

**Studia i Materiały  
Informatyki Stosowanej**



# **Studia i Materiały Informatyki Stosowanej**

Czasopismo młodych pracowników  
naukowych, doktorantów i studentów

Tom 12, Nr 1, 2020

Bydgoszcz 2020

**Studia i Materiały Informatyki Stosowanej**  
**Czasopismo młodych pracowników naukowych, doktorantów**  
**i studentów**

© Copyright 2020 by Uniwersytet Kazimierza Wielkiego

**Patronat naukowy:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. (052) 325 76 11  
e-mail: [simis@ukw.edu.pl](mailto:simis@ukw.edu.pl)

**ISSN 1689-6300**

**Projekt okładki: Łukasz Zawadzki (StudioStrzelec.pl)**  
**DTP: Dawid Ewald**

**Wydawca:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
Dyrektor:  
dr hab. inż. Izabela Rojek, prof. uczelni  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. +48 52 325 76 11  
*email: [izarojek@ukw.edu.pl](mailto:izarojek@ukw.edu.pl)*

**Kontakt:**

dr inż. Jacek Czerniak, prof. uczelni  
dr hab. inż. Marek Macko, prof. uczelni  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
*e-mail: [jczerniak@ukw.edu.pl](mailto:jczerniak@ukw.edu.pl)*  
*[mackomar@ukw.edu.pl](mailto:mackomar@ukw.edu.pl)*

**Druk (ze środków sponsora):**  
**Oficyna Wydawnicza MW**

**Nakład 250 egz.**

**Bydgoszcz 2020**

**Studies and Materials  
in  
Applied Computer  
Science**

Journal of young researchers,  
PhD students and students

Vol. 12, No.1, 2020

Bydgoszcz 2020

**Studies and Materials in Applied Computer Science**  
**Journal of young researchers, PhD students and students**

© Copyright 2020 by Kazimierz Wielki University in Bydgoszcz

**Scientific patronage:**  
Institute of Informatics  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. +48 52 325 76 11  
*e-mail: simis@ukw.edu.pl*

**ISSN 1689-6300**

**Cover designed by: Łukasz Zawadzki (StudioStrzelec.pl)**  
**DTP by: Dawid Ewald**

**Publisher:**

Institute of Informatics  
Kazimierz Wielki University  
Head:  
Izabela Rojek, PhD DSc Eng Assoc. Prof.  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. + 48 52 325 76 11  
*e-mail: izarojek@ukw.edu.pl*

**Contact:**

Jacek Czerniak, PhD. Eng. Assoc. Prof.  
Marek Macko, PhD. DSc. Eng., Assoc Prof  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
*e-mail: jczerniak@ukw.edu.pl*  
*mackomar@ukw.edu.pl*

**Printing (funded from non-profit programme):**  
**Oficyna Wydawnicza MW**

**Edition of 250 copies**

**Bydgoszcz 2020**

# Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

patronat: Polskie Towarzystwo Informatyczne



## Przewodniczący Rady Naukowej

prof. dr hab. inż. czł. rzec. PAN Janusz Aleksander Kacprzyk, IBS PAN

## Redaktorzy Naczelni

dr inż. Jacek Czerniak, UKW  
dr hab. inż. Marek Macko, prof. nadzw.

## Redaktor Zarządzający

dr inż. Łukasz Apiecionek, UKW

## Redaktor Statystyczny

dr Iwona Filipowicz, UKW

## Komitet Redakcyjny

dr inż. Mariusz Dramski, AM  
dr inż. Hubert Zarzycki, WWSIS  
dr inż. Marcin Łukasiewicz, UTP  
dr inż. Piotr Dziurzański, ZUT  
dr inż. Tomasz Kałaczyński, UTP  
dr hab. inż. Grzegorz Domek, prof. nadzw.  
dr Piotr Prokopowicz, UKW

## Redaktor Tematyczny (Informatyka)

prof. dr inż. Rafał A. Angryk, GSU

## Redaktor Tematyczny (Mechatronika)

prof. dr.h.c.mult. Peter Kopacek, VUT

## Redaktor Tematyczny (Metody numeryczne)

dr hab. Petro Filevych, LNUVB

## Redaktor Językowy (j.polski)

dr Małgorzata Kempieńska, FRM

## Redaktor Językowy (j.angielski)

Andrew Gill, Reed Elsevier, UK

## Rada Naukowa

dr hab. Stanisław	Ambroszkiewicz	Instytut Podstaw Informatyki PAN
prof. dr inż. Rafał A.	Angryk	Georgia State University, USA
dr hab. Zenon	Biniek	Wyższa Szkoła Technologii Informatycznych
prof. dr hab. inż. Ryszard	Budziński	Zachodniopomorski Uniwersytet Technologiczny
dr inż. Joanna	Chimiak-Opoka	University of Innsbruck, Austria
prof. dr hab. inż. Ryszard	Choraś	Uniwersytet Technologiczno-Przyrodniczy
dr hab. Petro	Filevych	Lviv National University of Veterinary and Biotechnologies, Ukraina
prof. dr hab. inż. Piotr	Gajewski	Wojskowa Akademia Techniczna
dr inż. Marek	Holyński	Prezes Polskiego Towarzystwa Informatycznego
prof. dr hab. inż. czł. rzec. PAN Janusz	Kacprzyk	Instytut Badań Systemowych PAN
dr hab. inż. Andrzej	Kobyliński	Szkoła Główna Handlowa
prof. dr.h.c.mult. Peter	Kopacek	Vienna University of Technology, Austria
prof. dr hab. inż. czł. koresp. PAN Józef	Korbicz	Uniwersytet Zielonogórski
prof. dr hab. inż. Jacek	Koronacki	Instytut Podstaw Informatyki PAN
prof. dr hab. inż. Marek	Kurzyński	Politechnika Wrocławska
prof. dr hab. inż. Halina	Kwaśnicka	Politechnika Wrocławska
prof. dr Miroslaw	Majewski	New York Institute of Technology, United Arab Emirates
dr inż. Dariusz	Mikolajewski	Uniwersytet Kazimierza Wielkiego
prof. dr hab. Andrzej	Marciniak	Politechnika Poznańska
dr hab. Marcin	Paprzycki	Instytut Badań Systemowych PAN
prof. dr hab. inż. czł. PAN Witold	Pedrycz	University of Alberta, Canada
prof. dr hab. inż. Andrzej	Piegat	Zachodniopomorski Uniwersytet Technologiczny
prof. dr hab. inż. Andrzej	Polański	Politechnika Śląska
prof. dr hab. inż. Orest	Popov	Zachodniopomorski Uniwersytet Technologiczny
prof. dr inż. George	Przybył Einstein	College of Medicine, USAT Montserrat
dr hab. inż. prof. nadzw. Izabela	Rojek	Uniwersytet Kazimierza Wielkiego
prof. dr hab. inż. Danuta	Rutkowska	Politechnika Częstochowska
prof. dr hab. inż. czł. koresp. PAN Leszek	Rutkowski	Politechnika Częstochowska
prof. dr hab. inż. Milan	Sága	Žilinská Univerzita, Słowacja
prof. dr hab. inż. czł. rzec. PAN Roman	Słowiński	Instytut Badań Systemowych PAN, Politechnika Poznańska
prof. dr hab. inż. Włodzimierz	Sosnowski	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż. Andrzej	Stateczny	Akademia Morska w Szczecinie
dr hab. inż. Jan	Studziński	Instytut Badań Systemowych PAN
prof. dr hab. Tomasz	Szapiro	Szkoła Główna Handlowa
dr hab. Janusz	Szczepański	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż. czł. rzec. PAN Ryszard	Tadeusiewicz	Akademia Górniczo-Hutnicza
prof. zw. dr hab. inż. czł. rzec. PAN Jan	Węglarz	Instytut Chemii Bioorganicznej PAN, Politechnika Poznańska
prof. dr hab. inż. Sławomir	Wierchoń	Instytut Podstaw Informatyki PAN
prof. dr hab. inż. Antoni	Wiliński	Zachodniopomorski Uniwersytet Technologiczny
dr hab. inż. Andrzej	Wiśniewski	Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach
dr hab. inż. Ryszard	Wojtyna	Uniwersytet Technologiczno-Przyrodniczy
prof. dr hab. Sławomir	Zadrożny	Instytut Badań Systemowych PAN
prof. dr. inż. Milan	Žmindák	Žilinská Univerzita, Słowacja
prof. dr hab. Zenon	Zwierzewicz	Akademia Morska w Szczecinie

## **SPIS TREŚCI**

<b>Od Redakcji .....</b>	<b>4</b>
<b>Aplikacja na platformę Android z systemem ekspertowym jako rozwiązanie problemu właściwego żywienia koni</b>	
Adrianna Piszcz, Dariusz Mikołajewski .....	5
<b>Zespół napędowy jako substytut pomostu</b>	
Krzysztof Galas, Józef Kubik .....	10
<b>Mobilne zarządzanie wynajmowaniem mieszkań. Aplikacja na platformę Android z wykorzystaniem serwera Node.js oraz bazy danych MS SQL</b>	
Belco Sangho, Zbyszko Królikowski .....	15
<b>Bydgoski egzoszkielec na rękę – koncepcja i wyniki wstępne</b>	
Jakub Kopowski, Izabela Rojek, Dariusz Mikołajewski, Marek Macko .....	22





## OD REDAKCJI

Szanowni Czytelnicy,

*Na progu Nowego 2020 roku przedstawiamy Czytelnikom kolejny numer naszego czasopisma poświęcony szerokiemu przekrojowi zagadnień z obszaru zastosowań informatyki i mechatroniki: od aplikacji na platformę Android wspomagających żywienie koni czy wynajem mieszkań aż po zaawansowane naukowo i technicznie rozwiązania platformy pływającej oraz egzoszkieletu na rękę. Pozwala to spojrzeć na opisywaną grupę zagadnień z punktu widzenia ich rozwoju oraz integracji w ramach przyszłościowych rozwiązań Internetu Rzeczy (ang. Internet of Things - IoT) oraz inteligentnego otoczenia (ang. Ambient Intelligence - AmI). Pozwoli to rozwiązywać problemy użytkowników kompleksowo, również dzięki łatwiejszemu udostępnieniu know-how, algorytmów czy gotowych narzędzi w ramach typowej dla IoT architektury sprzęt-się-chmura, lecz przy zapewnieniu typowej dla IoT zwiększonej efektywności przetwarzania danych oraz oddziaływania otoczenia na użytkownika. Potencjał rozwojowy tej grupy rozwiązań jest ogromny, a jej zastosowania mogą zarówno stanowić sukces naukowo-badawczy, jak i przyczynić się do poprawy kondycji gospodarczej, jak również jakości życia społeczeństwa.*

Redaktorzy Naczelni SiMIS,  
dr inż. Jacek Czerniak,  
dr hab. inż. Marek Macko, prof. nadzw.

# Aplikacja na platformę Android z systemem ekspertowym, jako rozwiązanie problemu właściwego żywienia koni

Adrianna Piszcz, Dariusz Mikołajewski

*Instytut Informatyki, Uniwersytet Kazimierza Wielkiego w Bydgoszczy*

**Streszczenie:** *Celem badania jest stworzenie aplikacji wspomagającej prawidłowe żywienie koni na platformę Android. Aplikacja realizowana jest w środowisku Xamarin w programie Visual Studio z wykorzystaniem języka CSharp. Dane niezbędne do działania aplikacji magazynowane są w plikach XML. Model systemu ekspertowego został przygotowany w oparciu o drzewa decyzyjne, natomiast projekt aplikacji opisuje diagram przypadków użycia oraz diagram klas. System ekspertowy, który został zaimplementowany w aplikacji, wykorzystuje do wygenerowania diety bazę wiedzy pozyskaną od ekspertów oraz moduł wnioskujący. Aplikacja może służyć zarówno celom użytkowym, jak i edukacyjnym. Będzie użyteczna dla osób ze specjalistycznym doświadczeniem, jak i bez niego.*

**Słowa kluczowe:** *Xamarin, aplikacja Android, XML, żywienie koni, system ekspertowy*

**Abstract:** *The aim of the study is to create application supporting the proper nutrition horses on the Android platform. The application is created in the environment Xamarin in Visual Studio IDE using CSharp language. The data needed for the application are stored in XML files. Expert system model was developed based on decision trees and the design of the application describes the use case diagram and class diagram. Expert system used in the application generates a diet using a knowledge base gained from experts and requesting module. The application can be used both practical purposes, as well as educational. It will be useful for people with specialized experience, and without it.*

**Keywords:** *Xamarin, Android application, XML, horse nutrition, expert system*

## 1. Wprowadzenie

Jeździectwo nieustannie się rozwija, stajnie stają się coraz bardziej rozbudowane i innowacyjne. Wzrasta świadomość i wiedza właścicieli i hodowców koni. Jednak nie sposób wymagać od wszystkich takiej samej wiedzy na temat żywienia, a jest to jeden z głównych aspektów zapewniających dobrobyt zwierzęcia. Często w stajniach wszystkie konie skarmiane są tymi samymi dawkami pokarmowymi, niezależnie od ilości ruchu, rasy czy innych czynników determinujących dobór optymalnej diety.

System ekspercki opiera się na wiedzy, która dostarcza rozwiązania problemów dotyczących danego zagadnienia poprzez wnioskowanie na podstawie bazy wiedzy. Bazy wiedzy tworzone są na podstawie umiejętności ekspertów, ich wiedzy merytorycznej, toku rozumowania oraz decyzyjności.

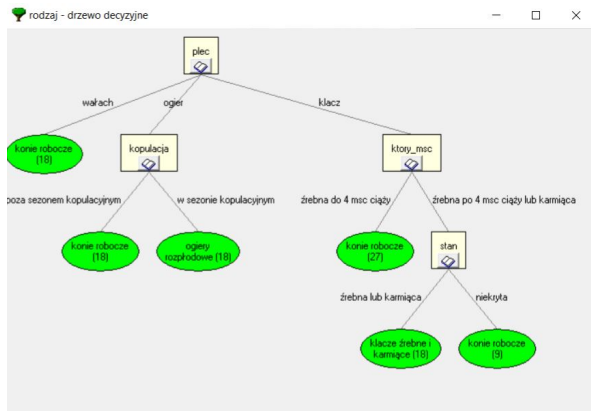
Aplikacja ma rozwiązywać problemy żywienia

koni w stajniach. System ekspertowy służyć będzie dobraniu najlepszej diety, poprzez zadawanie użytkownikowi szeregu pytań dotyczących konkretnego konia. Informacje te będą katalogowane w osobnej zakładce, aby użytkownik miał łatwy dostęp do każdej diety.

Celem jest zbudowanie intuicyjnego systemu, który dotrze do szerokiego grona odbiorców, niezależnie od stopnia wiedzy. System ten ma wspierać dobre praktyki żywieniowe koni. W łatwy sposób pozwoli dobrać oraz zapisać dietę dostosowaną dla konkretnego konia.

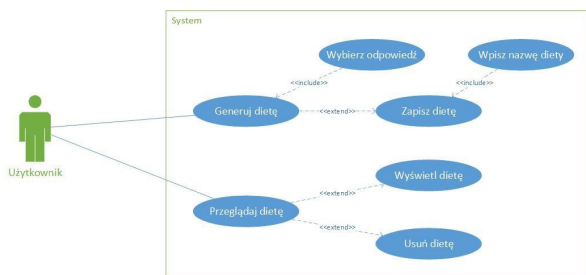
## 2. Metody

**Drzewa decyzyjne** Drzewo decyzyjne (Rysunek 1) składa się z korzenia, krawędzi, węzłów oraz liści. Liście to węzły, z których nie odchodzą już żadne krawędzie. Krawędzie są połączeniami istniejącymi w modelu. Drzewa decyzyjne mają strukturę hie-



Rysunek 1: Drzewo decyzyjne grupy głównej - DETREEX

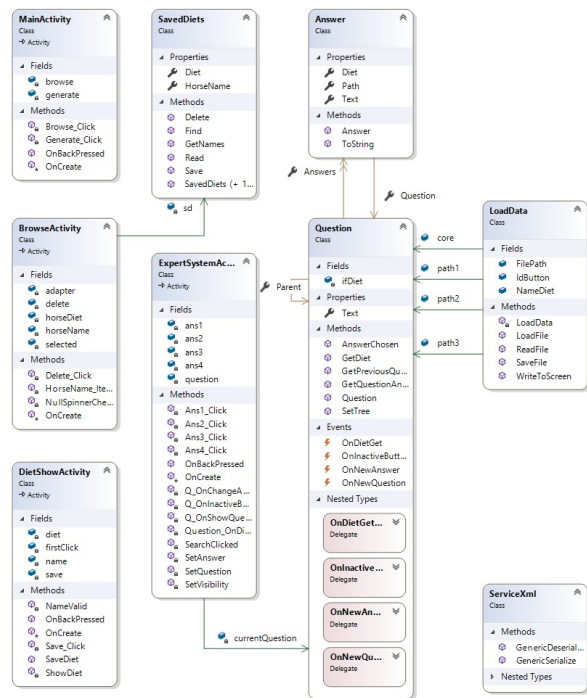
rarchiczną. W kolejnych krokach dzielony jest zbiór obiektów, poprzez odpowiadanie na pytania o wartości wybranych atrybutów. Ostateczna decyzja zależy od odpowiedzi na wszystkie pytania. Projekt drzew decyzyjnych został wykonany za pomocą systemu DETREEX.



Rysunek 2: Diagram przypadków użycia

**Diagram przypadków użycia** Diagram przypadków użycia pozwala spojrzeć na system od strony użytkownika. Charakterystyczną grafiką figurki człowieka oznaczany jest aktor, czyli pewien określony rodzaj użytkownika. W diagramie może występować jeden lub kilku aktorów. Aktor określa użytkownika o sprecyzowanych uprawnieniach i o możliwych do podjęcia przez niego akcji. Natomiast elipsa reprezentuje przypadek użycia, to znaczy jakąś określoną akcję. W tym diagramie zawsze aktor inicjuje akcje. W realizowanej aplikacji występuje jeden typ użytkownika. Może on generować lub przeglądać dietę. Generowanie diety jest bezpośrednio związane z wybieraniem odpowiedzi. Dietę wygenerowaną można zapisać, a to z kolei wiąże się z poprawnym wpisaniem nazwy diety. Przeglądanie diet rozszerzone jest o ich usuwanie oraz o wyświetlenie składu wybranej diety. Wykonany w progra-

mie MS Visio diagram przypadków użycia (Rysunek 2) obrazuje możliwe akcje wykonywane przez użytkownika systemu Paszomat.



Rysunek 3: Diagram klas

**Diagram klas** Klasa przedstawiona jest w postaci encji zawierającej atrybuty i funkcje, czyli działania jakie może wykonać obiekt tej klasy. Linie łączące poszczególne encje wskazują na relacje występujące pomiędzy klasami. Klasy mogą dziedziczyć z innych klas lub mogą być powiązane poprzez utworzenie obiektu jednej klasy w innej klasie. Na rysunku (Rysunek 3) przedstawiono diagram klas systemu Paszomat. Po prawej stronie widoczne są dwie klasy serwisowe: klasa ServiceXML, która nie posiada żadnych powiązań, ponieważ jej metody są statyczne oraz klasa LoadData, która inicjalizuje obiekty klasy Question. W trzeciej kolumnie widoczne są klasy Question i Answer, które mają pomiędzy sobą połączenie obustronne. Klasa Question implementuje kolekcje obiektów Answer, a klasa Answer obiekt Question. Dodatkowo można zauważyć, że klasa Question posiada połączenie z samą sobą. Oznacza to, że wewnątrz tej klasy tworzona jest jej instancja. Klasy aktywności nie posiadają powiązań z resztą systemu, ponieważ przepływ informacji pomiędzy nimi odbywa się za pośrednictwem eventów, za wyjątkiem klasy ExpertSystemActivity, która posiada powiązanie z klasą

Question, aby pobierać informacje o aktualnym pytaniu oraz klasą BrowseActivity, która posiada powiązanie z klasą SavedDiets (implementuje instancję tej klasy), po to aby móc zarządzać zapisanymi danymi.

**Baza danych XML** Extensible Markup Language, w skrócie XML to język znacznikowy należący do tej samej rodziny języków co HTML. Odróżnia go fakt, że znaczniki nie są definiowane odgórnie poprzez język, tylko jest to rolą użytkownika. Zaletą przechowywania danych w plikach xml w porównaniu do tradycyjnej bazy danych jest prostota projektowania, możliwość szybkiego przeszukiwania i pozyskiwania danych. Baza danych XML została wykorzystana w projekcie do przechowywania bazy wiedzy niezbędnej dla funkcjonowania systemu ekspertowego. Jej struktura jest zgodna ze strukturą wymodelowaną na drzewach decyzyjnych. Projekt wykorzystuje znaczniki o nazwach identycznych jak nazwy klas oraz pól tworzących strukturę projektu.

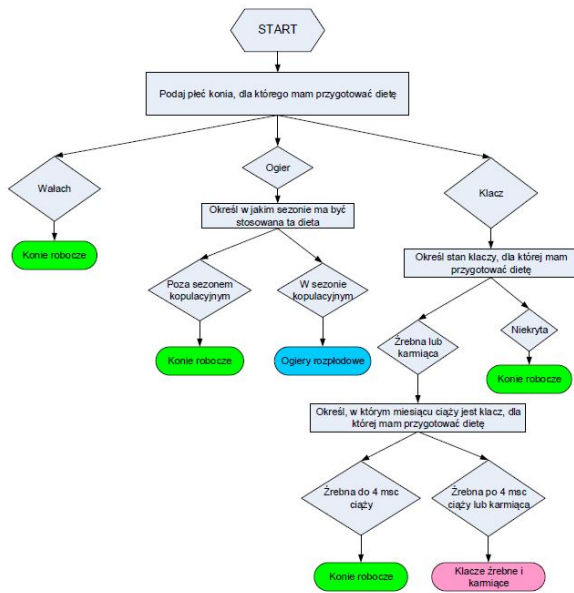
**Technologia Xamarin** Xamarin pozwala na natywne projektowanie aplikacji mobilnych, co sprawdza się przy realizacji projektów nastawionych na multiplatformowość. Kod Xamarin.Forms pisany jest w XAML-u. Został przygotowany zestaw kontrolki, które mapowane są na kontrolki używane w danym systemie.



Rysunek 5: Ekran menu głównego

### 3. Rezultaty

**System ekspertowy** Projekt systemu doradczego, wspomagającego dobór optymalnej diety dla konia, składa się z części głównej, którą jest wybór grupy, do której należy koń, a także z trzech części podrzędnych stanowiących kontynuację dla wyboru z grupy głównej. Zatem powstały cztery odrębne modele, z których jeden, jest modelem głównym (Rysunek 4) i determinuje przejście do jednego z trzech pozostałych.



Rysunek 4: Diagram grupy głównej

**Bazy wiedzy** Na podstawie opracowanych diagramów utworzono odpowiednie pliki formatu XML, wykorzystane w późniejszym etapie do implementacji logiki systemu. Każdemu modelowi odpowiada jeden plik XML, tak więc analogicznie istnieje plik grupy głównej oraz trzy pliki grup szczegółowych. Każdy z plików posiada jednakową strukturę odpowiadającą strukturze klasowej projektu. Elementem głównym plików jest etykieta *Question*, która obejmuje wszystkie pozostałe elementy danego pliku i jest korzeniem dla całego budowanego systemu. Posiada dwóch bezpośrednich potomków: *Text* oraz *Answers*. Etykieta *Text* przechowuje treść pytania, natomiast *Answers* jest węzłem dla odpowiedzi na to pytanie. Każda odpowiedź ma treść i może prowadzić do jednego z trzech elementów: do kolejnego pytania, do odpowiedzi jaką jest dieta lub do ścieżki wskazującej na jeden z trzech plików szczegółowych.



Rysunek 6: Ekran wygenerowanej diety



Rysunek 7: Ekran katalogu zapisanych diet

**Interfejs użytkownika** Interfejs użytkownika został opracowany z bardzo dużą starannością. Aplikacja z założenia ma być prosta i przyjazna mniej technologicznym użytkownikom. Składa się ona z czterech różnych ekranów – menu głównego (Rysunek 5), systemu ekspertowego, wygenerowanej diety oraz katalogu zapisanych diet. Pierwszym ekranem widocznym po uruchomieniu aplikacji jest menu główne. Dla wszystkich ekranów aplikacji został wykorzystany układ *RelativeLayout*, który charakteryzują zależności od innych elementów tego układu. Wybranie opcji „Generuj” przez użytkownika powoduje przekierowanie do najważniejszej części aplikacji – szeregu pytań i odpowiedzi prowadzących do jak najlepiej dobranej diety. Układ tego ekranu stanowi *TextView* wyświetlający treść aktualnego pytania oraz cztery kontrolki *Button* zawierające możliwe do wyboru odpowiedzi. Wybranie odpowiedzi powoduje przeładowanie widoku i wyświetlenie na tej samej formacie kolejnego pytania lub gdy było to ostatnie pytanie przeniesienie do trzeciego z kolei widoku aplikacji (Rysunek 6). Widok ten zawiera grafikę w tle, *TextView* z wygenerowaną dietą oraz *TextBox* wraz z przyciskiem służący do zapisania diety. Na tym etapie użytkownik ma możliwość cofania się do poprzednich pytań i zmiany wybranej odpowiedzi. Aplikacja podczas zapisu sprawdza czy wpisana nazwa figuruje już w bazie, jeśli tak użytkownik jest informowany o tym fakcie poprzez stosowny komunikat i ma możliwość wprowadzanie nowej, unikalnej na-

zwy. Istnieje dodatkowe zabezpieczenie, które zapobiega omyłkowemu cofnięciu użytkownika do menu głównego w przypadku, gdy została wygenerowana dieta. Jednokrotne naciśnięcie przycisku wstecz powoduje wyświetlenie alertu o niezapisanej diecie. Użytkownik ma możliwość ponownego naciśnięcia przycisku wstecz i przejścia do ekranu głównego lub zapisu diety. Ostatnim widokiem aplikacji jest widok przeglądania zapisanych diet (Rysunek 7). Wejście do tego ekranu odbywa się za pośrednictwem menu głównego po wybraniu opcji Przeglądaj i pod warunkiem, że dostępne są zapisane pozycje. Ekran zawiera *Spinner* z nazwami zapisanych diet, *ImageButton* przedstawiający kosz na śmieci oraz *TextView* ze składem diety aktualnie wybranego konia. Tło stanowi grafika konia na tle stajni.

**Testy** Pierwszym rodzajem testów, którym poddawana była aplikacja były testy manualne. Każda wdrażana funkcjonalność została ręcznie sprawdzona pod kątem oczekiwanych lub spodziewanych działań. Wszelkie odstępstwa były analizowane, a po znalezieniu źródła problemu, naprawiane. Po każdym usunięciu błędu testy były powtarzane. Kiedy proces tworzenia aplikacji został zakończony, aplikacja przeszła testy akceptacyjne. Grupa chętnych, potencjalnych użytkowników sprawdzała własnoręcznie możliwości i ograniczenia aplikacji. Zgłoszone uwagi zostały przełożone na usprawnienia aplikacji, tak aby system odpowiadał jak najlepiej na potrzeby osób, które będą go użytkowały.

## 4. Wnioski

Aplikacja wykorzystuje system ekspertowy do wygenerowania jak najlepiej dopasowanej diety dla określonego konia. Model systemu ekspertowego został zaprojektowany przy wsparciu osób posiadających doświadczenie w dziedzinie hodowli koni, zostały sformułowane pytania oraz zestawy odpowiedzi do każdego z nich, tak aby system mógł pozyskać dane niezbędne do wygenerowania diety.

System został zaimplementowany w języku CSharp. Bazy wiedzy oraz zawarte w nich reguły zostały przełożone na język obiektowy, został napisany system wnioskujący, którego zadaniem jest wyprowadzenie danych końcowych na ekran użytkownika.

Wybór konkretnej odpowiedzi determinuje ścieżkę do dalszego pytania, w ten sposób ilość zadawanych pytań została ograniczona do minimum. Sugerowane składy dawek pokarmowych z założenia są dawkami dziennymi, zatem można je podzielić na kilka posiłków.

Model jest otwarty na rozszerzenia o kolejne reguły i składy dawek. Zaimplementowanie w systemie zapamiętywania wprowadzonych przez użytkownika odpowiedzi pozwoliłoby stworzyć intuicyjny moduł zmian. Zmian w diecie można by było dokonywać zarówno poprzez ten moduł, edytując poszczególne parametry, np. ilość treningów w tygodniu lub też manualnie zmieniając dawkę z 3 kg na 2 kg. Możliwość zdalnej edycji diety swojego konia zachęcałaby ludzi zapracowanych, niemogących stawić się osobiście.

## Literatura

- [1] Ł. Bujak, *Drzewa decyzyjne*. Toruń: Uniwersytet Mikołaja Kopernika, 2008.
- [2] L. Kiełtyka, “Wykorzystanie systemów eksperckich w zarządzaniu wiedzą,” *Zeszyty Naukowe. Organizacja i Zarządzanie / Politechnika Łódzka*, vol. z. 53, 2013.
- [3] L. Madeyski, “Xml w bazach danych,” *Bazy Danych—Prace Naukowe Wydziałowego Zakładu Informatyki Politechniki Wrocławskiej*, (4), pp. 81–89, 2003.
- [4] K. Michalik, “Pc-shell/sphinx jako narzędzie tworzenia systemów ekspertowych,” *Prace Naukowe/Uniwersytet Ekonomiczny w Katowicach*, pp. 35–42, 2010.
- [5] D. Modlińska, *Konie i źrebięta : wszystko co o koniu powinien wiedzieć początkujący jeździec i hodowca*. Warszawa: Wydawnictwo Tenten, 1993.
- [6] J. Mulawka, *Systemy ekspertowe*. Warszawa: WNT, 1996.
- [7] A. Niederliński, *Regulowo-modelowe systemy ekspertowe rmse*. Gliwice: Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2006.
- [8] I. Rojek, J. Dorożyński, and D. A. Ośka, “Zarządzanie wiedzą przy użyciu systemu ekspertowego,” *Studies & Proceedings of Polish Association for Knowledge Management*, vol. 87, pp. 40–49, 2018.
- [9] W. Romowicz., *XML. Ćwiczenia praktyczne. Książka*. Gliwice: Helion, 2002. [Online]. Available: <https://helion.pl/ksiazki/xml-cwiczenia-praktyczne-wojciech-romowicz,cwxml.htm#format/d>
- [10] E. Sasimowski and M. Budzyński, *Żywnienie koni*. Warszawa: Państwowe Wydawnictwo Rolnicze i Lesne, 1981.
- [11] J. Schmuller and K. Masłowski, *UML : Ujednolicony Język Modelowania - wyrażanie związków między klasami w projektowaniu obiektowym*. Gliwice: Helion, 2003.
- [12] G. Taskos, *Xamarin. Tworzenie aplikacji cross-platform. Receptury*. Gliwice: Helion, 2017.
- [13] K. Waćkowski, J. Krawiec, J. Bereda, E. Chmielewska, and M. Malińska, *Informatyka: terminologia znormalizowana i wykaz norm*. Warszawa: Polski Komitet Normalizacyjny, 2006.
- [14] T. Wiczorek, S. Golak, R. Przyłucki, and M. Pilarczyk, “Innowacyjny system ekspertowy do oceny kopalń węgla kamiennego,” *Logistyka*, no. 5, CD 1, pp. 645–650, 2015.
- [15] R. Wieleba, “Inżynieria wiedzy w systemach ekspertowych,” *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki*, 2011.

# Zespół napędowy jako substytut pomostu

Krzysztof Galas, Józef Kubik

Wydział Mechatroniki, Uniwersytet Kazimierza Wielkiego w Bydgoszczy

**Streszczenie:** *Celem badania jest zaprezentowanie alternatywnego rozwiązania dla niszczących pomostów nad zbiornikami wodnymi. Projekt bazuje na dostępnym rozwiązaniu jakim jest platforma pływająca. Dobrane do tego został silnik krokowy, sterownik PLC oraz panel operatorski. Do symulacji tych połączeń wykorzystano środowisko WinProLadder oraz EasyBuilderPro. Przedstawione rozwiązanie obejmuje także projekt metalowej obudowy oraz obudowy w celu przechowywania na czas braku użytkowania. Do tego zostało wykorzystany program Autodesk Fusion 360. Rozwiązanie może pomóc poprawić komfort pobytu nad akwenami wodnymi, dzięki łatwiejszej konserwacji niż klasyczne rozwiązania.*

**Słowa kluczowe:** *PLC, język drabinkowy, silnik krokowy, sterowanie, pomost*

**Abstract:** *The aim of the study is to present an alternative solution to decaying bridges over water tanks. The project is based on the available floating platform solution. This has been matched to the stepper motor, PLC and operator panel. WinProLadder and EasyBuilderPro were used to simulate these connections. The illustrated embodiment also includes the design of the metal casing and the housing to hold for the duration of the absence of use. For this was used Autodesk Fusion 360. The solution can help improve the comfort of stay on waterwater, thanks to easier maintenance than classic solutions.*

**Keywords:** *PLC, ladder language, stepper motor, control, bridge*

## 1. Wprowadzenie

Rekreacja wodna w dzisiejszych czasach stała się bardzo popularna. Kiedy rozpoczyna się sezon letni, ludzie bardzo często wybierają, jako miejsce swojego urlopu jeziora czy też inne miejsca ulokowane blisko wody. Jedni preferują poranne łowienie ryb, inni sporty wodne, takie jak windsurfing czy kitesurfing. Wszystkich tych ludzi łączy jedna potrzeba - dostęp do wody.

Zaproponowano zastosowanie modułowego pomostu pływającego typu WaterFun, który charakteryzuje się dużą trwałością i jednocześnie lekkością. Ponadto pomost tego typu nie wymaga konstrukcji nośnej, a jedynie podpór w postaci słupków, które mają na celu zakotwiczenie pomostu, ale także zapobieganie chybotaniu lub nawet wywróceniu.

Projekt poszczególnych elementów konstrukcji zostanie wykonany w środowisku CAD, natomiast sterujący silnikiem program przy użyciu oprogramowania WinProLadder oraz EasyBuilderPro.

Istotą projektu jest zwiększenie komfortu osób korzystających ze standardowych pomostów. Podczas wakacji można zaobserwować zmartwionych rodziców, którzy obserwują swoje pociechy biegają-

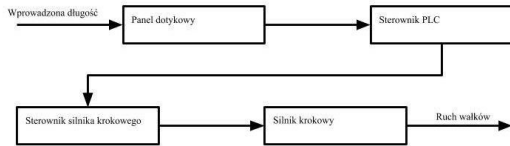
ce po wiekowych konstrukcjach, dorosłych, którzy z ostrożnością wykonują każdy krok.

To rozwiązanie świetnie sprawdzi się w przypadku bardzo ograniczonej przestrzeni, gdzie trzeba było wybierać pomiędzy kąpieliskiem, a postawieniem pomostu. Za pomocą odpowiedniego mechanizmu, pomost będzie wsuwany do stalowej obudowy, tak więc istnieje możliwość regulowania jego wysunięcia.

## 2. Metody

**Schemat układu sterowania** Na Rysunku 1 przedstawiono model działania sterowania wysuwaniem platformy pływającej. Opiera się ono na wykonywaniu ruchu obrotowego silnika krokowego. Jako narzędzie do wprowadzania danych wykorzystany został panel dotykowy. Przetwarza on dane wprowadzone przez użytkownika, które są analizowane poprzez sterownik PLC. Otrzymane dane następnie są przekazywane do sterownika silnika krokowego. Odpowiada on za wskazanie i kontrolę obrotów silnika krokowego. Następnie przekazywane impulsy elektryczne wprowadzają w ruch napęd całego systemu [2].

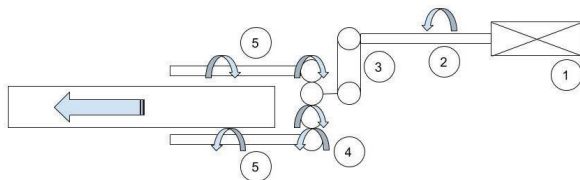




Rysunek 1: Schematyczne przedstawienie układu sterowania

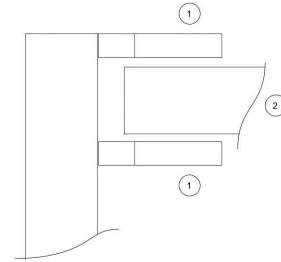
### Schemat dynamiczny układu napędowego

Przeniesienie napędu w taki sposób by powodował jednostajne i równomierne wysuwanie platformy zostało przedstawione na Rysunku 2. Przedstawiono graficznie kierunki oczekiwanego ruchu poszczególnych elementów. Ruch platformy powodują wałki wykorzystując zjawisko tarcia pomiędzy stykającymi się powierzchniami platformy i wałków. Maksymalne wysunięcie pomostu jest powiązane ze średnicą wałków. Na jej podstawie i wiedzy o ilości kroków silnika, obliczana jest odpowiednia liczba obrotów do wykonania wysunięcia. Aby zapobiec wysunięciu pomostu w warunkach niesprzyjających m.in. nadmiernego podniesienia poziomu wody, czujniki uniemożliwią wykonanie akcji oraz pojawi się stosowny komunikat na wyświetlaczu. Znaczne obniżenie poziomu wody uniemożliwi korzystanie z pomostu.

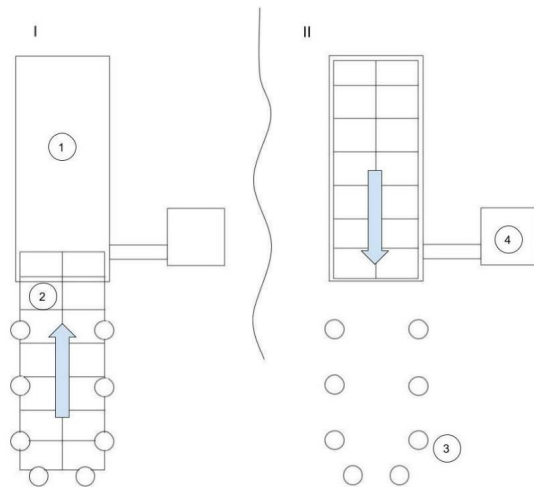


Rysunek 2: Szkic z naniesionymi elementami wewnętrznymi: silnik (1), wałek (2), łańcuch (3), koło zębate (4), wałek pokryty gumą (5).

**Koncepcja konstrukcji** Konstrukcja opiera się na ruchomym (chowanim) podeście oraz statycznej, nieruchomej bazie. Aby zapobiec nadmiernym ruchom części unoszącej się na wodzie zaproponowano zestaw słupków (Rysunek 3). Ograniczają one poziome jak i pionowe przesunięcia platformy. Rysunek 4 przedstawia ich rozmieszczenie jak i domyślny układ całej konstrukcji widziany z góry. Wysokość słupa będzie statycznie dostosowana do poziomu lustra wody, tak aby platforma wpasowała się między gumowymi uchwytami. Sam słup jest jednostką samodzielną nie łączącą się z całą konstrukcją.



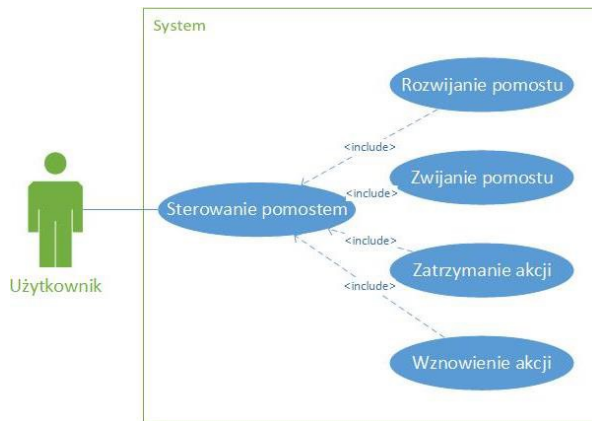
Rysunek 3: Szkic słupków podtrzymujących: gumowy uchwyt (1), pomost (2).



Rysunek 4: Szkic wysuniętego i schowanego pomostu: kieszeń (1), pomost (2), pał (3), skrzynka sterownicza (4).

**UML - Unified Modeling Language** Notacja, umożliwiająca wizualne przedstawienie swojego toku rozumowania innym osobom. Dzięki zastosowaniu, składających się na ten język modelowania, diagramów, projektowanie systemu przebiega w sposób zorganizowany i uwzględnione zostają różne punkty odniesienia. Przedstawiono dzięki niemu schemat przypadków użycia (Rysunek 5) pokazuje w uproszczeniu odpowiednie zachowanie systemu w przypadku ingerencji użytkownika. Cztery podstawowe funkcje zawierają się na całość sterowania pomostem, co jest realizowane przez program. Kolejne kroki działania systemu od momentu interakcji człowieka przedstawione są na diagramie (Rysunek 6) [?][?].

**PLC** Sterowniki PLC programowane są przeważnie w środowiskach dostarczanych przez producenta danego modelu za pomocą jednego z kilku dostępnych języków programowania. Jednym z takich języków jest ladder diagram (LAD) [7].



Rysunek 5: Diagram przypadków użycia

Można za jego pomocą zaprogramować wybrany wyświetlacz LCD, w tym także wyświetlacz dotykowy, co minimalizuje ilość fizycznych elementów układu. Brak wykorzystania przełączników, przycisków czy diod zmniejsza ryzyko uszkodzenia pojedynczych elementów układu, minimalizując tym samym konieczność częstego kontrolowania sprawności urządzenia. Wyświetlacz można łatwo dostosować do potrzeb zarówno użytkowników urządzenia, jak i środowiska w jakim ma pracować.

Do symulacji oraz programowaniem panelu wykorzystano WinProLadder oraz EasyBuilderPro. Programy te są środowiskiem programistycznym, w którym została zaimplementowana cała logika systemu.

### 3. Rezultaty

**Silnik krokowy** Jako narzędzie do wysuwania całej platformy wybrano silnik krokowy NEMA 34: M13430604.1), głównie ze względu na jego wysoki moment trzymający, który wynosi 12,4 [Nm]. Jest to atut z tego względu, iż nie ma potrzeby wprowadzenia ręcznego hamulca, który musiałby powstrzymać pomost przed odpłynięciem [1][6][10][14].

**Sterownik silnika krokowego** Wybrano sterownik silnika krokowego SSK-B16 ( Rys. 4.2), który jest kompatybilny z zastosowanym silnikiem krokowym. Dużą zaletą tego sterownika jest możliwość jego konfiguracji przez komputer osobisty, co także wspomaga diagnostykę oraz serwisowanie, w sytuacji wystąpienia awarii urządzenia. Posiada także zróżnicowaną gamę napięć zasilających, co daje możliwość wprowadzenia dowolnego źródła zasilania.



Rysunek 6: Diagram przebiegu programu

**Sterownik PLC** Zastosowano sterownik marki Fatek FBs-32MAT2 z tego powodu, iż posiada on możliwość rozbudowania, dzięki różnym modułom oferowanym przez producenta. Nie posiada on bardzo dużej ilości wejść w podstawowym zestawie, jednak jest ona wystarczająca na potrzeby projektu.

**Zasilanie** W tym przypadku można zastosować zasilanie fotowoltaiczne, gdyż konstrukcja domyślnie ma się znajdować na otwartej przestrzeni. Istnieją na rynku gotowe rozwiązania składające się z panelu słonecznego, akumulatora oraz regulatora.

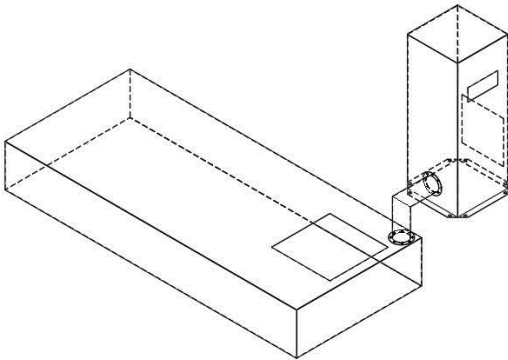
**Panel operatorski** Wybrano sterownik marki Weintec MT8101iE, posiada on szeroki zakres temperatury pracy (od 0 do 50 stopni Celsjusza), pulpit dotykowy (dla ułatwienia i usprawnienia wprowadzania danych oraz komunikacji z użytkownikiem). Wyposażony jest także w zegar czasu rzeczywistego (RTC), dzięki czemu może obliczać czas w jakim zostaną wykonane poszczególne akcje.[8] Dostęp do panelu jest ograniczony hasłem dostępu na ekranie głównym.



Rysunek 7: Widok panelu kontrolnego

**Wygląd interfejsu** Rysunek 7 przedstawia propozycję ekranu głównego panelu sterowania. Jego elementy graficzne są w pełni dostosowywać do potrzeb. Strzałki widniejące nad przełącznikami służą jako etykiety opisujące akcję, która zostanie wykonana po ich przełączeniu. Lewy odpowiada za wysuwanie pomostu i reset licznika odpowiadającego akcji wsuwania, a prawy to wsuwanie pomostu i reset licznika akcji wysuwania.

W celu zmiany wysunięcia pomostu należy wybrać opcję „Zmien długość”. Jego naciśnięcie wpływa na wartość wyświetlaną w polu tekstowym. Użytkownik naciskając przycisk określa ilość razy modyfikuje wartość, tak długo aż ta będzie zgodna z pożądaną. Po naciśnięciu przycisku „Koniec wprowadzania”, program zmodyfikuje wartość ustawioną przez użytkownika, wyrażoną w metrach.



Rysunek 8: Widok całej obudowy w rzucie izometrycznym

**Model CAD** Konstrukcja urządzenia i elementów jego obudowy zewnętrznej został wykonany w środowisku AutoCAD Fusion 360. Przedstawiono model w rzucie izometrycznym (Rysunek 8) [12].

Jako łączenie poszczególnych elementów zostało przewidziane spawanie oraz wykorzystanie śrub M12. W projekcie zostały uwzględnione dwa wejścia, tak aby umożliwić konserwację, wymianę lub montaż odpowiednich komponentów mechanizmu. Jako dodatkowe wzmocnienie zespołu części znajdujących się na ziemi przewidziano montaż do płyty betonowej bądź innego materiału.

Założono, że panel operatorski wraz z pozostałymi komponentami będzie znajdował się na powierzchni ziemi. Natomiast sam napęd będzie przenoszony przez wałek znajdujący się w obudowie, a następnie, za pomocą przekładni łańcuchowej, na kolejny wałek z gumowymi pierścieniami tak, aby tarcie mogło wypychać całość pomostu.

Konstrukcja musi również spełniać podstawowe normy szczelności dla sprzętu elektronicznego pracującego w takich warunkach. Konstrukcja powinna przynajmniej spełniać normę IP54 czyli ochronę przed wnikaniem pyłu w ilości mogącej zakłócić pracę urządzenia oraz ochronę przed kroplami wody padającymi pod dowolnym kierunkiem z różnych stron. Rozmiary konstrukcji nieprzekraczające  $25 m^2$  w mieście oraz  $35 m^2$  poza nim, nie wymagają posiadania pozwolenia na budowę. Sama ustawa nie przewiduje obostrzeń na temat konstrukcji na terenie plaży.

#### 4. Wnioski

Użytkownik za pomocą panelu dotykowego ma możliwość ustawienia kierunku ruchu pomostu, a także odległości jaką ma on pokonać. Po zatwierdzeniu ustawionych danych użytkownik uruchamia akcję za pomocą przycisku „START”. Do dyspozycji posiada także przyciski „LOCK” oraz „STOP”, które odpowiadają za zablokowanie lub zatrzymanie akcji.

Każde kliknięcie na ekranie panelu sterowania jest przetwarzane przez program sterownika PLC. Poszczególne działania powodują otwieranie się lub zamykanie wybranych cewek. Wystartowanie akcji za pomocą przycisku „START” wprowadza w ruch silniki krokowe, które po osiągnięciu wartości końcowej blokują się uniemożliwiając dalszy obrót.

Zastosowany system złożony jest z wielu elementów składowych, pozwala to na dowolną zmianę jednego z nich w nieskomplikowany sposób. Sterownik PLC jest urządzeniem łatwym w obsłudze i posiada duże możliwości personalizacji dzięki programowaniu z przeznaczonych do tego środowiskach. Język drabinkowy, który został wykorzystany do napisania programu sterującego, posiada jednakowe składowe niezależnie od wykorzystywanego środowiska pro-

gramistycznego. Wykorzystanie sterownika silnika krokowego umożliwia rozbudowę projektu o kolejny silnik bez konieczności wykorzystania innych skomplikowanych metod.

Projekt spełnia wszystkie poczynione założenia. Urządzenie nie posiada działającego prototypu, jednakże wykonane na modelu symulacje dają rezultat o wyniku pozytywnym. Widoczne dowody na poprawność jego wykonania objawiają się poprzez reakcję systemu sterującego na akcję wprowadzane za pośrednictwem emulacji panelu sterowniczego na komputerze.

## Literatura

- [1] K. Bisztyga, *Sterowanie i regulacja silników elektrycznych*. Warszawa: Wydawnictwo Naukowo-Techniczne, 1989.
- [2] B. Broel-Plater, *Układy wykorzystujące sterowniki PLC: projektowanie algorytmów sterowania*. Warszawa: Wydawnictwo naukowe PWN, 2008.
- [3] M. Doligalski, M.; Adamski, *Projektowanie strukturalnych programów dla sterowników PLC na podstawie modelu maszyny stanowej UML*. Zielonogóra: Uniwersytet Zielonogórski, 2007.
- [4] Z. Dyląg, Z.; Jakubowicz, A.; Orłoś, *Wytrzymałość materiałów T.1*. Wydawnictwo Naukowo-Techniczne, 1996.
- [5] J. Felis, H. Jaworowski, and J. Cieślik, *Teoria maszyn i mechanizmów. Cz.1 Teoria mechanizmów*. Kraków: AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne, 2008.
- [6] Z. Gientkowski, *Autonomiczne prądnice indukcyjne o wzbudzeniu kondensatorowym i przekształtnikowym*. Bydgoszcz: ATR, 1997.
- [7] T. Gilewski, *Szkola programisty PLC: język LAD w programowaniu sterowników przemysłowych*. Gliwice: Wydawnictwo Helion, 2018.
- [8] A. Hulewicz and Z. Krawiecki, *Sterownik PLC i panel operatorski w układzie automatyki inteligentnego budynku*. Poznań: Uniwersytet Poznański, 2017, no. 92.
- [9] J. Kwaśniewski, *Sterowniki PLC w praktyce inżynierskiej*. Legionowo: Wydawnictwo BTC, 2008.
- [10] E. Mitew, *Maszyny elektryczne. T.2*. Radom: Wydawnictwo Politechnika Radomska, 2005.
- [11] R. Orzechowski, Z.; Prywer, J.; Zarzycki, *Mechanika płynów w inżynierii środowiska*. Wydawnictwo Naukowo-Techniczne, 1997.
- [12] K. Paprocki, *Zasady zapisu konstrukcji*. Warszawa: Oficyna Wydawnicza PW, 2000.
- [13] W. Prosnak, *Statyka płynów i dynamika cieczy*. Warszawa: Państwowe Wydawnictwo Naukowe, 1970.
- [14] J. Przepiórkowski, *Silniki elektryczne w praktyce elektronika*. Warszawa: Wydawnictwo BTC, 2005.
- [15] A. Skibiński, *Obsługa i naprawy silników elektrycznych: zasady bezpiecznej pracy*. Warszawa: Instytut Wydawczy Związków Zawodowych, 1984.
- [16] J. Żółtowski, *Podstawy konstrukcji maszyn: połączenia, łożyskowane, sprzęgła*. Warszawa: Oficyna Wydawnicza PW, 2002.

# Mobilne zarządzanie wynajmowaniem mieszkań. Aplikacja na platformę Android z wykorzystaniem serwera Node.js oraz bazy danych stworzonej przy użyciu MySQL

Belco Sangho, Zbyszko Królikowski

*Instytut Informatyki, Uniwersytet Kazimierza Wielkiego w Bydgoszczy*

**Streszczenie:** *Urządzenia mobilne pozwalają na kreowanie aplikacji opierających się na ciągłej dostępności do Internetu i użytkownika, umożliwiając wprowadzenie na nowy poziom wcześniej istniejących koncepcji. Celem tego badania jest program, który umożliwi zarządzanie najmem mieszkań, automatyzację opłat oraz komunikację wynajmującego z lokatorami, jednocześnie znajdując się w zasięgu ręki. Zostało to zrealizowane przy pomocy aplikacji mobilnej napisanej w środowisku Java, serwera stworzonego przy pomocy środowiska Node.js, oraz bazy danych stworzonej w systemie zarządzania MySQL Workbench. Finalny produkt jest ogromnym ułatwieniem w zarządzaniu mieszkaniami, zarówno dla właściciela jak i lokatora.*

**Słowa kluczowe:** *Android Studio, MySQL Workbench, Node.js, aplikacja mobilna, serwer, Java*

**Abstract:** *Mobile devices allow us to create applications based on constant availability of internet and a user, which we can use to improve previously existing ideas. The aim of the study is to create program, capable of helping with rental management, automatic fees and communication between owner and tenant, while safely remaining within reach. Such application was created with a usage of environment Android Studio with help of Java language, while the backend server was made with a JavaScript runtime environment Node.js, and a database formed in relation database management system MySQL Workbench. Final product is an enormous help with rental management, both for owner and tenant.*

**Keywords:** *Android Studio, MySQL Workbench, Node.js, mobile application, server, Java*

## 1 Wprowadzenie

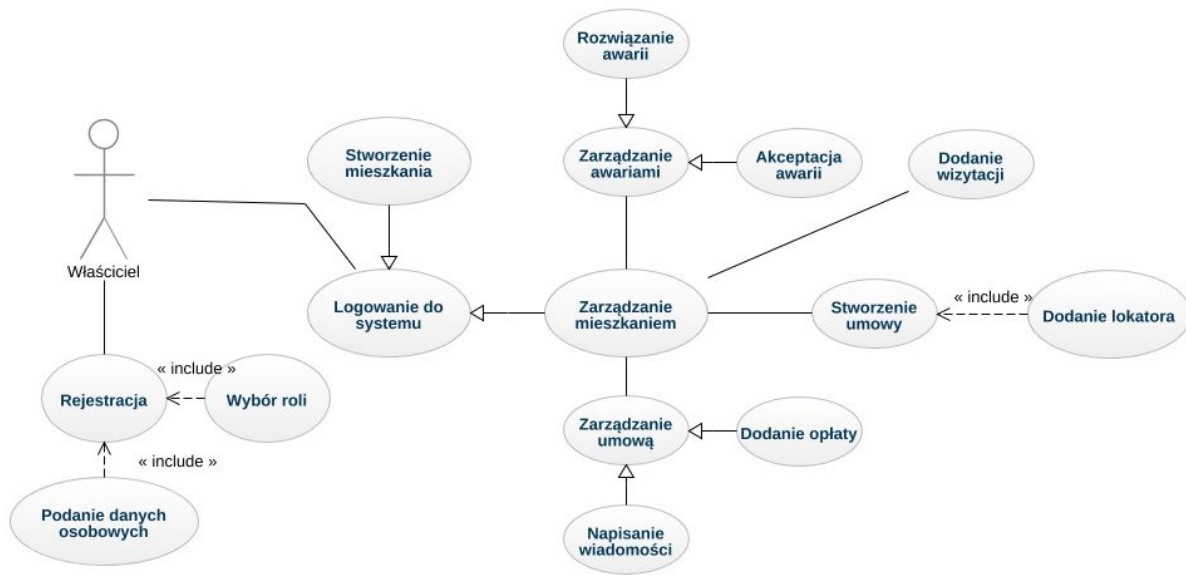
W związku z ogromnym wzrostem cen mieszkań na przestrzeni ostatnich pięćdziesięciu lat[1], popularnym źródłem zarobku wielu dużych i małych firm stało się wynajmowanie przestrzeni do życia dla rodzin. Na przeciw potrzebom takich firm wyszło wiele oprogramowań do zarządzania nieruchomościami, pozwalającymi sprawnie zawiązywać umowy, pilnować płatności oraz wyposażenia. Pozostawiło to jednak małe firmy i osoby prywatne, nierzadko posiadające tylko kilka mieszkań, w sytuacji, gdzie zakup i utrzymanie takich aplikacji byłoby nieopłacalne. Często też oprogramowanie to nie wspierało ciągle rozwijającego się rynku urządzeń mobilnych, który ułatwia monitorowanie poprzez prostotę i szybkość dostępu do informacji.

Idea stworzenia aplikacji która wyszłaby naprzeciw aktualnym potrzebom powstała podczas za-

jęć na Uniwersytecie Kazimierza Wielkiego w grupie współpracującej z firmą Logon. Finalnie została ona dokończona jako praca inżynierska. Projekt miał powstać przy pomocy najnowszych technologii, aby zapewnić mu zarówno prostotę obsługi, oraz prędkość działania.

Jak wynika z raportów popularności systemów operacyjnych zainstalowanych na urządzeniach mobilnych, ponad połowa rynku należy do Androida, jednak poszczególne raporty różnią się pod względem dokładnej wartości. Z tego też powodu zdecydowano się na wykorzystanie tej właśnie platformy. O popularności tego systemu zadecydowała głównie otwartość oprogramowania, która pod licencją Apache zezwala na dowolną dystrybucję i modyfikowanie jego kodu źródłowego, oraz możliwość wykorzystywania go za darmo. Pozwala to na dostosowanie systemu do wymagań twórcy urządzenia.

Aplikację serwerową zdecydowano się napisać w



Rysunek 1: Diagram przypadków użycia 1

Node.js, który oferuje nam możliwość napisania oprogramowania bazującego na nieskończonej pętli zdarzeń, odpowiadającej na przesyłane tam zapytania. Środowisko to działa bardzo dobrze w trybie asynchronicznym, co pozwala na maksymalne wykorzystanie dostępnych nam zasobów serwera.

Częścią wymaganą w takim rozwiązaniu jest baza danych, która została stworzona w języku MySQL. Relacyjne bazy danych charakteryzują się prostotą w tworzeniu oraz interakcji z serwerem, co było powodem takiego wyboru.

## 2 Metody

### 2.1 Diagram przypadków użycia

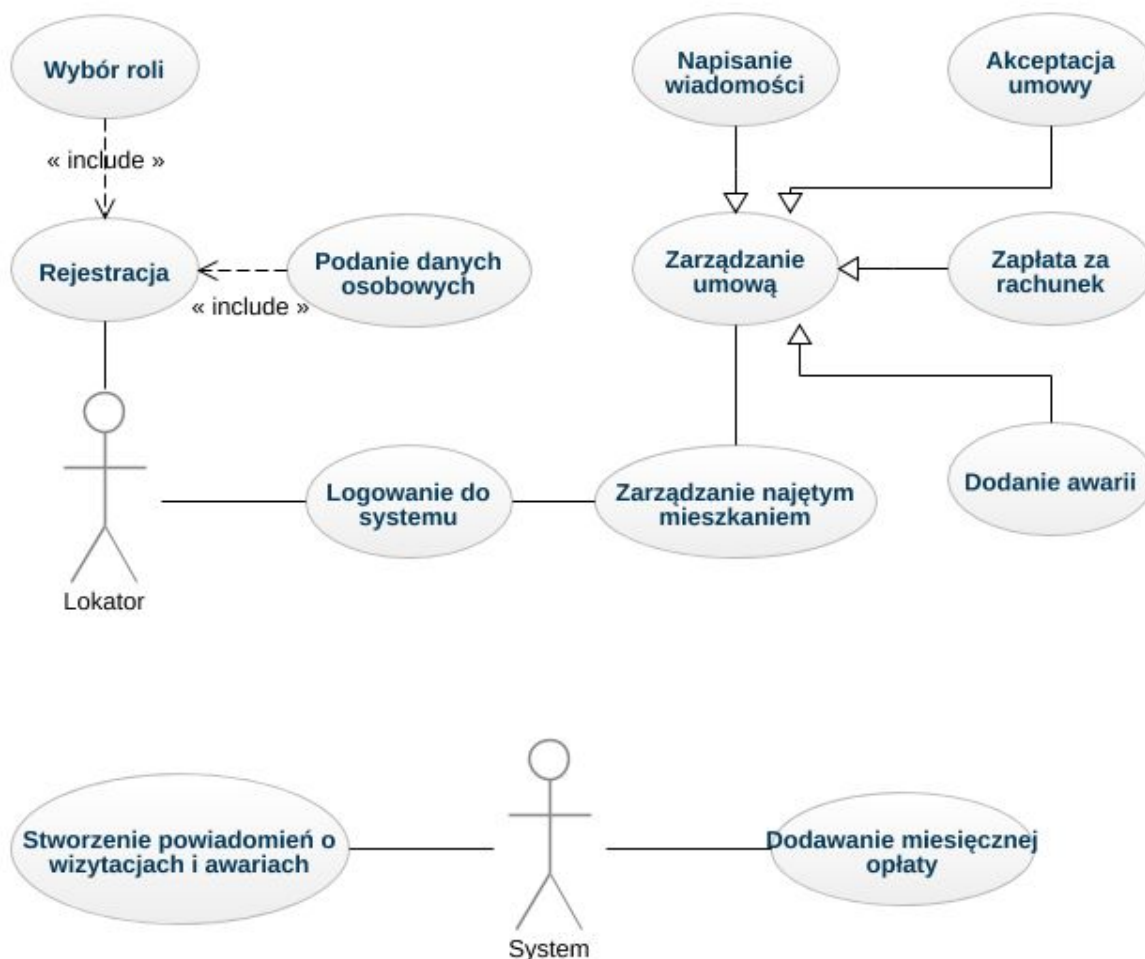
Aby stworzyć tak duży system, pierwszym krokiem powinno być określenie jego ram. Diagram przypadków użycia (Rys. 1 i 2) jest graficznym przedstawieniem wielkości i funkcjonalności architektury, dając wszystkim jasną wizję tego co należy zrobić[2]. Ważnym jego elementem jest uniezależnienie od implementacji, i pokazanie możliwych akcji (opisanych w elipsach) każdego z aktorów. Pozwala to w dalszym etapie programowania na uniknięcie sprzeczności odnośnie tego co jest wymagane, a co jedynie mogłoby się pojawić w finalnym produkcie. Jako że system jest dość rozległy, wymagania są dość duże. Część funkcjonalności jest jednak taka sama dla właściciela i lokatora.

### 2.2 MySQL Workbench

Aby stworzyć relacyjną bazę danych, zdecydowano się wykorzystać program posiadający interfejs graficzny[3], który przyspieszył inicjalną pracę podczas tworzenia schematu bazy. Zmniejszyło to ilość pisania bezpośredniego kodu MySQL. Środowisko to pozwala też na tworzenie wyzwalaczy, wykorzystywanych przy wykonywaniu akcji na rekordach tabeli. Procedury dają możliwość na wywoływanie operacji na bazie, przy minimalnym podawaniu danych i ekspozycji wewnętrznej struktury przez serwer. Za pomocą zdarzeń tworzy się włączające się o specyficznych danych funkcje, co pozwala na zautomatyzowanie pewnych części systemu, takich jak opłaty za mieszkanie. Widoki wykorzystuje się do łączenia danych z kilku tabel, wybierając interesujące nas informacje, za pomocą kluczy głównych identyfikujących rekord oraz kluczy obcych, które wskazują do którego rekordu należy informacja zawarta w innej tabeli.

### 2.3 Node.js

Środowisko Node.js zyskało swoją popularność jako część inicjatywy *JavaScript Everywhere*, odnosząc się do możliwości pisania w tym samym języku aplikacji webowych zarówno po stronie serwera, jak i klienta. Biorąc pod uwagę działanie w otwartym kodzie, każdy może modyfikować i dostosowywać to środowisko do swoich potrzeb[4]. Do-



Rysunek 2: Diagram przypadków użycia 2

myślnym menadżerem paczek jest Npm, a aktywne środowisko programistyczne dostarcza ogromną ilość bibliotek do pobrania. Ważnym aspektem środowiska jest działanie tylko wtedy, kiedy jest wymagana od niego praca, co daje możliwość budowania dużych aplikacji mądrze zarządzającej zasobami. Wraz ze stworzeniem nowych bibliotek istnieje nawet możliwość pisania aplikacji mobilnych w JavaScriptcie, aczkolwiek ta opcja nie została tu wykorzystana.

Jako że Node.js nie posiada domyślnego edytora tekstowego, wykorzystano Visual Studio Code. Środowisko to pozwala na dostosowanie go do każdego języka, oraz szybki dostęp do terminala, z którego pomocą wywoływano komendy Npm-a oraz Noda-a.

Najważniejszym komponentem wykorzystanym

podczas tworzenia serwera jest biblioteka Express. Obecne są w niej ścieżki[5] tworzące interfejs do komunikacji za pomocą zapytań Http. Są one przeszukiwane kolejno od góry do dołu, aż zostanie znaleziona odpowiednia trasa. Dodatkowo wykorzystano kilka pomocniczych bibliotek do komunikacji z bazą danych. Asynchroniczność całego środowiska dodatkowo pozwoliła na szybkie działanie.

Ze względu na powstawanie aplikacji mobilnej i serwera jednocześnie, testowanie ścieżek odbyło się przy pomocy dodatkowego programu Postman, który daje możliwość wysyłania zapytań http i sprawdzania odpowiedzi w każdym formacie, uniezależniając go od programu na urządzeniu mobilnym.

## 2.4 Android Studio

Zbudowane na otwartej wersji środowiska IntelliJ IDEA, Android Studio ładnie przypomina wyglądem i działaniem program który posłużył za jego podwaliny. Jest on darmowy, oraz posiada oficjalne wsparcie Google, producenta systemu Android. Z tego powodu nie należy się dziwić, że jest on obecnie jednym z najefektywniejszych sposobów pisania aplikacji na tą platformę. Domyślnym językiem programowania jest Java, i to ona została użyta w projekcie, aczkolwiek rosnące wsparcie dla Kotliny może niedługo oznaczać wyparcie pierwotnego języka. Szerokie wsparcie bibliotekami tworzonymi zarówno przez związanych i nie związanych z producentem programistów zostało wykorzystane przy integracji z serwerem.

Ekranu tworzone przy pomocy *Relative View*, który pozwala na pozycjonowanie elementów w zależności od siebie, oraz *Linear View*, który układa elementy jeden po drugim w orientacji poziomej bądź pionowej[6]. Duża część funkcjonalności wykorzystywała *RecyclerView*, pozwalający na proste i optymalne dla pamięci wyświetlenie podobnych do siebie elementów dodawanych dynamicznie.

Do komunikacji z serwerem zastosowano tak zwaną *singleton-ową* klasę, która może posiadać tylko jedną instancję. Chroni nas to przed jednoczesnym istnieniem wielu kolejek zapytań, co negatywnie wpływało by na logikę działa programu, oraz mogłoby potencjalnie doprowadzić do wykorzystania zbyt wielu zasobów, i w efekcie wymuszenie przez system zamknięcia programu.

Menadżerem paczek jest Gradle. Zarządza on pochodzeniem bibliotek ze zdalnych repozytoriów, co znacząco ułatwia ich importowanie, oraz pilnowanie ich wersji.

## 3 Rezultaty

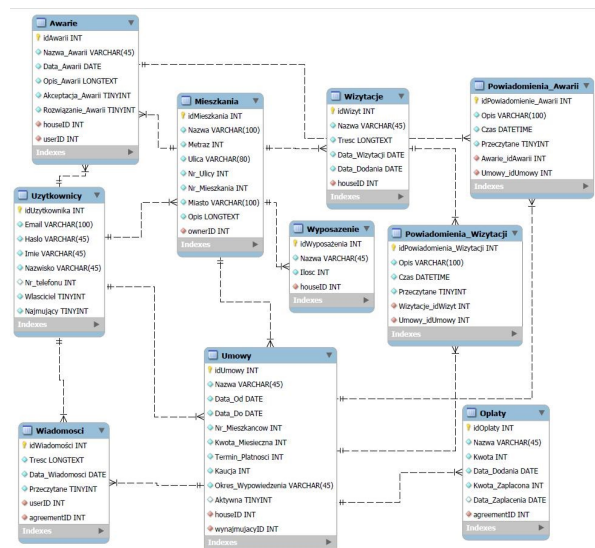
### 3.1 Baza danych

Efektom pracy z systemem zarządzania MySQL Workbench jest baza danych pozwalająca na spełnienie wymagań funkcjonalnych (Rys. 3). Posiada ona dziesięć tabel. Przyjęta została logika, aby nazywać po polsku pola własne tabeli, a klucze obce w języku angielskim. Do paru najważniejszych rzeczy należy:

- Wiele pól w tabeli Umowy, pozwalających na automatyczne wygasanie umów, naliczanie opłat do tabeli Opłaty oraz komunikację przy pomocy tabeli Wiadomości, między użytkownikami, których wiąże umowa.

- Generowanie powiadomień o awariach oraz wizytacjach, odpowiednio dla wszystkich lokatorów i właściciela w przypadku awarii, oraz tylko dla lokatorów w razie wizytacji, razem z przypomnieniem na dzień przed planowanym przybyciem posiadacza mieszkania.
- Duża ilość widoków, które są nieobecne na schemacie bazy danych, pozwalających na sprawne przekazywanie informacji do serwera.
- Kilka funkcji, automatycznie włączanych podczas wyświetlania wiadomości, aby przy następnym odpytaniu bazy były one zaznaczone jako przeczytane.

Finalnie baza pozwoliła na sprawne przekazywanie informacji do serwera.



Rysunek 3: Schemat bazy danych

### 3.2 Serwer

Przy użyciu biblioteki Express udało się utworzyć około 40 zapytań z serwera do bazy danych. Pozwalają one na utworzenie dróg dostępu do informacji, które są później przesyłane w formacie danych JSON do aplikacji mobilnej. Struktura tych danych jest tekstowa, oraz wszechobecna w większości języków programowania, co pozwoliło z kolei na proste odwzorowanie na obiekty w Javie.

Autoryzacja odbywa się poprzez sprawdzanie poprawnych danych podczas logowania, które w poprawnym przypadku pozwalają na dostęp do identyfikatora odpowiedniego użytkownika. Podczas programowania aplikacji upewniono się, że nawet w przypadku komunikacji z drugim użytkowni-

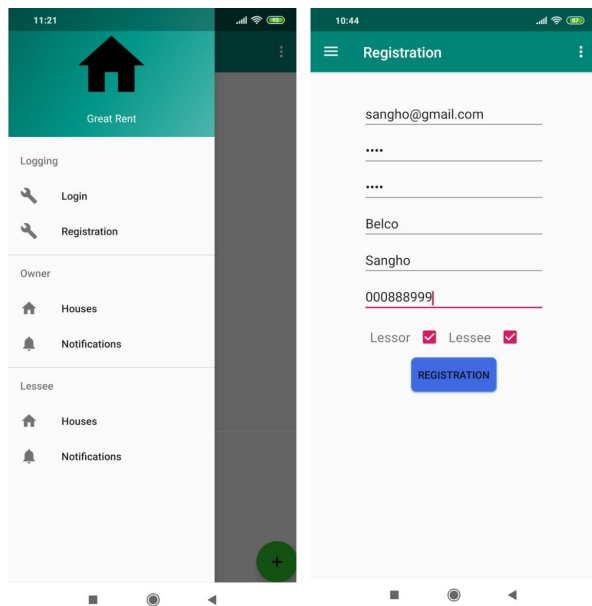


kiem, dane dotyczące haseł (zarówno tego użytkownika jak innych) bądź identyfikatora innego użytkownika nigdy nie dostawały się do programu, co jest dodatkowym źródłem zabezpieczenia.

W celu zapewnienia zdalnego dostępu do serwera, zdecydowano się na wykorzystanie platformy w chmurze Heroku[7]. Zapewnia ona darmową możliwość do stworzenia tego typu małych aplikacji webowych, jeśli nie przekroczymy danej wielkości bazy, oraz pogodzimy się z wolniejszą responsywnością.

### 3.3 Aplikacja mobilna

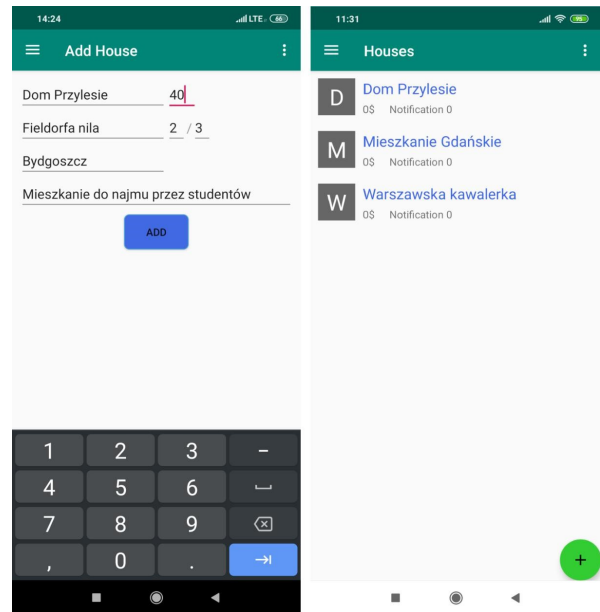
Interfejs użytkownika finalnie posiada dwadzieścia ekranów. Jednym z pierwszych ekranów jest rejestracja, gdzie mamy możliwość wybrania roli właściciela bądź lokatora. Istnieje możliwość posiadania obu ról (Rys. 4). Poruszanie się po aplikacji zostało zrealizowane za pomocą menu wysuwanego z lewej krawędzi. Pozwala to na przełączanie się między widokiem mieszkań a powiadomień, oraz na zmianę funkcji z najemcy z wynajmującego i odwrotnie.



Rysunek 4: Rejestracja i menu

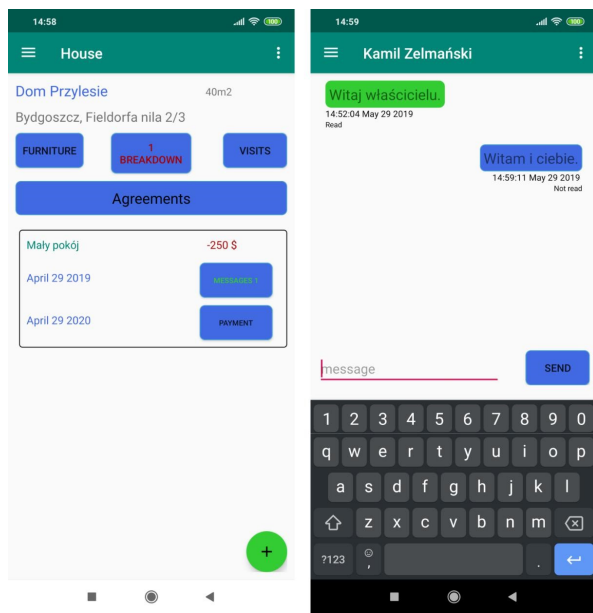
Pierwszy ekran widoczny po zalogowaniu ukazuje nam z perspektywy właściciela przynależne mu mieszkania (Rys. 5), bądź analogiczny widok dla lokatora. Za pomocą przycisku zlokalizowanego w prawym dolnym rogu mamy możliwość dodania nowego lokalu. Funkcjonalność ta powtarza się na każdym ekranie, który pozwala na dodanie nam

nowego elementu. Dodatkowo pod nazwą mieszkania widzimy ogólny bilans pieniężny dla wszystkich umów, oraz informacje o wiadomościach od lokatora i nowych awariach. Kolejne widoki to:



Rysunek 5: Przeglądanie i tworzenie mieszkań

- Ekran umów, widoczny jedynie dla właściciela, który pozwala podejrzeć wszystkie aktywne umowy z bilansami spłat rachunków przez lokatorów oraz możliwość wejścia w szczegółowe informacje o opłatach (Rys. 6). Na przyciskach pojawiają się też informacje o tym czy jest obecna nierozwiązana awaria, bądź nowa wiadomość od lokatora.
- Komunikator, pozwalający na wymianę wiadomości pomiędzy właścicielem oraz lokatorem (Rys. 6). Pod wiadomościami widać godzinę i datę wysłania, jak i to czy odczytaliśmy ją podczas ostatniej konwersacji.
- Przeglądanie awarii, pozwalające nam na ich akceptowanie (aby przekazać informację o zauważeniu do lokatorów), oraz rozwiązanie kiedy problem zostanie zażegnany (Rys. 8). Dodawanie awarii jest stosunkowo proste, ponieważ data dodaje się automatycznie. Przeglądanie i dodawanie wizytacji, mebli oraz opłat wygląda analogicznie, z zastrzeżeniem dodawania dla właściciela (lokator może dodać zapłaty).
- Powiadomienia widoczne ze strony lokatora, na których (Rys. 7) widać nowe umowy zaproponowane przez właścicieli, wizytacje i awarie



Rysunek 6: Przeglądanie umów oraz komunikator

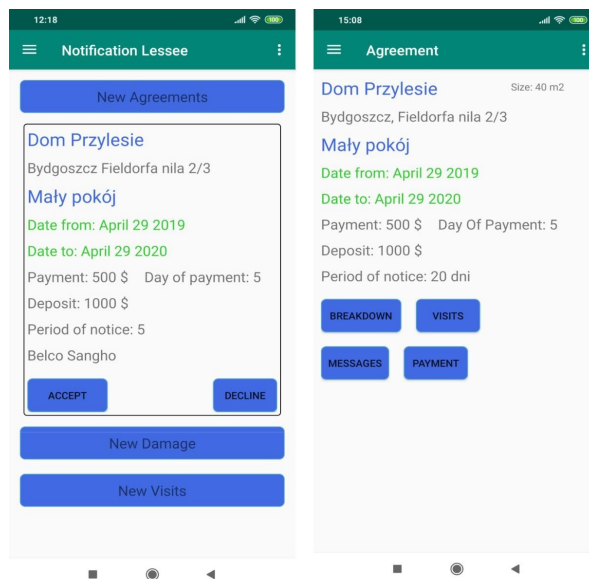
w mieszkaniach z aktywnymi umowami. Umowy w tej perspektywie wyglądają trochę inaczej, jako że mieszkania prowadzą tylko do jednej umowy.

### 3.4 Testy

Aplikacja została przetestowana manualnie za pomocą 6 różnych urządzeń, zarówno realnych, jak i wirtualnych włączanych za pomocą Android Studio. Dodatkowo przy pomocy ochotników (i ich urządzeń) udało się usprawnić pewne funkcjonalności, oraz poprawić wygląd. Serwer jak już zostało wspomniane był testowany przy pomocy programu Postman, aby pominąć możliwe błędy w interpretacji na telefonie. Baza była sprawdzana manualnie pod kontem poprawnych danych, korzystając z MySQL workbench, który pozwala na wykorzystanie graficznego środowiska do łatwiejszego sprawdzania widoków oraz tabel.

## 4 Wnioski

Aplikacja mobilna pozwala efektywnie zarządzać mieszkaniem poprzez zawarte w niej wiele opcji. Jednakże aby móc wprowadzać na rynek ten program, należałoby się uważniej przyjrzeć systemowi szyfrowania i autoryzacji, który musiałby spełniać wysokie standardy, z uwagi na przechowywanie wielu newralgicznych danych, takich jak adresy mieszkań, czy kwoty opłat ponoszonych z tytułu najmu,



Rysunek 7: Przeglądanie powiadomień oraz umowy lokatora

bądź imiona i nazwiska osób powiązanych umową. W przypadku dalszego rozwoju produktu, będzie to najważniejszy następny etap badań.

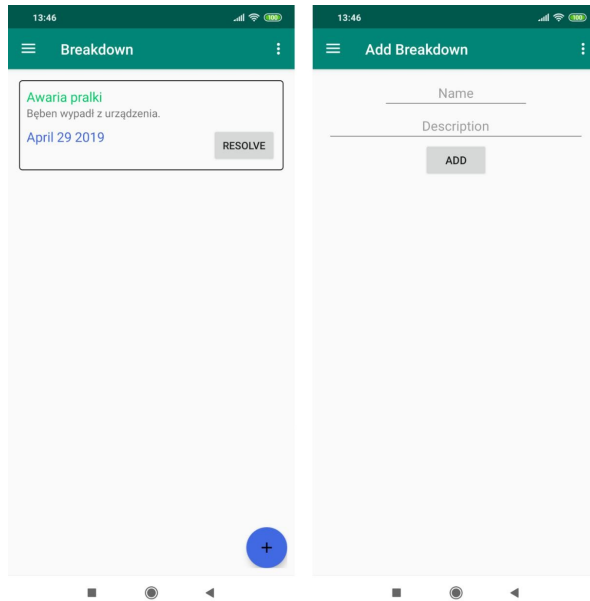
Jako że finalny produkt został stworzony w środowiskach zapewniających szeroką gamę rozszerzeń oraz bibliotek tworzonych przez środowisko programistyczne (Android Studio oraz Node.js), istnieje duży potencjał na rozwój poza aspektem bezpieczeństwa. Kilka kolejnych potencjalnych pól rozwoju to:

- Możliwość dalszego podziału mieszkań na pokoje.
- Podpisywanie umów przez wiele osób, oraz podział opłat zgodnie z ustaloną kwotą.
- Wyświetlanie powiadomień nawet kiedy aplikacja jest wyłączona, aby zmniejszyć potrzebę niepotrzebnego wchodzenia do aplikacji.
- Możliwość wykonywania płatności internetowych poprzez program.

Jak widać więc istnieje wiele możliwych usprawnień, które mogłyby podnieść jakość aplikacji.

## Literatura

- [1] Nathalie Girouard, Mike Kennedy, Paul van den Noord, and Christophe André. Recent house price developments. (475), 2006.



Rysunek 8: Przeglądanie i dodawanie awarii

- [2] Krzysztof Sacha. *Inżynieria Oprogramowania*. PWN Warszawa, 2010.
- [3] Oracle. *MySQL Workbench Manual*, 65006 edition, 02 2020.
- [4] *Node.js Manual Guides*.
- [5] StrongLoop IBN. *Express Routing Manual*.
- [6] Google. *Android Views*.
- [7] Heroku. *Getting Started on Heroku with Node.js*.

# Bydgoski egzoszkielec na rękę - koncepcja i wyniki wstępne

Jakub Kopowski<sup>1</sup>, Izabela Rojek<sup>2</sup>, Dariusz Mikołajewski<sup>2</sup>, Marek Macko<sup>3</sup>

<sup>1</sup> Wydział Psychologii, Uniwersytet Kazimierza Wielkiego w Bydgoszczy; <sup>2</sup> Instytutu Informatyki, Uniwersytet Kazimierza Wielkiego w Bydgoszczy; <sup>3</sup> Wydział Mechatroniki, Uniwersytet Kazimierza Wielkiego w Bydgoszczy

**Streszczenie:** *Możliwość chwytu oraz manipulacje różnymi obiektami stanowią podstawowe umiejętności funkcjonalne umożliwiające dalsze przejście do korzystania z narzędzi, pisania i innych czynności codziennego życia. W niniejszym artykule skupiono się na koncepcji egzoszkielec na rękę dorosłego człowieka oraz wstępnych wynikach w obszarze poprawy parametrów ręki dysfunkcyjnej w porównaniu z ręką zdrową. Pozwala to nie tylko na natychmiastową poprawę funkcji, ale również na kształtowanie jej w dłuższym okresie czasu podczas dalszych faz rehabilitacji.*

**Słowa kluczowe:** *funkcje ręki, model mechaniczny ręki, egzoszkielec.*

## Bydgosian hand exoskeleton – concept and preliminary findings

**Abstract:** *Possibility of grasping and manipulation of various object constitute basic functional abilities allowing for further development toward use of tools, hand writing, and other activities of daily living. This paper focuses on the concept of the hand exoskeleton for adult patients as far as preliminary findings in the area of improvement of the parameters of hand with deficit compared to parameters in healthy hand. It causes not only immediate functional recovery but also shapes this recovery during next phases of the rehabilitation.*

**Key words:** *hand function, mechanical model of the hand, exoskeleton.*

## 1. Wprowadzenie

Egzoszkielec (ang. *exoskeleton*) jako urządzenie mechaniczne służy pasywnemu (w oparciu o elementy sprężyste) lub aktywnemu (w oparciu o zestawy czujników i siłowników) wsparciu ruchów użytkownika. Szczególnym obszarem zastosowania egzoszkielec są funkcje ręki jako najbardziej wielofunkcyjnego naturalnego narzędzia człowieka. Deficyt w tym obszarze wpływa na jakość życia pacjenta, jego zdolność do nauki i pracy, a u dzieci – nauki i zabawy, sprzyjając stygmatyzacji w grupie rówieśniczej oraz dalszym opóźnieniom rozwojowym [1].

Rozwiązania robotyczne stanowią ważną część technicznego wsparcia nauk medycznych oraz praktyki klinicznej, stąd każdy postęp w tej dziedzinie stanowi ważny krok w terapii personalizowanej pacjentów z deficytem w obszarze kończyn górnych – nie tylko z deficytem neurologicznym, ale również z deficytem neurodegeneracyjnym w procesie starzenia się czy – docelowo - amputacjami w wyniku urazów [2-4].

Rozwijając koncepcję bydgoskiego egzoszkielec na rękę wypracowano wszystkie etapy opracowania i wdrożenia: od badania biomechaniki ręki i rozwinięcia jej modelu mechanicznego, poprzez skanowanie, modelowanie, modyfikowanie

parametrów modelu aż po dopasowanie ich do konkretnego pacjenta i praktyczne wykorzystanie w codziennym życiu [5-7]. W niniejszym artykule skupiono się na koncepcji egzoszkielec na rękę dorosłego człowieka oraz wstępnych wynikach w obszarze poprawy parametrów ręki dysfunkcyjnej w porównaniu z ręką zdrową.

## 2. Szczegółowy opis

### 2.1. Zarys koncepcji

Działający egzoszkielec zakładany na rękę użytkownika powinien spełniać następujące wymagania:

- budowa modułowa, dzięki czemu da się dostosować wielkość paliczek modelu do długości, szerokości oraz kształtu paliczek ręki użytkownika i innych wymagań użytkownika,
- “zatrzaski”, które w przypadku niektórych chorób umożliwią utrzymanie prawidłowej kształtu dłoni, w przypadku skurczów,
- kanały od pasków naciągowych, które umożliwią pasywną (jeśli są elastyczne) lub aktywną (jeśli są podłączone do serwo mechanizmów) regulację siły naciągu danego palca,

- możliwość zakładania całej “rękawicy” na dłoń człowieka bądź poszczególnych jej modułów na poszczególne człony (palczki) po kolei [5].

Podczas pracy nad egzoszkieletem wykorzystano pracę wieloetapową - inżynierię odwrotną (ang. *reverse engineering*). Skanowanie 3D, modyfikacje szablonu oraz druk 3D elementów egzoszkieletu umożliwiły dokładne wytworzenie modelu 3D. Należy jednak zaznaczyć, że ta technologia nie jest pozbawiona wad i wymaga spełnienia rygorystycznych założeń, m.in. obiekt skanowany powinien być całkowicie nieruchomy, by w modelu nie było żadnych artefaktów/błędów. W tym celu wykorzystano przezroczyste trzymadła do utrzymania ręki w pozycji nieruchomej.

Na rysunku 1 pokazano skanowaną rękę z już zeskanowanym modelem. Na podstawie skanu 3D ręki

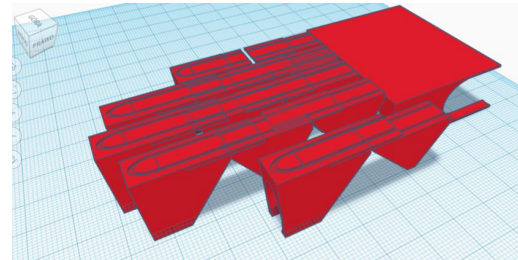
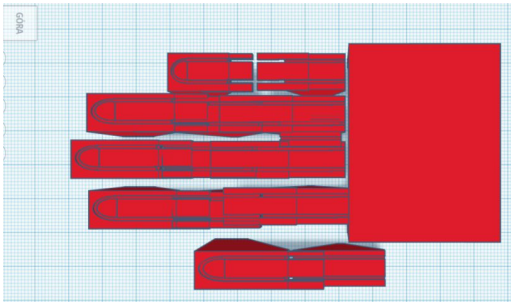
z dysfunkcjami wyznaczono długość poszczególnych elementów ręki (palczków oraz całych palców), potrzebnych do obliczeń i konstrukcji egzoszkieletu. Wyniki wykorzystano później do wykonania symulacji oraz potrzebnych obliczeń.

## 2.2. Opis najważniejszych parametrów

Egzoszkielet ma być zakładany na rękę (tj. odcinek kończyny górnej od palców do nadgarstka) dorosłego człowieka. Każda para kinematyczna (para kości palców) ma swoje odzwierciedlenie w egzoszkielecie jak przedstawiono na rysunku 2.



Rysunek 1. Zeskanowana ręka. Lewy kadr przedstawia przed obróbką, prawy po obróbce i koloryzacji.



Rysunek 2. Koncepcja egzoszkieletu na rękę (rzut z góry i z boku).

Palec ręki interpretuje się jako układ otwarty połączonych ciał o określonej masie i rozmiarach. Każde z tych ciał składa się z:

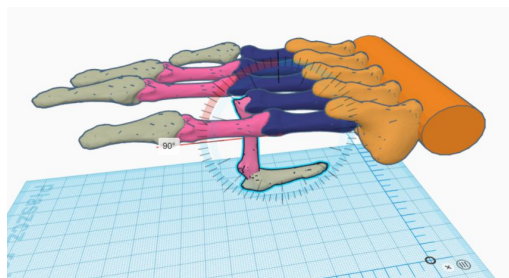
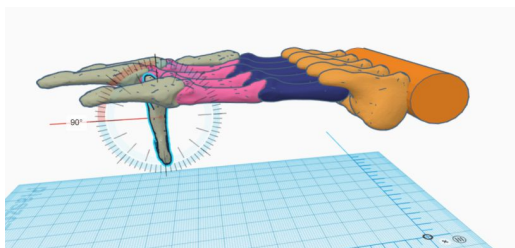
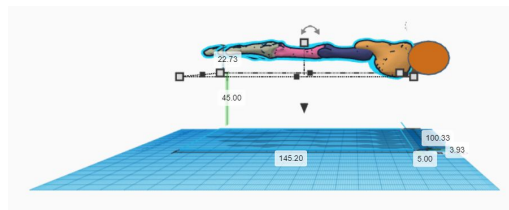
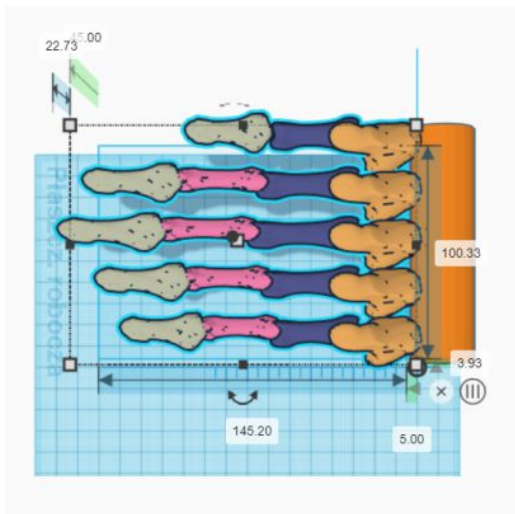
- kości - stanowiącej rusztowanie,
- układu mięśniowo-ścięgowego – stanowiącego siłownik wprawiający w ruch pary kinematyczne palczków,
- tkanki łącznej – łączącej wszystkie elementy, zapewniającej niezbędne paliwo do działania (tlen, węglowodany, tłuszcz) oraz materiały do budowy/naprawy całego mechanizmu.

W omawianym przypadku każdy palczek to osobny element w sensie mechanicznym, a każda para palczków bądź śródrecze-palczek to para kinematyczna. Najistotniejszymi z parametrów są:

- kąty zginania poszczególnych par kinematycznych, z których oblicza się przebytą drogę,
- czas pomiaru, który uzyskuje się z metadanych przy pobieraniu wektora danych z urządzenia pomiarowego,
- wymiary, a w szczególności długości palczków,

– masa poszczególnych paliczków, która potrzebna jest do obliczenia siły, z jaką palce zginają

i rozginają się.



Rysunek 3. Ręka jako układ mechaniczny z przedstawionym przykładem zginania palca oraz wyszczególnionymi paliczkami.

Prędkość chwilowa

Dla ruchu wzdłuż prostej prędkość definiuje się jako pochodną drogi po czasie, czyli granicę przyrostów przesunięcia do przyrostu czasu, w jakim nastąpił ten przyrost, dla nieskończenie małego przyrostu czasu. Droga  $s$  jest rozumiana jako długość odcinka krzywej (toru), po której porusza się ciało, od punktu początkowego do końcowego ruchu. Jeżeli prędkość zmienia się, to droga jest równa sumie małych odcinków drogi, na których uznaje się, że prędkość jest stała i jest określana jako prędkość chwilowa

$$v = \frac{s'}{t'} = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t} \quad (\text{wzór 1})$$

Ww. definicja została wybrana z powodu zasady działania urządzenia oraz programu pomiarowego. Karta i czujnik pomiarowy wysyłają w jak najmniejszym możliwym odstępie czasowym dane do programu pomiarowego. Program zapisuje dane dla wszystkich stawów oraz czas. Masa i długość paliczków jest wprowadzana osobno.

$$v = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t} \quad (\text{wzór 2})$$

gdzie:

$\Delta t$  - czas między odczytami czujnika,  
 $\Delta s$  - obliczone przemieszczenie (droga) pomiędzy odczytami czujnika, za pomocą wzoru na drogę po okręgu.

Zmiana prędkości

$$V \Delta = v - v \text{ poprzednie } v \quad (\text{wzór 3})$$

gdzie:

$v$  – aktualna wartość prędkości chwilowej  
 $v$  poprzednie – poprzedni pomiar prędkości chwilowej

Przyspieszenie dośrodkowe

Przyspieszenie dośrodkowe (przyspieszenie normalne) - rodzaj przyspieszenia, w którym ciało doznaje na skutek działania siły lub jej składowej prostopadłej do wektora prędkości ciała. Kierunek i zwrot tego przyspieszenia jest zgodny z kierunkiem i zwrotem tej siły.

W wyniku przyspieszenia normalnego nie zmienia się wartość prędkości, tylko jej kierunek

$$a_d = \frac{v^2}{r} \quad (\text{wzór 4})$$

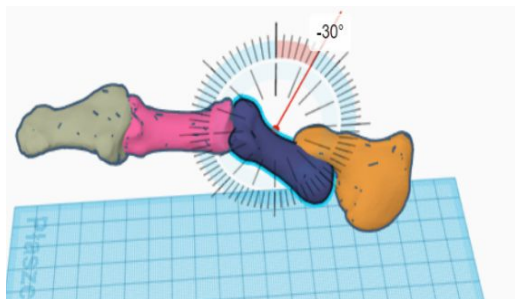
Przyspieszenie dośrodkowe pozwala określić kierunek działania siły czy w naszym przypadku kierunek zginania się danego paliczka. Na podstawie tego można określić czy i jaki kierunek ma ciało (paliczek). Ta informacja pozwala określić kierunek zmiany ruchu, nawet jeśli ręka porusza się ruchem jednostajnym. W przypadku nagłych zmian wartości przyspieszenia normalnego dostajemy informację, że nastąpiła nagle zmiana kierunku ruchu np. skurcz w parze paliczków (parze kinematycznej). Dzięki tej informacji egzozszkielet/komputer może zawnoczu przewidywać i zastosować odpowiednie środki zaradcze, które zniwelują anomalie w ruchu. Dzięki informacji o nagłych zmianach wartości oraz wiedzy o maksymalnych i minimalnych zakresach ruchów stawów można określić algorytm/ zestaw działań zaradczych w celu zniwelowania przegięć jak i w przypadkach problemu, kiedy ręką nie może być pełni zgjęta.

#### Przyspieszenie styczne

Przyspieszenie styczne to składowa styczna do toru ruchu obiektu równoległa do prędkości stycznej, która charakteryzuje zmianę wartości prędkości.

$$a_d = \frac{\Delta V}{\Delta t} \quad (\text{wzór 5})$$

Umożliwia określenie wartości przyspieszenia całkowitego. W przypadku zmian tego parametru dowiadujemy się o wartości przyspieszenia. Co wpływa na zmianę wartości siły.



Rysunek 4. Nieprawidłowe wyprostowane kolejnych segmentów/członów (par kinematycznych) palca.

Kolejnym problemem są wahania w wartościach przyspieszenia, czasu i kąta, pod jakim zginają

#### Przyspieszenie właściwe całkowite

$$Ac = \sqrt{a_d^2 + a_s^2} \quad (\text{wzór 6})$$

#### Siły

Siła dośrodkowa – siła powodująca zakrzywianie toru ruchu ciała, skierowana wzdłuż normalnej (prostopadle) do toru, w stronę środka jego krzywizny.

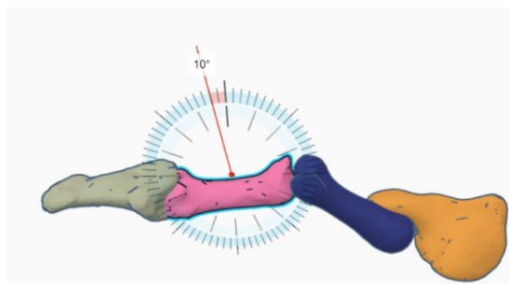
$$f_d = m \cdot a_d \quad (\text{wzór 7})$$

Siła styczna do prędkości (chwilowej) - siła styczna do toru ruchu i równoległa do przyspieszenia stycznego. Zmiana jej wartości powoduje zmianę wartości prędkości jak i przyspieszenia, ale nie zmienia kierunku.

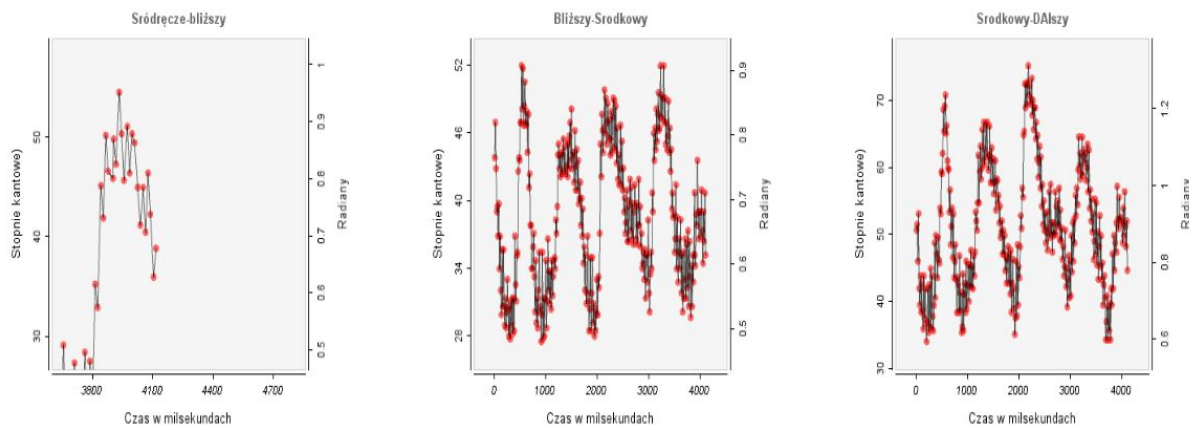
$$f_s = m \cdot a_s \quad (\text{wzór 8})$$

### 3. Opis przypadku

Jako pacjentkę do zbadania i zbudowania egzozszkieletu na rękę wybrano 23-letnią osobę z mózgowym porażeniem dziecięcym (MPD, ang. *cerebral palsy - CP*). Palce ręki lewej badano w następującej kolejności: palec wskazujący, palec środkowy, palec serdeczny, palec mały, kciuk. Ochotniczka z racji swej dysfunkcji miała problemy z prawidłowymi kątami zginania palców, co prezentuje najlepiej para kinematyczna paliczek środkowy dalszy (rys. 4). Należy zwrócić uwagę, że każda para np. śródrcze-bliższy zgina się prawidłowo.



i prostują się poszczególne pary kinematyczne w palcu (pary paliczków) (na rys. 5 i rys. 7).



**Rysunek 5.** Zakres kątów w czasie dla poszczególnych stawów, par kinematycznych paliczków.

Jak przedstawiono na rysunku 5 wartości zmian kątów poszczególnych par kinematycznych są stosunkowo duże w czasie: dla osoby z dysfunkcją ręki wahają się 25° do 50°. Pacjentka ma problemy z utrzymaniem ręki w określonej pozycji, w określonym chwycie i ma problem ze świadomym kontrolowaniem swojej ręki.

Takie nagłe zmiany wartości kąta w czasie przekładają się na zmiany w prędkościach i przyspieszeniach, które później przekładają się na siłę. Nagłe zmiany przyspieszeń statycznych i dośrodkowych w krótkich jednostkach czasu uniemożliwiają prawidłowe funkcjonowanie ręki.

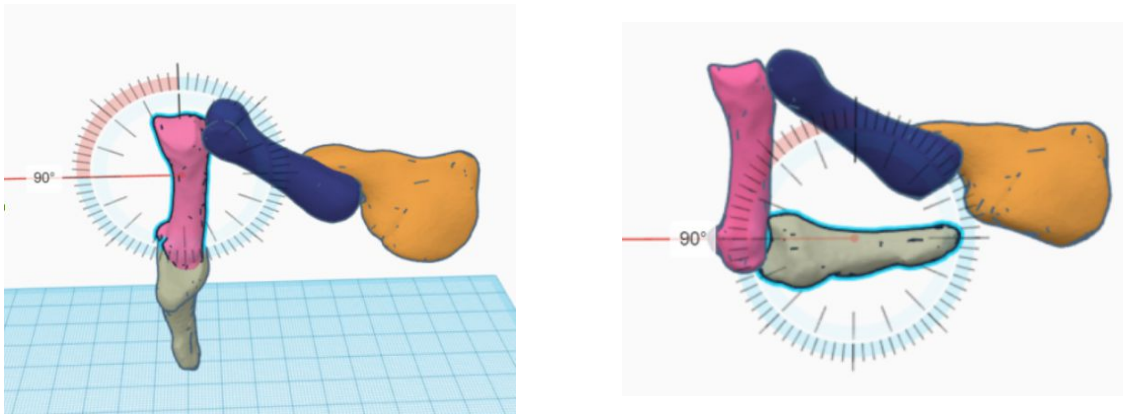
**Tabela 1** Przykładowe wyniki dla palca wskazującego w zakresie zginania palców (w stopniach)

Para kinematyczna	Dla zdrowej osoby		Dla osoby z deficytem	
	Wyprostowana ręka	Zaciśnięta w pięść	Wyprostowana ręka	Zaciśnięta w pięść
Środkowy -Dalszy	0	90	0	90
Bliższy-Środkowy	0	90	10	90
Śródrecze-Bliższy	0	90	-30	-30

W tabeli 1 pokazano wartości krańcowe dla ręki zdrowych oraz deficytem. W dwóch przypadkach, pierwszym osoba miała być maksymalnie wyprostowaną rękę i położyć ją na płaskiej powierzchni, drugim natomiast miała zaciśnąć rękę w pięść. Dla osoby zdrowej ręka wszystkie paliczki przyjmowały wartość 0°, czyli pozycję wyprostowaną. Natomiast dla osoby z deficytem funkcjonowania ręki zaobserwowano duże zmiany wartości w porównaniu z ręką zdrową, np. para śródrecze-bliższy wyginała się w tym miejscu w sposób patologiczny w przeciwnym kierunku niż u ręki zdrowej oraz kącie 30 stopni

przeciwnie do kierunku zginania ręki zdrowej. Taka anomalia wynika z długotrwałego nieprawidłowego wzrostu kości oraz systemu mięśniowego. Należy zwrócić uwagę, że pacjentka jest chora ma MPD od urodzenia, w okresie dojrzewania następowały nieprawidłowe ukształtowania kości oraz stawów, czyli elementów niezbędnych do prawidłowego funkcjonowania ręki człowieka. W parze kinematycznej paliczek bliższy-środkowy są też anomalie, ręka uniemożliwia wyprostowanie się do prawidłowej pozycji.





Rysunek 7. Anomalne zginanie palców

W przypadku zginania palców u osoby z deficytem występują również braki, szczególnie w przypadku pary śródręcze-bliższy występuje w tej parze uniemożliwienie zginania prawidłowego.

Czas i zakres kąta nie są całkowicie poza kontrolą pacjentki, ale w chwilach np. stresu ręka pacjentki zgina się i uniemożliwia jej wyprost albo wykonanie chwytu.

Egzoszkieleł na obecnym etapie pomaga ograniczyć nieprawidłowe kąty, ale nadal pozostały nagłe zmiany wartości i kierunku przyspieszeń o różnej nieregularnej częstotliwości. Trwają prace nad ulepszeniem egzoszkieletu, by zniwelować niekontrolowane wahania wartości i kierunku przyspieszeń, co spowoduje zwiększenie jakości życia pacjentki. Dużą pomocą okazało się stworzenie autorskiego oprogramowania, które umożliwi lepsze zilustrowanie i zrozumienie przyczyn takiego stanu rzeczy.

#### 4. Dyskusja

Następnymi krokami są:

- ulepszenie przyrządów pomiarowych, narzędzi oraz procedur badawczych w celach bardziej precyzyjnego oraz mniej czasochłonnego przeprowadzania badań,
- budowa bardziej ergonomicznego egzoszkieletu tak by osoba z dysfunkcjami mogła sama go zakładać,
- dopracowanie środowiska programistycznego w celu łatwiejszego i precyzyjniejszego współdziałania ręki z egzoszkieletem
- opracowanie nowych systemów sterowania, w tym opartych na interfejsach mózg-komputer [8, 9], z uwzględnieniem oddziaływania emocji [10-12].

Odrębną kwestią jest dobór parametrów technologii oraz materiałów właściwych dla optymalnego wykonania elementów egzoszkieletu [13,14], tak by spełniały one wymogi stawiane robotom rehabilitacyjnym [15,16]. Ww. badania są

przebiegane, a wyniki na skalę przemysłową są oczekiwane w najbliższych latach.

#### Wnioski

Obecnie zrealizowano prace w zakresie opisanego podstawowych parametrów biomechanicznych takich jak ich masa, szerokość długość poszczególnych palców, zakresy ruchu poszczególnych par (palcików) kinematycznych w palcach, a także czas, w jakim jest wykonywany ruch. Umożliwia to obliczenie przyspieszenia statycznego, dynamicznego oraz siły statycznej, dynamicznej i prędkości dla poszczególnych par kinematycznych dla poszczególnych elementów egzoszkieletu, co pozwoli lepiej dopasować egzoszkieleł do potrzeb pacjenta, a przyszłościowo: opracować i zoptymalizować procedurę w tym obszarze.

#### Literatura

1. Mikołajewska E. Terapia ręki. Diagnostyka i terapia. Soyer, Warszawa 2016.
2. Mikołajewska E., Mikołajewski D. Exoskeletons in neurological diseases – current and potential future applications. *Advances in Clinical and Experimental Medicine* 2011; 20(2):227–233.
3. Mikołajewska E., Mikołajewski D. Zastosowania automatyki i robotyki w wózkach dla niepełnosprawnych i egzoszkieleciach medycznych. *Pomiary Automatyka Robotyka* 2011; 5: 58–64.
4. Mikołajewska E. Egzoszkielety. W: Przeździecki B., Woźniowski M., Gieremek K., Janicki S. (red.). *Wyroby medyczne. Zaopatrzenie indywidualne*. Warszawa: PZWL, 2016.
5. Kopowski J., Mikołajewski D., Rojek I., Macko M. Bydgoski egzoszkieleł na rękę – aspekty mechaniczne. *Mechanik* 2019; 4: 271-274.
6. Kopowski J., Rojek I., Mikołajewski D., Macko M. 3D Printed Hand Exoskeleton - Own Concept W: Trojanowska J., Ciszka O., Mendes Machado J., Palenko I. *Advances in Manufacturing II. Vol. 1 - Solutions for Industry 4.0*. Springer, Cham 2019

- Seria: Lecture Notes in Mechanical Engineering, 2195-4356, ss. 298-306
7. Kopowski J., Mikołajewski D., Macko M., Rojek I. Bydgosian hand exoskeleton - own concept and the biomedical factors. *Bio-Algorithms and Med-Systems* 2019, 15(1):20190003.
  8. Mikołajewska E., Mikołajewski D. Non-invasive EEG-based brain-computer interfaces in patients with disorders of consciousness. *Military Medical Research* 2014; 1(1):14.
  9. Mikołajewska E., Mikołajewski D. Ethical considerations in the use of brain-computer interfaces. *Central European Journal of Medicine* 2013; 8(6):720-724.
  10. Mikołajewska E., Mikołajewski D. Informatyka afektywna w zastosowaniach cywilnych i wojskowych. *Zeszyty Naukowe/Wyższa Szkoła Oficerska Wojsk Lądowych im. gen. T. Kościuszki* 2013; 2:171-184.
  11. Picard R.W., *Affective computing*, MIT Press, Massachusetts 1997.
  12. Marechal C., Mikołajewski D., Tyburek K., Prokopowicz P., Bougueroua L., Ancourt C., Wegrzyn-Wolska K. Survey on AI-based multimodal methods for emotion Detection. *High-Performance Modelling and Simulation for Big Data Applications* 2019, pp. 307-324.
  13. Macko M., Mikołajewska E., Szczepański Z., Augustyńska B., Mikołajewski D. Repository of images for reverse engineering and medical simulation purposes. *Medical and Biological Sciences*, 2016; 30(3):23-29.
  14. Macko M., Szczepański Z., Mikołajewski D., Mikołajewska E., Listopadzki S. The method of artificial organs fabrication based on reverse engineering in medicine. *1st Renewable Energy Sources-Research and Business (RESRB-2016)*, June 22-24 2016, Wrocław, Poland pp. 353-365 Springer, Cham 2016.
  15. Mikołajewska E., Mikołajewski D. Roboty rehabilitacyjne. *Rehabil. Prakt* 2010; 4:49-53.
  16. Gorgey A. S. Robotic exoskeletons: The current pros and cons. *World J Orthop.* 2018; 9(9):112-119.