**Dokumentacja techniczna**

**Cel projektu:**

**Aplikacja do zgłaszania usterek mechanicznych w pojeździe samochodowym**

**Charakterystyka systemu**

**Analiza i specyfikacja wymagań**

**Analiza systemowa**

**Przed rozpoczęciem projektu grupa zapoznała się z jego celami. Ustalone ramy czasowe: do 16.06.2015r. Team Velocity – 5 osób w zespole, max. 6 godzin pracy na osobę w ciągu dnia oszacowane jako 80 pkt nakładu pracy na projekt sumarycznie.**

***Aplikacja webowa Usterki Mechaniczne***

***Użytkownicy – aktorzy role :***

1. ***Kierownik***
2. ***Logistyk (sekretarka)***
3. ***Księgowa***
4. ***Mechanik (nie więcej niż 10)***

***Wewnętrzny system dla pracowników serwisu mechanicznego, ma na starcie panel logowania - login, hasło szyfrowane. Jeden administrator – „kierownik” ma możliwość dodawania i edycji kont. Ma pełny podgląd aktywności pracowników. Logistyk posiada możliwość operowania trybem zgłaszania usterek. Mechanik (zwykły użytkownik) widzi po zalogowaniu swój harmonogram. Każdy z użytkowników ma wysublimowany kalendarz połączony z bazą danych pgSql. Ujmuje on dni oraz godziny pracy dla poszczególnych mechaników oraz obsługę klienta. Jest połączony z odpowiednią reakcją na zdarzenia na panelu. Szata graficzna ma w łatwy sposób sugerować dostępność w pracy mechaników (zajęte godziny wizytami klientów czerwone okienko, wolne zielone) oraz analogicznie pokazywać mechanikom na ich panelu dni wolne oraz robocze. Każdy z aktorów ma panel wyposażony w odpowiednią wyszukiwarkę klientów i usterek. Dodatkowo dostępna jest ogólna wyszukiwarka danych usług po dacie oraz obsłużonych pojazdach [nr rejestracyjne].***

**Synteza rozwiązań, cechy realizacji przedsięwzięcia, interakcje systemu ze światem zewnętrznym**

Mówiąc o systemie informatycznym mamy na myśli tak elementy sprzętowe systemu jak

i oprogramowanie systemowe oraz aplikacyjne.

Aplikacja działa na modelu MVC i serwisie RESTful. Model jest utrzymywany niezależnie od widoków, dzięki czemu w przyszłości można łatwo rozbudować aplikację, zupełnie zmieniając widoki.

***Język projektu:***

*Java*

***Wykorzystywana technologia:***

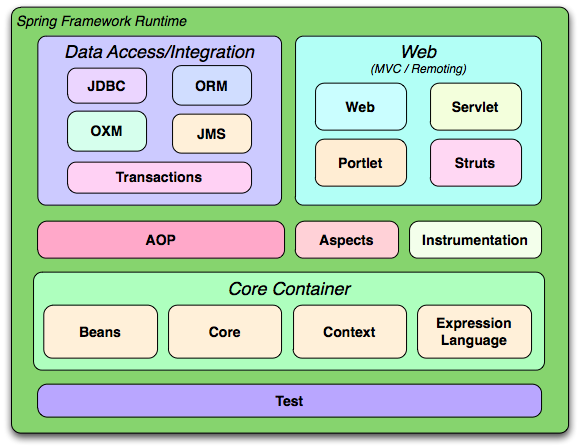
*- Java Spring Framework*

*- pgSql*

***Oprogramowanie:***

***Spring Tool Suite***

Spring Framework jest to platforma, której głównym celem jest uproszczenie procesu tworzenia oprogramowania klasy enterprise w technologii Java/J2EE. Rdzeniem Springa jest kontener wstrzykiwania zależności, który zarządza komponentami i ich zależnościami. Korzystamy z wersji SPRING BOOT SECURITY (m.in. ze względu na możliwość użycia specjalnego pola roles które jest bardzo funkcjonalne przy różnych uprawnieniach dla różnych kont).



Rysunek 1: Architektura Spring Framework, *źródło: Spring Framework Reference*

Spring jest rozwiązaniem modułowym. Szczególne części, które wykorzystujemy:

#### CoreContainer

**Core** i **Beans** – podstawowe moduły, zawierają funkcjonalność Inversion of Control i DependencyInjection. To dzięki nim możliwe jest oddzielenie konfiguracji i specyfikacji zależności od logiki biznesowej.

#### DAO

### **DAO** *Data Access Object* zapewnia wsparcie dla metod utrwalania obiektów, w szczególności odwzorowanie relacyjno-obiektowe ORM,JDBC, **OXM** (mapowanie XML), JMS (tworzenie i przetwarzanie wiadomości). Dostarcza gotową do wykorzystania pulę połączeń, a także możliwość deklaratywnego definiowania transakcji. Pozwala na łatwe mapowanie ResultSetów na listę obiektów klas domenowych.

#### Web

**Web** zawiera własny framework webowy – Spring Web MVC, który wspomaga proces ładowania plików na serwer.

#### Test

**Test** zawiera mechanizmy służące do testowania aplikacji (JUnit).

Inżynierię oprogramowania można podzielić na kilka zależnych od siebie warstw: narzędzia, metody , proces wytwórczy, dbanie o jakość.

***W projekcie trzymamy się zasad:***

* ***Solid***
* [Zasada jednej odpowiedzialności](http://pl.wikipedia.org/wiki/Zasada_jednej_odpowiedzialno%C5%9Bci) (S)
* [Zasada otwarte-zamknięte](http://pl.wikipedia.org/wiki/Zasada_otwarte-zamkni%C4%99te) (O)
* [Zasada podstawienia Liskov](http://pl.wikipedia.org/wiki/Zasada_podstawienia_Liskov) (L)
* [Zasada segregacji interfejsów](http://pl.wikipedia.org/w/index.php?title=Zasada_segregacji_interfejs%C3%B3w&action=edit&redlink=1) (I)
* [Zasada odwrócenia zależności](http://pl.wikipedia.org/w/index.php?title=Zasada_odwr%C3%B3cenia_zale%C5%BCno%C5%9Bci&action=edit&redlink=1) (D)
* ***DRY***

Don’t Repeat Yourself – reguła zalecająca unikanie różnego rodzaju powtórzeń wykonywanych przez programistów - na przykład unikanie tych samych czynności podczas kompilowania, unikanie wklejania (lub pisania) tych samych (lub bardzo podobnych) fragmentów kodu w wielu miejscach

* ***KISS***

Keep It Simple, Stupid– zasada utrzymania projektu w sposób prosty i zrozumiały dla każdego, jej istotą jest dążenie do utrzymania eleganckiej i przejrzystej [struktury](http://pl.wikipedia.org/wiki/Struktura)

* **CRUD**

***C****reate,* ***R****ead,* ***U****pdate and* ***D****elete* (pol. *utwórz, odczytaj, aktualizuj i usuń*) - cztery podstawowe funkcje w aplikacjach korzystających z pamięci trwałej, które umożliwiają zarządzanie nią

***metod:***

* ***Refactoringu***
* ***Continous Integration.***

***Korzystamy z metodyki projektowania systemów informatycznych Scrum przechodząc wg modelu iteracyjnego przez poszczególne fazy projektowania:***

1. ***Analiza wymagań użytkownika i modelowanie systemu.***
2. ***Projektowanie systemu (zgodnie z rozpiską sprintów).***
3. ***Implementacja systemu.***
4. ***Testowanie systemu.***
5. ***Pielęgnacja i dalszy rozwój.***

**Kierownik**

|  |
| --- |
| Dodawanie kont wszystkich pracowników |
| Edycja kont, ustawianie haseł i loginów pracowników |
| Ustawia: imię, nazwisko, stanowisko, godziny pracy, urlopy, dostępność |
| Może podglądnąć wszystkie operacje wykonywane przez każdego pracownika |
| Widzi wszystkie opcje z każdego konta |
| Po wejściu na stronę jest panel logowania i wyniki zależne od stanowiska |

**Logistycy (sekretarki)**

|  |
| --- |
| Po zalogowaniu widzi 2 tryby |
| 1. tryb- dane: imię, nazwisko, stanowisko, kalendarz z godzinami pracy, urlopami |
| 2.tryb- zgłaszanie danej awarii : |
| Wyszukiwarka klientów po nazwisku i nr. rejestracyjnym pojazdu |
| Gdy jest klient, z jego konta ustala się nową wizytę, gdy nie jest, dodaje się go |
| Z konta klienta ustala nową wizytę (pole tekstowe) |
| Wybiera się z kalendarza dzień, w którym ma być wizyta |
| W wybranym dniu jest lista mechaników (Zielony-wolny, Czerwony-Zajęty) |
| Wybiera się mechanika, ten automatycznie w bazie robi się Czerwony i akceptuje wizytę |

**Mechanicy**

|  |
| --- |
| Widzi harmonogram (kalendarz), dni kiedy ma wolne (czerwone) i kiedy ma być w pracy(zielone) |
| Zielony dzień jest klikalny, widzi tam rozpiskę godzinową oraz: |
| \*nieedytowalne pole z notatkami sekretarki |
| \*pola do uzupełnienia: cena swojej naprawy, cena za części, textbox- co zrobił |
| \*Dwa przyciski: Zapisz i Nie udało się wykonać naprawy |

**Księgowe**

|  |
| --- |
| Dokładna wyszukiwarka usług, sortująca po: |
| Nazwisku, nr. rejestracyjnym, po cenie usługi, za części i sumie, operatory: <>= |
| Jak wyszuka dane, pokazuje jej się opcja zaznaczenia tych znalezionych osób |
| Po zaznaczeniu uaktywnia się przyciski: Zapisz do pdf i zapisuje się na komputerze |

**Wszyscy**

|  |
| --- |
| Każdy ma zakładkę z wyszukiwarką wykonanych usług i może szukać po: |
| dacie (od -do), nr. rejestracyjnym, nazwisku klienta, nazwisku mechanika (Można łączyć) |

**Sprawy techniczne**

|  |
| --- |
| Stworzenie panelu logowania |
| Stworzenie tabel w bazie danych |
| Stworzenie bazy danych, podpięcie jej |

**User Stories**

* Jako niezalogowany użytkownik mogę wyświetlić panel logowania.
* Po zalogowaniu się do systemu widzę swój panel użytkownika.
* Jako administrator mogę przejść do panelu dodawania nowych użytkowników, aby zarejestrować konta pracowników.
* Po podaniu odpowiednich danych mogę dodać nowego użytkownika lub anulować dodawanie pracownika.
* Jako dowolny aktor widzę na panelu po zalogowaniu odpowiednio swój kalendarz
* Jako sekretarka mogę wyszukiwać klientów, dzięki temu mogę dokonać weryfikacji czy dana osoba jest już klientem w bazie
* Jako administrator mogę dodawać godziny pracy mechanika
* Jako dowolny aktor nie mogę zobaczyć panelu innego aktora, dzięki temu zapewnione jest bezpieczeństwo systemu.
* Jako mechanik mogę wyświetlić swój panel użytkownika, po zalogowaniu się.
* Jako mechanik mogę wyświetlić swój kalendarz pracy i sprawdzić grafik na dany dzień.
* Jako mechanik mogę wprowadzić dane dotyczące wykonanej naprawy i zatwierdzić zmiany.

**Scenariusze testowe**

**#D1 Loguję użytkownika**

Przypadek użycia zaczyna się gdy wyświetlam witrynę systemu Car Repair. Widzę panel logowania. Wpisuję login i hasło użytkownika. Następuje przekierowanie na moje konto. Przypadek użycia kończy się.

Nadzwyczajny ciąg zdarzeń: podanie złego loginu lub hasła bądź nie podanie jednego z pól – zalogowanie użytkownika nie nastąpi.

**#D2 Dodaję nowego użytkownika**

Przypadek użycia zaczyna się gdy przechodzę do panelu dodawania nowych użytkowników jako administrator. Podaję dane. Po ich zatwierdzeniu przypadek użycia kończy się.

Nadzwyczajny ciąg zdarzeń: anulowanie dodawania pracownika, nie podanie wymaganych danych, nie zaznaczenie rodzaju konta – przypadek użycia kończy się nie dodaniem nowego konta i wyświetleniem komunikatu błędu. Podaniem złej formy imienia, nazwiska lub hasła- komunikat błędu z opisanymi wymaganiami i nie dodanie nowego konta.

**#D3 Dodaję klienta**  
  
Przypadek użycia zaczyna się, gdy wchodzę w wyszukiwarkę sekretarki. Podaję dane nowego klienta i zatwierdzam. Po zatwierdzeniu w bazie, jak i w wyszukiwarce klientów, pojawia się nowo dodana osoba. Przypadek użycia kończy się.  
  
Nadzwyczajny ciąg zdarzeń: Podany klient już istnieje- konflikt z kluczem głównym, którym jest numer tablicy rejestracyjnej, nie podanie którejkolwiek z danych bądź zakończenie przyciskiem anulowania. Przypadek użycia kończy się nie dodaniem klienta do bazy i wyświetleniem stosownego komunikatu.

**#D4 Wyszukuję klienta**

Przypadek użycia zaczyna się, gdy wchodzę w wyszukiwarkę sekretarki. Podaję dane do wyszukania: nazwisko lub nr. rejestracyjny pojazdu. Po ich zatwierdzeniu otrzymuję klientów z bazy spełniających kryteria. Przypadek użycia kończy się.

Nadzwyczajny ciąg zdarzeń: brak klientów spełniających podane kryteria w bazie.

**#D5 Zaznaczam godziny pracy**

Przypadek użycia zaczyna się, gdy zalogowany jako kierownik wybieram z listy rozwijanej pracownika, dla którego chcę ustalić harmonogram dnia pracy. Następnie klikam na ikonę kalendarza i wybieram dzień. Wybieram z listy godziny pracy. Przypadek użycia kończy się po zapisaniu zmian.

**#D6 Dodaję nowe zlecenie**

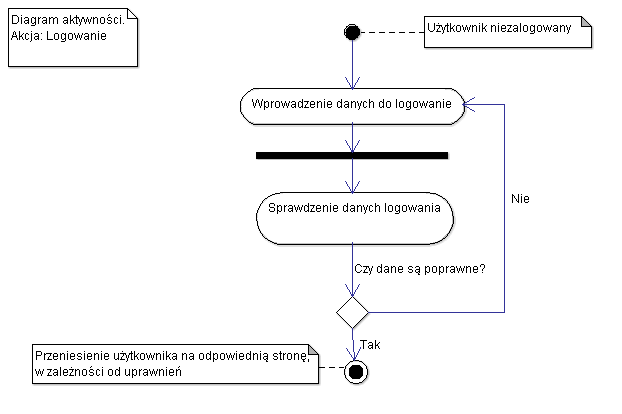
Przypadek użycia zaczyna się po zalogowaniu na panel sekretarki Podaję nazwę i opis zadania do wykonania, wybieram mechanika. Następnie wybieram dzień wykonania z kalendarza. Szacuję ilość godzin na zlecenie. Przypadek użycia kończy się po kliknięciu przycisku ‘Dodaj’.

**#D7 Generuję raport w formacie PDF**

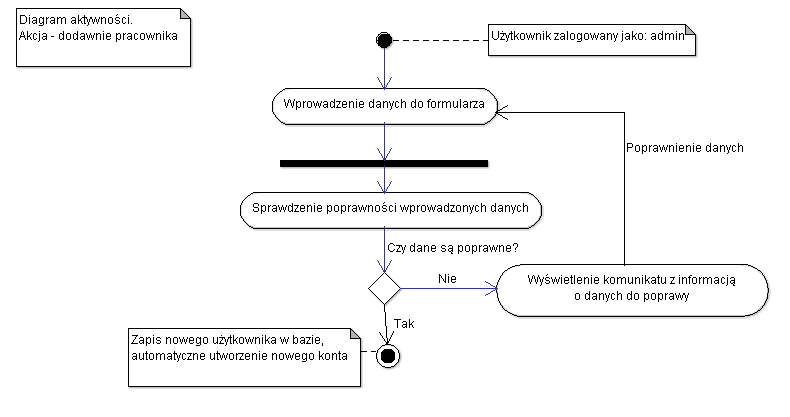
Przypadek użycia zaczyna się, gdy zalogowany jako księgowa klikam przycisk „Pobierz raport”. W zależności od przeglądarki internetowej może zostać uruchomione okno dialogowe w celu wybrania lokalizacji na dysku do zapisu raportu, lub raport zostanie automatycznie zapisany w domyślnej lokalizacji. Przypadek użycia kończy się po zapisaniu dokumentu w formacie PDF.

**Diagramy**

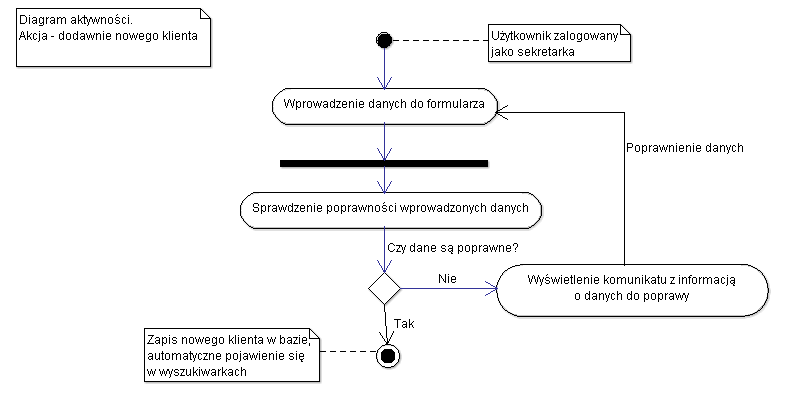
**#D1 Loguję użytkownika**

****

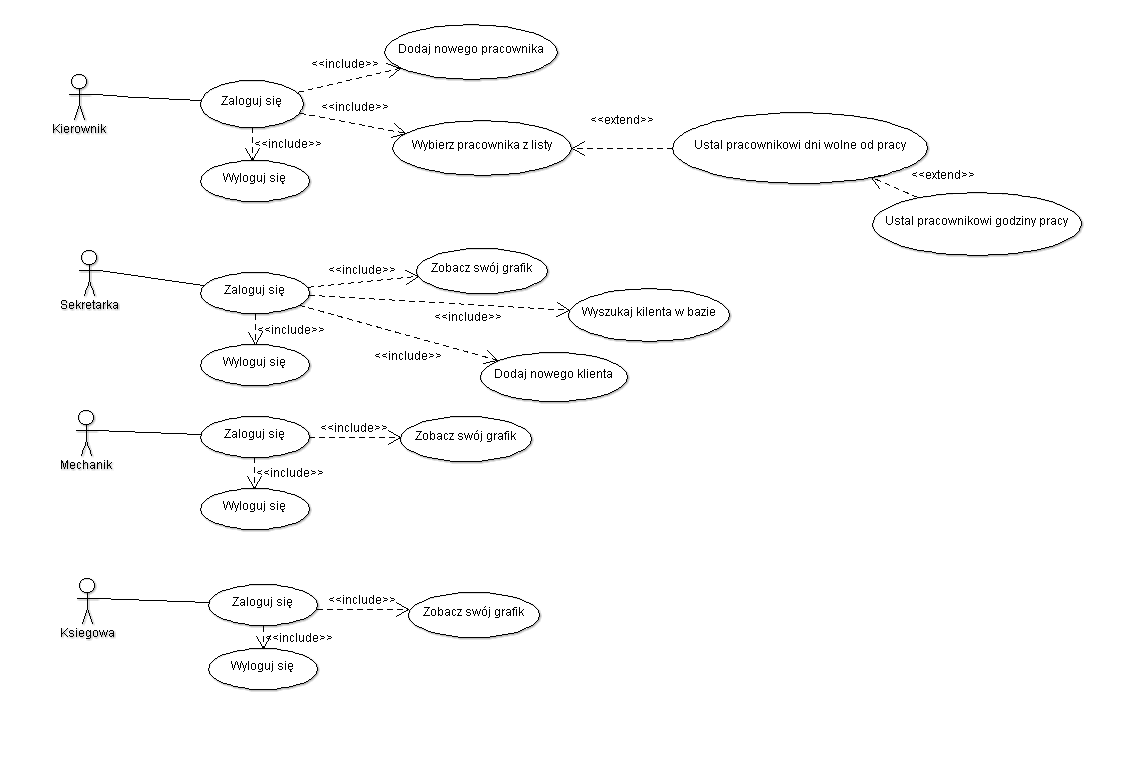
**#D2 Dodaję nowego użytkownika**

****

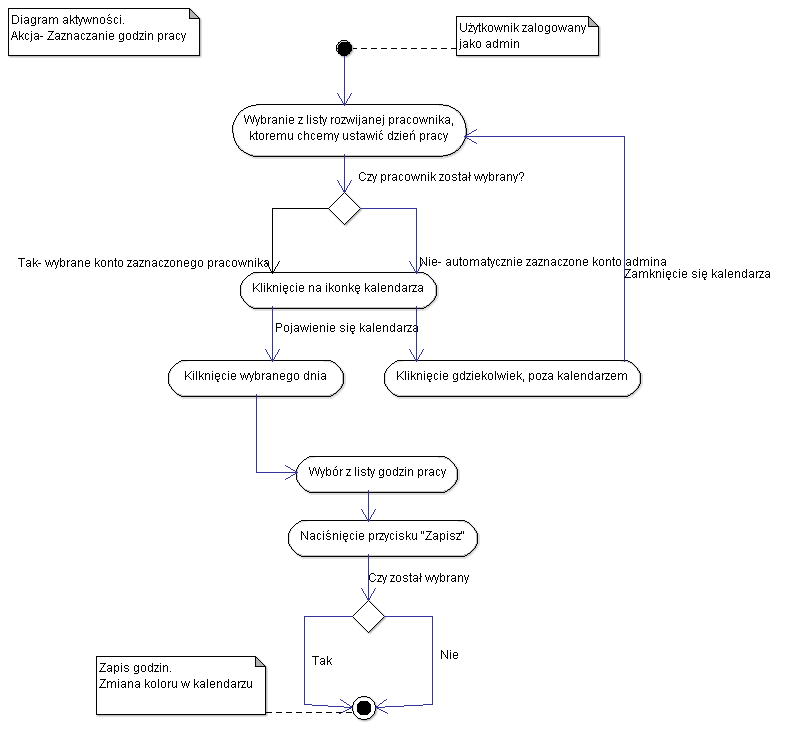
**#D3 Dodaję klienta**

****

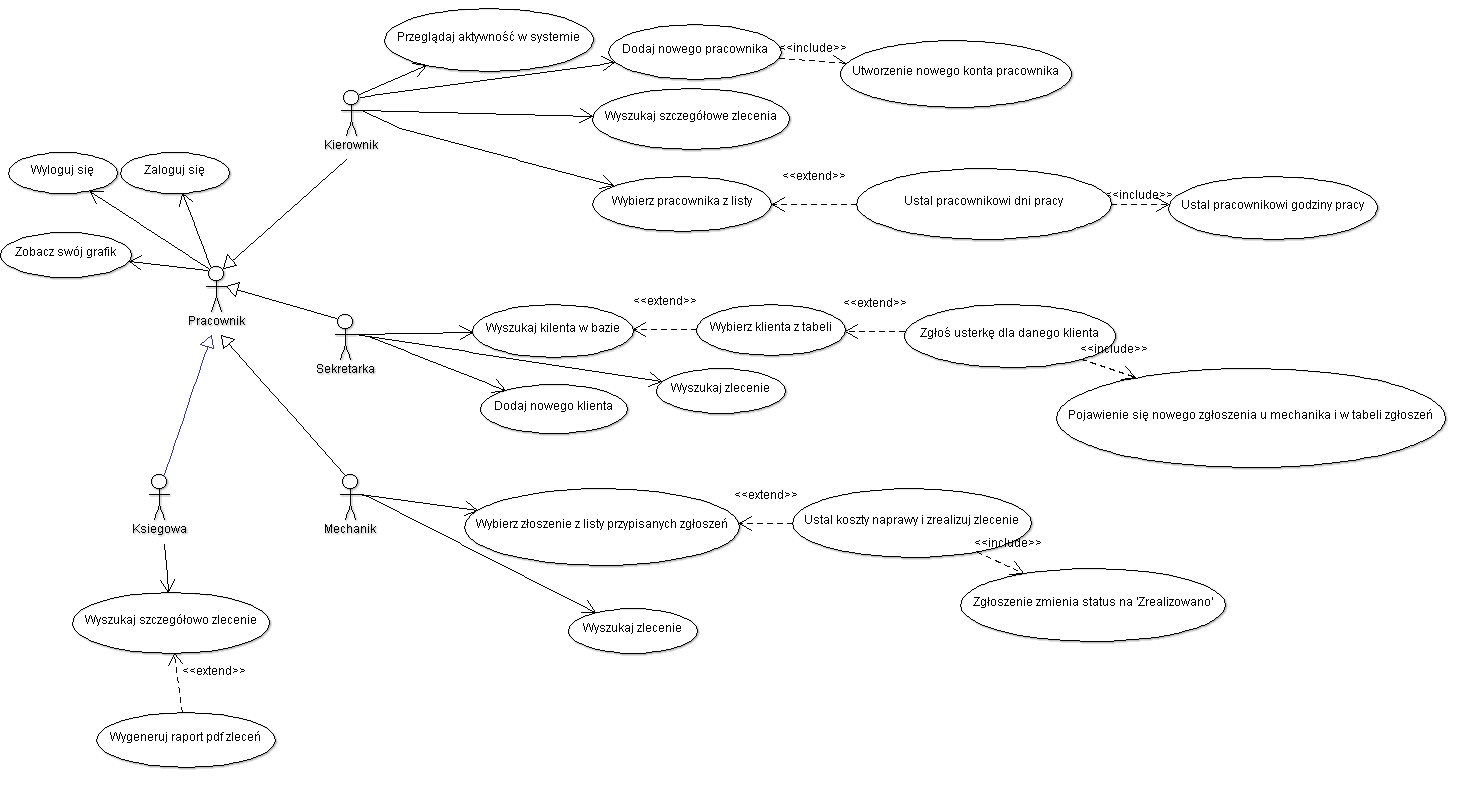
**Use Case (#UC1, #UC2, #UC3, #UC4 )**

****

**#D5 Zaznaczam godziny pracy**

****

**Use Case**

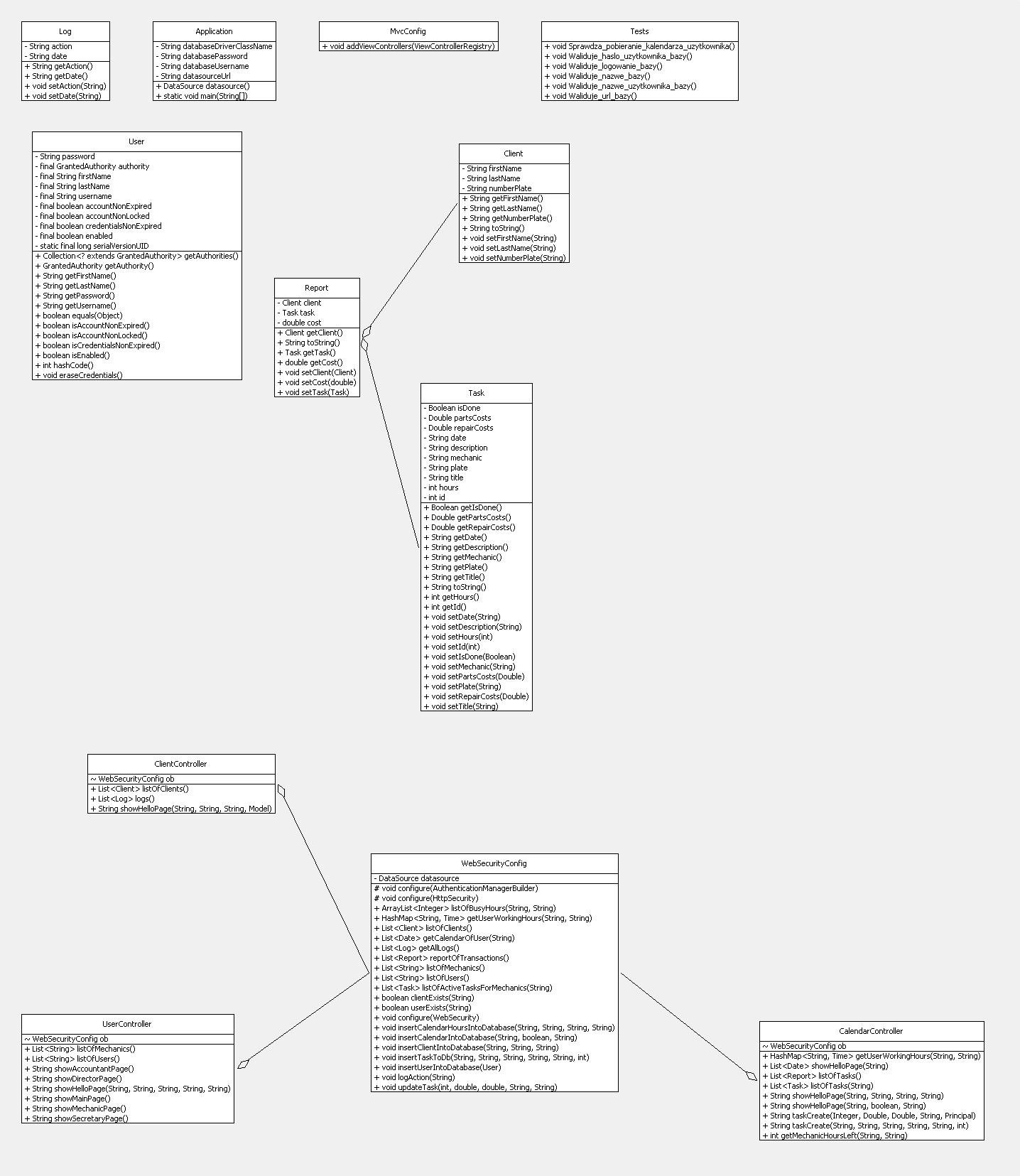
****

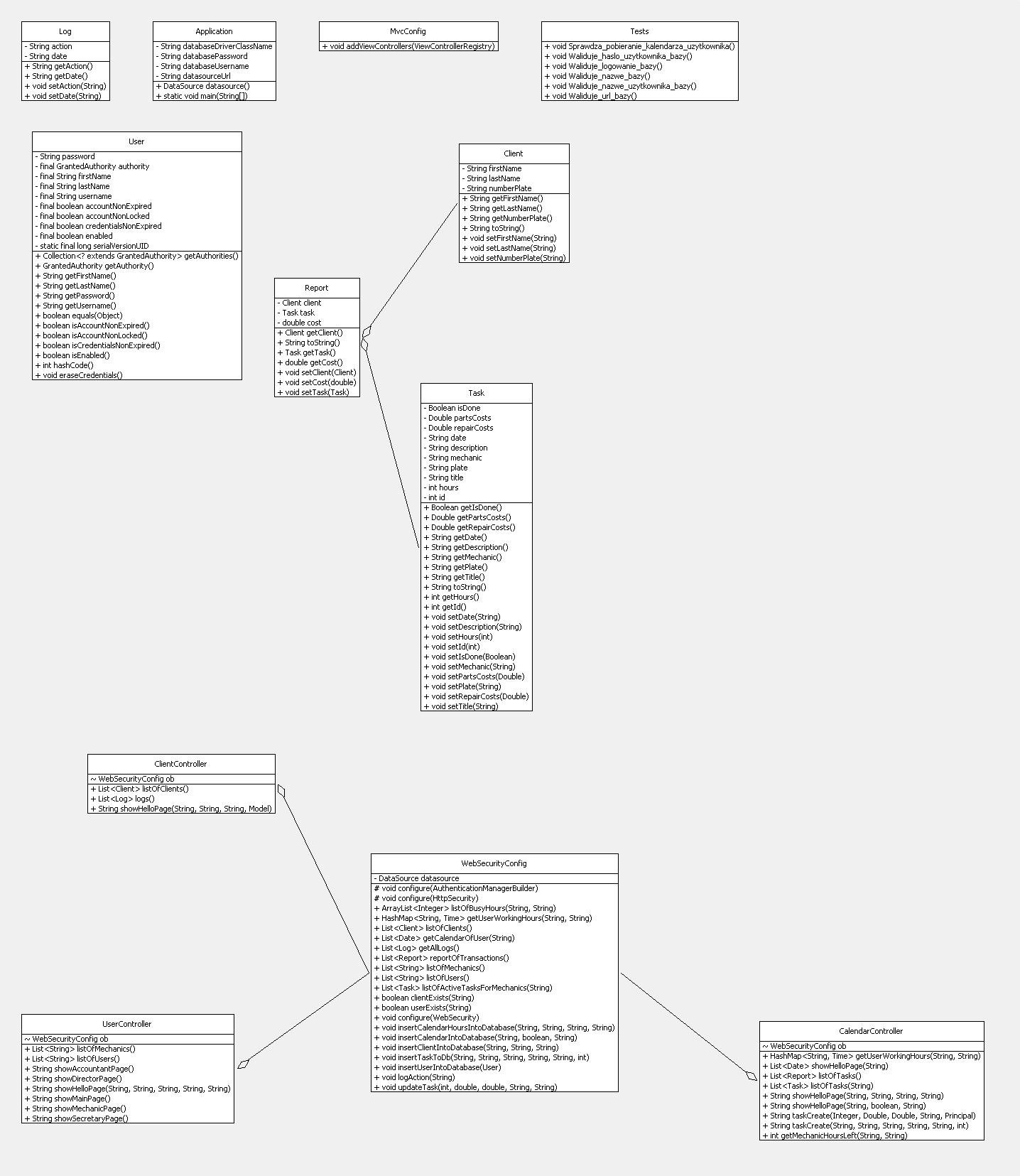
**Model obiektów**

Przez sposób zachowania obiektu rozumiemy operacje jakie mogą być dokonane na nim przez inne obiekty, operacje jakie może on wykonywać na innych obiektach oraz konsekwencje tych operacji czyli zmiany stanu obiektu i obiektów, które były z nim w interakcji. Często obiekty w celu wykonania operacji informują się wzajemnie o takiej potrzebie poprzez mechanizm tzw. przesyłania komunikatu. Relacje pomiędzy klasami tzw. relacji dziedziczenia czyli wyrażenia faktu, iż jedna klasa jest uogólnieniem innej (lub innych) oraz tzw. polimorfizmu (mieć wiele postaci).

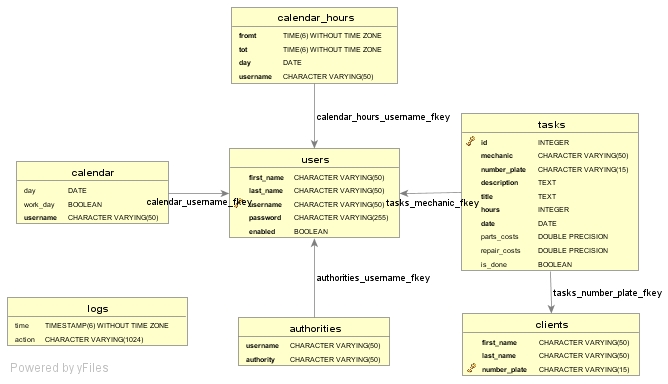
**Model obiektów jest reprezentowany graficznie za pomocą diagramów (układ hierarchiczny z więzami) składających się z klas tworzących hierarchie oraz połączonych ze sobą wiązaniami. Na tym modelem buduje się pozostałe dwa: dynamiczny i funkcjonalny.**

**Diagram klas**

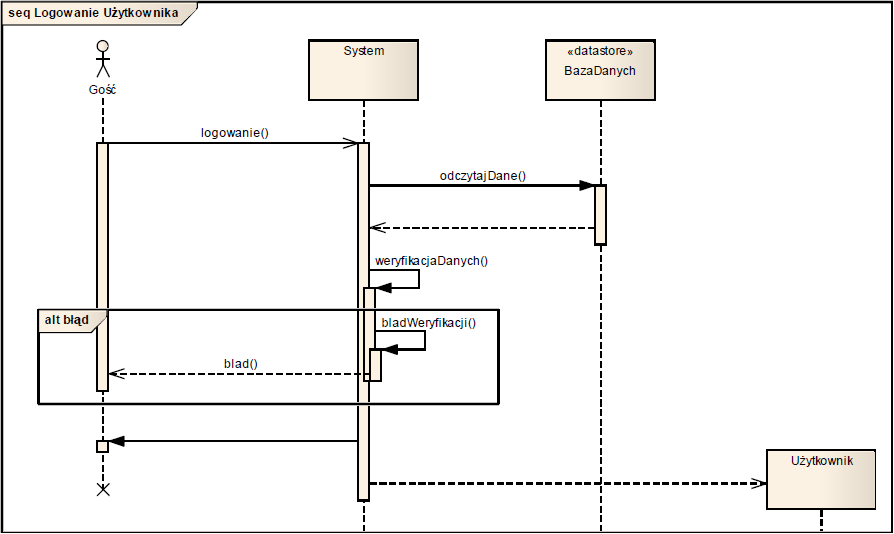
****

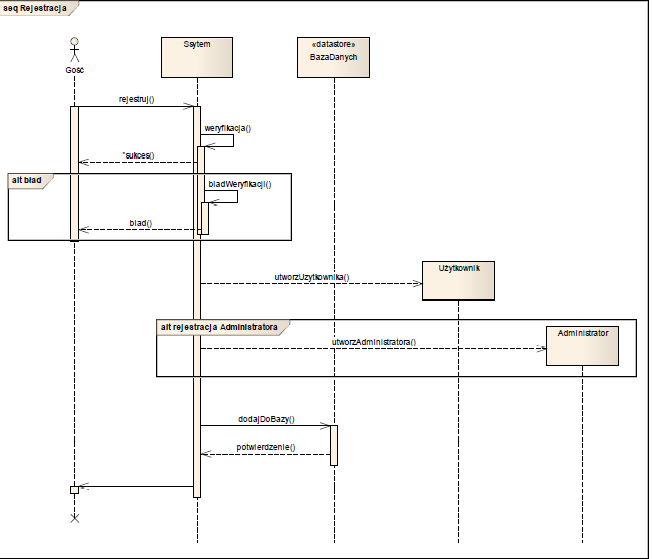
****

**Diagram bazy danych**

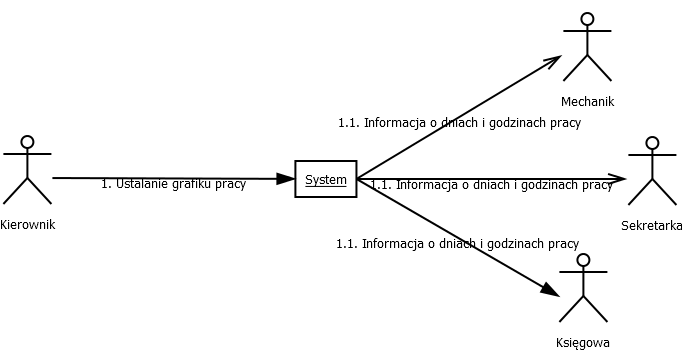
****

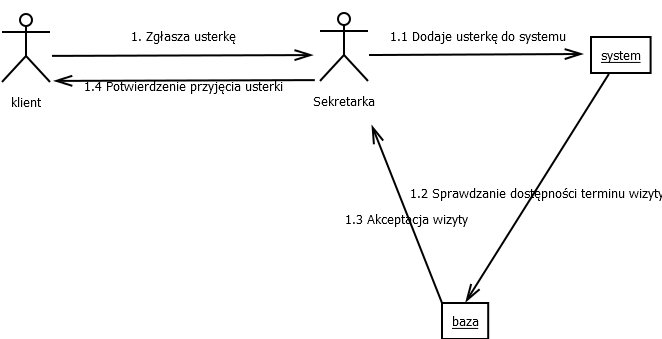
**Model dynamiczny – diagramy sekwencji**

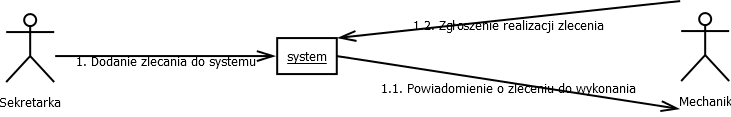
****

****

**Diagramy komunikacji**

****

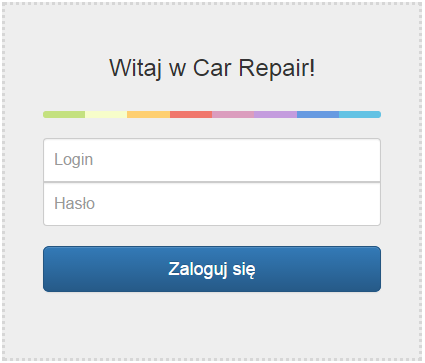
****

****

**Obsługa aplikacji**

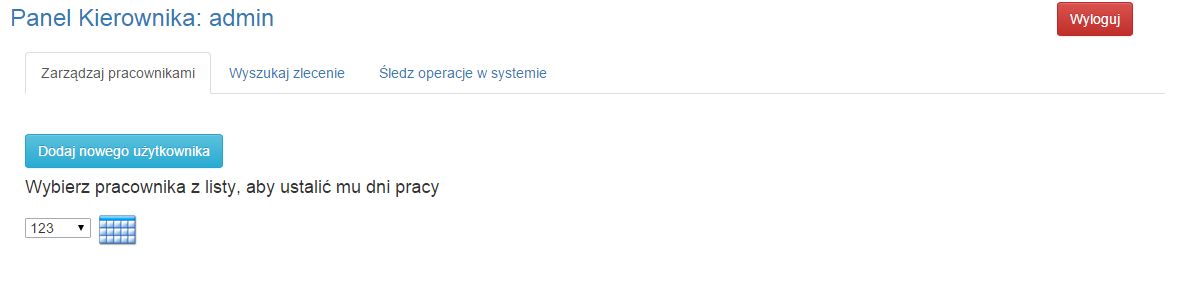
W celu uruchomienia aplikacji wymagana jest przeglądarka internetowa z zainstalowanym dodatkiem Java.

Po uruchomieniu wyświetla się Panel logowania, wyglądający następująco:



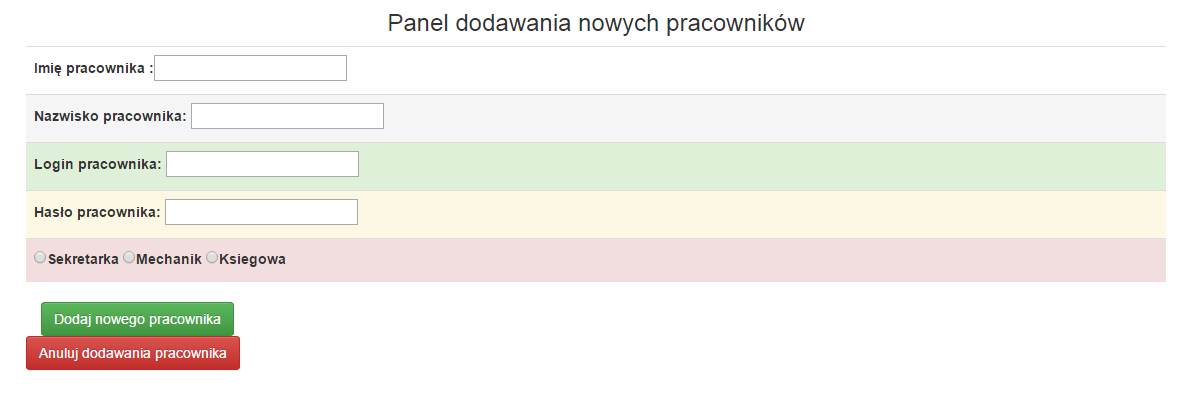
W celu zalogowania należy wprowadzić login i hasło oraz kliknąć przycisk „Zaloguj się”.

Panel Kierownika wygląda następująco:



Aby się wylogować należy kliknąć przycisk „Wyloguj”, znajdujący się w prawym górnym rogu aplikacji.

Kierownik ma możliwość dodawania nowych pracowników. Aby tego dokonać należy kliknąć przycisk „Dodaj nowego użytkownika” Po kliknięciu wyświetli się Panel dodawania nowych pracowników:



Należy wprowadzić dane takie jak Imię, Nazwisko pracownika, unikalny login, hasło, oraz określić stanowisko, na którym dany użytkownik jest zatrudniony: Sekretarka, Mechanik, Księgowa.

W celu dodania pracownika do bazy należy kliknąć przycisk „Dodaj nowego użytkownika”.

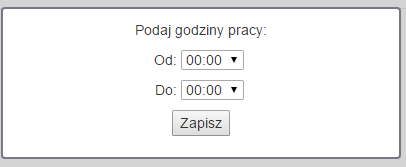
W każdej chwili można anulować dodawania pracownika. Aby tego dokonać należy kliknąć przycisk „Anuluj dodawania pracownika”.



Kierownik ma również możliwość ustalania dni, oraz godzin pracy dla każdego pracownika.

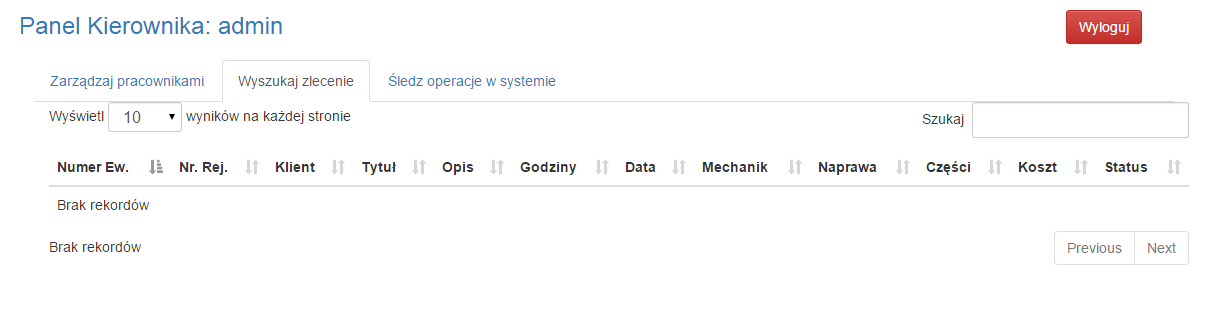
W tym celu należy z rozwijanej listy wybrać pracownika, następnie kliknąć na ikonę kalendarza, znajdującą się po prawej stronie listy, aby otworzyć kalendarz.

Po kliknięciu na wybrany dzień wyświetli się okno do wprowadzenia godzin pracy, wyglądające następująco:



Należy z list rozwijanych wybrać godziny pracy i w celu zapisania zmian, kliknąć przycisk „Zapisz”.

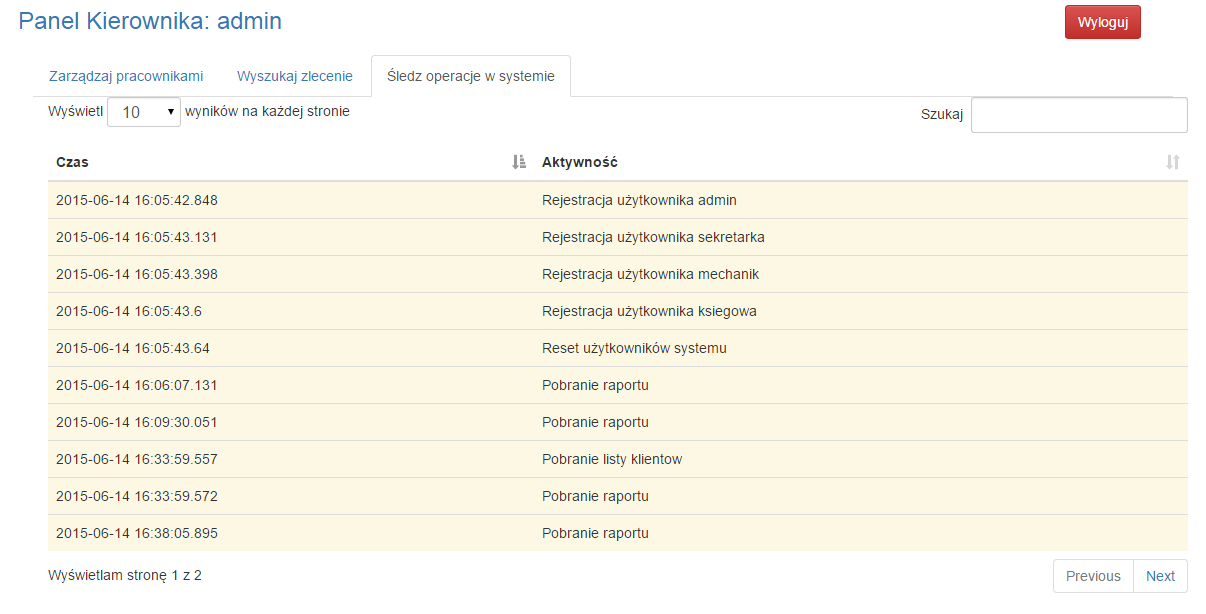
W zakładce „Wyszukaj zlecenie” można wyszukiwać konkretne zlecenia.



W polu szukaj należy wpisać dane do wyszukania. Przyciski Previous/Next służą do nawigacji pomiędzy stronami.

Można wybrać ilość rekordów wyświetlanych na stronie. Służy do tego rozwijana lista.

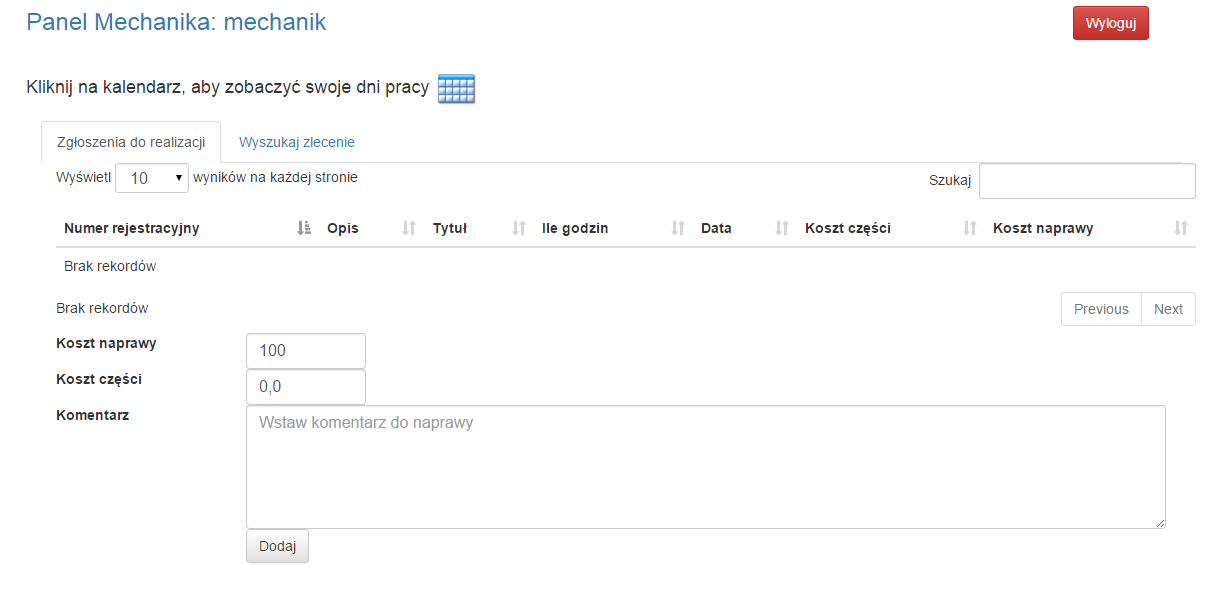
W zakładce „Śledź operacje w systemie” można zobaczyć historię zmian.



W polu szukaj należy wpisać dane do wyszukania. Przyciski Previous/Next służą do nawigacji pomiędzy stronami.

Można wybrać ilość rekordów wyświetlanych na stronie. Służy do tego rozwijana lista.

Panel Mechanika wygląda następująco:

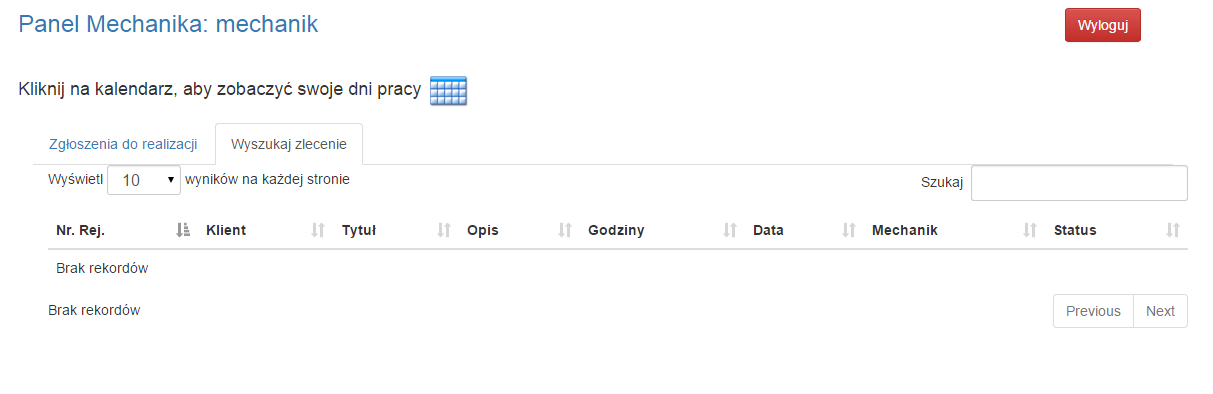


Aby się wylogować należy kliknąć przycisk „Wyloguj”, znajdujący się w prawym górnym roku aplikacji.

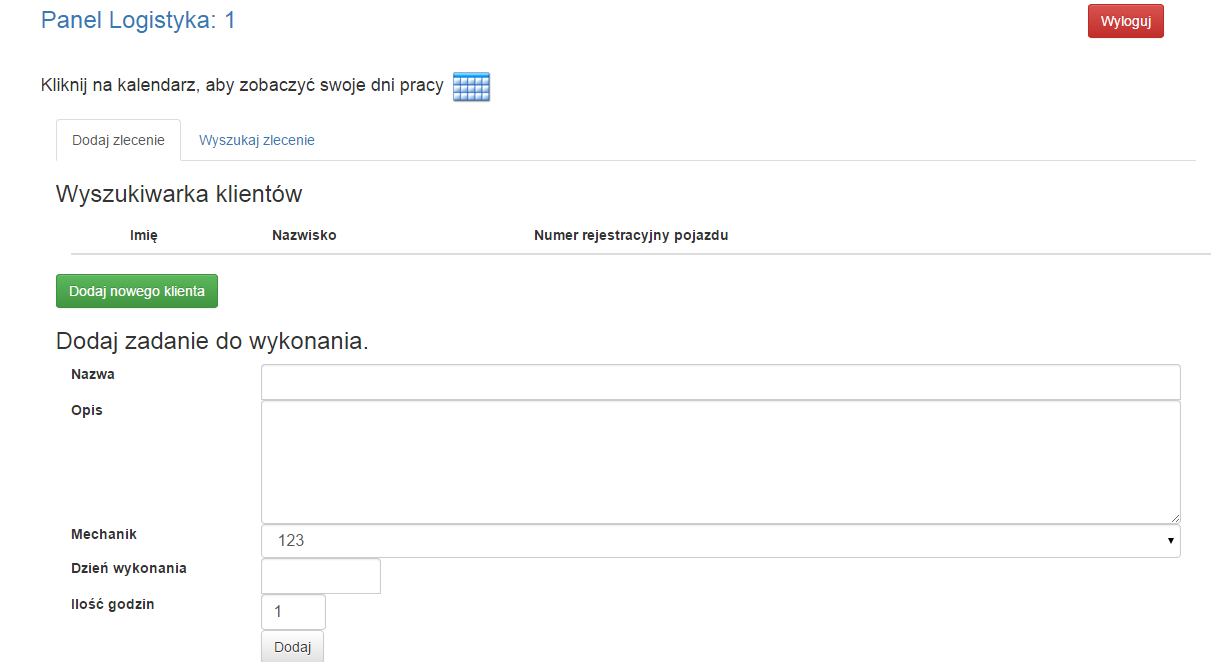
Po kliknięciu w ikonę kalendarza wyświetla się okno kalendarza, analogicznie jak w panelu kierownika. Mechanik ma możliwość sprawdzenia dni pracy oraz godzin.

W formularzu mechanik wprowadza dane dotyczące wykonanej naprawy. Należy wprowadzić koszt naprawy, koszt części użytych do wykonania naprawy, umieścić stosowny komentarz odnośnie naprawy. W celu zapisania zmian należy, po uprzednim wypełnieniu wszystkich danych, kliknąć przycisk „Dodaj”.

W zakładce „Wyszukaj zlecenie” można wyszukiwać konkretne zlecenia.



Panel logistyka wygląda następująco:

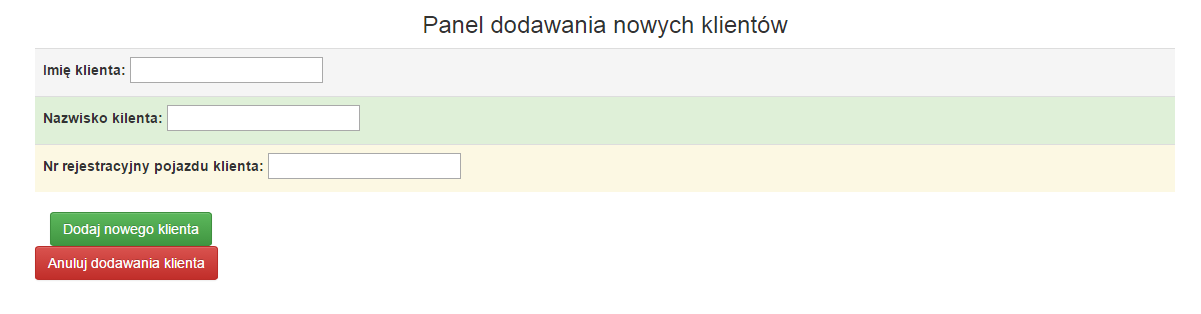


Aby się wylogować należy kliknąć przycisk „Wyloguj”, znajdujący się w prawym górnym roku aplikacji.

Po kliknięciu w ikonę kalendarza wyświetla się okno kalendarza, analogicznie jak w panelu kierownika. Logistyk ma możliwość sprawdzenia dni pracy oraz godzin.

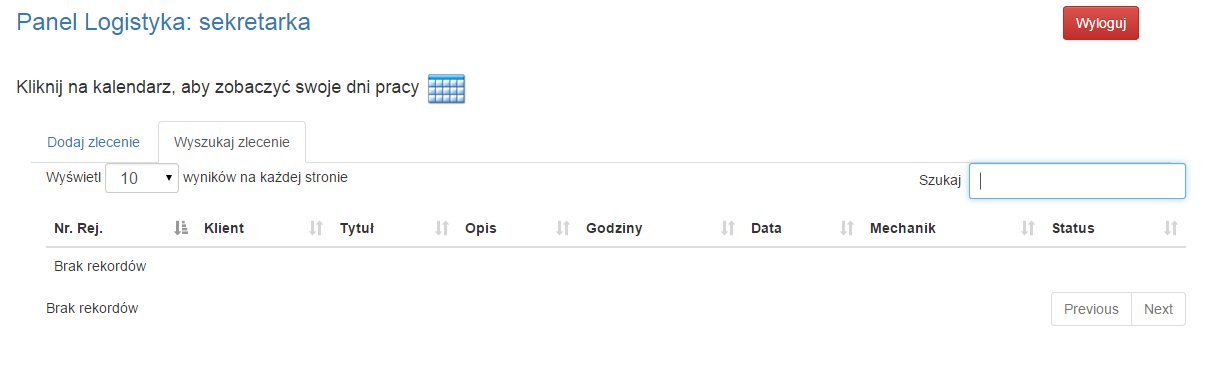
Logistyk może Dodawać zadanie do wykonania. Aby to zrobić należy wypełnić pola formularza znajdujące się na jego panelu. Należy podać nazwę zadania, opis, wybrać mechanika, który ma to zadanie wykonać, dzień wykonania zadania oraz oszacować czas potrzebny na jego wykonanie. Aby zadanie zostało dodawanie do bazy należy kliknąć przycisk „Dodaj”.

Logistyk może dodawać nowych klientów. Aby tego dokonać należy kliknąć przycisk „Dodaj nowego klienta”



Należy wprowadzić imię i nazwisko klienta oraz numer rejestracyjny jego pojazdu. Aby zapisać zmiany należy kliknąć przycisk „Dodaj nowego klienta”. Można przerwać dodawanie nowego klienta klikając przycisk „Anuluj dodawanie klienta”.

W zakładce „Wyszukaj zlecenie” można wyszukiwać konkretne zlecenia.

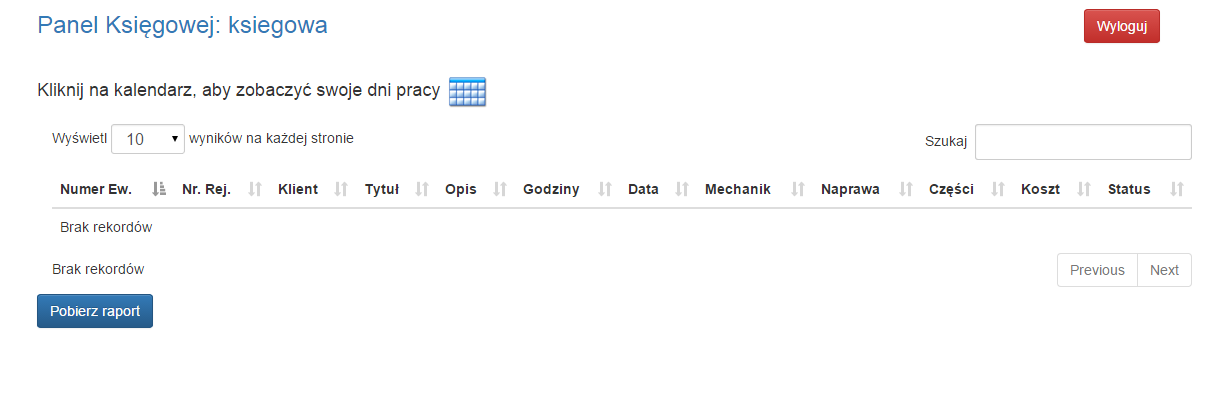


W polu szukaj należy wpisać dane do wyszukania. Przyciski Previous/Next służą do nawigacji pomiędzy stronami.

Można wybrać ilość rekordów wyświetlanych na stronie. Służy do tego rozwijana lista.

Aby posortować rekordy należy kliknąć ikonki strzałek przy interesującej danej.

Panel księgowej wygląda następująco:



Aby się wylogować należy kliknąć przycisk „Wyloguj”, znajdujący się w prawym górnym roku aplikacji.

Po kliknięciu w ikonę kalendarza wyświetla się okno kalendarza, analogicznie jak w panelu kierownika. Księgowa ma możliwość sprawdzenia dni pracy oraz godzin.

Księgowa może wyszukiwać konkretne rekordy. Są to zgłoszone zadania, koszty oraz informacja o statusie wykonania.

W polu szukaj należy wpisać dane do wyszukania. Przyciski Previous/Next służą do nawigacji pomiędzy stronami.

Można wybrać ilość rekordów wyświetlanych na stronie. Służy do tego rozwijana lista.

Aby posortować rekordy należy kliknąć ikonki strzałek przy interesującej danej.

Księgowa może wygenerować raport przedstawiający dane. W celu wykonania raportu należy kliknąć przycisk „Pobierz raport”. Raport zostanie wygenerowany w formacie PDF. W zależności od przeglądarki internetowej może zostać uruchomione okno dialogowe w celu wybrania lokalizacji na dysku do zapisu raportu, lub raport zostanie automatycznie zapisany w domyślnej lokalizacji. Do poprawnego otwarcia raportu wymagany jest zainstalowany program obsługujący odczyt plików w formacie PDF, np.: Adobe Reader.

**Backlog produktu**

**(w ramach stosowania lekkiej metodyki wytwarzania Agile)**

**Kontrola zapewnienia jakości**

Projekt jest efektem racy nad zestawem zadań podlegających zarządzaniu, połączonych wspólnym celem osiąganym w ramach zadanych ograniczeń.

Według ustalonego Definition of Done – stan w którym akceptujemy zrealizowany scenariusz w formie tasku:

Testy akceptacyjne kończą się sukcesem.

Kryteria akceptacyjne zostały spełnione.

Zastosowane wzorce są adekwatne do specyfiki systemu informatycznego, nie występują konflikty projektowe.

Responsywność systemu na poziomie zgodnym z wymaganiami.

Testy funkcjonalne wg ustalonych scenariuszy testowych kończą się zgodnie z oczekiwanymi rezultatami.

W przypadku warunków początkowych nie przewidzianych, program powinien zachować się w sposób przewidywalny.

Pozytywne wyniki testowania systemu są podstawą do odbioru oprogramowania konstruowanego w trakcie projektu.

**Spis treści:**

1. Sformalizowanie zbioru wymagań stawianych przed systemem (*str.3*)
2. Zdefiniowanie funkcjonalnego modelu systemu

-Synteza rozwiązań, cechy realizacji przedsięwzięcia, interakcje systemu ze światem zewnętrznym (*str.4*)

1. Konstrukcja modelu wymagań użytkownika

-User Stories (*str.8*)

-Scenariusze testowe (+adekwatne diagramy akcji, Use Case) (*str.9*)

1. Pełny diagram Use Case (*str.14*)
2. Model obiektów (*str.15*)

-diagram klas (model fizyczny systemu)

-diagram bazy danych

1. Model dynamiczny – diagramy sekwencji (*str.17*)
2. Diagramy komunikacji (*str.19*)
3. Obsługa aplikacji, oczekiwane rezultaty działań (*str.20*)
4. Backlog (*str.27*)
5. Kontrola zapewnienia jakości (*str.28*)