

## **Temat projektu:**

# System zarządzania kręgielnią

## **Wykonawcy:**

Jakub Puchalski

Łukasz Włodarczyk

Kuba Wyrzykowski

GitHub: <https://github.com/infernus0520/System-zarz-dzania-kr-gielni->

## **1. Opis projektu**

Aplikacja została stworzona w celu ułatwienia zarządzania salonami usługowymi takimi jak kręgielnia. Ma ona za zadanie wspomagać pracę właściciela poprzez wyświetlanie statusu aktualnie zarezerwowanych terminów i torów, a także pozwalać ustalać osobny grafik dla każdego pracownika. Pracownikom zaś, ma ona ograniczyć zakres papierowej pracy, a przede wszystkim tę pracę ułatwić. Wszystkie dane o kliencie będą od teraz bezpiecznie przechowywane w bazie. Pracownicy nie będą musieli przeglądać kalendarza w celu znalezienia dostępnych terminów wizyt, czy też sprawdzenia jakie tory zostały zarezerwowane, gdyż wszystko zawarte będzie w systemie.

## **2. Kontekst systemu**

System do zarządzania kręgielnią jest aplikacją webową dostępną poprzez przeglądarkę internetową.

Głównymi użytkownikami, a jednocześnie osobami do których skierowany jest system to właściciele firm oraz pracownicy owej placówki. To oni będą mieli dostęp do najważniejszych funkcji takich jak:

- rezerwacje
- obsługa klienta
- obsługa pracowników

Kolejnym odbiorcą jest klient korzystający z usług salonu, który po zalogowaniu się na swój profil będzie miał możliwość wglądu w swoje dane znajdujące się obecnie w systemie. Mini-profil klienta zawiera również informację o aktualnym stanie punktów lojalnościowych.

## **3. Cele projektu**

- ułatwienie zarządzania kręgielnią, pracownikami, terminami wizyt w kręgielni oraz rezerwacją torów
- przywiązanie klientów do kręgielni poprzez punkty lojalnościowe i newsletter
- personalizacja klienta
- indywidualny koszt usługi

## **4. Technologia oraz docelowe środowisko**

### **4.1. Java**

Java 8 to najnowsza wersja oprogramowania Java, która zawiera nowe funkcje, rozszerzenia oraz poprawki mające na celu zwiększenie wydajności programowania i obsługi programów Java.

### **4.2. Vaadin**

Vaadin jest narzędziem służącym do budowania bogatych aplikacji internetowych. Posiada architekturę która znajduje się po stronie serwerowej. Vaadin używa języka Java jako języka programistycznego. Narzędzie to bazuje na programowaniu opartym o zdarzenia (ang. event-driven programming). Jest to narzędzie które pozwala w bardzo prosty sposób stworzyć graficzny interfejs oraz zaprogramować do niego zdarzenia. Vaadin to bardzo dobry wybór dla systemów w których często wykonuje się operację dodawania, edytowania a także usuwania danych.

### **4.3. Gradle**

Gradle jest elastycznym narzędziem do budowy projektów , w tym przypadku Java'owych. Umożliwia pisanie skryptów budujących w opartym na Groovym DSLu, co sprawia, że wyrażenie w nich algorytmów staje się banalnie proste. Gradle pozwala na tworzenie zależności między zadaniami, i opiera się na paradygmacie "*convention over configuration*" polegającym na tym że konfiguracja jest zawsze możliwa i łatwa do przeprowadzenia. Gradle wykracza poza możliwości innych narzędzi, jednocześnie stosuje się do przyjętych przez społeczność Javy niepisanych standardów.

### **4.4. H2**

Szybka relacyjna baza danych która zajmuje bardzo małą ilość pamięci. Używana głównie w systemach gdzie nie mamy do czynienia z ogromnymi ilościami danych. Dla naszego systemu zawierającego stosunkowo niewielką ilość danych jest ona najlepszym wyborem.

### **4.5. Spring**

Spring jest szkieletem tworzenia aplikacji w języku Java. Spring zapewnia generyczną warstwę abstrakcji dla zarządzania transakcjami. Pozwala na dołączanie własnych komponentów zarządzających transakcjami (ang. transaction manager) i łatwe odgraniczenie transakcji bez zagłębiania się w niskopoziomowe szczegóły. Spring zapewnia bazujące na JavaBeans zarządzanie konfiguracją przez stosowanie zasad Odwrócenia sterowania. Pozwala to na szybsze i prostsze składanie aplikacji. IoC kojarzone jest także z "Wstrzykiwaniem zależności", aczkolwiek to drugie jest pojęciem węższym. Wszystkie udogodnienia związane z tym narzędziem pozwalają znacząco zredukować ilość kodu w projekcie a także zwiększyć jego czytelność.

## 5. Narzędzia

### 5.1.IntelliJ

IntelliJ IDEA jest komercyjnym zintegrowanym środowiskiem programistycznym (IDE) dla Javy firmy JetBrains. Jest to środowisko za pomocą którego stworzony został projekt.

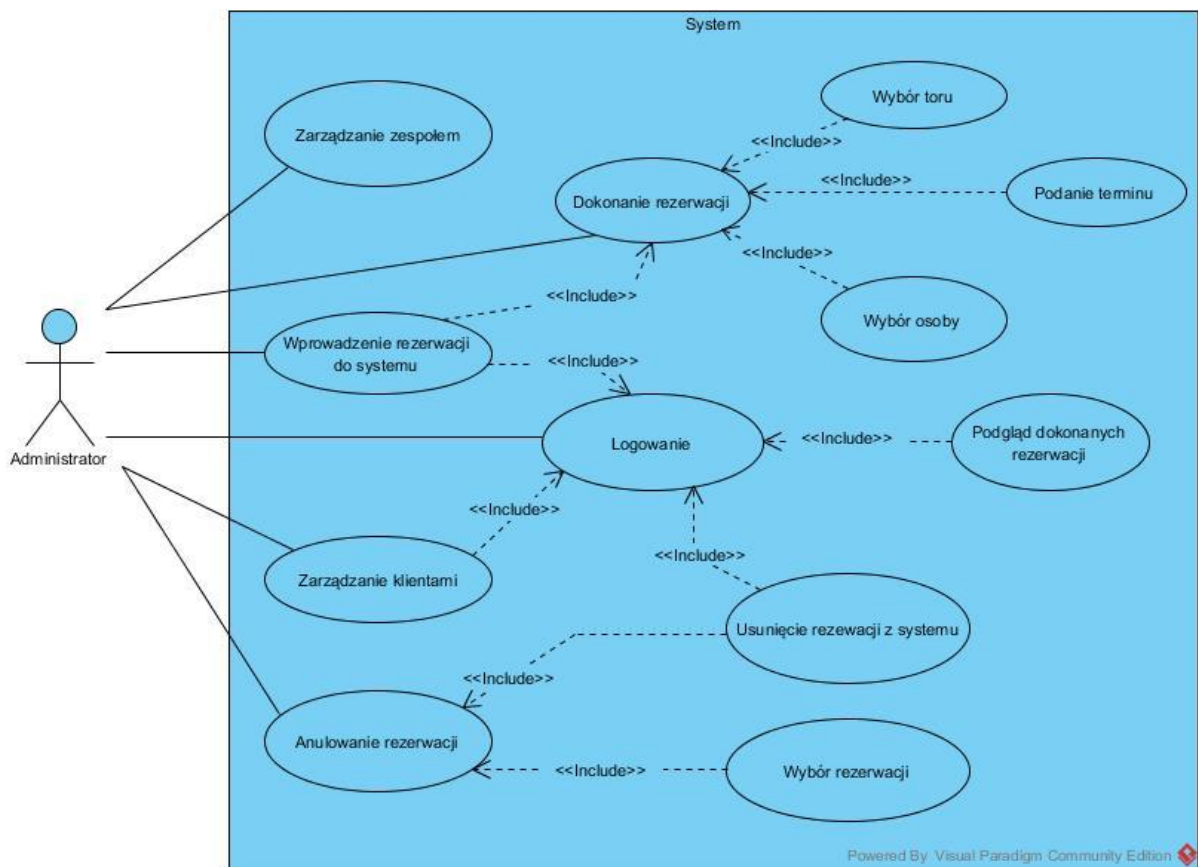
### 5.2.TortoiseGit

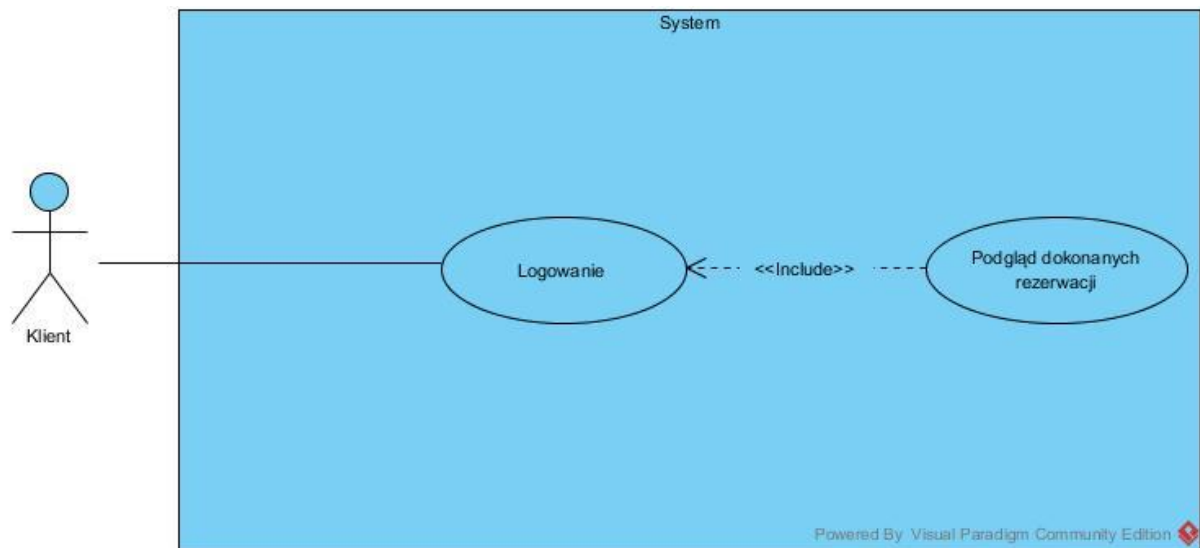
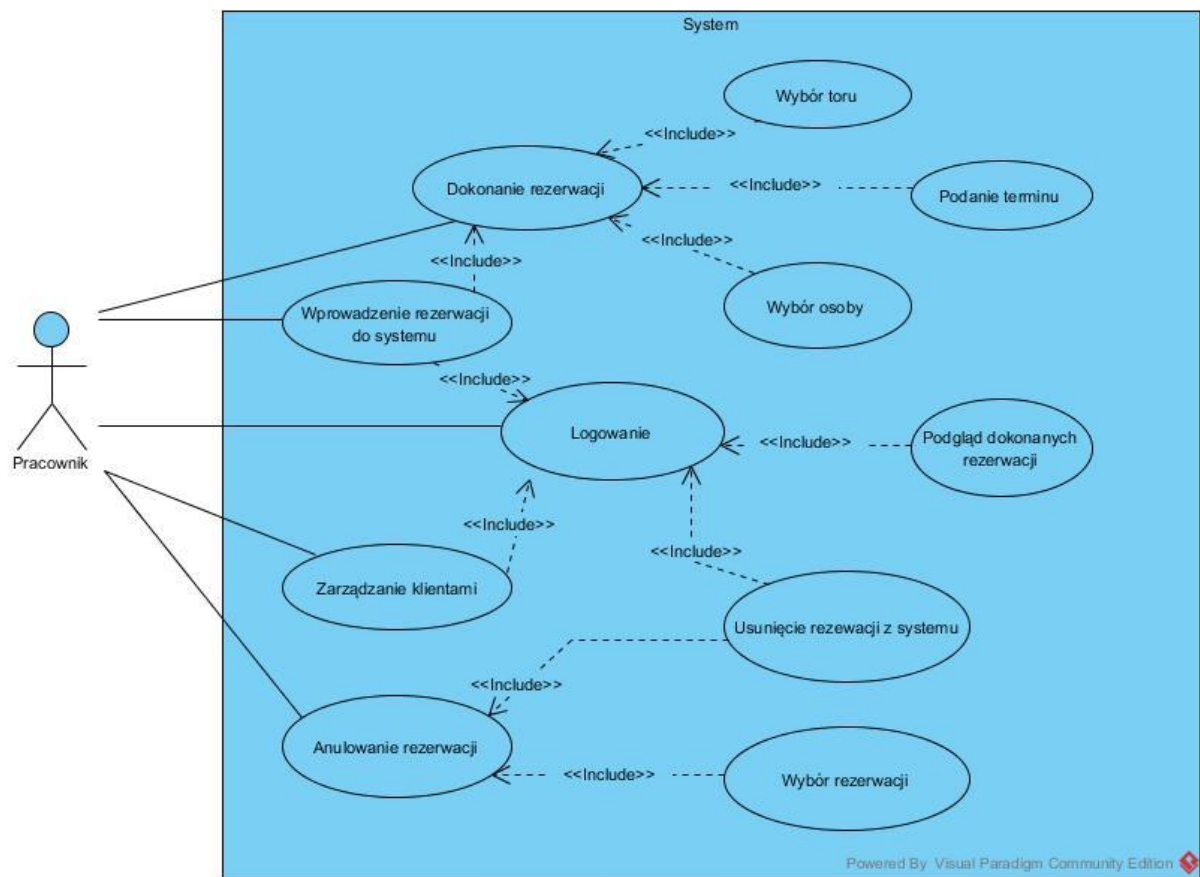
Wersja graficzna systemu kontroli wersji Git. Zdecydowanie ułatwia oraz przyspiesza wykonywania operacji na repozytorium, ponieważ nie wymaga znajomości oraz używania jakichkolwiek komend w konsoli.

### 5.3.Github

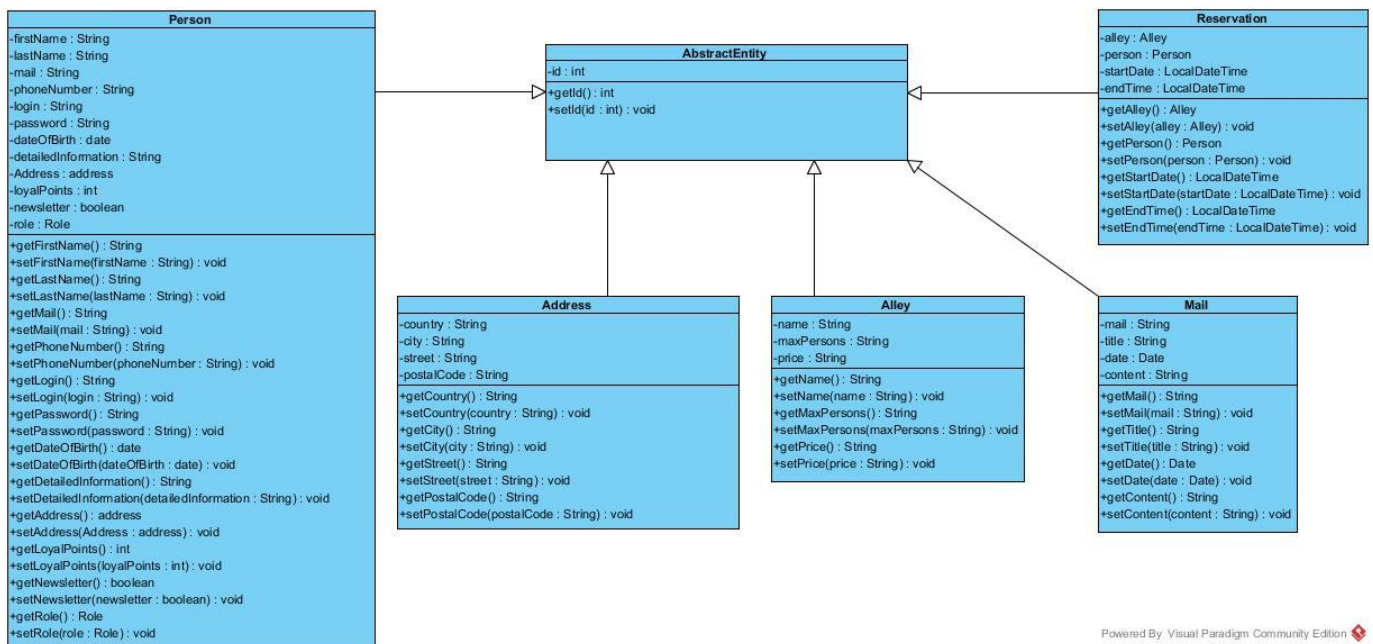
Github - jest to miejsce do przechowywania projektów w chmurze(wirtualnym dysku), wykorzystujące system kontroli wersji GIT albo Mercurial. Dzięki tej usłudze do projektu może mieć dostęp kilka osób w każdej chwili, na każdym urządzeniu.

## 6. Diagram przypadków użycia

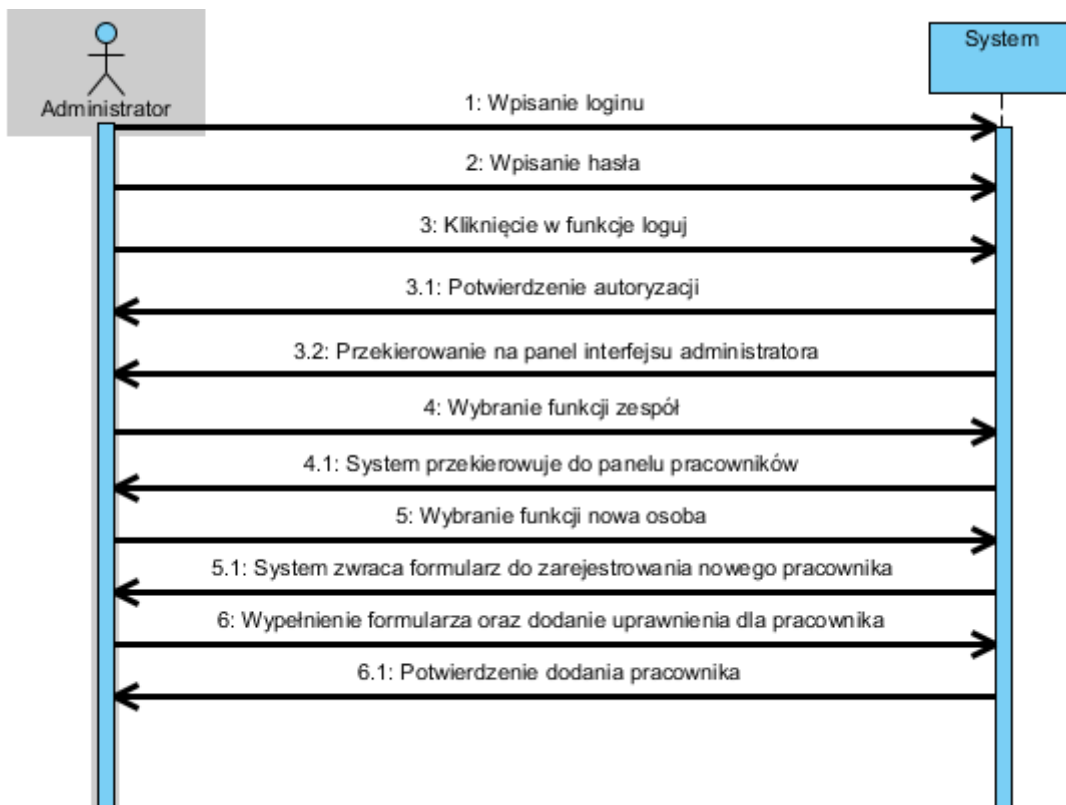




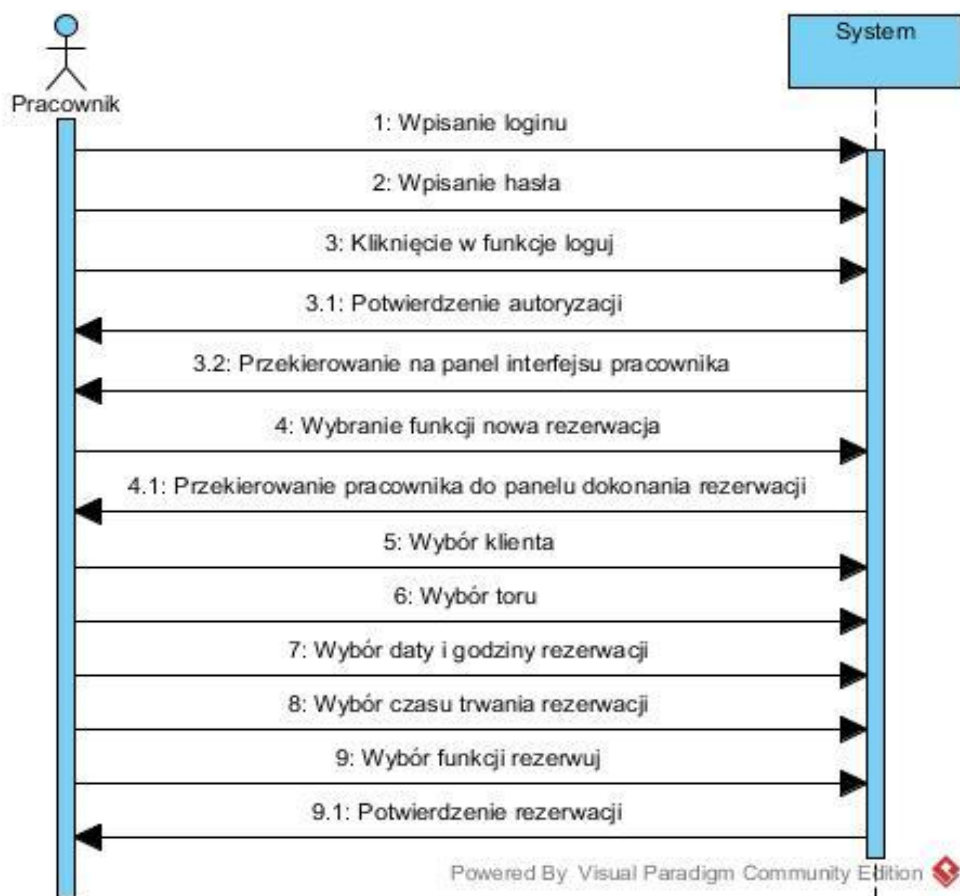
## 7. Diagram klas



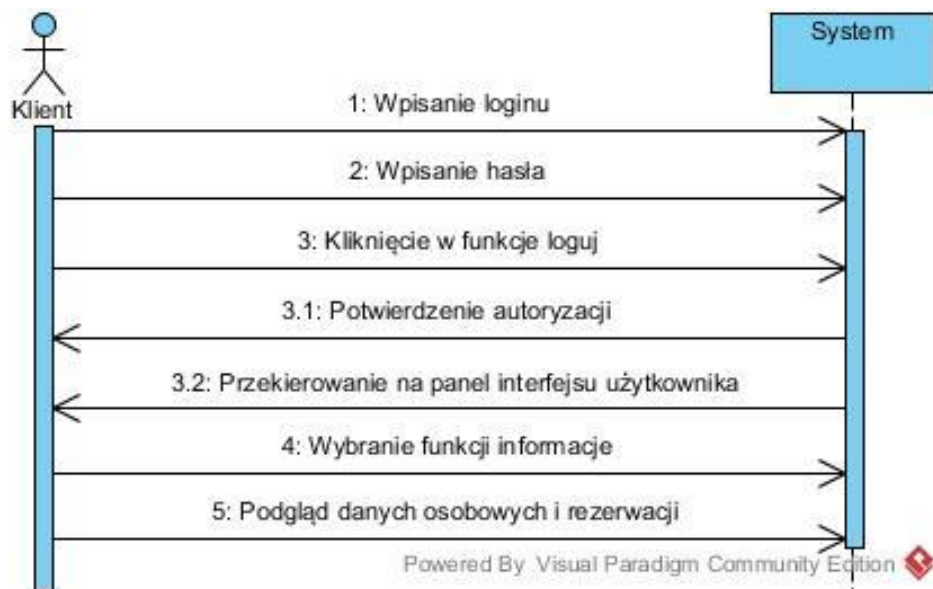
## 8. Diagramy sekwencji



Rys. Diagram przedstawia zarejestrowanie nowego pracownika przez administratora



Rys. Diagram przedstawia wprowadzenie rezerwacji do systemu przez pracownika



Rys. Diagram przedstawia sprawdzenie informacji odnośnie rezerwacji przez klienta

## **Rozkład pracy**

### **Łukasz Włodarczyk:**

- Mechanizm logowania
- Poszczególne poziomy uprawnień
- Mechanizm wysyłania e-maili
- Implementacja modelu
- Implementacja poszczególnych interfejsów i metod
- Inicjalizacja projektu i bazy danych
- Tworzenie poszczególnych komponentów
- Tłumaczenia

### **Jakub Puchalski:**

- Wygląd strony logowania
- Wygląd interfejsu użytkownika, pracownika, administratora
- Wygląd menu głównego
- Tworzenie grafiki do projektu
- Wygląd panelów funkcyjnych

### **Kuba Wyrzykowski :**

- Tworzenie dokumentacji
- Tworzenie diagramów
- Pomoc przy tworzeniu modelu
- Pomoc w planowaniu wyglądu strony
- Pomoc w tłumaczeniach