

## Programowanie zespołowe

# Donation Application

<b>Nazwa</b>	Aplikacja darowizn
<b>Kierownik</b>	Jakub Regeńczuk
<b>Początkowy skład grupy projektowej</b>	Marcin Głęba Katarzyna Lebioda Michał Swat
<b>Prowadzący</b>	dr inż. Marcin Zawada

# Wstęp

Celem niniejszego projektu było stworzenie aplikacji dla darczyńców oraz osób zbierających na różne cele niekomercyjne oraz realizację swoich projektów.

Systemu miał być ułatwieniem, platformą umożliwiającą obustronny kontakt użytkownikom zbierającym oraz chcącym przekazać datki na wybraną zbiórkę. Ponadto innowacyjną funkcją tworzonego systemu są płatności w najpopularniejszej obecnie kryptowalucie Bitcoin.

## Zakres projektu

Podstawową funkcjonalnością systemu miała być łatwość użytkowania zarówno w środowisku mobilnym jak i desktopowym. Pierwotnie planowane funkcjonalności obejmowały stworzenie:

- Aplikacji mobilnej umożliwiającej korzystanie z wszystkich funkcji systemu oraz posiadającej zintegrowany system powiadomień i płatności
- Serwisu internetowego z funkcjonalnościami równoważnymi aplikacji
- Strony funkcjonalnej aplikacji składającej się z:
  - Serwera umieszczonego w jednej z chmur umożliwiającej bezpłatny i zdalny dostęp
  - Bazy danych bezpiecznie przechowującej dane użytkowników oraz hasła

## Podział pracy

Podział pracy oraz obowiązków został stworzony w pierwszych tygodniach semestru i uwzględnia preferencje oraz propozycje członków zespołu. Początkowo poszczególne odpowiedzialności zostały przydzielone następująco:

Za stronę aplikację mobilną odpowiedzialna jest Katarzyna Lebioda oraz Marcin Głaba. Analizą oraz późniejszą implementacją i obsługą systemu płatności i notyfikacji zajmuje się Marcin Głaba.

Część serwerową i bazodanową tworzy Michał Swat.

Jakub Regeńczuk odpowiedzialny jest za implementację strony klienckiej aplikacji oraz serwis internetowy. Zajmuje się również koordynacją pracy członków oraz tworzeniem dokumentacji.

### **Uwaga:**

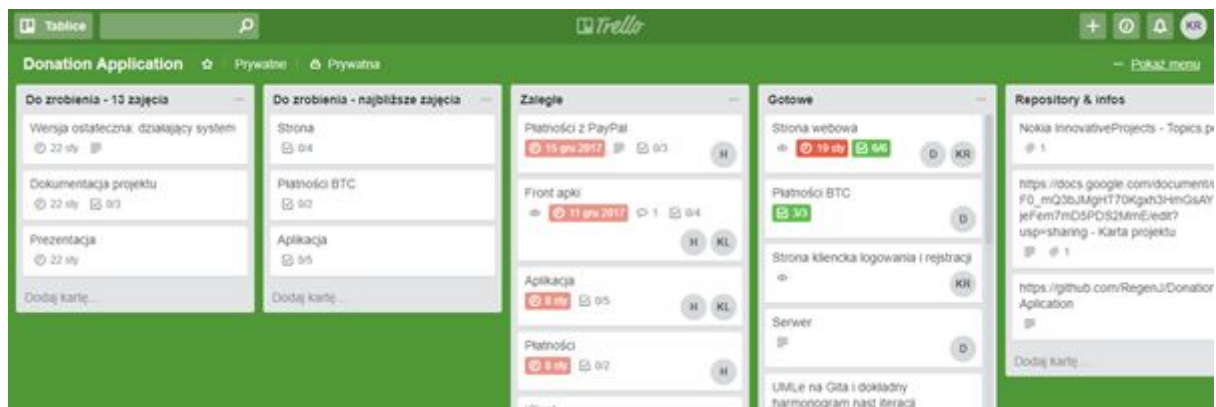
Od 8.01 z powodu braku pracy i kontaktu z niektórymi członkami zespołu funkcjonalności oraz zakres projektu zostały zmienione w następujący sposób:

Część serwerową i bazodanową nadal tworzy Michał Swat. Dodatkowo współtworzy również stronę webową oraz samodzielnie zajmuje się systemem płatności przez Bitcoin.

Jakub Regeńczuk zajmuje się wyłącznie tworzeniem strony internetowej czyli dwóch z jej widoków oraz ogólnie stroną stylistyczną aplikacji przeglądarkowej. Jak wcześniej również zajmuje się dokumentacją oraz koordynacją pracy.

## Organizacja pracy

Grupa projektowa korzystała z Trello - prostego, szybkiego oraz skutecznego narzędzia do zarządzania zadaniami.



Rys. 1 - tablice Trello

Projekt zbudowany jest w oparciu o tablice, na których to znajdują się listy zadań. Listy te pozwalają tworzyć karty (zadania), które w bardzo prosty sposób możemy sortować, czy przenosić pomiędzy różnymi listami. Listy te podzielone są według statusu poszczególnych zadań: „Do zrobienia”, „Zaległe”, „Gotowe” oraz „Informacje”.

# Harmonogram i jego realizacja

1. Stworzenie pełnej specyfikacji systemu i podział pracy zespołu.
  - a. Rozeznanie w dostępnych technologiach i wybranie najodpowiedniejszych dla danych części systemu
  - b. Założenie repozytorium oraz ustalenie kanałów i zasad komunikacji
  - c. Postawienie serwera

Termin: do 4 zajęć (6.11)

**Realizacja:** w terminie wypełniono punkty a., b. i c.

2. Projekt systemu, w tym przynajmniej diagramu klas/obiektów i diagramu przypadków użycia
  - a. Analiza systemów płatności oraz wybranie najodpowiedniejszego dla aplikacji
  - b. Rozplanowanie architektury, wyglądu i funkcji aplikacji - spotkanie zespołu
  - c. Stworzenie diagramów UML
    - i. Pierwsza wersja
    - ii. Poprawki
  - d. Rozpoczęcie implementacji poszczególnych części systemu
    - i. Konfiguracja środowiska dla aplikacji mobilnej, stworzenie pierwszych widoków w aplikacji
    - ii. Serwer: endpointy dla rejestracji użytkownika (sprawdzanie czy już taki istnieje)
    - iii. Serwer: umożliwienie korzystania z podstawowej bazy pozostałym członkom zespołu: chmura lub skrypt stawiający i konfigurujący
  - e. Dodanie stworzonych funkcjonalności na repozytorium
    - i. Stworzenie własnych branchy przez pozostałych członków od takiego wstępnego mastera

Termin: do 7 zajęć (27.11)

**Zrealizowano:** system płatności (przelewy24) wstępnie udało się wybrać na spotkaniu zespołu (a. i b.), wtedy stworzono również pierwsze wersje diagramów. Konfiguracja serwera przebiegła również w czasie.

**Nie zrealizowano:** analizy systemu płatności i łatwości jej zaimplementowania

3. Pierwsza iteracja systemu (wersja alpha): działający prototyp z implementacją podstawowych funkcjonalności systemu
  - a. Serwer i baza danych:
    - i. Rejestracja i logowanie
    - ii. Tworzenie zbiórki przez użytkownika i przechowywanie zbiorów użytkownika
    - iii. Zmiana hasła i wylogowywanie
  - b. Klient:
    - i. Dokończenie metod logowania rejestracji

- ii. Komunikacja z serwerem: obsługa requestów
- c. Aplikacja mobilna
  - i. Widoki logowania/rejestracji
  - ii. Tworzenia zbiorów: widok z formularzem (nazwa, opis, data, suwak z kwotą)
  - iii. Widok zbiórki
- d. Aplikacja webowa
  - i. Strona rejestracji i logowania
  - ii. Tworzenie zbiórki
- e. Płatności

Termin: 10 zajęcia (18.12)

**Zrealizowano:** już od 20.11 możliwe było zalogowanie się na podstawową stronę aplikacji oraz tworzenie zbiorów. Również w podobnym czasie powstała wizualna strona aplikacji mobilnej, potem również widoki logowania, rejestracji i tworzenia zbiórki. Później dodano również w miarę sprawną komunikację z serwerem oraz działającą autoryzację oraz rejestrację użytkownika z aplikacji.

**Nie zrealizowano:** pełnej obsługi requestów z serwera, połączenia widoków w aplikacji, możliwości rejestracji i przeglądania jako zalogowany aplikacji. System płatności z Przelewy24, z powodu trudności w implementacji biblioteki twórców, został zmieniony na PayPal. Nie został on jednak podłączony do aplikacji. Nie został również zamieszczony na repozytorium żaden związany z płatnościami kod.

4. Druga iteracja (wersja beta): system z większością działających funkcjonalności
  - a. Aplikacja mobilna:
    - i. Dodanie głównego widoku aplikacji, zbiorów subskrybowanych przez użytkownika
    - ii. Obsługa błędów oraz poprawne wysyłanie requestów oraz odbieranie odpowiedzi podczas tworzenia zbiórki
    - iii. Ostatecznie podłączenie i zintegrowanie systemu płatności
  - b. Aplikacja webowa:
    - i. Wyszukiwanie zbiorów
    - ii. Szczegółowy widok zbiórki
    - iii. Widok mojej fundacji
    - iv. Płatności BTC
  - c. Serwer:
    - i. Obsługa metod do aplikacji webowej
  - d. Płatności
    - i. Dodanie widoku płatności przez Paypal do strony oraz aplikacji
    - ii. Płatności przez BTC?

Termin: 12 zajęcia (15.01)

**Zrealizowano:** Wyszukiwanie oraz szczegółowy widok zbiórki w aplikacji webowej. Michał zaimplementował również działający system płatności przez Bitcoin, z możliwością przesyłania wybranej kwoty w BTC na numer portfela.

**Nie zrealizowano:** od 16.12 żadnego postępu ze stroną wizualną aplikacji, brak możliwości przełączania między widokami. Nie został podłączony widok logowania, tworzenia zbiórki oraz system płatności. Próby zrobienia tego ze strony Marcina (były takie) nie zostały w żaden sposób dodane do repozytorium. Brak kontaktu z nim oraz z Kasią.

5. Ostateczna wersja systemu
  - a. Poprawki widoków, widok główny
  - b. Finalizacja płatności przez Bitcoin
  - c. Dokumentacja

Termin: 13 zajęcia (22.01)

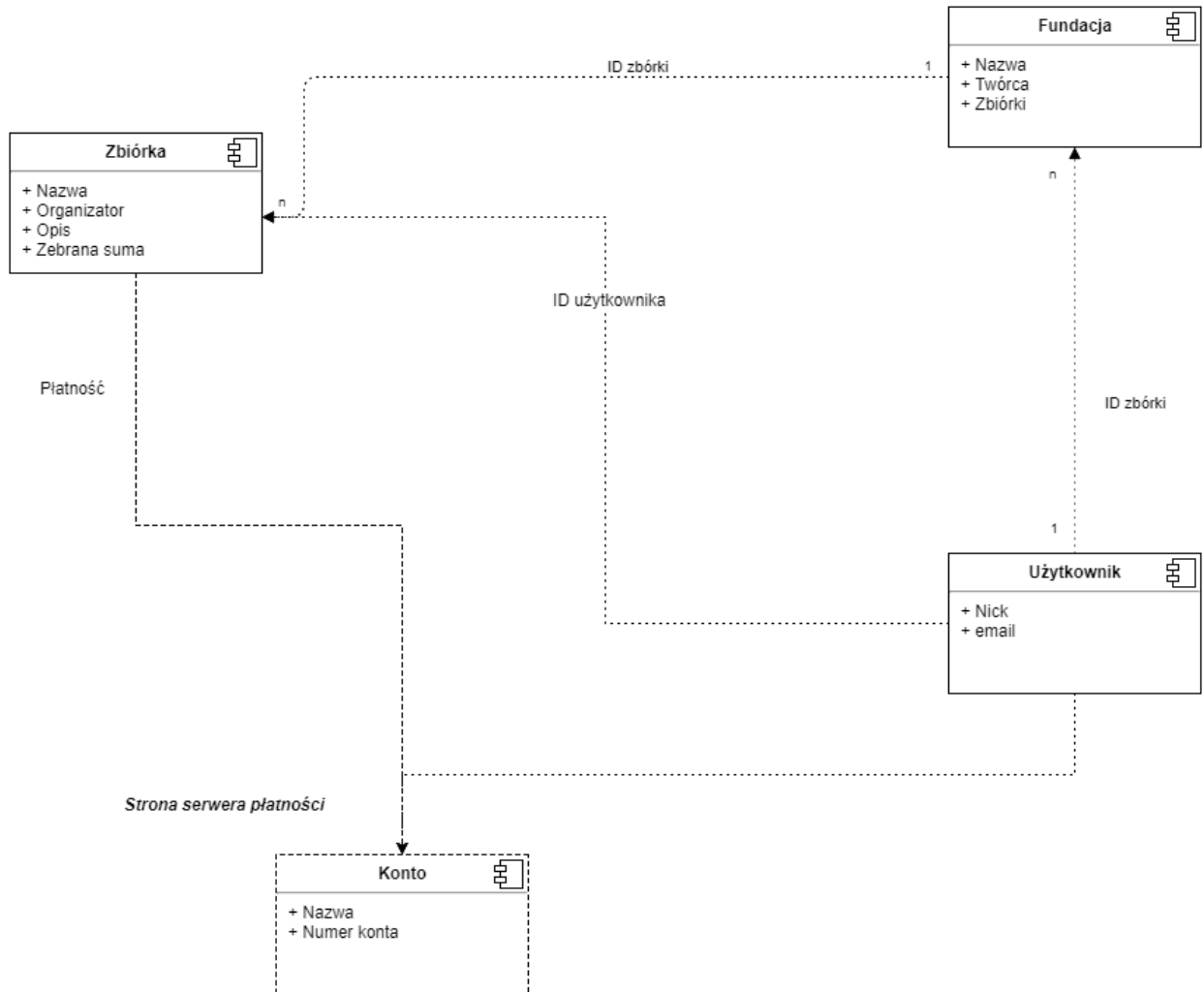
**Zrealizowano:** Stylistyczne poprawki widoków, widok główny z wyświetlaniem subskrybowanych zbiorów. Dokończona i poprawiona płatność z podstawową obsługą błędów.

Niniejsza dokumentacja i prezentacja.

**Nie zrealizowano:** funkcjonalności widoku fundacji, jakkolwiek działającej aplikacji mobilnej.

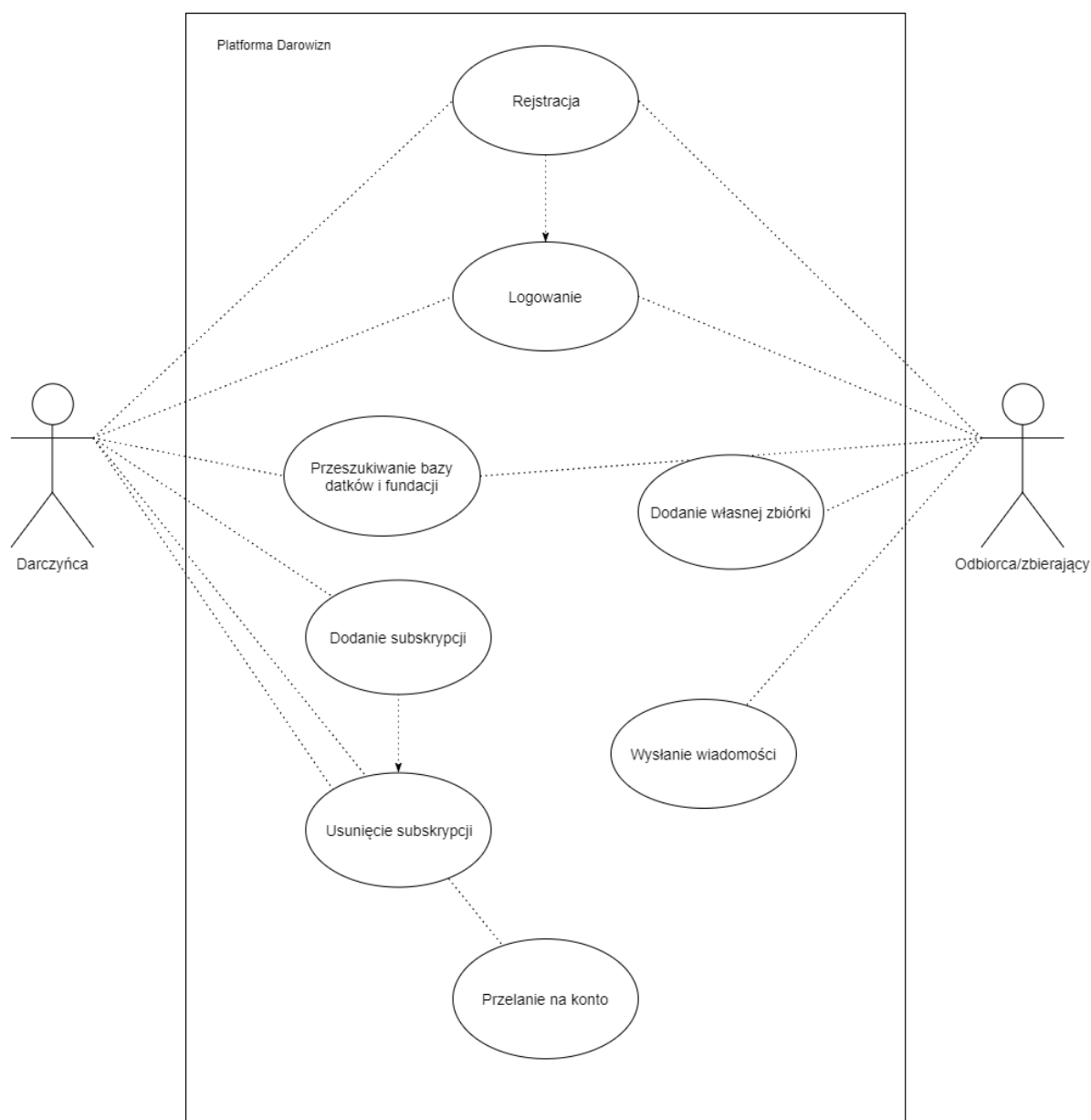
# Projekt systemu

Projekt systemu opisywany jest przez poniższe diagramy UML



Rys. 2 - diagram komponentów

Diagram komponentów opisuje główne składowe systemu: Zbiórkę, Fundację, Użytkownika i Konto. Każdy z nich posiada pola które łączą relacje 1 do 1 oraz 1 do wielu. Relacje wraz z kluczami oznaczone są przerywanymi liniami.



Rys. 3 - diagram przypadków użycia

Diagram przypadków użycia standardowo pokazuje przykładowych użytkowników aplikacji oraz czynności które mogą wykonywać korzystając z aplikacji.

Czynności wspólne oraz związki przyczynowe oznaczone są na diagramie pionowymi strzałkami, których zwrot oznacza kolejność wykonywania poszczególnych czynności.



# Implementacja systemu

Ostateczna wersja systemu składa się ze aplikacji webowej opartej na frameworku Django oraz serwisu internetowego opartego na standardowym zestawie HTML5, CSS3 oraz JavaScript. Poszczególne części systemu oraz użyte w nich technologie zostały opisane poniżej.

## Serwer oraz baza danych

Podstawą strony serwera jest framework Django, który ułatwia komunikację z bazą danych poprzez automatyczną serializację obiektów Pythona. Ponadto Django zapewnia wbudowany system autentykacji użytkowników oraz zapewnia podstawowe wymagania bezpieczeństwa (bezpieczne przechowywanie hasła - hash + sól, obrona przed XSS i CSRF).

Baza danych oparta jest na PostgreSQL, która pozwala na efektywne przechowywanie i zarządzanie dużymi zbiorami danych.

Aplikacja Django działa w oparciu o rozwiązanie chmurowe Heroku, które umożliwia elastyczny sposób deploymentu oraz zarządzania aplikacją. Ponadto Heroku dba o skalowalność uruchomionych w chmurze aplikacji.

## Płatności

System płatności został zrealizowany w oparciu o pythonowy moduł Bit, który pozwala na łatwe realizowanie płatności za pomocą kryptowaluty Bitcoin. Dla każdego użytkownika podczas jego rejestracji tworzony jest nowy portfel Bitcoin. Jego adres przypisywany jest do każdej zbiórki stworzonej przez użytkownika.

## Aplikacja webowa

Tworzony serwis internetowy spełnia wszystkie podstawowe założenia projektu: użytkownik po zalogowaniu ma wgląd do spisu zbiorów, możliwość ich wyszukiwania i sortowania, tworzenia nowej oraz przekazania donacji.

Panel użytkownika jest rozplanowany zgodnie z założeniami i metodyką znaną z frameworka Bootstrap. Do nawigacji po podstronach służy górny pasek nawigacji, wraz z polami przekierowującymi na odpowiednie strony.



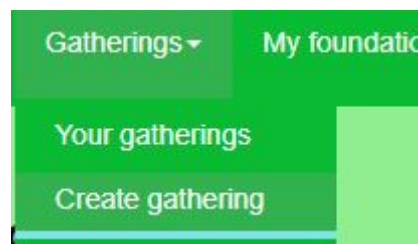
Rys. 4 - pasek nawigacji

## ***Home***

Głównym widokiem w aplikacji jest strona Home. To na niej znajdują się podstawowe informacje o projekcie Helping Hand, przydatne dla użytkowników informacje oraz formularze kontaktowe.

## ***Gatherings***

Drugim w kolejności polem jest Gatherings. Klikając w nie użytkownik przejdzie do rozwijanego menu, w którym może nową zbiórkę oraz przejrzeć te dotychczas stworzone.



Rys. 5 - Gatherings

Stworzenie nowej zbiórki możliwe jest po wybraniu suwakiem kwoty (w Bitcoinach), którą chcemy przekazać zbierającemu. Dokładność kwot to tysięczne części BTC.

Następnie trzeba wybrać datę zakończenia oraz tytuł i opis zbiórki.



Rys. 6 - tworzenie nowej zbiórki

## ***My foundation***

Widok i funkcjonalności nie zostały w tej iteracji w pełni zaimplementowane. Powstał jedynie szablon podstrony wraz z wysyłaniem zapytań do bazy.

## Donate Others

Kolejną zakładką w pasku nawigacji jak i również widokiem w aplikacji jest Donate Others. Tu użytkownik znajdzie wszystkie dostępne zbiórki oraz znajdzie te które go najbardziej zainteresują. Do wyszukiwania służy pole tekstowe połączone z odpowiednią funkcją zwracającą żądane przez użytkownika zbiórki. Wyszukiwanie jest na razie możliwe tylko po tytule zbiórki.



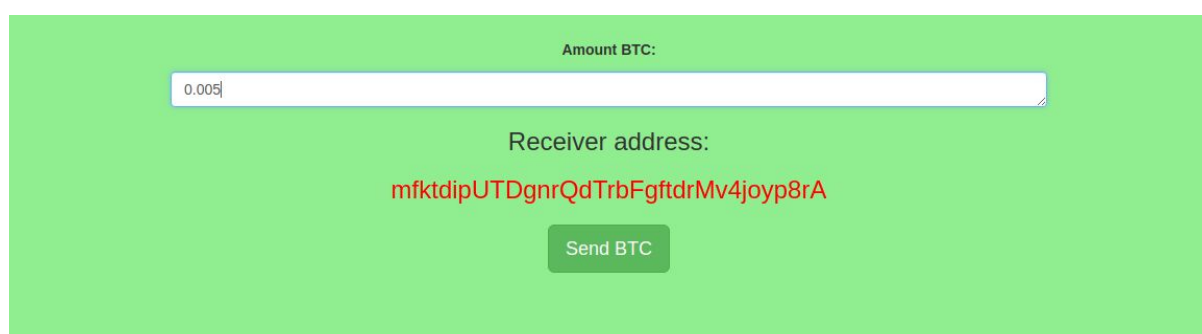
Start date	End date	Owner	Title	Target	Actual	Progress	Donation
01. 14. 2018	2018-01-30	kamcia	qwert	3980.0 PLN	0.0 PLN	<input type="text"/>	<button>Send BTC</button>
03. 04. 2018	2018-01-14	testUser	Na poprawki	1810.0 PLN	0.0 PLN	<input type="text"/>	<button>Send BTC</button>
01/03/2018	2018-01-25	uzyszkodnik	na pleco	4010.0 PLN	3900.0 PLN	<div>97%</div>	<button>Send BTC</button>

Rys. 7 - widok Donate Others

W teoretycznym dalszym rozwoju aplikacji możliwe jest dodanie kolejnych pól do wyszukiwania np. po założycielu zbiórki czy jej dacie zakończenia.

## Płatności Bitcoin

Kluczowa funkcja systemu to oczywiście płatność na daną zbiórkę. Przekierowanie do obsługującej jej podstrony następuje po naciśnięciu „Send BTC” przy interesującej nas zbiórce.



Amount BTC:

Receiver address:

mfktdipUTDgnrQdTrbFgftdrMv4joyp8rA

Send BTC

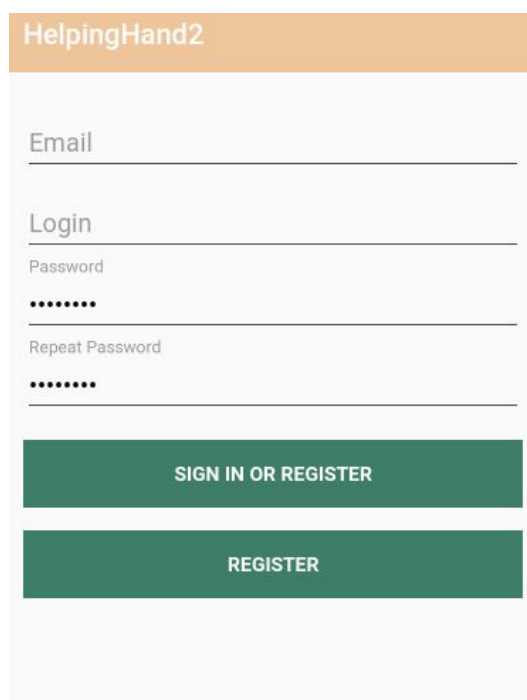
Rys. 8 – płatność

Aby dokonać płatności użytkownik musi wypełnić kwotę płatności, adres jego portfela zostaje pobrany z bazy. Po wciśnięciu „Send BTC” użytkownik zostaje autoryzowany i przekierowany na stronę płatności. Szczegóły i mechanizm dokonywania płatności opisane zostały w sekcji *Płatności*

# Aplikacja mobilna - GUI

## *Logowanie i rejestrowanie*

Utworzenie layout'u logowania, którego założeniem było, aby po naciśnięciu przycisku *Sign or register*, w zależności od tego jakie dane zostały podane sprawdzać w bazie danych usera i następnie logować, albo odkrywać schowane komórki, aby następnie zarejestrować użytkownika po naciśnięciu przycisku *Register*.

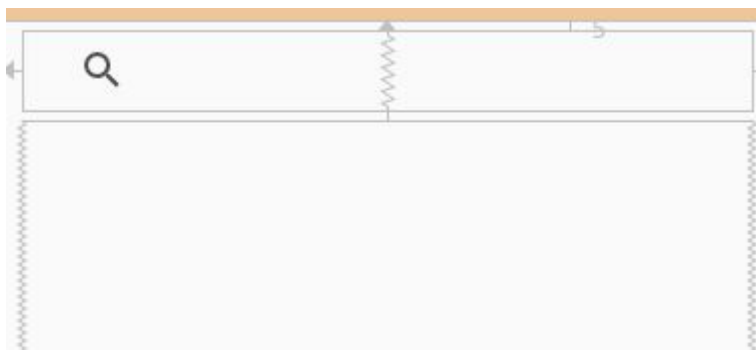


The mockup shows a mobile app interface for login and registration. At the top is an orange header bar with the text 'HelpingHand2'. Below the header is a light gray background. The form consists of several input fields: 'Email', 'Login', 'Password' (with a masked password '\*\*\*\*\*'), and 'Repeat Password' (with a masked password '\*\*\*\*\*'). Below these fields are two green buttons: 'SIGN IN OR REGISTER' and 'REGISTER'.

Rys.9 - logowanie GUI

## *Karta główna*

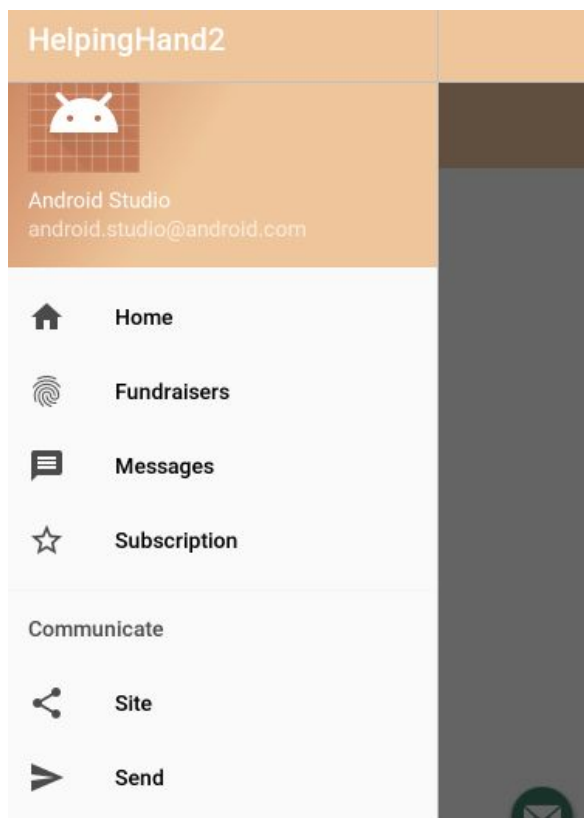
Layout główny, którego funkcją w założeniu była możliwość przeszukiwania bazy danych w poszukiwaniu utworzonych zbiorów oraz wyświetlaniu wyników w liście poniżej pola wyszukań.



Rys. 10 - karta główna GUI

### ***Nawigacja - menu boczne***

Widok menu bocznego wysuwanego z prawej strony bądź poprzez naciśnięcie przycisku na górnej listwie widoku. Służyć miała do nawigacji, przemieszczania się między kartami aplikacji. Header menu zawierać miał zapamiętane po zalogowaniu dane użytkownika. A kolejne pola menu odpowiadają za powrót do strony domowej, tworzenie własnej zbiórki, wiadomości, które mógł tworzyć jedynie posiadacz własnej zbiórki, oraz pola listy zbiorok obserwowanych, przekierowanie do strony.



Rys. 11 - menu boczne GUI

### ***Karta fundacji***

Ten widok zawiera pola informacyjne na temat wybranej przez użytkownika zbiórki, które w założeniu miały wypełniać się danymi pobranymi z bazy danych. A po naciśnięciu przycisku miała przekierowywać do płatności.

Donation Name

Bank Account Number

Kwota

DESCRIPTION

➤

Rys. 12 - karta zbiórki GUI

### ***Karta tworzenia zbiórki***

Ten widok miał służyć jako ankieta do generowania własnej zbiórki, która po właściwym uzupełnieniu pól ankiety i zaakceptowaniu przez użytkownika, miała dodać nową zbiórkę do bazy danych.

Create Your Own Donation

Enter the collection name

Name

Enter the bank account

Account Number

Set the amount of your collection

Amount

Choose a collection end date

Date

Describe the purpose of your donation

Description

CREATE DONATION

Rys. 13 - karta tworzenia zbiórki GUI

# Podsumowanie

Projekt zakończył się połowicznym sukcesem. Udało się stworzyć funkcjonujący i ogólnodostępny system w serwisie chmurowym, również wszystkie podstawowe założenia (ogólnodostępność, intuicyjność, dostępne płatności) zostały skutecznie zaimplementowane.

Do niezrealizowanych części należy z pewnością aplikacja mobilna. Na jej porażkę składało się kilka czynników. Pierwszym jest nieodpowiednia organizacja zasobów i pracy: brak zaangażowania odpowiedzialnych za nią członków powinien zostać wcześniej zgłoszony i w razie braku odzewu ich odpowiedzialności powinny zostać usunięte z projektu. Kolejnym czynnikiem jest ten wspomniany brak zaangażowania członków zespołu, mogący być spowodowany brakiem spotkań zespołu.

Podsumowując, mimo trudności komunikacyjnych i znacznego zmniejszenia grupy udało się zaimplementować założone funkcjonalności a nawet poszerzyć je o innowacyjny system płatności poprzez Bitcoin. Jednakże, z powodów wyżej wymienionych nie udało się zaimplementować działającej aplikacji mobilnej, a jedynie GUI.