

PIOTR DUDKIEWICZ

Numer albumu: 39257

Kierunek studiów: Informatyka

Specjalność: Systemy komputerowe i oprogramowanie

Forma studiów: studia stacjonarne

APLIKACJA DO PODZIAŁU WSPÓLNYCH WYDATKÓW WŚRÓD ZNAJOMYCH

Application to sharing common expenses among friends.

Praca dyplomowa
napisana pod kierunkiem:

dr inż. Radosław Maciaszczyk

Katedra Architektury Komputerów i Telekomunikacji

Data wydania tematu pracy: 17.06.2020

Data dopuszczenia pracy do egzaminu:

Szczecin, 2021

**OŚWIADCZENIE
AUTORA PRACY DYPLOMOWEJ**

Oświadczam, że praca dyplomowa inżynierska pn.

Aplikacja do podziału wspólnych wydatków wśród znajomych.

(temat pracy dyplomowej)

napisana pod kierunkiem:

dr inż. Radosław Maciaszczyk

(tytuł lub stopień naukowy imię i nazwisko opiekuna pracy)

jest w całości moim samodzielnym autorskim opracowaniem sporządzonym przy wykorzystaniu wykazanej w pracy literatury przedmiotu i materiałów źródłowych.

Złożona w dziekanacie **Wydziału Informatyki**

(wydział)

treść mojej pracy dyplomowej w formie elektronicznej jest zgodna z treścią w formie pisemnej.

Oświadczam ponadto, że złożona w dziekanacie praca dyplomowa ani jej fragmenty nie były wcześniej przedmiotem procedur procesu dyplomowania związanych z uzyskaniem tytułu zawodowego w uczelniach wyższych.

.....*Piotr Dudkiewicz*.....
podpis dyplomanta

Szczecin, dn. 28.04.21r

Streszczenie

Niniejsza praca inżynierska dotyczy tworzenia responsywnej aplikacji webowej do podziału wydatków wśród znajomych, która ma ułatwiać zarządzanie i księgowanie. Praca zawiera zagadnienia związane z systemami mobilnymi, sposobami wytwarzania aplikacji oraz aplikacjami wykorzystującymi przetwarzanie chmurowe. Aplikacja została wykonana przy użyciu technologii VueJS oraz ExpressJS.

Słowa kluczowe: RWD, przetwarzanie chmurowe, responsywne aplikacje mobilne, aplikacje internetowe

Abstract

This engineering thesis concerns the creation of a responsive web application for sharing expenses among friends, which is to facilitate management and bookkeeping. The work covers issues related to mobile systems, methods of application development and applications that use cloud computing. The application was made using VueJS or ExpressJS technology.

Keywords: RWD, cloud computing, responsive mobile applications, web applications

Spis treści

Streszczenie	3
1. Wstęp	6
1.1 Cel i zakres pracy	6
2. Przetwarzanie chmurowe	7
2.1 Podział ze względu na dostarczane usługi	7
2.1.1 Platforma jako usługa (PaaS)	7
2.1.2 Oprogramowanie jako usługa (SaaS)	8
2.1.3 Infrastruktura jako usługa (IaaS)	8
2.2 Podział ze względu na dostępność	8
2.2.1 Chmura publiczna	8
2.2.2 Chmura prywatna	8
2.2.3 Chmura hybrydowa	8
3. Sposoby wytwarzania aplikacji mobilnych	9
3.1 Aplikacje natywne	9
3.2 Aplikacje hybrydowe	10
3.3 Aplikacje cross-platformowe	10
3.4 Aplikacje Progresywne (PWA)	10
4. Podobne aplikacje istniejące na rynku	12
4.1 Splitwise	12
4.2 Tricount	13
4.3 Settle Up	13
4.4 Splid	15
5. Narzędzia	15
5.1 Serwer	15
5.1.1 Mongoose	15
5.1.2 ExpressJS	17
5.1.3 Biblioteki	18
5.2 Klient	18
5.2.1 VueJS	18
5.2.2 Bootstrap-Vue	19
5.2.3 Biblioteki	20
6. Prototyp	20
6.1 Funkcjonalność	20

6.2	Wzorzec architektoniczny MVVM	20
6.3	Struktura bazy danych	21
6.4	Interfejs użytkownika	24
6.4.1	Widok logowania	24
6.4.2	Widok rejestracji	25
6.4.3	Widok resetowania hasła.....	27
6.4.4	Widok przypomnienia loginu.....	29
6.4.5	Widok strony głównej	29
6.4.6	Widok listy zrzutek	32
6.4.7	Widok szczegółów grupy	35
6.4.8	Widok szczegółów grupowej zrzutki	36
6.4.9	Widok szczegółów pojedynczej zrzutki	38
6.4.10	Widok tworzenia grupy	39
6.4.11	Widok edycji informacji o grupie	41
6.4.12	Widok edycji członków grupy i uprawnień	42
6.4.13	Widok wyboru rodzaju tworzonej zbiórki.....	43
6.4.14	Widok tworzenia grupowej zrzutki	44
6.4.15	Widok tworzenia pojedynczej zbiórki.....	48
6.4.16	Widok edycji pojedynczej zrzutki.....	49
6.4.17	Widok edycji informacji o zrzutce grupowej	50
6.4.18	Widok edycji należności	51
6.4.19	Widok listy powiadomień	52
6.4.20	Widok listy zaproszeń	53
6.4.21	Widok ustawień użytkownika	53
6.4.22	Widok edycji informacji o użytkowniku.....	54
6.4.23	Widok zmiany hasła	55
6.4.24	Widok potwierdzenia zmiany e-mail , aktywacji konta lub przywrócenia konta 55	
6.4.25	Usuwanie zrzutki, grupy lub użytkownika.....	56
6.4.26	Okno z komunikatem	56
6.5	Mechanizm odnawiania zrzutek	56
7.	Testy	57
8.	Podsumowanie	60
8.1	Możliwe przyszłe rozwiązania	60
9.	Bibliografia.....	62

1. Wstęp

W dzisiejszych czasach urządzenia mobilne są nieodłączną częścią życia człowieka. Nie ma znaczenia, czy prowadzimy rozmowę, szukamy dobrej restauracji na sobotni obiad, dokonujemy płatności, czy sterujemy zdalnie urządzeniem odkurzającym mieszkanie. Do tych wszystkich, różnorodnych czynności wykorzystujemy telefony komórkowe. W większości codziennych rzeczy pomagają nam przeznaczone do tego aplikacje. Wzrost zapotrzebowania na aplikacje, które ułatwiają płatności za usługi lub zakupy, stworzyło zapotrzebowanie na aplikacje, które pomagają w planowaniu i podziale kosztów – wydatków. Takie narzędzie usprawniłoby podział kosztów znajomych wychodzących razem do restauracji, jadących na wspólny urlop, po rozliczenia kosztów stałych za media wśród współlokatorów. Podział wydatków przy pomocy takiej aplikacji zaoszczędzi czas, który zajmuje podział rachunków na poszczególne osoby.

Równolegle do całej technologii aplikacji mobilnych, rozwija się niezależna technologia chmur obliczeniowych, które dzięki swojej wszechstronności umożliwiają przechowywanie wielu danych z prywatnych urządzeń na serwerach zewnętrznych, gwarantujących dostęp do tych danych niezależnie od miejsca w jakim się znajdujemy. Prędkość przesłania tych danych jest błyskawiczna i pozwala na ich natychmiastowy odczyt i edycję. Połączenie aplikacji z chmurą pozwala na uzyskanie szerokich możliwości. Telefon zapewnia podręczne narzędzie, które jest elastyczne i ogólnodostępne a wykorzystanie chmury zapewnia stały i natychmiastowy dostęp do danych.

1.1 Cel i zakres pracy

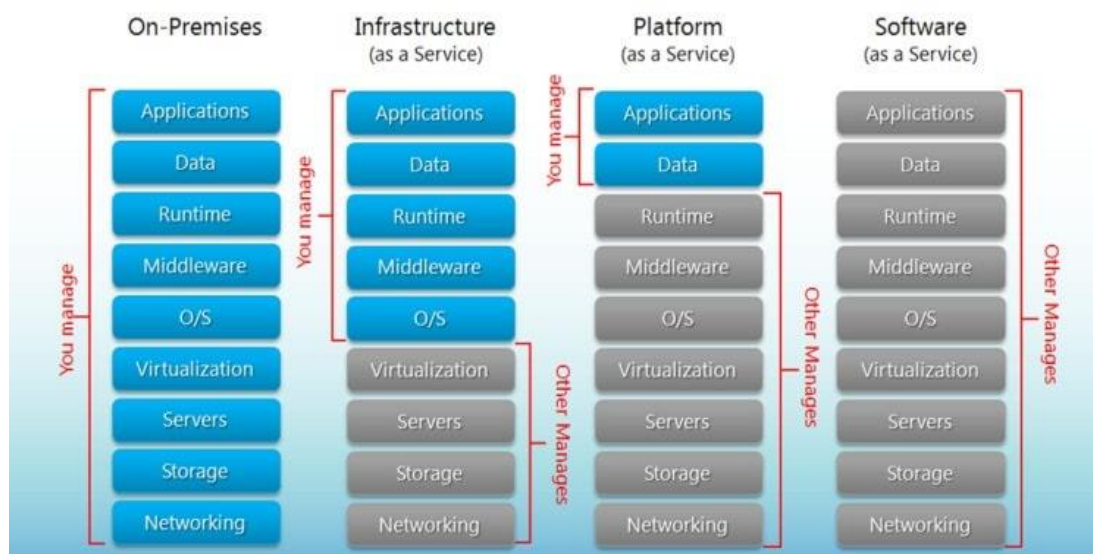
Tematem przedstawionej pracy inżynierskiej jest stworzenie aplikacji w formie responsywnej strony internetowej (ang. *Responsive Web Design*), która dostosowuje się do urządzenia, na którym jest wyświetlana. Dodatkowym atutem jest fakt, że strona responsywna na taki sam adres zarówno na urządzeniach mobilnych jak i na komputerach, niezależnie od systemu operacyjnego. Wygodą stron responsywnych jest również to, że aby korzystać z aplikacji nie trzeba jej pobierać ani instalować na urządzeniu. Czas wykonania takiej strony jest znacznie krótszy niż czas potrzebny do stworzenia aplikacji mobilnej.

Aplikacja przedstawiona w niniejszej pracy, ma za zadanie usprawnić podział kosztów stałych lub jednorazowych wśród współlokatorów, znajomych, lub innych członków danej grupy generującej wspólne wydatki.

2. Przetwarzanie chmurowe

Przetwarzanie chmurowe jest to udostępnianie zasobów obliczeniowych (pamięć, procesory, aplikacje, usługi, sieć) na żądanie przy pomocy Internetu. Zapewnia ono współdzielenie oraz dostępność danych z dowolnego miejsca. Usługobiorca może dostosować zasoby do swoich potrzeb i płacić za faktyczne wykorzystanie ich oraz dzięki zastosowaniu tej technologii unika części bądź całości kosztów, które musiałby ponieść przy budowie i utrzymaniu własnej infrastruktury.

2.1 Podział ze względu na dostarczane usługi



Rysunek 1 Porównanie IaaS, PaaS i SaaS [15]

2.1.1 Platforma jako usługa (PaaS)

Jest to pracujące w chmurze środowisko, które dostarcza nam niezbędne komponenty takie jak serwer aplikacji (np. serwer NodeJS), serwer bazodanowy, serwer e-mail oraz wiele innych, dzięki czemu eliminuje koszty związane z zakupem i utrzymaniem sprzętu oraz oprogramowania. Przykładem platformy opartej o PaaS jest Heroku (<https://www.heroku.com/>), która obsługuje różne technologie na przykład Java, JavaScript, PHP i Ruby.

2.1.2 Oprogramowanie jako usługa (SaaS)

Polega na udostępnianiu aplikacji działając w chmurze na oddalonych komputerach, utrzymywana przez dostawcę, zazwyczaj użytkownicy łączą się z nią poprzez przeglądarkę lub dedykowaną aplikację. Zaletami takiego rozwiązania są szybka możliwość korzystania z usługi, dostępność z dowolnego komputera, zapis danych w chmurze oraz brak kosztów wytworzenia aplikacji. Przykładem rozwiązania typu SaaS jest Slack, czyli aplikacja, która dostarcza nam gotowe narzędzie do komunikacji między członkami zespołu.

2.1.3 Infrastruktura jako usługa (IaaS)

To usługa dająca możliwość korzystania z zasobów takich jak pamięć, procesory i sieć, dzięki czemu możemy uniknąć kosztów związanych z zakupem oraz utrzymaniem sprzętu. Przykładem usługi typu IaaS jest Microsoft Azure (<https://azure.microsoft.com/>), gdzie możemy zakupić serwer VPS i skonfigurować go według własnych potrzeb (możemy np. zainstalować środowisko Node.JS i uruchomić własną aplikację).

2.2 Podział ze względu na dostępność

2.2.1 Chmura publiczna

Obecnie jest to najpopularniejsze rozwiązanie, polegające na korzystaniu z zasobów obliczeniowych zewnętrznych usługodawców. Takie rozwiązanie daje nam niskie koszty, ponieważ nie musimy wydawać pieniędzy na zakup sprzętu i oprogramowania, dodatkowo możemy przenieść odpowiedzialność na dostawcę oraz zapewnić większą niezawodność.

2.2.2 Chmura prywatna

Jest to infrastruktura zewnętrzna bądź wewnętrzna przeznaczona dla jednej organizacji, zarządzana samodzielnie lub przez zewnętrzną firmę. Zaletą tego rozwiązania jest większa kontrola nad zasobami oraz brak współdzielenia ich z innymi jednostkami.

2.2.3 Chmura hybrydowa

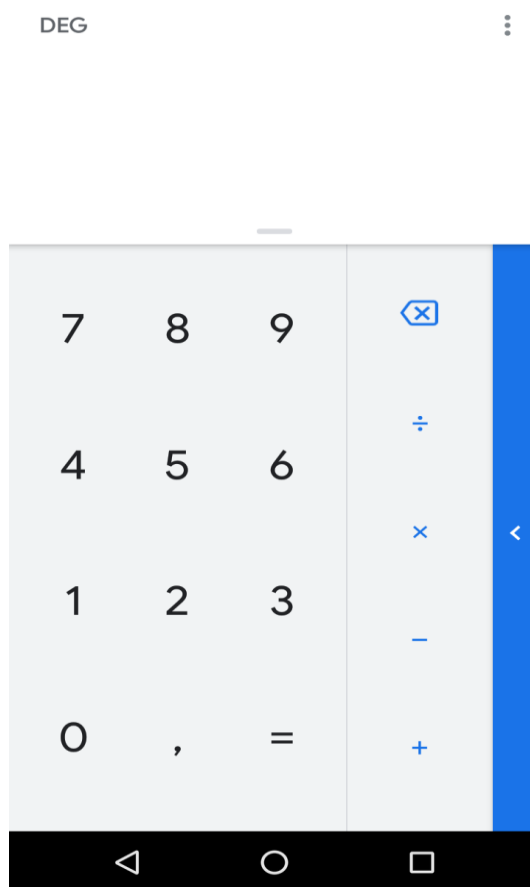
Przetwarzanie chmurowe, wykorzystujące chmurę prywatną i publiczną, czyli część infrastruktury organizacji pracuje w przestrzeni publicznej, a część w przestrzeni prywatnej. Dzięki takiemu rozwiązaniu możemy wykorzystać zalety obu rozwiązań.

3. Sposoby wytwarzania aplikacji mobilnych

Aplikacje mobilne są to aplikacje, z których możemy korzystać na urządzeniach przenośnych takich jak smartphone, tablet czy iPod. Obecnie na rynku istnieje kilka sposobów wytwarzania tego typu aplikacji. Wybór sposobu jest zależny od posiadanych środków pieniężnych, grupy docelowej i nakładu czasowego.

3.1 Aplikacje natywne

Są to aplikacje stworzone dla określonej platformy (Android, Windows Phone lub IOS) w specjalnej dla danego środowiska technologii. Dla systemu IOS będzie to Objective-C lub Swift (zyskuje na popularności), dla Windows Phone C#, a dla Android Java lub Kotlin (zyskuje na popularności). Zaletami tego rozwiązania są wysoka wydajność (technologia przystosowana do danego środowiska), większy dostęp do komponentów urządzenia, interfejs spójny z danym środowiskiem oraz lepsze wrażenia podczas korzystania z aplikacji (UX). Z wad należy wspomnieć o dużym nakładzie pieniężnym i czasowy (na każde urządzenie należy stworzyć osobną aplikację) oraz konieczność instalacji aplikacji na urządzeniu.



Rysunek 2 Aplikacja natywna

3.2 Aplikacje hybrydowe

To aplikacje, które łączą rozwiązanie webowe i natywne. Oznacza to, że aplikacja jest tworzona podobnie do aplikacji webowej, czyli przy użyciu takich technologii jak HTML, CSS i JavaScript z tą różnicą, że dodatkowo dostarczane są wtyczki, które dają możliwości natywne, a sama aplikacja renderowana jest przy użyciu natywnego WebView jako strona internetowa, która imituje aplikację natywną. Do zalet tego rozwiązania należą niskie nakłady czasowe i finansowe, łatwiejsze utrzymanie aplikacji oraz dostęp do funkcji urządzenia. Do wad należy zaliczyć słabszą wydajność przy bardziej zaawansowanych aplikacjach, działa tylko online oraz mogą występować problemy podczas korzystania z funkcji występujących tylko na jednej platformie. Przykładowymi aplikacjami są Twitter i Gmail, a narzędziami do tworzenia aplikacji hybrydowych są PhoneGap (<https://build.phonegap.com/>), Ionic (<https://ionicframework.com/>) oraz Moblie Angular UI (<http://mobileangularui.com/>).

3.3 Aplikacje cross-platformowe

Aplikacje te tworzone są przy użyciu tej samej technologii, ale mające możliwość uruchomienia się na wielu platformach. Oznacza to, że posiadają wspólną bazę kodu przy jednocześnie występujących różnicach w interfejsie. Zaletami tego sposobu są niski koszt, krótki czas realizacji, zbliżona wydajność do aplikacji natywnych oraz łatwiejsze utrzymanie aplikacji. Do wad należy większe skomplikowanie procesu tworzenia (niekiedy trzeba pisać osobną część kodu dla danej platformy). Przykładowymi technologiami cross-platformowymi są Xamarin [17] oraz React Native (<https://reactnative.dev/>), a aplikacjami Facebook i Slack (<https://slack.com/>).

3.4 Aplikacje Progresywne (PWA)

Jest to pewnego rodzaju rozszerzenie aplikacji wykonanych w technologii RWD (patrz 1.1) z dodatkową możliwością zapisu na telefonie, działaniem w trybie offline, dostępem do powiadomień typu push oraz dostępem do funkcji telefonu np. aparat, GPS. Do zalet tego rozwiązania należą multiplatformowość, szybkie i tanie wytworzenie aplikacji, nie trzeba jej instalować oraz działanie w trybie offline. Istnieją również wady takie jak większe zużycie energii, słabe wsparcie przez IOS oraz gorszy dostęp do funkcjonalności urządzenia. Przykładowymi aplikacjami progresywnymi są Uber (<https://www.uber.com/>) i Tinder (<https://tinder.com/>).



Q Dodaj miejsce odbioru

Teraz



Zezwól na dostęp do lokalizacji

[Idealny odbiór](#)

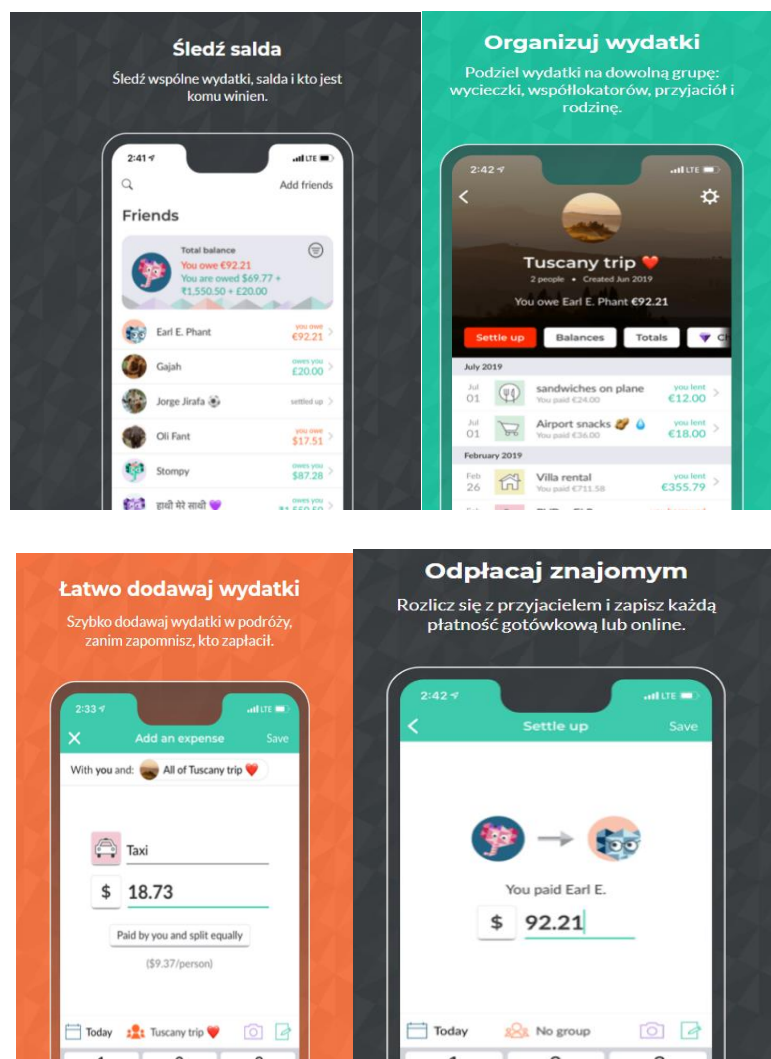
Rysunek 3 Przykładowa aplikacja PWA

4. Podobne aplikacje istniejące na rynku

Do tej pory na rynku aplikacji mobilnych, które pozwalają na podział kosztów, ukazało się niewiele. Dodatkowo, każda z nich jest dostępna w formie właśnie aplikacji mobilnej, którą trzeba pobrać i zainstalować na urządzeniu.

4.1 Splitwise

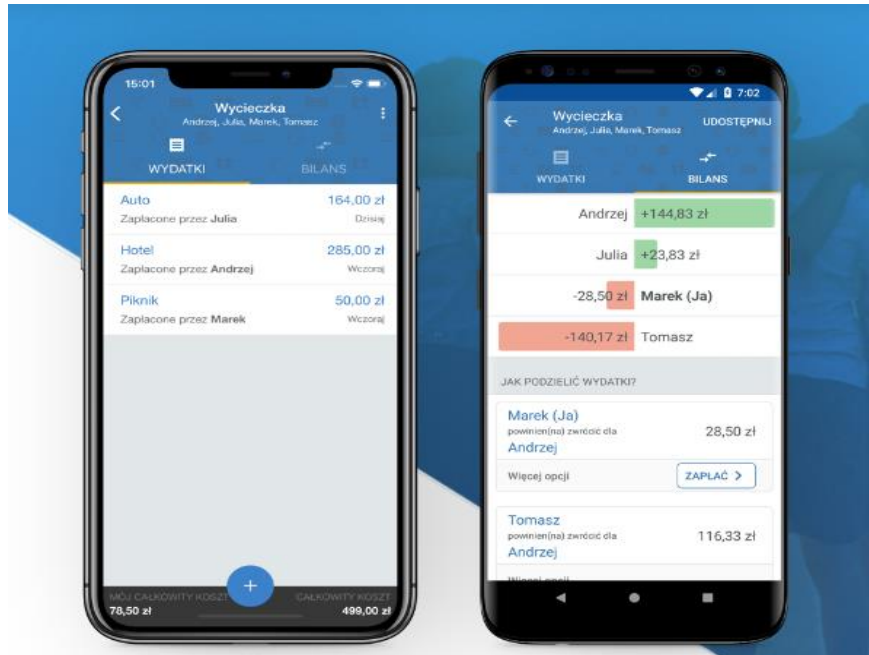
Aplikacja Splitwise pozwala na podział wydatków ze znajomymi oraz współlokatorami, w równych i nierównych % płatności, planowanie i kategoryzowanie wydatków. Obsługiwana w siedmiu językach, pozwala na działanie w trybie offline oraz synchronizuje i przechowuje dane na chmurze. Dostępna jest wersja rozszerzona, która podsumowuje wydatki użytkownika i przedstawia je w formie wykresów oraz pozwala na skanowanie paragonów. Aplikacja działa na urządzeniach z systemem Android oraz IOS.



Rysunek 4. Aplikacja Splitwise (<https://www.splitwise.com/>)

4.2 Tricount

Aplikacja do podziału wydatków wśród ludzi. Każdy z użytkowników znajdujący się w danej grupie znajomych może dodawać wydatki do wspólnego konta. Aplikacja wysyła powiadomienie niezwłocznie po dodaniu, edytowaniu lub usunięciu przez osobę, kolejnego kosztu. Dodatkowym atutem jest rozliczanie w wielu walutach. Aplikacja pozwala na nierównomierny podział kosztów, oraz może wygenerować podsumowanie rachunków i kosztów w formie PDF. Umożliwia ona pracę w trybie offline.



Rysunek 5. Aplikacja Tricount (<https://www.tricount.com/>)

4.3 Settle Up

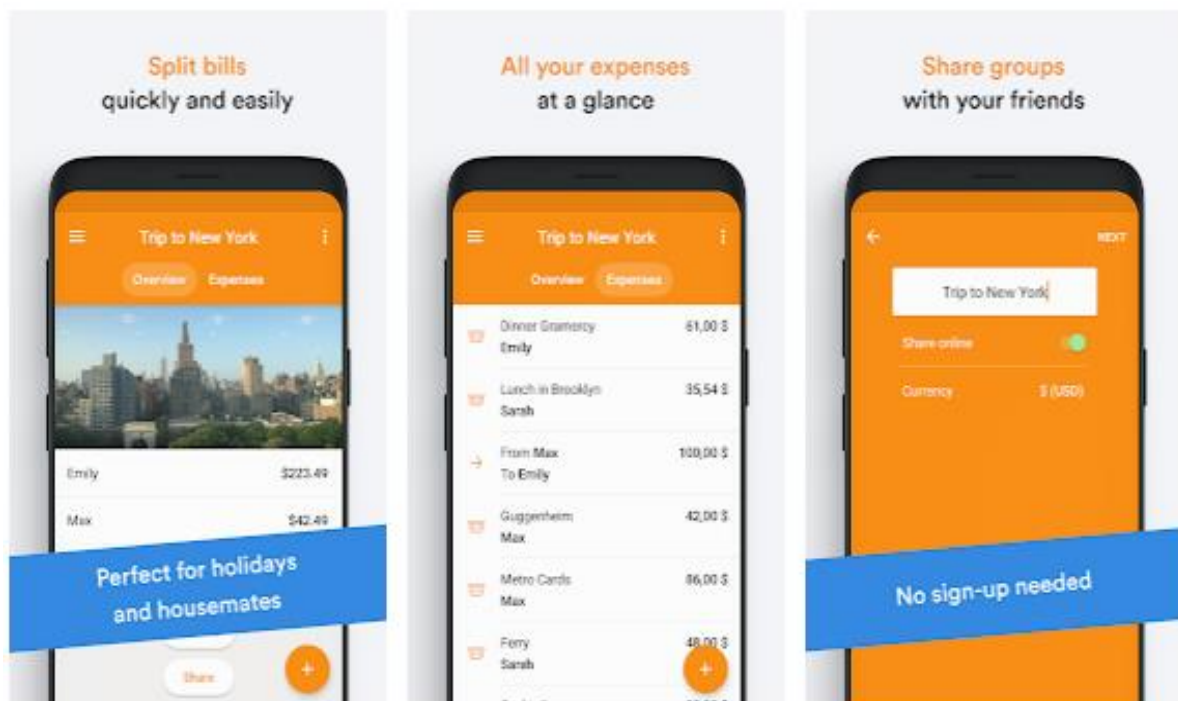
W tej aplikacji wszystkie wydatki są archiwizowane, oraz zsynchronizowane dla grupy osób stworzonej z użytkowników. Aplikacja pozwala na minimalizowanie transakcji oraz na ograniczenie sporów o podział kosztów. Jak oferują twórcy, jest to idealne rozwiązanie dla znajomych, współlokatorów oraz par, niezbędne przy dzieleniu kosztów codziennych, rachunków oraz wydatków np. wakacyjnych. Aplikacja jest obsługiwana przez system Android, IOS i aplikację webową.



Rysunek 6. Aplikacja Settle Up (<https://settleup.io/>)

4.4 Splid

Aplikacja obsługuje skomplikowane transakcje, w około 150 walutach, co ułatwia podział rachunków ze znajomymi. Dodatkowo znajduje najłatwiejszy podział kosztów co minimalizuje liczbę płatności. Aplikacja odznacza się bardzo przejrzystym interfejsem, działa w trybie online jak i offline. Nie wymaga rejestracji. Podlicza koszty z poszczególnych transakcji.



Rysunek 5 Aplikacja Splid (<https://splid.app/>)

5. Narzędzia

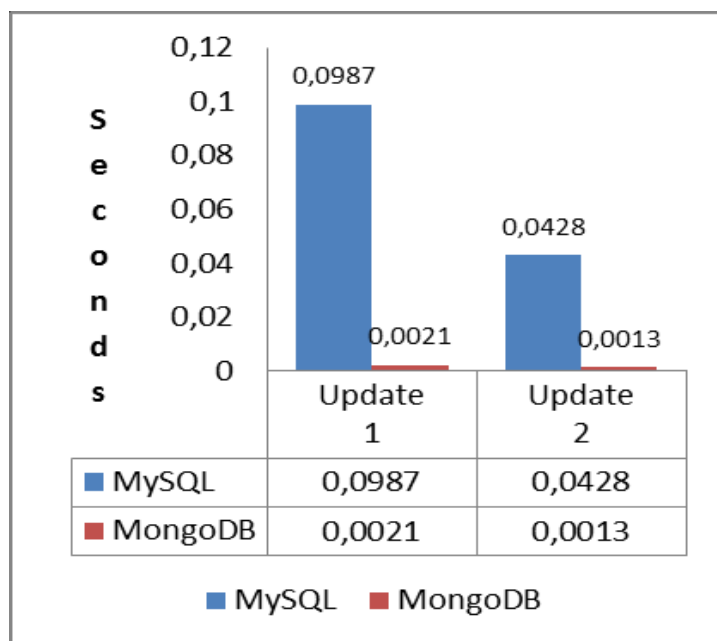
5.1 Serwer

5.1.1 Mongoose

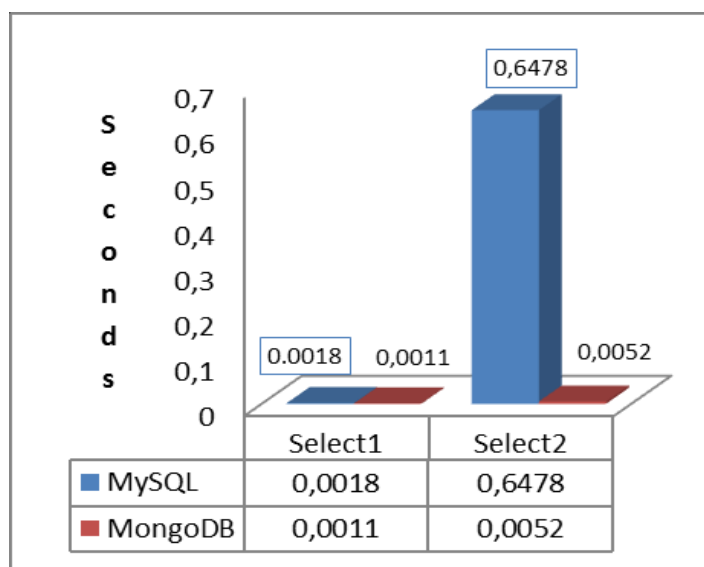
MongoDB (<https://www.mongodb.com/>), jest to nierelacyjny system zarządzania bazą danych, który w przeciwieństwie do relacyjnych systemów nie posiada ustrukturyzowanego sposobu zarządzania danymi. Dane przechowywane są w dokumentach typu BSON, a te umieszczane są w kolekcjach, dzięki czemu wyszukiwanie informacji jest bardzo szybkie,

a dodatkowo przypominają one obiekty znajdujące się w aplikacji. Dodatkową zaletą jest elastyczność, dzięki czemu możemy zmieniać strukturę dokumentów.

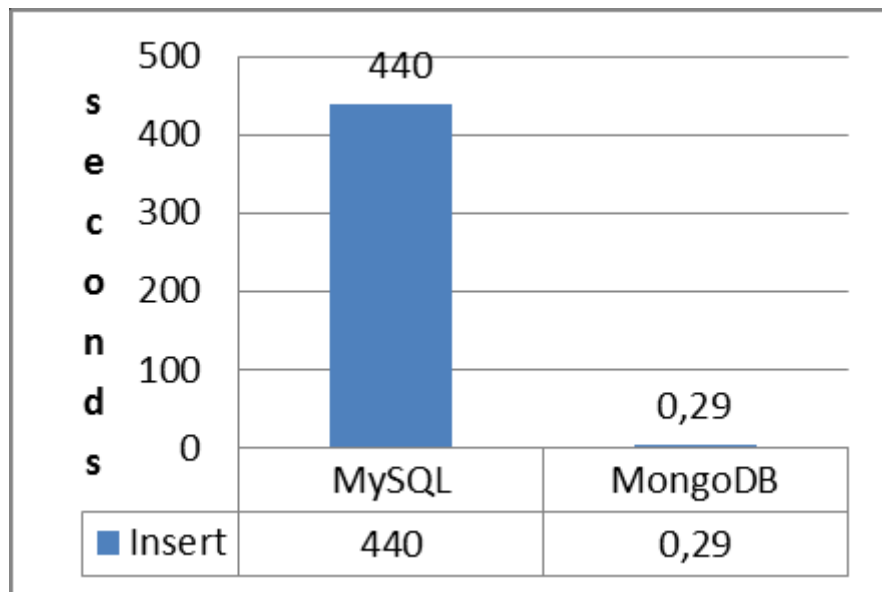
Mongoose (<https://mongoosejs.com/>) jest to natomiast bibliotek ODM do JavaScript ułatwiająca pracę z MongoDB, która daje na możliwość modelowania, walidacji, rzutowania oraz tworzenia populacji, czyli pewnego rodzaju relacji.



Rysunek 7 Porównanie czasu aktualizacji danych w MongoDB i MySQL [16]



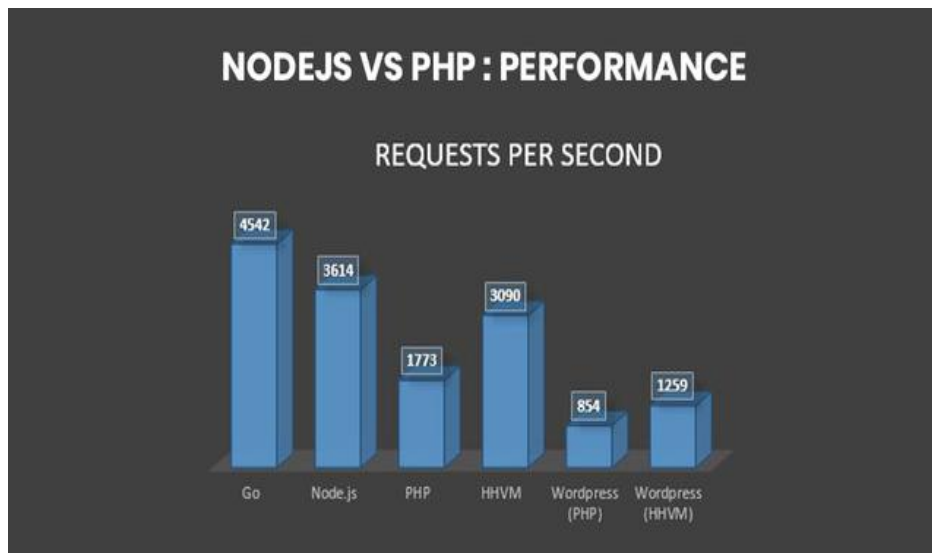
Rysunek 8 Porównanie czasu wyszukiwania danych w MongoDB i MySQL [16]



Rysunek 9 Porównanie czasu zapisu danych w MongoDB i MySQL [16]

5.1.2 ExpressJS

NodeJS (<https://nodejs.org/en/>) jest to środowisko oparte o silnik V8 (<https://v8.dev/>) od Google, które pozwala nam uruchomić kod JavaScript poza przeglądarką internetową. Dodatkowo udostępnia nam API dające nam dostęp do systemu operacyjnego, czyli systemu plików, bibliotek i procesów, dzięki czemu JavaScript może działać jako serwer aplikacji internetowych. Zaletą tego środowiska jest duża wydajność, użycie tego samego języka do napisania klienta i serwera, można zainstalować go na różnych systemach operacyjnych oraz dostęp do bogatego menadżera NPM (<https://www.npmjs.com/>), który dostarcza nam dużo różnych bibliotek, ułatwiających tworzenie aplikacji. ExpressJS (<https://expressjs.com/>) jest to framework routingowy do NodeJS, który służy do obsługi żądań http, wykonywania działań pośrednich (middleware), renderowania widoków oraz daje możliwość konfigurowania ustawień serwera. Jest najpopularniejszym frameworkiem NodeJS.



Rysunek 10 Porównanie wydajności NodeJS i PHP (<https://hackernoon.com/>)

5.1.3 Biblioteki

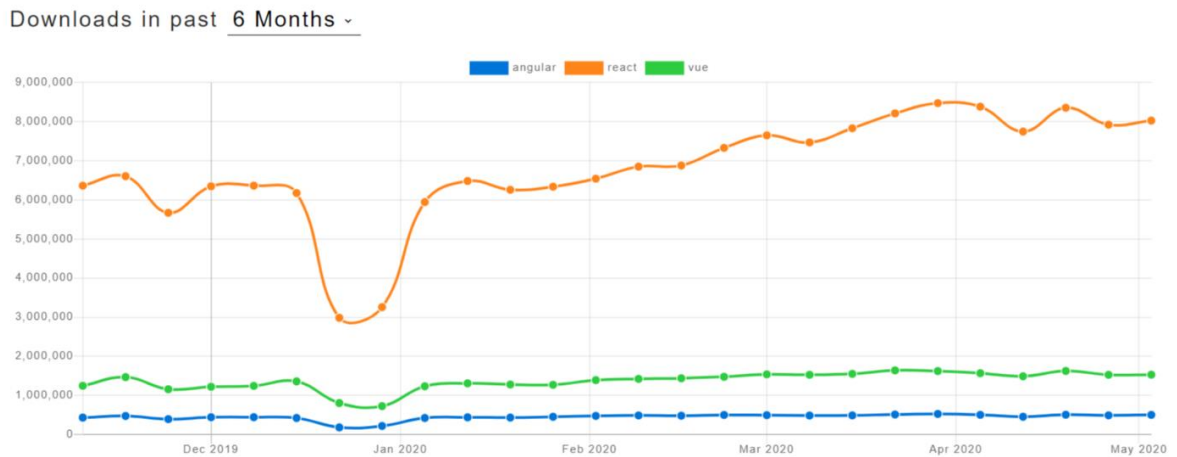
- Dayjs - biblioteka ułatwiająca operowanie datami (edycja, formatowanie, porównywanie)
- Passport – moduł służący do implementacji mechanizmów uwierzytelniających, działający jako middleware, posiadający gotowe schematy uwierzytelniające.
- Bcrypt – biblioteka kryptograficzna
- Nodemailer – biblioteka do obsługi poczty e-mail

5.2 Klient

5.2.1 VueJS

VueJS [11] jest to lekki i progresywny framework do tworzenia dynamicznych interfejsów użytkownika, zarówno w aplikacjach mobilnych, jak i internetowych, łączący takie technologie jak HTML, CSS i JavaScript. Został stworzony w celu usprawnienia procesu wytwórczego aplikacji. Jego twórcą jest Evan You, którego założeniem było wyciągnięcie tego co najlepsze z Reacta (<https://pl.reactjs.org/>) oraz AngularJs (<https://angular.io/>) (dwa najpopularniejsze frameworki), i stworzenie jednego idealnego. Z Angular posiada uproszczone szablony, a z React jednokierunkowy przepływ danych oraz komponenty. Vue posiada bardzo przejrzystą dokumentację oraz ze względu na swoją prostotę ma niski próg wejścia. Aplikacje można budować na dwa sposoby: dodając do standardowego pliku HTML kod Vue lub przy użyciu narzędzia vue-cli, które tworzy

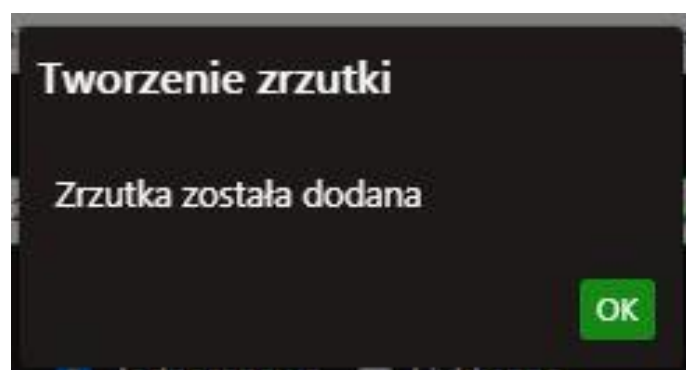
specjalną i bardziej przejrzystą strukturę projektu. Ten framework posiada również vue-router, czyli specjalną bibliotekę do tworzenia aplikacji typu SPA oraz vuex, czyli pewnego rodzaju magazynu do składowania danych.



Rysunek 11 Popularność poszczególnych frameworków (<https://merehead.com/>)

5.2.2 Bootstrap-Vue

Bootstrap (<https://getbootstrap.com/>) jest to specjalny framework autorstwa Twitter do tworzenia responsywnych aplikacji internetowych (RWD). Składa się on ze specjalnych stylów CSS oraz skryptów JavaScript, dzięki czemu można korzystać z gotowych elementów na przykład formularze, obiekty typu karuzela zdjęć lub okna modale. Bootstrap-Vue (<https://bootstrap-vue.org/>) natomiast opakowuje te wszystkie elementy w gotowe komponenty Vue.



Rysunek 12 Przykładowy komponent Bootstrap-Vue (źródło własne)

5.2.3 Biblioteki

- axios – moduł do wykonywania asynchronicznych zapytań http
- dayjs – biblioteka ułatwiająca operowanie datami (edycja, formatowanie, porównywanie)

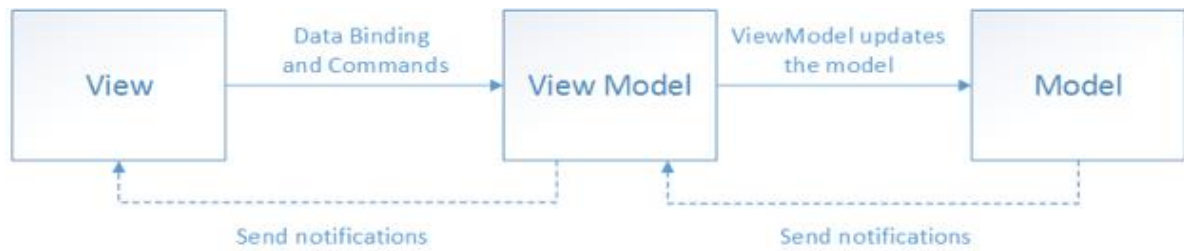
6. Prototyp

6.1 Funkcjonalność

- Aplikacja dostępna na urządzenia mobilne i desktopowe
- Logowanie i rejestracja użytkownika
- Tworzenie i edycja grup
- Dodawanie użytkowników do grupy
- Usuwanie użytkowników z grupy
- Edycja uprawnień użytkowników w grupie
- Tworzenie i edycja zrzutek pojedynczych (między dwoma użytkownikami bez użycia grupy)
- Tworzenie i edycja zrzutek grupowych
- Zrzutki cykliczne, które odnawiają się w określonych odstępach czasowych
- Zrzutki jednorazowe
- Powiadomienia w aplikacji o np. zmianie uprawnień w grupie, dodaniu użytkownika do zrzutki
- Powiadomienia e-mail o kończącym się terminem wpłaty na zrzutkę

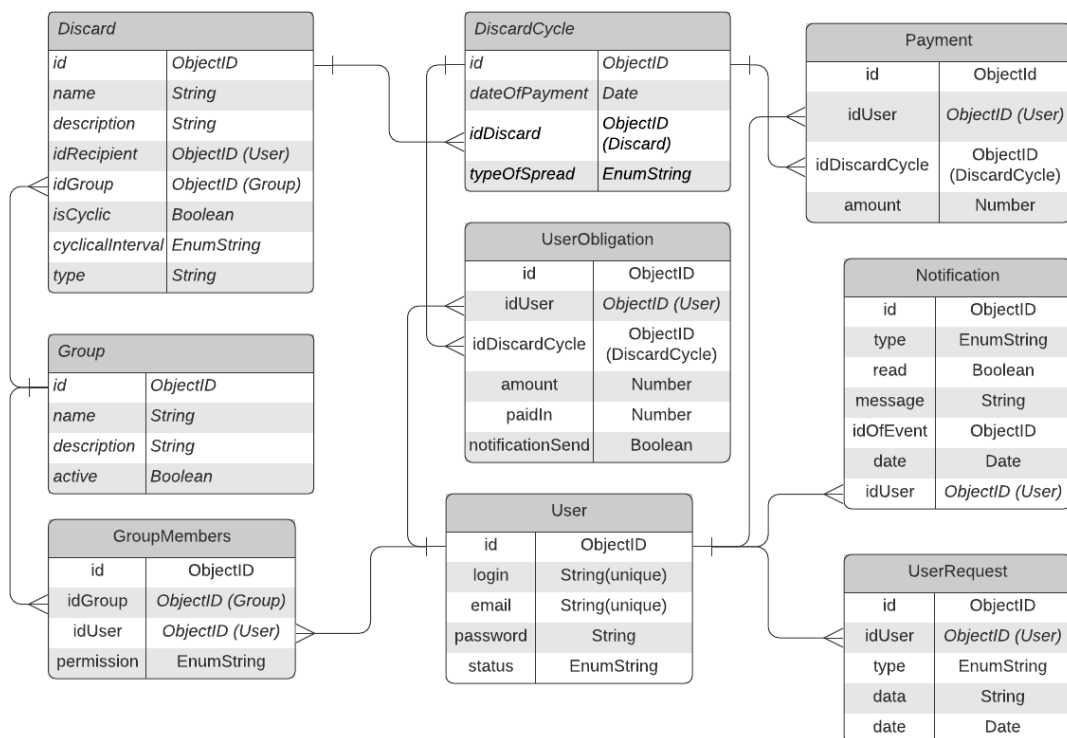
6.2 Wzorzec architektoniczny MVVM

W aplikacji został użyty wzorzec architektoniczny MVVM (Model-View-ViewModel), w którym warstwa prezentacji i warstwa logiki biznesowej zostały odseparowane. Dzięki temu rozwiązaniu aplikacja staje się reaktywna, czyli potrafi reagować na zmianę danych, a kod może być ponownie użyty (reużywalność). MVVM składa się z następujących elementów: model, viewmodel i view. Viewmodel to główny element tej architektury, który przetwarza pobrane dane, łączy je z modelem oraz dostarcza mechanizm odpowiedzialne za interakcję z użytkownikiem. Model natomiast przechowuje dane, które zostały dostarczone przez viewmodel, a view tworzy graficzną reprezentację danych oraz elementy interfejsu służące do interakcji z użytkownikiem np. przyciski.



Rysunek 13 Schemat architektury MVVM [18]

6.3 Struktura bazy danych



Rysunek 14 Schemat bazy danych

Schemat (Rysunek 14 Schemat bazy danych) przedstawia strukturę bazy danych aplikacji, która składa się z następujących kolekcji:

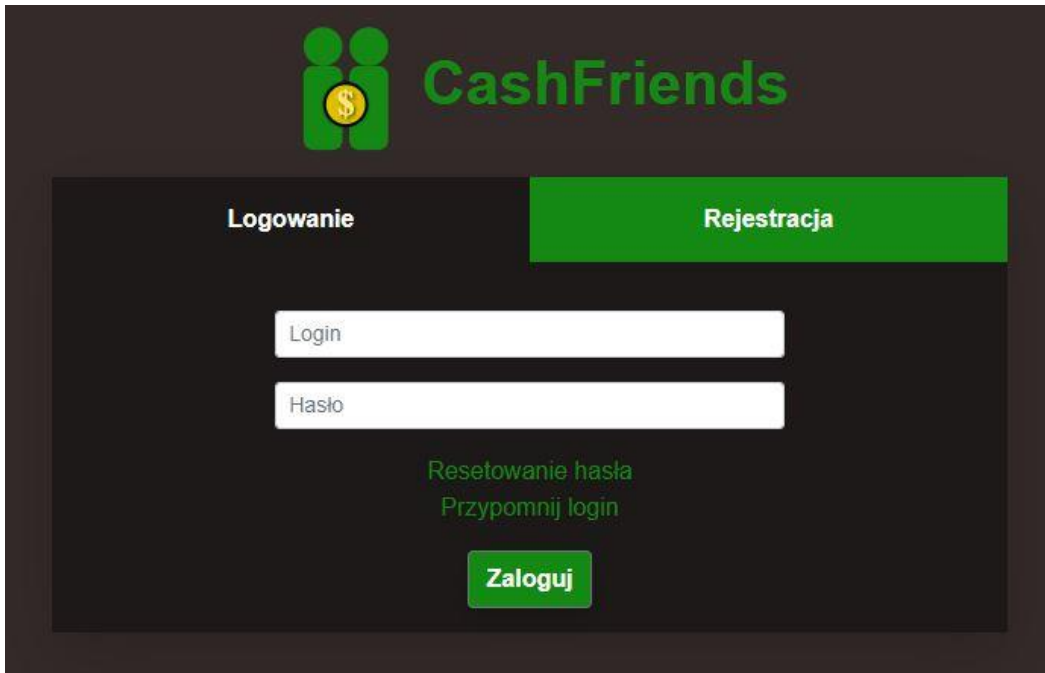
- Discard – przechowuje ona wspólne dane dla każdego cyklu zrzutki, składa się z następujących pól:
 - Name – nazwa zrzutki
 - Description – opis zrzutki
 - idRecipient – id odbiorcy zrzutki, to pole wymagane jest w przypadku zrzutki pojedynczej

- idGroup – id grupy w której została utworzona zrzutka, pole wymagane w przypadku zrzutki grupowej
- isCyclical – czy zrzutka jest cykliczna
- cyclicalInterval – odstęp między poszczególnymi cyklami zrzutki
- type – typ zbiórki, grupowa czy pojedyncza
- DiscardCycle – przechowuje ona informacje o cyklu zrzutki, składa się z:
 - DateOfPayment – ostateczny termin dokonania wpłaty
 - idDiscard – id rekordu z kolekcji Discard
 - typeOfSpread – typ podziału należności
- UserObligation – przechowuje informacje na temat należności użytkownika, zawiera pola:
 - idUser – id użytkownika, którego dotyczy należność
 - idDiscardCycle – id cyklu
 - amount – kwota do zapłaty
 - paidIn – ile zostało wpłacone
 - notificationSend – czy zostało wysłane powiadomienie e-mail informujące o kończącym się terminie zrzutki
- Payment – przechowuje prośby o zatwierdzenie wpłaty, składa się z następujących pól
 - idUser – id użytkownika
 - idDiscardCycle – id zrzutki
 - amount – ile wpłacono
- Group – przechowuje informacje na temat grup:
 - Name – nazwa grupy
 - Description – opis grupy
 - Active – czy grupa jest aktywna
- GroupMembers – przechowuje członków danej grupy wraz z uprawnieniami:
 - idGroup – id grupy
 - idUser – id użytkownika
 - permission – uprawnienia użytkownika posiada następujące wartości: „owner”, „admin”, „member” (zwykły członek grupy), „pending” (zaproszenie nie zostało jeszcze zaakceptowane), „deleted” (usunięty z grupy)

- User – przechowuje użytkowników aplikacji, składa się z pól:
 - Login – nazwa użytkownika
 - Password – hash hasła
 - E-mail – adres poczty elektronicznej użytkownika
 - Status – status konta zawiera trzy opcje: „active”, „notActive” (nie aktywowano konta) i „deleted” (konto usunięte)
- Notification – przechowuje powiadomienia, zawiera pola:
 - Type – typ powiadomienia
 - Read – czy powiadomienie zostało odczytane
 - Message – treść powiadomienia
 - idOfEvent – id elementu do którego odnosi się powiadomienie, np. zrzutka, grupa.
 - Date -data powiadomienia
 - idUser – id użytkownika
- UserRequest – żądania użytkownika np. zmiana hasła, zmiana e-mail, aktywacja konta
 - idUser – id użytkownika
 - type - typ żądania
 - data – dodatkowe informacje potrzebne do wykonania żądania
 - date – data utworzenia żądania

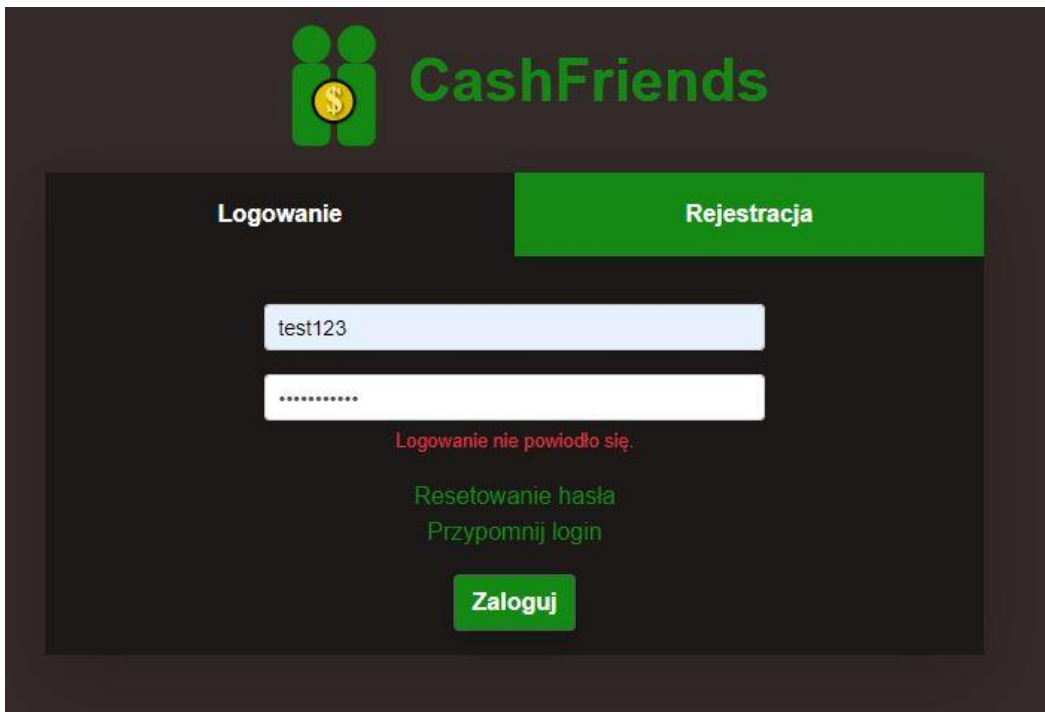
6.4 Interfejs użytkownika

6.4.1 Widok logowania



The screenshot shows the login interface for 'CashFriends'. At the top, there is a logo consisting of two green stylized figures holding a yellow coin with a dollar sign, followed by the text 'CashFriends' in green. Below the logo, there are two tabs: 'Logowanie' (Login) and 'Rejestracja' (Registration). The 'Logowanie' tab is active. Under this tab, there are two input fields: 'Login' and 'Hasło' (Password). Below the password field, there are two links: 'Resetowanie hasła' (Reset password) and 'Przypomnij login' (Remember login). At the bottom, there is a green button labeled 'Zaloguj' (Login).

Rysunek 15 Widok ekranu logowania



This screenshot shows the same login interface as Figure 15, but with an error message. The 'Login' field now contains the text 'test123' and has a light blue background. The 'Hasło' field is empty and has a white background. Below the password field, there is a red error message: 'Logowanie nie powiodło się.' (Login failed). The 'Resetowanie hasła' and 'Przypomnij login' links are still present, and the 'Zaloguj' button remains at the bottom.

Rysunek 16 Widok ekranu logowania po wprowadzeniu nieprawidłowych danych

Po wejściu do aplikacji, gdy użytkownik nie jest zalogowany ukazuje się formularz logowania (Rysunek 15 Widok ekranu logowania) . Aby się zalogować należy wprowadzić poprawne dane (login i hasło) i nacisnąć przycisk „Zaloguj” lub wcisnąć klawisz „Enter”. Jeżeli dane są niepoprawne ukazuje się widok z Rysunek 16 Widok ekranu logowania po wprowadzeniu nieprawidłowych danych.

6.4.2 Widok rejestracji

Rysunek 17 Widok formularza rejestracji

The image shows a web registration form for 'CashFriends'. At the top, there is a logo with two green figures and a yellow coin, followed by the text 'CashFriends'. Below the logo are two tabs: 'Logowanie' (Login) and 'Rejestracja' (Registration). The 'Rejestracja' tab is active. The form contains several input fields with associated labels and validation messages in red text:

- Podaj Login***: A text input field labeled 'Login'. Below it, a red message reads: '- jest zbyt krótki - musi rozpoczynać się od litery lub cyfry - musi kończyć się literą lub cyfrą'.
- Podaj E-mail***: A text input field labeled 'E-mail'. Below it, a red message reads: 'E-mail jest nieprawidłowy'.
- Podaj Hasło***: A text input field labeled 'Hasło'. Below it, a red message reads: '- zbyt krótkie - musi zawierać jedną dużą literę - musi zawierać jedną małą literę - musi zawierać jedną cyfrę'.
- Powtórz Hasło***: A text input field labeled 'Powtórz Hasło'. Below it, a red message reads: 'Hasła są od siebie różne'.
- Avatar**: A section with a text input field labeled 'Wgraj avatar' and a 'Browse' button.

At the bottom of the form, there is a green button labeled 'Zarejestruj'. A small note at the bottom states: 'Pola oznaczone * są obowiązkowe'.

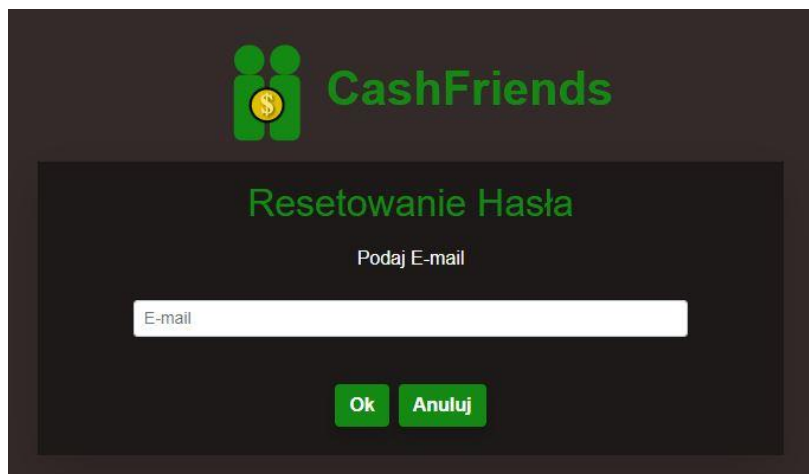
Rysunek 18 Formularz rejestracji po wprowadzeniu niepoprawnych danych

Widok (Rysunek 17) ten ukazuje się po kliknięciu w zakładkę „Rejestracja” w widoku logowania (Rysunek 15). Aby się zarejestrować należy poprawnie wypełnić formularz i nacisnąć przycisk „Zarejestruj”. Po nieprawidłowym wypełnieniu formularza ukaże się widok z Rysunek 18.

Zasady poprawności danych:

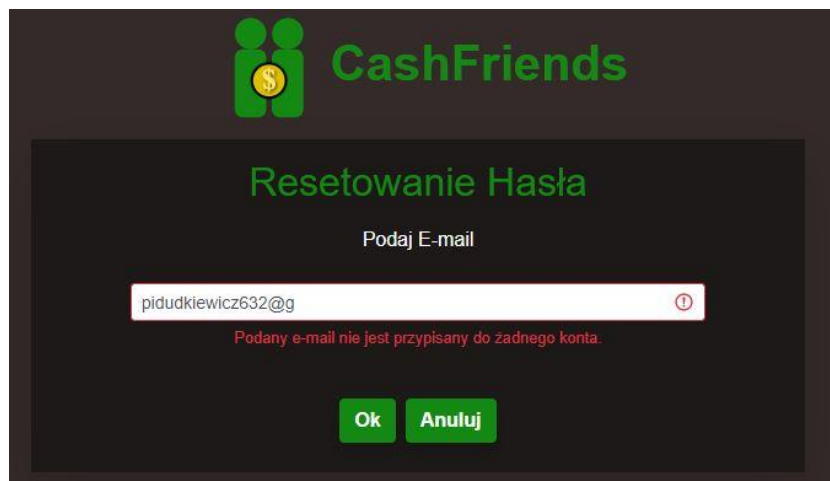
- Login – musi być unikalny, długość od 5 do 21 znaków, musi zaczynać i kończyć się literą lub cyfrą, może zawierać litery i cyfry oraz takie znaki jak: „_” i „-”.
- E-mail – musi być unikalny i być e-mailem
- Hasło – musi zawierać jedną cyfrę, jedną małą literę i jedną dużą literę, minimalna długość to 8 znaków.
- Avatar – pole nieobowiązkowe, maksymalny rozmiar pliku to 1MB, dopuszczalne rozszerzenia to jpg i png.

6.4.3 Widok resetowania hasła



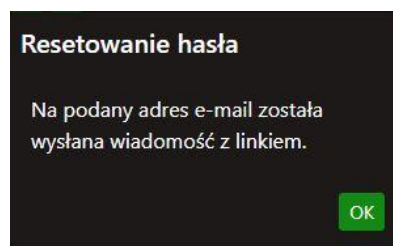
The screenshot shows the 'Resetowanie Hasła' (Reset Password) form for CashFriends. At the top is the CashFriends logo, consisting of two green stylized figures with a yellow dollar sign between them, followed by the text 'CashFriends' in green. Below the logo, the title 'Resetowanie Hasła' is displayed in green. Underneath the title is the instruction 'Podaj E-mail' (Provide E-mail). A white text input field contains the placeholder text 'E-mail'. At the bottom of the form are two green buttons: 'Ok' and 'Anuluj' (Cancel).

Rysunek 19 Widok formularza resetowania hasła



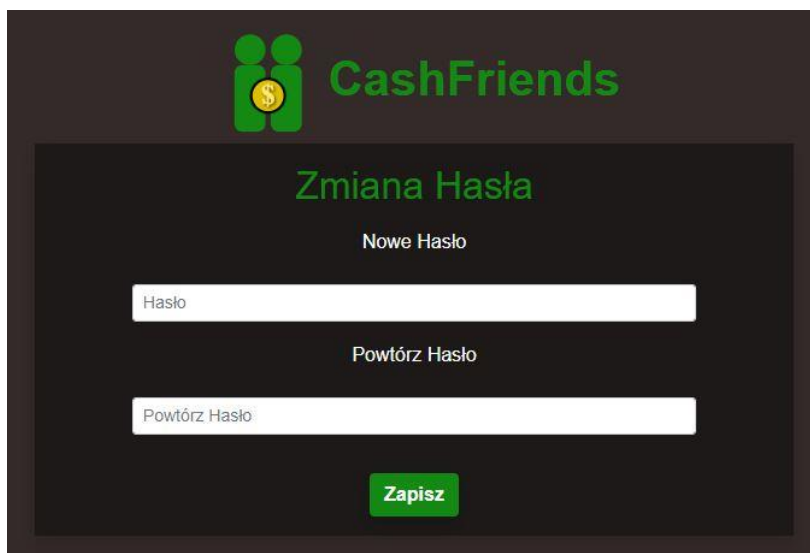
This screenshot shows the same 'Resetowanie Hasła' form as Figure 19, but with an email address entered and an error message. The text input field now contains 'pidudkiewicz632@g' and has a red border and a red information icon on the right. Below the input field, a red error message reads: 'Podany e-mail nie jest przypisany do żadnego konta.' (The provided email is not assigned to any account). The 'Ok' and 'Anuluj' buttons remain at the bottom.

Rysunek 20 Widok po wprowadzeniu nieprawidłowego e-mail



The screenshot shows a small dialog box titled 'Resetowanie hasła' (Reset password). The text inside the dialog reads: 'Na podany adres e-mail została wysłana wiadomość z linkiem.' (A message with a link has been sent to the provided email address). At the bottom right of the dialog is a green button labeled 'OK'.

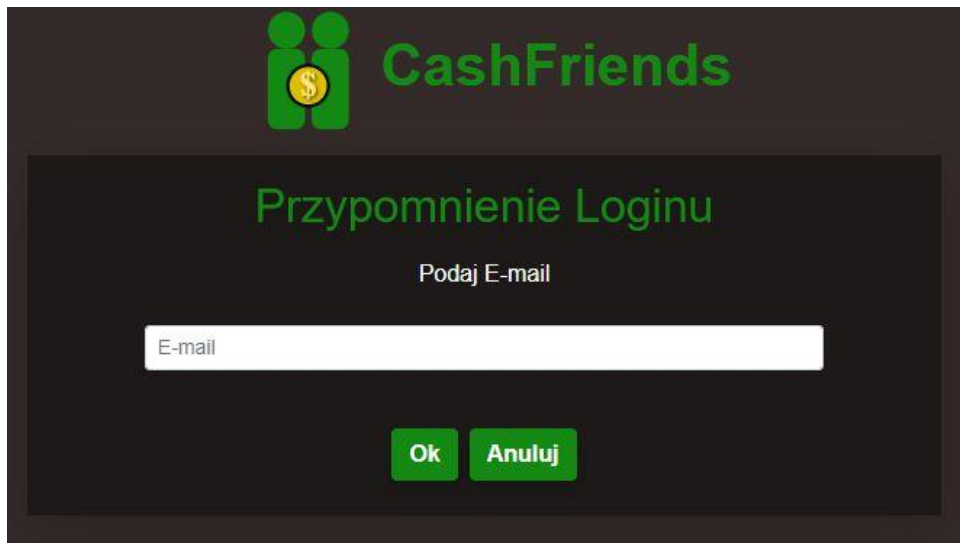
Rysunek 21 Komunikat po poprawnym wypełnieniu formularza

The image shows a web interface for changing a password. At the top, there is a logo consisting of two green stylized figures holding a yellow coin, followed by the text "CashFriends" in green. Below this, the title "Zmiana Hasła" (Change Password) is displayed in green. Underneath the title, the text "Nowe Hasło" (New Password) is shown. This is followed by a white input field with the placeholder text "Hasło". Below this field, the text "Powtórz Hasło" (Repeat Password) is shown. This is followed by another white input field with the placeholder text "Powtórz Hasło". At the bottom of the form, there is a green button with the white text "Zapisz" (Save).

Rysunek 22 Widok zmiany hasła

W widoku resetowania hasła (Rysunek 19) należy podać e-mail przypisany do konta i kliknąć przycisk „Ok”, jeżeli pole zostało poprawnie wypełnione to pojawi się komunikat o wysłaniu wiadomości na wskazany adres (Rysunek 21), a jeżeli błędnie to pojawi się komunikat widoczny na Rysunek 20. Po kliknięciu w link we wiadomości wysłanej na e-mail pojawia się widok zmiany hasła (Rysunek 22 Widok zmiany hasła). Zasada walidacji hasła jest identyczna jak przy rejestracji.

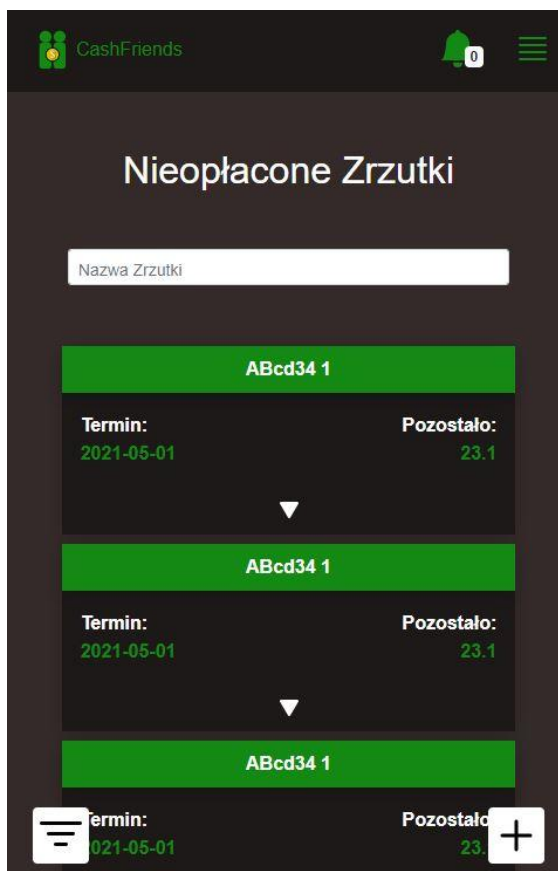
6.4.4 Widok przypomnienia loginu



Rysunek 23 Widok przypominania loginu

Ten widok (Rysunek 23) jest analogiczny do widoku resetowania hasła.

6.4.5 Widok strony głównej



Rysunek 24 Widok strony głównej

Filtrowanie

Data

Od:

Do:

1.05.2021

23.05.2021

Składka

Od:

Do:

12

34

Typ

Wszytkie

Sortowanie

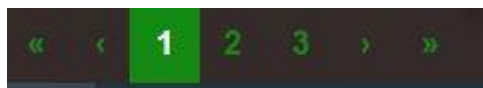
Rosnąco po dacie

Na stronie

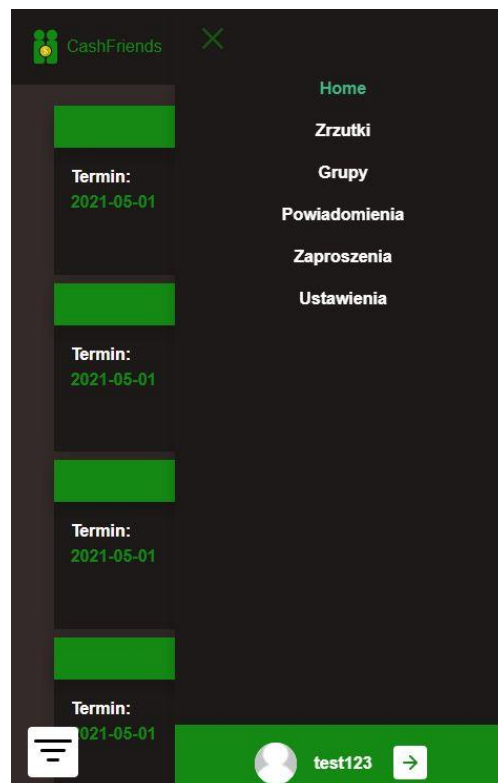
10

Filtruj

Rysunek 25 Okienko filtrowania



Rysunek 26 Stronicowanie



Rysunek 27 Panel boczny



Rysunek 28 Element listy po kliknięciu

Widok strony głównej wyświetla się od razu po zalogowaniu do aplikacji, znajdują się na nim zrzutki, które nie zostały jeszcze opłacone przez użytkownika. Rozwinięcie danej zrzutki powoduje pojawienie się dodatkowej informacji oraz dwóch przycisków (Rysunek 28), które prowadzą do widoku tworzenia płatności lub szczegółów zrzutki. Po kliknięciu w przycisk znajdujący się w lewym dolnym rogu wyświetla się formularz filtrowania (Rysunek 25), który umożliwia filtrowanie danych po dacie, typie oraz kwocie, a także umożliwia wybór sortowania oraz ilość wyświetlanych elementów na stronie. W prawym dolnym rogu znajduje się przycisk dodawania nowej zrzutki, który przekierowuje nas do widoku tworzenia zrzutki. Natomiast w prawym górnym rogu znajdują się dwa przyciski: powiadomień, który wyświetla liczbę nowych powiadomień, a po kliknięciu w niego następuje przekierowanie do widoku powiadomień, oraz przycisk menu, który rozwija panel boczny (Rysunek 27). W panelu bocznym znajduje się menu, avatar, login oraz przycisk wylogowania. Również w tym widoku występuje stronicowanie znajdujące się pod zrzutkami (Rysunek 26), które występują także w innych widokach. Na stronie jest dostępne również wyszukiwanie zrzutki po nazwie.

6.4.6 Widok listy zrzutek

CashFriends

0

Zrzutki

Nazwa

ABcd34 1

Termin: 2021-05-01 Pozostało: 23.1

ABcd34 1

Termin: 2021-05-01 Pozostało: 23.1

ABcd34 1

Termin: 2021-05-01 Pozostało: 23.1

+

Rysunek 29 Widok list zrzutek

Filtrowanie

Data

Od: 1.05.2021 Do: 23.05.2021

Składka

Od: 12 Do: 34

Typ

Wszystkie

Gdzie

Wszystko

Sortowanie

Rosnąco po dacie

Pokaż

☒ Opłacone ☒ Nieopłacone

Na stronie

10

Filtruj

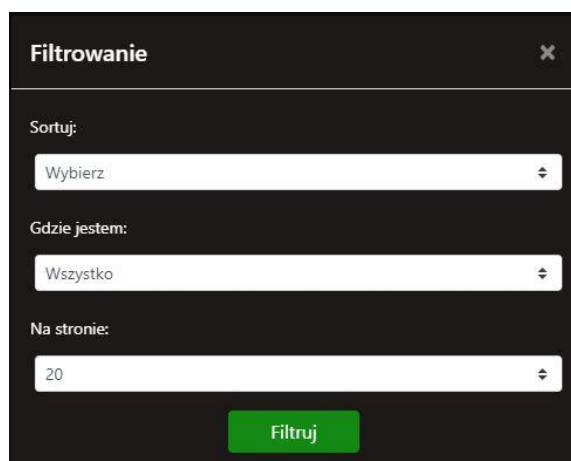
Rysunek 30 Okno formularza filtrowania

W tym widoku dostępna jest lista wszystkich zrzutek, czyli zrzutek opłaconych i nieopłaconych, pojedynczych w których jesteśmy odbiorcą lub dłużnikiem oraz grupowych, w których jesteśmy dłużnikiem lub jesteśmy administratorem grupy z której pochodzi ta zbiórka. Kliknięcie w dany element powoduje pojawienie się dodatkowej informacji oraz przycisków zapłaty i szczegółów zbiórki (Rysunek 28). Nad listą zbiórek wyświetlane jest pole do wyszukiwania po nazwie zbiórki. Na dole znajdują się dwa przyciski: po prawej przekierowujące do widoku tworzenia zbiórki, po lewej wyświetlające okno z formularzem filtrowania (Rysunek 30), które umożliwia na wybranie zrzutek po dacie, składce, typie, gdzie jesteśmy dłużnikami lub odbiorcami, wybór grupy lub użytkownika w zależności od wybranego typu zbiórki, oraz wybór rodzaju sortowania i liczby zrzutek na stronie.

Widok listy grup



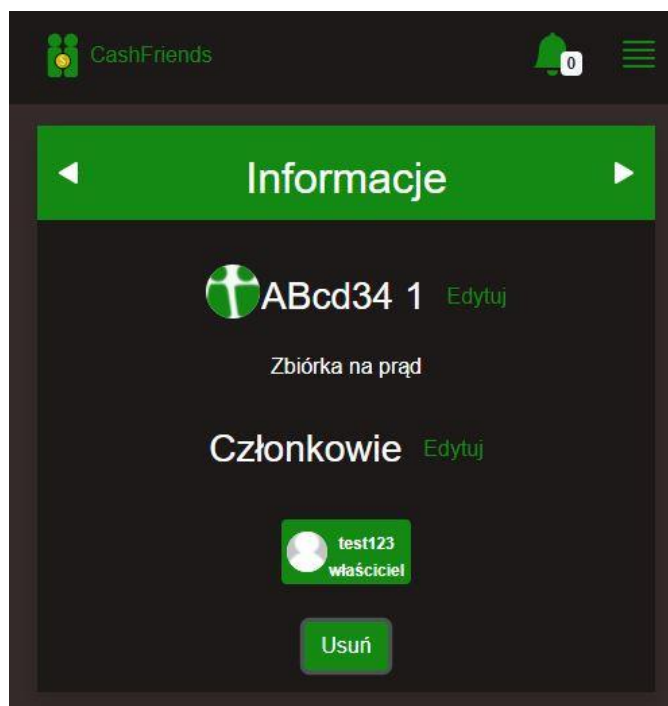
Rysunek 31 Widok listy grup



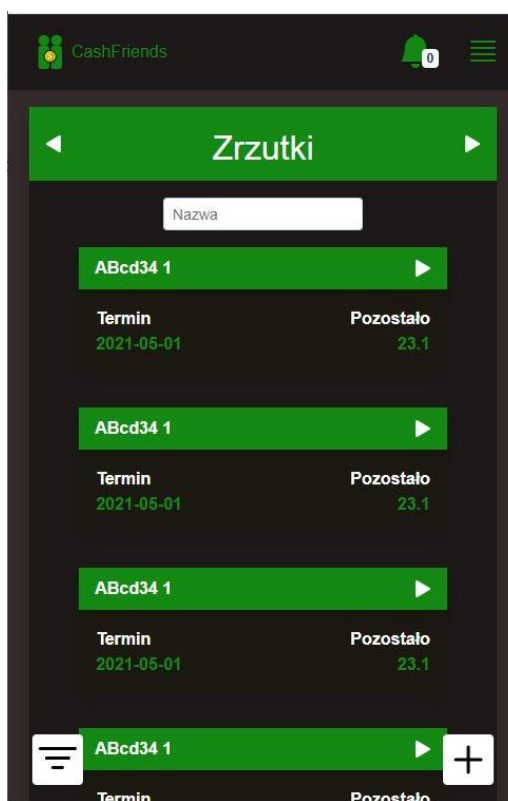
Rysunek 32 Okno filtrowania

W tym widoku wyświetla na jest lista wszystkich grup do których należy użytkownik. Podobnie jak w innych widokach, u góry znajduje się pole do wyszukiwania po nazwie grupy, a na dole przycisk przekierowania do formularza dodawania oraz przycisk wyświetlania okna filtrowania. Po kliknięciu w wybraną grupę następuje przekierowanie do widoku szczegółów grupy (Rysunek 33).

6.4.7 Widok szczegółów grupy



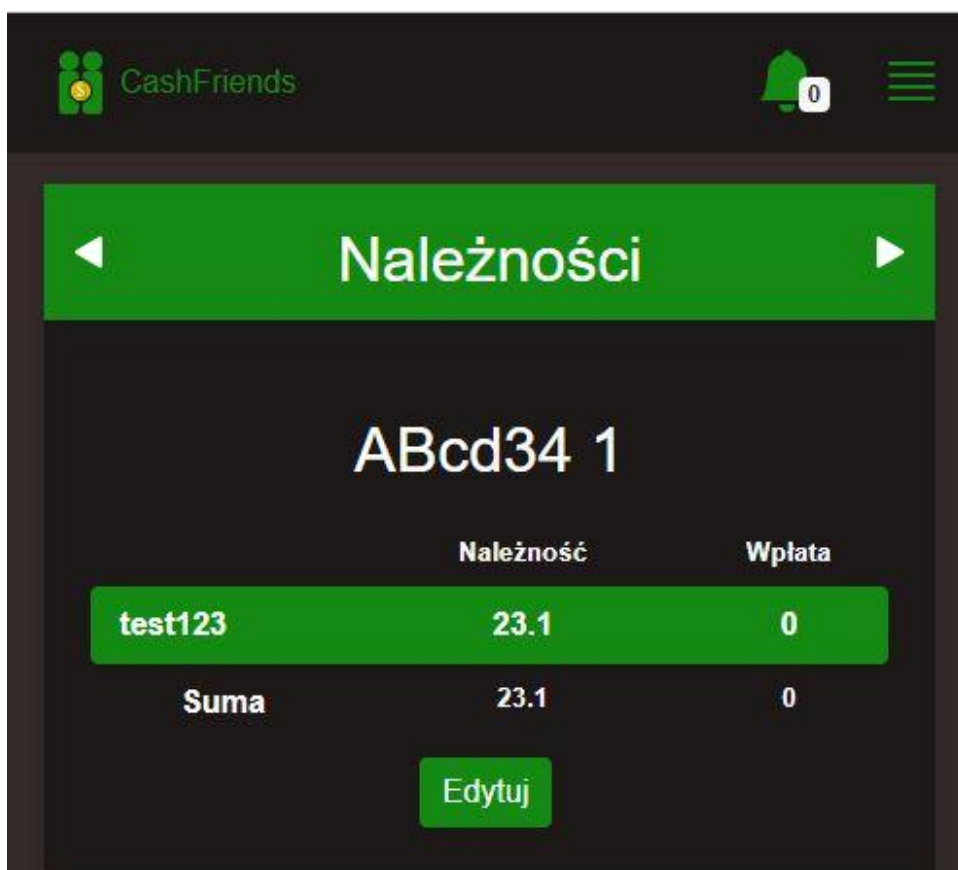
Rysunek 33 Zakładka informacji o grupie



Rysunek 34 Zakładka z zrzutkami znajdującymi się grupie

Widok ten składa się z dwóch zakładek: „Informacje” oraz „Zrzutki”. Pierwsza (Rysunek 33) z nich wyświetla nazwę, opis, logo, członków grupy oraz przyciski do edycji informacji, edycji członków oraz usunięcia grupy (przyciski dostępne dla właściciela oraz administratora). Drugi zaś (Rysunek 34) wyświetla listę zrzutek znajdujących się w grupie, dla administratora i właściciela będą to wszystkie zrzutki, a dla zwykłego członka będą to zrzutki w których bierze udział. Przyciski znajdujące się na dole służą do dodawania nowej zrzutki w grupie (dostępne tylko dla administratora lub właściciela) oraz wyświetlania okna z formularzem filtrowania, które działa analogicznie do formularza znajdującego się w widoku listy zrzutek (Rysunek 30).

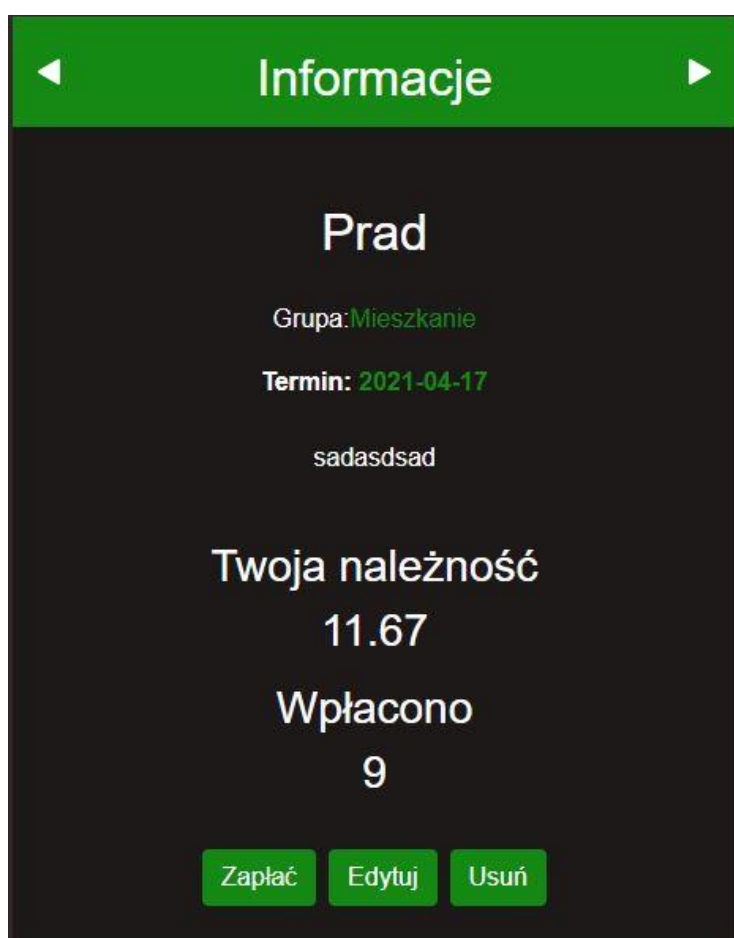
6.4.8 Widok szczegółów grupowej zrzutki



Rysunek 35 Zakładka należności



Rysunek 36 Zakładka z wpłatami

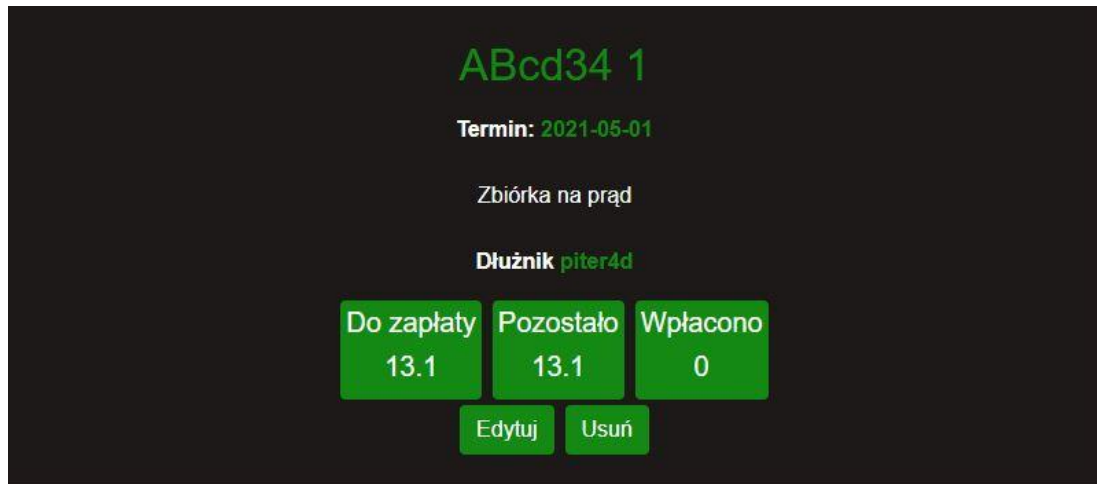


Rysunek 37 Zakładka z informacjami

Ten widok składa się z trzech zakładek: „Informacje”, „Należności” (dostępna dla administratora i właściciela) oraz „Wpłaty” (dostępna dla administratora i właściciela). Pierwsza z nich (Rysunek 37) zawiera informacje takie jak nazwa, grupa, opis, należność użytkownika, ile wpłacono, przyciski zapłaty, edycji oraz usuwania (edycja i usuwanie dostępne dla administratora i właściciela). Druga zakładka (Rysunek 35) zawiera informacje

o należnościach i wpłatach oraz przycisk do edycji, a trzecia zakładka (Rysunek 36) zawiera prośby potwierdzenia wpłaty.

6.4.9 Widok szczegółów pojedynczej zrzutki



ABcd34 1

Termin: 2021-05-01

Zbiórka na prąd

Dłużnik piter4d

Do zapłaty	Pozostało	Wpłacono
13.1	13.1	0

Edytuj Usuń

Rysunek 38 Widok szczegółów pojedynczej zrzutki

Widok ten zawiera szczegółowe informacje (nazwa, termin, opis, informacje o wpłacie) na temat zbiórki oraz przyciski, dla odbiorcy będą to edycja i usuwanie, a dla dłużnika przycisk zapłaty.

6.4.10 Widok tworzenia grupy

CashFriends

Dodawanie Grupy

Nazwa

Opis

Logo

Wybierz plik Nie wybrano pliku

Dodawanie Użytkowników

Rysunek 39 Widok formularza tworzenia grupy

Dodawanie Grupy

Nazwa

Długość od 3 do 40 znaków.

Opis

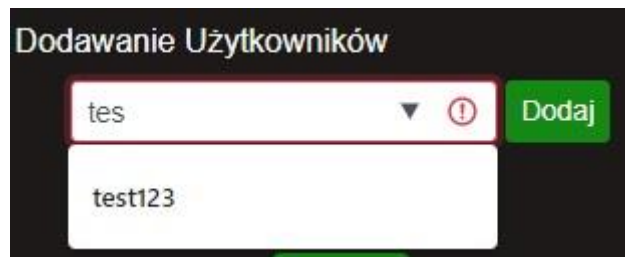
Logo

Wybierz plik Nie wybrano pliku

Dodawanie Użytkowników

Nie ma takiego użytkownika.

Rysunek 40 Widok nieoprawnej walidacji



The image shows a web interface for adding users. At the top, the title 'Dodawanie Użytkowników' is displayed. Below it is a form with a text input field containing the text 'tes'. To the right of the input field is a green button labeled 'Dodaj'. Below the input field is a suggestion box that displays 'test123'. The entire form is set against a dark background.

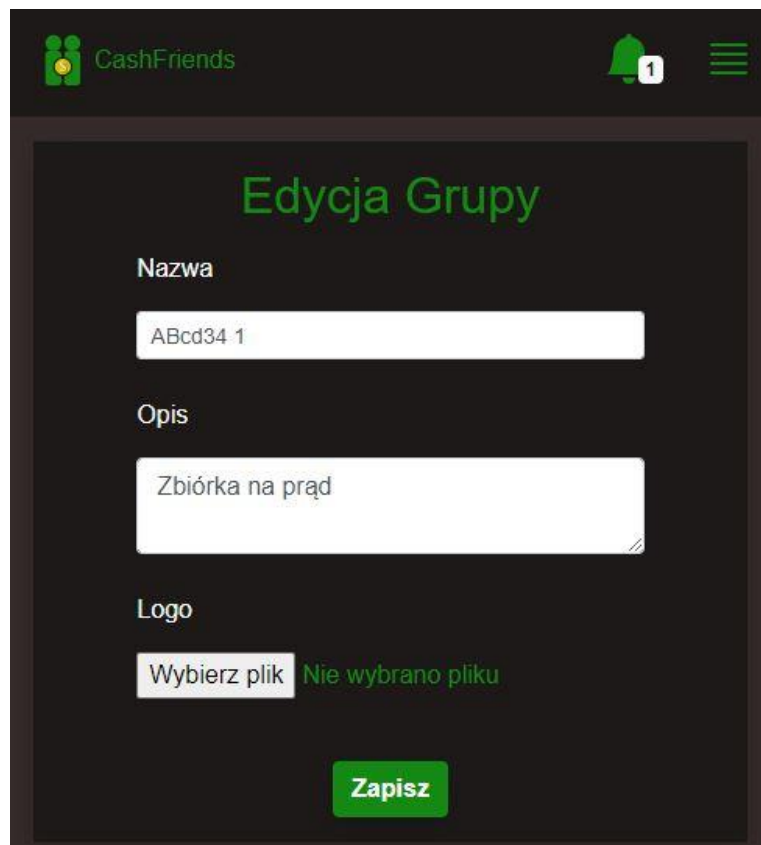
Rysunek 41 Pole do dodawania użytkownika

Widok ten zawiera formularz tworzenia grupy, który zawiera pola takie jak:

- Nazwa – długość od 3 do 40 znaków, dozwolone litery, cyfry oraz spację.
- Opis – pole nieobowiązkowe, maksymalna długość to 100 znaków, Dozwolone litery, cyfry, spację, enter oraz (- , () ! ?).
- Logo - pole nieobowiązkowe, maksymalny rozmiar pliku to 1MB, dopuszczalne rozszerzenia to jpg i png.
- Dodawanie użytkowników – pole nieobowiązkowe, wyświetla proponowanych użytkowników (Rysunek 41) po wpisaniu trzech znaków.

Po wprowadzeniu nieprawidłowych informacji wyświetlają się komunikaty widoczne na Rysunek 40. Po poprawnym wypełnieniu formularza należy kliknąć przycisk „Dodaj”, jeżeli proces dodawania grupy przebiegł pomyślnie to wyświetli się okienko z komunikatem.

6.4.11 Widok edycji informacji o grupie



CashFriends

1

Edycja Grupy

Nazwa

ABcd34 1

Opis

Zbiórka na prąd

Logo

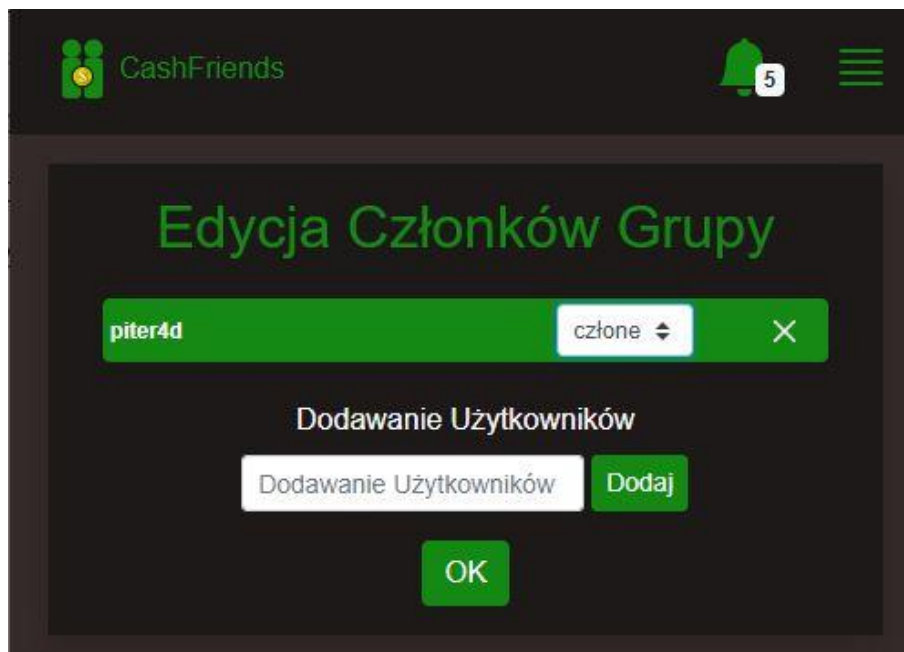
Wybierz plik Nie wybrano pliku

Zapisz

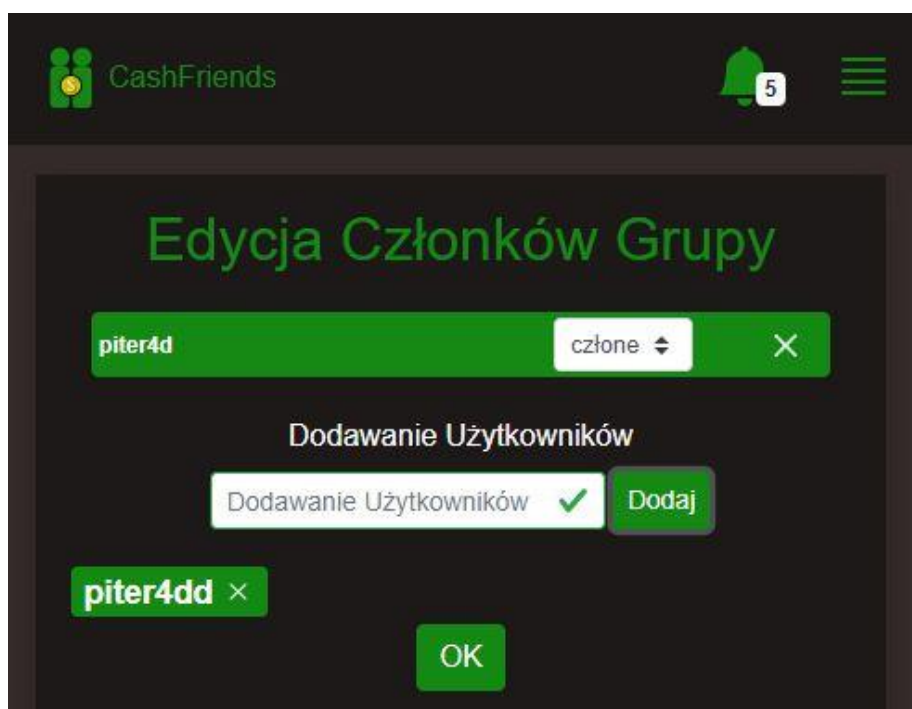
Rysunek 42 Widok formularza edycji grupy

Widok działa analogicznie do formularza tworzenia grupy, różnią się tylko brakiem pola dodawania użytkowników oraz wypełnionym formularzem aktualnymi informacjami o grupie.

6.4.12 Widok edycji członków grupy i uprawnień



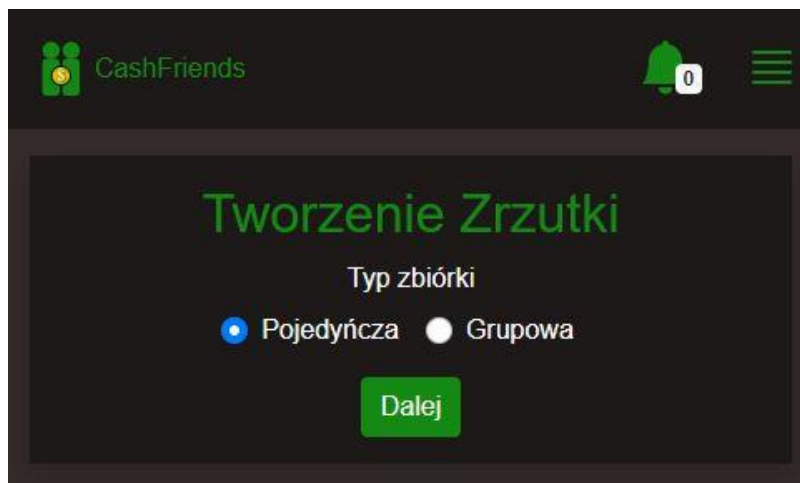
Rysunek 43 Widok edycji członków grupy



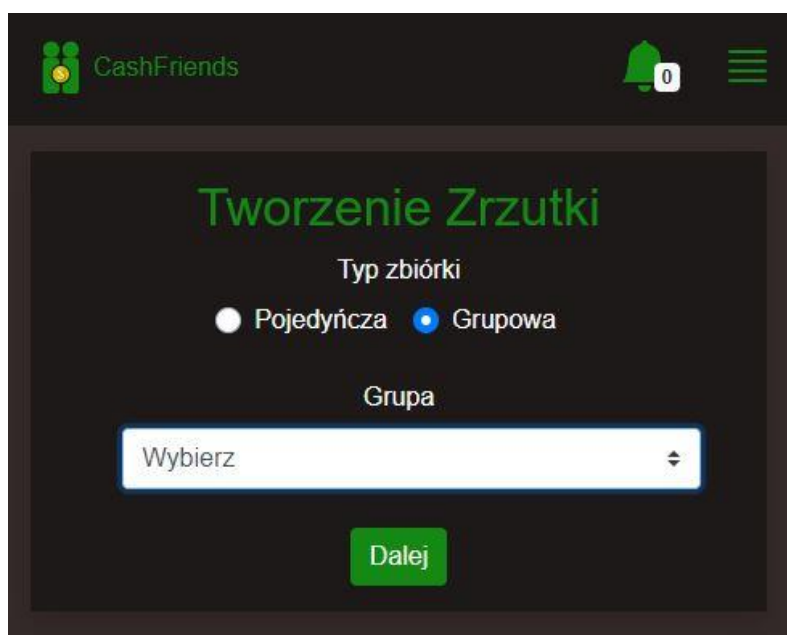
Rysunek 44 Widok po wprowadzeniu nowego użytkownika

Widok ten pozwala na usuwanie, przywracanie, zmianę uprawnień oraz dodawaniu użytkowników. Pole dodawania użytkowników działa podobnie jak w formularzu tworzenia grupy (Rysunek 41). Po wprowadzeniu zmian należy kliknąć przycisk „Ok”.

6.4.13 Widok wyboru rodzaju tworzonej zbiórki



Rysunek 45 Widok wyboru rodzaju tworzonej zbiórki



Rysunek 46 Widok po wybraniu grupowej zrzutki

Ten widok jest wyświetlany po naciśnięciu przycisku „dodaj” na stronie głównej lub widoku listy wszystkich zrzutek. Służy on do wyboru rodzaju tworzonej zbiórki. Jeżeli wybierzemy zbiórkę grupową to należy wybrać grupę (Rysunek 46). Po naciśnięciu przycisku „dalej” następuje przekierowanie do odpowiedniego formularza dodawania zrzutki.

6.4.14 Widok tworzenia grupowej zrzutki

The screenshot shows the 'Tworzenie zrzutki' (Creating a fundraiser) screen in the CashFriends app. The header includes the CashFriends logo, a notification bell with '0', and a menu icon. The main title is 'Tworzenie zrzutki' in green. Below it is the section 'Informacje' (Information). The form contains the following fields:

- Nazwa*** (Name): A text input field with the placeholder 'Prąd' (Current).
- Wybierz datę płatności z listy*** (Choose payment date from list): A dropdown menu with the selected option 'Za dwa tygodnie' (In two weeks).
- lub podaj własną** (or provide your own): A label for the next date field.
- 11.05.2021**: A date input field with a calendar icon.
- Opis** (Description): A text input field with the placeholder 'Opis'.

Below the description field, there is a note: 'Można też tu umieścić opis sposobu płatności' (You can also put a description of the payment method here). At the bottom, a note states: 'Pola oznaczone * są obowiązkowe' (Fields marked * are mandatory). A green 'Dalej' (Next) button is at the bottom center.

Rysunek 47 Krok pierwszy tworzenia grupowej zbiórki

The screenshot shows the 'Tworzenie zrzutki' (Creating a fundraiser) screen in the CashFriends app, specifically the 'Typ Zrzutki' (Type of Fundraiser) section. The header is identical to the previous screen. The main title is 'Typ Zrzutki'. The form contains the following elements:

- Cykliczna** (Recurring): A radio button that is selected.
- Jednorazowa** (One-time): A radio button that is not selected.
- Odnawiaj co** (Renew every): A label for the dropdown menu.
- Wybierz opcję** (Choose an option): A dropdown menu.

At the bottom, there are two green buttons: 'Dalej' (Next) and 'Wstecz' (Back).

Rysunek 48 Krok drugi

CashFriends

Tworzenie zrzutki

Dodawanie użytkowników

Wybierz użytkowników

☐ piter4d

☐ wszyscy

Dalej Wstecz

Rysunek 49 Krok trzeci

Tworzenie zrzutki

Przydział składek

Rozłożenie kwoty

Kwota

0

Każdy użytkownik będzie musiał zapłacić 0

Zakończ Wstecz

Rysunek 50 Przydział składek - rozłożenie kwoty

Tworzenie zrzutki

Przydział składek

Określona składka

Kwota

0

Suma: 0

Zakończ Wstecz

Rysunek 51 Przydział składek – określona składka

Tworzenie zrzutki

Przydział składek

Ręczne

piter4d

Suma: 0

Zakończ Wstecz

Rysunek 52 Przydział składek – ręczny przydział

Proces tworzenia zrzutki grupowej jest został podzielony na cztery widoki:

- Pierwszy „Informacje” (Rysunek 47) należy wypełnić takie pola jak:
 - Nazwa – długość od 3 do 40 znaków, dozwolone litery, cyfry oraz spację.
 - Data – można wybrać gotową datę z listy (za tydzień, dwa tygodnie, miesiąc, dwa miesiące, pół roku lub rok) lub wprowadzić własną, data nie może być większa niż rok oraz mniejsza od aktualnej daty.

- Opis – pole nieobowiązkowe, maksymalna długość to 150, dozwolone litery, cyfry oraz spację, enter oraz (- , () ! ?).
- Drugi „Typ zrzutki” (Rysunek 48) zawierający opcję:
 - Cykliczna – zrzutka, która odnawia się co określony czas (tydzień, dwa tygodnie, miesiąc, dwa miesiące, pół roku lub rok), należy go wybrać z pola wyboru.
 - Jednorazowa – zrzutka ta odbywa się tylko raz.
- Trzeci „Dodawanie użytkowników” (Rysunek 49) – w którym należy wybrać przynajmniej jednego użytkownika
- Czwarty „Przydział składek”, który zawiera cztery opcję:
 - Rozłożenie kwoty (Rysunek 50) – kwota podana w polu jest dzielona na wszystkich użytkowników np. jeżeli podamy w polu kwotę 50, a liczba użytkowników biorących udział w zrzutce wynosi 5, to każdemu z nich zostanie przypisana kwota 10.
 - Określona składka (Rysunek 51) – kwota podana w polu zostanie przypisana każdemu użytkownikowi, np. jeżeli podamy 10 to każdemu zostanie przypisana kwota wynosząca 10.
 - Ręczny (Rysunek 52 Przydział składek – ręczny przydział) – każdemu użytkownikowi przypisujemy inną kwotę.

Kwota przypisana każdemu użytkownikowi musi wynosić przynajmniej 1.

Po poprawnym wypełnieniu ostatniego pola i naciśnięciu przycisku „Zakończ” tworzona jest zrzutka oraz wyświetla się komunikat.

6.4.15 Widok tworzenia pojedynczej zbiórki

Tworzenie pojedynczej zbiórki

Nazwa*

Opis

Można też tu umieścić opis sposobu płatności

Wybierz datę płatności z listy*

Wybierz

lub podaj własną

Termin płatności

Typ*

☒ Jednorazowa ☐ Cykliczna

Jestem*

☒ Odbiorcą ☐ Spłacającym

Spłacający

Kwota*

0

Pola oznaczone * są wymagane

Dodaj

Rysunek 53 Widok tworzenia pojedynczej zbiórki

Widok ten zawiera formularz dodawania zbiórki pojedynczej, który zawiera następujące pola:

- Nazwa – od 3 do 40 znaków, dozwolone litery, cyfry oraz spacje.
- Opis - pole nieobowiązkowe, maksymalna długość to 150, dozwolone litery, cyfry oraz spację, enter oraz (- , () ! ?).
- Data - można wybrać gotową datę z listy (za tydzień, dwa tygodnie, miesiąc, dwa miesiące, pół roku lub rok) lub wprowadzić własną, data nie może być większa niż rok oraz mniejsza od aktualnej daty.

- Typ – po wybraniu cyklicznej pojawia się pole z wyborem co jaki czas ma odnawiać się zrzutka (opcję te same co w zrzutce grupowej)
- Jestem – wybór czy użytkownik jest odbiorcą czy spłacającym
- Spłacający/Odbiorca – pole działające analogicznie do pola z formularzu dodawania grupy (Rysunek 41)
- Kwota – nie może być mniejsza niż 1.

Jeżeli formularz zostanie wypełniony niepoprawnie to wyświetlają się na czerwono odpowiednie komunikaty, natomiast po poprawnym wypełnieniu i naciśnięciu przycisku „dodaj”, zostanie utworzona zbiórka oraz pojawi się okienko z komunikatem

6.4.16 Widok edycji pojedynczej zrzutki

Edycja zrzutki

Nazwa*

dasda

Opis

Mozna też tu umieścić opis sposobu płatności

Wybierz datę płatności z listy*

lub podaj własną

25.05.2021

Typ*

☒ Jednorazowa ☐ Cykliczna

Kwota*

34

Wpłacono

0

Pola oznaczone * są wymagane

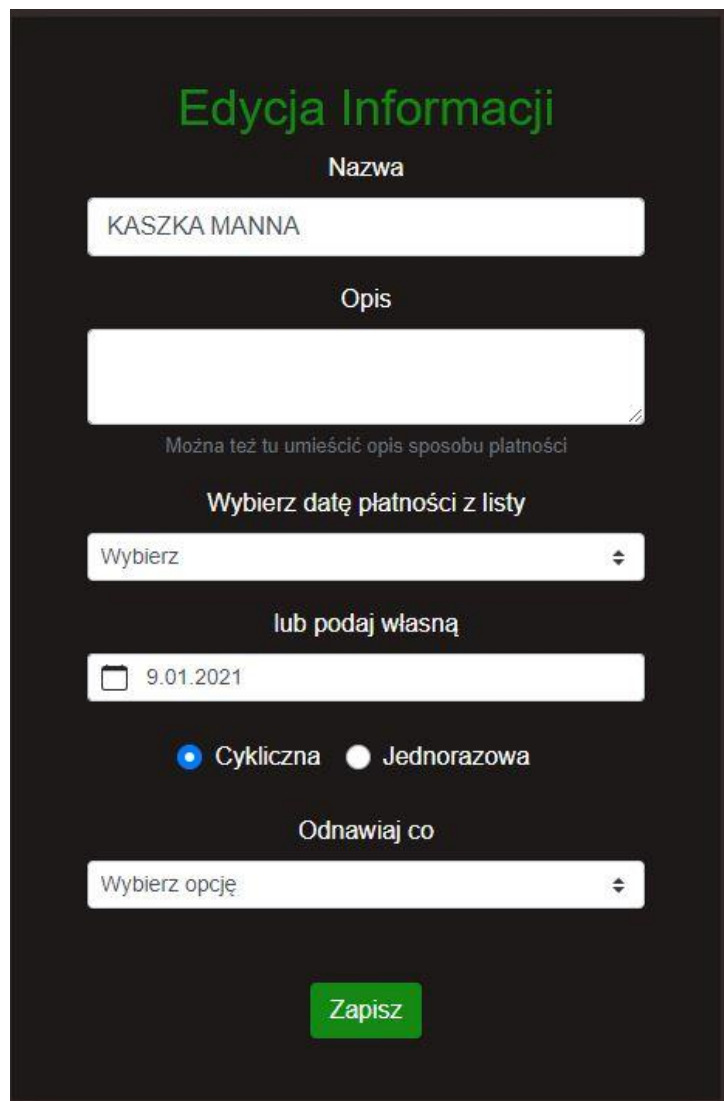
Zapisz

Rysunek 54 Widok edycji pojedynczej zrzutki

Formularz jest podobny do tego z tworzenia zrzutki, różni się jedynie:

- Brakiem wyboru użytkownika
- Polem „Wpłacono”, gdzie podajemy kwotę, którą wpłacił dłużnik, nie może być ona mniejsza od zera.
- W polu data nie można wprowadzić mniejszej daty od edytowanej.

6.4.17 Widok edycji informacji o zrzutce grupowej



Edycja Informacji

Nazwa

KASZKA MANNA

Opis

Można też tu umieścić opis sposobu płatności

Wybierz datę płatności z listy

Wybierz

lub podaj własną

9.01.2021

☒ Cykliczna ☐ Jednorazowa

Odnawiaj co

Wybierz opcję

Zapisz

Rysunek 55 Formularz edycji informacji o zrzutce grupowej

W tym formularzu walidacja jest podobna do tej z procesu tworzenia, jedyną różnicą jest to, że data nie może być mniejsza od edytowanej daty.

6.4.18 Widok edycji należności



Edycja Należności

KASZKA MANNA

Rozłożenie kwoty

Kwota

6,66

	Należność	Wpłata	
 piter4dd	3.33	<div>0</div>	×
 piter4ddd	3.33	<div>0</div>	×
Suma:		6.66	

piter4ddddd

piter4d

Dodaj

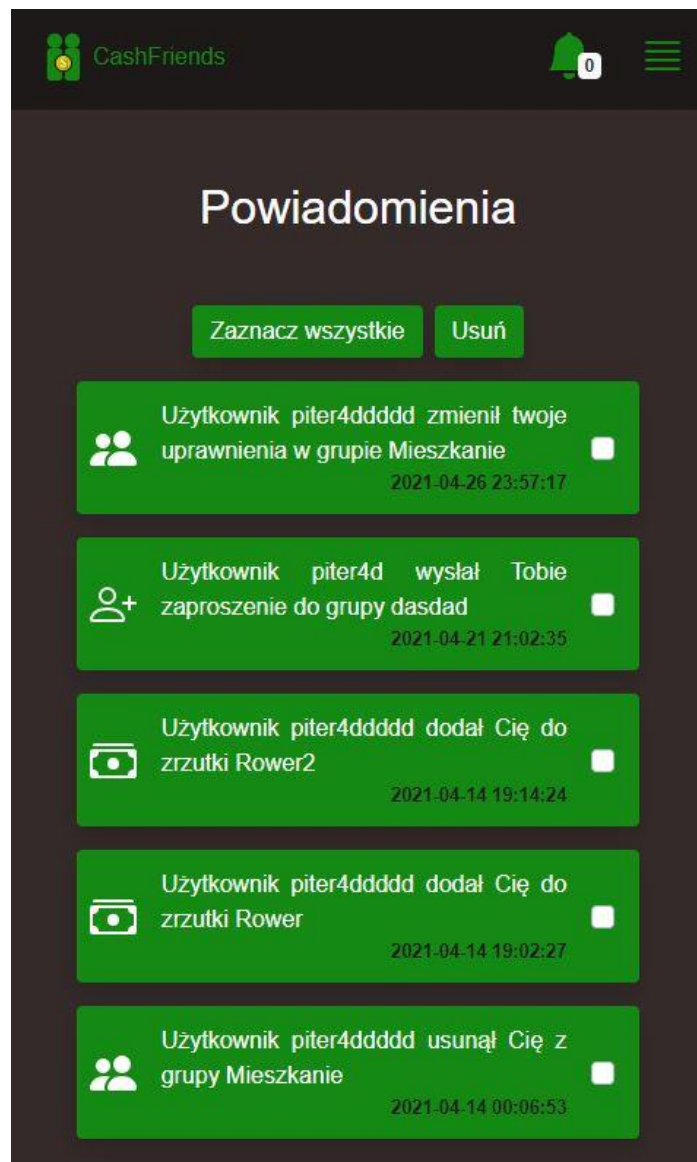
Zapisz

Wyczyść Pola

Rysunek 56 Widok edycji należności

Widok edycji należności działa podobnie do etapu czwartego („Przydział składek”) tworzenia zrzutki grupowej, różnica polega na tym, że zawiera dodatkowo możliwość edycji wpłat (nie może być mniejsza od 0) oraz możliwość usunięcia lub dodania użytkownika.

6.4.19 Widok listy powiadomień



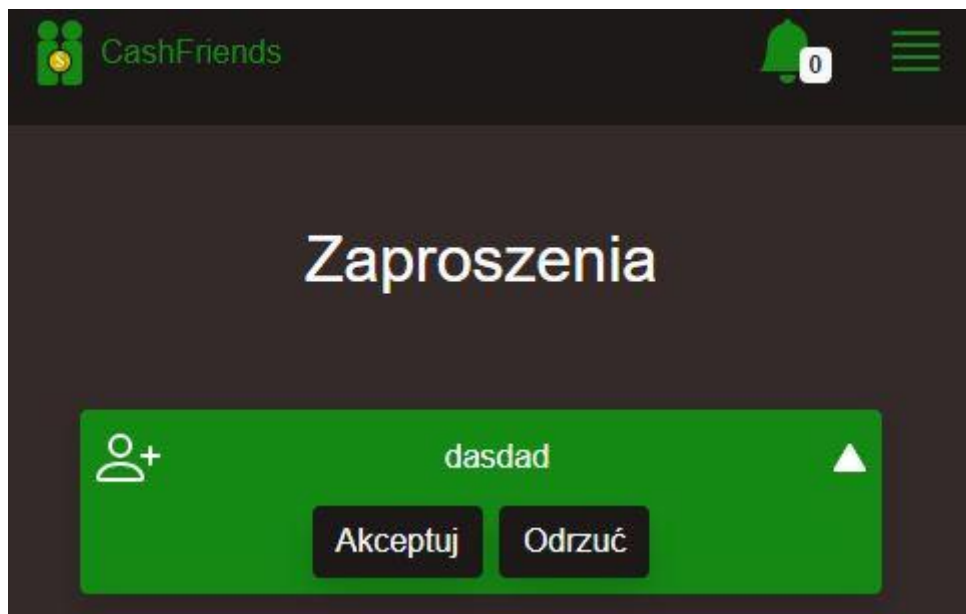
Rysunek 57 Widok listy powiadomień



Rysunek 58 Wyświetlone powiadomienie

Widok ten zawiera listę powiadomień odczytanych i nieodczytanych, nieodczytane mają kolor zielony, a odczytane czarny (Rysunek 58 Wyświetlone powiadomienie). U góry znajdują się dwa przyciski, po lewej do zaznaczenia wszystkich elementów, a po prawej do usunięcia zaznaczonych. Pod listą znajduje się stronicowanie oraz wybór ilość wyświetlanych elementów na stronie.

6.4.20 Widok listy zaproszeń



Rysunek 59 Widok listy zaproszeń

Widok ten zawiera listę zaproszeń do grup, każdy element listy jest rozwijalny i zawiera dwa przyciski, jeden do akceptacji, a drugi do odrzucenia zaproszenia. Po akceptacji/odrzućeniu zaproszenia pojawia się okno z komunikatem.

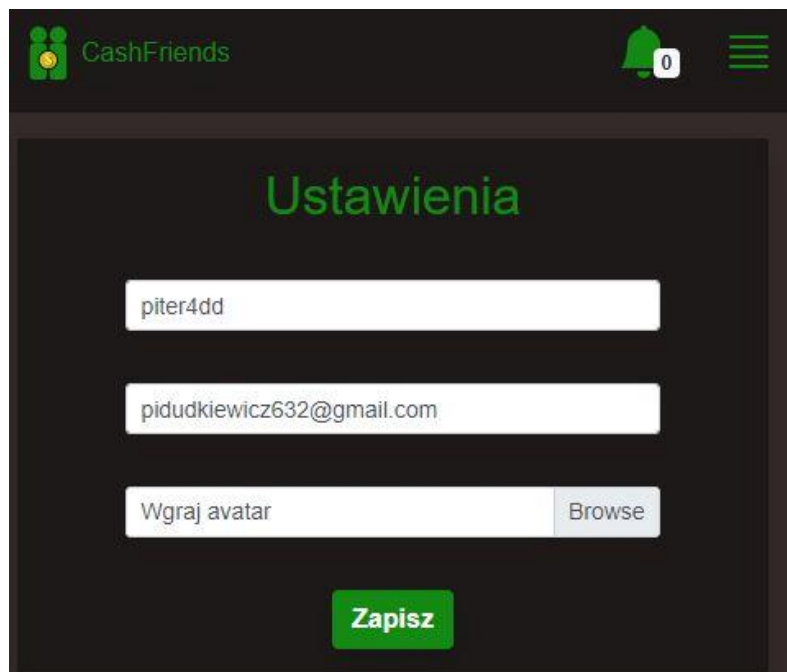
6.4.21 Widok ustawień użytkownika



Rysunek 60 Widok ustawień użytkownika

Widok ten prezentuje avatar użytkownika, nazwę, adres e-mail oraz trzy przyciski: edycji informacji, edycji hasła oraz usunięcia konta.

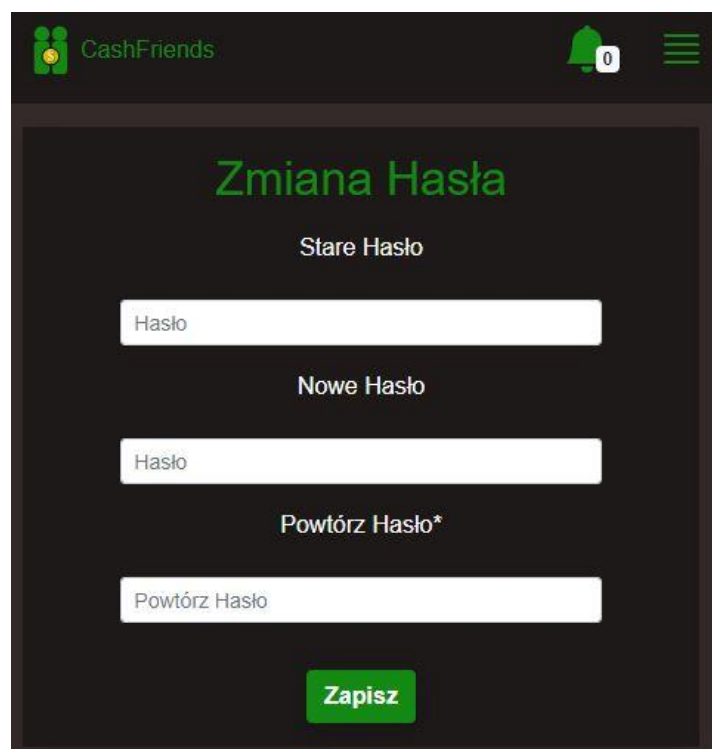
6.4.22 Widok edycji informacji o użytkowniku



Rysunek 61 Widok edycji informacji o użytkowniku

Widok edycji informacji o użytkowniku umożliwia zmianę nazwy, adresu e-mail lub avatara, zasady walidacji są identyczne jak podczas rejestracji. Po nieprawidłowym wypełnieniu pola wyświetla się na czerwono pod polem odpowiednia informacja. Po prawidłowym wypełnieniu formularza i naciśnięciu przycisku „Zapisz”, zmiana jest zapisywana oraz wyświetla się odpowiedni komunikat. Jeżeli adres e-mail został zmieniony, to na starym adresie e-mail pojawi się wiadomość z linkiem do potwierdzenia zmiany adresu e-mail.

6.4.23 Widok zmiany hasła

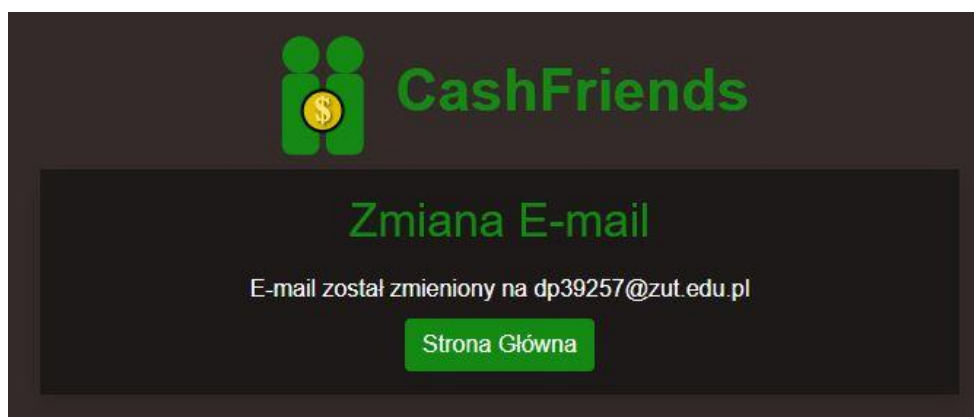


The screenshot shows the 'Zmiana Hasła' (Change Password) interface. At the top, there is a header with the 'CashFriends' logo on the left, a notification bell icon with a '0' badge on the right, and a hamburger menu icon. The main content area has a dark background. The title 'Zmiana Hasła' is displayed in green. Below it, the label 'Stare Hasło' (Old Password) is followed by a white input field containing the placeholder text 'Hasło'. The label 'Nowe Hasło' (New Password) is followed by another white input field with the placeholder 'Hasło'. Below that, the label 'Powtórz Hasło*' (Repeat Password*) is followed by a third white input field with the placeholder 'Powtórz Hasło'. At the bottom of the form is a green button with the white text 'Zapisz' (Save).

Rysunek 62 Widok zmiany hasła

Aby zmienić hasło należy podać stare hasło oraz nowe hasło i powtórzyć je. Zasada walidacji nowego hasła jest identyczna jak przy rejestracji. Jeżeli pole zostanie wypełnione niepoprawnie to pod pole pojawi się na czerwono odpowiedni komunikat. Po zatwierdzeniu zmiany wyświetla się okno z komunikatem.

6.4.24 Widok potwierdzenia zmiany e-mail , aktywacji konta lub przywrócenia konta

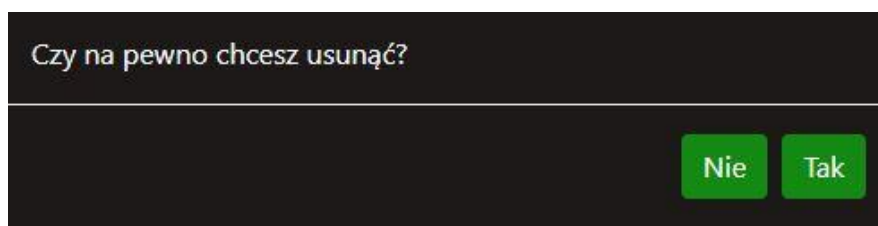


The screenshot shows the 'Zmiana E-mail' (Change Email) confirmation screen. At the top, there is a header with the 'CashFriends' logo on the left and the text 'CashFriends' in green on the right. The main content area has a dark background. The title 'Zmiana E-mail' is displayed in green. Below it, the text 'E-mail został zmieniony na dp39257@zut.edu.pl' is shown in white. At the bottom is a green button with the white text 'Strona Główna' (Main Page).

Rysunek 63 Widok potwierdzenia zmiany e-mail

Widoki aktywacji oraz przywrócenia konta wyglądają analogicznie.

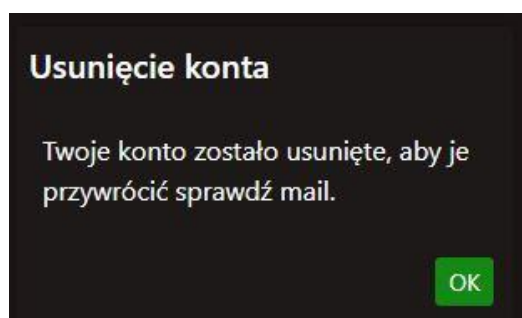
6.4.25 Usuwanie zrzutki, grupy lub użytkownika



Rysunek 64 Okno potwierdzenia

Okno potwierdzenia usunięcia wyświetla się, gdy naciśniemy przycisk „Usuń”. Jeżeli naciśniemy przycisk „Tak” to operacja się wykona. Gdy usuwamy konto na adres e-mail wysyłany jest link z możliwością usunięcia konta.

6.4.26 Okno z komunikatem



Rysunek 65 Okno z komunikatem

Okno z komunikatem wyświetlanej po pomyślnym wykonaniu operacji lub w sytuacji wystąpienia błędu.

6.5 Mechanizm odnawiania zrzutek

Zrzutka cykliczna jest odnawiana wtedy, gdy data ostatniego cyklu pomniejszona o dwa dni jest mniejsza od dzisiejszej daty. Jeżeli różnica pomiędzy dzisiejszą datą a datą ostatniego cyklu jest większa od zadanej długości cyklu, to data kolejnej zrzutki będzie równa sumie dzisiejszej daty z długością cyklu, w przeciwnym razie data kolejnej zrzutki będzie równa sumie ostatniej daty cyklu z długością cyklu. W przypadku zrzutki pojedynczej należność jest taka sama jak w poprzedniej. W grupowej natomiast do nowego cyklu przypisywani są tylko użytkownicy z aktywnymi kontami i ci, którzy aktualnie znajdują się w grupie. Jeżeli w ostatniej zrzutce wybrane było „rozbicie kwoty”, to liczona jest suma ostatnich należności i rozbijana jest ona na członków nowego cyklu, a jeżeli było wybrane „ręczne” bądź „określona składka” to należności są takie same jak w poprzednim.

Zrzutki dodatkowo nie są odnawiane, kiedy w zrzutce pojedynczej jeden z

użytkowników usunął konto lub jeżeli w zrzutce grupowej wszyscy uczestnicy poprzedniej zrzutki usunęli konto bądź nie należą już do grupy.

7. Testy

Testy we współczesnych projektach informatycznych odgrywają ważną rolę i są niezbędnym etapem poprzedzającym wdrażenie aplikacji. Ważne jest to, aby w testach uwzględnić możliwie wszystkie przypadki i wykryć te błędy, które są dużym zagrożeniem dla bezpieczeństwa i stabilności działania aplikacji. Dlatego w obecnych czasach w wielu firmach wyznaczony jest specjalnie wykwalifikowany zespół, którego zadaniem jest opracowanie i przeprowadzenie testów aplikacji.

Istnieje wiele rodzajów testów, a różnią się one typem oraz poziomem. Ze względu na poziom testowania rozróżniamy:

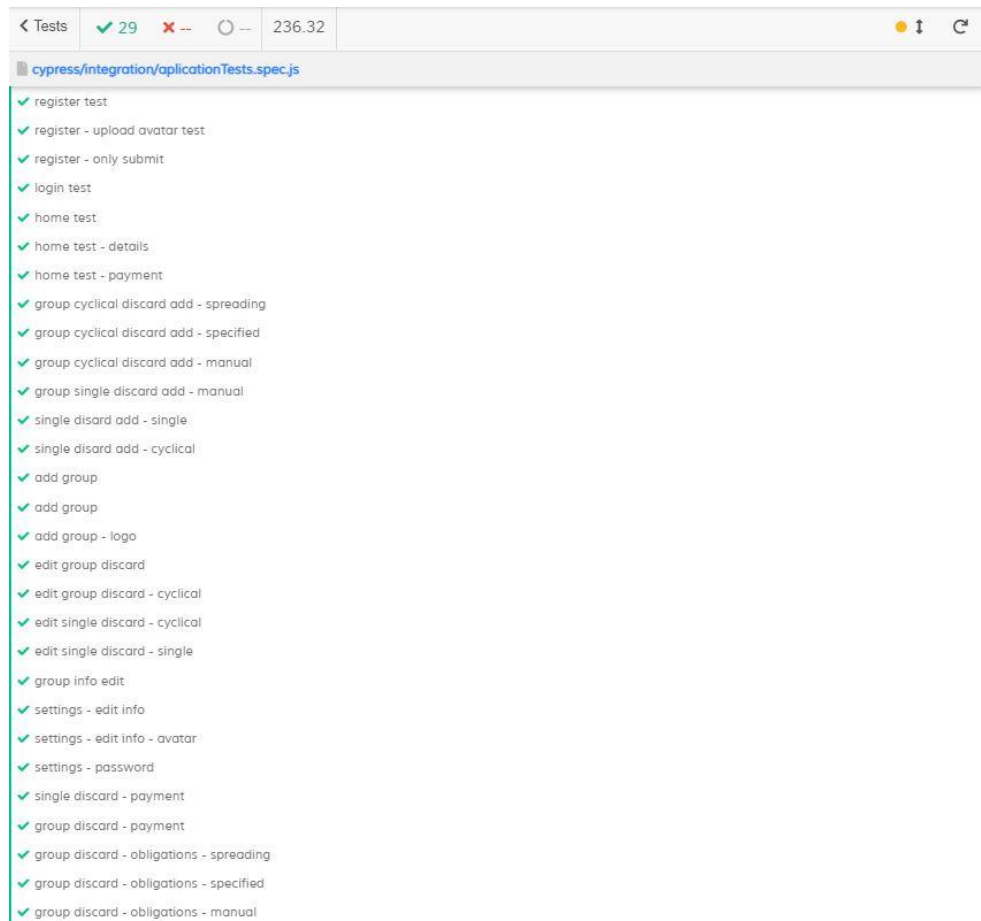
- Testy jednostkowe – odbywają się na kodzie źródłowym, testowane są funkcje, metody, klasy oraz moduły, które działają w aplikacji. Są to bardzo proste i zautomatyzowane testy.
- Testy integracyjne – testowana jest komunikacja między poszczególnymi elementami systemu.
- Testy end to end – są to testy, które symulują zachowanie użytkownika podczas korzystania z aplikacji.
- Testy akceptacyjne – mają na celu sprawdzenie czy spełnione są wymagania biznesowe, opierają się głównie na powieleniu zachowań użytkowników, z reguły testują wydajność i stabilność pracy.

Natomiast ze względu na typ rozróżniamy:

- Testy funkcjonalne – sprawdzają czy spełnione zostały wymagania funkcjonalne.
- Testy wydajnościowe – sprawdzają jak działa system pod wpływem obciążenia.
- Testy dymne – sprawdzają główną funkcjonalność oraz działanie oprogramowania.
- Testy eksploracyjne -sprawdzają czy w nowych wersjach aplikacji występują błędy z poprzednich wersji.

W niniejszej pracy zostały użyte testy dymne oraz end to end, które były przeprowadzone automatycznie i manualnie. Automatyczne testy ze względu na trudność przeprowadzenia ich skupiały się tylko na głównych funkcjonalnościach aplikacji, czyli

logowaniu, rejestracji, dodawaniu i edycji zrzutek, grup, należności. Zostały one wykonane przy użyciu narzędzia Cypress, które umożliwia wykonywanie testów end to end. Manualne testy skupiały się na wszystkich funkcjonalnościach aplikacji, sprawdzane było zarówno działanie klienta jak i serwera, a przeprowadzane były one przez cały okres tworzenia prototypu. Uwzględnione zostały możliwie wszystkie przypadki, dzięki czemu błędy zostały wykryte, a później naprawione.



Rysunek 66 Wynik testów automatycznych w narzędziu Cypress

2	get	#login		
3	- type	test123		
	(xhr)	● GET 200 /sockjs-node/info?t=1619550636014		
4	get	#password		
5	- type	1Qwerty		
6	get	button		
7	- click			
	(xhr)	● POST 400 /api/user/Login		
	(req)	● http://77.55.214.54:8000/api/user/Login Complete		apiCheck
8	wait	@apiCheck 1		
9	- assert	expected 400 to not equal 200		
10	get	#login		
11	- clear			
12	get	#login		
13	- type	test12		
14	get	button		
15	- click			
	(xhr)	● POST 400 /api/user/Login		
	(req)	● http://77.55.214.54:8000/api/user/Login Complete		apiCheck
16	wait	@apiCheck 2		
17	- assert	expected 400 to not equal 200		
18	get	#login		
19	- clear			
20	get	#password		
21	- clear			
22	get	#login		
23	- type	test123		
24	get	#password		
25	- type	1Qwerty.		
26	get	button		
27	- click			
	(xhr)	● POST 200 /api/user/Login		
	(req)	● http://77.55.214.54:8000/api/user/Login Complete		apiCheck
28	wait	@apiCheck 3		
	(page Load)	--page Loaded--		
	(new url)	http://192.168.0.10:8000/		
	(xhr)	● GET 200 /api/notification/number		
	(xhr)	● GET 200 /api/discard/user?isPaid=false		
	(xhr)	● GET 200 /sockjs-node/info?t=1619550639132		
29	- assert	expected 200 to equal 200		

Rysunek 67 Test logowania

8. Podsumowanie

Celem niniejszej pracy inżynierskiej było stworzenie i umieszczeniu w chmurze aplikacji do podziału wspólnych wydatków wśród znajomych, która była skierowana głównie na urządzenia mobilne.

Często bywa tak, że ludzie posiadają wspólne wydatki np. współlokatorzy muszą płacić razem za mieszkanie, prąd czy Internet, dlatego motywacją do stworzenia tej aplikacji było ułatwienie zarządzania i księgowania oraz możliwość monitorowania nadchodzących tego typu wydatków. Stworzony prototyp wyróżnia się na tle dostępnych na rynku aplikacji, dostępnością na urządzeniach mobilnych i desktopowych, możliwością tworzenia cyklicznych zrzutek, prostym interfejsem oraz szybkością działania.

Do stworzenia aplikacji zostały wykorzystane między innymi takie technologie jak VueJS oraz ExpressJS. Pierwsza z nich została użyta do stworzenia klienta, a wybrana została ze względu na prostotę i szybkość działania oraz możliwość tworzenia aplikacji jednostronicowych (z ang. SPA), czyli niewymagających przeładowania strony w celu zmiany widoku, dzięki czemu aplikacja przypomina bardziej aplikację natywną. ExpressJS wykorzystany był przy tworzeniu serwera, został wybrany ze względu na szybkość pracy, posiadanie przejrzystego wzorca projektowego oraz możliwością użycia tego samego języka programowania, co przy tworzeniu klienta, czyli JavaScript.

Podsumowując, niniejsza praca zakończyła się sukcesem, a jej efektami są aplikacja z dalszymi możliwościami rozwojowymi oraz poszerzenie wiedzy autora w dziedzinie przetwarzania chmurowego oraz tworzenia aplikacji.

8.1 Możliwe przyszłe rozwiązania

W obecnych czasach płatności elektroniczne są bardzo popularne, według Izby Gospodarki Elektronicznej (raport „Płatności elektroniczne 2018”) około 62% Polaków posiada internetowe konta bankowe, a liczba płatności elektronicznych z roku na rok rośnie, tak więc warto byłoby w planach rozwojowych aplikacji umieścić implementację płatności elektronicznych wewnątrz aplikacji. Takie rozwiązanie znacząco ułatwiłoby księgowanie wpłat dokonywanych przez uczestników zrzutki, ponieważ użytkownik zarządzający wpłatami nie musiałby monitorować i zatwierdzać wpłat, robiłaby to za niego aplikacja. Najlepszym rozwiązaniem byłoby skorzystanie z usług zewnętrznej firmy, która dostarczałaby możliwość transakcji internetowych, jedną z takich firm jest Shoper

(<https://www.shoper.pl/>), która umożliwia płatności GPay, Blik, MasterCard i Visa. Dzięki temu część odpowiedzialności przenosimy na firmę zewnętrzną.

Warto również rozważyć wprowadzenie statystyk użytkownika i grupy, dzięki czemu użytkownicy mogliby w łatwy sposób sprawdzać swoje zaległości, ile mają należności do uregulowania w danym okresie oraz aktywność członków grupy w zrzutka, dzięki czemu w łatwiejszy sposób mogliby kontrolować swoje wydatki. Kolejnymi możliwościami rozwoju są importowanie zrzutek do kalendarza oraz generowanie raportów do arkusza kalkulacyjnego, co jeszcze bardziej ułatwiłoby użytkownikowi zarządzanie wydatkami oraz pamiętanie o nadchodzących zrzutkach.

9. Bibliografia

1. <https://it-develop.pl/blog/aplikacja-na-smartfona-czy-rwd.html> (dostęp dnia 10.01.2021)
2. <https://www.splitwise.com/> (dostęp dnia 10.01.2021)
3. <https://www.tricount.com/> (dostęp dnia 10.01.2021)
4. <https://settleup.io/> (dostęp dnia 10.01.2021)
5. <https://splid.app/> (dostęp dnia 11.01.2021)
6. <https://play.google.com/> (dostęp dnia 11.01.2021)
7. <https://www.ibm.com/pl-pl/cloud/learn/cloud-computing-gbl> (dostęp dnia 17.04.2021)
8. <https://itcraftapps.com/pl/blog/aplikacje-cross-platformowe-vs-natywne-vs-hybrydowe-vs-pwa/> (dostęp dnia 18.04.2021)
9. <https://www.mongodb.com/what-is-mongodb> (dostęp dnia 20.04.2021)
10. https://developer.mozilla.org/pl/docs/Learn/Server-side/Express_Nodejs/Introduction (dostęp dnia 20.04.2021)
11. <https://vuejs.org/> (dostęp dnia 21.04.2021)
12. <https://mansfeld.pl/webdesign/bootstrap-co-to-jest-czy-warto-uzywac/> (dostęp dnia 21.04.2021)
13. <https://blog.helion.pl/mvvm-wprowadzenie-wzorca/> (dostęp dnia 24.04.2021)
14. <https://it-academy.pl/poziomy-typy-testow/> (dostęp dnia 24.04.2021)
15. <https://bestarion.com/iaas-vs-paas-vs-saas-enter-the-ecommerce-vernacular> (dostęp dnia 10.01.2021)
16. https://www.researchgate.net/publication/278302676_A_Comparative_Study_MongoDB_vs_MySQL (dostęp 21.04.2021)
17. <https://dotnet.microsoft.com/apps/xamarin> (dostęp 21.04.2021)
18. <https://docs.microsoft.com/pl-pl/xamarin/xamarin-forms/enterprise-application-patterns/mvvm> (dostęp 21.04.2021)