

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

**PROJEKT Z INTERNETOWYCH BAZ  
DANYCH**

**Aplikacja wspomagająca zarządzaniem budynkiem**

Termin zajęć: Czwartek, 14:15–16:55

AUTORZY:

Arkadiusz Sokołowski

Dmytro Verkhovsky

Indeks: 230435, 231844

PROWADZĄCY ZAJĘCIA:

Dr inż. Roman Ptak, W4/K9

Wrocław, 2018 r.

## **Spis treści:**

1.Wstęp .....	5
1.1.Analiza SWOT .....	5
1.2.Cel projektu .....	8
1.3.Zakres projektu .....	8
1.4.Techniczna wykonalność .....	8
1.5.Kosztorys .....	8
1.6.Harmonogram .....	9
2.Analiza wymagań .....	10
2.1.Wymagania funkcjonalne.....	10
2.2.Wymagania niefunkcjonalne.....	10
2.2.1.Wykorzystywane technologie i narzędzia .....	10
2.2.2.Wymagania dotyczące rozmiaru bazy danych.....	11
2.2.3.Wymagania dotyczące bezpieczeństwa systemu .....	11
3. Projekt Systemu .....	12
3.1. Ogólna Architektura projektu.....	12
3.2. Projekt Bazy Danych .....	12
3.2.1 Uproszczony model konceptualny .....	12
3.2.2 Model logiczny i normalizacji.....	13
3.2.3 Model fizyczny i ograniczenia integralności danych.....	14
3.3. Projekt Aplikacji użytkownika .....	20
3.3.1 Architektura aplikacji i diagramy projektowe .....	20
3.2.2 Projekt wybranych funkcjonalności systemu .....	22
3.4. Scenariusze przypadków użycia .....	30
3.5. Wybrane fragmenty kodu.....	34
3.6. Instrukcja wdrożeniowa .....	40
4. Tutorial .....	43
5. Testy .....	48
5. Podsumowanie .....	49

# Spis rysunków

Rysunek 1 — Ogólna architektura projektu .....	11
Rysunek 2 — Model konceptualny .....	11
Rysunek 3 — Model logiczny .....	12
Rysunek 4 — Model fizyczny .....	13
Rysunek 5 — Diagram Przypadek Użycia.....	19
Rysunek 6 — Okno logowania .....	19
Rysunek 7 — Okno rejestracji .....	20
Rysunek 8 — PU dodaj zamówienie.....	21
Rysunek 9 — PU EditOrder .....	22
Rysunek 10 — Dodanie nowego zamówienia .....	23
Rysunek 11 — Edycja istniejącego zamówienia.....	24
Rysunek 12 — Usunięcie domownika .....	25
Rysunek 13 — Sprawdzenie czy istnieje .....	26
Rysunek 14 — Sprawdzenie czy istnieje .....	27
Rysunek 15 — Diagram decyzyjny .....	28
Rysunek 16 — Instalacja XAMPP .....	40
Rysunek 17 — Menu główne XAMPP .....	41
Rysunek 18 — Menu Volumes .....	41
Rysunek 19 — Folder «htdocs» .....	41
Rysunek 20 — Directory .....	42
Rysunek 21 — Strona 192.168.64.2.....	42
Rysunek 22 — Tworzenie bazy .....	42
Rysunek 23 — Główna Strona .....	43
Rysunek 24 — Strona Rejestracji .....	43
Rysunek 24 — Rejestracja z sukcesem .....	44
Rysunek 26 — Logowanie .....	44
Rysunek 27 — Strona po zalogowaniu .....	45
Rysunek 29 — New Order .....	45
Rysunek 30 — Formularz dodawania zlecenia .....	45
Rysunek 31 — Przyciski .....	45
Rysunek 32 — Save .....	45
Rysunek 33 — Confirm .....	46
Rysunek 34 — Chowanie kolumn.....	46
Rysunek 35 — Add item to Storage .....	46
Rysunek 36 — Ekran dodawania zasobu do schowka .....	46
Rysunek 37 — Add user to house .....	47
Rysunek 38 — Formularz dodawania użytkownika .....	47
Rysunek 39 — Wykres pieniędzy wydanych przez użytkowników .....	47
Rysunek 40 — Lighthouse .....	48

# Spis Listingów

Listing 1 — Połączenie w PHP .....	34
Listing 2 — Komórki w tabelach .....	35
Listing 3 — Obsługa przycisków .....	37
Listing 4 — plik order.php .....	38
Listing 5 — Wstawienie do Bazy .....	39

## 1. Wstęp

### 1.1. Analiza SWOT

INTERNAL FACTORS	
STRENGTHS (+)	WEAKNESSES (-)
<p>Jak już wszyscy przyzwyczają się do korzystania z samej aplikacji, gdy np. Coś się zepsuje w domu, lub czegoś będzie brakować, będzie to znacznie prostsze i będzie nam dawało dużo więcej satysfakcji. Nasze rozwiązanie będzie również bardzo mobilne dzięki temu, że będzie to aplikacja internetowa. Będzie również darmowe, oraz łatwo skalowalne dzięki czemu będzie można je wykorzystać zarówno do mieszkania jak i nawet akademiku.</p>	<p>Podstawową słabą stroną naszego systemu będzie konieczność połączenia internetowego, oraz samego prądu, urządzenia z akumulatorem, które będzie w stanie pobrać z aplikacji co należy w domu zrobić, oraz czego brakuje.</p>

EXTERNAL FACTORS	
OPPORTUNITIES (+)	THREATS (-)
<p>Naszą główną szansą, jest zapotrzebowanie rynku, a ściślej brak konkretnych rozwiązań problemu, którym jest odpowiedni podział obowiązków między domowników. Same czynności wprowadzane do naszej aplikacji będą zawierały również czas który będzie można przełożyć bezpośrednio na tabelę, który domownik poświęcił ile z wolnego czasu dla domu w danym miesiącu, co z pewnością zapobiegnie wielu.</p>	<p>Nasza Szansa poniekąd przekłada się również na nasze zagrożenie, grupa docelowa, którą są ludzie, z pewnością przywykła do obecnego stanu rzeczy, czasem szybciej wysłać sms niż wpisać coś do aplikacji, co jednak według nas kończy się zazwyczaj brakiem równo rozłożonych obowiązków przez co u wielu domowników wytwarza się poczucie niesprawiedliwości.</p>

INTERNAL FACTORS	
STRENGTHS (+)	WEAKNESSES (-)
<ul style="list-style-type: none"> <li>● Szerokie grono odbiorców systemu (5%)</li> <li>● Wykwalifikowani i doświadczeni programiści tworzą system (10%)</li> <li>● Mobilność oprogramowania (35%)</li> <li>● Prosta i intuicyjna obsługa interfejsu użytkownika (20%)</li> <li>● Duża oszczędność czasu i pieniędzy (20%)</li> <li>● Podobne tematy pracy inżynierskiej (10%)</li> </ul>	<ul style="list-style-type: none"> <li>● Niewielkie środki finansowe (5%)</li> <li>● Ograniczony czas realizacji projektu (50%)</li> <li>● Duży natłok innych zajęć pracowników (35%)</li> <li>● Brak specjalistów od reklamy (10%)</li> </ul>

EXTERNAL FACTORS	
OPPORTUNITIES (+)	THREATS (-)
<ul style="list-style-type: none"> <li>● Wprowadzenie aplikacji do google play (5%)</li> <li>● Stworzenie systemu który podbije rynek (15%)</li> <li>● Zdobycie oceny 5.0 z „Internetowych baz danych” (30%)</li> <li>● Sklonność ludzi do korzystania z aplikacji dzięki którym oszczędzają czas i pieniądze (10%)</li> <li>● Sklonność ludzi do podporządkowania życia pod aplikacje mobilne (15%)</li> <li>● Brak podobnych systemów na rynku (25%)</li> </ul>	<ul style="list-style-type: none"> <li>● Ataki hackerów na system, przez co ludzie będą mogli ponieść duże koszty (DDOS, wykradanie oraz wprowadzanie nowych danych) (25%)</li> <li>● Baza danych oparta o SQL (60%)</li> <li>● Przyzwyczajenie ludzi do braku tego oprogramowania (25%)</li> </ul>

Wykwalifikowani i doświadczeni programiści tworzą system (10%) + Podobne tematy pracy inżynierskiej (10%) < Ograniczony czas realizacji projektu (50%) + Duży natłok innych zajęć pracowników

Strengths < Weakness -> 0.5

Duża oszczędność czasu i pieniędzy(20%) + Szerokie grono odbiorców systemu (5%)

+

Wykwalifikowani i doświadczeni programiści tworzą system (10%) > Niewielkie środki finansowe (5%) + Brak specjalistów od reklamy (10%)

Strengths > Weakness -> 0.35

Niewielkie środki finansowe (5%) + Brak specjalistów od reklamy (10%)

Weakness 0.15

Sklonność ludzi do korzystania z aplikacji dzięki którym oszczędzają czas i pieniądze (10%) + Sklonność ludzi do podporządkowania życia pod aplikacje mobilne (15%) = Przyzwyczajenie ludzi do braku tego oprogramowania (25%)

OPPORTUNITIES = THREATS -> 0

Brak podobnych systemów na rynku (25%) < Baza danych oparta o SQL (60%)

Threats -> 0.6

Wprowadzenie aplikacji do google play (5%) + Zdobycie oceny 5.0 z „Internetowych baz danych” (30%)

opportunities -> 0.35

Threats -> Ataki hackerów na system, przez co ludzie będą mogli ponieść duże koszty (DDOS, wykradanie oraz wprowadzanie nowych danych) (25%) threats

Weakness 0.65 Strengths 0.35

***Threats 0.6 opportunities 0.35***

Dlatego właśnie zdecydowaliśmy się na wybranie strategii Defensywnej.

## **1.2. Cel projektu**

Celem projektu jest stworzenie funkcjonalnego, prostego w obsłudze systemu dla ludzi który chcą zarządzać swoim domem, system ten będzie miał za zadanie wspierać domowników w dbaniu o dom, oraz o wszystkie jego zapasy. Aplikacja internetowa ma korzystać z bazy danych, aby ułatwić przetwarzanie informacji domu.

## **1.3. Zakres projektu**

- 1.Określenie wymagań stawianych oprogramowaniu, wstępne założenia projektowe
- 2.Projekt struktury bazy danych, mechanizmów poprawności i kontroli dostępu do danych
- 3.Implementacja i testy bazy danych
- 4.Projekt i implementacja aplikacji
- 5.Testy stworzonego oprogramowania

## **1.4. Techniczna wykonalność**

Jako, iż projekt trwa jedynie przez jeden semestr postanowiliśmy ograniczyć się do wąskiej grupy docelowej jaką jest rodzina zamieszkująca raczej w centrum zurbanizowanego miasta. Postanowiliśmy skupić się jedynie na domownikach, zasobach mieszkania, oraz podstawowych obowiązkach jakie wykonuję się w mieszkaniu. W zakresie domowników zajmiemy się jedynie ich dostępnością która będzie wyznaczana poprzez grafik, który każdy użytkownik będzie sam dla siebie tworzył, rozgraniczymy również czynności które może wykonywać każdy z domowników, oraz te do których domownik musi być pełnoletni, lub w pełni zdrowy fizycznie/psychicznie. W zasobach, oraz podstawowych obowiązkach rozważymy jedynie możliwości jakie występują w mieszkaniach, nie będziemy póki co rozważać innych typów - tj. Kawalerka, blok mieszkaniowy, dom parterowy, przyczepa campingowa, dom w gospodarstwie itd. Odnośnie technologii postanowiliśmy obsłużyć jedynie użytkowników zdolnych do korzystania z przeglądarki internetowej Safari 12.0 dostępnej dla macOS Mojave i tylko dla niej zapewnimy niezawodność w miarę naszych możliwości.

## **1.5. Kosztorys**

Przez to, że pracujemy na oprogramowaniu dostępnym bezpłatnie, koszt wytworzenia aplikacji będzie ograniczał się wynagrodzeniem naszej pracy i opłatą narzędzia potrzebnego do wsparcia naszej aplikacji.Nasze wynagrodzenie było by 14 tysięcy złotych w całości za 2 miesięczny okres dla 2 członków grupy, co jak łatwo przeliczyć przekłada się na pensję 3500 złotych miesięcznie dla każdego z nas. Też dla naszej strony będzie potrzebna domena i serwer na którym można było by to wszystko rozmieścić. Łączna cena za Hosting i Domenę na przykładowy

adres «homekeeper.eu» jest 500 zł na rok. Potrzebujemy też usługi która będzie pozytywnie wpływać na SEO (Search Engine Optimization zwana także pozycjonowaniem) która kosztuje 150 zł na rok. Nam nie jest potrzebna licencja MySQL bo nie nam nie jest potrzebne wsparcie od Oracle i my nie będziemy modyfikować kodu MySQL lub osadzać MySQL. Inne technologii z XAMPP są pod licencją GNU GPL.

Koszt aplikacji jest 14650 zł.

## 1.6. Harmonogram

Przygotowania do realizacji według nas kończy ten punkt, jako iż obecny semestr trwa dość krótko modelowanie zamierzamy skończyć do końca października, dzięki czemu przez cały listopad będziemy mogli pracować nad wdrażaniem bazy danych, oraz dodawaniem wszystkich funkcjonalności do aplikacji internetowej.

1. Przygotowanie do realizacji - 18.10.2018
2. Modelowanie aplikacji - 29.10.2018
3. Tworzenie bazy danych i aplikacji internetowej - 15.11.2018
4. Przygotowanie prezentacji - 29.11.2018

## **2. Analiza wymagań**

Wybór i opracowanie wstępnych założeń dotyczących wybranych tematów projektów.

### **2.1. Wymagania funkcjonalne**

1. Główna domu chcącą zarządzać swoim domem tworzy dom w naszej aplikacji podając jego nazwę adres numer telefonu stacjonarny oraz komórkowy swoje Imię, Nazwisko, Mail budynku, oraz mail administratora, oraz hasło. Od tego momentu w bazie istnieje dom, oraz jego administrator, który ma możliwość założenia konta wszystkim domownikom.
2. Wszyscy użytkownicy mają możliwość stworzenia zlecenia, czyli np. uzupełnienia zapasu, posprzątania jakiegoś pokoju, wymienienia żarówki. Mają również możliwość przypisania go do siebie.
3. Administrator ponadto jako jedyny ma możliwość usunięcia przypisania domownika do zlecenia
4. System umożliwi Administratorowi dodawanie, usuwanie oraz edycję danych domownika takich jak: imię nazwisko email hasło data urodzenia pesel numer telefonu ilość.
5. Wszyscy będą mogli wpisać swój grafik pracy, oraz zajęć obowiązkowych w celu obliczenia ile z wolnego czasu poświęcili domowi by była możliwość zrównoważenia obowiązków.

### **2.2. Wymagania niefunkcjonalne**

#### **2.2.1. Wykorzystywane technologie i narzędzia**

1. System powinien pracować na systemach macOS Mojave.
2. Technologia wykorzystana do stworzenia strony internetowej to HTML.
3. W ramach realizacji bazy danych będziemy korzystać z MariaDB SERVER.
4. Połączenie między serwerem a stroną wykonujemy za pomocą PHP.
5. Dla łatwiejszego zarządzania bazą korzystamy z phpMyAdmin.

## **2.2.2. Wymagania dotyczące rozmiaru bazy danych**

1. Baza danych może zawierać do 4000 klientów. Każdy klient ma wprowadzić swoje dane osobowe i też login z hasłem
2. Baza danych będzie przechowywać dane o zgłoszeniach za ostatnie 5 lat. Jeżeli będzie ze 200 zgłoszeń w tygodniu( co jest za dużo ) to za rok będzie mniej więcej z 9 mln zgłoszeń, za 5 lat 45 mln zgłoszeń.

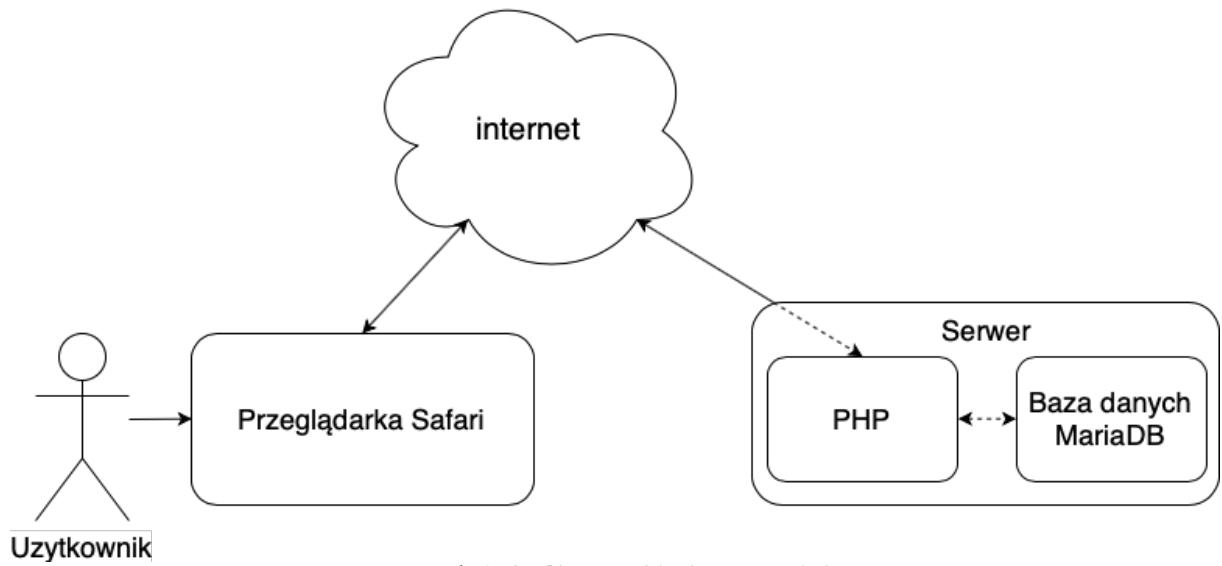
## **2.2.3. Wymagania dotyczące bezpieczeństwa systemu**

1. Dostęp do odpowiednich kont pracowników (klientów administratorów, pracowników) będzie możliwy poprzez wprowadzenie loginu i hasła. Hasła będą przechowywane w postaci zahashowanej. Sama baza będzie posiadała backup, który wyeliminuje utratę wszystkich danych.

### 3. Projekt systemu

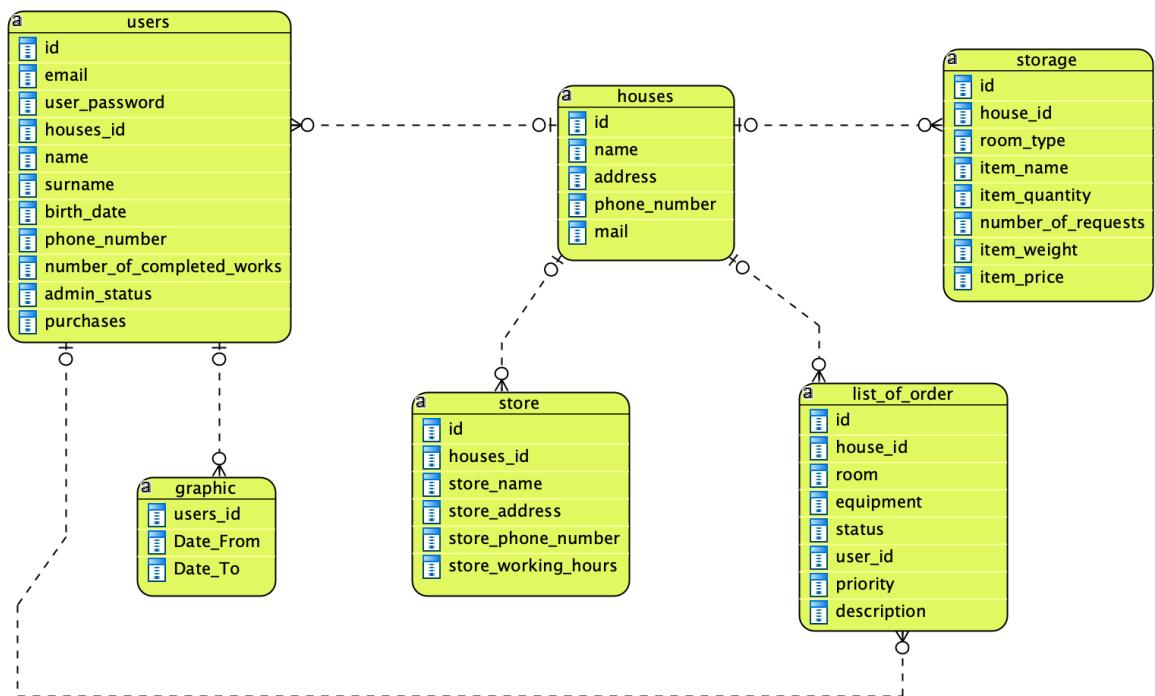
#### 3.1 Ogólna architektura projektu

#### 3.2. Projekt bazy danych



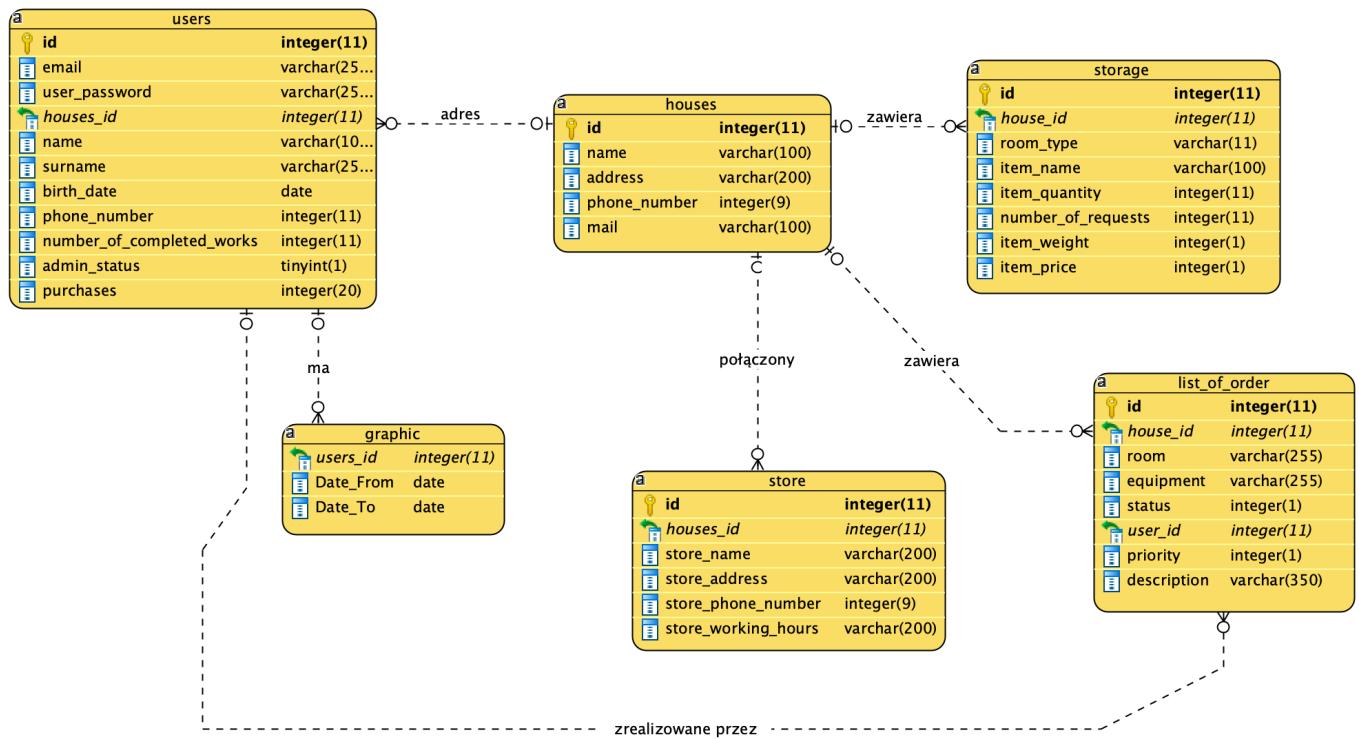
Rysunek 1 *Ogólna architektura projektu*

#### 3.2.1. Uproszczony model konceptualny



Rysunek 2 *Model konceptualny*

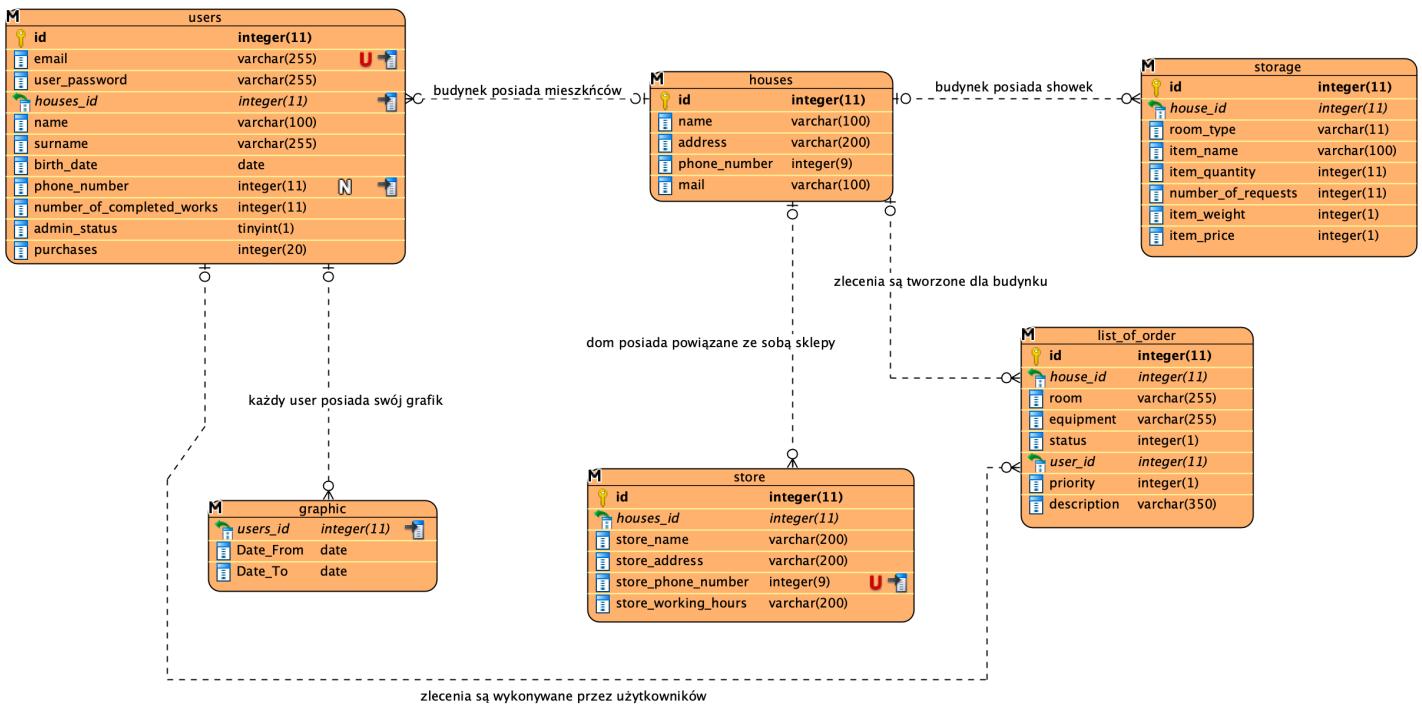
### 3.2.2. Model logiczny i normalizacja



Rysunek 3 *Model logiczny*

Baza danych jest znormalizowana na różnych płaszczyznach. Wprowadzone klucze główne zapobiegają powielaniu się informacji, jak również pomagają przy różnego typu operacjach.

### 3.2.3. Model fizyczny i ograniczenia integralności danych



Rysunek 4 *Model fizyczny*

Encja users zawiera dane dotyczące użytkownika aplikacji

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
id	Identyfikator osoby	INT(11)	+
email	Poczta osoby	Varchar(255)	+
user_password	Hasło osoby	Varchar(255)	+
houses_id	Identyfikator budynku osoby	INT(11)	+
name	Imię osoby	Varchar(100)	+
surname	Nazwisko osoby	Varchar(200)	+
birth_date	Data urodzenia osoby	Date	+
phone_number	Numer telefonu osoby	INT(11)	-
number_of_completed_works	Ilość wykonanych prac osoby	INT(11)	+
admin_status	Status osoby( czy jest administratorem czy nie jest)	tinyint(1)	+
purchases	ilość zakupów osoby	INT(20)	+

Tabela 1. Encja Users

Nazwa atrybutu	Znaczenie
<b>id</b>	Unikalny identyfikator klienta, automatycznie nadawany przez system, kucz główny
<b>email</b>	Email kilenta, dowolny ciąg znaków maksymalnie do 255 znaków, dowolne znaki, musi zawierać '@' i '.'
<b>user_password</b>	Hasło nadane przez klienta, dowolny ciąg znaków maksymalnie do 255, dowolne znaki
<b>houses_id</b>	Unikalny identyfikator budynku, Klucz obcy
<b>name</b>	Imię klienta, dowolny ciąg znaków o maksymalnej długości do 100 znaków, zaczyna się z wielkiej litery, dozwolone litery polskie
<b>surname</b>	Nazwisko klienta, dowolny ciąg znaków o maksymalnej długości do 200 znaków, zaczyna się z wielkiej litery, dozwolone litery polskie
<b>birth_date</b>	Data urodzenia klienta, ciąg znaków o długości 8, podany przez klienta, tylko liczby formatu date
<b>phone_number</b>	Telefon użytkownika, ciąg o długości 9, zawiera tylko liczby
<b>number_of_completed_works</b>	Ilość wykonanych prac użytkownika, zawiera tylko liczby, maksymalny rozmiar 11 znaków
<b>admin_status</b>	Status osoby, czy jest administratorem czy nie jest, zawiera tylko liczbę, długość 1
<b>purchases</b>	Ilość zakupów osoby, zawiera tylko liczby, maksymalny rozmiar 20 znaków

Tabela 2. Encja Users

Encja store zawiera dane dotyczące sklepów do zakupów

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
<b>id</b>	Identyfikator sklepu	INT(11)	+
<b>houses_id</b>	Identyfikator budynku	INT(11)	+
<b>store_name</b>	Nazwa sklepu	Varchar(200)	+
<b>store_adress</b>	Adress sklepu	Varchar(200)	+
<b>store_phone_number</b>	Numer telefonu sklepu	INT(9)	+
<b>store_working_hours</b>	Czasy pracy sklepu	Varchar(200)	+

Tabela 3. Encja store

Nazwa atrybutu	Znaczenie
id	Unikalny identyfikator sklepa, automatycznie nadawany przez system, kucz główny
houses_id	Unikalny identyfikator budynku, Klucz obcy, tylko liczby o długości 11
store_name	Nazwa sklepu, ciąg znaków o długości 200
store_adress	Adres sklepu, ciąg znaków o długości 200
store_phone_number	Numer telefonu sklepu, ciąg znaków o długości 9, tylko liczby
store_working_hours	Czasy pracy sklepu, ciąg znaków o długości 200

Tabela 4. Encja store

Encja storage zawiera dane dotyczące przechowywanych rzeczy w showku

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
id	Identyfikator pracownika	int(11)	+
house_id	Identyfikator budynku	int(11)	+
room_type	Typ pokoju	Varchar(11)	+
item_name	Nazwa przedmiotu	Varchar(100)	+
item_quantity	Ilość przedmiotów	int(11)	+
number_of_requests	Ilość zapytań o przedmiot	int(11)	+
item_weight	Waga przedmiotu	int(1)	+
item_price	Cena przedmiotu	int(1)	+

Tabela 5. Encja storage

Nazwa atrybutu	Znaczenie
id	Unikalny identyfikator przedmiota, automatycznie nadawany przez system, kucz główny
house_id	identyfikator budynku, Klucz obcy, tylko liczby o długości 11
room_type	Typ pokoju w którym ten przedmiot może być zainstalowany, dowolny ciąg znaków maksymalnie do 11
item_name	Nazwa przedmiotu znajdującego się w showku, dowolny ciąg znaków maksymalnie do 100
item_quantity	Ilość przedmiotów znajdujących się w showku, tylko liczby o długości 11
number_of_requests	Ilość zapytań o przedmiot, tylko liczby o długości 11
item_weight	Waga przedmiotu, tylko liczby o długości 1
item_price	Cena przedmiotu, tylko liczby o długości 1

*Tabela 6. Encja storage*

Encja list\_of\_order zawiera dane dotyczące zamówień

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
id	Identyfikator zamówienia	int(11)	+
house_id	Identyfikator budynku	int(11)	+
room	Numer pokoju	Varchar(100)	+
Equipment	Nazwa przedmiotu	Varchar(100)	+
status	Status zamówienia	int(1)	+
user_id	id użytkownika	int(11)	+
Priority	Prioritet zamówienia	int(1)	+
description	Opis zamówienia	Varchar(350)	+

*Tabela 7. Encja list\_of\_order*

Nazwa atrybutu	Znaczenie
id	Unikalny Identyfikator zamówienia, automatycznie nadawany przez system, kucz główny
house_id	identyfikator budynku, Klucz obcy, tylko liczby o długości 11
room	Numer pokoju w którym się znajduje przedmoit do naprawy, ciąg znaków o maksymalnej długości 100
Equipment	Nazwa przedmiotu do naprawy, ciąg znaków o maksymalnej długości 100
status	Status zamówienia, może być 0, 1 lub 2 co oznacza że jest wolny, w trakcie naprawy i naprawiony
user_id	Id użytkownika który zgłosił problem, klucz obcy o długości 11
Priority	Priorytet wykonania zlecenia integer długości 1
description	Opis zamówienia, opisuję jakieś dodatkowe pojęcia dotyczące zamówienia, komentarzy użytkownika, ciąg znaków o maksymalnej długości 350

Tabela 8. Encja list\_of\_order

Encja houses zawiera dane dotyczące budynków

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
id	Identyfikator budynku	int(11)	+
Name	nazwa budynku	Varchar(100)	+
Address	Adres budynku	Varchar(200)	+
phone_number	Numer telefonu budynku	int(9)	+
Mail	Poczta budynkowa	Varchar(100)	+

Tabela 9. Encja houses

Nazwa atrybutu	Znaczenie
id	Unikalny Identyfikator budynku, automatycznie nadawany przez system, kucz główny
Name	Nazwa budynku, ciąg znaków o maksymalnej długości 100
Address	Adres budynku, ciąg znaków o maksymalnej długości 200
phone_number	Numer telefonu budynku, ciąg znaków o maksymalnej długości 9, tylko cyfry
Mail	Adres pocztowy budynku, ciąg znaków o maksymalnej długości 100

*Tabela 11. Encja houses*

Encja graphic zawiera dane dotyczące grafika pracowników

Nazwa atrybutu	Opis atrybutu	Typ	OBL (+)/ OPC (-)
user_id	Identyfikator użytkownika	int(11)	+
Date_from	Data od której	datetime	+
Date_to	Data do której	datetime	+

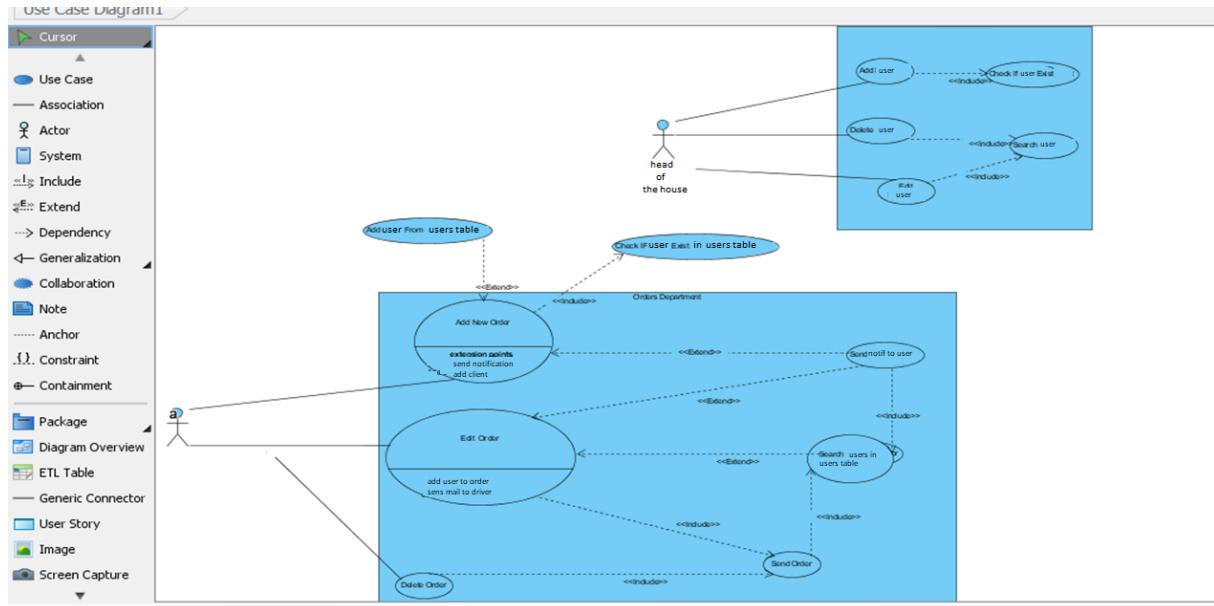
*Tabela 12. Encja graphic*

Nazwa atrybutu	Znaczenie
user_id	Unikalny Identyfikator użytkownika, automatycznie nadawany przez system, kucz główny
Date_from	Data od której użytkownik pracuje
Date_to	Data do której użytkownik pracuje

*Tabela 13. Encja graphic*

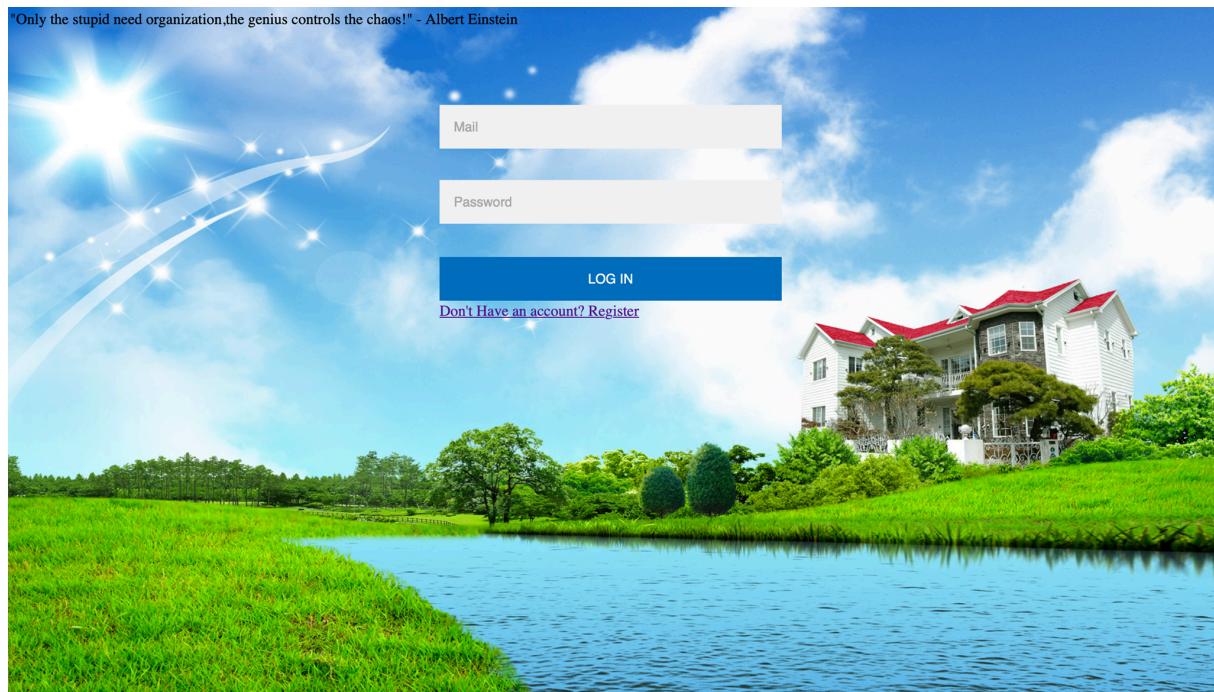
### 3.3 Projekt aplikacji użytkownika

#### 3.3.1 Architektura aplikacji i diagramy projektowe



Rysunek 5 *Diagram Przypadków Użycia*

#### 3.3.2 Architektura aplikacji i diagramy projektowe



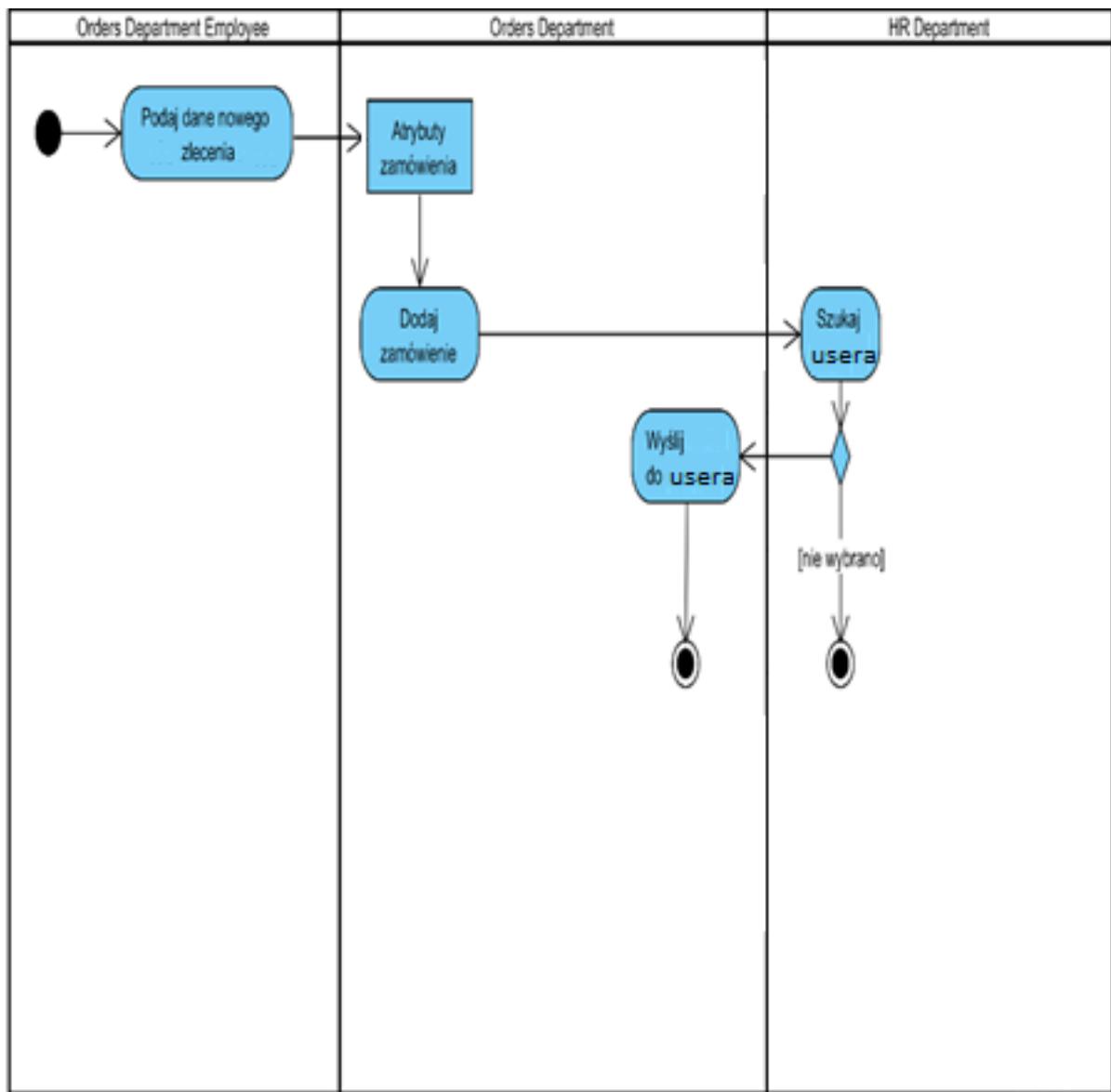
Rysunek 6 *Okno logowania*

A registration form window with a background image of a white house with a red roof situated near a body of water. The form consists of several input fields and buttons:

- Landline number
- Phone number
- Name
- Surname
- Date of Birth: Year Month Day
- House E-Mail
- Admin Mail
- Password
- Repeat Password
- Accept terms of usage
- Nie jestem robotem
- reCAPTCHA  
Prywatność - Warunki
- ZAREJESTRUJ SIĘ
- Cofam

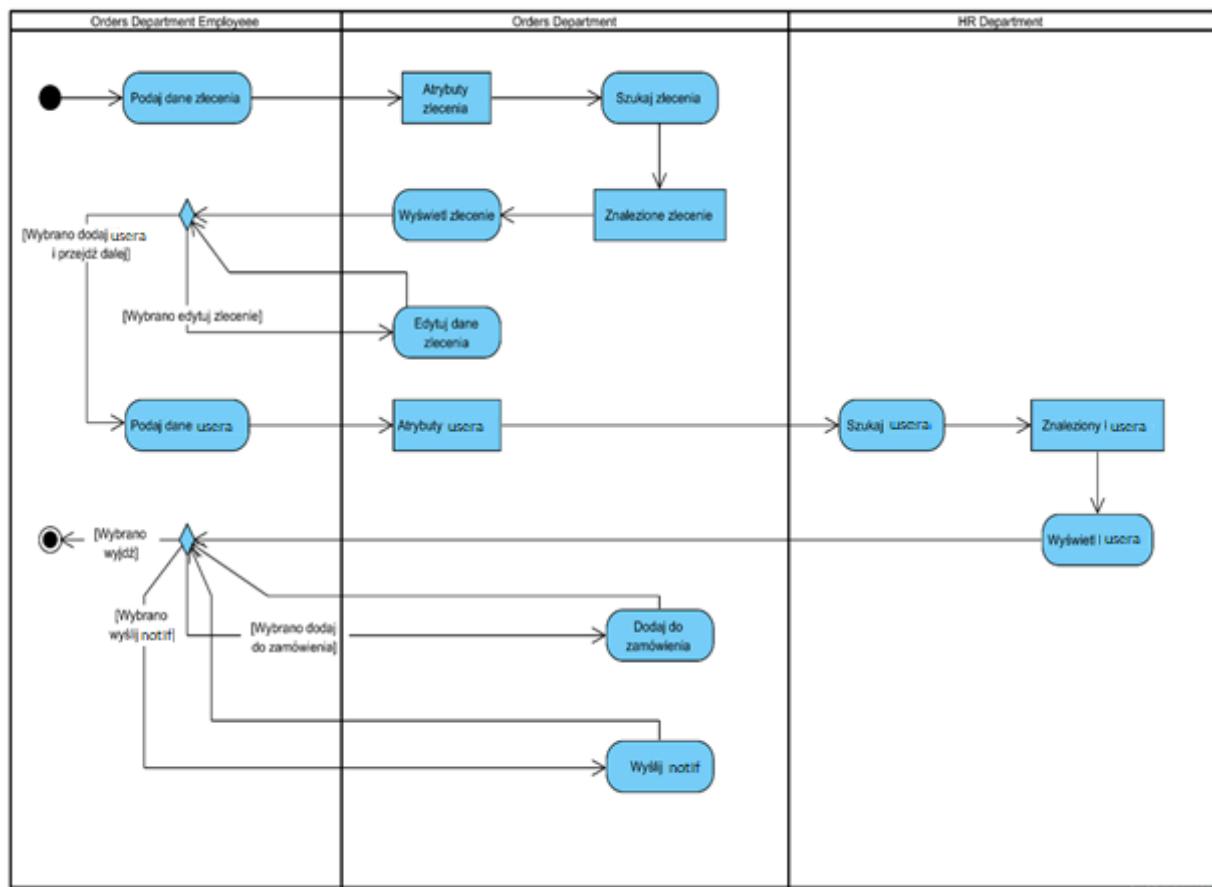
Rysunek 7 *Okno rejestracji*

### 3.3.3 Projekt wybranych funkcjonalności systemu



Rysunek 8 *PU dodaj zamówienie*

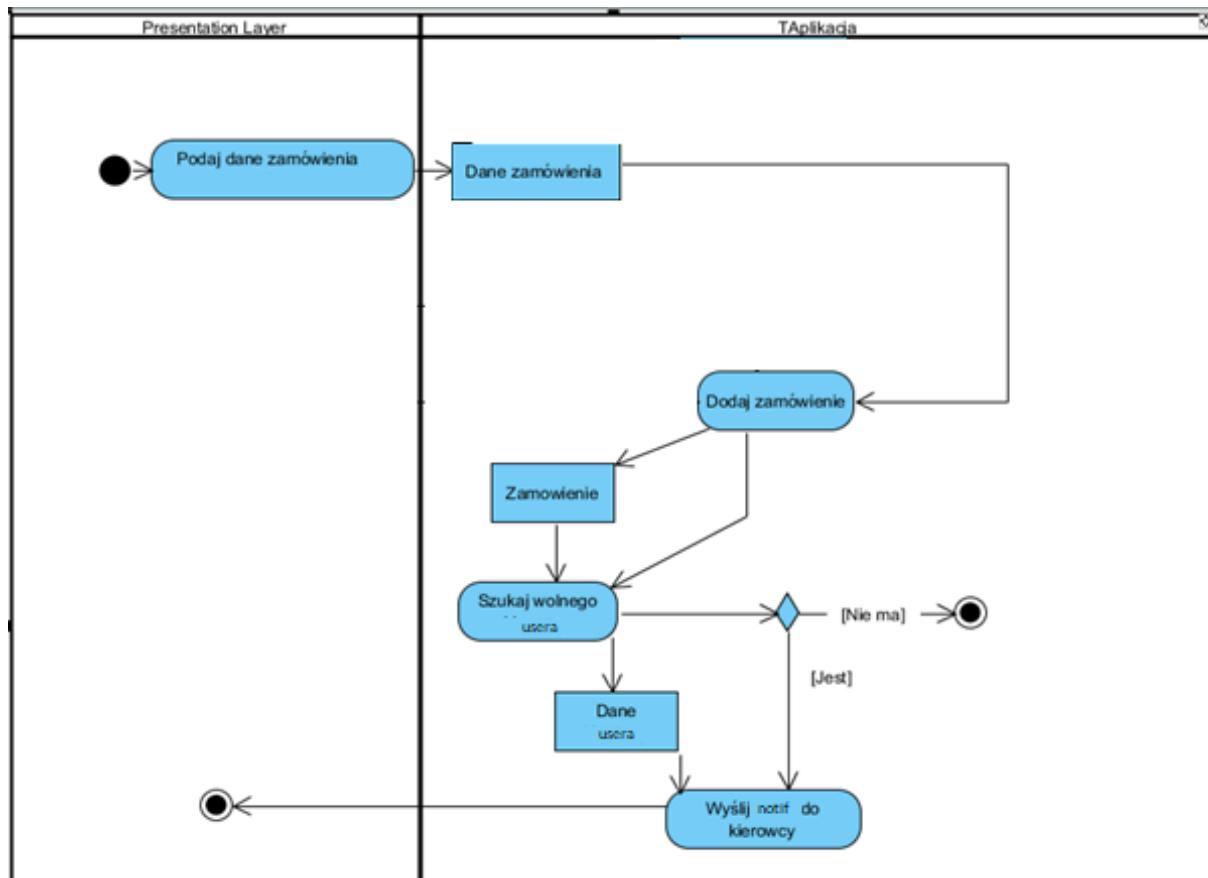
Diagram aktywności dla PU *EditOrder*:



Rysunek 9 *PU EditOrder*

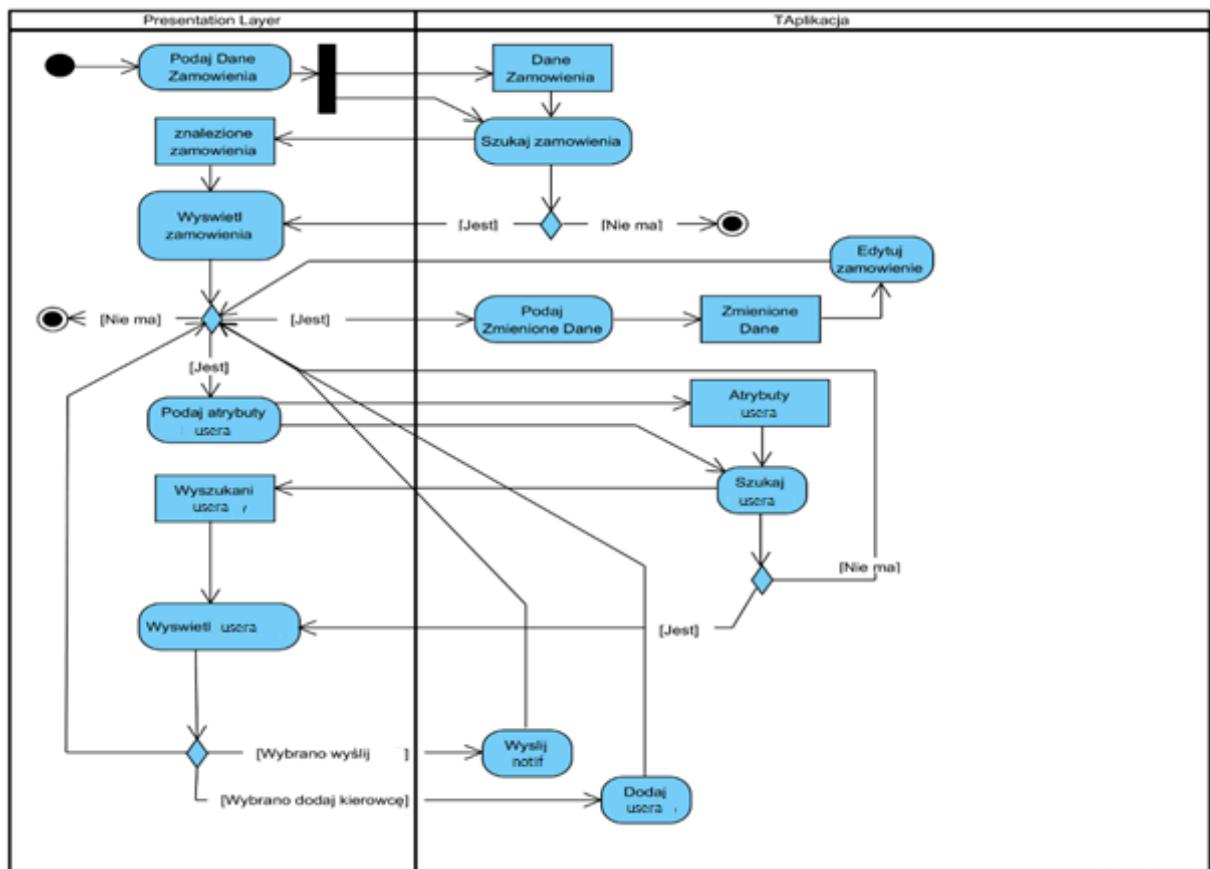
Diagramy czynności i stanów (definiowanie zachowania przypadków użycia) :

*Dodanie nowego zamówienia:*



Rysunek 10 **Dodanie nowego zamówienia**

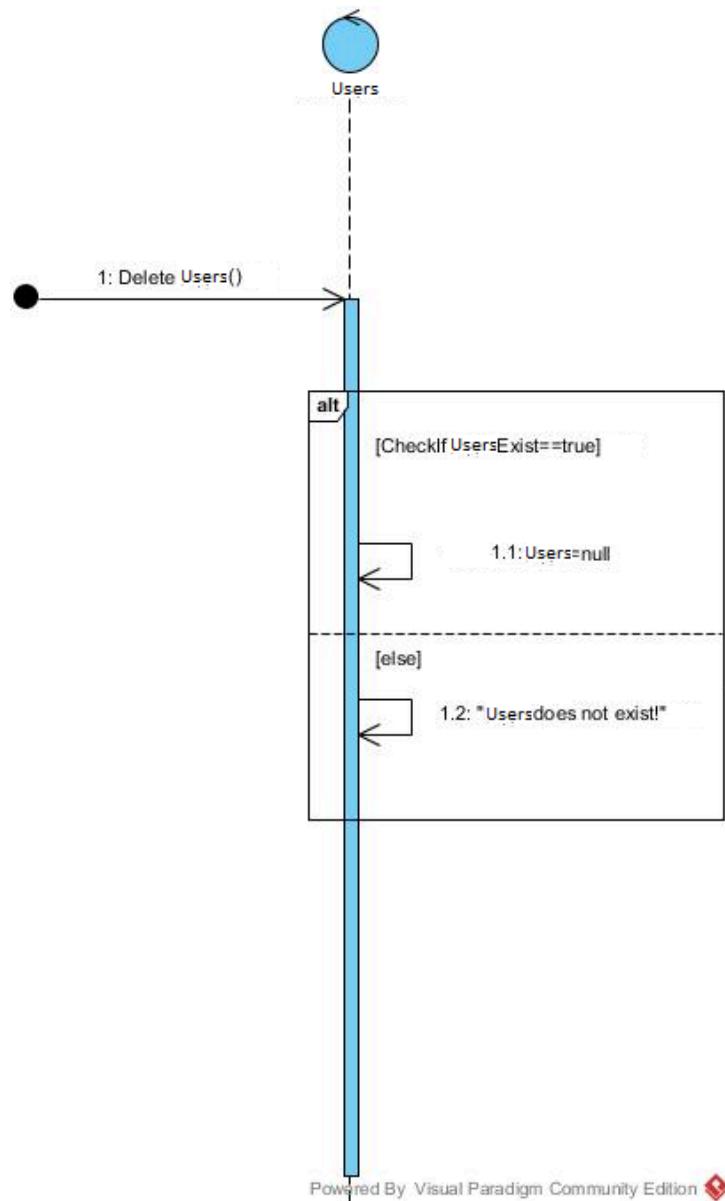
*Edycja istniejącego zamówienia:*



Rysunek 11 *Edycja istniejącego zamówienia*

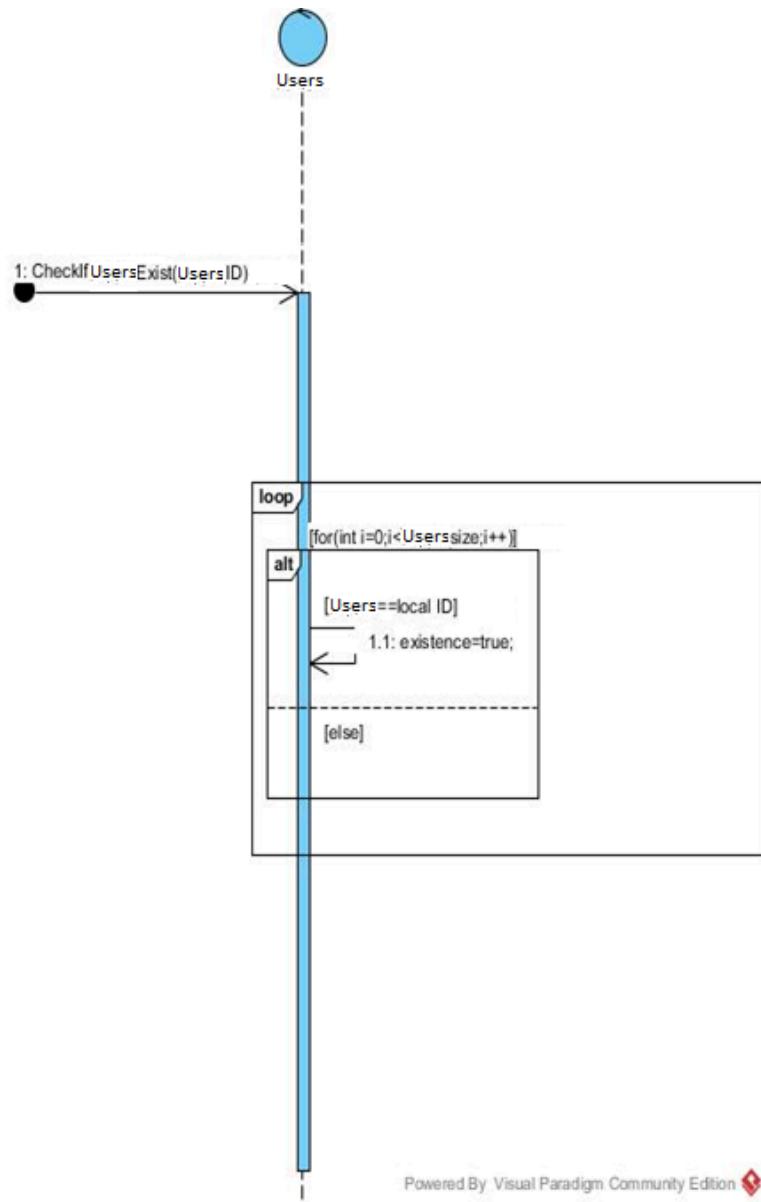
## Diagramy sekwencji

*Usunięcie domownika:*



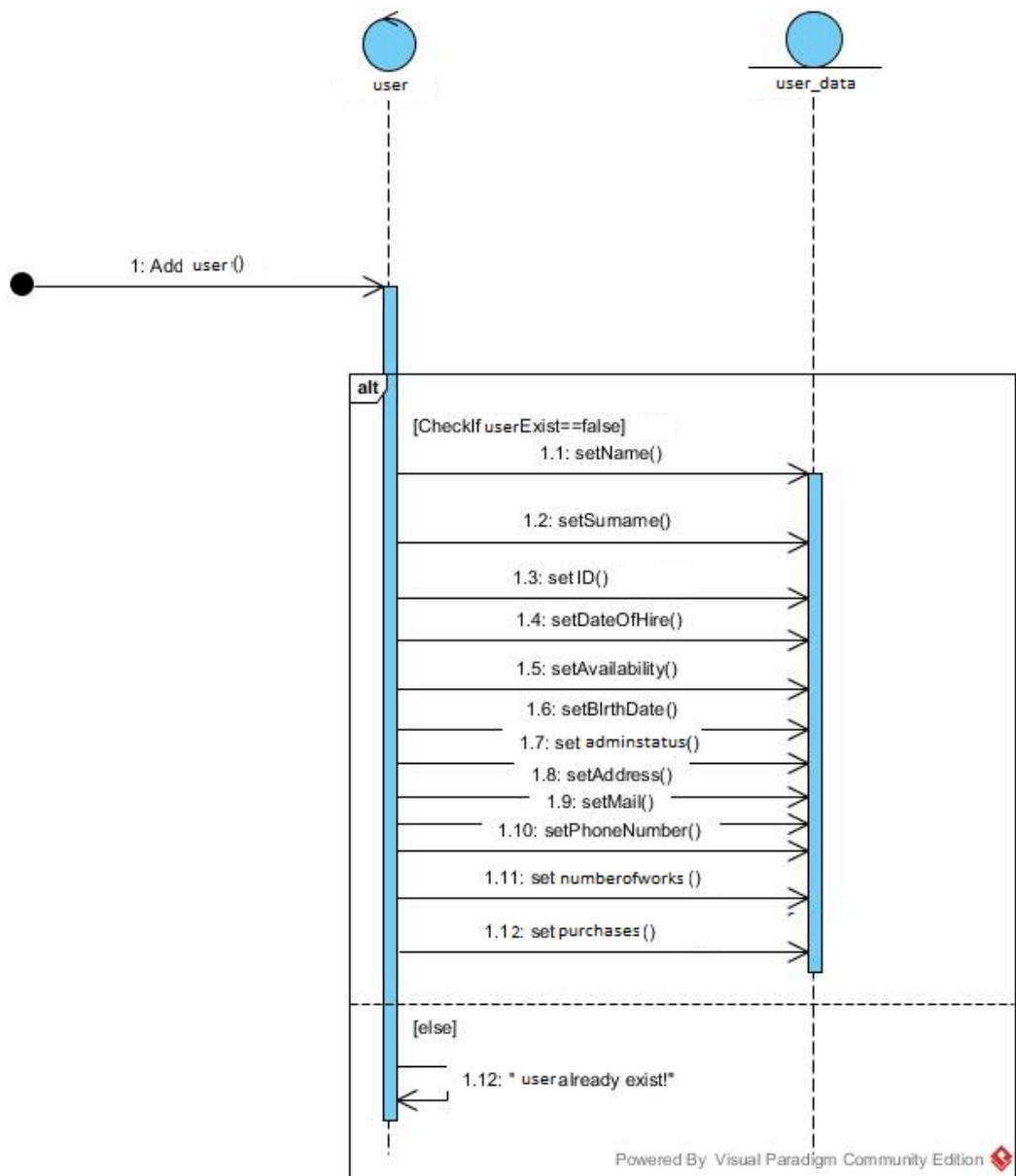
Rysunek 12 *Usunięcie domownika*

*Sprawdzenie czy istnieje:*



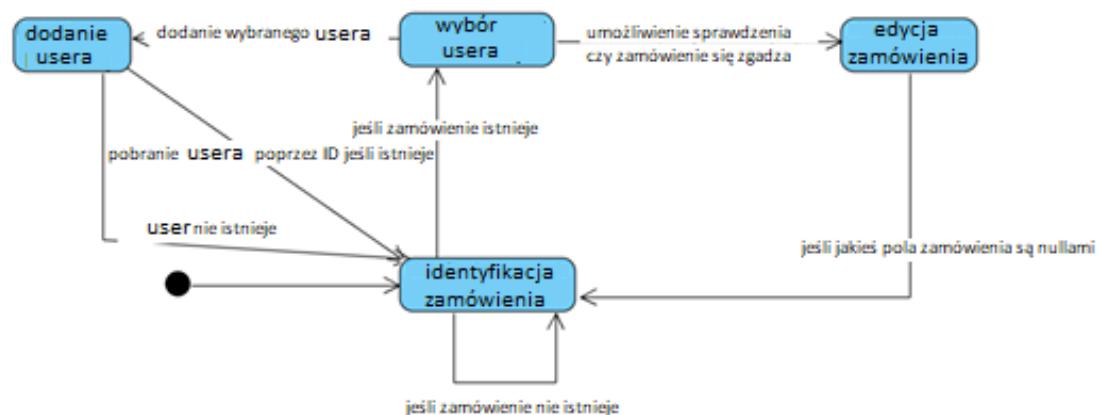
Rysunek 13 *Sprawdzenie czy istnieje*

*Dodanie użytkownika:*



Rysunek 14 *Dodanie użytkownika*

*Diagram decyzyjny:*



Rysunek 15 **Diagram decyzyjny**

## 3.4 Scenariusze przypadków użycia

### PU Rejestracja

Cel: Utworzenie nowego konta w systemie.

Warunki początkowe: Brak

Warunki końcowe: Konto klienta zostaje utworzone w systemie, wraz z profilem jego domu, którego pierwszym administratorem staje się użytkownik. Klient może się -

### PU Logowanie

Scenariusz:

1. Użytkownik wchodzi na stronę główną systemu
2. Użytkownik wybiera opcję „Zarejestruj się”.
3. System prezentuje formularz rejestracji
4. Użytkownik wypełnia formularz, po czym zatwierdza go i wysyła.
5. Konto Użytkownika zostaje utworzone.

### PU Logowanie użytkownika

Cel: Uzyskanie dostępu do funkcjonalności systemu wymagających uwierzytelnienia.

Warunki początkowe: brak

Warunki końcowe: Użytkownik zostaje zalogowany do systemu

Scenariusz:

1. Użytkownik wchodzi na stronę główną systemu
2. Użytkownik wypełnia formularz logowania, który znajduje się na głównej stronie
3. Użytkownik wybiera opcję „Zaloguj się”
4. Użytkownik wypełnia formularz podając swój email oraz hasło, po czym go wysyła.  
4b. W przypadku nieprawidłowych danych użytkownik dostaje o tym informacje oraz może powtórzyć krok 4
5. Użytkownik zostaje zalogowany

## **PU Dodanie zlecenia**

Cel: Dodanie nowego zlecenia w naszym domu.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: zlecenie zostaje dodane do domu, do którego należy użytkownik, oraz zostaje przypisane do podanego użytkownika.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Użytkownik wybiera opcję „Add Order”
3. Użytkownik wypełnia komórki: *Room*, *Equipment*, *Description*, oraz wybiera interesujące go opcje z list odpowiadających *statusowi*, *użytkownikowi*, oraz *Priorytetowi* który znajdujące się na stronie dodania zlecenia
4. Użytkownik wybiera Submit-dodanie zlecenia, lub Add More-dodanie wiersza, który również należy uzupełnić zgodnie z instrukcją w kroku trzecim, lub Back-jeśli nie interesuje go dodanie zlecenia
5. Po wybraniu opcji submit, Użytkownik wraca na główny ekran aplikacji, lub dostaje informację o niepowodzeniu wraz z jego przyczyną

## **PU Przeglądanie swoich, oraz niczych zleceń**

Cel: Wgląd w zlecenia, które są przypisane do domu, do którego należymy.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Użytkownikowi zostaje wyświetlona odpowiednia lista zleceń.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Znajdując się na głównej stronie (lub jednej z podstron, oprócz podstrony dodawania użytkownika, oraz dodawania zlecenia) użytkownik wybiera, „My orders”
3. Użytkownikowi zostaje wyświetlona odpowiednia lista.

## **PU Przeglądanie zleceń całego domu**

Cel: Wgląd w zlecenia, które są przypisane do domu, do którego należymy.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Użytkownikowi zostaje wyświetlona odpowiednia lista zleceń.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Znajdując się na głównej stronie (lub jednej z podstron, oprócz podstrony dodawania użytkownika, oraz dodawania zlecenia) użytkownik wybiera, opcję „List of Orders”
3. Użytkownikowi zostaje wyświetlona lista

## **PU Przeglądanie historii zleceń całego domu**

Cel: Wgląd w zlecenia, które są przypisane do domu, do którego należymy.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Użytkownikowi zostaje wyświetlona odpowiednia lista zleceń.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Znajdując się na głównej stronie (lub jednej z podstron, oprócz podstrony dodawania użytkownika, dodawania zlecenia, oraz dodawania zasobu do schowka) użytkownik wybiera, opcję „History of Orders”
3. Użytkownikowi zostaje wyświetlona lista

## **PU Przeglądanie zasobów w schowka**

Cel: Wgląd w listę zasobów w schowku naszego domu.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Użytkownikowi zostaje wyświetlona lista z wszystkimi zasobami w domu.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Znajdując się na głównej stronie (lub jednej z podstron, oprócz podstrony dodawania użytkownika, oraz dodawania zlecenia), użytkownik wybiera „Storage”
3. Następnie użytkownikowi zostaje wyświetlona lista zasobów dostępnych w domu

## **PU Dodanie zasobu do schowka**

Cel: Dodanie zasobu do schowka naszego domu.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Do schowka należącego do domu zostaje dodany zasób.

Scenariusz:

4. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
5. Znajdując się na stronie Storage użytkownik wybiera, opcję „Add Item to Storage”
6. Następnie podobnie jak przy dodawaniu zleceń należy wypełnić odpowiednie komórki, w tym przypadku Room, Equipment, Quantity, Number of requests, Price, oraz wybranie z listy wagi.
7. Użytkownikowi zostaje wyświetlona lista

## **PU Przeglądanie użytkowników w domu**

Cel: Wgląd w listę użytkowników w schowku naszego domu.

Warunki początkowe: Posiadanie konta w systemie, oraz zalogowanie się.

Warunki końcowe: Użytkownikowi zostaje wyświetlona lista z wszystkimi użytkownikami w domu.

Scenariusz:

1. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika**
2. Znajdując się na głównej stronie (lub jednej z podstron, oprócz podstrony dodawania użytkownika, oraz dodawania zlecenia), użytkownik wybiera „Users”
3. Następnie użytkownikowi zostaje wyświetlona lista użytkowników dostępnych w domu

## **PU Dodanie użytkownika do domu**

Cel: Dodanie użytkownika naszego domu.

Warunki początkowe: Posiadanie konta administratora w systemie, oraz zalogowanie się.

Warunki końcowe: Do naszego domu zostaje dodany użytkownik.

Scenariusz:

4. Użytkownik poprawnie wykonuje **PU Logowanie użytkownika** logując się na konto administratora
5. Znajdując się na stronie Storage użytkownik wybiera, opcję „Add New User”
6. Następnie podobnie jak przy dodawaniu zleceń należy wypełnić odpowiednie komórki, w tym przypadku Phone number, Name, Surname, Date of Birth, Account Type, Mail, Password, Repeat Password .
7. Ostatnim krokiem jest wybór opcji Add User, lub powrót do poprzedniego ekranu.

### 3.5 Wybrane fragmenty kodu

Poniższy fragment kodu odpowiada za ustalenie połączenia na podstawie zmiennych, które znajdują się w osobnym pliku dla lepszego bezpieczeństwa. Ponadto wykonuje kilka ustawień dla bazy by było możliwe używanie znaków w konwencji utf-8. Następnie do zmiennej records przypisuje odpowiedź z „*Społenie*” na którym wykonuje query \$sql czyli pobór wszystkich zleceń użytkownika, oraz tych nieprzypisanych do nikogo. W tym fragmencie postanowiłem również stworzyć tablicę z imionami wszystkich użytkowników by móc w łatwy sposób stworzyć nowe zlecenie wybierając użytkownika z wszystkich dostępnych w naszym domostwie.

```
$polaczenie = new mysqli($host,$db_user,$db_password,$db_name);
$sql="SELECT * FROM list_of_order where user_id=$houses_id or
user_id=$houses_id*-1";
$polaczenie->query('set character_set_server=utf8');
$polaczenie->query('set character_set_client=utf8');
$polaczenie->query('set character_set_connection=utf8');
$polaczenie->query('set character_set_results=utf8');
$polaczenie->query('set character_set_server=utf8');
$records=$polaczenie->query($sql);
$usersql="SELECT name FROM users where houses_id=$houses_id ";
$usertable=$polaczenie->query($usersql);
while( $user=$usertable->fetch_assoc() ) {
    foreach( $user AS $value ) {
        $clmNames[] = $value;
    }
}
$numberofusers = count($clmNames);
```

Listing 1 – Połączenie w PHP

Kolejnym etapem, którego należało się podjąć było stworzenie komórek tabeli, z których to Status User i Priority są listami, reszta natomiast zwykłymi polami(id, oraz house są nieedytowalne). Oprócz tabeli znajdują się również przyciski z których Add More dodaje nowy wiersz, dzięki któremu możemy dodać kilka zleceń jednocześnie, Submit odpowiada za wysłanie „*Insert*” z nowymi zleceniami do bazy, oraz back umożliwiający powrót do poprzedniej strony.

```
<table style="background: rgba(255,255,255,0.6); " class="table table-bordered"
id="dynamic_field">
<tr>
  <td><input type="text" name="id[]" value="Id: 1" class="form-control house_list"
readonly/></td>
  <td><input type="text" name="name[]" value='<?php
echo 'house: '.$house_name['name'];?>' class="form-control house_list" readonly/></td>
  <td><input type="text" name="room[]" placeholder="Room" class="form-control room_list" /></td>
  <td><input type="text" name="equipment[]" placeholder="Equipment" class="form-control equipment_list" /></td>
  <td>
    <meta charset="UTF-8">

    <select name="status[]">
      <option value="0">To Do</option>
      <option value="1">Doing</option>
      <option value="2">Done</option>
      <option value="3">Deleted</option>
    </select>
  </td>
  <td class="col1">
    <select name="user[]" style="width:100%">
      <?php
        foreach($clmNames as $usernames) {
      ?>
        <option
          <?php
            for($q=0;$q<sizeof($clmNames);$q++)
            {
              if($usernames==$clmNames[$q])
              {
                $nameofuser=$clmNames[$q];
                $usersql="SELECT id FROM users WHERE
name LIKE '$usernames'";
                $usertable=$mysqli->query($usersql);
                $user=$usertable->fetch_assoc();
                $uszer=$user['id'];
              }
            }
          ?>value=<?php echo $uszer; ?>
        <?php $result = $usernames;
        echo $result; ?> </option>
      <?php
        }
      ?>
    </select>
  </td>
</tr>
```

```
<?php } ?>
</select>
</td>
<td>
<select name="priority[]" class="col1">
<option value="0">Low</option>
<option value="1">Medium</option>
<option value="2">High</option>
</select>
</td>
<td><input type="text" name="description[]" placeholder="Description" class="form-control description_list" /></td>

<td><button type="button" name="add" id="add" class="btn btn-success">Add More</button></td>
</tr>
</table>
<input type="button" name="submit" id="submit" class="btn btn-info" value="Submit" />
<input type="button" name="back" id="back" class="btn btn-info" value="Back" onclick="history.back();"/>
```

Listing 2 – Komórki w tabelach

Ważne było dodanie też Scriptów, które umożliwią obsługę tych przycisków poprzez odwołanie ich do odpowiednich fragmentów kodu znajdujących się w innym pliku .php

```
<script>
$(document).ready(function(){
    var i=1;
    $('#add').click(function(){
        i++;
        $('#dynamic_field').append('<tr id="row'+i+'><td><input type="text" name="id[]" value="Id: '+i+'" class="form-control house_list" readonly/></td><td><input type="text" name="name[]" placeholder="<?php echo 'house: '.$house_name['name']?>" class="form-control name_list" readonly/></td><td><input type="text" name="room[]" placeholder="Room" class="form-control room_list" /></td><td><input type="text" name="equipment[]" placeholder="Equipment" class="form-control equipment_list" /></td><td><select name="status[]> <option value="0">To Do</option><option value="1">Doing</option><option value="2">Done</option><option value="3">Deleted</option> </select></td><td class="col1"><select name="user[]" style="width:100%"><?php foreach($clmNames as $usernames) { ?> <option <?php for($q=0;$q<sizeof($clmNames);$q++){if($usernames==$clmNames[$q]){$nameofuser=$clmNames[$q];$usersql="SELECT id FROM users WHERE name ='".$usernames."";$usertable=$mysqli->query($usersql);$user=$usertable->fetch_assoc();$uszer=$user['id'];}}?>value=<?php echo $uszer; ?>><?php echo $usernames; ?> </option><?php } ?></select> </td><td><select name="priority[]" class="col1"><option value="0">Low</option><option value="1">Medium</option><option value="2">High</option></select></td><td><input type="text" name="description[]" placeholder="Description" class="form-control description_list" /></td><td><button type="button" name="remove" id="'+i+'" class="btn btn-danger btn_remove">X</button></td></tr>');
    });
    $(document).on('click', '.btn_remove', function(){
        var button_id = $(this).attr("id");
        $('#row'+button_id+'').remove();
        i--;
    });

    $('#submit').click(function(){
        $.ajax({
            url:"order.php",
            method:"POST",
            data:$('add_order').serialize(),
            contentType: "application/x-www-form-urlencoded; charset=utf-8",
            success:function(data)
            {
                alert(data);
                $('#add_order')[0].reset();
            }
        });
    });
});
</script>
```

Listing 3 – Obsługa przycisków

Skoro odwołuje się tu do pliku order.php to postanowiłem również pokazać i jego składnie wraz z krótkim omówieniem. Jako iż dodawanie zlecenia opiera się na 3 polach tekstowych warto sprawdzić czy wszystkie z nich są wypełnione, oraz przypisać wartości z \$\_POST do zmiennych.

```
$number1 = count($_POST["room"]);
$number2 = count($_POST["equipment"]);
$number3 = count($_POST["description"]);
$value1 = $_POST["room"];
$value2 = $_POST["equipment"];
$value3 = $_POST["status"];
$value4 = $_POST["user"];
$value5 = $_POST["priority"];
$value6 = $_POST["description"];
```

Listing 4 – plik order.php

Teraz jeśli każda z nich istnieje chociaż raz, oraz ich liczba jest taka sama, jesteśmy w stanie wykonać Insert do bazy danych w taki sposób by nowe zlecenie było przypisane zgodnie z wprowadzonymi przez nas danymi, oraz było ono przypisane do naszego domu. W zależności od wyniku dostajemy komunikat, albo o poprawnym dodaniu zlecenia, albo o porażce. Wartości również tuż pod forem miały być oczyszczane z niepotrzebnych oraz wadliwych znaków, lecz nie znalazłem jeszcze sposobu, by oczyszczenie pozostawiało znaki UTF-8.

```

if(( $number1 >= 1 && $number2 >= 1 && $number3>=1 )&&($number1==$number2 &&
$number2==$number3))
{
    for($i=0; $i<$number1; $i++)
    {
        $value1_clean = $value1[$i];
        $value2_clean = $value2[$i];
        $value3_clean = $value3[$i];
        $value4_clean = $value4[$i];
        $value5_clean = $value5[$i];
        $value6_clean = $value6[$i];

        if($value1_clean != '' && $value2_clean != '' && $value3_clean != '' &&
$value4_clean != '' && $value5_clean != '' && $value6_clean != '')

        {
            $sql = 'INSERT INTO
list_of_order(house_id,room,equipment,status,user_id,priority,description)
VALUES('.$houses_id.'', '.$value1_clean.'', '.$value2_clean.'', ''.
$value3_clean.'', '.$value4_clean.'', '.$value5_clean.'', '.$value6_clean.'');';

// "INSERT INTO list_of_order(name)
VALUES(''.mysqli_real_escape_string($connect, $_POST["name"][$i]).'');
$polaczenie->query($sql);
// mysqli_query($polaczenie, $sql);
echo "order ".$i." was insterter properly";
echo $sql;
}

else{
    // echo "ERROR order ".$i." was not insterter properly";
    echo $sql;
    echo $value4_clean[$i];

}

}
}
else
{
    echo "Please, insert all data";
}

```

Listing 5 – Wstawienie do Bazy

### 3.6 Instrukcja wdrożeniowa

Najpierw musimy zainstalować na nasz serwer aplikacje XAMPP. Wymagania sprzętowe tej aplikacji możemy zobaczyć poniżej:

- Minimum 350 MB miejsca na dysku dla instalacji.
- 4 GB miejsca wolnego na dysku dla typowego systemu.
- Minimum procesora - Pentium 4, 3,2 GHz.
- 1 GB pamięci RAM.
- Karta sieciowa.

Wymagania dotyczące systemu operacyjnego:

- OS X 10.6 lub późniejsza

Instrukcja po instalacji aplikacji XAMPP:

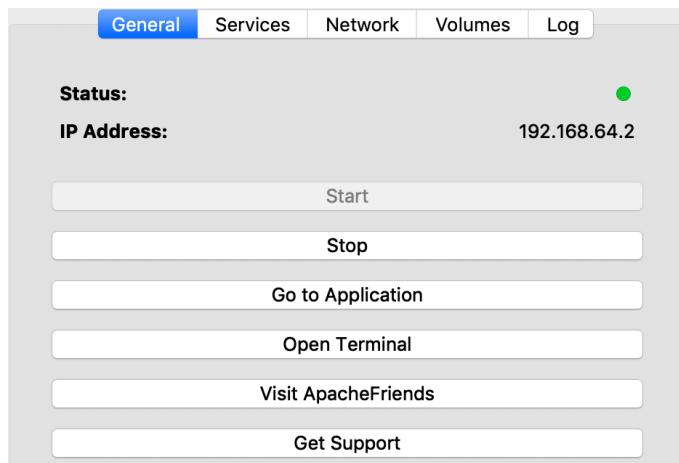
- Trzeba przyjść na stronę oficjalną XAMPP – <https://www.apachefriends.org/index.html>, na której pobieramy wersja 7.2.11, która jest najnowsza na dzień dzisiejszy.
- Po pobraniu otrzymamy plik z rozszerzeniu .dmg, który w przypadku tej wersji nazywa się xampp-osx-7.2.11-vm.dmg.
- Jak odpalimy plik musimy przeciągnąć jego do folderu naszych aplikacji, jak jest pokazane na poniższym rysunku.



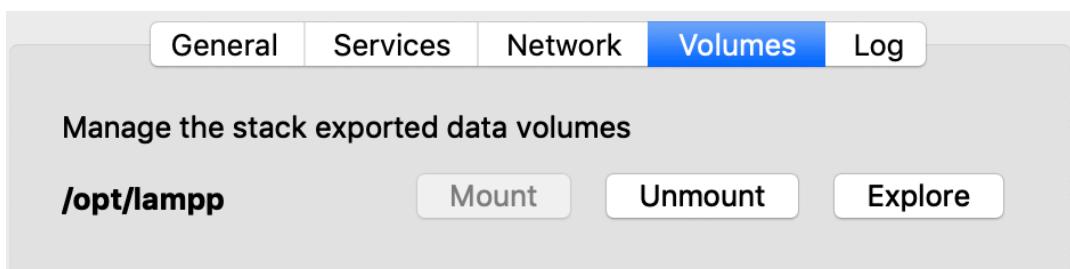
Rysunek 16 — Instalacja XAMPP

Instrukcja po konfiguracji aplikacji XAMPP:

- Jak aplikacja będzie zainstalowana musimy przenieść folder *inż* do foldera *htdocs*, w tym folderze leżą funkcje napisane w PHP.
- Następnym krokiem jest wystartowania serwera za pomocą przycisku Start w menu głównym. Menu główne możemy zobaczyć na rysunku powyższym
- W kolejnym kroku musimy przejść do zakładki która się nazywa *Volumes* i wybrać opcje *mount*, a za tym opcje *explore*, jak widać na rysunku poniższym.

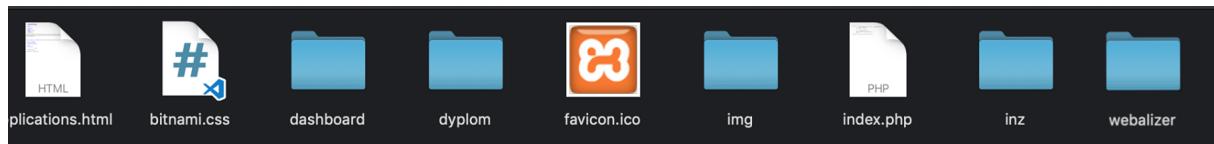


Rysunek 17 *Menu główne XAMPP*



Rysunek 18 *Menu «Volumes»*

Po wykonaniu takich kroków zможемy zobaczyć folder *htdocs* w który wstawimy nasz folder *inz*. Folder *htdocs* po zakończeniu wszystkich kroków możemy zaobserwować na rysunku poniższym.



Rysunek 19 *Folder «htdocs»*

Następnym krokiem jest zezwolenie na wejście do phpMyAdmin, które się znajduje w XAMPP. Nie możemy tego zrobić najpierw bo wejście jest zabronione.

Odblokujemy phpMyAdmin dla zaimportowania stworzonej bazy danych do serwera.

- Znowu musimy przenieść do zakładki która się nazywa *Volumes* z menu głównego i wybrać *Mount* i *Explore*.
- Po tym trzeba przejść po ścieżce *etc/extra/httpd-xampp.conf* i otworzyć plik. Na 18 linijce, w punkcie **Directory**, trzeba wstawić tekst, który można zobaczyć na rysunku poniższym.

```
<Directory "/opt/lampp/phpmyadmin">
    AllowOverride AuthConfig Limit
    Order allow,deny
    Require all granted
    Allow from all
```

Rysunek 20 *Directory*

Ostatnim etapem będzie dodanie stworzonej bazy danych do phpMyAdmin.

- Odpalamy przeglądarkę i przechodzimy do strony podanej w głównym menu aplikacji, które można zobaczyć na rysunkach poprzednich, Adresem jest – 192.168.64.2.



Rysunek 21 *Strona 192.168.64.2*

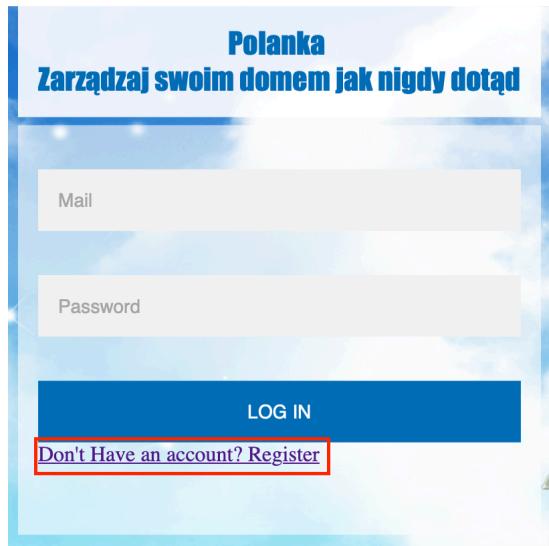
- Na podanej stronie przechodzimy do phpMyAdmin, strona pod adresem 192.168.64.2 jest pokazana na rysunku powyższym.
- Za tym mamy stworzyć nową bazę danych do której potem zimportujemy swoją . Musimy stworzyć bazę o nazwie *praca\_inzynierska* i kodowaniem *utf8\_polish\_ci*, jak na rysunku poniższym.

Rysunek 22 *Tworzenie bazy*

- Potem trzeba wybrać bazę w menu, przejść do wkładki *Import* w menu górnym, w której musimy podać linka do pliku *praca\_inzynierska*, w którym znajduje nasza stworzona baza.

## 4. Tutorial

### Rejestracja



Rysunek 23 *Główna Strona*

Znajdując się na głównej stronie wybieramy zaznaczoną opcję po czym przenosi nas na ekran rejestracji który wygląda tak

A screenshot of the Polanka registration form. It consists of several input fields arranged vertically: "House name", "Address", "Landline number", "Phone number", "Name", "Surname", and a "Date of Birth" field with dropdown menus for Year, Month, and Day. Below these are "House E-Mail" and "Admin Mail" fields, followed by "Password" and "Repeat Password" fields. There is also a checkbox for "Accept terms of usage" with a link "More info here". Underneath the checkbox is a "reCAPTCHA" field with the text "Nie jestem robotem" and the reCAPTCHA logo. At the bottom of the form is a large blue button labeled "ZAREJESTRUJ SIĘ" and a smaller link "Cofnij".

Rysunek 24 *Strona Rejestracji*

W formularzu bardzo ważne jest by wszystkie pola były wypełnione, a daty były zgodne z stanem faktycznym. Akceptując regulamin, zgadzamy się na przetwarzanie danych RODO

W razie porażki dostaniemy informację pod polem co zrobiliśmy źle.  
Natomiast po udanej rejestracji naszym oczom ukazuje się napis:

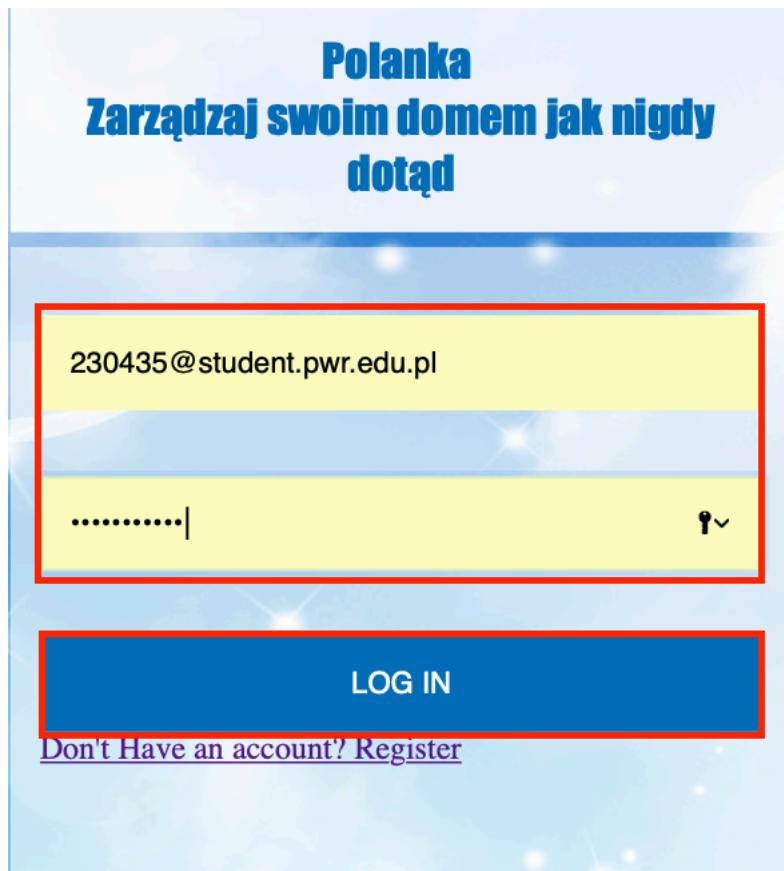
Dziękujemy za rejestrację w serwisie! Możesz już zalogować się na swoje konto!

Zaloguj się na swoje konto!

Rysunek 25 *Rejestracja z sukcesem*

Po kliknięciu Zaloguj się na swoje konto!

## Logowanie



Rysunek 26 *Logowanie*

Aby poprawnie zalogować się należy wypełnić pola danymi, które zostały podane podczas rejestracji, oraz kliknąć przycisk **LOG IN**.

W razie podania nieprawidłowych danych dostaniemy o tym komunikat i zostaniemy na stronie.

Gdy dane podamy prawidłowo przejdziemy na stronę główną, którą jest zaznaczona niżej My Orders.

The screenshot shows the Polanka application interface. At the top, there is a navigation bar with several tabs: 'New Order' (blue), 'My Orders' (green, highlighted with a red border), 'List Of Orders' (yellow), 'History Of Orders' (orange), 'Storage' (teal), 'Graphic' (red), and 'Users' (purple). Below the navigation bar, the main content area has a title 'Polanka - Zarządzaj swoim domem jak nigdy dotąd'. There is a 'Show All' button above a table. The table has columns for Room (Room, Hide Rooms), Equipment (Equipment, Hide Equipment), Status (Status, Hide Status), User (User, Hide User), Priority (Priority, Hide Priority), Description (Description, Hide Description), Price (Price, Hide Price), Effort Hours (Effort Hours, Hide Hours), and two buttons ('Edit' and 'Trash'). The table contains data rows labeled 'a', 'b', 'To do', 'none', 'Low', 'c', '100', '150', 'Edit', and 'Trash'.

Rysunek 28 *Strona po zalogowaniu*

## Dodawanie zlecenia

Ostatnią i chyba najważniejszą funkcjonalnością jest Dodawanie nowego zlecenia w domu dostępne pod przyciskiem New Order:

The screenshot shows the 'New Order' page. The navigation bar at the top is identical to the one in the previous screenshot, with the 'New Order' tab highlighted by a red border. Below the navigation bar, the main content area has a title 'Add New Order To House: pierwszy'. There is a form with fields for 'Id' (with placeholder 'house: pierwszy'), 'Room', 'Equipment', 'To Do' (dropdown with 'none'), 'Priority' (dropdown with 'Low'), 'Description', 'Price', 'Effort Hours', and a green 'Add More' button. At the bottom left are 'Submit' and 'Back' buttons.

Rysunek 29 *New Order*

Przycisk przenosi nas na kolejną stronę, na której należy tak jak w przypadku dodawania zasobów do schowka wypełnić wszystkie komórki:

The screenshot shows the 'Add New Order To House' form. The title is 'Add New Order To House: pierwszy'. The form includes fields for 'Id' (set to 'house: pierwszy'), 'Room', 'Equipment', 'To Do' (set to 'none'), 'Priority' (set to 'Low'), 'Description', 'Price', 'Effort Hours', and a green 'Add More' button. At the bottom left are 'Submit' and 'Back' buttons.

Rysunek 30 *Formularz dodawania zlecenia*

Formularz ten różni się od dodawania zasobów do schowka jedynie tym, iż nadajemy tutaj również priorytet, oraz przypisujemy zgłoszenie do konkretnego użytkownika.

## Edycja zgłoszeń

Znajdując się na tej stronie widzimy wszystkie zlecenia, które są przypisane do nas, oraz które nie są przypisane do nikogo. Mamy również możliwość edycji tabeli poprzez naciśnięcie przycisku Edit, lub usunięcia zlecenia poprzez naciśnięcie Trash.



Rysunek 31 *Przyciski*

Przy Edycji należy wypełnić pola, których zmiana nas interesuje, a następnie nacisnąć save.



Rysunek 32 *Save*

Przy usunięciu natomiast należy potwierdzić to przyciskiem Confirm.



Rysunek 33 *Confirm*

Mamy również możliwość chowania i pokazywania poszczególnych kolumn dzięki przyciskom pokazanym na obrazku poniżej

Show All							
Room	Equipment	Status	User	Priority	Description	Price	Effort Hours
Show Rooms Hide Rooms	Show Equipment Hide Equipment	Show Status Hide Status	Show User Hide User	Show Priority Hide Priority	Show Description Hide Description	Show Price Hide Price	Show Effort Hours Hide Hours

Rysunek 34 *Chowanie kolumn*

W zakładkach List of orders, różnica polega tylko na tym, że przedstawiona lista jest listą wszystkich zleceń w domu, natomiast History of Orders, przedstawia historię zgłoszeń w domu, zarówno tych wykonanych jak i usuniętych.

## Dodawanie zasobów do schowka

W zakładce Storage dysponujemy jeszcze jednym przyciskiem, który odpowiada za dodanie zasobu do domostwa.

Polanka - Zarządzaj swoim domem jak nigdy dotąd							
Add Item to Storage							
Show Rooms Hide Rooms	Show Item Hide Items	Show Quantity Hide Quantity	Show Requests Hide Requests	Show Weights Hide Weights	Show Prices Hide Prices		
Room	Item name	Quantity	Number Of Requests	Weight	Price		

Rysunek 35 *Add item to Storage*

Odbiera się to na osobnym ekranie, który wygląda następująco:

### Add New Item To Storage Of House: pierwszy

Id: 1	house: pierwszy	Room	Equipment	Quantity	Requests	Low	Price	Add More
Submit	Back							

Rysunek 36 *Ekran dodawania zasobu do schowka*

Na podanym ekranie nie jesteśmy w stanie zmienić komórek ID, oraz House, natomiast jesteśmy zobowiązani do wprowadzenia reszty informacji, możemy również dodać kolejny rząd(przycisk **Add More**), aby mieć możliwość dodania kilku zasobów za jednym razem.

W zakładce Users, do której mamy dostęp jeśli jesteśmy administratorem, jesteśmy w stanie zobaczyć wszystkie informacje na temat użytkowników, jesteśmy również w stanie dodać nowego.

## Dodawanie użytkowników



Rysunek 37 *Add user to house*

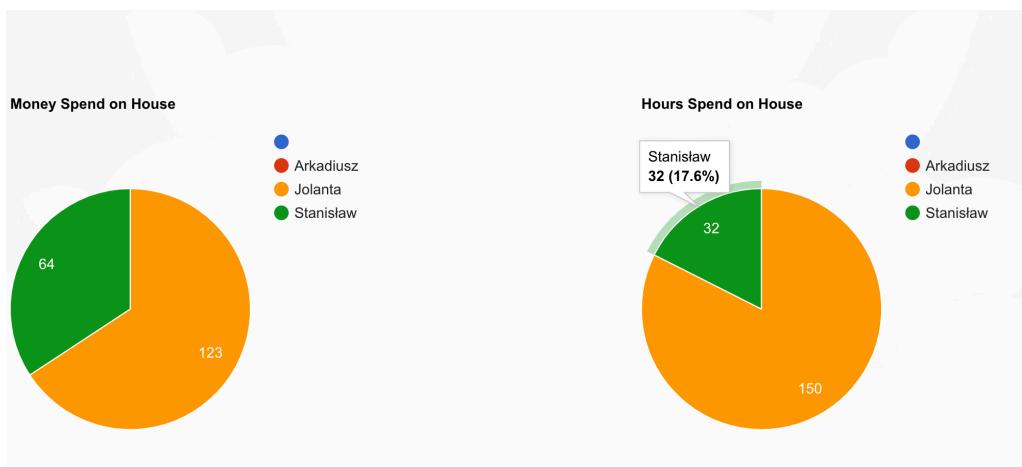
To również odbywa się na osobnym ekranie, w którym musimy również wypełnić wszystkie wiersze:

The screenshot shows a form titled "Add New User to House: pierwszy". The form fields include: "Phone number", "Name", "Surname", "Date of Birth" (with dropdown menus for Year, Month, and Day), "Account Type" (set to "Admin"), "Mail" (with placeholder "Mail"), "Password" (with placeholder "Password"), "Repeat Password" (with placeholder "Repeat Password"), and a blue "ADD USER" button at the bottom.

Rysunek 38 *Formularz dodawania użytkownika*

## Wykresy

W aplikacji pozostała jeszcze jedna funkcjonalność, którą jest graphic:

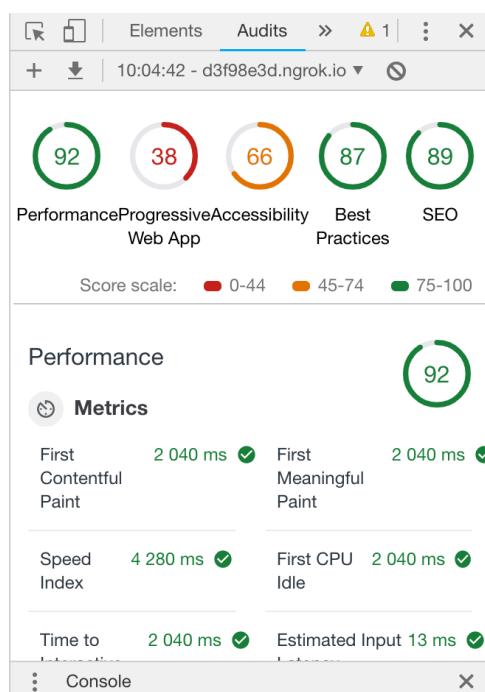


Rysunek 39 *Wykres pieniędzy i czasu wydanych przez użytkowników*

Służy on do wyświetlenia podsumowania, dotyczącego wydanych pieniędzy przez każdego domownika, oraz czasu poświęconego domowi.

## 5. Testy

Naszą aplikację postanowiliśmy również przetestować zarówno za pomocą wbudowanego w przeglądarkę programu „lighthouse”, oraz oprogramowania do testów automatycznych „[achecker.ca](#)”. Z narzędzia *lighthouse* byliśmy w stanie korzystać bez żadnych dodatkowych programów, po przetestowaniu, uzyskaliśmy następujące wyniki:



Rysunek 40 *lighthouse*

Po czym musieliśmy zainstalować narzędzie *ngrok aby korzystać z „[achecker.ca](#)”*, jako iż *achecker* nie pokazuje podsumowania, postanowiliśmy go jedynie użyć do poprawiania błędów.

## **6. Podsumowanie**

Podsumowując, podstawowym problemem okazał się język PHP, którego obaj nie znaliśmy, oraz ogromna ilość innych zajęć, przez co zrobiliśmy funkcjonalną aplikację, lecz bez funkcjonalności grafiku. Ogromną część czasu zajęło nam przygotowanie dokumentacji, przez co aplikacja nie została wykończona tak jak zakładaliśmy i nie jest ona przystosowana do urządzeń mobilnych. Co nie oznacza, że nie da się z niej korzystać, jest ona funkcjonalna, lecz GUI nie jest przystosowane do mniejszych ekranów.