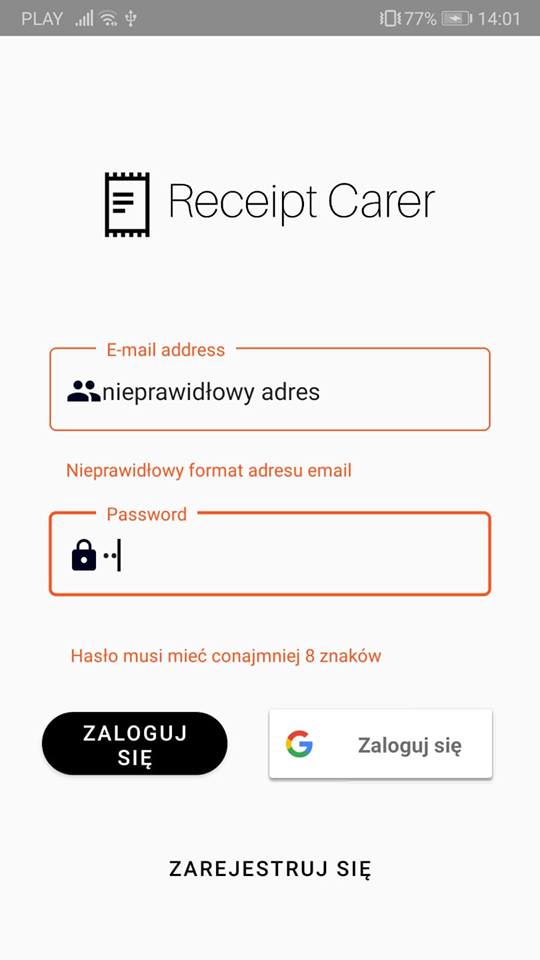
- zamianę znaku w napisie na inny znak.

W heurystykach miara wykorzystywana jest do wyzaczania jak dużo (w procentach) potrzeba zminieć dłuższy ciąg znaków by zaminić go w krótszy ciąg. Dla przykładu słowo „Lid1” aby zamienić na słowo „Lidl” wymagane jest zmienienie 25% słowa (jedna z czterech liter). W przypadku słów „Lidl” oraz „Lib1” byłoby to już 50% (dwie litery na cztery). W Heurystykach dopuszczalny procent zmian to 25% ze względu na fakt, że najkrótsza nazwa sklepu jest czteroliterowa i dopuszcalne jest zmienieni jej jednej litery.

# Implementacja

W niniejszym rozdziale została omówiona implementacja poszczególnych funkcjonalności aplikacji.

## Logowanie

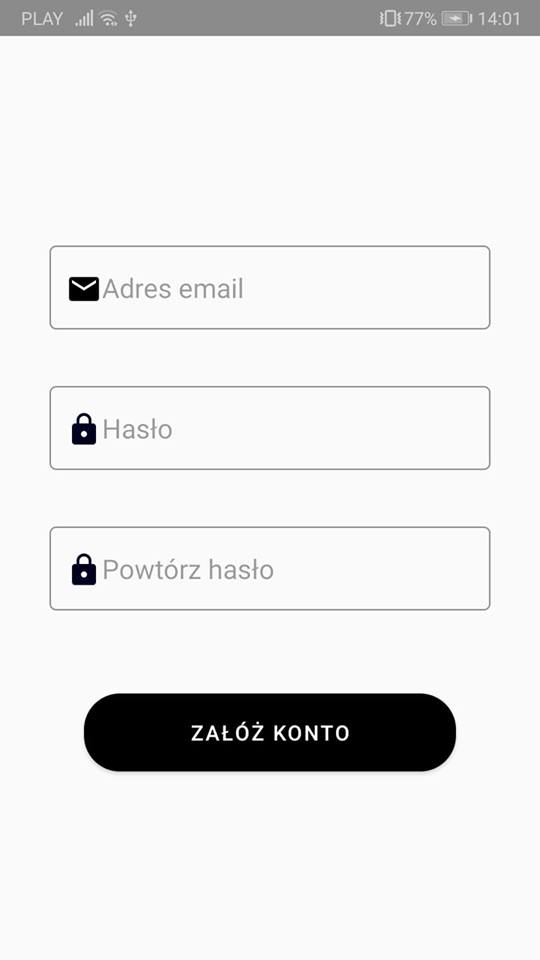
*Rysunek 6.1 Ekran logowania – widok główny (po lewo), informacja o błędach (po prawo)   
[Źródło: opracowanie własne]*

Włączając aplikacje pierwszym ekranem jest ekran logowania (Rysunek 6.1) (pod warunkiem, że do aplikacji wcześniej się nie zalogowano). Jeżeli użytkownik zarejestrował się wcześniej do aplikacji to może wprowadzić adres email i hasło, a następnie kliknąć przycisk *Zaloguj się*. Po pozytywnej weryfikacji adresu email oraz hasła użytkownik zostanie przeniesiony do ekranu głównego, listy paragonów (rysunek 6.3), opisanego szczegółowo w dalszej części pracy. Nieprawidłowy email lub zbyt krótkie hasło powodują wyświetlenie informacji o błędzie (Rysunek 6.1).

Po prawej stornie od przycisku logowania się znajduje przycisk *Zaloguj* *się* od Google, który pozwala na możliwość logowania i rejestrowania się za pośrednictwem konta Google.

Na samym dole użytkownik ma możliwość zarejestrowania się poprzez wciśnięcie przycisku Zarejestruj się. Przenosi on użytkownika do ekranu rejestracji (Rysunek 6.2) się opisanego szczegółowo w dalszej części pracy.

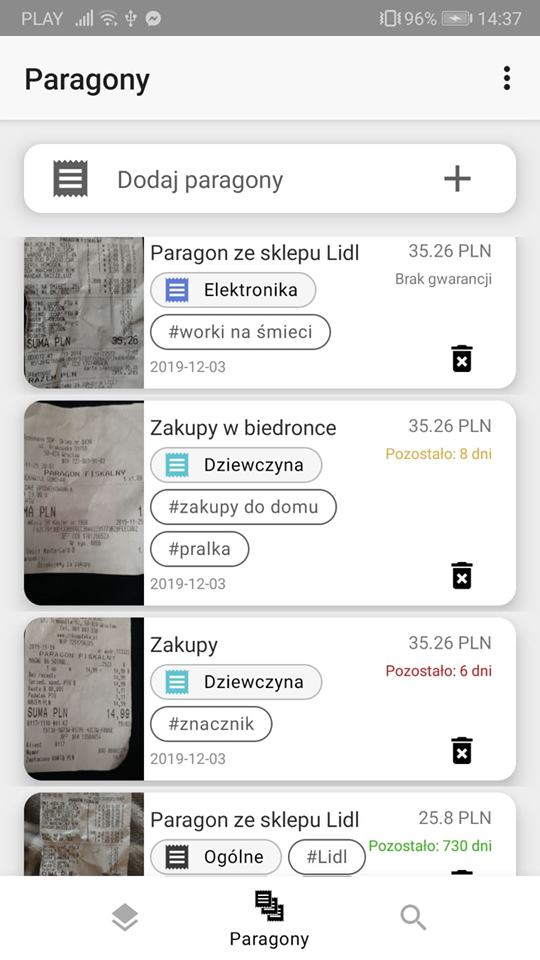
## Rejestracja

*Rysunek 6.2 Ekran rejestracji – widok główny (po lewo), informacja o błędach (po prawo)   
[Źródło: opracowanie własne]*

Naciskając przycisk *Zarejestruj się* użytkownik zostaje przeniesiony do ekranu rejestracji. Może on wprowadzić dane potrzebne do późniejszej jego weryfikacji w systemie. Nieprawidłowe wprowadzenie danych skutkuje wyświetleniem się informacji o błędzie. Dla bezpieczeństwa, że użytkownik nie pomylił się przy wprowadzaniu hasła, hasło należy wpisać dwukrotnie w celu pełnej rejestracji użytkownika. W przypadku prawidłowych danych zostaje utworzony użytkownik w systemie, a następnie zapisany w bazie danych. Do wykonania tych czynności konieczny jest dostęp do Internetu. Po samej rejestracji użytkownik przenoszony jest do ekranu głównego – listy paragonów (rysunek 6.3).

## Lista paragonów



*Rysunek 6.3 Ekran listy paragonów [Źródło: opracowanie własne]*

Po zalogowaniu do aplikacji ekranem, który widzi użytkownik jest ekran listy paragonów które dodał do aplikacji. Znajdujące się elementy na ekranie to lista paragonów, przycisk dodawnai apragonów, Toolbar z podpisem na jakim ekranies ei znajdujemy z guzikiem pozwalającym na wyświetelnie opcji dodatkowych. W pierwszej wersji aplikacji z tamtego poziomu menu można się wylogować. Każdy element listy paragonów pokazuje następujące informacje:

- nazwa paragonu,

- miniaturka paragonu,

- data wydania paragonu

- grupa do której należy paragon,

- kwota pieniężna znajdująca się na paragonie,

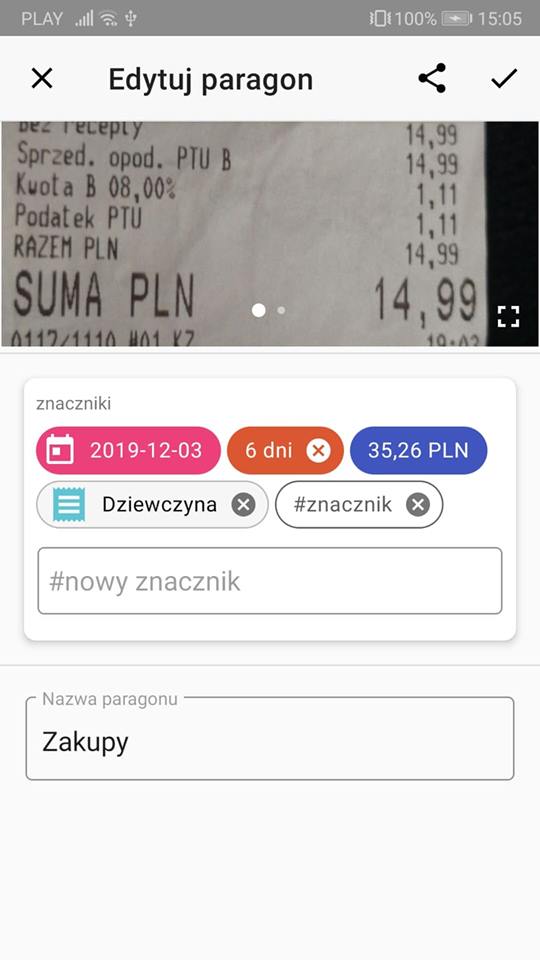
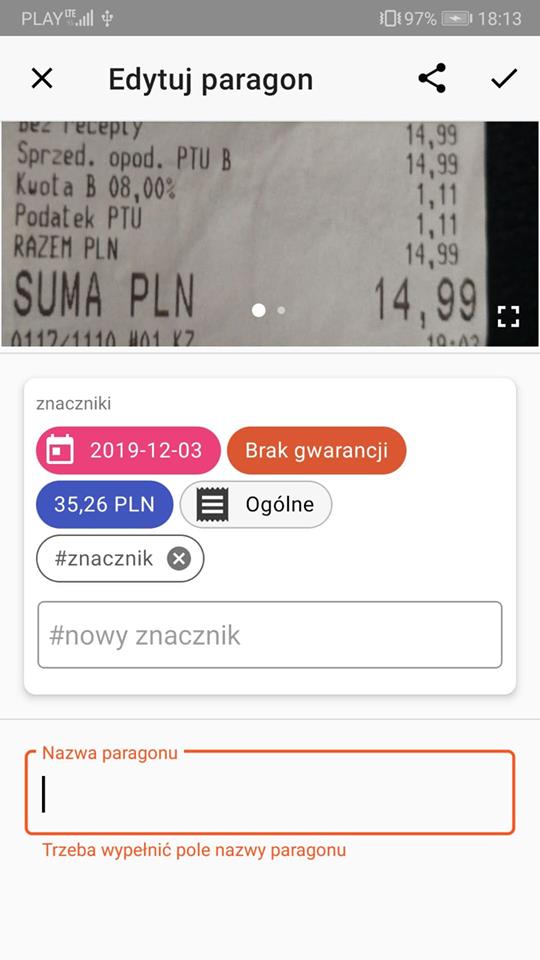
- liczba dni jakie pozostały do końca gwarancji, lub informacja o braku gwarancji,

- znaczniki przypisane do paragonu.

W zależności od liczby dni pozostałych do końca trwania gwarancji informacja o niej jest podkreślana w inny sposób. Gdy liczba dni jest mniejsza niż 7 dni to wyświetla tę informację na czerwono, gdy pozostało mniej niż 30 dni, a więcej niż 7 to na pomarańczowo, a w przypadku liczby większej niż 40 dni na zielono. Informacja ta ma pomóc użytkownikowi w obserwowaniu, które paragony dotyczą produktów, które doc-odzą do terminów kończących teminy wanzosci gwarancji.

Uzytkownik może kliknąć w element listy i zostanie wtedy przeniesiony do szczeółów danego paragonu wraz z możliwościąjjego edytowania i eksportowania. Użytkownik dostaje także możliwość usunięci paragonu klikając w ikone kosza znajdującą się w prawym dolnym rogu każdego elementu lsity apragonów.

## Paragon – widok szczegółowy i edycja

*Rysunek 6.4 Ekran edytowania paragonu – widok edycji (po lewej),   
informacja o błędzie nazwy (po prawej)  
 [Źródło: opracowanie własne]*

Po kliknięciu w dany element na liście paragonów użytkownik zostaje przeniesiony do ekranu edytowania paragonu (Rysunek 6.4). Na tym ekranie znajdują się te same informacje co na liście paragonów przedstawione w poprzednim podrozdziale z tą różnicą że poszczególne elementy mogą być edytowane. Na górze ekranu znajduje się możliwość podglądu zdjęć paragonów. Akcje edytowania można anulować lub zaakceptować za pomocą przycisków akceptacji (prawy górny róg). Po akceptacji sprawdzana jest poprawność danych i w razie błędów wyświetlany jest odpowiedni komunikat. Po zatwierdzeniu edycji dane są synchronizowane z bazą danych.

W aplikacji rozróżnialne są tak zwane specjalne typy znaczników o innych charakterystykach w wyglądzie i funkcjonalności. Do specjalnych znaczników należą:

- znacznik specjalny wyboru daty (kolor różowy z ikoną kalendarza po lewej stronie),

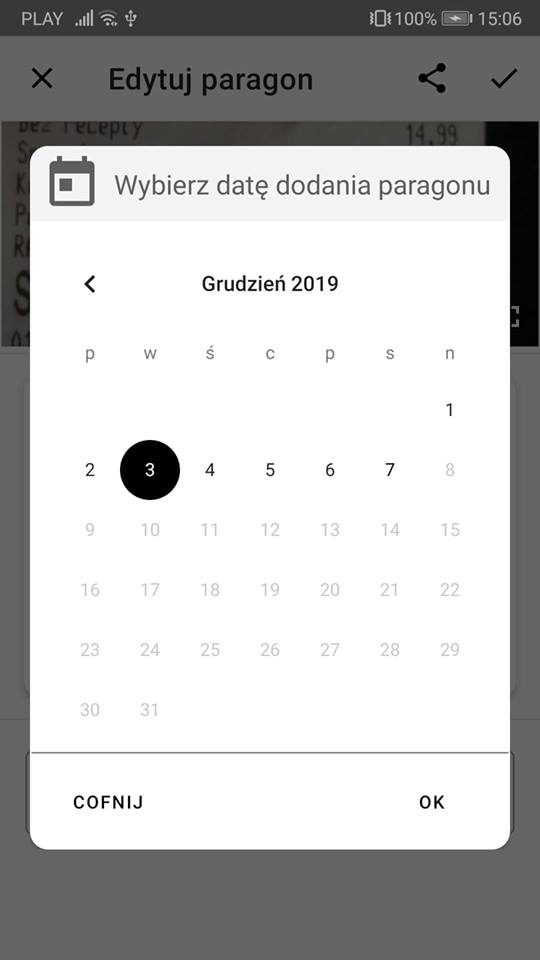
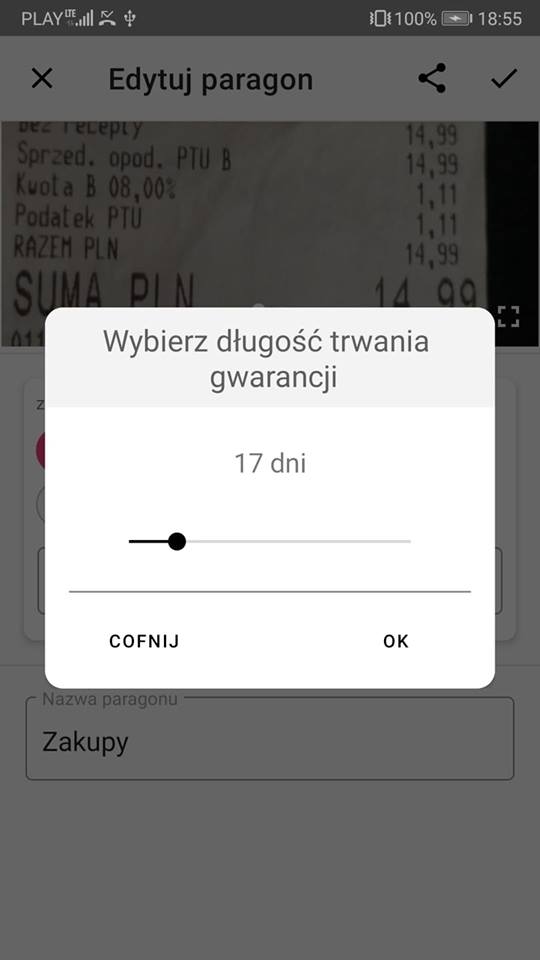
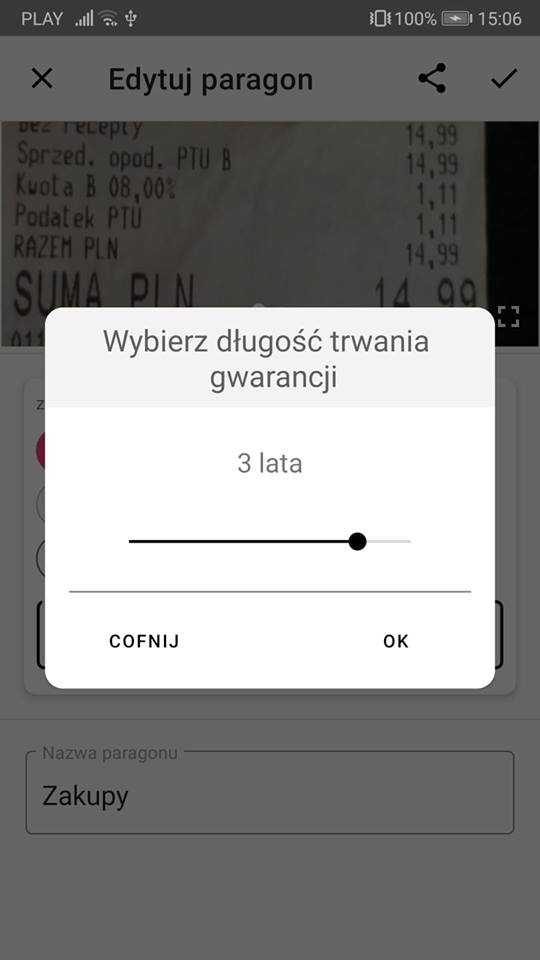
- znacznik specjalny wyboru długości trwania gwarancji (kolor pomarańczowy),

- znacznik specjalny wyboru kwoty pieniężnej (kolor niebieski,

- znacznik specjalny wyboru grupy (kolor szary z ikoną paragonu po lewej stronie).

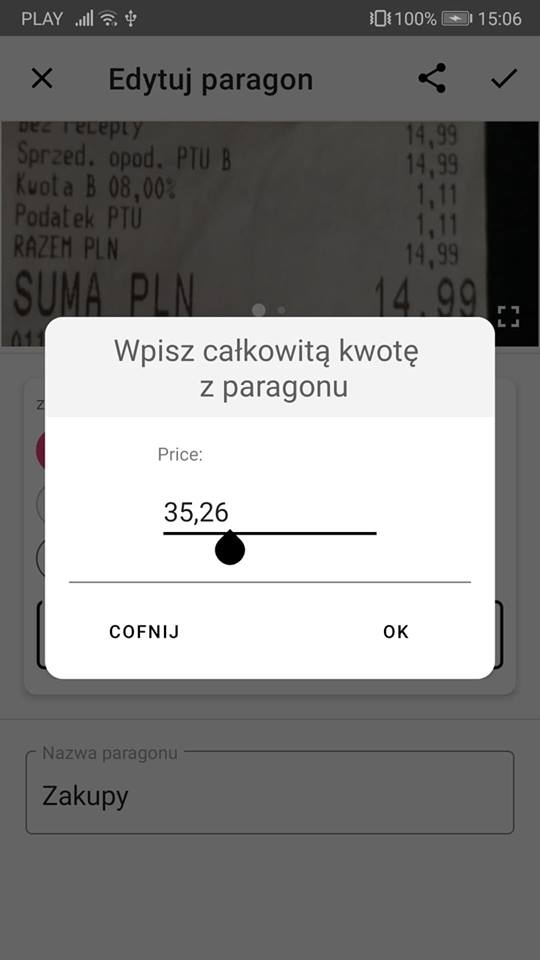
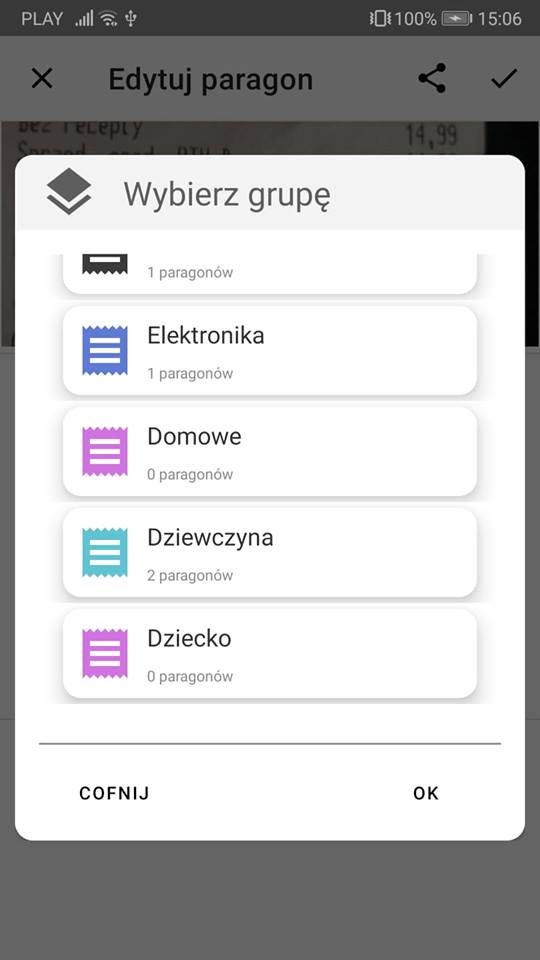
W każdy znacznik specjalny można kliknąć i wykonać związane z nimi możliwe akcje. System taki został wybrany ze względu na uproszczenie widoku edycji paragonu. Po kliknięciu w dany znacznik pojawia się dodatkowy ekran z odpowiednim tytułem, pozwalający na wykonanie, zaakceptowanie lub odrzucenie danej akcji związanych ze znacznikami.

Znacznik wyboru daty ukazuje ekran kalendarza, który pozwala na wybranie dowolnej daty wcześniejszej od dnia, w którym następuje edycja (Rysunek 6.5). Znacznik wyboru długości trwania gwarancji pozwala na wybór za pomocą slajdera długości trwania gwarancji (Rysunek 6.5). W zależności od pozycji slajdera gwarancja może być wyrażona w danich, tygondiach, miesiącach lub latach. Rysunek 6.5 prezentuje tę kolej rzeczy.

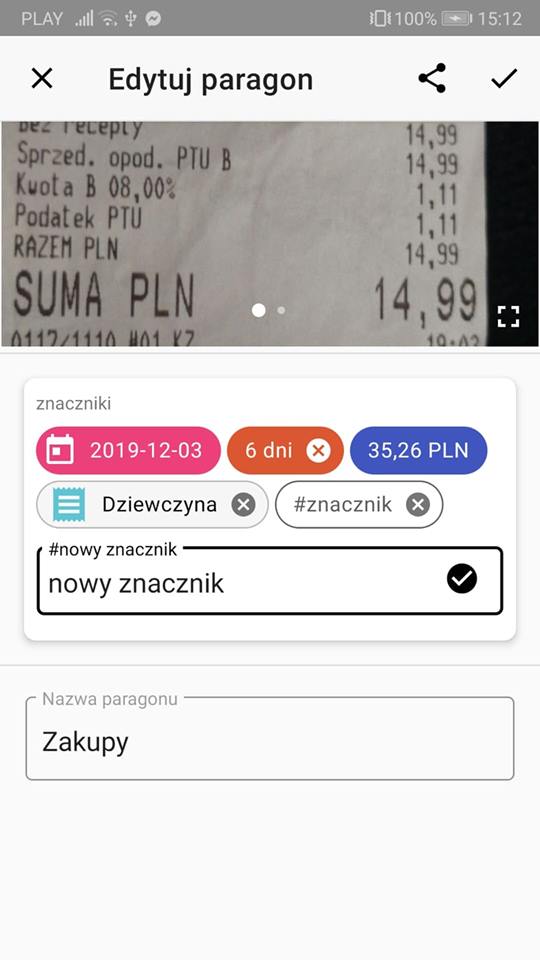
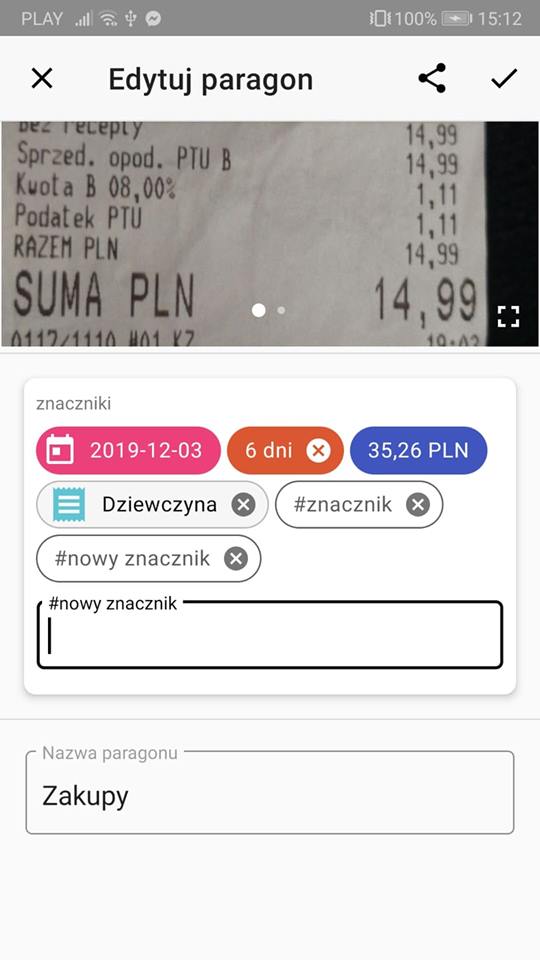
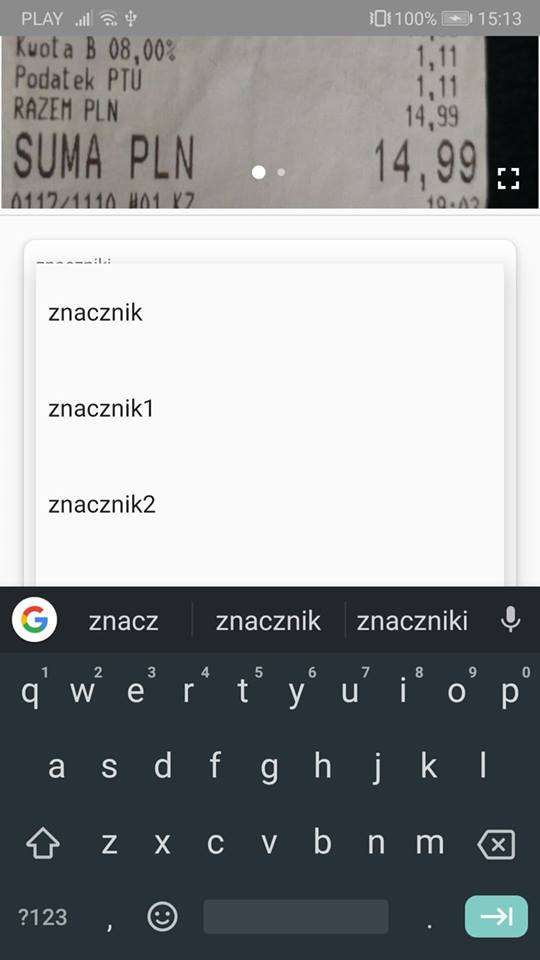
*Rysunek 6.5 Ekrany specjalne: wybór daty dodania paragonu (po lewej),  
wybór długości trwania gwarancji (środek i po prawej) [Źródło opracowanie własne]*

W przypadku znacznika wyboru całkowitej kwoty z paragonu, ekran zawiera jedynie pole kwoty (Rysunek 6.6), które można edytować. Znacznik dotyczący wyboru grupy pozwala na wybranie grupy i przypisanie jej do paragonu (Rysunek 6.6). W przypadku nie wybrania żadnej grupy zostaje on przypisany do grupy ogólnej o nazwie „Ogólne”.

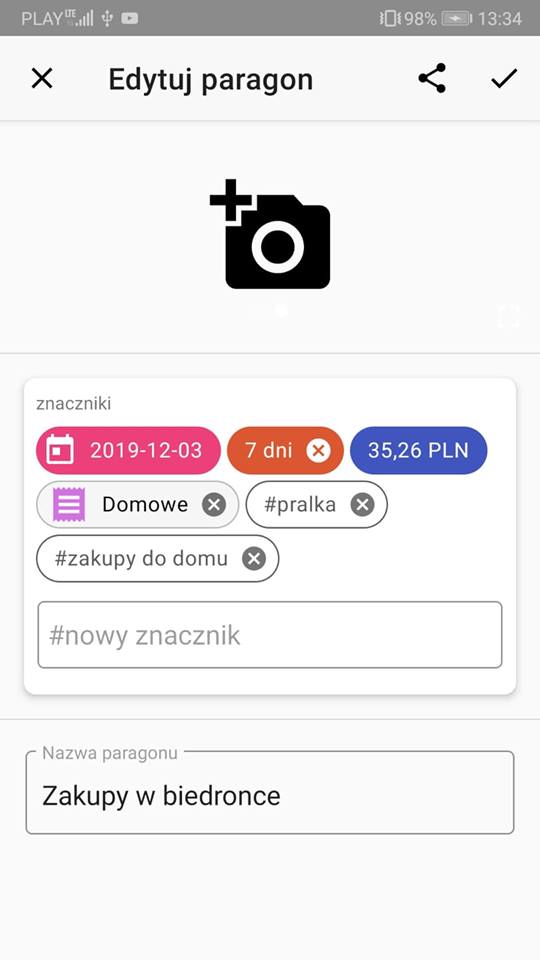
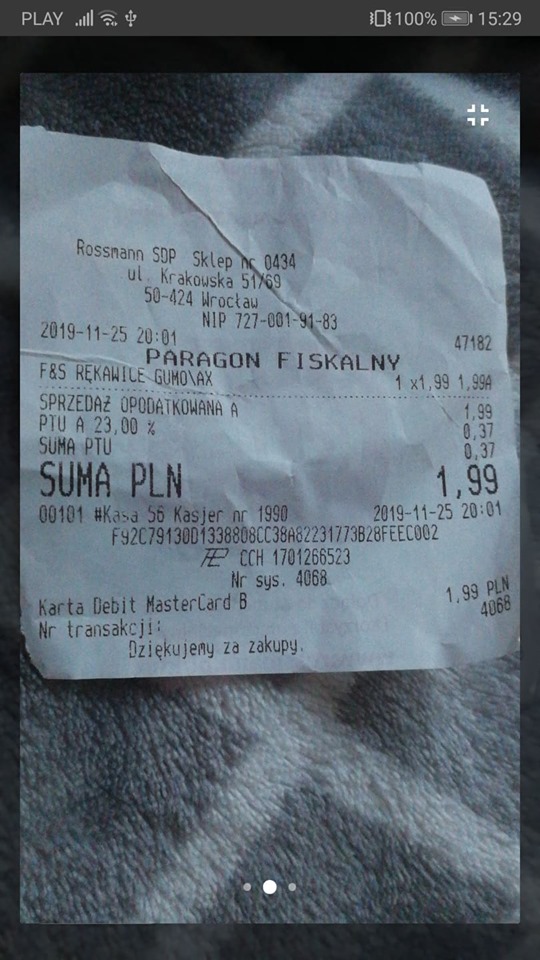
*Rysunek 6.6 Ekrany specjalne: wybór całkowitej kwoty z paragonu (po lewej),  
wybór grupy do której należy paragon (po prawej) [Źródło: opracowanie własne]*

Kolejną funkcjonalnością związaną z paragonami jest zarzadzanie tak zwanymi zwykłymi znacznikami związanymi z danym paragonem. Znaczniki służą do gromadzenia bardziej szczegółowych informacji na temat paragonów i są wykorzystywane przy ich przeszukiwaniu.

*Rysunek 6.7 Dodawanie nowego znacznika – wpisanie nazwy znacznika (krok 0, po lewej),  
zaakceptowanie znacznika do paragonu (krok 1, po prawej),  
wybór znacznika z lity podpowiadającej znaczniki [Źródło: opracowanie własne]*

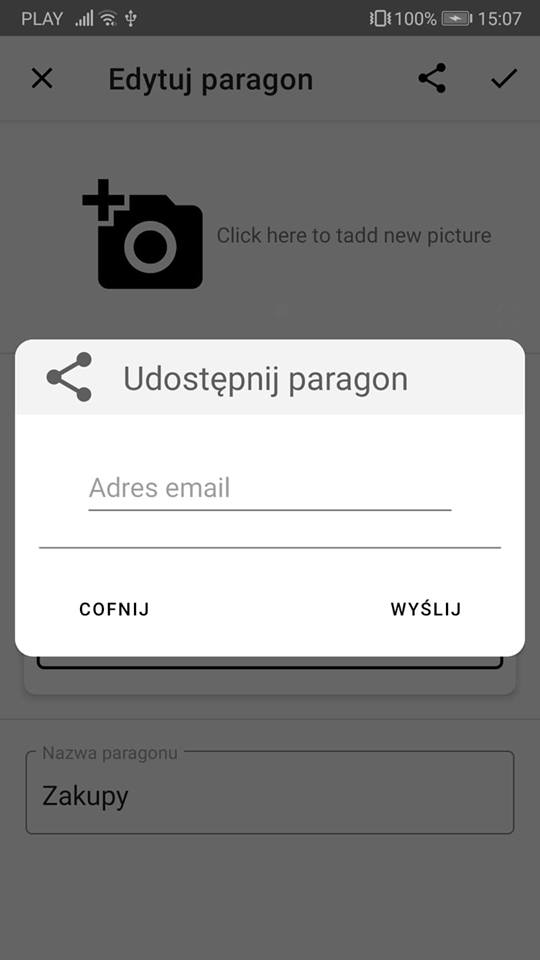
Rysunek 6.7 przedstawia możliwość dodania znacznika do paragonu. Po wpisaniu w edytowalne pole tekstowe ciągu znaków pojawia się po prawej stronie ikona akceptacji pozwalająca na dodanie ich do paragonu. Taki znacznik można usunąć z paragonu klikając w ikonę usunięcia znacznika znajdującego się po prawej stronie w nim. Istnieje także możliwość by użytkownik wybrał znacznik z listy już istniejacych znaczników. Ukazane jest to na rysunku 6.8. Przy wpisywaniu frazy do pola nowego znaicznika pojawia się lsita znaczników znadujących się w całej aplikacji które zawierają wpisywaną frazę. Pozwala to na lepsze porządknowanie znaczników.

*Rysunek 6.8 Możliwość dodania nowych zdjęć do paragonu (po lewo),  
podgląd zdjęć paragonów (po prawo) [Źródło: opracowanie własne]*

Kolejną możliwościami związanymi z edycją paragonu jest możliwość dodania więcej zdjęć do paragonu oraz przeglądanie zdjęć paragonów. Dostęp do dodawnia nowych zdęc uzyskuej się poprzez przesunięcie na samo prawo podglądu zdjeć paraogonu. Klikając w ikone paratu z plusem nastepuje przekierowanie do ekranu dodwania zdjeć. Sposób dodawania zdjęć został opisany szczegółowo w dalszej części pracy przy okazji dodawania paragonów do aplikacji. Klikając w dowolną miniarurkę na przeglądzie zdjęć na ekranie edytowania paraognu wyświetlany jest podgląd zdjęc pozwalający na usuwanie paragonów (musi zostać conajmnije jedno zdjęcie paragonu).

## Eksportowanie paragonów

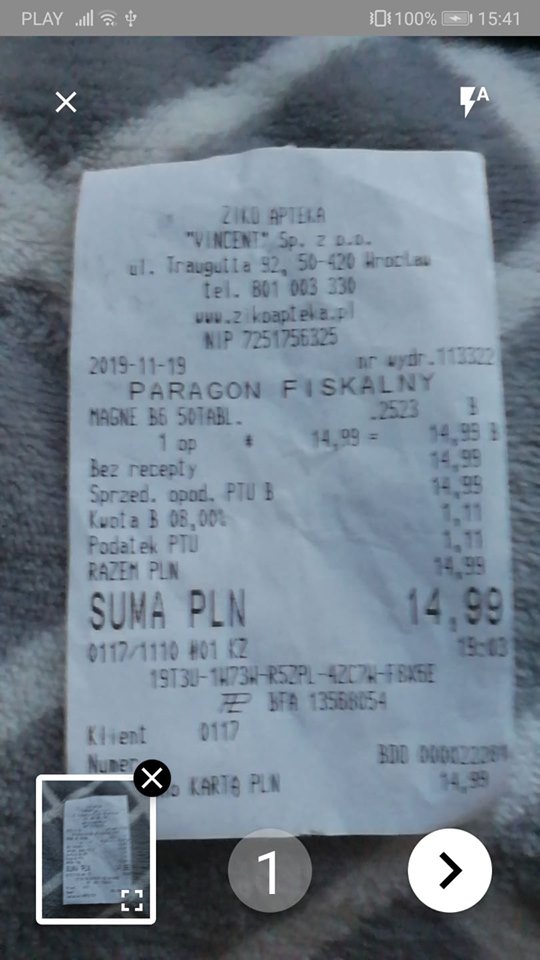


*Rysunek 6.9 Eksport paragonu [Źródło: opracowanie własne]*

Z ekranu edycji paragonu można eksportować paragon na wybrany email. Służy do tego przycisk znajdujący się po lewej od przycisku akceptacji zmian paragonu. Na rysunku 6.10 przedstwiono ekran eksportu. Po wpisaniu i zweryfikowaniu poprawności w pisowni adresu email nastepuje odesłanie do zewnętrznej apliakcji zajmującej się wysyłką emaili na telefonie użytkownika.

## Dodawanie nowych paragonów

Z ekranu listy paragonów klikjać w rpzycisk dodawnia apaaragonów użytkonwik przechowidz do ekranu wykonywania zdjęć. Ekran ten został przedstawiony na rysnku 6.11. Ten sam ekran wraz z działaniem przedstawiany jest przy dodawnaiu kolejnych zdjęć do apragonów podczas ich edycji.

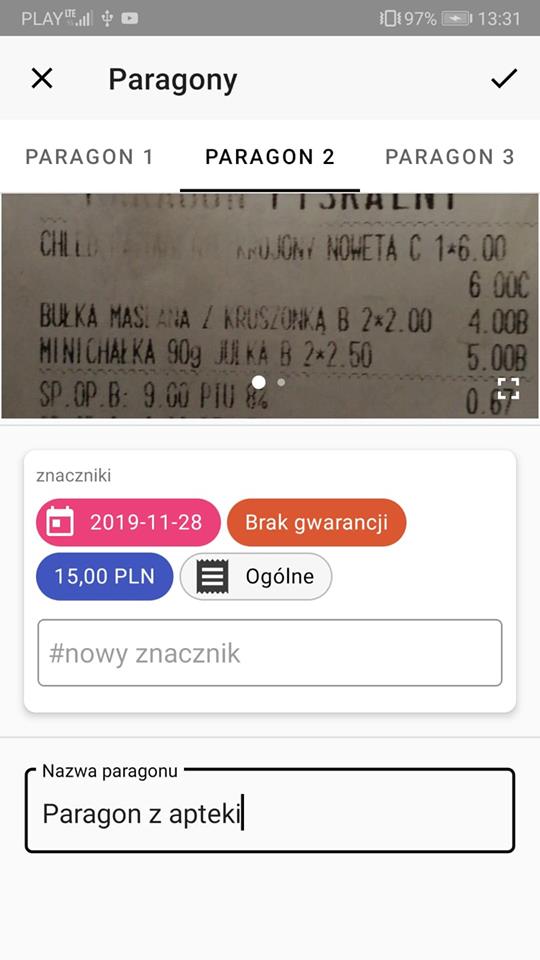
*Rysunek 6.10 ekran dodawania zdjęć – brak wykonanych zdjęć (po lewej),  
wykonano jedno zdjęcie, wykonano 2 zdjęcia (środek oraz po prawo) [Źródło: opracowanie własne]*

Przez cały czas działania tego ekranu widoczny est podglądu apartu tylniego telefnu, Nie istnieje możliwość wykonywania zdjęcia araptem rpzednim. W lewym górnym rogu znajduje się przycisk anulowania akcji. Przycisniecie go skkutnuje przeniesiem do poprzedniego ekranu użytkownaego przez użytkownika. W [rawym gornym rogu znajduje się możliwość wybrania trybu dizała świata flesz pod warunkiem, że urządzenie na którym jest zaintaloswana apliakcja posiad ataką możliwość. Wprzeciwnym wypadku guzik ten jest niewidoczny. Aplikacja obsługuje 3 tryby działania swiatła flesz: włączone, wyłączone lub automatyczne. Użytkownik klikjac w ikone światła przełącza je na kolejny stan.

Na dole na śordku znajduje się przycisk wykonywania zdjęcia. Po jego wciśnięciu w lewym dolnym rogu pojaiwa się podgląd ostatnio wykonanego zdjęcia wraz z guzikiem krzyżyka pozwalającego na usuwanie ostatnio dodanego zdjęcia. Po wciśnięciu podglądu zdjęcia pojawia się ekran podglądu zdjęć taki sam jaki przedstawony został na rsynku 6.9. Jeśli zostało wykonane conajmnije jedno zdjęcie w prawym dolnym rogu pojawia się przycisk umożliwiający akceptacje zdjęć i przełączenie do kolejnego ekranu. W przypadku gdy akcja rozpoczynajaca ekran wykonywania zdjęć to dodawnaie paragonów, użytkownik zostaje przeniesiony do ekranu przedstawionego na rysunku 6.11.

Po zatwierdzeniu zdjęć paragonów nastepuje następuje skanowanie paragonów przy wykorzystaniu systemu OCR oferowanego przez Firebase wbudowanego w aplikację. Po zeskanowaniu przy wykorzystaniu opisanych w fazie projektowania heurystyk wyodrębniane są następujące dane: data dodania paragonu, nazwa sklepu z którego pochodzi paragon oraz kwota pieniężna znajdująca się na paragonie.

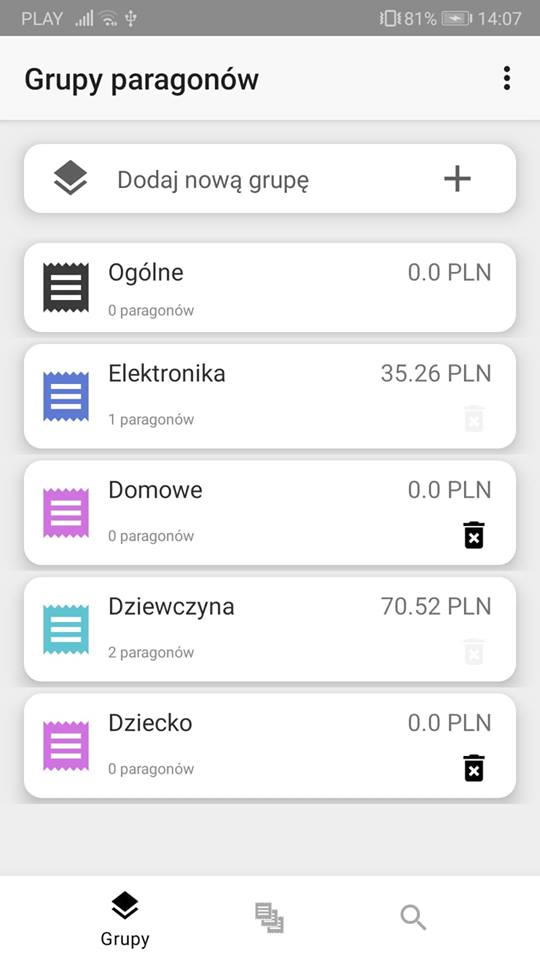
Rysunek 6.11 przedstawia ekran edytowania paragonów. Można przełączać się pomiędzy paragonami za pomocą tab.



*Rysunek 6.11 Ekran edycji dodawanych paragonów [Źródło: opracowanie własne]*

Uzytkownik edytuje oraz zatwierdza edycję nowych paragonów w taki sam sposób w jaki został opisany przy okazji edycji paragonów. Po akceptacji i weryfikacji poprawności wprowadznych danych nowe apragony są synchornizowane z bazą danych oraz użytkownik przenoszony jest do ekranu głównego – listy wszystkich paragonów.

## Lista grup paragonów



*Rysunek 6.12 Ekran z listą grup paragonów [Źródło: opracowanie własne]*

Na rysunku 6.12 przedstawiony jest ekran zawierający listy grup paragonów oraz przycisk dodawania nowej grupy. Przycisk ten przenosi użytkownika do ekranu dodawania nowej grupy przedstaiwnej w dalszej części pray. Każdy element listy zawiera następujące informacje na temat grupy:

- nazwę grupy,

- liczba paragonów znajdujące się w grupie,

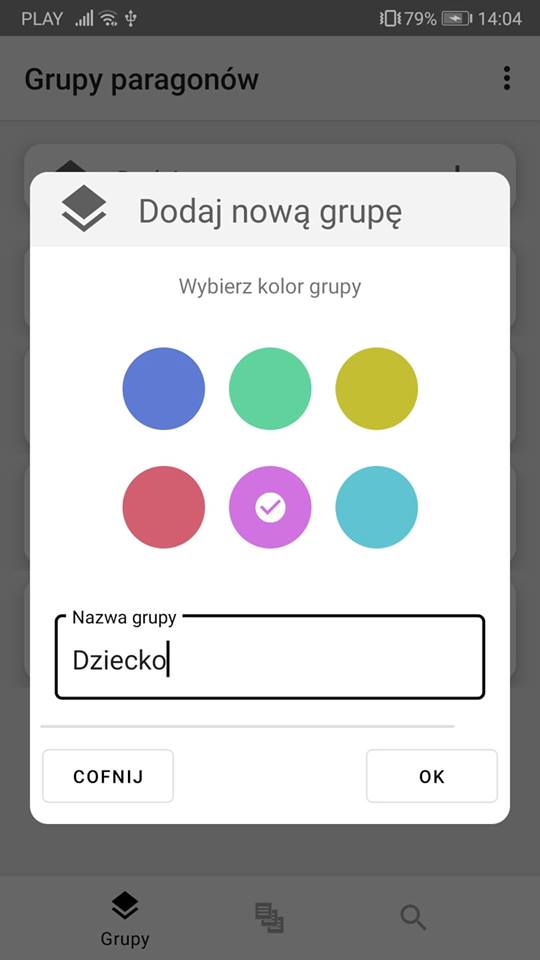
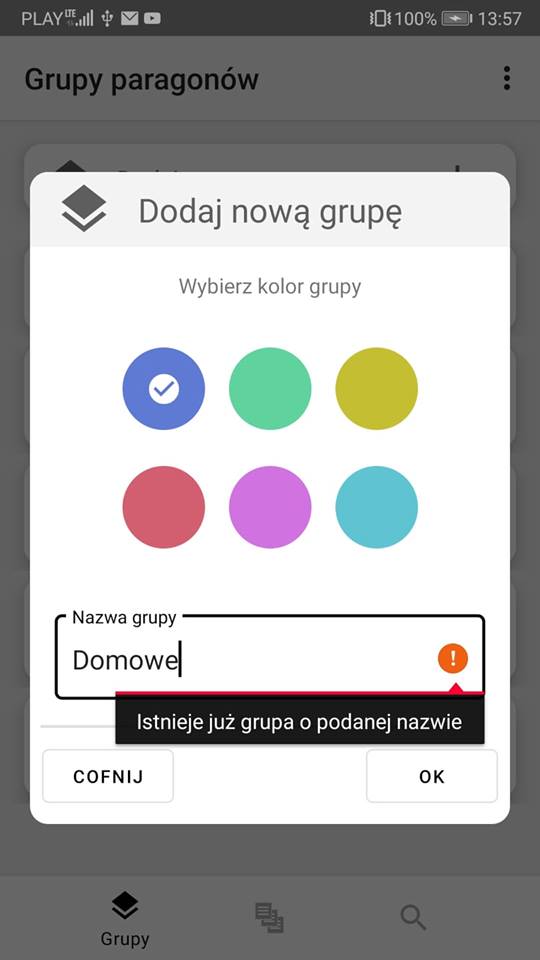
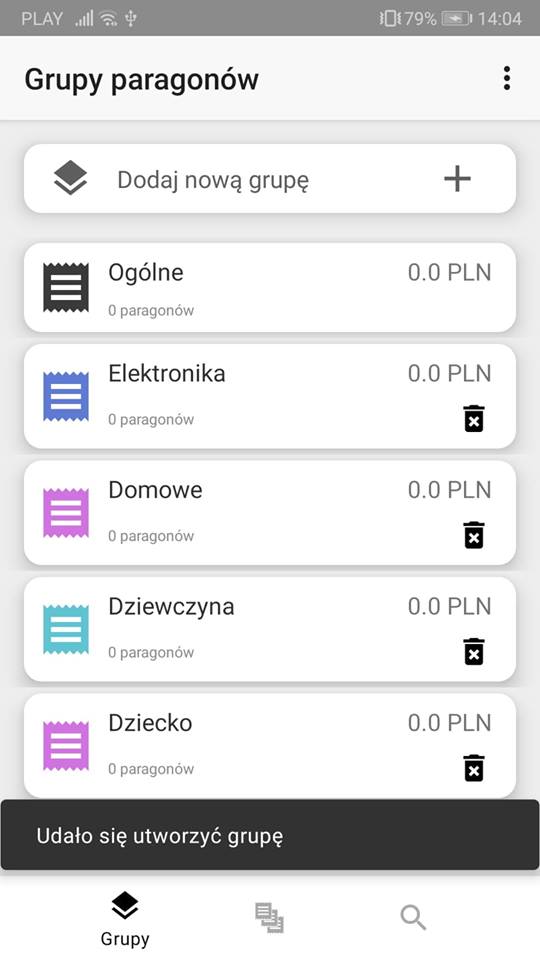
- sumaryczna kwota, wszystkich paragonów należących do grupy,

- ikona grupy wraz jej symbolizującym kolorem,

- przycisk usunięcia grupy.

Grupę można usunąć tylko pod warunkiem, że żaden paragon nie jest do niej przypisany. Grupa o nazwie „Ogólne” nie posiada możliwości usunięcia ani edycji i każdy użytkownik ją posiada. Pozwala to na uniknięcie sytuac istnienia paragonów ,któ®e należą do ządnej grupy. Jest to niedozwolone w aplikajci. Klikając w dowolny element różny od wcześniej wskazano użytkownik przenoszony jest do ekranu edycji grupy opisanej szczegolowe w dalszej części pracy.

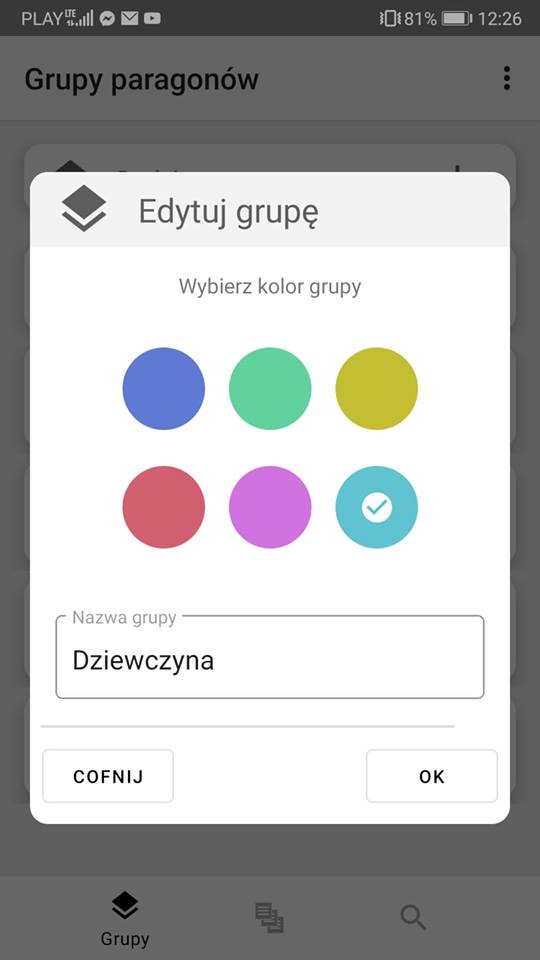
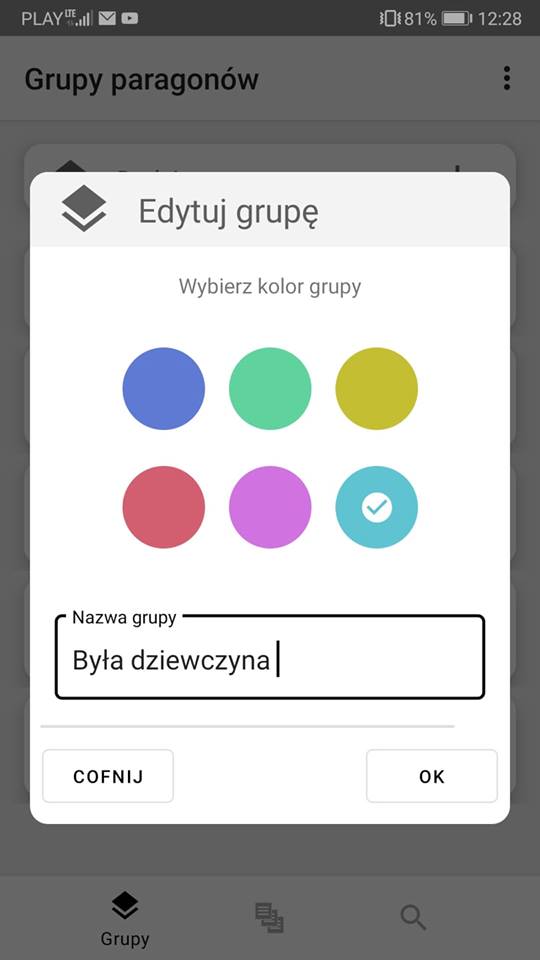
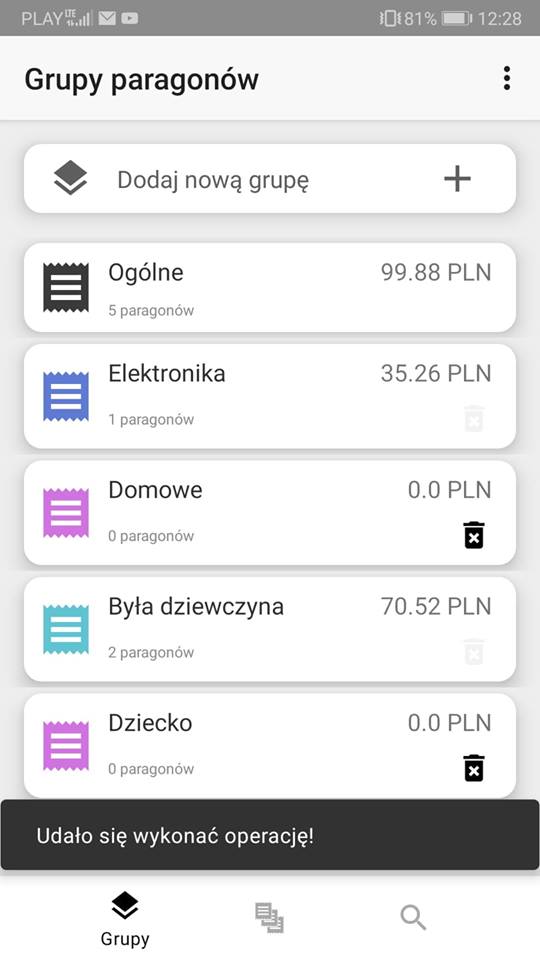
## Dodawanie nowej grupy

*Rysunek 6.13 Ekran dodawania nowej grupy paragonów [Źródło: opracowanie własne]*

Rysunek 6.13 przedstawia ekran, który pojawia się po kliknieciu w przycisk dodania nowej grupy paragonów. Użytkownik może wybirać jeden z sześciu dostępnych kolorów dla grupy oraz wpraowdzić nazwę dla nowej grupy. Nazwa grupy musi być unikalna inaczej użytkownik po zatwierdzeniu dowania nowej grupy zostanie poinformawany o niemożliwości dodaniu ze względu na powtórzoną nazwę grupy. Po zatwierdzeniu i poprawnym zweryfikowaniu nazwy grupa zostaje dodana do apliakjci i synchronizowana z baza danych. Na ekranie pojawia się informacja o poprawności w utworzeniu nowej grupy paragonów.

## Edycja grupy

*Rysunek 6.14 Ekrany edycji grupy [Źródło: opracowanie własne]*

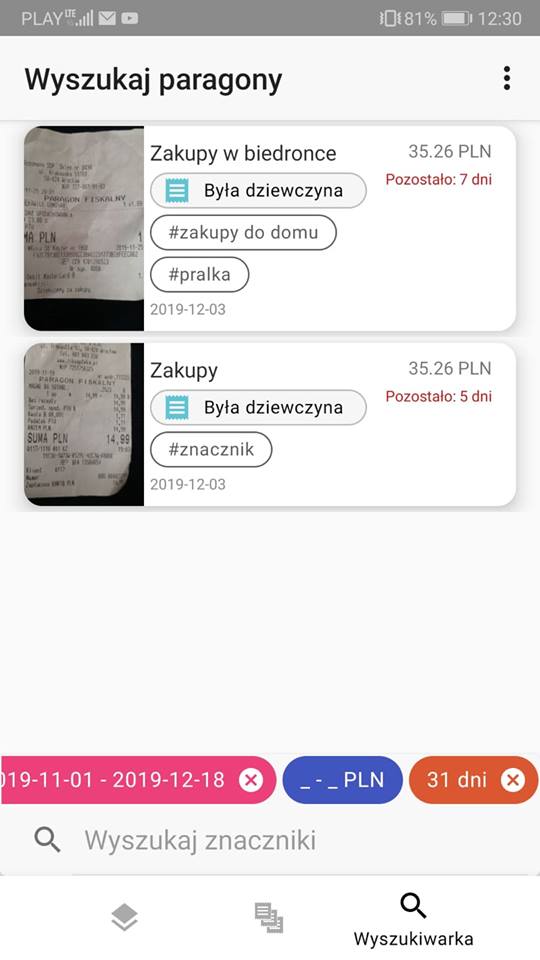
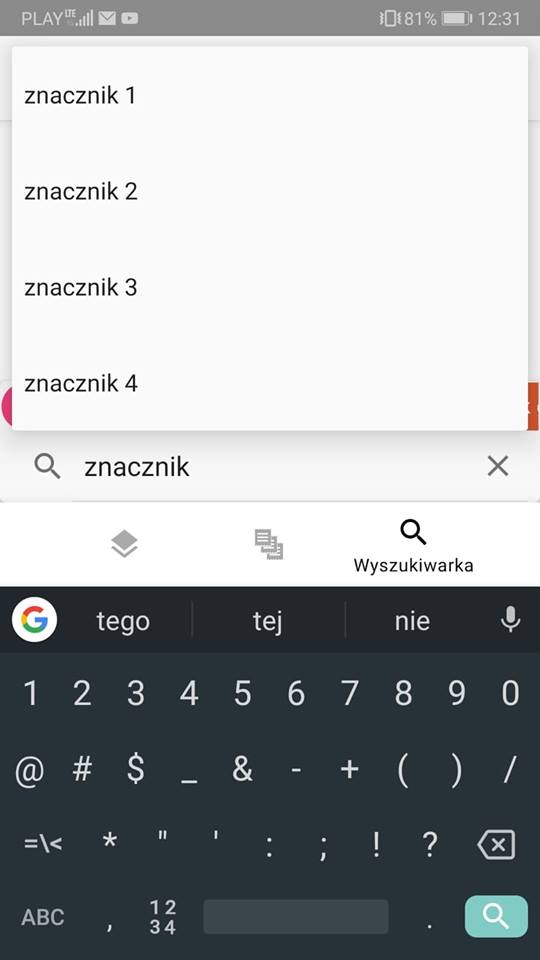
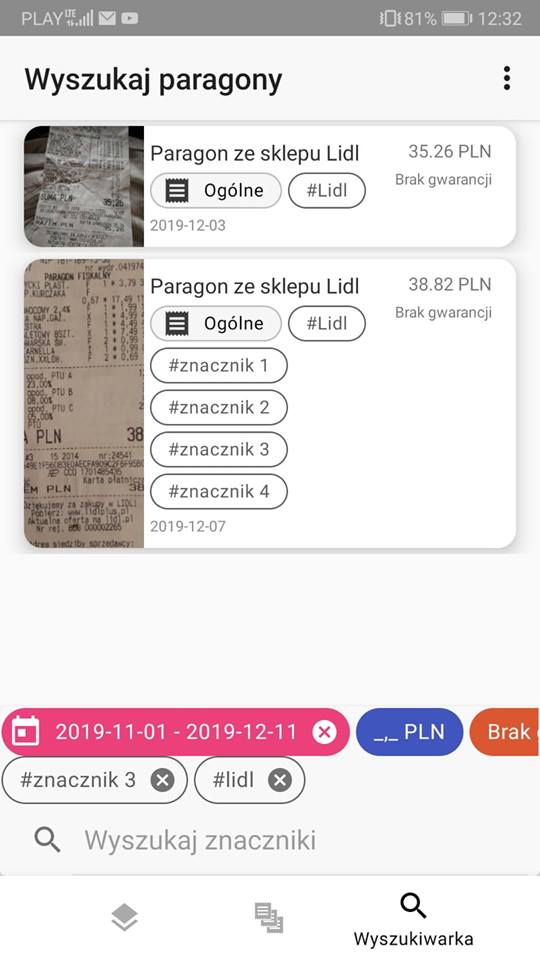
Rysunek 6.14 przedstawia ekran edycji pragonów. Użytkownik po kliknięciu w odpowiednią grupę zostaje przeniesiony do ekranu identycznego z ekranem dodwania dnowej grupy (Rysunek 6.13), z różnicą w tytule znajdującej się na górze oraz elementy są wcześniej wypełnione.

## Wyszukiwanie paragonów

*Rysunek 6.15 Ekrany związane z wyszukiwaniem paragonów [Źródło: opracowanie własne]*

Rysunek 4.11 przedstawia diagram pakietów opisujący wzorzec architektoniczny zastosowany w aplikacji.

*Rysunek 6.16*

# Testy

## Testy jednostkowe



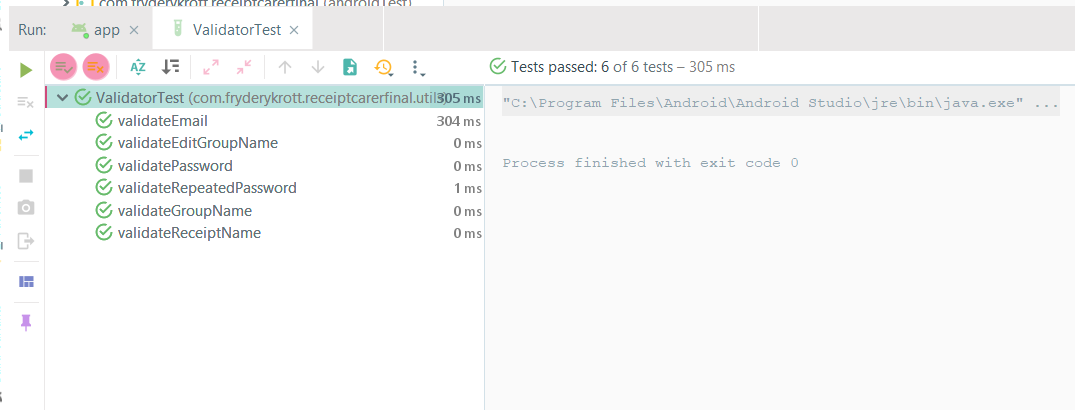
*Rysunek 7.1*



*Rysunek 7.2*



*Rysunek 7.3*



*Rysunek 7.4*

## Testy funkcjonalne

*Tabela 11 Przypadek testowy nr 1 dla PU Eksportuj paragon [Źródło: opracowanie własne]*

|  |  |
| --- | --- |
| Identyfikator | PTU-1 |
| Tytuł | Eksportowanie zdjęć paragonu na podany przez użytkownika prawidłowy adres email. |
| Opis | Celem testu, jest sprawdzenie, czy eksportowanie zdjęć paragonów przebiega w sposób prawidłowy i umożliwia. |
| Warunki wstępne | Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych. |
| Kroki | 1. Otwórz aplikację. 2. Kliknij w wybrany paragon. 3. Kliknij w przycisk eksportu paragonu. 4. Wprowadź adres email. 5. Kliknij przycisk Wyślij. |
| Oczekiwany rezultat | Aplikacja przeniesie użytkownika do wybranej przez niego aplikacji umożliwiającej wysyłanie maili na podany adres email wraz z treścią zawierającą linki URL do zdjęć paragonów wypranego paragonu. Wyświetlony zostaje komunikat o wysłanym emailu w aplikacji. |

*Tabela 12 Przypadek testowy nr 2 dla PU Edytuj grupę*

|  |  |
| --- | --- |
| Identyfikator | PTU-2 |
| Tytuł | Edytowanie grupy paragonów i próba zmiany nazwy grupy na istniejącą już w aplikacji nazwę grupy. |
| Opis | Celem testu, jest sprawdzenie, czy walidacja danych podczas edycji grupy paragonów przebiega w sposób prawidłowy. |
| Warunki wstępne | Użytkownik jest zalogowany w systemie. |
| Kroki | 1. Otwórz aplikację 2. Kliknij w przycisk *Grupy* na dolnym pasku nawigacji. 3. Kliknij w wybraną grupę paragonów. 4. Zmień nazwę grupy na istniejącą już w aplikacji inną nazwę grupy. 5. Kliknij w przycisk zatwierdzania (o nazwie *OK*). |
| Oczekiwany rezultat | Grupa nie może zostać edytowania ze względu na fakt, że istnieje już w systemie grupa o takiej samej nazwie. Wyświetlony zostaje komunikat z informacją, że nie można edytować grupy na grupę o takiej nazwie. |

Zakończenie

1. Podsumowanie pracy
2. Kierunki dalszego rozwoju

# Bibliografia

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://pl.wikipedia.org/wiki/Google. [Data uzyskania dostępu: 27 10 2019]. |
| [2] | [Online]. Available: https://allegro.pl/. [Data uzyskania dostępu: 27 10 2019]. |
| [3] | [Online]. Available: https://dlakupujacych.allegro.pl/bezpieczenstwo/program-ochrony-kupujacych/warunki-skorzystania. [Data uzyskania dostępu: 26 10 2019]. |
| [4] | [Online]. Available: https://www.uokik.gov.pl/. [Data uzyskania dostępu: 22 11 2019]. |
| [5] | [Online]. Available: https://www.uokik.gov.pl/aktualnosci.php?news\_id=11630. [Data uzyskania dostępu: 22 11 2019]. |
| [6] | [Online]. Available: https://www.panparagon.pl/. [Data uzyskania dostępu: 26 10 2019]. |
| [7] | [Online]. Available: https://play.google.com/store/apps/details?id=com.modernfactory.paragon&hl=pl. [Data uzyskania dostępu: 26 10 2019]. |
| [8] | [Online]. Available: https://pl.wikipedia.org/wiki/Java. [Data uzyskania dostępu: 22 11 2019]. |
| [9] | [Online]. Available: https://developer.android.com/studio. [Data uzyskania dostępu: 25 11 2019]. |
| [10] | [Online]. Available: https://firebase.google.com/docs/firestore. [Data uzyskania dostępu: 25 11 2019]. |
| [11] | [Online]. Available: https://pl.wikipedia.org/wiki/Optyczne\_rozpoznawanie\_znak%C3%B3w. [Data uzyskania dostępu: 06 12 2019]. |
| [12] | [Online]. Available: https://pl.wikipedia.org/wiki/Git\_(oprogramowanie). [Data uzyskania dostępu: 25 11 2019]. |
| [13] | [Online]. Available: https://online.visual-paradigm.com/. [Data uzyskania dostępu: 25 11 2019]. |
| [14] | [Online]. Available: https://square.github.io/picasso/. [Data uzyskania dostępu: 06 12 2019]. |
| [15] | [Online]. Available: https://pl.wikipedia.org/wiki/Zintegrowane\_%C5%9Brodowisko\_programistyczne. [Data uzyskania dostępu: 26 11 2019]. |
| [16] | [Online]. Available: https://pl.wikipedia.org/wiki/Android\_(system\_operacyjny). [Data uzyskania dostępu: 25 11 2019]. |
| [17] | [Online]. Available: https://www.jetbrains.com/. [Data uzyskania dostępu: 26 11 2019]. |
| [18] | [Online]. Available: https://www.jetbrains.com/idea/promo/ultimate/?gclid=Cj0KCQiAt\_PuBRDcARIsAMNlBdo2eqaR6SfGD9Rf5x7zCSaYyA\_pd4jkg4b8asVE1bdEGASXtvzHtm8aAlV5EALw\_wcB. [Data uzyskania dostępu: 26 11 2019]. |
| [19] | [Online]. Available: https://codelabs.developers.google.com/codelabs/mlkit-android/#5. [Data uzyskania dostępu: 06 12 2019]. |
| [20] | [Online]. Available: https://pl.wikipedia.org/wiki/Odleg%C5%82o%C5%9B%C4%87\_Levenshteina. [Data uzyskania dostępu: 06 12 2019]. |
| [21] | J. Tracewicz. [Online]. Available: https://www.spidersweb.pl/2018/12/aplikacja-panparagon-warta-ubezpieczenie.html. [Data uzyskania dostępu: 28 10 2019]. |
| [22] | J. Bieniek. [Online]. Available: https://bezprawnik.pl/reklamacja-bez-paragonu-jest-jak-najbardziej-mozliwa-dowodem-moze-byc-potwierdzenie-platnoscia-karta/. [Data uzyskania dostępu: 02 11 2019]. |
| [23] | [Online]. Available: https://play.google.com/store/apps/details?id=com.glority.receipt&hl=pl. [Data uzyskania dostępu: 26 10 2019]. |
| [24] | [Online]. Available: https://play.google.com/store/apps/details?id=com.receiptbank.android&hl=en\_US. [Data uzyskania dostępu: 26 10 2019]. |
| [25] | [Online]. Available: https://play.google.com/store/apps/details?id=com.andrzejsalwin.eparagon&gl=PL. [Data uzyskania dostępu: 22 11 2019]. |
| [26] | [Online]. Available: https://pl.wikipedia.org/wiki/Base64. [Data uzyskania dostępu: 25 11 2019]. |
| [27] | [Online]. Available: https://en.wikipedia.org/wiki/GitHub. [Data uzyskania dostępu: 25 11 2019]. |
| [28] | [Online]. Available: https://api.cloudmersive.com/swagger/index.html?urls.primaryName=Optical%20Character%20Recognition%20(OCR)%20API. [Data uzyskania dostępu: 25 11 2019]. |
| [29] | [Online]. Available: https://api.cloudmersive.com/. [Data uzyskania dostępu: 25 11 2019]. |

Spis rysunków

[*Rysunek 4.1 Diagram klas (domenowy)* 16](#_Toc26704962)

[*Rysunek 4.2 Diagram przypadków użycia* 19](#_Toc26704963)

[*Rysunek 4.3 Dodaj nową grupę paragonów – Lista grup (po lewej). Widok dodawania grupy (po prawej)* 22](#_Toc26704964)

[*Rysunek 4.4* *Dodaj nową grupę paragonów – Uzupełnienie parametrów grupy (po lewej). Dodanie nowej grupy do listy grup wraz z komunikatem (po prawej).* 22](#_Toc26704965)

[*Rysunek 4.5* Przeszukaj paragony *– kryteria wyszukania. Wybór daty dodania paragonu (po lewo), wybór kwot (środek) oraz wybór długości trwania gwarancji (po prawo)* 24](#_Toc26704966)

[*Rysunek 4.6 Przeszukaj paragony – brak kryteriów (po lewo), wypełnienie kryteriów i wynik wyszukania (środek), komunikat w przypadku braku znalezienia paragonów dla poszczególnych kryteriów.* 24](#_Toc26704967)

[Rysunek 4.7 *Dodaj paragony – kolejno krok 0 (z lewej) oraz krok 1 (z prawej)* 26](#_Toc26704968)

[*Rysunek 4.8 Dodaj paragony –kolejno krok 2 (z lewej) oraz krok 3 (z prawej)* 26](#_Toc26704969)

[*Rysunek 5.1 Diagram ERD bazy danych* 29](#_Toc26704970)

[*Rysunek 5.2 Diagram rozmieszczenia* 30](#_Toc26704971)

[*Rysunek 5.3 Diagram pakietów – wzorzec MVP* 31](#_Toc26704972)

[*Rysunek 5.4 Wynik algorytmu OCR od Firebase w formie bloczków tekstu [Źródło:* [19]*]* 32](#_Toc26704973)

[*Rysunek 5.5 Przykładowe formy zapisu kwoty sumarycznej na paragonie [Źródło: opracowanie własne]* 33](#_Toc26704974)

[*Rysunek 6.1 Ekran logowania – widok główny (po lewo), informacja o błędach (po prawo) [Źródło: opracowanie własne]* 35](#_Toc26704975)

[*Rysunek 6.2 Ekran rejestracji – widok główny (po lewo), informacja o błędach (po prawo) [Źródło: opracowanie własne]* 36](#_Toc26704976)

[*Rysunek 6.3 Ekran listy paragonów [Źródło: opracowanie własne]* 37](#_Toc26704977)

[*Rysunek 6.4 Ekran edytowania paragonu – widok edycji (po lewej), informacja o błędzie nazwy (po prawej) [Źródło: opracowanie własne]* 38](#_Toc26704978)

[*Rysunek 6.5 Ekrany specjalne: wybór daty dodania paragonu (po lewej), wybór długości trwania gwarancji (środek i po prawej) [Źródło opracowanie własne]* 39](#_Toc26704979)

[*Rysunek 6.6 Ekrany specjalne: wybór całkowitej kwoty z paragonu (po lewej), wybór grupy do której należy paragon (po prawej) [Źródło: opracowanie własne]* 39](#_Toc26704980)

[*Rysunek 6.7 Dodawanie nowego znacznika – wpisanie nazwy znacznika (krok 0, po lewej), zaakceptowanie znacznika do paragonu (krok 1, po prawej), wybór znacznika z lity podpowiadającej znaczniki [Źródło: opracowanie własne]* 40](#_Toc26704981)

[*Rysunek 6.8 Możliwość dodania nowych zdjęć do paragonu (po lewo), podgląd zdjęć paragonów (po prawo) [Źródło: opracowanie własne]* 41](#_Toc26704982)

[*Rysunek 6.9 Eksport paragonu [Źródło: opracowanie własne]* 42](#_Toc26704983)

[*Rysunek 6.10 ekran dodawania zdjęć – brak wykonanych zdjęć (po lewej), wykonano jedno zdjęcie, wykonano 2 zdjęcia (środek oraz po prawo) [Źródło: opracowanie własne]* 43](#_Toc26704984)

[*Rysunek 6.11 Ekran edycji dodawanych paragonów [Źródło: opracowanie własne]* 44](#_Toc26704985)

[*Rysunek 6.12 Ekran z listą grup paragonów [Źródło: opracowanie własne]* 45](#_Toc26704986)

[*Rysunek 6.13* 46](#_Toc26704987)

[*Rysunek 6.14* 47](#_Toc26704988)

[*Rysunek 6.15* 47](#_Toc26704989)

[*Rysunek 6.16* 48](#_Toc26704990)

Spis tabel

[*Tabela 1 Opis użytkownika systemu* 6](#_Toc26704991)

[*Tabela 2 Potrzeby i cechy* 7](#_Toc26704992)

[*Tabela 3 Inne wymagania produktowe* 7](#_Toc26704993)

[*Tabela 4 Porównanie oferowanych funkcjonalności* 14](#_Toc26704994)

[*Tabela 5 Słownik pojęć* 15](#_Toc26704995)

[*Tabela 6 Macierz śladowania wymagań* 18](#_Toc26704996)

[Tabela 7 Skrócone opisy przypadków użycia 20](#_Toc26704997)

[*Tabela 8 Dodaj grupę* 21](#_Toc26704998)

[*Tabela 9 Przeszukuj paragony - Scenariusz* 23](#_Toc26704999)

[*Tabela 10 Dodanie paragonów - Scenariusz* 25](#_Toc26705000)

Załącznik