



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

Praca dyplomowa – inżynierska

## Aplikacja mobilna wspierająca zarządzanie paragonami

Fryderyk Rott

słowa kluczowe:  
paragon, dowód zakupu,  
skanowanie, organizacja,  
firebase, java

krótkie streszczenie:

Praca przedstawia specyfikację wymagań, projekt, implementację oraz testy aplikacji umożliwiającej zarządzanie dokumentami stanowiącymi dowody zakupu (paragony).

opiekun pracy	dr inż. Bogumiła Hnatkowska	.....	.....
dyplomowej	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego	.....	.....	.....
	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do: \*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

\* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław, rok 2020



## **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, wyszukiwanie, a także gromadzenie informacji o kończących się terminach gwarancji.

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 podsumowanie 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, searching them through and collecting data about ending warranty periods 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

Wstęp .....	1
1. Wprowadzenie do problematyki .....	1
2. Geneza i motywacja pracy .....	2
3. Cel i zakres pracy .....	3
4. Struktura pracy .....	3
1. Sposoby organizacji paragonów .....	4
1.1. Aspekt prawny .....	4
1.2. Sposoby wykorzystania paragonów .....	5
2. Wizja systemu .....	6
2.1. Wprowadzenie .....	6
2.2. Zidentyfikowanie grupy użytkowników .....	6
2.3. Potrzeby i cechy .....	7
2.4. Inne wymagania produktowe .....	7
3. Rozwiązania konkurencyjne .....	8
3.1. PanParagon .....	8
3.2. Paragon: Paragony i karty lojalnościowe w jednym miejscu .....	10
3.3. Receipt Lens-Expense Tracking & Reporting .....	12
3.4. Receipt Bank .....	13
3.5. Podsumowanie .....	14
4. Specyfikacja i analiza wymagań .....	15
4.1. Słownik pojęć .....	15
4.2. Reguły biznesowe i model domenowy .....	16
4.3. Historyjki użytkowników .....	17
4.4. Diagram Przypadków użycia .....	20
4.5. Skrócony opis przypadków użycia .....	21
4.6. Scenariusze przypadków użycia wraz z prototypami interfejsów .....	22
5. Projekt aplikacji ReceiptCarer .....	28
5.1. Opis wybranych wykorzystanych środowisk i narzędzi programistycznych .....	28
5.2. Architektura rozwiązania .....	30
5.2.1. Architektura fizyczna .....	30
5.2.1. Architektura logiczna .....	31
5.3. Projekt Bazy Danych .....	32
5.4. Ekstrahowanie informacji z paragonów .....	34
6. Implementacja aplikacji ReceiptCarer .....	38
6.1. Rozwiązanie problemu synchronizacji zdjęć .....	38
6.2. Logowanie .....	38

6.3.	Rejestracja .....	39
6.4.	Lista paragonów .....	40
6.5.	Paragon – widok szczegółowy i edycja.....	41
6.6.	Dodawanie nowych paragonów .....	45
6.7.	Eksportowanie paragonów .....	47
6.8.	Lista grup paragonów .....	48
6.9.	Dodawanie nowej grupy .....	49
6.10.	Edycja grupy .....	50
6.11.	Wyszukiwanie paragonów .....	50
7.	Testy .....	52
7.1.	Testy jednostkowe .....	52
7.2.	Testy funkcjonalne .....	54
7.2.1.	Testy dla PU Eksportuj paragon .....	54
7.2.2.	Testy dla PU Dodaj paragony .....	55
7.2.3.	Testy dla PU Edytuj grupę .....	57
	Zakończenie .....	59
1.	Podsumowanie pracy .....	59
2.	Kierunki dalszego rozwoju .....	59
	Bibliografia .....	60
	Spis rysunków .....	62
	Spis tabel .....	63

# Wstęp

## 1. 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 niezadowolających 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.

## **2. 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 odsprzedaży produktu – dzięki możliwości udowodnienia ceny produktu w momencie zakupu oraz wciąż trwającego okresu gwarancji, osoby odsprzedażające produkty mogą wynegocjować lepsze ceny sprzedaży.

Podczas rozmów z ludźmi zainteresowanymi 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.

### **3. 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, wyszukiwanie, a także gromadzenie informacji o kończących się terminach gwarancji.

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.

### **4. 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.

# 1. 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 zarządzane 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 scentralizowania wszystkich dowodów zakupów w jednym miejscu.

## 1.1. 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 jasno 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.

## **1.2. Sposoby wykorzystania paragonów**

Dowody zakupów znajdują zastosowanie w wielu różnych sytuacjach. Przeważnie wykorzystywane są one do udowodnienia zakupu w celu reklamacji zakupionego produktu (lub w celu jego zwrotu, 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 pełnią bardzo ważną rolę przy zarządzaniu wydatkami. Ze względu na to, że są najpopularniejszą formą dowodów zakupów, przechowywanie i kategoryzowanie ich pozwala także na prowadzenie analizy wydatków.

## 2. Wizja sytemu

### 2.1. 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 paragonów poprzez skanowanie ich i ekstrakowanie podstawowych informacji z nimi związanymi. Aplikacja w pierwszej wersji nie przewiduje 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.

### 2.2. 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
<b>Użytkownik indywidualny</b>	Osoba zajmująca się organizacją paragonów na potrzeby własne.	Zbieranie i ewidencjonowanie paragonów.  Pilnowanie nadchodzących terminów kończących gwarancje.  Przekazywanie dowodów zakupów w razie potrzeby.

## 2.3. Potrzeby i cechy

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

*Tabela 2 Potrzeby i cechy*

Potrzeba	Priorytet	Cechy
<b>Użytkownicy indywidualni potrzebują wsparcia dla zarządzania paragonami</b>	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.
<b>Użytkownicy indywidualni potrzebują wsparcia do dodawania wielu paragonów w jak najszybszym czasie</b>	Wysoki	Możliwość dodawania wielu paragonów w ramach jednej operacji dodawania.
<b>Użytkownicy indywidualni potrzebują dostawać powiadomień o nadchodzących kończących się terminach gwarancji.</b>	Wysoki	Automatyczne pojawianie się powiadomień o kończących się terminach gwarancji.

## 2.4. Inne wymagania produktowe

Tabela 3. Przedstawia dodatkowe wymagania нефункционалне, 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 urządzeniach mobilnych z wbudowanym aparatem umożliwiającym 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*

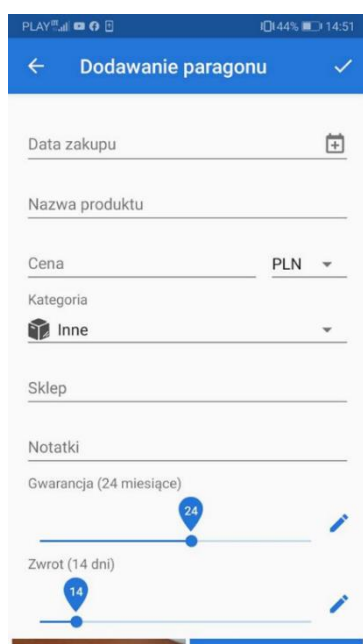
Potrzeba	Priorytet	Cechy
<b>Kompatybilność z systemami Android w wersji co najmniej 9.0</b>	Wysoki	Możliwość zainstalowania i użytkowania aplikacji na systemach z androidem w wersji co najmniej 9.0.
<b>Przechowywanie danych użytkownik w chmurze</b>	Wysoki	Możliwość rejestrowania i logowania się w aplikacji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

### 3. Rozwiązania konkurencyjne

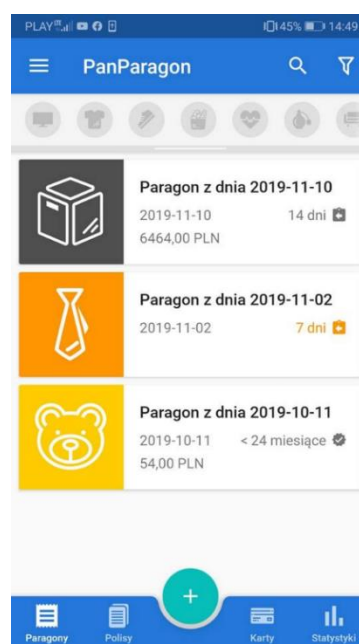
W tym rozdziale zostaną przedstawione rozwiązania konkurencyjne, które w całości lub po części pozwalają na organizację paragonów.

#### 3.1. PanParagon

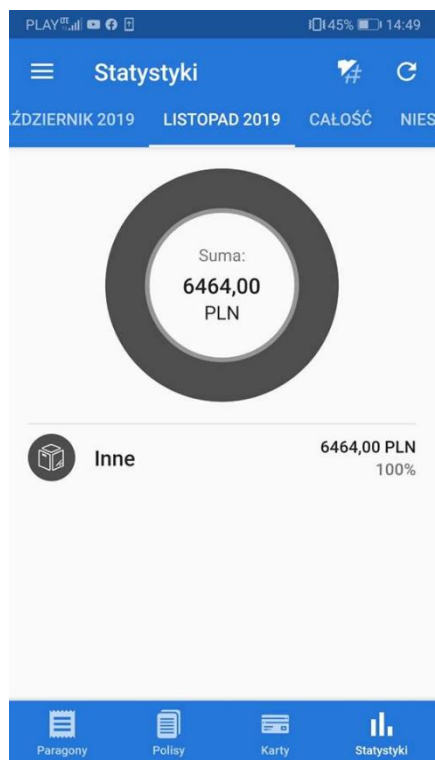
Pierwsza na liście rozwiązań konkurencyjnych jest najbardziej popularna aplikacja mobilna na systemy Android, co za tym idzie największa konkurencja dla projektowanego systemu. PanParagon [6] oferuje przechowywanie (rysunek 3.1), dodawanie (rysunek 3.2) i zarządzanie paragonami. Nie posiada jednak możliwości 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 kategorie 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 3.1 – PanParagon [6]. ekran główny, widok paragonów [Źródło: opracowanie własne]



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

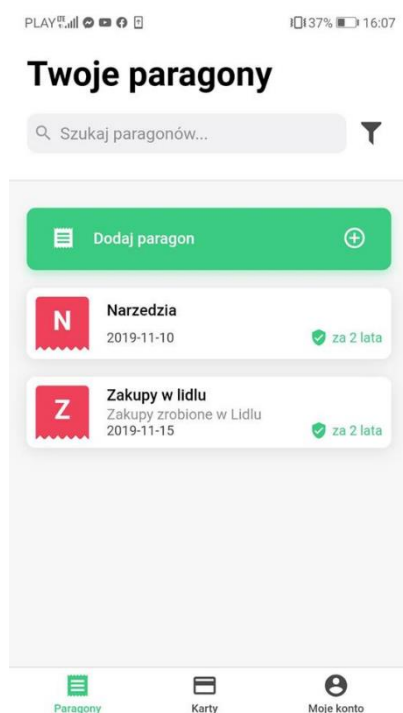


Rysunek 3.3 – PanParagon [6], widok statystyk dla listopada 2019 r. [Źródło: opracowanie własne]

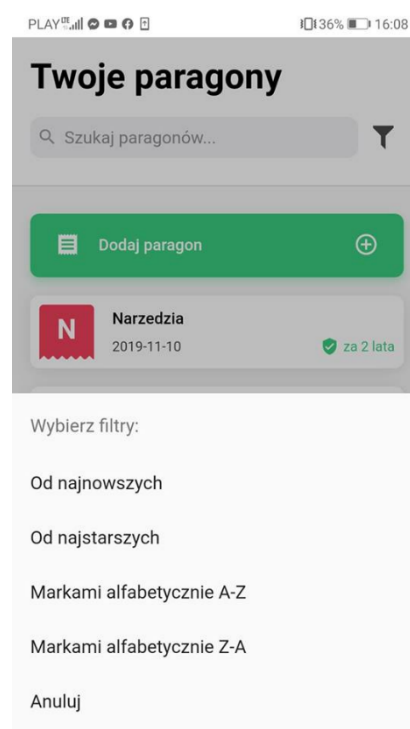
Rysunek 3.4 – PanParagon [6], widok filtrowania paragonów. [Źródło: opracowanie własne]

### 3.2. Paragon: Paragony i karty lojalnościowe w jednym miejscu

Kolejnym rozwiązaniem konkurencyjnym jest zdecydowanie mniej zaawansowana od poprzedniego rozwiązania konkurencyjnego 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, robi to tylko z poziomu aplikacji (nie wysyła powiadomień poza aplikacją) za pomocą informacji kolorystycznych.



Rysunek 3.5 – Paragon: Paragony i karty lojalnościowe w jednym miejscu [26], widok zarządzania paragonami [Źródło: opracowanie własne]



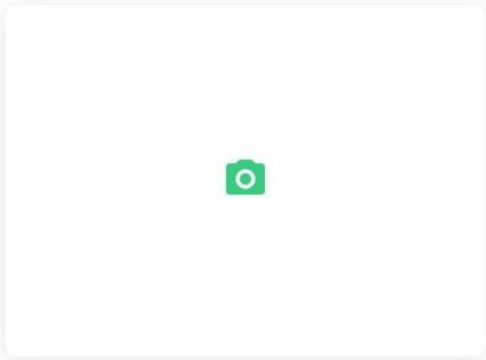
Rysunek 3.6 – Paragon: Paragony i karty lojalnościowe w jednym miejscu [26], widok filtrowania paragonów [Źródło: opracowanie własne]



PLAY 36% 16:08

← Edytuj paragon

Zdjęcie



Nazwa produktu


Sklep

Kolor paragonu

☒ Domyślny kolor

Data zakupy

10.11.2019

 Zapisz zmiany

PLAY 36% 16:08

← Edytuj paragon

Nazwa produktu

Sklep

Kolor paragonu

☒ Domyślny kolor


Data zakupy

10.11.2019

Okres gwarancji (w miesiącach)

24

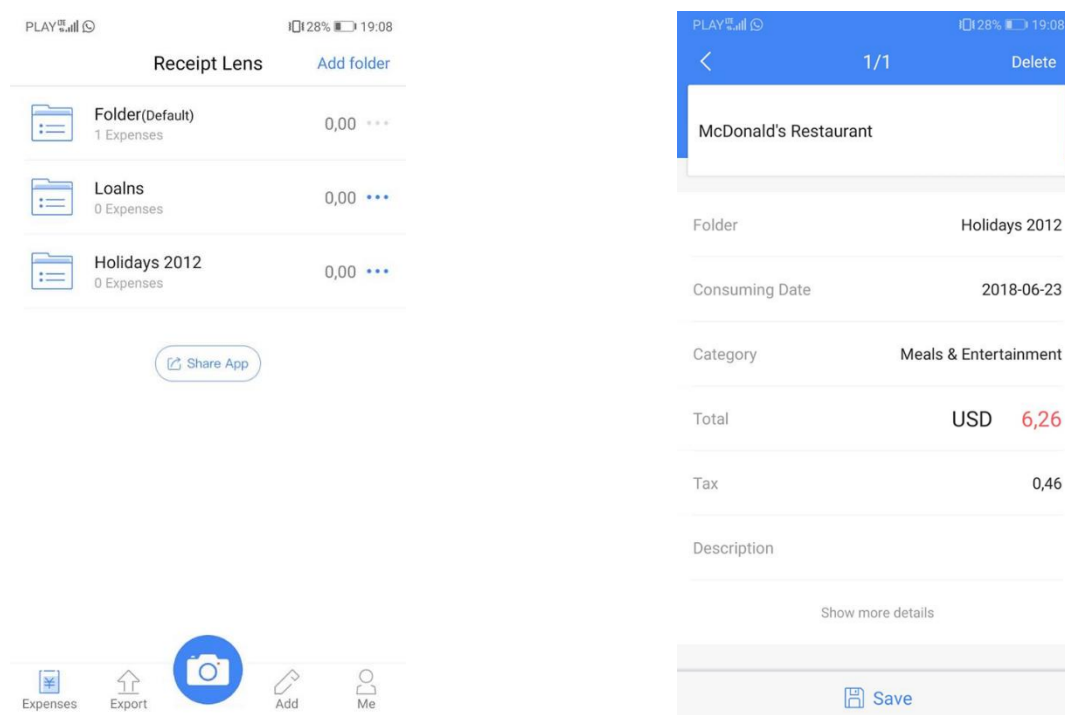
Kilka słów o paragonie...

 Zapisz zmiany

Rysunek 3.7 – Paragon: Paragony i karty lojalnościowe w jednym miejscu [26], widok dodawania paragonu [Źródło: opracowanie własne]

### 3.3. 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 kończących się terminach gwarancji. Aplikacja posiada możliwość tworzenia konta i przechowywania danych w chmurze. Można z niej korzystać lokalnie bez logowania się.

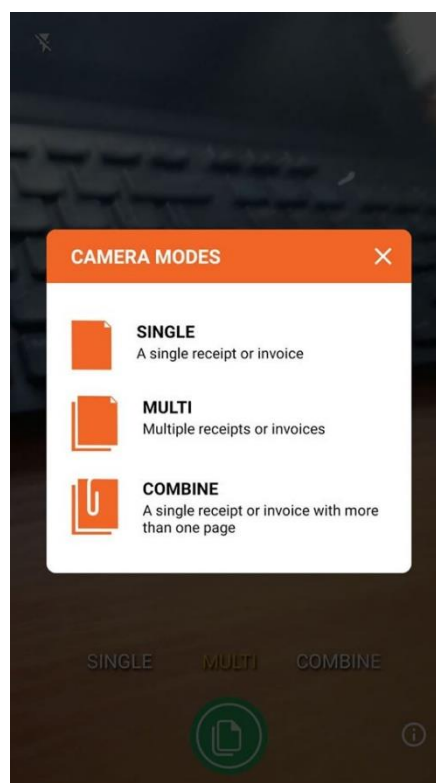
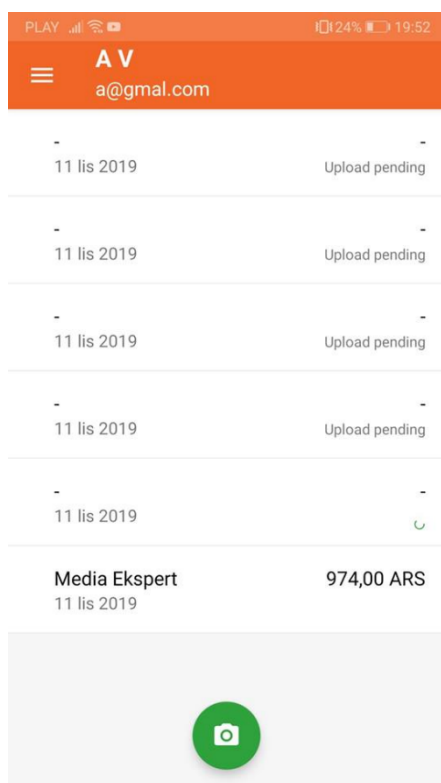


Rysunek 3.8 - Receipt Lens-Expense Tracking & Reporting [25], widok ekranu głównego z folderami. [Źródło: opracowanie własne]

### 3.4. 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 [8]. 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 [24], widok ekranu głównego. [Źródło: opracowanie własne]

### 3.5. 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 najszerzej 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 [Źródło: opracowanie własne]*

	PanParagon	Paragon: Paragony i karty lojalnościowe w jednym miejscu	Receipt Lens- Expense Tracking & Reporting	Receipt Bank	Proponowa ne rozwiązanie
<b>Przechowywanie danych w chmurze</b>	+ Lokalna i zewnętrzna baza danych	+ Zewnętrzna baza danych	+ Lokalna i zewnętrzna baza danych	+ Zewnętrzna baza danych	+ Zewnętrzna baza danych
<b>Dodawanie paragonów</b>	Pojedynczo	Pojedynczo	Pojedynczo lub kilka naraz	Pojedynczo lub kilka naraz	Pojedynczo lub kilka naraz
<b>Skanowanie paragonów</b>	+		+		+
<b>Kategoryzowanie</b>	+ Ustalane z góry kategorie		+ System tworzonych przez użytkownika folderów	+ Ustalane z góry kategorie	+ System tworzenia przez użytkownika grup
<b>Przeszukiwanie paragonów</b>	+ Zaawansowane filtrowanie		+ Tylko po datach dodania i w obrębie jednego folderu		+
<b>Sortowanie paragonów</b>	+	+			
<b>Eksportowanie paragonów</b>	+ 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)
<b>Przechowywanie informacji na temat gwarancji</b>	+	+			+
<b>Powiadamianie o kończących się terminach gwarancji</b>	+				+

## 4. Specyfikacja i analiza wymagań

### 4.1. Słownik pojęć

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

*Tabela 5 Słownik pojęć [Źródło: opracowanie własne]*

Termin	Synonim	Definicja
<b>Eksport paragonów</b>	Wysyłka paragonów	Oznacza wysłanie zdjęcia lub zdjęć paragonów poza aplikację na dany adres mailowy.
<b>Filtrowanie</b>	Przeszukiwanie	Możliwość wyodrębniania poszczególnych elementów spośród listy tych elementów na podstawie unikatowych informacji posiadanych przez elementy.
<b>Grupa</b>	Grupa paragonów	Grupy służą do grupowania paragonów. Posiadają informację na temat liczby znajdujących się w niej paragonów. Są zarządzane przez użytkowników.
<b>Gwarancje</b>		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) oraz przed upłynięciem terminu ważności gwarancji.
<b>Paragon</b>	Dowód zakupu, Rachunek,	Unikatowy dokument papierowy, stanowiący dowód zakupu produktu lub usługi.
<b>Powiadomienie</b>		To czynne poinformowanie użytkownika aplikacji o sytuacji zaistniałej w systemie.
<b>Rejestracja</b>		To proces zakładania konta użytkownika. Po założeniu konta użytkownik może się logować do aplikacji. Na podstawie danych wprowadzonych przy rejestracji użytkownik jest weryfikowany w systemie podczas logowania.
<b>Skanowanie</b>	Skanowanie paragonu	Możliwość ekstrahowania informacji na podstawie zdjęcia paragonu.
<b>Termin gwarancji</b>		Graniczna data, do której istnieje możliwość oddania zakupionego, wadliwego towaru w ramach gwarancji.
<b>Użytkownik</b>		Osoba będąca w posiadaniu aplikacji i korzystająca z jej funkcjonalności.
<b>Znacznik</b>	Tag	Słowo lub słowa kluczowe przechowujące informację na temat paragonu.

## 4.2. 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. W regułach biznesowych zostały pominięte reguły związane z logowaniem i rejestracją gdyż jest to zapewnione przez zewnętrzne narzędzie.

### **Użytkownik**

- REG001. Użytkownik może lecz nie musi posiadać wiele paragonów
- REG002. Użytkownik musi posiadać co najmniej jedną grupę
- REG003. Użytkownik musi posiadać grupę o nazwie „Ogólne”
- REG004. Grupa o nazwie „Ogólne” nie może zostać usunięta przez użytkownika
- REG005. Użytkownik musi posiadać nazwę użytkownika
- REG006. Nazwa użytkownika musi być unikalna
- REG007. Użytkownik musi posiadać adres email
- REG008. Adres email musi być unikalny

### **Paragon**

- REG009. Paragon musi należeć tylko do jednego użytkownika
- REG010. Paragon musi należeć tylko do jednej grupy
- REG011. Paragon musi posiadać datę dodania
- REG012. Paragon może posiadać datę kończącą gwarancję
- REG013. Data kończąca gwarancję musi być późniejsza niż data dodania
- REG014. Paragon musi posiadać nazwę paragonu
- REG015. Nazwa paragonu nie może być pustym ciągiem znaków
- REG016. Paragon nie musi posiadać żadnego znacznika
- REG017. Paragon musi posiadać co najmniej jedno zdjęcie
- REG018. Paragon musi posiadać kwotę pieniężną (znajdującą się na paragonie)
- REG019. Suma pieniężna musi być większa lub równa zero

### **Znacznik**

- REG020. Znacznik musi należeć do co najmniej jednego paragonu
- REG021. Znacznik musi posiadać nazwę znacznika
- REG022. Nazwa znacznika musi być unikalna
- REG023. Nazwa znacznika musi rozpoczynać się od znaku „#”

### **Grupa**

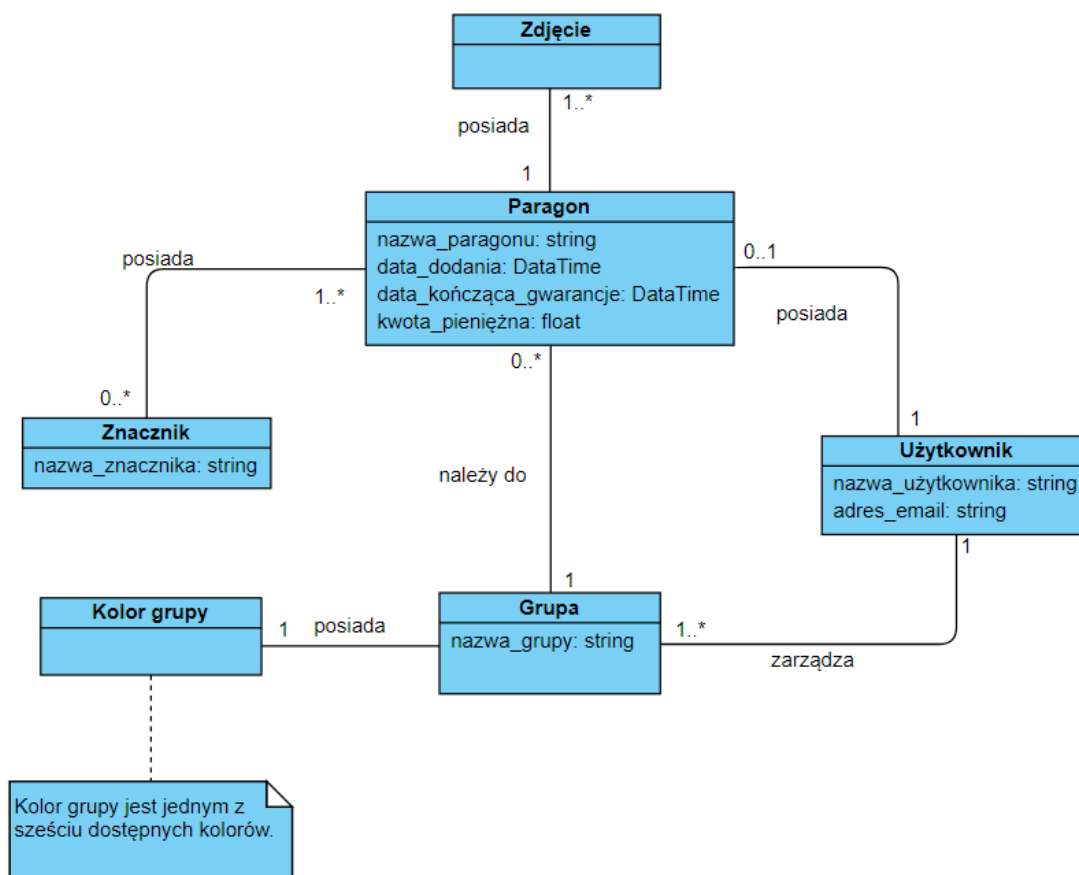
- REG024. Grupa musi być przypisana do użytkownika
- REG025. Do grupy nie musi być przypisany żaden paragon
- REG026. Grupa musi posiadać nazwę grupy
- REG027. Nazwa grupy musi być unikalna
- REG028. Grupa musi posiadać kolor grupy

### **Kolor grupy**

- REG029. Kolor grupy musi być jednym z sześciu dostępnych w systemie kolorów

### **Zdjęcie**

- REG030. Zdjęcie musi być przypisane do paragonu



Rysunek 4.1 Diagram klas (domenowy) [Źródło: opracowanie własne]

### 4.3. Historyjki użytkowników

Po analizie potrzeb użytkowników systemu przedstawiono wymagania systemu w formie historyjek użytkownika. W tabeli 6 przedstawiono macierz śładowania 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 paragonu 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 uznaję, 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.

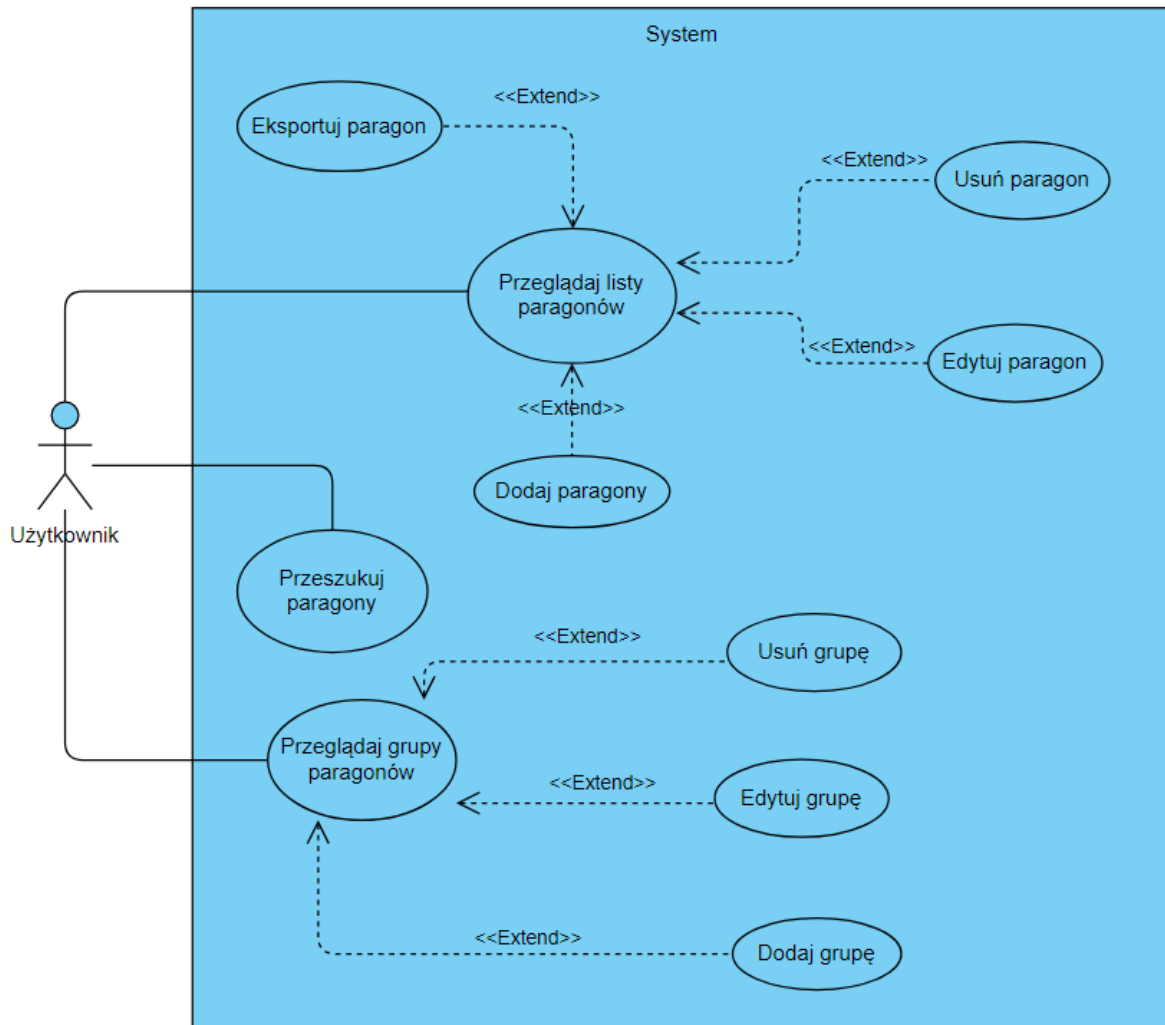


Tabela 6 Macierz śladowania wymagań [Źródło: opracowanie własne]

	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
<b>HU01.</b>		+											
<b>HU02.</b>			+										
<b>HU03.</b>					+								
<b>HU04.</b>				+									
<b>HU05.</b>						+							
<b>HU06.</b>	+												
<b>HU07.</b>							+	+					
<b>HU08.</b>								+					
<b>HU09.</b>										+			
<b>HU10.</b>									+				
<b>HU11.</b>											+		
<b>HU12.</b>												+	
<b>HU13.</b>													+

#### 4.4. 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 przedstawione przypadki użycia związane z logowaniem i rejestracją użytkowników.



Rysunek 4.2 Diagram przypadków użycia  
[Źródło: opracowanie własne]

#### 4.5. Skrócony opis przypadków użycia

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

*Tabela 7 Skrócone opisy przypadków użycia [Źródło: opracowanie własne]*

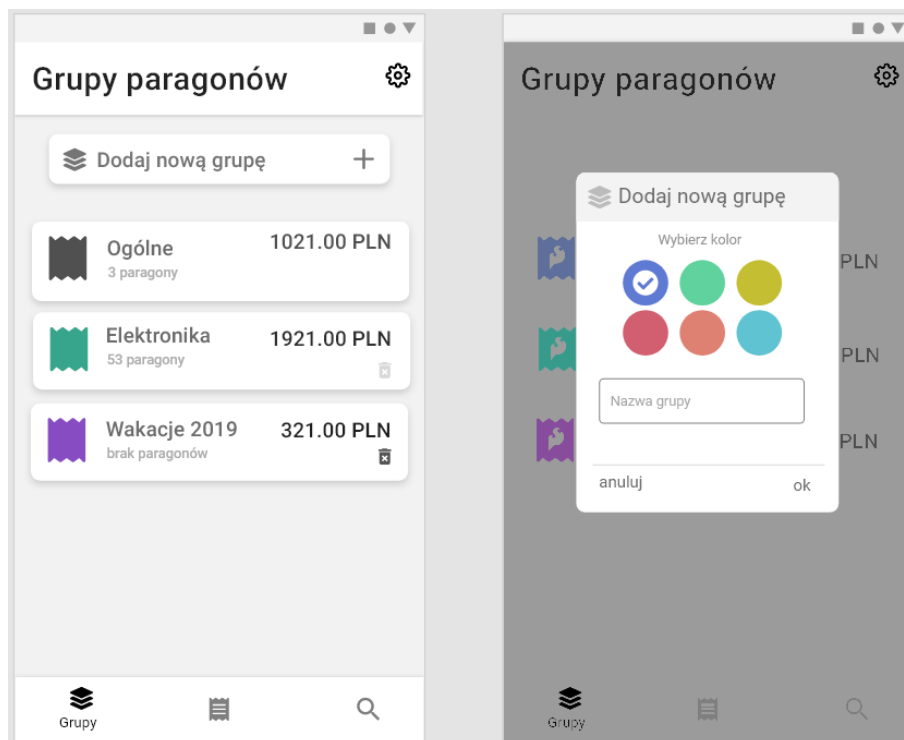
Nazwa przypadku użycia	Skrócony opis
<b>Dodaj grupę</b>	System umożliwia dodanie nowej grupy paragonów o podanych parametrach.
<b>Dodaj paragony</b>	System umożliwia dodanie nowych paragonów o podanych parametrach.
<b>Edytuj grupę</b>	System umożliwia zmianę parametrów istniejącej grupy paragonów.
<b>Edytuj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

#### 4.6. Scenariusze przypadków użycia wraz z prototypami interfejsów

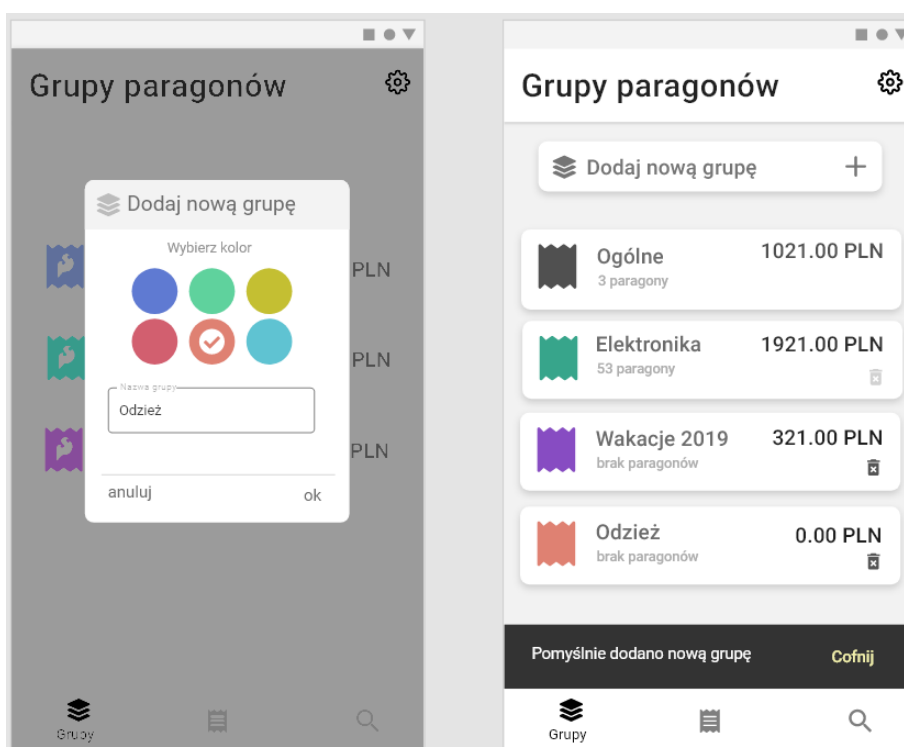
W następującym podrozdziale przedstawiono przykładowe przebiegi przypadków użycia wraz ze scenariuszami alternatywnymi i wyjątkowymi. W tabeli 8. Przedstawiono przebieg scenariusza związanego z oddawaniem nowej grupy. Następnie na rysunkach 4.3 oraz 4.4 przedstawiono kolejno ekrany związane z przedstawionym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

*Tabela 8 Dodaj grupę [Źródło: opracowanie własne]*

Nazwa przypadku użycia	Dodaj grupę
<b>Aktorzy</b>	Użytkownik
<b>Warunki wstępne</b>	Użytkownik jest zalogowany, system wyświetla listę grup. [Rysunek 4.3] (PU przeglądaj grupy paragonów)
<b>Warunki końcowe</b>	Do systemu zostaje dodana nowa grupa paragonów
<b>Scenariusz główny</b>	<ol style="list-style-type: none"> <li>1. Użytkownik zleca dodanie grupy paragonów.</li> <li>2. System wyświetla parametry do wypełnienia dla nowej grupy paragonów (nazwa grupy, kolor grupy). [Rysunek 4.3]</li> <li>3. Użytkownik wprowadza dane nowej grupy paragonów. [Rysunek 4.4]</li> <li>4. Użytkownik zatwierdza akcję dodania.</li> <li>5. System sprawdza poprawność danych (REG027)</li> <li>6. System zapamiętuje grupę paragonów i wyświetla powiadomienie o jego dodaniu. [Rysunek 4.4]</li> <li>7. Zakończenie scenariusza.</li> </ol>
<b>Scenariusz alternatywny 1</b>	<ol style="list-style-type: none"> <li>4a. Użytkownik anuluje akcję dodania.</li> <li>1. Powrót do punktu 1. Scenariusza głównego.</li> </ol>
<b>Scenariusz alternatywny 2</b>	<ol style="list-style-type: none"> <li>3a. Użytkownik zatwierdza akcję dodania przed wypełnieniem pól.</li> <li>1. System wyświetla powiadomienie o braku obowiązkowych danych.</li> <li>2. Następuje powrót do punktu 2. Scenariusza głównego.</li> </ol>
<b>Scenariusz wyjątku</b>	<ol style="list-style-type: none"> <li>5* W systemie istnieje grupa paragonów o takiej nazwie (REG027).</li> <li>1. System wyświetla powiadomienie o tym, że grupa o podanej nazwie już istnieje w systemie.</li> <li>2. Następuje powrót do punktu 1. Scenariusza głównego.</li> </ol>



*Rysunek 4.3 Dodaj nową grupę paragonów – Lista grup (po lewej).  
Widok dodawania grupy (po prawej) [Źródło: opracowanie własne]*

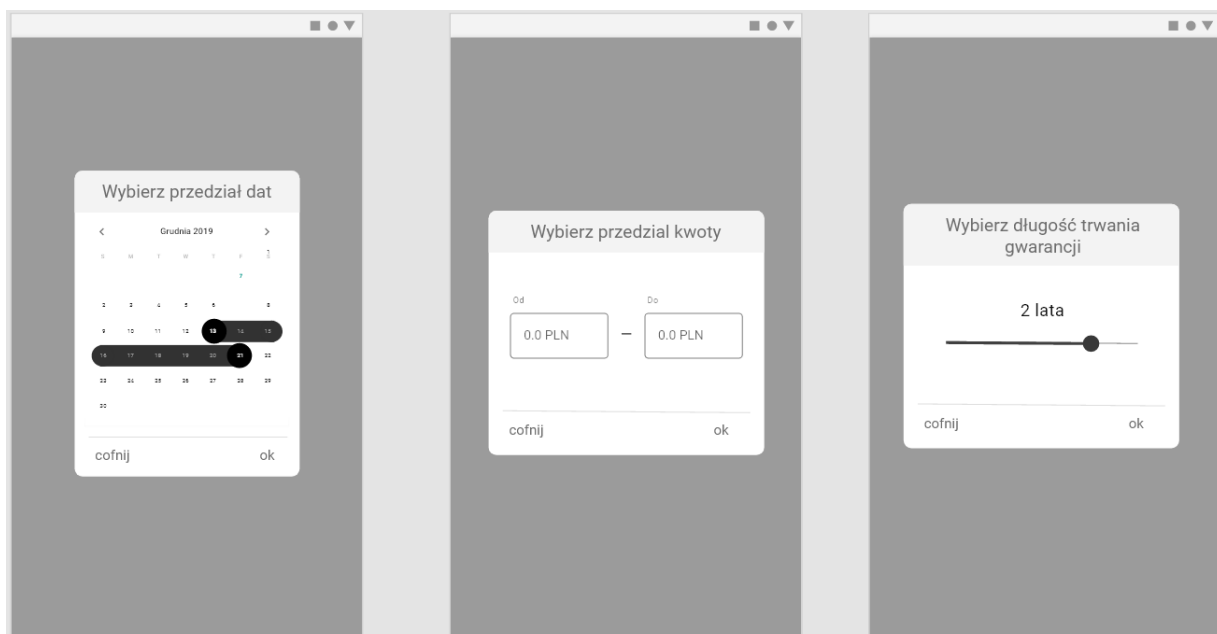


*Rysunek 4.4 Dodaj nową grupę paragonów – Uzupełnienie parametrów grupy (po lewej).  
Dodanie nowej grupy do listy grup wraz z komunikatem (po prawej).  
[Źródło: opracowanie własne]*

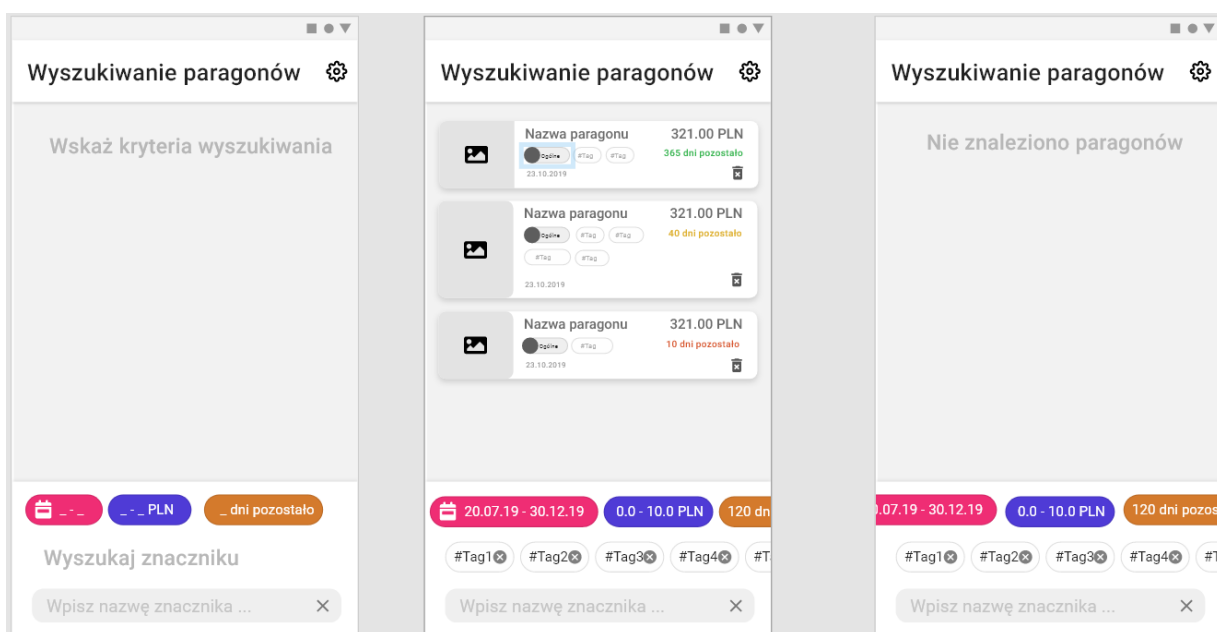
W tabeli 9. Przedstawiono przebieg scenariusza związanego z przeszukiwaniem paragonów. Następnie na rysunkach 4.5 oraz 4.6 przedstawiono kolejno ekrany związane ze przedstawionym scenariuszem.

*Tabela 9 Przeszukuj paragony - Scenariusz*

Nazwa przypadku użycia	Przeszukuj paragony
<b>Aktorzy</b>	Użytkownik
<b>Warunki wstępne</b>	Użytkownik jest zalogowany.
<b>Warunki końcowe</b>	System wyświetla paragony wskazane przez kryteria wyszukiwania.
<b>Scenariusz główny</b>	<ol style="list-style-type: none"> <li>1. Użytkownik zleca przeszukiwanie paragonów</li> <li>2. System wyświetla możliwe kryteria wyszukiwania: [Rysunek 4.5] <ul style="list-style-type: none"> <li>- daty dodania paragonu</li> <li>- sumy pieniężne na paragonach,</li> <li>- gwarancje paragonów,</li> <li>- znaczki na paragonach.</li> </ul> </li> <li>3. Użytkownik wybiera kryteria wyszukiwania</li> <li>4. System wyszukuje paragony według wskazanych przez użytkownika kryteriów wyszukiwania. [Rysunek 4.6]</li> <li>5. System wyświetla znalezione paragony według kolejności wstawiania. Wyświetlone dane to: nazwa paragonu, znaczki należące do paragonu, długość trwania gwarancji przypisana do paragonu, kwota widniejąca na paragonie, grupa, do której należy paragon oraz miniaturka paragonu. [Rysunek 4.6]</li> <li>6. Zakończenie scenariusza</li> </ol>
<b>Scenariusz alternatywny 1</b>	3a. Użytkownik anuluje akcję wyszukiwania paragonów Powrót do punktu 1. Scenariusza głównego
<b>Scenariusz alternatywny 2</b>	5a System nie znalazł żadnych paragonów pasujących do wskazanych przez użytkownika kryteriów wyszukiwania <ol style="list-style-type: none"> <li>1. System wyświetla komunikat o braku zgadzających się z kryteriami wyszukiwania paragonów [Rysunek 4.6]</li> </ol> Następuje powrót do punktu 2. Scenariusza głównego



Rysunek 4.5 Przeszukaj paragony – kryteria wyszukiwania. Wybór daty dodania paragonu (po lewo), wybór kwot (środek) oraz wybór długości trwania gwarancji (po prawo) [Źródło: opracowanie własne]



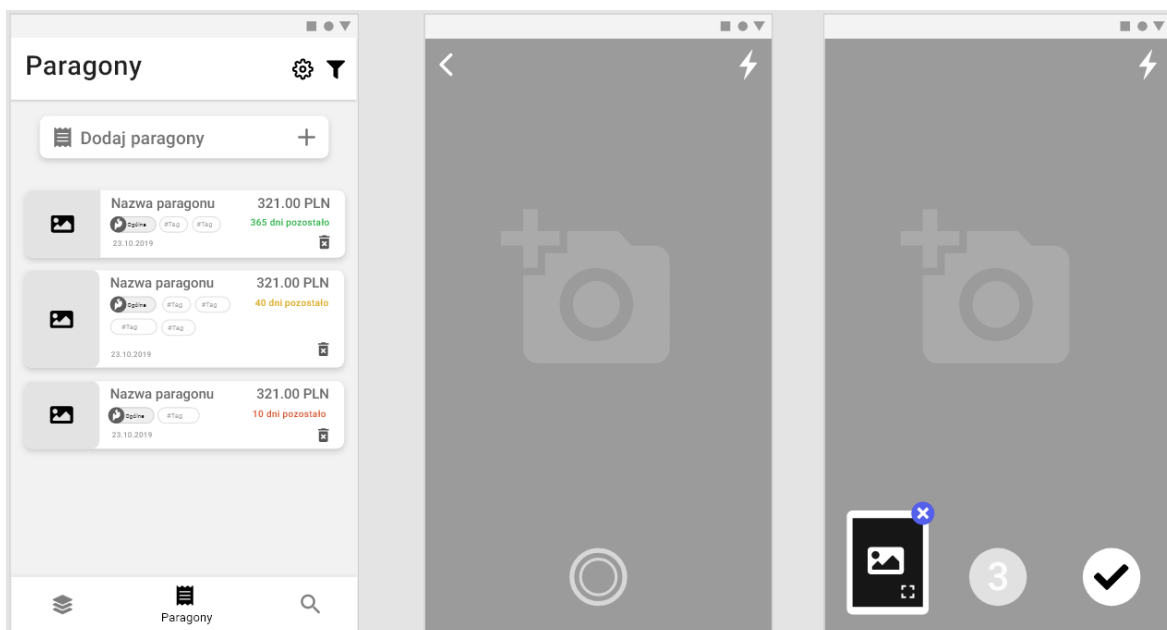
Rysunek 4.6 Przeszukaj paragony – brak kryteriów (po lewo), wypełnienie kryteriów i wynik wyszukiwania (środek), komunikat w przypadku braku znalezienia paragonów dla poszczególnych kryteriów. [Źródło: opracowanie własne]

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.

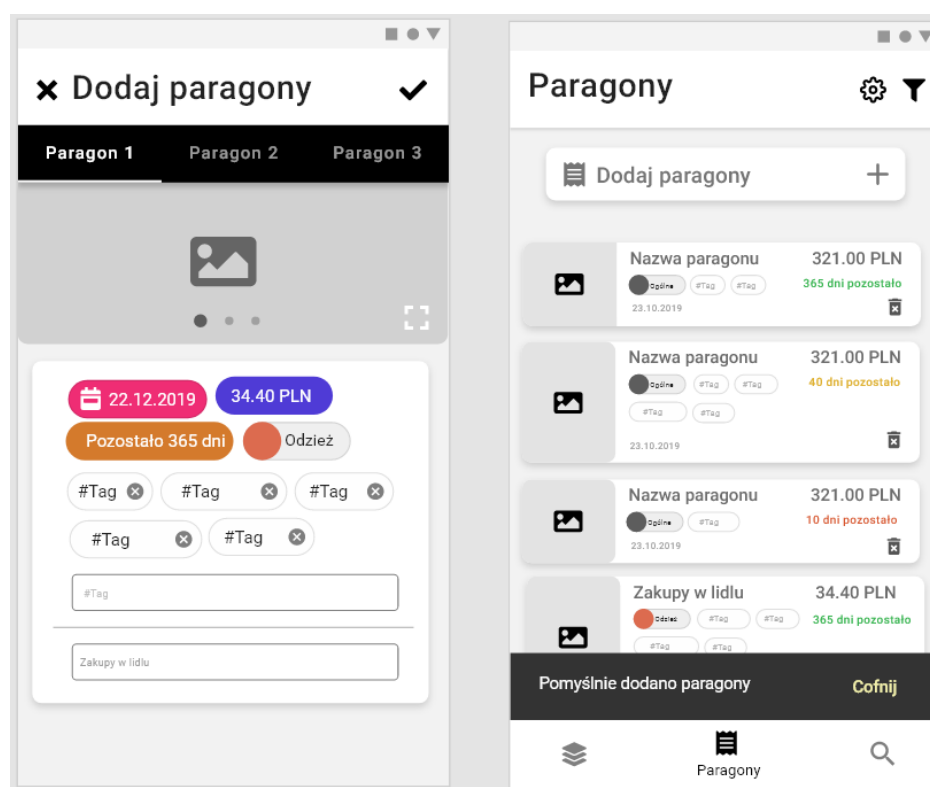
*Tabela 10 Dodaj paragony - Scenariusz*

Nazwa przypadku użycia	Dodaj paragony
<b>Aktorzy</b>	Użytkownik
<b>Warunki wstępne</b>	Użytkownik jest zalogowany, system wyświetla listę paragonów (PU przeglądanie listy paragonów)
<b>Warunki końcowe</b>	Do systemu zostaje dodany jeden lub wiele paragonów
<b>Scenariusz główny</b>	<ol style="list-style-type: none"> <li>1. Użytkownik wybiera akcję dodania nowych paragonów</li> <li>2. System wyświetla ekran podglądu widoku kamery</li> <li>3. Użytkownik zleca wykonanie zdjęcia kamerze.</li> <li>4. System zapamiętuje zdjęcie.</li> <li>5. System wyświetla miniaturkę zdjęcia. (Uwaga: użytkownik może dodać wiele paragonów na raz, akcje 3-5 mogą być powtarzane wielokrotnie).</li> <li>6. Użytkownik zatwierdza zdjęcia paragonów.</li> <li>7. System skanuje zdjęcia w celu wyekstrahowania informacji na temat paragonów.</li> <li>8. System zapamiętuje zinterpretowane na podstawie zeskanowanych zdjęć dane i dodaje je do poszczególnych paragonów.</li> <li>9. System wyświetla ekran z parametrami paragonów (zdjęcia paragonów, daty dodania paragonów, informacje na temat gwarancji paragonów, suma pieniężna, nazwy paragonów oraz znaczki związane z paragonami).</li> <li>10. Użytkownik edytuje parametry.</li> <li>11. Użytkownik zatwierdza dodawanie paragonów.</li> <li>12. System zapamiętuje paragony i wyświetla powiadomienie o udanym ich dodaniu.</li> <li>13. Zakończenie scenariusza.</li> </ol>
<b>Scenariusz alternatywny 1</b>	6a Użytkownik zleca wykonanie kolejnego zdjęcia 1. Następuje punkt 4. Scenariusza głównego
<b>Scenariusz alternatywny 2</b>	6b Użytkownik usuwa ostatnio dodane zdjęcie 1. System usuwa ostatnio dodane zdjęcie 2. Powrót do punktu 6. Scenariusza głównego
<b>Scenariusz alternatywny 3</b>	6c Użytkownik anuluje akcje dodawania paragonów 1. Powrót do punktu 1. Scenariusza głównego
<b>Scenariusz wyjątku</b>	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 4.7 Dodaj paragony – kolejno od lewej krok 0, krok 2 oraz krok 5  
[Źródło: opracowanie własne]



Rysunek 4.8 Dodaj paragony – kolejno od lewej krok 9 oraz krok 12  
[Źródło: opracowanie własne]

## 5. 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.

### 5.1. Opis wybranych wykorzystanych środowisk i narzędzi programistycznych

Będąc przedmiotem projektu aplikacja mobilna przeznaczona jest na urządzenia mobilne z systemem Android. Całość napisana jest w języku **Java** [9] w środowisku programistycznym **Android Studio** [10]. Wykorzystano możliwości narzędzia **Firebase** [11] do zaimplementowania systemu logowania się i rejestracji do aplikacji. Baza danych została zrealizowana za pomocą **Firestore** [11] oraz **Cloud Storage**. **Firestore** stanowi bazę danych, gdzie przechowywane są wszystkie dane potrzebne do prawidłowego funkcjonowania aplikacji. **Cloud Storage** pozwala na przechowywanie plików graficznych (plików, które mają rozmiar większy niż 1 mb) i dostęp do nich. Firebase udostępnia także narzędzie **OCR** [12] (do rozpoznawania tekstu z plików graficznych) o nazwie **Firestore Vision**. Umożliwiło ono ekstrahowanie tekstu ze zdjęć paragonów.

Za pomocą systemu kontroli wersji **Git** [13] utrzymano stabilność i skalowalność projektu.

Do przygotowania wszelkiego rodzaju diagramów zastosowano **Visual Paradigm Online** [14].

Przy użyciu biblioteki **Picasso** [15] zrealizowano dynamiczne pobieranie oraz ładowanie zdjęć zapisanych w **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 zaprojektowane do produkcji aplikacji mobilnych na systemy Android [16] od firmy Google [1]. Android studio zostało opracowane 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 rozproszony 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 [14] 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.

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

Firestore jest mobilną i webową platformą programistyczną udostępniającą wiele produktów związanych z przetwarzaniem danych. Jedną z udostępnianych usług, z której korzysta aplikacja jest Firestore Cloud Firestore [11]. 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 Firestore: Firestore Cloud Storage. Pliki przechowywane są za jej pomocą w folderach i pod folderach. Dostęp do nich jest poprzez adresy url.

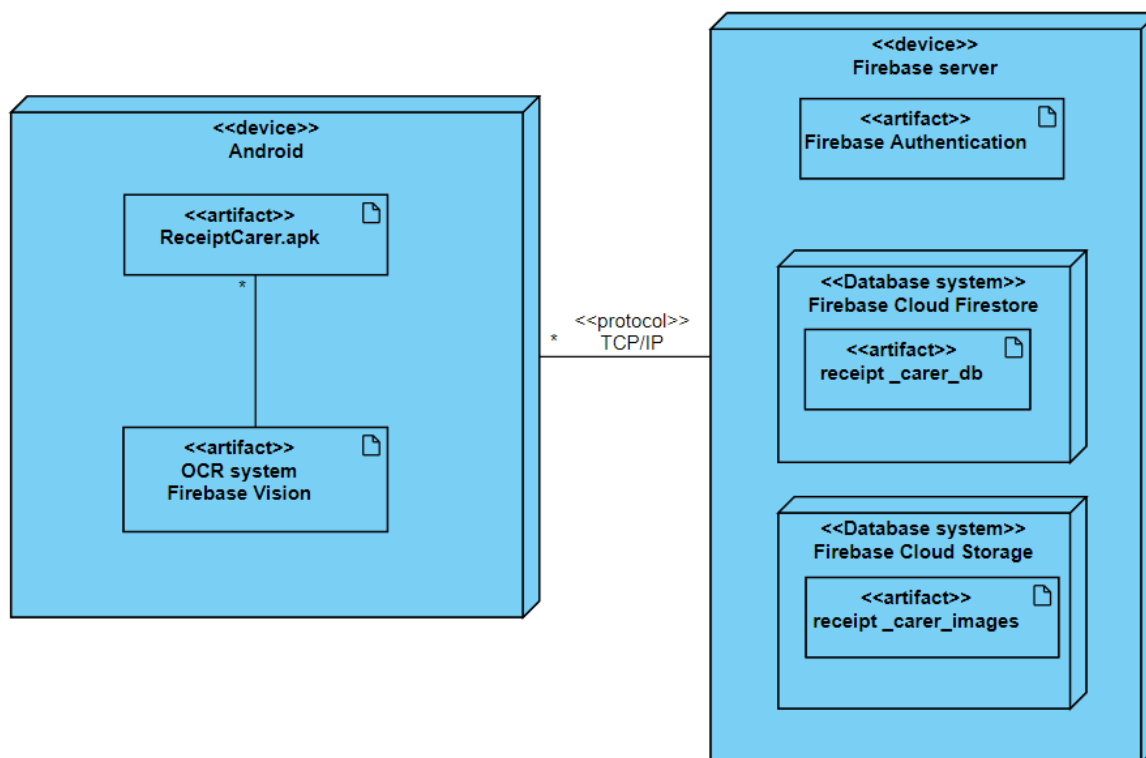
Firestore Authentication jest serwisem oferującym możliwość logowania i rejestracji do systemu. Do skanowania paragonów zostało użyte kolejne narzędzie od Firestore służące do rozpoznawania znaków ze zdjęć o nazwie Firestore Vision. Pozwala na wykorzystanie modułów rozpoznawania tekstu zarówno online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

Firestore 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 implementację. Z tego powodu Firestore i oferowane przez niego usługi zostały wykorzystane w aplikacji.

## 5.2. Architektura rozwiązania

### 5.2.1. Architektura fizyczna

Poszczególne komponenty aplikacji w formie diagramu rozmieszczenia zostały przedstawione na rysunku 5.1.

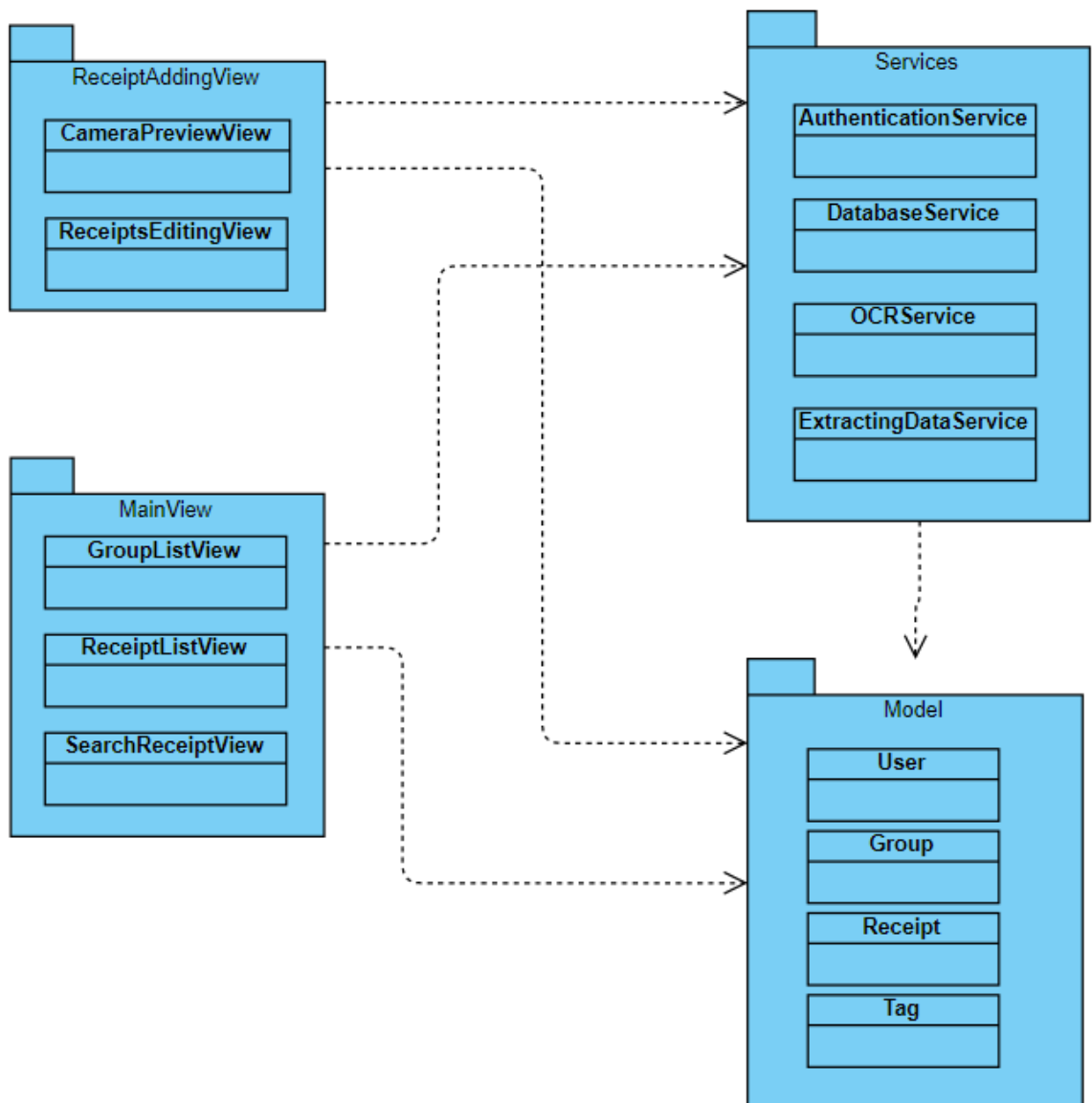


Rysunek 5.1 Diagram rozmieszczenia [Źródło: opracowanie własne]

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

### 5.2.1. Architektura logiczna

Rysunek 5.2 przedstawia diagram pakietów opisujący strukturę zależności w aplikacji.

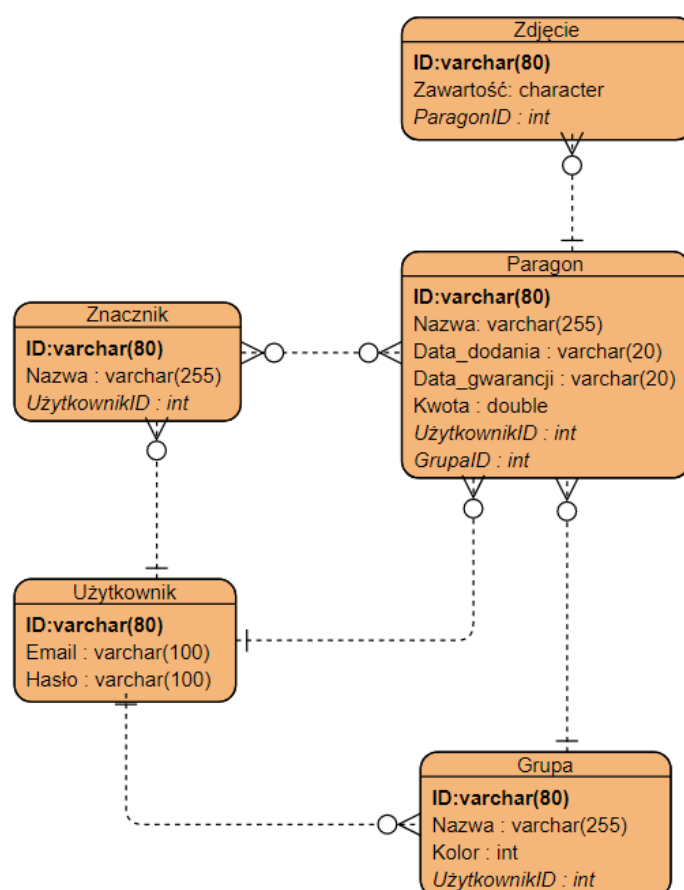


Rysunek 5.2 Diagram pakietów [Źródło: opracowanie własne]

Warstwa prezentacji została podzielona na dwa oddzielne pakiety. Pakiet **MainView** reprezentuje widoki związane z wyświetlaniem list grup i paragonów oraz modułu wyszukiwania. Pakiet **ReceiptAddingView** reprezentuje widoki związane z dodawaniem nowych paragonów do aplikacji. Komunikują się one z serwisami należącymi do pakietu **Services**. **MainView**, **ReceiptAddingView** oraz **Services** wykorzystują klasy należące do pakietu **Model**, czyli klasy modelu reprezentujące warstwę danych w aplikacji.

### 5.3. Projekt Bazy Danych

Aplikacja ReceiptCarer wykorzystuje bazę danych NoSQL [19]. W celu uproszczonego zaprezentowania struktury bazy danych, na rysunku 5.3 zamieszczono diagram ERD (charakterystyczny dla relacyjnych baz danych [20], 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 względu 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.



Rysunek 5.3 Diagram ERD bazy danych [Źródło: opracowanie własne]

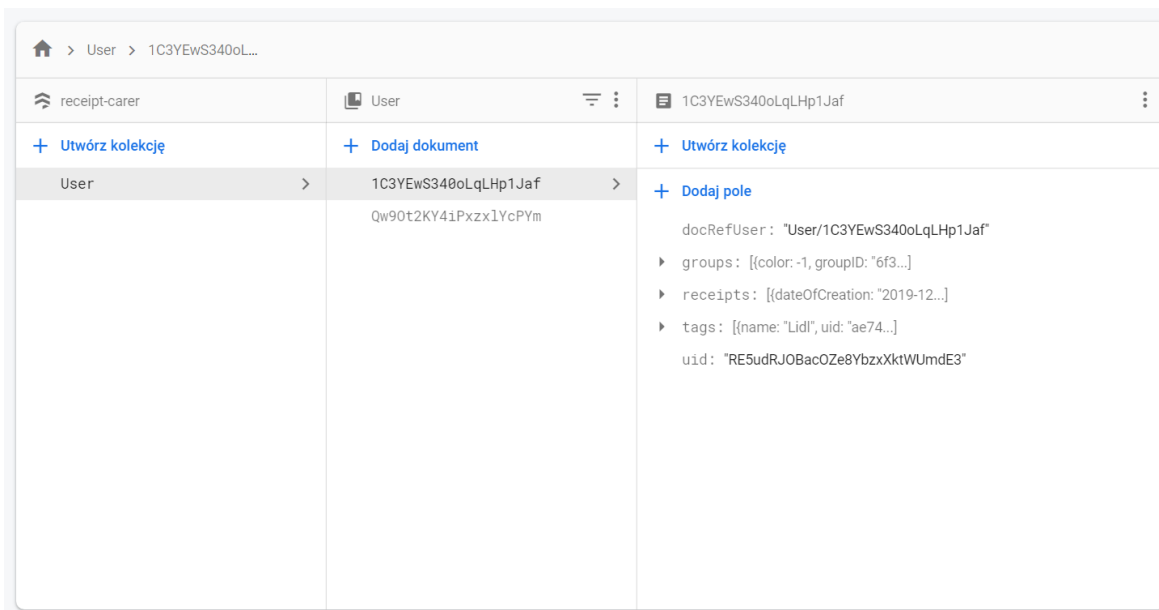
Wykorzystana baza danych pozwala na zagnieżdżanie danych. Pozwala przechowywać obiekty, które mogą mieć pola typu lista i przechowywać wewnątrz wiele elementów określonego typu. Pozwala także na mapowanie obiektów bezpośrednio z bazy

danych na klasy modelu w samej aplikacji – wystarczy, że klasa, na którą chcemy zmapować obiekt pobierany z bazy, posiada pola o tych samych typach i nazwach, co zdefiniowane w bazie danych. Oznacza to, że definiując obiekt w aplikacji, możemy go w tej samej formie przesyłać bezpośrednio na serwer wraz z jego zawartością i analogicznie pobierać dane z bazy. Taka technologia pozwala na projektowanie bazy danych w wygodny sposób - implementowany model danych w aplikacji nie potrzebuje być w żaden sposób tłumaczony przed zapisaniem w bazie danych.










Aplikacja projektowana w pracy wykorzystuje ten mechanizm w następujący sposób - aplikacja posiada klasy modelu o nazwach: *Paragon* (ang: Receipt), *Znacznik* (ang: Tag), *Użytkownik* (ang: User), *Grupa* (ang: Group). Obiekt typu *Użytkownik* posiada identyfikator, definiowany podczas rejestracji i na jego podstawie pobierane są z bazy danych wszystkie dane, które dotyczą tego użytkownika. Użytkownik w bazie przechowywany jest wraz z zawartością, na którą składają się trzy listy następujących typów: *Paragon*, *Grupa*, *Znacznik*. Na rysunku 5.4 przedstawiono strukturę bazy danych odpowiedzialnej za przechowywanie informacji związanych z użytkownikami w celu łatwiejszego zobrazowania jej struktury.

Obiekt typu *Paragon* posiada listę identyfikatorów znaczników oraz listę linków do zdjęć do niego należących. Zdjęcia paragonów przechowywane są w oddzielnej bazie danych Firebase Cloud Storage przeznaczonej na przechowywanie plików. Na rysunku 5.5 przedstawiono widok przechowywanych w bazie danych (przeznaczonej na przechowywanie plików) zdjęć paragonów dostępny dla autora pracy w konsoli Firebase. Dostęp do tych zdjęć jest za pomocą linków URL. *Paragon* posiada także identyfikator grupy do której należy.

Pobierając użytkownika z bazy danych pobierane są wszystkie najważniejsze elementy potrzebne do funkcjonowania aplikacji. Następnie wewnątrz aplikacji każdy identyfikator służy za klucz obcy rozpoznawany już wewnątrz i w trakcie działania aplikacji. Gdy następuje modyfikacja danych wewnątrz aplikacji, są one synchronizowane z bazą danych Firestore. Taka implementacja bardzo ułatwiła zarządzanie bazą danych.



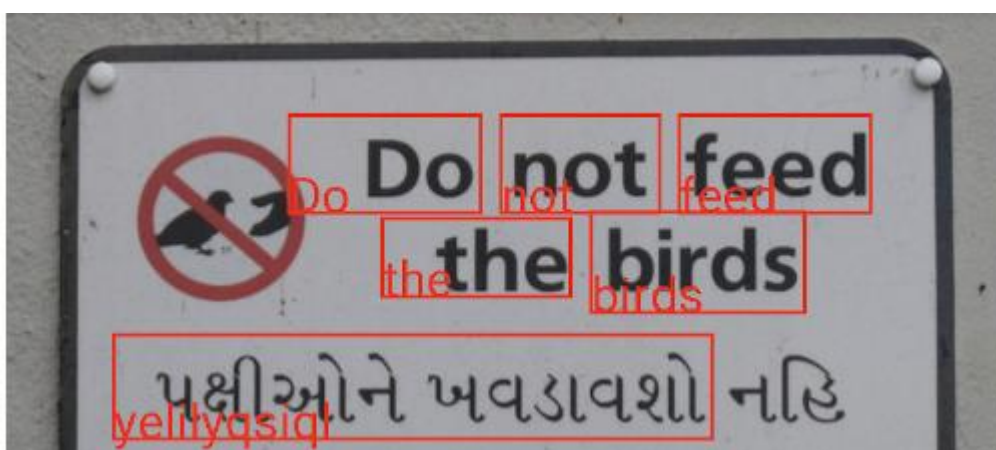
Rysunek 5.4 Widok struktury bazy danych (przeznaczonej do przechowywania informacji na temat użytkowników) w konsoli Firebase dostępnej dla aplikacji projektowanej w pracy.  
[Źródło: opracowanie własne]

gs://receipt-carer.appspot.com > receiptimages				Prześlij plik
<input type="checkbox"/>	Nazwa	Rozmiar	Typ	Ostatnia modyfikacja
<input type="checkbox"/>	 23ce07a5-c18e-4800-9310-3c36efa76db2.png	4.82 MB	image/png	7 gru 2019
<input type="checkbox"/>	 396f3ff1-e7d7-4305-9070-6911a4d43b0b.png	2.98 MB	image/png	7 gru 2019
<input type="checkbox"/>	 3cec6655-9f05-46aa-a1c2-ced898ff29a1.png	2.01 MB	image/png	5 gru 2019
<input type="checkbox"/>	 40a5593a-8c2b-47f3-b284-ef687db92cc3.png	4.44 MB	image/png	5 gru 2019
<input type="checkbox"/>	 4adaeb90-1786-4356-a600-afe9364b8096.png	3.03 MB	image/png	5 gru 2019
<input type="checkbox"/>	 562f8db0-5082-4af9-8b05-6b1846b8182b.png	3.7 MB	image/png	5 gru 2019
<input type="checkbox"/>	 62bd37f2-fef2-47e6-9ab7-d55148a2f352.png	2.44 MB	image/png	5 gru 2019
<input type="checkbox"/>	 66cb63a4-eff5-4969-a6fa-48d1063665eb.png	3.68 MB	image/png	4 gru 2019
<input type="checkbox"/>	 6dd06001-4714-4eb5-a77b-0ce8fb06600c.png	4.84 MB	image/png	7 gru 2019

Rysunek 5.5 Widok przechowywanych w bazie danych (przeznaczonych na przechowywanie plików) zdjęć paragonów w konsoli Firebase dostępnej dla aplikacji projektowanej w pracy.  
[Źródło: opracowanie własne]

## 5.4. Ekstrahowanie informacji z paragonów

Jedną z funkcjonalności, jakie ma posiadać aplikacja, jest możliwość ekstrahowania informacji z paragonów. Rozpoznawanie tekstu przez algorytm OCR od Firebase jest tylko początkiem ekstrahowania danych. Pozwala on na wyodrębnienie tekstu w formie tzw. Bloków, tak jak przedstawia rysunek 5.6. Każdy blok zawiera informacje o tekście, który się znajduje w jego granicach oraz o punktach (o współrzędnych x, y względem zdjęcia, którego dotyczy), oznaczających wierzchołki tych prostokątów. Taki podział pozwala na wykorzystanie przy rozpoznawaniu specyficznych elementów paragonów. Punkty wierzchołków w aplikacji zostały wykorzystane w przypadku rozpoznawania kwoty sumarycznej widniejącej na paragonie.



Rysunek 5.6 Wynik algorytmu OCR od Firebase w formie blozków tekstu [Źródło: [21]]

W przypadku ekstrahowania informacji z paragonów samo skanowanie tekstu nie wystarcza. Nastąpiła więc potrzeba wymyślenia 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 indywidualnego podejścia do ekstrahowania, które będzie omówione w dalszej części pracy.

Dla zwiększenia szansy znalezienia fraz porównywanych w heurystyce, związanych z poszczególnymi danymi z paragonu, każda wartość tekstowa jest zamieniana na małe litery. Dzięki temu usuwany jest problem związany z tym, że formalnie w wykorzystanym języku programowania (Java), dwa wyrażenia są traktowane jako różne teksty pomimo ich oczywistego podobieństwa. Przykładem może być słowo *suma*, które może być różnie zapisane na różnych paragonach („SUMA” lub „Suma”), Dzięki przetworzeniu słowa - zamienieniu wszystkich jego znaków na małe litery, oba słowa prawidłowo zostają uznane za to samo słowo, niezależnie od tego, jaką czcionką były zapisane na paragonie.

Każdy blok tekstu badany jest za pomocą heurystyk:

1. Heurystyka rozpoznawania nazwy sklepu,
2. Heurystyka rozpoznawania daty wydania paragonu,
3. Heurystyka rozpoznawania kwoty sumarycznej na paragonie,
4. Dodatkowa heurystyka

### **Nazwa sklepu**

Heurystyka rozpoznawania nazwy sklepu sprawdza, czy dany blok zawiera jedną z zapisanych w formie listy nazw sklepów. Na potrzeby implementacji i testów w aplikacji brane jest pod uwagę siedem nazw najpopularniejszych sklepów, m.in.: „Lidl”, „Żabka”, „Biedronka”. Trzeba jednak pamiętać, że im dłuższa lista sprawdzanych sklepów, tym więcej przy każdym bloku musi zostać wykonanych obliczeń, zwiększając tym czas trwania ekstrahowania danych, co jest niepożądane w przypadku omawianej aplikacji. Dodatkowo, obserwując paragony (z listy wybranych sklepów), można zauważyć, że nie ma potrzeby badania wszystkich bloków pod kątem nazwy sklepu, jeśli w pierwszych kilkunastu blokach nie została ona znaleziona. Nazwa sklepu znajduje się przeważnie na początku dowodu zakupu. Z tego powodu, jeśli przy 20 bloku nie została odnaleziona nazwa sklepu, heurystyka porzuca jej poszukiwania. Liczba ta została oszacowana na podstawie 15 paragonów z różnych sklepów. Lista sklepów, których nazwy są sprawdzane, jest predefiniowana przez autora pracy (jest stała, a użytkownicy jej nie widzą).

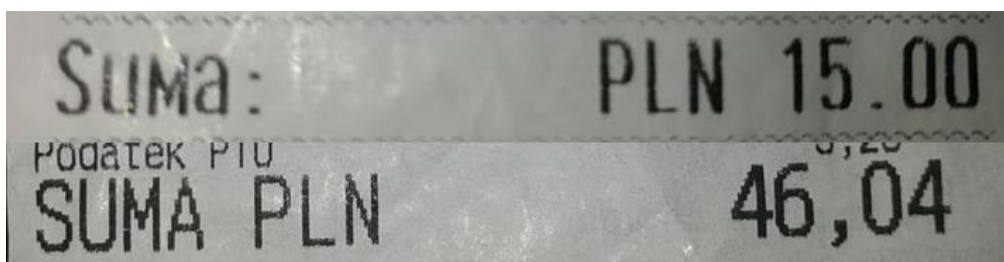
### **Data wydania paragonu**

Data na paragonie może być przedstawiana 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. Sprawdzana jest także sensowność szukanych dat za pomocą wyrażeń regularnych oraz wbudowanych w język Java bibliotek.

### **Cena sumaryczna widniejąca na paragonie**

Największym wyzwaniem okazało się znalezienie kwoty sumarycznej na paragonie. Jest tak ze względu na fakt, że różne sklepy inaczej podchodzą do sprawy zapisu sumy na paragonach. Oczywiście jest, że rozpoznawanie samego tekstu, czy jest liczbą dziesiętną nie jest wystarczające, gdyż na paragonie znajdują się wiele takich liczb (choćby sumy częściowe za poszczególne produkty). Trzeba znaleźć więc punkt odniesienia, który

posłuży za znalezienie kwoty sumarycznej. Dla przykładu na rysunku 5.7 przedstawiono dwie możliwe formy zapisu kwoty sumarycznej na paragonie.



Rysunek 5.7 Przykładowe formy zapisu kwoty sumarycznej na paragonie [Źródło: opracowanie własne]

Wspólnym punktem odniesienia w obu przypadkach przedstawionych na rysunku (jak i biorąc pod uwagę każdy paragon, z którym autor pracy miał styczność) jest słowo „PLN”. Problemem okazuje się fakt, że słowa te są w różnych odległościach od wyszukiwanej kwoty sumarycznej, co wpływa znacząco na to, w jaki sposób algorytm tworzy bloki. W zależności od przypadku, kwota sumaryczna może znajdować się w tym samym bloku, co słowo klucz („PLN”) lub w sąsiedztwie nawet kilku bloków za lub przed blokiem zawierającym punkt odniesienia. Z tego powodu heurystyka w pierwszej kolejności, jeśli znajdzie blok, który zawiera frazę klucz, przeszukuje ten blok, czy nie zawiera liczby dziesiętnej. Kolejnym krokiem w przypadku nieznalezienia, jest zebranie bloków znajdujących się w najbliższym sąsiedztwie bloku frazy klucz. W tym przypadku nie wystarczy tylko przeszukać wszystkich sąsiednich bloków pod kątem wystąpienia liczb dziesiętnych, ponieważ w okolicach frazy kluczowej może być takich kilka. Pomocnym staje się dodatkowa informacja niesiona przez bloki tekstu, jaką są punkty wierzchołków prostokątów wyznaczonych przez algorytm rozpoznawania tekstu. Można na ich podstawie sprawdzić, czy sąsiednie bloki znajdują się na tej samej linii, co blok, zawierający frazę „PLN”. Heurystyka wymyślona na potrzeby aplikacji sprawdza więc, czy jeden z wybranych wierzchołków w stosunku do innych bloków jest mniej więcej na tej samej linii. Sprawdzana jest mianowicie współrzędna „y” punktów znajdujących się w lewym górnym rogu prostokątów bloków. Ze względu na fakt, że bloki nie muszą znajdować się w tej samej linii, sprawdzane jest, czy znajdują się w położeniu bliskim tej linii (czyli uwzględniając lekkie odchylenia spowodowane obróceniem paragonu na zdjęciu, np. gdy zostało nieco krzywo wykonane), zakładając przy tym, że mogą być odsunięte o kilka pikseli względem siebie. Wartością, która wykazała największą skuteczność bez popełniania błędów, było odchylenie 50-sięciu pikseli względem współrzędnej y. Taka wartość pozwala na obsłużenie drobnych przekrzywień zdjęć paragonów.

W przypadku badań tego typu, czyli czy bloki znajdują się na jednej linii, ważnym jest by zdjęcie które jest wykonywane było w miarę prosto zrobione (by paragon nie był obrócony). Na tym etapie można by zaimplementować algorytmy, pozwalające na naprawianie zaburzeń związanych z ułożeniem paragonu. Przekształcałyby oryginalne zdjęcie paragonu dla lepszej jego czytelności przez heurystyki. Algorytmy wykorzystywane do rozpoznawania zdjęcia doskonale sobie radzą z tego typu zaburzeniami, lecz w przypadku ekstrakowania informacji z paragonów autor pracy zdecydował się uprościć implementowane heurystyki, by nie zwiększać ich złożoności działania. Pozwoliło to na możliwość wprowadzenia większej bazy sklepów lub formatów dat. Aplikacja z założenia ma być szybka i zakłada się, że zdjęcia wykonywane przez użytkownika nie będą znacząco zniekształcone.

### **Dodatkowa heurystyka**

Istnieje możliwość, że wyszukiwane słowa klucz, takie jak porównywane nazwy sklepów ze znalezionymi przez algorytm blokami tekstów, mogą być nierozpoznane przez algorytmy OCR w sposób zgodny z rzeczywistością. Przykładem może być nazwa sklepu „Lidl” gdzie częstym przypadkiem rozpoznawania przez Firestore Vision było nie słowo „Lidl” lecz słowo „Lid1” (z zamienioną jedną literką na cyfrę „1”), ze względu na podobieństwo znaków. Jest to częsty przypadek i całkowicie zrozumiały. Nastąpiła więc konieczność ochrony przez tego typu błędami. Wykorzystano do tego miarę odmienności napisów (skończonych ciągów znaków) o nazwie *Odległość Levenshteina* [22]. Odległością między dwoma napisami nazywamy w przypadku tego algorytmu najmniejszą liczbę działań prostych, przekształcających jeden napis w drugi. Do działań prostych zaliczamy: wstawienie nowego znaku do napisu, usunięcie znaku z napisu, zamianę znaku w napisie na inny znak.

W heurystykach miara wykorzystywana jest do wyznaczania jak dużo (w procentach) potrzeba zmienić dłuższy ciąg znaków, by zamienić 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 dopuszczalne jest zmienienie jej jednej litery.

## 6. Implementacja aplikacji ReceiptCarer

W niniejszym rozdziale przedstawiono prezentację zaimplementowanego rozwiązania, opisując poszczególne funkcjonalności aplikacji. Aplikacja nazywa się ReceiptCarer (z ang. Opiekun Paragonów).

### 6.1. Rozwiązanie problemu synchronizacji zdjęć

Głównym walorem aplikacji jest możliwość przechowywania zdjęć paragonów. Pojawia się jednak problem związany z synchronizacją zdjęć z zewnętrzną bazą danych w czasie rzeczywistym. Pobierając wszystkie potrzebne dane z bazy danych, przy każdej synchronizacji aplikacji z bazą, wraz ze wzrostem liczby paragonów dla danego użytkownika, pojawia się szybko problem wynikający z dużego rozmiaru plików graficznych. Jednym z rozwiązań i zarazem założeń bazy danych w czasie projektowania było zastosowanie podziału na dwie osobne bazy - bazę danych do przechowywania informacji na temat użytkowników, paragonów, grup oraz znaczników i bazę danych służącą do przechowywania samych zdjęć. Każdy paragon powinien posiadać linki URL do zapisanych zdjęć w bazie znajdującej się na serwerze Firebase. Taki podział został zaimplementowany i pozwolił na pobieranie zdjęć paragonów o dużym rozmiarze tylko, gdy zachodzi taka potrzeba. Synchronizacja ta została zapewniona dzięki darmowej bibliotece dla aplikacji na systemy android – bibliotece *Picasso* [15]. Dostarczając jedynie linki URL, zapisane dla każdego z paragonów, pobiera ona w tle potrzebne zdjęcia, a następnie je zapamiętuje i wykorzystuje w taki sposób, by zminimalizować wypełnianie się łącza internetowego. Zdjęcia są wczytywane dopiero w momencie gdy zostaje wyświetlony podgląd paragonu. W aplikacji istnieje lista paragonów posiadająca miniaturki pierwszych zdjęć każdego paragonu. Dzięki bibliotece *Picasso* nie jest pobierane pełne zdjęcie z bazy danych lecz tylko jego zdecydowanie mniejsza wersja. Nie ma więc potrzeby na tworzenie miniaturk ręcznie.

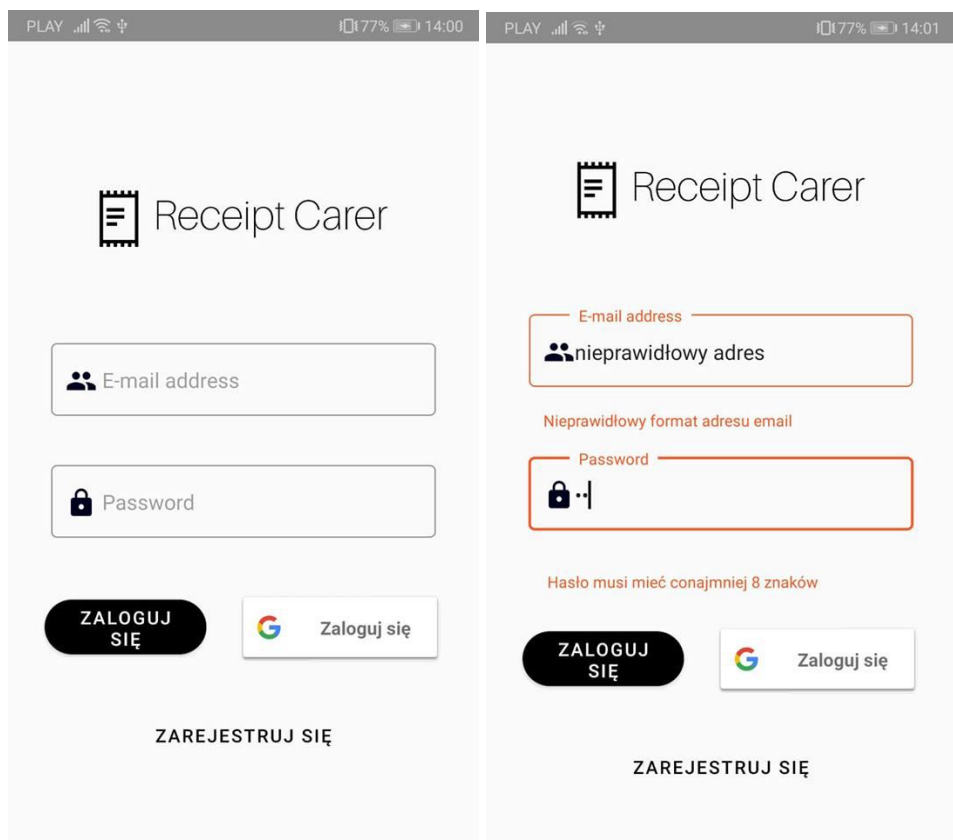
Sama baza danych ze wszystkimi informacjami na temat paragonów, grup i znaczników jest bardzo mała, nawet dla bardzo dużej liczby paragonów i tylko ona jest wczytywana podczas synchronizacji. Mechanizmy zastosowane w aplikacji zapewniają jej szybkie działanie nawet w przypadku bardzo wielu paragonów.

### 6.2. Logowanie

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

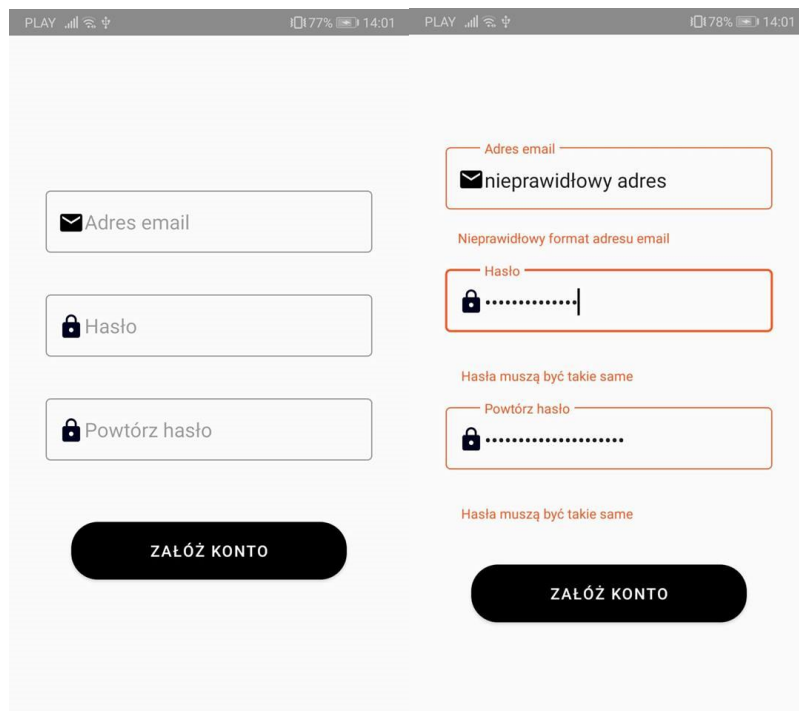
Na samym dole ekranu widnieje przycisk, za pomocą którego użytkownik ma możliwość zarejestrowania się (poprzez wciśnięcie przycisku *Zarejestruj się*). Przenosi on użytkownika do ekranu rejestracji (Rysunek 6.2) opisanego szczegółowo w dalszej części pracy.



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

### 6.3. Rejestracja

Po naciśnięciu przycisku *Zarejestruj się*, użytkownik zostaje przeniesiony do ekranu rejestracji. Użytkownik może 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, aby móc się zarejestrować w systemie. 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).



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

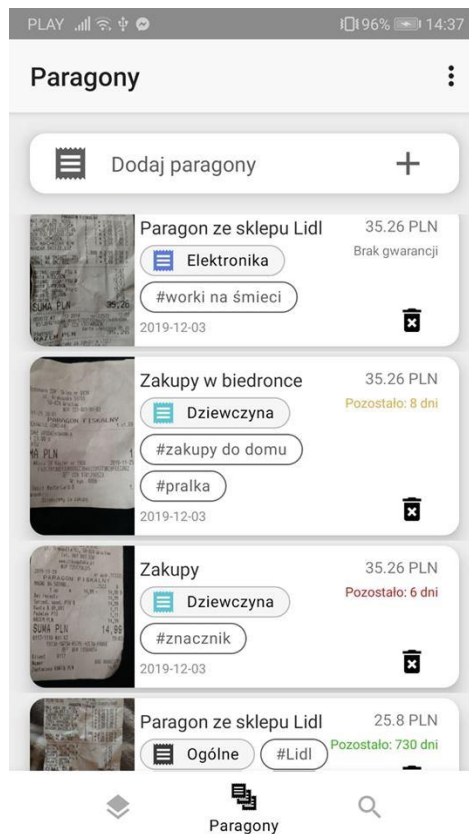
## 6.4. Lista paragonów

Po zalogowaniu do aplikacji ekranem, który widzi użytkownik jest ekran listy paragonów, które dodał do aplikacji. Znajdujące się na ekranie elementy to lista paragonów, przycisk dodawania paragonów oraz przycisk znajdujący się na górnym pasku, którego kliknięcie wyświetla przycisk umożliwiający wylogowanie użytkownika z aplikacji. 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,
- znaczki przypisane do paragonu.

W zależności od liczby dni pozostałych do końca trwania gwarancji informacja o niej jest uwydatniana w inny sposób. Gdy liczba dni jest mniejsza niż 7 dni to wyświetla się ta informacja 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, którym z dodanych paragonów (czyli którym produktom) kończy się ważność gwarancji.

Użytkownik może kliknąć w element listy i zostanie wtedy przeniesiony do szczegółów danego paragonu wraz z możliwością jego edytowania i eksportowania. Użytkownik ma także możliwość usunięcia paragonu klikając w ikonę kosza znajdującą się w prawym dolnym rogu każdego elementu listy paragonów. Opisywany ekran przedstawiono na rysunku 6.3.



Rysunek 6.3 Ekran listy paragonów [Źródło: opracowanie własne]

## 6.5. Paragon – widok szczegółowy i edycja

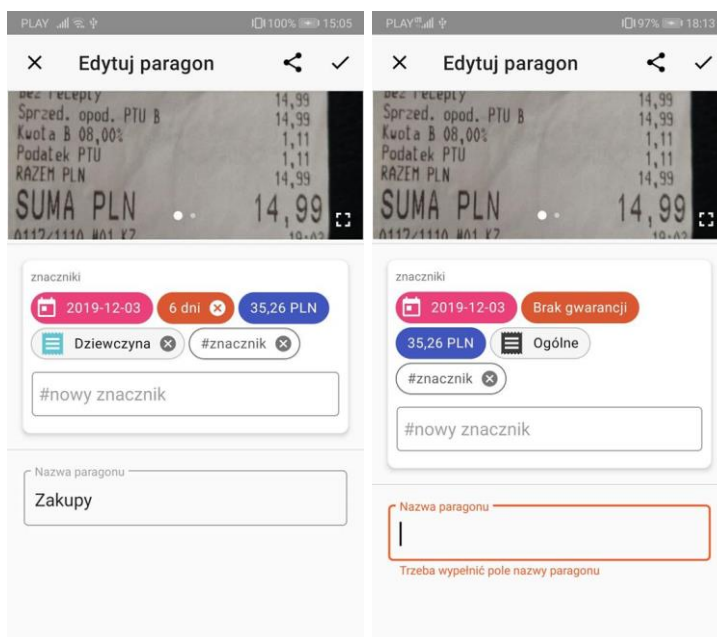
Po kliknięciu w dany element na liście paragonów użytkownik zostaje przeniesiony do ekranu edytowania paragonu (Rysunek 6.4). Na tym ekranie znajdują się te same informacje co na liście paragonów (przedstawione w poprzednim podrozdziale) z tą różnicą, że poszczególne elementy mogą być edytowane. Na ekranie możliwy jest również podgląd 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 tzw. specjalne typy znaczników, które charakteryzują się innym wyglądem i funkcjonalnościami. Do specjalnych znaczników należą:

1. Znacznik specjalny wyboru daty (kolor różowy z ikoną kalendarza po lewej stronie) - pozwala na wybranie daty dodania paragonów.
2. Znacznik specjalny wyboru długości trwania gwarancji (kolor pomarańczowy) - pozwala na wybór długości trwania gwarancji w dniach, tygodniach, miesiącach lub latach.
3. Znacznik specjalny wyboru kwoty pieniężnej (kolor niebieski) – pozwala na wpisanie kwoty sumarycznej widniejącej na paragonie.
4. Znacznik specjalny wyboru grupy (kolor szary z ikoną paragonu po lewej stronie) – pozwala na wybór grupy, do której ma należeć paragon.

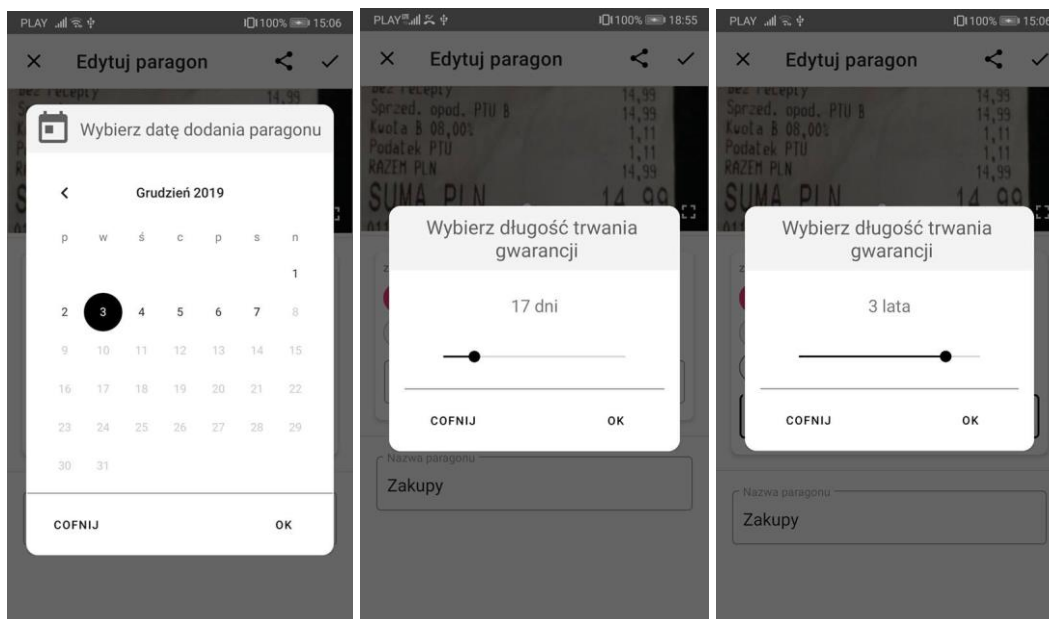
W każdy znacznik specjalny można kliknąć i wykonać związane z nimi możliwe akcje. System taki został wybrany ze względu na uproszczenie widoku edycji paragonu. Po kliknięciu w dany znacznik pojawia się dodatkowy ekran z odpowiednim tytułem,

pozwalający na wykonanie, zaakceptowanie lub odrzucenie danej akcji związanej ze znacznikami.



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

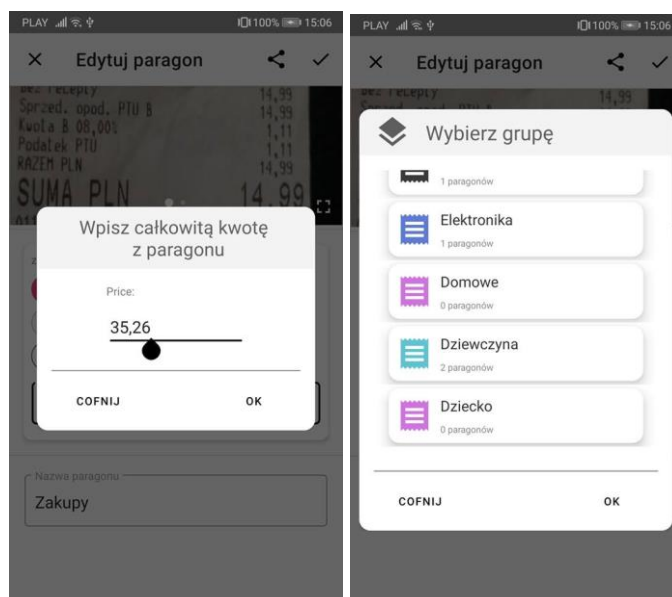
Znacznik wyboru daty ukazuje ekran kalendarza, który pozwala na wybranie dowolnej daty wcześniejszej od dnia, w którym następuje edycja (Rysunek 6.5). Znacznik wyboru długości trwania gwarancji pozwala na wybór za pomocą paska (którego wartość można ustawiać, przesuwając po nim) długości trwania gwarancji (Rysunek 6.5). W zależności od pozycji ustawionej na pasku gwarancja może być wyrażona w dniach, tygodniach, miesiącach lub latach. Rysunek 6.5 prezentuje omówione funkcjonalności.



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

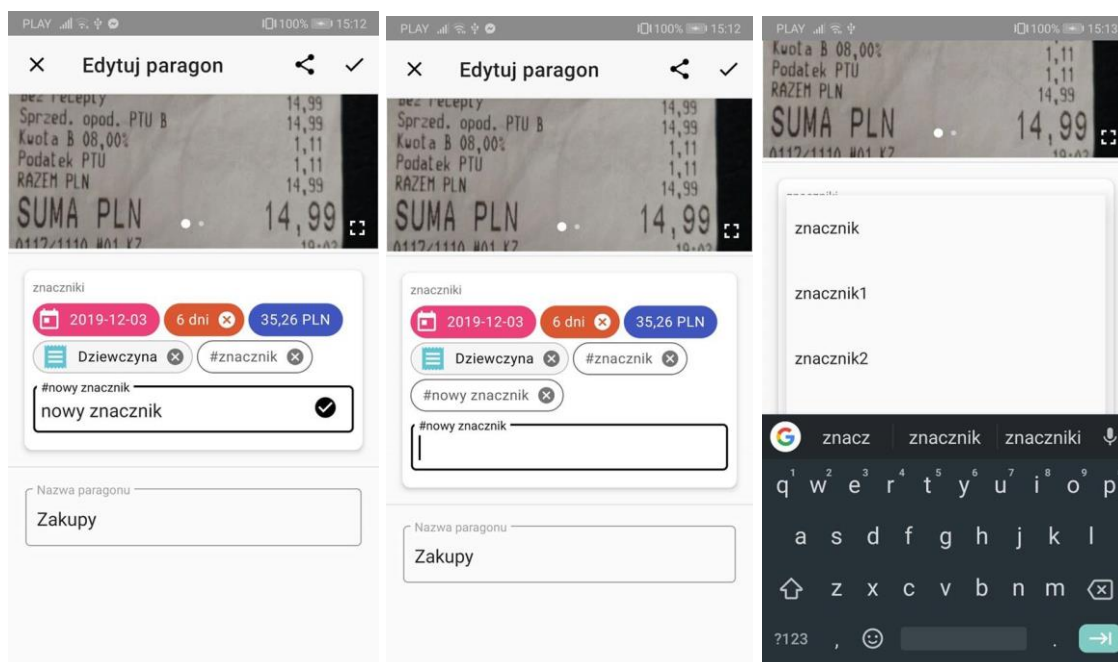


W przypadku znacznika wyboru całkowitej kwoty z paragonu, ekran zawiera jedynie pole kwoty (Rysunek 6.6), które można edytować. Znacznik dotyczący wyboru grupy pozwala na wybranie grupy i przypisanie jej do paragonu (Rysunek 6.6). W przypadku nie wybrania żadnej grupy zostaje on przypisany do grupy ogólnej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



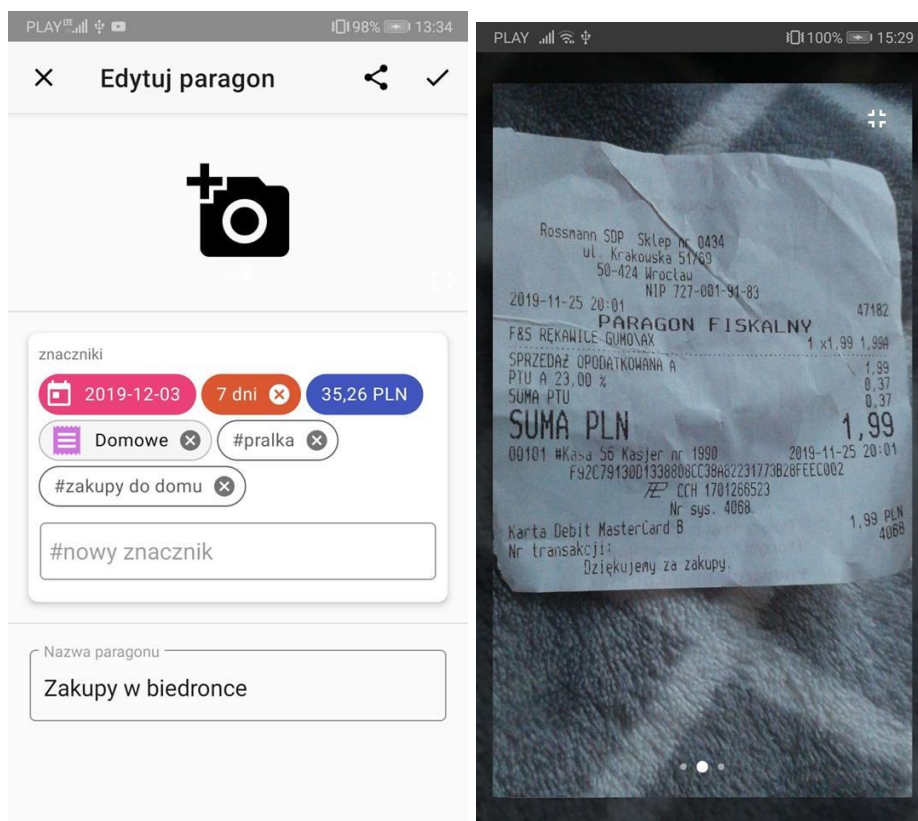
*Rysunek 6.6 Ekrany specjalne: wybór całkowitej kwoty z paragonu (po lewej), wybór grupy do której należy paragon (po prawej) [Źródło: opracowanie własne]*

Kolejną funkcjonalnością związaną z paragonami jest zarządzanie tzw. 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 listy 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 jego prawej stronie. Istnieje także możliwość, by użytkownik wybrał znacznik z listy już istniejących znaczników. Ukazane jest to na rysunku 6.8. Przy wpisywaniu frazy do pola nowego znacznika pojawia się lista znaczników znajdujących się już w całej aplikacji, które zawierają wpisywaną frazę. Pozwala to na lepsze i bardziej sprawne porządkowanie znaczników.



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

Kolejną możliwością związaną z edycją paragonu, jest możliwość dodania więcej zdjęć do paragonu oraz przeglądanie zdjęć paragonów. Dostęp do dodawania nowych zdjęć uzyskuje się poprzez przesunięcie do końca w prawo podglądu zdjęć paragonu. Klikając w ikonę aparatu z plusem następuje przekierowanie do ekranu dodawania zdjęć. Sposób dodawania zdjęć został opisany szczegółowo w dalszej części pracy przy opisie dodawania paragonów do aplikacji. Po kliknięciu w dowolną miniaturkę na przeglądzie zdjęć na ekranie edytowania paragonu, wyświetlany jest podgląd zdjęć pozwalający na usuwanie paragonów (przy czym musi zostać co najmniej jedno zdjęcie paragonu).

## 6.6. Dodawanie nowych paragonów

Z ekranu listy paragonów klikając w przycisk dodawania paragonów, użytkownik zostaje przeniesiony do ekranu wykonywania zdjęć. Ekran ten został przedstawiony na rysunku 6.9. Ten sam ekran wraz z działaniem przedstawiany jest przy dodawaniu kolejnych zdjęć do paragonów podczas ich edycji.



Rysunek 6.9 ekran dodawania zdjęć – brak wykonanych zdjęć (po lewej), wykonano jedno zdjęcie, wykonano 2 zdjęcia (środek oraz po prawo) [Źródło: opracowanie własne]

Przez cały czas wyświetlania tego ekranu widoczny jest podgląd aparatu tylnego telefonu. Nie istnieje możliwość wykonywania zdjęcia aparatem przednim. W lewym górnym rogu znajduje się przycisk anulowania akcji. Przeciśnięcie go skutkuje przeniesieniem do poprzedniego ekranu używanego przez użytkownika. W prawym górnym rogu znajduje się możliwość wybrania trybu działania światła flesz pod warunkiem, że urządzenie na którym jest zainstalowana aplikacja posiada taką możliwość. W przeciwnym wypadku guzik ten jest niewidoczny. Aplikacja obsługuje 3 tryby działania światła flesz: włączone, wyłączone lub automatyczne. Użytkownik klikając w ikonę światła, przełącza je na kolejny stan.

Na dole, na środku znajduje się przycisk wykonywania zdjęcia. Po jego wciśnięciu w lewym dolnym rogu pojawia 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 przedstawiony został na rysunku 6.9. Jeśli zostało wykonane co najmniej jedno zdjęcie, w prawym dolnym rogu pojawia się przycisk umożliwiający akceptację zdjęć i przełączenie do kolejnego ekranu. W przypadku gdy akcja wyświetlająca ekran wykonywania zdjęć, to dodawanie paragonów, użytkownik zostaje przeniesiony do ekranu przedstawionego na rysunku 6.10.

Po zatwierdzeniu zdjęć paragonów następuje skanowanie paragonów przy wykorzystaniu systemu OCR oferowanego przez Firebase wbudowanego w aplikację. Po zeskanowaniu, z wykorzystaniem 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 6.10 przedstawia ekran edytowania paragonów. Można przełączać się pomiędzy paragonami za pomocą zakładek na górnym pasku (*Paragon 1, Paragon 2 itd.*).

The screenshot shows a mobile application interface for editing receipts. At the top, there's a status bar with 'PLAY' logo, signal strength, and battery level at 97%. Below the status bar is a header with a close button (X), the title 'Paragony', and a checkmark button. Under the header are three tabs: 'PARAGON 1', 'PARAGON 2' (which is selected), and 'PARAGON 3'. The main content area displays a receipt image with the following items and prices:

Item	Price
CHLEB (KROJONY) NOWETA C 1*6.00	6.00C
BULKA MASLANA / KRUSZONKA B 2*2.00	4.00B
MINI CHAJKA 90g JULKA B 2*2.50	5.00B
SP. OP. B: 9.00 PTO 8%	0.67

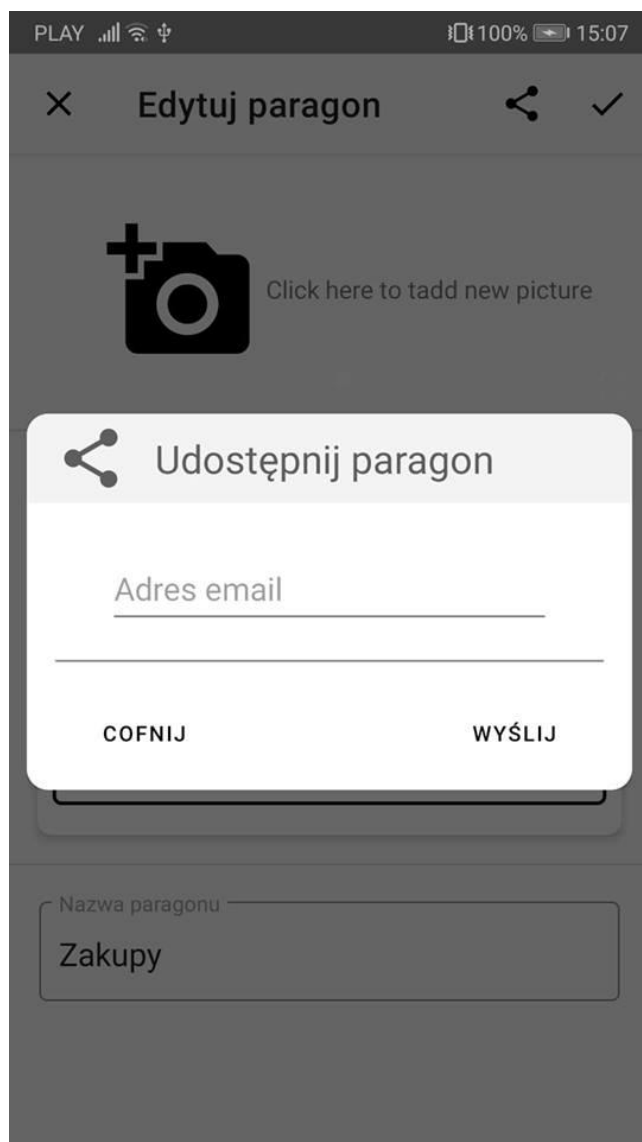
Below the receipt image is a section titled 'znaczniki' (tags) containing several buttons: a date button '2019-11-28', a button 'Brak gwarancji', a price button '15,00 PLN', and a button 'Ogólne'. There is also a text input field with the placeholder '#nowy znacznik'. At the bottom, there is a text input field labeled 'Nazwa paragonu' with the text 'Paragon z apteki' entered.

Rysunek 6.10 Ekran edycji dodawanych paragonów [Źródło: opracowanie własne]

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

## 6.7. Eksportowanie paragonów

Z ekranu edycji paragonu można eksportować paragon, wysyłając go na podany adres e-mail. Służy do tego przycisk znajdujący się po lewej stronie od przycisku akceptacji zmian paragonu. Na rysunku 6.11 przedstawiono ekran eksportowania. Po wpisaniu przez użytkownika i zweryfikowaniu przez system poprawności podanego adresu e-mail, następuje odesłanie do zewnętrznej aplikacji zajmującej się wysyłką wiadomości e-mail na telefonie użytkownika.

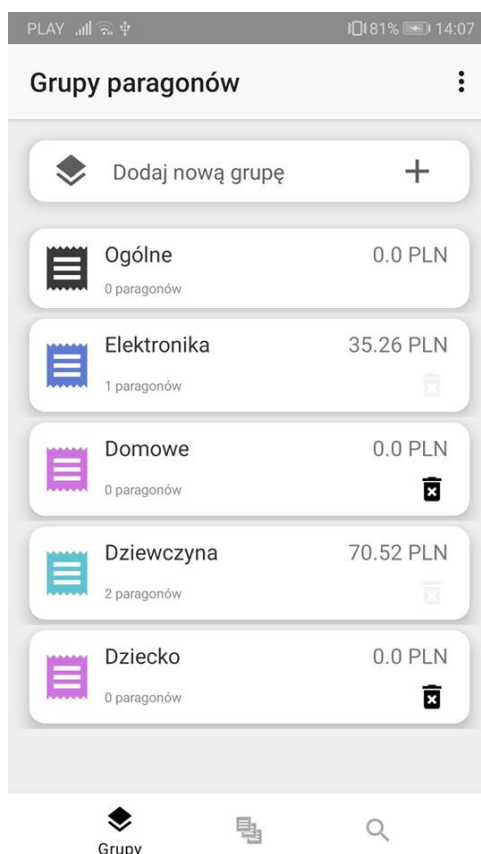


Rysunek 6.11 Eksport paragonu [Źródło: opracowanie własne]

## 6.8. Lista grup paragonów

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 (przedstawione w dalszej części pracy). 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.



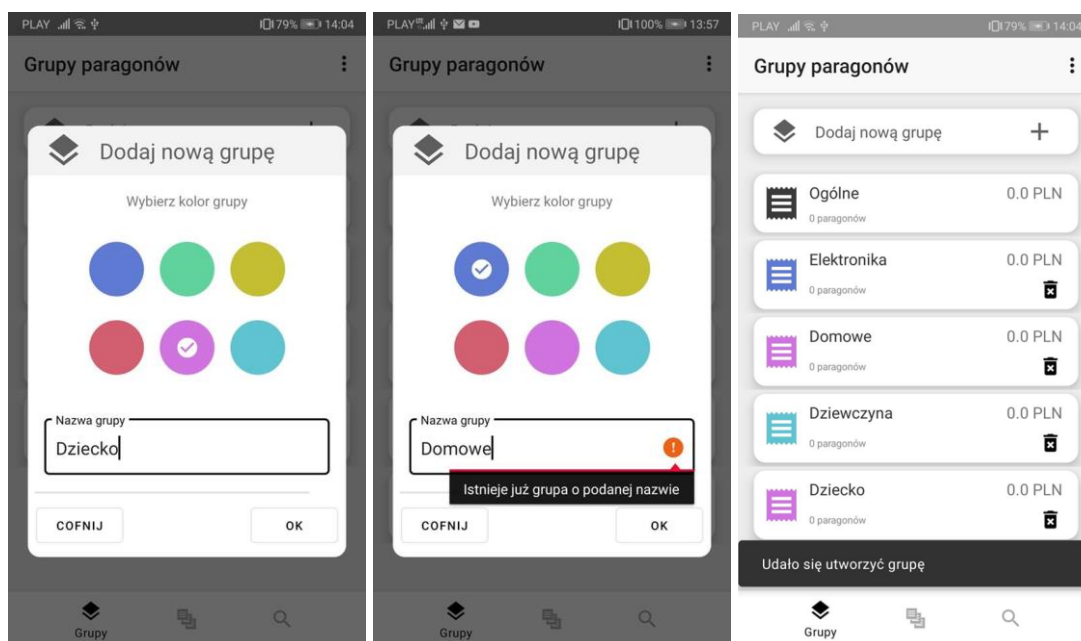
Rysunek 6.12 Ekran z listą grup paragonów [Źródło: opracowanie własne]

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 sytuacji istnienia paragonów, które nie należą do żadnej grupy (jest to niedozwolone w aplikacji). Klikając w dowolny element różny od wcześniej wskazanego, użytkownik przenoszony jest do ekranu edycji grupy opisaną szczegółowo w dalszej części pracy.



## 6.9. Dodawanie nowej grupy

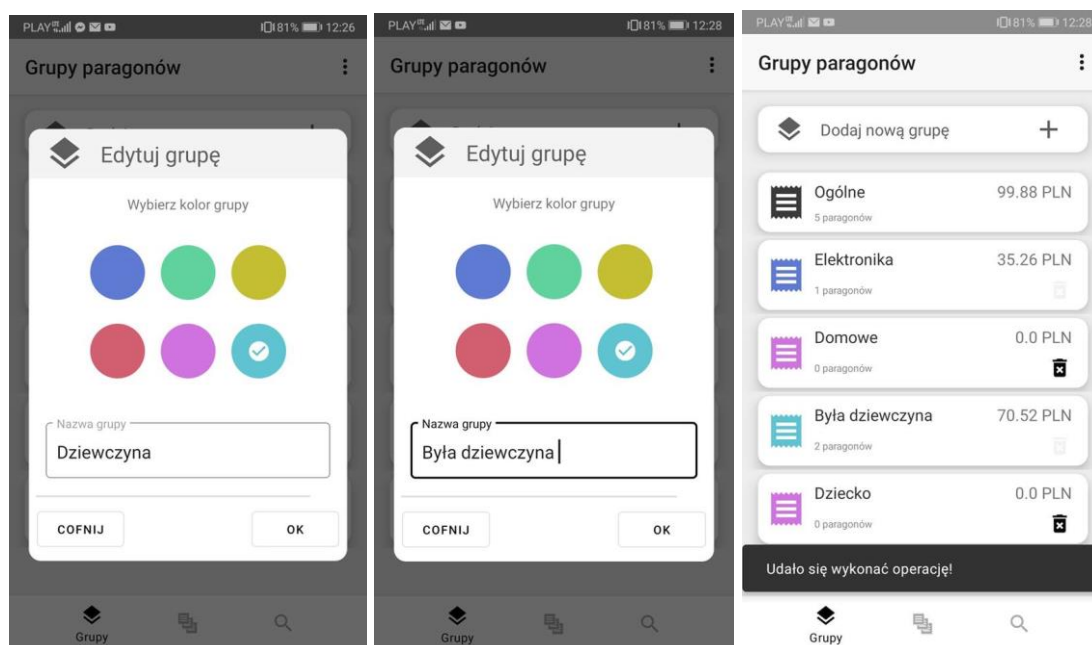
Rysunek 6.13 przedstawia ekran, który pojawia się po kliknięciu w przycisk dodania nowej grupy paragonów. Użytkownik może wybrać jeden z sześciu dostępnych kolorów dla grupy oraz wprowadzić nazwę dla nowej grupy. Nazwa grupy musi być unikalna, inaczej użytkownik po zatwierdzeniu dodawania nowej grupy zostanie poinformowany o braku możliwości dodania ze względu na powtórzoną nazwę grupy. Po zatwierdzeniu i poprawnym zweryfikowaniu nazwy, grupa zostaje dodana do aplikacji i synchronizowana z bazą danych. Na ekranie pojawia się informacja o poprawności operacji utworzenia nowej grupy paragonów.



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

## 6.10. Edycja grupy

Rysunek 6.14 przedstawia ekran edycji paragonów. Użytkownik po kliknięciu w odpowiednią grupę zostaje przeniesiony do ekranu identycznego z ekranem dodawania nowej grupy (Rysunek 6.13), z różnicą w tytule znajdującym się na górze ekranu, oraz elementy są już wypełnione.

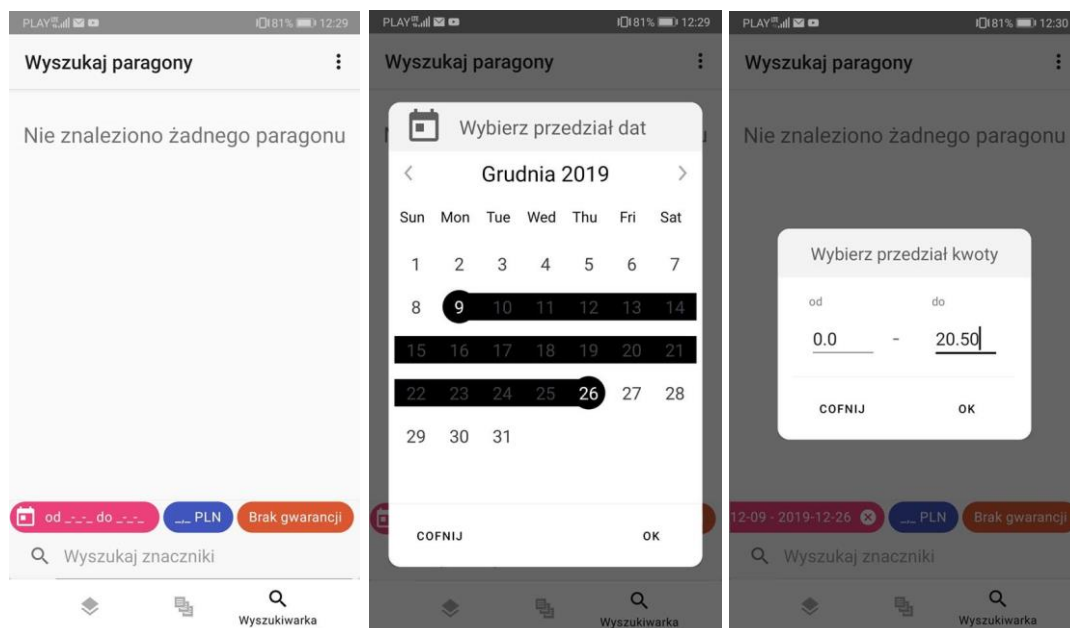


Rysunek 6.14 Ekrany edycji grupy [Źródło: opracowanie własne]

## 6.11. Wyszukiwanie paragonów

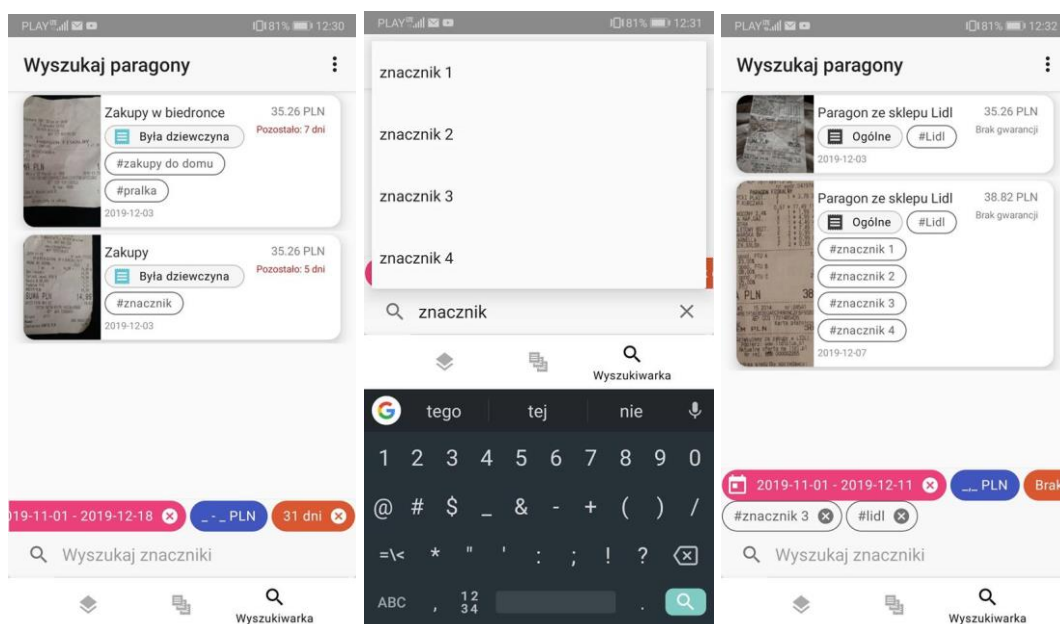
Rysunek 6.15 przedstawia ekran wyszukiwania paragonów oraz ekrany wyświetlające się po kliknięciu w wybrany znacznik specjalny, dotyczący daty wydania paragonu lub znacznik kwoty widniejącej na paragonie. Znaczniki są podobne do tych widniejących na ekranach dodawania i edycji paragonów, z tym, że te pozwalają na wybieranie przedziałów, zarówno daty od - do oraz kwoty od - do. Ponadto można wybrać także długość trwania gwarancji według której mają być wyszukiwane paragony. Na dole pod znacznikami specjalnymi znajduje się także pole umożliwiające wyszukiwanie znaczników po ich nazwach. Do znaczników należą także nazwy grup oraz nazwy paragonów. Każdorazowa zmiana kryteriów wyszukiwania powoduje wyszukanie poszczególnych elementów i pojawienie się ich w formie listy paragonów spełniającej podane kryteria (lub odpowiedniej informacji jeśli nie zostały znalezione żadne paragony o podanych kryteriach).





Rysunek 6.15 Wyszukiwanie paragonów – brak kryteriów wyszukiwania (po lewo), ekran znacznika specjalnego wyboru przedziału dat (środek), ekran znacznika specjalnego wyboru przedziału kwoty pieniężnej (po prawo) [Źródło: opracowanie własne]

Wyświetlane paragony przedstawiają takie same informacje, co przedstawiane są na liście paragonów opisanej w poprzednich podrozdziałach, poza możliwością usunięcia paragonu. Po kliknięciu w paragon wyświetla się jego podgląd w taki sam sposób, jak w przypadku kliknięcia na element listy na ekranie przedstawiającym listę paragonów. Na rysunku 6.16 pokazane są także możliwości związane z dodawaniem znaczników jako kryteria wyszukiwania. Wyszukiwane są wszystkie paragony, które zawierają chociaż jeden z wybranych przez użytkownika znaczników.



Rysunek 6.16 Wyszukiwanie paragonów – wynik wyszukiwania dla podanych kryteriów [Źródło: opracowanie własne]

## 7. Testy

### 7.1. Testy jednostkowe

Do przetestowania aplikacji zostały wykonane testy jednostkowe za pomocą JUnit 4 [23]. Na potrzeby tych testów zaprojektowana została klasa *ValidatorTest* sprawdzająca poprawność funkcji walidujących parametry podawane przez użytkownika, takie jak: nazwa grupy, nazwa paragonu czy hasło przy logowaniu się do aplikacji. Przed rozpoczęciem testów inicjowane są grupy paragonów o podanych parametrach jak przedstawiono na rysunku 7.1. Mają one za zadanie imitację prawdziwych globalnych grup znajdujących się na telefonie podczas logowania użytkownika.

```
public class ValidatorTest {

    ArrayList<Group> groups;

    String groupNameTest1 = "Grup Name 1";
    String groupNameTest2 = "Grup Name 2";

    @Before
    public void setUp() throws Exception {

        Group group_1 = new Group();
        group_1.setName(groupNameTest1);

        Group group_2 = new Group();
        group_2.setName(groupNameTest2);

        groups = new ArrayList<>(initialCapacity: 2);
        groups.add(group_1);
        groups.add(group_2);

        Utils.user = new User();
        Utils.user.setGroups(groups);
    }
}
```

Rysunek 7.1 Inicjacja grup w systemie na potrzeby testów jednostkowych [Źródło: opracowanie własne]

Na rysunkach 7.2 oraz 7.3 przedstawione zostały przykładowe testy jednostkowe zaimplementowane w celu przetestowania poprawności działania funkcji walidujących poprawność adresu email oraz haseł wpisywanych podczas rejestracji nowego użytkownika.

```
@Test
public void validateEmail() {
    String email_1 = "adres";
    String email_2 = "adres@";
    String email_3 = "adres@fds";
    String email_4 = "adres@dsa.pl";
    String email_5 = "";

    assertFalse(Validator.validateEmail(email_1));
    assertFalse(Validator.validateEmail(email_2));
    assertFalse(Validator.validateEmail(email_3));
    assertTrue(Validator.validateEmail(email_4));
    assertFalse(Validator.validateEmail(email_5));
}
```

*Rysunek 7.2 Przykładowy test jednostkowy badający poprawność walidacji adresu email [Źródło: opracowanie własne]*

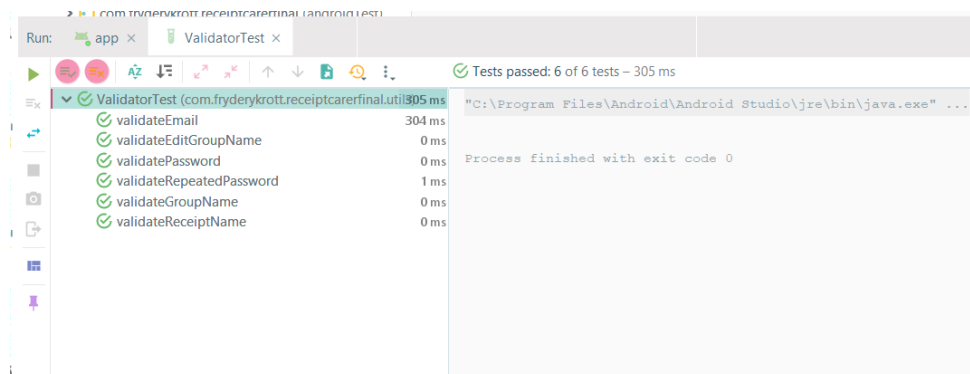
```
@Test
public void validateRepeatedPassword() {
    String password_test_1 = "short";
    String password_test_2 = "PerfectExample";
    String password_test_3 = "";

    assertFalse(Validator.validateRepeatedPassword(password_test_1, password_test_2));
    assertFalse(Validator.validateRepeatedPassword(password_test_3, password_test_1));
    assertFalse(Validator.validateRepeatedPassword(password_test_3, password_test_2));
    assertFalse(Validator.validateRepeatedPassword(password_test_3, password_test_3));
    assertFalse(Validator.validateRepeatedPassword(password_test_1, password_test_1));

    assertTrue(Validator.validateRepeatedPassword(password_test_2, password_test_2));
}
```

*Rysunek 7.3 Przykładowy test jednostkowy badający poprawność walidacji hasła wpisywanego podczas rejestracji nowego użytkownika [Źródło: opracowanie własne]*

Na rysunku 7.4 przedstawiono wynik wszystkich testów jednostkowych związanych z walidacją danych w aplikacji ReceiptCarer. Podczas testów znalezione zostały przypadki niezgodzące się z przewidywanymi rezultatami, co pozwoliło na wprowadzenie poprawek, a co za tym idzie na lepszą dbałość o prawidłowe działanie aplikacji.



Rysunek 7.4 Wynik testów jednostkowych związanych z walidacją danych [Źródło: opracowanie własne]

## 7.2. Testy funkcjonalne

Najważniejsze funkcjonalności aplikacji zostały przetestowane za pomocą testów manualnych. Przypadki testowe zostały zapisane w formie tekstowej w tabeli, a następnie zweryfikowane pod kątem zgodności z rzeczywistym działaniem w aplikacji. Poniżej zostały przedstawione przykładowe testy funkcjonalne dla dwóch przypadków użycia: PU *Eksportuj paragon* oraz PU *Edytuj grupę*.

### 7.2.1. Testy dla PU Eksportuj paragon

W tabeli 11 oraz tabeli 12 znajdują się przykładowe przypadki testowe dla PU *Eksportuj paragon*. Następnie w tabeli 13 przedstawiono wyniki testów.

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

Identyfikator	Przypadek testowy 1
<b>Tytuł</b>	Eksportowanie zdjęć paragonu na podany przez użytkownika prawidłowy adres e-mail.
<b>Opis</b>	Celem testu jest sprawdzenie, czy eksportowanie zdjęć paragonów przebiega w sposób prawidłowy (czy umożliwia wysłanie zdjęć paragonu na wybrany adres e-mail).
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych. W systemie istnieje co najmniej jeden paragon (dodany przez użytkownika).
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w wybrany paragon.</li> <li>2. Kliknij w przycisk eksportu paragonu.</li> <li>3. Wprowadź adres e-mail.</li> <li>4. Kliknij przycisk Wyślij.</li> </ol>
<b>Oczekiwany rezultat</b>	Aplikacja przenosi użytkownika do wybranej przez niego aplikacji umożliwiającej wysyłanie wiadomości e-mail na podany adres e-mail wraz z treścią zawierającą linki URL do zdjęć paragonów wybranego paragonu. Wyświetlony zostaje komunikat o wysłanym e-mailu w aplikacji.

Tabela 12 Przypadek testowy nr 2 dla PU Eksportuj paragon [Źródło: opracowanie własne]

Identyfikator	Przypadek testowy 2
<b>Tytuł</b>	Eksportowanie zdjęć paragonu na podany przez użytkownika nieprawidłowy adres e-mail.
<b>Opis</b>	Celem testu, jest sprawdzenie, czy system zareaguje na nieprawidłowe uzupełnienie danych związanych z adresem e-mail, na który wysłany ma zostać paragon.
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w wybrany paragon.</li> <li>2. Kliknij w przycisk eksportu paragonu.</li> <li>3. Wprowadź nieprawidłowy adres e-mail.</li> <li>4. Kliknij przycisk Wyślij.</li> </ol>
<b>Oczekiwany rezultat</b>	Email nie może zostać wysłany z powodu nieprawidłowego wpisanego adresu e-mail. Wyświetlony zostaje komunikat o nieprawidłowym adresie e-mail.

Tabela 13 Wynik testów dla PU Eksportuj paragon [Źródło: opracowanie własne]

Identyfikator	Wynik testu
<b>Przypadek testowy 1</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.
<b>Przypadek testowy 2</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.

#### 7.2.2. Testy dla PU Dodaj paragony

W tabelach 14, 15 oraz tabeli 16 znajdują się przykładowe przypadki testowe dla PU *Dodaj paragony*. Następnie w tabeli 17 przedstawiono wyniki testów.

Tabela 14 Przypadek testowy nr 1 dla PU Dodaj paragony [Źródło: opracowanie własne]

Identyfikator	Przypadek testowy 3
<b>Tytuł</b>	Dodanie jednego paragonu z wieloma zdjęciami do aplikacji.
<b>Opis</b>	Celem testu, jest sprawdzenie, istnieje możliwość dodania wielu zdjęć do jednego paragonu podczas dodawania paragonów do systemu..
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w przycisk <i>Dodaj paragony</i> w górnej części ekranu.</li> <li>2. Kliknij w przycisk wykonywania zdjęcia.</li> <li>3. Kliknij w przycisk akceptacji wykonanych zdjęć.</li> <li>4. Przesuń podgląd zdjęć paragonu na samo prawo.</li> <li>5. Kliknij w przycisk dodawania nowych zdjęć do paragonu.</li> <li>6. Kliknij dwukrotnie w przycisk wykonywania zdjęcia.</li> <li>7. Kliknij w przycisk akceptacji wykonywanych zdjęć.</li> <li>8. Przejrzyj czy wszystkie zdjęcia dodane do paragonu się zgadzają z wcześniej wykonanymi zdjęciami</li> <li>9. Otwórz widok powiększonego podglądu zdjęć poprzez kliknięcie</li> </ol>

	<p>w dowolne zdjęcie w podglądzie zdjęć paragonu.</p> <p>10. Przejrzyj czy wszystkie zdjęcia się zgadzają się z wcześniej dodanymi zdjęciami paragonów</p> <p>11. Zamknij widok powiększonego podglądu.</p>
<b>Oczekiwany rezultat</b>	Zdjęcia dodają się do paragonu prawidłowo. Wyświetlane są prawidłowe zdjęcia paragonu zarówno w widoku podglądu paragonu jak i powiększonym podglądzie zdjęć paragonu.

Tabela 15 Przypadek testowy nr 2 dla PU Dodaj paragony [Źródło: opracowanie własne]

Identyfikator	Przypadek testowy 4
<b>Tytuł</b>	Dodanie jednego paragonu do systemu z pustą nazwą paragonu.
<b>Opis</b>	Celem testu jest sprawdzenie, czy walidacja nazwy paragonu przebiega w sposób prawidłowy
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w przycisk <i>Dodaj paragony</i> w górnej części ekranu.</li> <li>2. Kliknij w przycisk wykonywania zdjęcia.</li> <li>3. Kliknij w przycisk akceptacji wykonanych zdjęć.</li> <li>4. Przesuń podgląd zdjęć paragonu w prawo.</li> <li>5. Zmień nazwę paragonu na pustą nazwę</li> <li>6. Kliknij w przycisk akceptacji dodawania paragonów.</li> <li>7. Kliknij w przycisk akceptacji wykonywanych zdjęć.</li> </ol>
<b>Oczekiwany rezultat</b>	Paragon nie może zostać dodany do systemu ponieważ nazwa paragonu nie może być pustym ciągiem znaków. Wyświetlana zostaje informacja o błędzie w nazwie paragonu.

Tabela 16 Przypadek testowy nr 3 dla PU Dodaj paragony [Źródło: opracowanie własne]

Identyfikator	Przypadek testowy 5
<b>Tytuł</b>	Dodanie jednego paragonu z wieloma zdjęciami do aplikacji usuwając jedno ze zdjęć paragonu.
<b>Opis</b>	Celem testu jest sprawdzenie, czy istnieje możliwość usunięcia dodanych zdjęć paragonów spośród zdjęć dodawanego paragonu.
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w przycisk <i>Dodaj paragony</i> w górnej części ekranu.</li> <li>2. Kliknij w przycisk wykonywania zdjęcia.</li> <li>3. Kliknij w przycisk akceptacji wykonanych zdjęć.</li> <li>4. Przesuń podgląd zdjęć paragonu w prawo.</li> <li>5. Kliknij w przycisk dodawania nowych zdjęć do paragonu.</li> <li>6. Kliknij dwukrotnie w przycisk wykonywania zdjęcia.</li> <li>7. Kliknij w przycisk akceptacji wykonywanych zdjęć.</li> <li>8. Otwórz widok powiększonego podglądu zdjęć poprzez kliknięcie w dowolne zdjęcie w podglądzie zdjęć paragonu.</li> <li>9. Usuń wybrane zdjęcie paragonu za pomocą przycisku usuwania zdjęć znajdującego się w lewym górnym rogu powiększonego podglądu zdjęć paragonu.</li> </ol>

	10. Zamknij widok powiększonego podglądu zdjęć paragonu. 11. Przejrzyj wszystkie zdjęcia paragonów.
<b>Oczekiwany rezultat</b>	Zdjęcia usuwają się prawidłowo. Wyświetlane są prawidłowe zdjęcia paragonu zarówno w widoku podglądu paragonu jak i powiększonym podglądzie zdjęć paragonu.

*Tabela 17 Wyniki testów dla PU Dodaj paragony [Źródło: opracowanie własne]*

Identyfikator	Wynik testu
<b>Przypadek testowy 3</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.
<b>Przypadek testowy 4</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.
<b>Przypadek testowy 5</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.

### 7.2.3. Testy dla PU Edytuj grupę

W tabeli 18 oraz tabeli 19 znajdują się przykładowe przypadki testowe dla PU *Eksportuj paragon*. Następnie w tabeli 20 przedstawiono wyniki testów.

*Tabela 18 Przypadek testowy nr 1 dla PU Edytuj grupę [Źródło: opracowanie własne]*

Identyfikator	Przypadek testowy 6
<b>Tytuł</b>	Edytowanie grupy paragonów i zmianę nazwy grupy na inną.
<b>Opis</b>	Celem testu jest sprawdzenie, czy walidacja danych podczas edycji grupy paragonów przebiega w sposób prawidłowy oraz dokonanie zmiany nazwy grupy.
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie. Aplikacja ma dostęp do bazy danych.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w przycisk <i>Grupy</i> na dolnym pasku nawigacji.</li> <li>2. Kliknij w wybraną grupę paragonów.</li> <li>3. Zmień nazwę grupy na inną prawidłową nazwę grupy.</li> <li>4. Kliknij w przycisk zatwierdzania (o nazwie <i>OK</i>).</li> </ol>
<b>Oczekiwany rezultat</b>	Następuje zmiana nazwy grupy. Grupa jest synchronizowana z bazą danych. Wyświetlony zostaje komunikat z informacją, o poprawnej edycji grupy.

*Tabela 19 Przypadek testowy nr 2 dla PU Edytuj grupę [Źródło: opracowanie własne]*

Identyfikator	Przypadek testowy 7
<b>Tytuł</b>	Edytowanie grupy paragonów i próba zmiany nazwy grupy na istniejącą już w aplikacji nazwę grupy.
<b>Opis</b>	Celem testu, jest sprawdzenie, czy walidacja danych podczas edycji grupy paragonów przebiega w sposób prawidłowy.
<b>Warunki wstępne</b>	Użytkownik jest zalogowany w systemie.
<b>Kroki</b>	<ol style="list-style-type: none"> <li>1. Kliknij w przycisk <i>Grupy</i> na dolnym pasku nawigacji.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Kliknij w wybraną grupę paragonów.</li> <li>3. Zmień nazwę grupy na istniejącą już w aplikacji inną nazwę grupy.</li> <li>4. Kliknij w przycisk zatwierdzania (o nazwie <i>OK</i>).</li> </ol>
<b>Oczekiwany rezultat</b>	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.

*Tabela 20 Wyniki testów dla PU Edytuj grupę [Źródło: opracowanie własne]*

Identyfikator	Wynik testu
<b>Przypadek testowy 6</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.
<b>Przypadek testowy 7</b>	Test przebiegł prawidłowo. System zachowuje się zgodnie z oczekiwaniami.



## **Zakończenie**

### **1. Podsumowanie pracy**

Praca miała na celu zaprojektowanie i zaimplementowanie aplikacji umożliwiającej efektywne organizowanie dowodów zakupów, a także ich skanowanie w celu usprawnienia procesu ich ewidencji. Aplikacja miała także pozwalać na eksportowanie paragonów, wyszukiwanie ich oraz gromadzenie informacji o kończących się terminach gwarancji. Wszystkie założenia projektowe zostały spełnione. Aplikacja została zaimplementowana wraz ze wszystkimi zakładanymi funkcjonalnościami.

Zastosowany zestaw narzędzi oferowany przez Firebase pozwolił na ujednolicenie i uproszczenie implementacji. Firebase oferuje bardzo szeroki zakres usług, który okazał się bardzo wygodny w użyciu i poleciłbym każdemu korzystanie z niego, szczególnie na potrzeby implementacji aplikacji mobilnej.

### **2. Kierunki dalszego rozwoju**

Aplikacja zaprojektowana i zaimplementowana w pracy pozwala na efektywną organizację dowodów zakupów. Nie zmienia to jednak faktu, że istnieją jeszcze udogodnienia, które można wprowadzić w ramach kolejnych aktualizacji do aplikacji. Jednym z pomysłów na udogodnienie jest wprowadzenie do aplikacji możliwości prowadzenia statystyk na temat gromadzonych paragonów. Sprawiłoby to, że aplikacja rozszerzyłaby swoje zastosowanie do nie tylko organizacji dowodów zakupów, ale także pozwalałaby na prowadzenie analizy wydatków. Kolejnym ulepszeniem do aplikacji mogłoby być zaimplementowanie algorytmów wspomagających ekstrakowanie danych z paragonów, które pozwalałby na skuteczniejszą oraz szybszą analizę zdjęć. Istnieje także możliwość rozszerzenia możliwości skanowania do obsługi innych dowodów zakupów, takich jak faktury.

## Bibliografia

- [1] „Google - strona oficjalna,” [Online]. Available: <https://about.google/intl/pl/>. [Data uzyskania dostępu: 27 10 2019].
- [2] „Allegro - platforma transakcyjna on-line,” [Online]. Available: <https://allegro.pl/>. [Data uzyskania dostępu: 27 10 2019].
- [3] „Allegro - Program Ochrony Kupujących,” [Online]. Available: <https://dlakupujacych.allegro.pl/bezpieczenstwo/program-ochrony-kupujacych/warunki-skorzystania>. [Data uzyskania dostępu: 26 10 2019].
- [4] „Urząd Ochrony Konkurencji i Konsumentów - strona oficjalna,” [Online]. Available: <https://www.uokik.gov.pl/>. [Data uzyskania dostępu: 22 11 2019].
- [5] „Urząd Ochrony Konkurencji i Konsumentów - reklamacje bez paragonów,” [Online]. Available: [https://www.uokik.gov.pl/aktualnosci.php?news\\_id=11630](https://www.uokik.gov.pl/aktualnosci.php?news_id=11630). [Data uzyskania dostępu: 22 11 2019].
- [6] „Aplikacja Pan Paragon - strona oficjalna,” [Online]. Available: <https://www.panparagon.pl/>. [Data uzyskania dostępu: 26 10 2019].
- [7] „Pan Paragon,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.modernfactory.paragon&hl=pl>. [Data uzyskania dostępu: 26 10 2019].
- [8] „Internet - computer network,” [Online]. Available: <https://www.britannica.com/technology/Internet>. [Data uzyskania dostępu: 9 12 2019].
- [9] „Java - język programowania,” [Online]. Available: <https://www.java.com/pl/>. [Data uzyskania dostępu: 22 11 2019].
- [10] „Android Studio - oficjalna strona dla deweloperów,” [Online]. Available: <https://developer.android.com/studio>. [Data uzyskania dostępu: 25 11 2019].
- [11] „Cloud Firestore - strona oficjalna,” [Online]. Available: <https://firebase.google.com/docs/firestore>. [Data uzyskania dostępu: 25 11 2019].
- [12] „OCR - definicja,” [Online]. Available: <https://sjp.pwn.pl/slowniki/OCR.html>. [Data uzyskania dostępu: 06 12 2019].
- [13] „Oprogramowanie Git - strona oficjalna,” [Online]. Available: <https://git-scm.com/>. [Data uzyskania dostępu: 25 11 2019].
- [14] „Visual Paradigm Online,” [Online]. Available: <https://online.visual-paradigm.com/>. [Data uzyskania dostępu: 25 11 2019].
- [15] „Biblioteka Picasso,” [Online]. Available: <https://square.github.io/picasso/>. [Data uzyskania dostępu: 06 12 2019].
- [16] „Android - strona oficjalna,” [Online]. Available: [https://www.android.com/intl/pl\\_pl/](https://www.android.com/intl/pl_pl/). [Data uzyskania dostępu: 25 11 2019].
- [17] „Jetbrains - strona oficjalna,” [Online]. Available: <https://www.jetbrains.com/>. [Data uzyskania dostępu: 26 11 2019].
- [18] „IntelliJ IDEA - strona oficjalna,” [Online]. Available: [https://www.jetbrains.com/idea/promo/ultimate/?gclid=Cj0KCQiAt\\_PuBRDcARIsAMNIBdo2eqaR6SfGD9Rf5x7zCSaYyA\\_pd4jkg4b8asVE1bdEGASXtvzHtm8aAlV5EALw\\_wcB](https://www.jetbrains.com/idea/promo/ultimate/?gclid=Cj0KCQiAt_PuBRDcARIsAMNIBdo2eqaR6SfGD9Rf5x7zCSaYyA_pd4jkg4b8asVE1bdEGASXtvzHtm8aAlV5EALw_wcB). [Data uzyskania dostępu: 26 11 2019].
- [19] R. A. Kłopotek, „Bazy danych NoSQL,” [Online]. Available:

- [https://rklopotek.blog.uksw.edu.pl/files/2017/02/Bazy\\_danych\\_wyk%c5%82ad\\_14\\_draft.pdf](https://rklopotek.blog.uksw.edu.pl/files/2017/02/Bazy_danych_wyk%c5%82ad_14_draft.pdf). [Data uzyskania dostępu: 08 12 2019].
- [20] H. Mazur i Z. Mazur, Projektowanie relacyjnych baz danych, Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2004.
- [21] „Przykład zastosowania Firebase Vision,” [Online]. Available: <https://codelabs.developers.google.com/codelabs/mlkit-android/#5>. [Data uzyskania dostępu: 06 12 2019].
- [22] „Odległość Levenshteina - wyjaśnienie algorytmu,” [Online]. Available: <http://www.algorytm.org/przetwarzanie-tekstu/odleglosc-levenshteina-odleglosc-edycyjna.html>. [Data uzyskania dostępu: 06 12 2019].
- [23] „JUnit 4 - strona oficjalna,” [Online]. Available: <https://junit.org/junit4/>. [Data uzyskania dostępu: 9 12 2019].
- [24] „Aplikacja Receipt Bank,” [Online]. Available: [https://play.google.com/store/apps/details?id=com.receiptbank.android&hl=en\\_US](https://play.google.com/store/apps/details?id=com.receiptbank.android&hl=en_US). [Data uzyskania dostępu: 26 10 2019].
- [25] „Aplikacja Receipt Lens-Expense Tracking & Reporting,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.glority.receipt&hl=pl>. [Data uzyskania dostępu: 26 10 2019].
- [26] „Aplikacja Paragon - karty lojalnościowe i gwarancje pod ręką,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.modernfactory.paragon&hl=pl>. [Data uzyskania dostępu: 22 11 2019].
- [27] A. Stasiewicz, Android. Podstawy tworzenia aplikacji, Helion, 2013.

## Spis rysunków

Rysunek 4.1 Diagram klas (domenowy) [Źródło: opracowanie własne] .....	17
Rysunek 4.2 Diagram przypadków użycia [Źródło: opracowanie własne] .....	20
Rysunek 4.3 Dodaj nową grupę paragonów – Lista grup (po lewej). Widok dodawania grupy (po prawej) [Źródło: opracowanie własne] .....	23
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] .....	23
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] .....	25
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] .....	25
Rysunek 4.7 Dodaj paragony – kolejno od lewej krok 0, krok 2 oraz krok 5 [Źródło: opracowanie własne] .....	27
Rysunek 4.8 Dodaj paragony – kolejno od lewej krok 9 oraz krok 12 [Źródło: opracowanie własne] .....	27
Rysunek 5.1 Diagram rozmieszczenia [Źródło: opracowanie własne] .....	30
Rysunek 5.2 Diagram pakietów [Źródło: opracowanie własne] .....	31
Rysunek 5.3 Diagram ERD bazy danych [Źródło: opracowanie własne] .....	32
Rysunek 5.4 Widok struktury bazy danych (przeznaczonej do przechowywania informacji na temat użytkowników) w konsoli Firebase dostępnej dla aplikacji projektowanej w pracy. [Źródło: opracowanie własne] .....	33
Rysunek 5.5 Widok przechowywanych w bazie danych (przeznaczonej na przechowywanie plików) zdjęć paragonów w konsoli Firebase dostępnej dla aplikacji projektowanej w pracy. [Źródło: opracowanie własne] .....	34
Rysunek 5.6 Wynik algorytmu OCR od Firebase w formie blozków tekstu [Źródło: [21]]	34
Rysunek 5.7 Przykładowe formy zapisu kwoty sumarycznej na paragonie [Źródło: opracowanie własne] .....	36
Rysunek 6.1 Ekran logowania – widok główny (po lewo), informacja o błędach (po prawo) [Źródło: opracowanie własne] .....	39
Rysunek 6.2 Ekran rejestracji – widok główny (po lewo), informacja o błędach (po prawo) [Źródło: opracowanie własne] .....	40
Rysunek 6.3 Ekran listy paragonów [Źródło: opracowanie własne] .....	41
Rysunek 6.4 Ekran edytowania paragonu – widok edycji (po lewej), informacja o błędzie nazwy (po prawej) [Źródło: opracowanie własne] .....	42
Rysunek 6.5 Ekrany specjalne: wybór daty dodania paragonu (po lewej), wybór długości trwania gwarancji (środek i po prawej) [Źródło: opracowanie własne] .....	42
Rysunek 6.6 Ekrany specjalne: wybór całkowitej kwoty z paragonu (po lewej), wybór grupy do której należy paragon (po prawej) [Źródło: opracowanie własne] .....	43
Rysunek 6.7 Dodawanie nowego znacznika – wpisanie nazwy znacznika (krok 0, po lewej), zaakceptowanie znacznika do paragonu (krok 1, po prawej), wybór znacznika z listy podpowiadającej znaczniki [Źródło: opracowanie własne] .....	43
Rysunek 6.8 Możliwość dodania nowych zdjęć do paragonu (po lewo), podgląd zdjęć paragonów (po prawo) [Źródło: opracowanie własne] .....	44
Rysunek 6.9 ekran dodawania zdjęć – brak wykonanych zdjęć (po lewej), wykonano jedno zdjęcie, wykonano 2 zdjęcia (środek oraz po prawo) [Źródło: opracowanie własne] .....	45
Rysunek 6.10 Ekran edycji dodawanych paragonów [Źródło: opracowanie własne] .....	46

<i>Rysunek 6.11 Eksport paragonu [Źródło: opracowanie własne]</i> .....	47
<i>Rysunek 6.12 Ekran z listą grup paragonów [Źródło: opracowanie własne]</i> .....	48
<i>Rysunek 6.13 Ekran dodawania nowej grupy paragonów [Źródło: opracowanie własne]</i> .....	49
<i>Rysunek 6.14 Ekrany edycji grupy [Źródło: opracowanie własne]</i> .....	50
<i>Rysunek 6.15 Wyszukiwanie paragonów – brak kryteriów wyszukiwania (po lewo), ekran znacznika specjalnego wyboru przedziału dat (środek), ekran znacznika specjalnego wyboru przedziału kwoty pieniężnej (po prawo) [Źródło: opracowanie własne]</i> .....	51
<i>Rysunek 6.16 Wyszukiwanie paragonów – wynik wyszukiwania dla podanych kryteriów [Źródło: opracowanie własne]</i> .....	51
<i>Rysunek 7.1 Inicjacja grup w systemie na potrzeby testów jednostkowych [Źródło: opracowanie własne]</i> .....	52
<i>Rysunek 7.2 Przykładowy test jednostkowy badający poprawność walidacji adresu email [Źródło: opracowanie własne]</i> .....	53
<i>Rysunek 7.3 Przykładowy test jednostkowy badający poprawność walidacji hasła wpisywanego podczas rejestracji nowego użytkownika [Źródło: opracowanie własne]</i> ....	53
<i>Rysunek 7.4 Wynik testów jednostkowych związanych z walidacją danych [Źródło: opracowanie własne]</i> .....	54

## Spis tabel

<i>Tabela 1 Opis użytkownika systemu</i> .....	6
<i>Tabela 2 Potrzeby i cechy</i> .....	7
<i>Tabela 3 Inne wymagania produktowe</i> .....	7
<i>Tabela 4 Porównanie oferowanych funkcjonalności [Źródło: opracowanie własne]</i> .....	14
<i>Tabela 5 Słownik pojęć [Źródło: opracowanie własne]</i> .....	15
<i>Tabela 6 Macierz śladowania wymagań [Źródło: opracowanie własne]</i> .....	19
<i>Tabela 7 Skrócone opisy przypadków użycia [Źródło: opracowanie własne]</i> .....	21
<i>Tabela 8 Dodaj grupę [Źródło: opracowanie własne]</i> .....	22
<i>Tabela 9 Przeszukuj paragony - Scenariusz</i> .....	24
<i>Tabela 10 Dodaj paragony - Scenariusz</i> .....	26
<i>Tabela 11 Przypadek testowy nr 1 dla PU Eksportuj paragon [Źródło: opracowanie własne]</i> .....	54
<i>Tabela 12 Przypadek testowy nr 2 dla PU Eksportuj paragon [Źródło: opracowanie własne]</i> .....	55
<i>Tabela 13 Wynik testów dla PU Eksportuj paragon [Źródło: opracowanie własne]</i> .....	55
<i>Tabela 14 Przypadek testowy nr 1 dla PU Dodaj paragony [Źródło: opracowanie własne]</i> .....	55
<i>Tabela 15 Przypadek testowy nr 2 dla PU Dodaj paragony [Źródło: opracowanie własne]</i> .....	56
<i>Tabela 16 Przypadek testowy nr 3 dla PU Dodaj paragony [Źródło: opracowanie własne]</i> .....	56
<i>Tabela 17 Wyniki testów dla PU Dodaj paragony [Źródło: opracowanie własne]</i> .....	57
<i>Tabela 18 Przypadek testowy nr 1 dla PU Edytuj grupę [Źródło: opracowanie własne]</i> ..	57
<i>Tabela 19 Przypadek testowy nr 2 dla PU Edytuj grupę [Źródło: opracowanie własne]</i> ..	57
<i>Tabela 20 Wyniki testów dla PU Edytuj grupę [Źródło: opracowanie własne]</i> .....	58