

# Internetowy Serwis Aukcyjny

*Jakub Niewczas, Damian Klimek, Tomasz Zabrzewski, Łukasz Myszkowski*

Zespołowe przedsięwzięcie inżynierskie

Informatyka

Rok. akad. 2017/2018, sem. I

Prowadzący: dr hab. Marcin Mazur

## Spis treści

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Opis projektu</b>                             | <b>3</b> |
| 1.1      | Członkowie zespołu . . . . .                     | 3        |
| 1.2      | Cel projektu (produkt) . . . . .                 | 3        |
| 1.3      | Potencjalny odbiorca produktu (klient) . . . . . | 3        |
| 1.4      | Metodyka . . . . .                               | 3        |
| <b>2</b> | <b>Wymagania użytkownika</b>                     | <b>3</b> |
| 2.1      | User story 1 . . . . .                           | 3        |
| 2.2      | User story 2 . . . . .                           | 3        |
| 2.3      | User story 3 . . . . .                           | 4        |
| 2.4      | User story 4 . . . . .                           | 4        |
| 2.5      | User story 5 . . . . .                           | 4        |
| 2.6      | User story 6 (Opcjonalnie) . . . . .             | 4        |
| 2.7      | User story 7 . . . . .                           | 4        |
| 2.8      | User story 8 . . . . .                           | 4        |
| 2.9      | User story 9 . . . . .                           | 5        |
| 2.10     | User story 10 . . . . .                          | 5        |
| 2.11     | User story 11 . . . . .                          | 5        |
| 2.12     | User story 12 . . . . .                          | 5        |
| 2.13     | User story 13 . . . . .                          | 5        |
| 2.14     | User story 14 . . . . .                          | 5        |
| 2.15     | User story 15 . . . . .                          | 5        |
| 2.16     | User story 16 . . . . .                          | 6        |
| 2.17     | User story 17 . . . . .                          | 6        |
| 2.18     | User story 18 . . . . .                          | 6        |
| <b>3</b> | <b>Harmonogram</b>                               | <b>6</b> |
| 3.1      | Rejestr zadań (Product Backlog) . . . . .        | 6        |
| 3.2      | Sprint 1 . . . . .                               | 6        |
| 3.3      | Sprint 2 . . . . .                               | 6        |
| 3.4      | Sprint 3 . . . . .                               | 7        |
| 3.5      | Sprint 4 . . . . .                               | 7        |
| 3.6      | Sprint 5 . . . . .                               | 7        |
| <b>4</b> | <b>Product Backlog</b>                           | <b>7</b> |
| 4.1      | Backlog Item 1 . . . . .                         | 7        |
| 4.2      | Backlog Item 2 . . . . .                         | 9        |
| 4.3      | Backlog Item 3 . . . . .                         | 9        |
| 4.4      | Backlog Item 4 . . . . .                         | 10       |
| 4.5      | Backlog Item 5 . . . . .                         | 10       |
| 4.6      | Backlog Item 6 . . . . .                         | 11       |
| 4.7      | Backlog Item 7 . . . . .                         | 11       |
| 4.8      | Backlog Item 8 . . . . .                         | 11       |
| 4.9      | Backlog Item 9 . . . . .                         | 12       |
| 4.10     | Backlog Item 10 . . . . .                        | 12       |
| 4.11     | Backlog Item 11 . . . . .                        | 12       |
| 4.12     | Backlog Item 12 . . . . .                        | 13       |
| 4.13     | Backlog Item 13 . . . . .                        | 13       |

|          |                                   |           |
|----------|-----------------------------------|-----------|
| 4.14     | Backlog Item 14 . . . . .         | 13        |
| 4.15     | Backlog Item 15 . . . . .         | 14        |
| 4.16     | Backlog Item 16 . . . . .         | 14        |
| 4.17     | Backlog Item 17 . . . . .         | 14        |
| 4.18     | Backlog Item 18 . . . . .         | 15        |
| 4.19     | Backlog Item 19 . . . . .         | 15        |
| 4.20     | Backlog Item 20 . . . . .         | 16        |
| 4.21     | Backlog Item 21 . . . . .         | 16        |
| 4.22     | Backlog Item 22 . . . . .         | 16        |
| 4.23     | Backlog Item 23 . . . . .         | 17        |
| <b>5</b> | <b>Sprint 1</b>                   | <b>17</b> |
| 5.1      | Cel . . . . .                     | 17        |
| 5.2      | Sprint Planning/Backlog . . . . . | 17        |
| 5.3      | Realizacja . . . . .              | 17        |
| 5.4      | Sprint Review/Demo . . . . .      | 20        |
| <b>6</b> | <b>Sprint 2</b>                   | <b>20</b> |
| 6.1      | Cel . . . . .                     | 20        |
| 6.2      | Sprint Planning/Backlog . . . . . | 20        |
| 6.3      | Realizacja . . . . .              | 21        |
| 6.4      | Sprint Review/Demo . . . . .      | 22        |
| <b>7</b> | <b>Sprint 3</b>                   | <b>22</b> |
| 7.1      | Cel . . . . .                     | 22        |
| 7.2      | Sprint Planning/Backlog . . . . . | 22        |
| 7.3      | Realizacja . . . . .              | 23        |
| 7.4      | Sprint Review/Demo . . . . .      | 23        |
| <b>8</b> | <b>Sprint 4</b>                   | <b>24</b> |
| 8.1      | Cel . . . . .                     | 24        |
| 8.2      | Sprint Planning/Backlog . . . . . | 24        |
| 8.3      | Realizacja . . . . .              | 24        |
| 8.4      | Sprint Review/Demo . . . . .      | 25        |

# 1 Opis projektu

## 1.1 Członkowie zespołu

1. Damian Klimek (kierownik projektu);
2. Jakub Niewczas;
3. Tomasz Zabrzewski;
4. Łukasz Myszkowski;

## 1.2 Cel projektu (produkt)

Głównym i najważniejszym celem projektu jest stworzenie platformy webowej, która pojęcie „zakupy” pchnie krok dalej. Potencjalni klienci będą mieć możliwość robienia zakupów (*tj. sprzedawanie i kupowanie przedmiotów*) bez wychodzenia z domu. Kolejną ważną rzeczą jest zaimplementowanie modułu indywidualnych kont dla użytkowników (*logowanie się na swoje konto oraz rejestrowanie nowego*). Również stawiamy nacisk na stworzenie miłego dla oka, prostego a przede wszystkim intuicyjnego designu strony internetowej aby użytkownicy mogli w łatwy sposób przemieszczać się po platformie i szybko wyszukiwać interesujące ich przedmioty.

## 1.3 Potencjalny odbiorca produktu (klient)

Konkretnym klientem może być każdy, kto potrzebuje pieniędzy przez co dostaje możliwość sprzedania czegośkolwiek lub osoby potrzebujące jakiegoś dobra. Widelki wieku klientów nie są definiowalne - każdy kto posiada dostęp do internetu może skorzystać z usług jakie oferuje produkt.

## 1.4 Metodyka

Projekt będzie realizowany przy użyciu (zaadaptowanej do istniejących warunków) metodyki *Scrum*.

# 2 Wymagania użytkownika

## 2.1 User story 1

Jako nowy użytkownik serwisu chciałbym mieć możliwość założenia swojego indywidualnego konta, żebym mógł korzystać w pełni z usług dostarczanych przez serwis.

## 2.2 User story 2

Jako użytkownik chciałbym mieć możliwość zalogować się na swoje indywidualne konto, by móc korzystać z wszystkich usług platformy.

### 2.3 User story 3

Jako użytkownik chciałbym mieć możliwość wystawienia przedmiotu na aukcję, żebym mógł sprzedawać przedmioty oraz zarabiać pieniądze.

### 2.4 User story 4

Jako użytkownik podczas wystawiania przedmiotu na aukcję chciałbym mieć możliwość wybrania formy aukcji (*tj. licytacja, opcja „KUP TERAZ” lub obie formy*), żebym mógł sprzedać przedmiot w wybranej formie aukcji.

### 2.5 User story 5

Jako użytkownik podczas wystawiania przedmiotu na aukcję chciałbym mieć możliwość wprowadzenia opisu aukcji:

- konkretna nazwa aukcji;
- dodatkowe informacje o sprzedawanym przedmiocie;
- cena (*wywoławcza podczas licytacji lub stała za przedmiot podczas opcji „KUP TERAZ”*);
- wgranie zdjęcia przedmiotu;

żebym mógł zachęcić potencjalnego kupca do zakupu przedmiotu.

### 2.6 User story 6 (Opcjonalnie)

Jako użytkownik chciałbym mieć możliwość komentowania produktów innych osób wystawiających przedmioty na aukcji, , żebym mógł dowiedzieć się interesujących informacji od innych osób, które wcześniej kupiły przedmiot na jego temat bądź podzielić się własną opinią.

### 2.7 User story 7

Jako użytkownik chciałbym mieć możliwość dodania przedmiotów które mnie interesują do zakładek (*tj. Ulubione*), żebym mógł szybko i łatwo je odnaleźć.

### 2.8 User story 8

Jako użytkownik chciałbym mieć możliwość dodania swoich danych:

- imię;
- nazwisko;
- miejscowość;
- numer telefonu;
- e-mail;
- numer konta bankowego;

żeby umożliwić innym użytkownikom kontakt ze mną i rozliczeń.

## 2.9 User story 9

Jako użytkownik chciałbym mieć możliwość zmiany:

- istniejącego hasła na nowe;
- istniejącego e-maila na nowy;
- danych osobowe na nowe;

żeby zaktualizować email oraz wprowadzić nowe hasło bądź też poprawić błąd w danych osobowych lub całkowicie je zmienić.

## 2.10 User story 10

Jako użytkownik chciałbym mieć możliwość zobaczenia ile razy odwiedziono moją aukcję (*tj. licznik odsłon*), żebym mógł zobaczyć jakie jest zainteresowanie moją aukcją.

## 2.11 User story 11

Jako użytkownik chciałbym mieć możliwość edytowania danych aukcji (*tj. cena, opis*) oraz usunięcia danej aukcji, żebym mógł poprawić błędy lub zaktualizować dane aukcji.

## 2.12 User story 12

Jako użytkownik chciałbym mieć możliwość wyszukiwania interesujących mnie aukcji poprzez podanie słów kluczowych w wyszukiwarce, żebym mógł znaleźć przedmiot, który chciałbym zakupić.

## 2.13 User story 13

Jako użytkownik chciałbym by po dokonaniu transakcji (kup teraz bądź wygrana licytacja) wysyłała się do kupującego wiadomość z moim numerem konta oraz informacjami dotyczącymi transakcji, by mógł on dokonać wpłaty za zakupiony przedmiot.

## 2.14 User story 14

Jako użytkownik chciałbym mieć możliwość sprawdzenia starych aukcji które zostały zakończone (przedmiot sprzedano, przedmiot kupiono) bądź też wygasły, żebym mógł sprawdzić historie swoich transakcji.

## 2.15 User story 15

Jako użytkownik chciałbym mieć możliwość wysłania wiadomości prywatnej do innego użytkownika serwisu, żebym mógł się z nim skontaktować.

## 2.16 User story 16

Jako użytkownik chciałbym mieć możliwość przeglądania wszystkich dostępnych aukcji na serwisie w postaci listy wraz z ich podstawowymi danymi (*tj. nazwa aukcji, forma aukcji, cena, data rozpoczęcia oraz zakończenia aukcji*), żebym mógł przeglądać aukcje oferowane przez serwis.

## 2.17 User story 17

Jako użytkownik chciałbym mieć możliwość zakupu produktu natychmiast (opcja kup teraz), bądź licytowania go (opcja licytuj), żebym mógł go kupić.

## 2.18 User story 18

Jako użytkownik chciałbym mieć możliwość posiadania paska nawigacyjnego w którym zostanie umiejscowiona wyszukiwarka, oraz panel konta użytkownika.

# 3 Harmonogram

## 3.1 Rejestr zadań (Product Backlog)

- Data rozpoczęcia: 24.10.2017;
- Data zakończenia: 31.10.2017;

## 3.2 Sprint 1

- Data rozpoczęcia: *31.10.2017*;
- Data zakończenia: *14.11.2017*;
- Scrum Master: *Łukasz Myszkowski*;
- Product Owner: *Tomasz Zabrzewski*;
- Development Team: *Jakub Niewczas, Damian Klimek, Łukasz Myszkowski*;

## 3.3 Sprint 2

- Data rozpoczęcia: *14.11.2017*;
- Data zakończenia: *28.11.2017*;
- Scrum Master: *Tomasz Zabrzewski*;
- Product Owner: *Damian Klimek*;
- Development Team: *Łukasz Myszkowski, Jakub Niewczas, Damian Klimek*;

### 3.4 Sprint 3

- Data rozpoczęcia: *28.11.2017*;
- Data zakończenia: *12.12.2017*;
- Scrum Master: *Damian Klimek*;
- Product Owner: *Jakub Niewczas*;
- Development Team: *Tomasz Zabrzewski, Łukasz Myszkowski, Jakub Niewczas*;

### 3.5 Sprint 4

- Data rozpoczęcia: *12.12.2017*;
- Data zakończenia: *09.01.2018*;
- Scrum Master: *Jakub Niewczas*;
- Product Owner: *Damian Klimek*;
- Development Team: *Tomasz Zabrzewski, Łukasz Myszkowski, Damian Klimek*;

### 3.6 Sprint 5

- Data rozpoczęcia: *09.01.2018*;
- Data zakończenia: *23.01.2018*;
- Scrum Master: *Łukasz Myszkowski*;
- Product Owner: *Tomasz Zabrzewski*;
- Development Team: *Damian Klimek, Jakub Niewczas, Tomasz Zabrzewski*;

## 4 Product Backlog

### 4.1 Backlog Item 1

**Tytuł zadania:** Baza danych

**Opis zadania:** Przygotowanie struktury bazy danych.

**Priorytet:** Wysoki



**Definition of Done:** Baza danych powinna zawierać tyle tabel ile potrzeba na stworzenie aplikacji. Środowisko bazodanowe **MySQL**. Baza danych serwisu musi posiadać następujące struktury.

Struktura tabeli **users**:

- **id** → [typ - INT, AUTO\_INCREMENT];
- **username** → [typ - VARCHAR(32), kodowanie - UTF8\_GENERAL\_CI];
- **password** → [typ - VARCHAR(32), kodowanie - UTF8\_GENERAL\_CI];
- **firstname** → [typ - VARCHAR(64), kodowanie - UTF8\_GENERAL\_CI];
- **surname** → [typ - VARCHAR(64), kodowanie - UTF8\_GENERAL\_CI];
- **email** → [typ - VARCHAR(64), kodowanie - UTF8\_GENERAL\_CI];
- **phone** → [typ - VARCHAR(9), kodowanie - UTF8\_GENERAL\_CI];
- **place** → [typ - VARCHAR(128), kodowanie - UTF8\_GENERAL\_CI];
- **bank** → [typ - VARCHAR(26), kodowanie - UTF8\_GENERAL\_CI];
- **avatar** → [typ - VARCHAR(128), kodowanie - UTF8\_GENERAL\_CI];

Struktura tabeli **categories**:

- **id** → [typ - INT, AUTO\_INCREMENT];
- **name** → [typ - VARCHAR(64), kodowanie - UTF8\_GENERAL\_CI];

Struktura tabeli **auctions**:

- **id** → [typ - INT, AUTO\_INCREMENT];
- **name** → [typ - VARCHAR(128), kodowanie - UTF8\_GENERAL\_CI];
- **category** → [typ - INT, relacja z kolumną id z tabeli categories];
- **buy\_prize** → [typ - VARCHAR(16), kodowanie - UTF8\_GENERAL\_CI];
- **bidding\_prize** → [typ - VARCHAR(16), kodowanie - UTF8\_GENERAL\_CI];
- **description** → [typ - VARCHAR(512), kodowanie - UTF8\_GENERAL\_CI];
- **image** → [typ - VARCHAR(256), kodowanie - UTF8\_GENERAL\_CI];
- **last\_bidder** → [typ - INT, relacja z kolumną id z tabeli users];
- **buyer** → [typ - INT, relacja z kolumną id z tabeli users];

Struktura tabeli **comments**:

- **id** → [typ - INT, AUTO\_INCREMENT];

- **message** → [typ - VARCHAR(512), kodowanie - UTF8\_GENERAL\_CI];
- **owner** → [typ - INT, relacja z kolumną id z tabeli users];
- **auction** → [typ - INT, relacja z kolumną id z tabeli auctions];

Struktura tabeli **messages**:

- **id** → [typ - INT, AUTO\_INCREMENT];
- **owner** → [typ - INT, relacja z kolumną id z tabeli users];
- **recipient** → [typ - INT, relacja z kolumną id z tabeli users];
- **title** → [typ - VARCHAR(128), kodowanie - UTF8\_GENERAL\_CI];
- **message** → [typ - VARCHAR(512), kodowanie - UTF8\_GENERAL\_CI];
- **readed** → [typ - INT];

## 4.2 Backlog Item 2

**Tytuł zadania:** Rejestracja

**Opis zadania:** Możliwość założenia nowego konta;

**Priorytet:** Wysoki

**Definition of Done:** Każda osoba powinna mieć możliwość założenia swojego indywidualnego konta. Należy stworzyć formularz z podstawowymi danymi (*tj. login, hasło, imię, nazwisko, e-mail, miejscowość, numer telefonu, numer konta bankowego oraz avatar*). Wpisane przez użytkownika dane muszą być pobierane metodą *POST*. Finalnym efektem rejestracji jest wysłanie zapytania do bazy danych, który utworzy nowy rekord w tabeli *users* z danymi użytkownika.

## 4.3 Backlog Item 3

**Tytuł zadania:** Zabezpieczenie hasła

**Opis zadania:** Wymuszenie od użytkownika wprowadzenia danych spełniających określone kryteria;

**Priorytet:** Niski

**Definition of Done:** Użytkownik tworzący nowe konto musi podać hasło składające się z minimum 8 znaków w tym:

- wielkie litery od A do Z;
- małe litery od a do z;
- cyfry od 0 do 9;
- znaki niealfabetyczne (np. !, @, #, &);

Należy stworzyć skrypt używający *wyrażenia regularne*, które w łatwy sposób mogą sprawdzić poprawność hasła z wyżej ustalonymi zasadami. Hasło, które będzie podawał użytkownik rejestrując lub logując się na swoje konto powinno być niewidoczne dla osób postronnych (wykropkowane).

#### 4.4 Backlog Item 4

**Tytuł zadania:** Weryfikowanie danych

**Opis zadania:** Sprawdzanie wprowadzanych danych przez użytkownika podczas rejestracji;

**Priorytet:** Wysoki

**Definition of Done:** Każda osoba wprowadzająca dane do serwisu podczas rejestracji powinna podać poprawne dane:

- imię - powinno zawierać od 3 do 32 znaków, w tym tylko od a do z;
- nazwisko - powinno zawierać od 3 do 32 znaków, w tym tylko od a do z;
- e-mail - powinien przypominać maskę (tj. mójemail09@domena.pl);
- miejscowość - powinna zawierać od 3 do 64 znaków, w tym tylko od a do z;
- numer telefonu - powinna składać się z 9 cyfr.
- konto bankowe - powinno składać się z 26 cyfr.

Gdy użytkownik nie zastosuje się do wyżej wymienionych zasad, należy wysłać do niego w postaci *JavaScript Popup Alert box*, wiadomość zwrotną podającą, w którym miejscu popełnił błąd oraz jak poprawnie wypełnić input formularza.

#### 4.5 Backlog Item 5

**Tytuł zadania:** Logowanie

**Opis zadania:** Możliwość zalogowania się na swoje konto;

**Priorytet:** Wysoki

**Definition of Done:** Stworzenie formularza logowania (tj. login oraz hasło) pobierające owe dane, następnie w postaci zapytania MySQL musi sprawdzać czy istnieją dane w bazie danych oraz czy pasują do jednego użytkownika. Należy zaimplementować wiadomość zwrotną gdy użytkownik wpisał złe dane (nie ma takiego rekordu w bazie danych bądź hasło nie pasuje do loginu). Po udanej operacji (logowanie powiodło się) należy stworzyć użytkownikowi sesję oraz przekierować go na stronę główną serwisu.

## 4.6 Backlog Item 6

**Tytuł zadania:** Pasek nawigacji

**Opis zadania:** Utworzenie paska nawigacji umieszczonego w górnej części ekranu;

**Priorytet:** Wysoki

**Definition of Done:** Należy utworzyć pasek nawigacji, który będzie umiejscowiony na górze ekranu przeglądarki. Pozycjonowanie paska należy ustawić jako *fixed*, aby w przypadku długiej strony „przykleił” się do ekranu i podążał za scrolowaną stroną. Szerokość paska powinna być uzależniona od szerokości ekranu, na którym będzie odwiedzany serwis, zaś wysokość powinna być ustawiona na *75px*.

## 4.7 Backlog Item 7

**Tytuł zadania:** Zmiana danych

**Opis zadania:** Umożliwienie użytkownikowi zmiany swoich podstawowych danych;

**Priorytet:** Średni

**Definition of Done:** Każda osoba posiadająca swoje konto powinna mieć możliwość zmiany swoich podstawowych danych na nowe (tj. imię, nazwisko, e-mail, miejscowość, numer telefonu, numer konta bankowego, hasło oraz avatar). Należy stworzyć formularz z wcześniej wymienionymi danymi oraz jako domyślne *value* kolejnych inputów formularza powinny być przypisane dane pobrane z bazy danych. Obowiązkowo należy zaimplementować przycisk *submit*, po kliknięciu którego zostaną pobrane dane z formularza metodą *POST* oraz wysłane do bazy danych w postaci zapytania MySQL. Wysłane dane muszą podmieniać już istniejące.

## 4.8 Backlog Item 8

**Tytuł zadania:** Wyszukiwarka

**Opis zadania:** Umożliwienie użytkownikowi wyszukiwania interesujących przedmiotów;

**Priorytet:** Średni

**Definition of Done:** Każda osoba (zalogowana lub niezalogowana) może używać wyszukiwarki. Wyszukiwarke w postaci formularza należy umieścić na pasku nawigacji, który znajduje się na górze ekranu. Wpisane hasło w input formularza zwrócone przez funkcję *htmlspecialchars()* należy umieścić w zapytaniu MySQL w poszukiwaniu przedmiotów. Zapytanie należy zaprojektować tak, aby wyszukiwał każdy przedmiot, w którym zawiera się wpisane przez użytkownika hasło (*np. dom, domek, domeczek, domuś itp.*).

## 4.9 Backlog Item 9

**Tytuł zadania:** Panel konta

**Opis zadania:** Zgrupowanie w jedno miejsce podstawowych akcji użytkownika;

**Priorytet:** Średni

**Definition of Done:** Panel należy podzielić na dwie grupy - dla osób zalogowanych oraz dla osób niezalogowanych. Oba panele powinny być umiejscowione na pasku nawigacji na górze ekranu. Panel dla zalogowanego użytkownika powinien być w postaci avataru użytkownika, w który można kliknąć a następnie rozwija się menu z podstawowymi akcjami (*tj. edytuj dane, moje aukcje, historia zakupów, wyloguj*). Panel dla osoby niezalogowanej powinien być w postaci napisu „Moje konto”, które również rozwija się po kliknięciu w napis, z którego można wybrać dwie akcje (*tj. zarejestruj się, zaloguj się*). Po kliknięciu, w któryś z odnośników z rozwijanych menu każdy powinien być przekierowywany na konkretne podstrony serwisu.

## 4.10 Backlog Item 10

**Tytuł zadania:** Dodanie aukcji

**Opis zadania:** Umożliwienie użytkownikowi wystawienia nowej aukcji;

**Priorytet:** Wysoki

**Definition of Done:** Każda osoba zalogowana na swoje konto powinna mieć tę opcję dostępną. Osoba, która nie jest zalogowana na swoje konto powinna dostać wiadomość zwrotną, że ta opcja jest tylko dla osób zalogowanych lub całkowicie zablokować dostęp. Tworzenie nowego rekordu w bazie danych w tabeli *auction*.

## 4.11 Backlog Item 11

**Tytuł zadania:** Formularz aukcyjny

**Opis zadania:** Umożliwienie użytkownikowi podanie niezbędnych danych dotyczących aukcji;

**Priorytet:** Wysoki

**Definition of Done:** Osoba zalogowana podczas wystawiania nowego przedmiotu na aukcję musi mieć możliwość wpisania niezbędnych danych, które pozwolą odróżniać od siebie inne aukcje (tj. konkretna nazwa aukcji, opis przedmiotu, cena, zdjęcie). Formularz powinien pobierać dane metodą *POST*.

#### 4.12 Backlog Item 12

**Tytuł zadania:** Edytowanie aukcji

**Opis zadania:** Umożliwienie użytkownikowi edytowania danych swojej aukcji;

**Priorytet:** Średni

**Definition of Done:** Użytkownik, który wystawił przedmiot na aukcję powinien mieć możliwość edytowania danych związanych z aukcją. Należy stworzyć formularz, w którym domyślnymi wartościami kolejnych inputów będą dane wczytane z bazy danych. Dane z formularza powinny być pobierane metodą *POST*. Kliknięcie przycisku *submit* skutować powinno nadpisanie istniejących danych w bazie poprzez wykonanie zapytania *UPDATE*.

#### 4.13 Backlog Item 13

**Tytuł zadania:** Usuwanie aukcji

**Opis zadania:** Umożliwienie użytkownikowi usunięcia swojej aukcji;

**Priorytet:** Średni

**Definition of Done:** Użytkownik, który wystawił przedmiot na aukcję powinien mieć możliwość całkowitego usunięcia aukcji z serwisu. Usuwana aukcja powinna usunąć się również z bazy danych.

#### 4.14 Backlog Item 14

**Tytuł zadania:** Lista aukcji

**Opis zadania:** Umożliwienie użytkownikowi wyświetlania dostępnych aukcji;

**Priorytet:** Wysoki

**Definition of Done:** Każda osoba odwiedzająca serwis powinna mieć możliwość zobaczenia wszystkich dostępnych aukcji. Na stronie głównej powinny być wczytane wszystkie aukcje z bazy danych, które są aktywne (nie przeminął czas zakończenia aukcji). Aukcje powinny być wyświetlane w postaci tabelki (*tj. miniaturka zdjęcia, nazwa aukcji, rodzaj aukcji oraz ceny*). Nazwa aukcji powinna być odsyłaczem, który po kliknięciu przekieruje na podstronę przypisaną do aukcji.

#### 4.15 Backlog Item 15

**Tytuł zadania:** Wiadomość informacyjna

**Opis zadania:** Wyświetlenie wiadomości informacyjnej dla niezalogowanych użytkowników;

**Priorytet:** Niski

**Definition of Done:** Należy stworzyć box, który będzie umieszczony na stronie głównej (zaraz pod paskiem nawigacji). Box powinien być widoczny tylko dla osób niezalogowanych oraz tylko na stronie głównej - po zalogowaniu całkowicie znika. Należy umieścić w nim wiadomość informacyjną odnośnie serwisu (co to za witryna) oraz powinno zawrzeć się odnośniki do zarejestrowania nowego konta oraz zalogowania.

#### 4.16 Backlog Item 16

**Tytuł zadania:** Kupowanie przedmiotu

**Opis zadania:** Umożliwienie użytkownikowi zakupu;

**Priorytet:** Średni

**Definition of Done:** Osoba zalogowana powinna mieć możliwość zakupu interesującego przedmiotu poprzez opcję KUP TERAZ lub wygranie licytacji. Kiedy użytkownik zakupi przedmiot poprzez KUP TERAZ automatycznie dana aukcja powinna się zakończyć (wysyłane jest zapytanie do bazy danych edytujące rekord odpowiadający aukcji, do którego dopisywana jest wartość *id* osoby kupującej do kolumny *buyer*). Jeśli wartość jest 0, to znaczy, że nikt przedmiotu nie kupił. W przypadku wybrania formy aukcji jako licytacja osoba zalogowana może wpisać w formularzu cenę, którą jest w stanie zapłacić za przedmiot pod warunkiem, że jest ona większa od ostatniej licytacji. Formularz pobrany metodą *POST*. Zapytanie z nową zlicytowaną ceną nadpisuje się w bazie danych.

#### 4.17 Backlog Item 17

**Tytuł zadania:** Informacja zwrotna

**Opis zadania:** Przesyłanie danych osób biorących udział w transakcji;

**Priorytet:** Wysoki

**Definition of Done:** Po sfinalizowaniu każdej transakcji powinny automatycznie przesyłać się dane, które będą pobierane z bazy danych:

- od kupującego do sprzedającego - imię, nazwisko, adres, numer telefonu;
- od sprzedającego do kupującego - imię, nazwisko, numer konta bankowego, numer transakcji, numer telefonu;

#### 4.18 Backlog Item 18

**Tytuł zadania:** Historia zakupów

**Opis zadania:** Umożliwienie użytkownikowi przeglądania swoich starych transakcji;

**Priorytet:** Niski

**Definition of Done:** Każda osoba posiadająca swoje konto powinna mieć możliwość przeglądania swoich wszystkich wcześniejszych transakcji. Stworzyć podstronę przypisaną do każdego użytkownika serwisu, na którym będą wypisywane w postaci tabeli (*tj. miniaturka przedmiotu, nazwa, cena zakupu oraz jedna z kategorii [przedmiot: kupiony / sprzedany, aukcja wygasła]*). Inny użytkownik nie może podglądać historii zakupów innego użytkownika - ma dostęp tylko do swojej podstrony.

#### 4.19 Backlog Item 19

**Tytuł zadania:** Wiadomości

**Opis zadania:** Umożliwienie użytkownikowi wysyłania i odbierania prywatnych wiadomości od innych użytkowników;

**Priorytet:** Średni

**Definition of Done:** Każda osoba posiadająca konto powinna mieć możliwość wysyłania prywatnych wiadomości do innych użytkowników serwisu oraz również odbierania. Należy stworzyć formularz pozwalający wysyłanie wiadomości (*tj. temat wiadomości, do kogo ma być wysłana wiadomość oraz pole wiadomości*). Formularz powinien być pobierany metodą *POST*. Pobrane dane powinny być umieszczone w bazie danych w tabeli *messages*. Domyślnie argument *readed* powinien być ustawiany na 0. W przypadku gdy *readed* ma wartość 0 wyświetlany jest na stronie głównej komunikat w postaci boxa pod paskiem nawigacji, informujący o nowej nieprzeczytanej prywatnej wiadomości. Jeśli użytkownik wejdzie w wiadomość automatycznie wartość argumentu *readed* zmienia się na 1, zapisuje się w bazie - przez to nie będzie wyświetlane powiadomienie



o nieprzeczytanej wiadomości. Należy stworzyć podstronę, w której będą wyświetlać się wszystkie prywatne wiadomości użytkownika w postaci odsyłacza, który po kliknięciu będzie przekierowywał na kolejną podstronę przypisaną do konkretnej wiadomości. Na podstronie musi znajdować się podgląd wiadomości (*tj. od kogo została przysłana, temat wiadomości oraz cała wiadomość*). Dane wczytane bezpośrednio z bazy danych.

## 4.20 Backlog Item 20

**Tytuł zadania:** Komentarze

**Opis zadania:** Umożliwienie użytkownikowi komentowania konkretnych aukcji;

**Priorytet:** Niski

**Definition of Done:** Każda osoba zalogowana na swoje konto powinna mieć możliwość skorzystania z opcji komentowania aukcji. Należy stworzyć formularz tylko dla zalogowanych osób pod aukcją (*tj. wiadomość*). Dane pobierane poprzez metodę *POST*. Dane powinny być wysyłane do bazy danych wraz z *id* konta, z którego pisany jest komentarz. Pod formularzem należy stworzyć podgląd komentarzy do aukcji. W przypadku gdy aukcja nie posiada żadnych komentarzy należy wyświetlić wiadomość informującą o tym stanie. W przeciwnym wypadku należy wyświetlić w postaci tabeli komentarz (*tj. login użytkownika, który napisał komentarz, avatar oraz wiadomość komentarza*). Osoby niezalogowane pod aukcją widzą tylko komentarze, formularz jest dla nich całkowicie niewidoczny.

## 4.21 Backlog Item 21 (Opcjonalnie)

**Tytuł zadania:** Ulubione

**Opis zadania:** Umożliwienie użytkownikowi zapisania aukcji;

**Priorytet:** Niski

**Definition of Done:** Każda osoba zalogowana na swoje konto powinna mieć możliwość przypisania linku interesującej go aukcji do odpowiedniej zakładki w swoim profilu.

## 4.22 Backlog Item 22 (Opcjonalnie)

**Tytuł zadania:** Licznik

**Opis zadania:** Umożliwienie użytkownikowi sprawdzenia ilości odsłon aukcji;

**Priorytet:** Niski

**Definition of Done:** Na każdej stronie aukcji w dogodnym miejscu pod ogłoszeniem powinien znajdować się licznik odwiedzin danej aukcji. Licznik nalicza się gdy osoba wejdzie na podstronę aukcji. Dana powinna być zapisywana i wczytywana z bazy danych.

## 4.23 Backlog Item 23

**Tytuł zadania:** Kategorie

**Opis zadania:** Umożliwienie użytkownikowi wyszukiwania aukcji po kategoriach;

**Priorytet:** Niski

**Definition of Done:** Każda osoba odwiedzająca serwis powinna mieć dostępny pasek boczny z dostępnymi kategoriami. Kategorie powinny być wczytywane z bazy danych z tabeli *categories*, w postaci odsyłacza który umożliwi przekierowanie osoby na podstronę wybranej kategorii, gdzie zostaną wyświetlone wszystkie aukcje należące do danej kategorii.

# 5 Sprint 1

## 5.1 Cel

Celem pierwszego Sprintu jest umożliwienie użytkownikowi założenia konta dzięki któremu będzie on mógł uzyskać dostęp do wszystkich opcji serwisu i korzystania z niego. Aby to mogło się odbyć wcześniej zostanie stworzona baza danych w której będą zapisywane wszystkie dane.

## 5.2 Sprint Planning/Backlog

**Tytuł zadania.** Baza danych.

- Estymata: XL

**Tytuł zadania.** Rejestracja

- Estymata: S

**Tytuł zadania.** Logowanie

- Estymata: S

**Tytuł zadania.** Weryfikowanie danych

- Estymata: S

## 5.3 Realizacja

**Tytuł zadania.** Logowanie

**Wykonawca.** Damian Klimek

**Realizacja.** Został stworzony formularz logowania który wymaga od użytkownika podania danych (*tj. login oraz hasło*), następnie pobiera dane z formularza metodą *POST* i wartości przypisuje do zmiennych, które później sprawdzane są w bazie danych za pomocą zapytania MySQL. Jeśli nie istnieją takie dane to użytkownik otrzymuje informację zwrotną w postaci wyskakującego okienka, że konto nie istnieje, a w przeciwnym razie zostaje utworzona sesja oraz zostaje zalogowany.

**Tytuł zadania.** Baza danych;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Baza danych została stworzona w środowisku *MySQL*. Struktura bazy danych tworzona graficznie za pomocą *PHPMysqlAdmin*. Wszystkie dane będą umieszczane w bazie o nazwie *zpi\_project*, domyślne porównywanie napisów (kodowanie) zostało ustawione na *UTF8\_GENERAL\_CI*. Baza danych zawiera następujące tabele (*tj. auctions, categories, comments, messages, users*). Wykonując to zadanie nie napotkałem żadnych problemów lecz było dość czasochłonne ponieważ trzeba było przemyśleć całą strukturę działania serwisu.

Struktura tabeli *auctions*:

| # | Nazwa                | Typ          | Metoda porównywania napisów | Atrybuty | Null | Ustawienia domyślne | Dodatkowo      |
|---|----------------------|--------------|-----------------------------|----------|------|---------------------|----------------|
| 1 | <b>id</b>            | int(11)      |                             |          | Nie  | Brak                | AUTO_INCREMENT |
| 2 | <b>name</b>          | varchar(128) | utf8_general_ci             |          | Nie  | Brak                |                |
| 3 | <b>category</b>      | int(11)      |                             |          | Nie  | Brak                |                |
| 4 | <b>buy_price</b>     | varchar(16)  | utf8_general_ci             |          | Nie  | Brak                |                |
| 5 | <b>bidding_price</b> | varchar(16)  | utf8_general_ci             |          | Nie  | Brak                |                |
| 6 | <b>description</b>   | varchar(512) | utf8_general_ci             |          | Nie  | Brak                |                |
| 7 | <b>image</b>         | varchar(256) | utf8_general_ci             |          | Nie  | Brak                |                |
| 8 | <b>last_bidder</b>   | int(11)      |                             |          | Nie  | Brak                |                |
| 9 | <b>buyer</b>         | int(11)      |                             |          | Nie  | Brak                |                |

Rysunek 1: Tabela auctions

Struktura tabeli *categories*:

| # | Nazwa       | Typ         | Metoda porównywania napisów | Atrybuty | Null | Ustawienia domyślne | Dodatkowo      |
|---|-------------|-------------|-----------------------------|----------|------|---------------------|----------------|
| 1 | <b>id</b>   | int(11)     |                             |          | Nie  | Brak                | AUTO_INCREMENT |
| 2 | <b>name</b> | varchar(64) | utf8_general_ci             |          | Nie  | Brak                |                |

Rysunek 2: Tabela categories

Struktura tabeli *comments*:

Struktura tabeli *messages*:

| #                          | Nazwa          | Typ          | Metoda porównywania napisów | Atrybuty | Null | Ustawienia domyślne | Dodatkowo      |
|----------------------------|----------------|--------------|-----------------------------|----------|------|---------------------|----------------|
| <input type="checkbox"/> 1 | <b>id</b>      | int(11)      |                             |          | Nie  | Brak                | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <b>message</b> | varchar(512) | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 3 | <b>owner</b>   | int(11)      |                             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 4 | <b>auction</b> | int(11)      |                             |          | Nie  | Brak                |                |

Rysunek 3: Tabela comments

| #                          | Nazwa            | Typ          | Metoda porównywania napisów | Atrybuty | Null | Ustawienia domyślne | Dodatkowo      |
|----------------------------|------------------|--------------|-----------------------------|----------|------|---------------------|----------------|
| <input type="checkbox"/> 1 | <b>id</b>        | int(11)      |                             |          | Nie  | Brak                | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <b>owner</b>     | int(11)      |                             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 3 | <b>recipient</b> | int(11)      |                             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 4 | <b>title</b>     | varchar(128) | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 5 | <b>message</b>   | varchar(512) | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 6 | <b>readed</b>    | int(1)       |                             |          | Nie  | Brak                |                |

Rysunek 4: Tabela messages

Struktura tabeli *users*:

| #                           | Nazwa            | Typ          | Metoda porównywania napisów | Atrybuty | Null | Ustawienia domyślne | Dodatkowo      |
|-----------------------------|------------------|--------------|-----------------------------|----------|------|---------------------|----------------|
| <input type="checkbox"/> 1  | <b>id</b>        | int(11)      |                             |          | Nie  | Brak                | AUTO_INCREMENT |
| <input type="checkbox"/> 2  | <b>username</b>  | varchar(32)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 3  | <b>password</b>  | varchar(32)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 4  | <b>firstname</b> | varchar(64)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 5  | <b>surname</b>   | varchar(64)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 6  | <b>email</b>     | varchar(64)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 7  | <b>phone</b>     | varchar(9)   | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 8  | <b>place</b>     | varchar(128) | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 9  | <b>bank</b>      | varchar(26)  | utf8_general_ci             |          | Nie  | Brak                |                |
| <input type="checkbox"/> 10 | <b>avatar</b>    | varchar(128) | utf8_general_ci             |          | Nie  | Brak                |                |

Rysunek 5: Tabela users

**Tytuł zadania.** Rejestracja;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Tworząc skrypt rejestracji przypisałem pobrane dane z formularza metodą *POST* do zmiennych. W postaci zapytania *MySQL* dane zawarte w zmiennych będą dodawane do bazy danych do tabeli *users*. Zadanie nie wymagało zbyt wielkiego poświęcenia czasu gdyż składa się tylko z paru operacji.

**Tytuł zadania.** Weryfikacja danych;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Do wcześniejszego zadania **Rejestracja** dodałem warunki, które muszą być spełnione aby użytkownik mógł założyć swoje konto. Użyłem zmiennych, które zostały zadeklarowane w bazowym zadaniu. Na początku wykonałem dwa zapytania do bazy danych aby sprawdzić czy wpisany *login* oraz *hasło* istnieją już w bazie - jeśli tak to zwracana jest wiadomość, że konto na takie dane już istnieje. Lista warunków z backlogu została oskryptowana, w przypadku gdy skrypt ustali, że użytkownik popełnił błąd zostanie mu zwrócona informacja w postaci wyskakującego okienka *JavaScriptu*.

## 5.4 Sprint Review/Demo

Sprint został zakończony powodzeniem. Wszystkie założone zadania zostały zrealizowane w ustalonym czasie. Zostały zrealizowane następujące zadania:

- Została założona baza danych oparta na Środowisku bazodanowym MySQL, baza danych (która została uzupełniona w odpowiednie wiersze) umożliwia użytkownikowi założenie konta oraz logowanie się na stronie.
- Został utworzony formularz który ma na celu umożliwienie logowania się na swoje konto.
- Została stworzona weryfikacja danych wprowadzanych przy Rejestracji oraz Logowaniu: mająca na celu sprawdzić poprawność wpisanych danych przez użytkownika, w przypadku kiedy użytkownik nie wprowadzi poprawnych danych wyświetli się komunikat w postaci alertu podający w którym miejscu użytkownik popełnił jakiś błąd, oraz jaki to był błąd.
- Formularz sprawdza czy istnieje w bazie danych podany login oraz hasło i czy te dane są przypisane tylko do jednego użytkownika. Została zaimplementowana wiadomość zwrotna gdy użytkownik wpisze złe dane “nie ma takiego rekordu w bazie danych bądź hasło nie pasuje do loginu” jeśli dane są poprawne wyświetli się komunikat “logowanie powiodło się” i użytkownik zostanie przekierowany na główną stronę serwisu.

## 6 Sprint 2

### 6.1 Cel

Celem drugiego sprintu jest utworzenie “Panelu konta”, który ma ułatwić użytkownikowi zarządzanie danymi oraz w przyszłości aukcjami. Zostanie również stworzona wyszukiwarka która w przyszłości pozwoli wyszukać konkretne produkty.

### 6.2 Sprint Planning/Backlog

**Tytuł zadania.** Pasek nawigacji;

- Estymata: S;

**Tytuł zadania.** Zmiana danych;

- Estymata: S;

**Tytuł zadania.** Wyszukiwarka;

- Estymata: M;

**Tytuł zadania.** Panel konta;

- Estymata: M;

## 6.3 Realizacja

**Tytuł zadania.** Pasek nawigacji;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Pasek nawigacji stworzyłem u góry ekranu, zablokowałem go w tym miejscu na stałe używając wbudowanej opcji Bootstrap'a (framework, który używamy w projekcie) - `navbar-fixed-top`, przez co będzie podążał razem ze stroną podczas scrollowania. Wygląd całego paska stworzyłem w *CSS* - szerokość dopasowuje się do wielkości ekranu, na którym wyświetlana jest strona a sama zawartość paska jest umiejscowiana od środka.

**Tytuł zadania.** Wyszukiwarka;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Obecne zadanie jest kontynuacją poprzedniego zadania (*tj. pasek nawigacji*). Na pasku dodałem jeden większy input formularza z wiadomością podpowiadającą do czego służy (gdy użytkownik napisze jakieś hasło w inputcie wiadomość domyślna automatycznie się usuwa). Wpisane przez osobę hasło pobieram metodą *POST* oraz przepuszczam przez funkcję *htmlspecialchars()*, zwróconą wartość umieściłem w zapytaniu MySQL w postaci `([...]name LIKE %zmienna%[...])`. Ostatnim etapem jest przeniesienie osoby na podstronę, na której są wypisane aukcje, które zawierają ciąg wpisany w wyszukiwarkę.

**Tytuł zadania.** Panel konta;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Zadanie jest kontynuacją poprzedniego zadania (*tj. pasek nawigacji*). W przypadku gdy osoba nie jest zalogowana na swoje konto stworzyłem napis po prawej stronie wyszukiwarki *Moje konto*, w które jeśli klikniemy pojawi się kwadracik a w nim mini-menu (*tj. zarejestruj się oraz zaloguj się*) - po kliknięciu zostajemy przekierowani na odpowiednie podstrony. W przypadku gdy osoba jest zalogowana to obok wyszukiwarki wyświetla się awatar osoby w kształcie okręgu, który również można kliknąć i pojawi się mini-menu (*tj. edytuj dane, moje aukcje, historia zakupów, wyloguj*) - podobnie jak wyżej, po kliknięciu w menu skrypt przekierowuje nas na odpowiednią podstronę.

**Tytuł zadania.** Zmiana danych;

**Wykonawca.** Damian Klimek;

**Realizacja.** Obecne zadanie jest uzupełnieniem do zadania *Panel konta*. Polegało na stworzeniu podstrony do "Zmień dane" w zakładce Panelu konta. Został stworzony formularz z danymi (tj. imię, nazwisko, email, miejscowość, numer telefonu, numer konta bankowego, hasło oraz avatar) który pobiera dane z tabeli *users*, a następnie umożliwia zmianę ich poprzez wysłanie zapytania *UPDATE* do bazy danych. Został również dodany przycisk submit (Wyślij).

## 6.4 Sprint Review/Demo

Sprint został zakończony powodzeniem. Wszystkie założone zadania zostały zrealizowane w ustalonym czasie. Zostały zrealizowane następujące zadania:

- Został utworzony pasek nawigacyjny umożliwiający użytkownikowi nawigację po serwisie.
- Został utworzony panel konta w którym możemy zarządzać swoim kontem oraz zmienić dane.
- Została utworzona wyszukiwarka na pasku nawigacyjnym dzięki której możemy wyszukać interesującą nas aukcję.

## 7 Sprint 3

### 7.1 Cel

Celem trzeciego sprintu jest umożliwienie zalogowanemu użytkownikowi dodawania aukcji oraz jej edytowania. Zostanie również stworzona lista dostępnych aukcji oraz formularz aukcyjny.

### 7.2 Sprint Planning/Backlog

**Tytuł zadania.** Dodanie aukcji;

- Estymata: L;

**Tytuł zadania.** Lista aukcji;

- Estymata: M;

**Tytuł zadania.** Formularz aukcyjny;

- Estymata: S;

**Tytuł zadania.** Edytowanie aukcji;

- Estymata: M;

## 7.3 Realizacja

**Tytuł zadania.** Dodanie aukcji;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Z poziomu panelu konta użytkownika (kliknięcie w avatar na pasku nawigacji po czym wybranie pozycji *dodaj aukcję*) następuje przekierowanie na nową podstronę. Jeśli użytkownik nie jest zalogowany na swoje konto dostaje wiadomość, iż nie posiada dostępu na przebywanie na podstronie i zostaje wyświetlona mu strona główna. Stworzyłem zapytanie, które tworzy nowy rekord w bazie *auctions*.

**Tytuł zadania.** Formularz aukcyjny;

**Wykonawca.** Kuba Niewczas;

**Realizacja.** Obecne zadanie jest rozszerzeniem powyższego zadania (tj. *dodanie aukcji*). Stworzyłem formularz z następującymi danymi (*nazwa aukcji, kategoria, opis aukcji, zdjęcie przedmiotu, cena kup teraz, cena licytacji*). Dane pobieram metodą POST oraz przypisuję je do zmiennych a następnie ładuję one do zapytania.

**Tytuł zadania.** Lista aukcji;

**Wykonawca.** Damian Klimek;

**Realizacja.** Została stworzona lista aukcji, która jest pobierana z bazy danych (tabela *auctions*) oraz wyświetlana na stronie głównej dla wszystkich użytkowników, lista ta: wyświetla obrazek aukcji, nazwę aukcji oraz jej cenę. W przyszłości umożliwi przejście do danej aukcji.

**Tytuł zadania.** Edytowanie aukcji;

**Wykonawca.** Tomasz Zabrzewski;

**Realizacja.** Zadanie to polegało na stworzeniu podstrony z możliwością edycji aukcji tylko dla osoby która tą aukcję wystawiła. Został stworzony formularz z danymi (tj. nazwa, kategoria, cena kupna, cena licytacji, opis, zdjęcie) który pobiera dane z tabeli *auctions*, a następnie umożliwia zmianę ich poprzez wysłanie zapytania *UPDATE* do bazy danych.

## 7.4 Sprint Review/Demo

Sprint został zakończony powodzeniem. Wszystkie założone zadania zostały zrealizowane w ustalonym czasie.



## 8 Sprint 4

### 8.1 Cel

Celem czwartego sprintu jest m.in. umożliwienie właścicielowi danej aukcji usunięcia jej, dodanie możliwości grupowania aukcji wg. kategori. Będzie teraz można wejść w daną aukcję i zapoznać się z dokładną jej treścią, a następnie ją kupić. Zostanie również udostępniona opcja komentowania wybranych aukcji.

### 8.2 Sprint Planning/Backlog

**Tytuł zadania.** Usuwanie aukcji;

- Estymata: L;

**Tytuł zadania.** Kategorie;

- Estymata: M;

**Tytuł zadania.** Kupowanie przedmiotu;

- Estymata: S;

**Tytuł zadania.** Komentarze;

- Estymata: M;

**Tytuł zadania.** Podgląd aukcji;

- Estymata: M;

### 8.3 Realizacja

**Tytuł zadania.** Usuwanie aukcji;

**Wykonawca.** Damian Klimek

**Realizacja.** Zadanie to polegało na stworzeniu podstrony z możliwością usunięcia aukcji tylko dla osoby która tą aukcje wystawiła, a następnie umożliwiała zmianę ich poprzez wysłanie zapytania *DELETE* do bazy danych.

**Tytuł zadania.** Kategorie;

**Wykonawca.** Damian Klimek

**Realizacja.** Zostały stworzone kategorie, które pobierają się z bazy danych z tabeli *categories*. Kategorie te są wyświetlane na stronie głównej obok listy aukcji, po kliknięciu w wybraną kategorię następuje przekierowanie na podstronę z aukcjami przypisanymi do danej kategorii.

Każda osoba odwiedzająca serwis powinna mieć dostępny pasek boczny z dostępnymi kategoriami. Kategorie powinny być wczytywane z bazy danych z tabeli *categories*, w postaci odsyłacza który umożliwi przekierowanie osoby na podstronę wybranej kategorii, gdzie zostaną wyświetlone wszystkie aukcje należące do danej kategorii.

**Tytuł zadania.** Podgląd aukcji;

**Wykonawca.** Kuba Niewczas

**Realizacja.** Stworzyłem oddzielną podstronę, na której widnieją wszystkie informacje związane z daną aukcją. Każda osoba ma dostęp do tej podstrony. Dane wyświetlają się w postaci tabelki.

**Tytuł zadania.** Kupowanie przedmiotu;

**Wykonawca.**

**Realizacja.**

**Tytuł zadania.** Komentarze;

**Wykonawca.** Kuba Niewczas

**Realizacja.** Zadanie jest rozszerzeniem do zadania *Podgląd aukcji*. Pod podglądem aukcji stworzyłem sekcję komentarzy, które są widoczne dla każdej osoby (zalogowanej lub niezalogowanej). Poniżej komentarzy znajduje się *textarea*, z której dane pobierane są metodą POST a następnie wysyłane w postaci zapytania do bazy do tabeli *comments*. Komentarz jest przypisany do osoby, która go wysłała (w podglądzie komentarza na stronie aukcji widnieje avatar oraz nazwa użytkownika). Dodawać komentarze mogą tylko osoby zalogowane.

## 8.4 Sprint Review/Demo

Sprint został zakończony częściowym powodzeniem. Nie wszystkie założone zadania zostały zrealizowane w ustalonym czasie. Zadania które zostały w dniu dzisiejszym (09-01-2018) zaprezentowane to:

- Zostało utworzone usuwanie aukcji, dzięki któremu można usunąć swoją aukcję.
- Został utworzony podgląd aukcji, w którym można zobaczyć wszystkie informacje związane z daną aukcją.
- Został utworzony panel kategorii dzięki któremu możemy przejść do interesującej nas kategorii.

- Została stworzona sekcja komentarzy, dzięki której możemy jako zalogowany użytkownik dodawać komentarze pod aukcją. W czasie sprint review został zauważony problem z dodawaniem komentarzy, który został rozwiązany po dokładnym przeanalizowaniu kodu.

Zadania niezrealizowane, które przechodzą na następny sprint to:

- Wystąpiły niespodziewane komplikacje z kupowaniem przedmiotu.

## 9 Sprint 5

### 9.1 Cel

Celem piątego sprintu jest umożliwienie wysyłania wiadomości, możliwość wyświetlenia wszystkich swoich aukcji na stronie, kupowanie przedmiotu oraz otrzymanie informacji zwrotnej po kupieniu przedmiotu.

### 9.2 Sprint Planning/Backlog

**Tytuł zadania.** Wiadomości;

- Estymata: L;

**Tytuł zadania.** Informacja zwrotna;

- Estymata: M;

**Tytuł zadania.** Kupowanie przedmiotu;

- Estymata: XL;

**Tytuł zadania.** Moje aukcje;

- Estymata: L;

### 9.3 Realizacja

**Tytuł zadania.** Wiadomości

**Wykonawca.** Kuba Niewczas

**Realizacja.** WIP

**Tytuł zadania.** Informacja zwrotna

**Wykonawca.** Kuba Niewczas

**Realizacja.** WIP

**Tytuł zadania.** Kupowanie przedmiotu

**Wykonawca.** Kuba Niewczas

**Realizacja.** WIP

**Tytuł zadania.** Moje aukcje

**Wykonawca.** Kuba Niewczas

**Realizacja.** WIP

## 9.4 Sprint Review/Demo

### Literatura

- [1] S. R. Covey, *7 nawyków skutecznego działania*, Rebis, Poznań, 2007.
- [2] Tobias Oetiker i wsp., Nie za krótkie wprowadzenie do systemu L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>,  
<ftp://ftp.gust.org.pl/TeX/info/lshort/polish/lshort2e.pdf>
- [3] K. Schwaber, J. Sutherland, *Scrum Guide*, <http://www.scrumguides.org/>, 2016.
- [4] <https://agilepainrelief.com/notesfromatooluser/tag/scrum-by-example>
- [5] [https://www.tutorialspoint.com/scrum/scrum\\_user\\_stories.htm](https://www.tutorialspoint.com/scrum/scrum_user_stories.htm)