**W A R S Z A W S K A  
W Y Ż S Z A S Z K O Ł A I N F O R M A T Y K I**

PRACA DYPLOMOWA

STUDIA PIERWSZEGO STOPNIA

Przemysław Foltyn

Numer albumu 7134

ANALIZA, PROJEKT I IMPLEMENTACJA SYSTEMU WSPIERAJĄCEGO WYBRANE FUNKCJE ZARZĄDZANIA PRACĄ W ORGANIZACJI PRODUKCYJNEJ

Promotor:

mgr inż. Jerzy Stankiewicz

*Praca spełnia wymagania stawiane pracom dyplomowym na studiach pierwszego stopnia.*

W A R S Z A W A 2018

**Wersja dokumentu 1.0.7**

Spis treści

[1. Wstęp 5](#_Toc503640117)

[1.1 Temat pracy dyplomowej 5](#_Toc503640118)

[1.2 Cel i zakres pracy dyplomowej 5](#_Toc503640119)

[1.3 Wprowadzenie do problemu 5](#_Toc503640120)

[2. Analiza 6](#_Toc503640121)

[2.1 Założenia ogólne 6](#_Toc503640122)

[2.1.1 Podstawowe pojęcia związane z zadaniem projektowym 6](#_Toc503640123)

[2.1.2 Szczegóły opisu 7](#_Toc503640124)

[2.1.3 RCP(rejestracja czasu pracy): 7](#_Toc503640125)

[2.1.4 RZP(Realizacja zadań produkcyjnych) : 8](#_Toc503640126)

[2.2 Przegląd produktów rynkowych 10](#_Toc503640127)

[2.3 Wymagania dotyczące systemu 10](#_Toc503640128)

[2.3.1 Wymagania funkcjonalne 10](#_Toc503640129)

[2.3.2 Wymagania Pozafunkcjonalne – dopiszę więcej, na razie nie mam pomysłów 12](#_Toc503640130)

[2.3.3 Aktorzy 13](#_Toc503640131)

[2.4 Modelowanie 14](#_Toc503640132)

[2.4.1 Diagramy przypadków użycia 14](#_Toc503640133)

[2.4.2 Diagramy sekwencji 32](#_Toc503640134)

[2.4.3 Identyfikacja aktywności 32](#_Toc503640135)

[2.4.4 Diagram hierarchii funkcji 32](#_Toc503640136)

[2.4.5 Diagram klas 32](#_Toc503640137)

[2.5 Wybór architektury 34](#_Toc503640138)

[3. Projektowanie 34](#_Toc503640139)

[3.1 Projekt bazy danych 34](#_Toc503640140)

[3.1.1 Tabele 34](#_Toc503640141)

[3.1.2 Widoki 34](#_Toc503640142)

[3.1.3 Procedury składowane 34](#_Toc503640143)

[3.1.4 Procedury wyzwalane 34](#_Toc503640144)

[3.1.5 Funkcje skalarne/tabularne 34](#_Toc503640145)

[3.2 Projekt interfejsu 34](#_Toc503640146)

[3.2.1 Ogólna koncepcja interfejsu 34](#_Toc503640147)

[3.2.2 Interfejs głównego formularza 34](#_Toc503640148)

[3.2.3 Interfejs pozostałych formularzy 34](#_Toc503640149)

[3.2.4 System pomocy i komunikatów 34](#_Toc503640150)

[4. Dokumentacja systemu 34](#_Toc503640151)

[4.1 Opis instalacji systemu 34](#_Toc503640152)

[4.2 Dokumentacja użytkowa 34](#_Toc503640153)

[4.3 Testowanie systemu 34](#_Toc503640154)

[5. Zakończenie 34](#_Toc503640155)

[5.1 Podsumowanie 34](#_Toc503640156)

[5.2 Wnioski 34](#_Toc503640157)

[5.3 Możliwości dalszego rozwoju 34](#_Toc503640158)

[Dodatek A: Słownik terminów użytych w pracy 34](#_Toc503640159)

[Dodatek B: Wykaz rysunków 35](#_Toc503640160)

[Dodatek C: Słownik tabel 35](#_Toc503640161)

[Dodatek D: Wykaz literatury 35](#_Toc503640162)

[Bibliografia 35](#_Toc503640163)

# Wstęp

## Temat pracy dyplomowej

Analiza, projekt i implementacja systemu wspierającego wybrane funkcje zarządzania pracą w organizacji produkcyjnej

## Cel i zakres pracy dyplomowej

Celem pracy dyplomowej jest opracowanie systemu/aplikacji służącej do rejestracji czasu pracy oraz definiowania i rozliczania zadań produkcyjnych. Zakres pracy obejmować będzie:

* Zaprojektowanie bazy danych
* Zaprojektowanie mechanizmów bazodanowych
* Utworzenie działającej aplikacji
* Wprowadzenie danych testowych
* Testowanie aplikacji

## Wprowadzenie do problemu

Przedsiębiorstwo Meblex produkujące meble buduje nową siedzibę oraz halę produkcyjną. Decyzją zarządu, nowa lokalizacja będzie wyposażona w nowoczesny system informatyczny. System który będzie spełniał wymagania powinien mieć możliwość:

* ewidencjonowania osoby - pracowników,
* zarządzania czasem pracy pracowników
* zarządzania zadaniami dla pracowników
* budowania okresowych raportów czasowych

Przedsiębiorstwo zatrudnia około 500 osób na różnych stanowiskach. System powinien być dostosowany do takiej ilości pracowników oraz zapewnić wydajny dostęp do jego zasobów przez 24h na dobę i siedem dni w tygodniu.

# Analiza

W tym rozdziale przedstawiono wymagania funkcjonalne i pozafunkcjonalne

## Założenia ogólne

Przedsiębiorstwo ma swoją strukturę podzieloną na wydziały. Każdy z wydziałów charakteryzuje się innym systemem pracy. Niektóre działy pracują w stałym systemie czasowym i o stałych porach, a inne pracują w systemie zmianowym. Najbardziej zróżnicowany jest dział produkcyjny gdzie oprócz rozliczania czasu pracy, rozliczana jest wydajność produkcyjna.

Pracownicy biurowi mają zazwyczaj określone stałe godziny pracy i są one zmieniane w jedynie w szczególnych przypadkach. Pracownicy produkcyjni mają ustalane grafiki miesięczne i mogą rozpoczynać pracę na dwie zmiany. Niektóre stanowiska pracy nie wymagają konkretnej godziny przyjścia do pracy, ale jedynie wypracowania określonego czasu.

W części produkcyjnej przedsiębiorstwa pracownicy są dodatkowo rozliczani z zadań produkcyjnych. Zadania produkcyjne są generowanie przez dział handlowy który je przekazuje w postaci zamówień. Zamówienia mogą być składane przez klientów jak również przez inny wewnętrzny dział przedsiębiorstwa. Zamówienia są przekazywane codziennie rano, kierownikom w postaci kartek z wypisanymi produktami. Kierownicy na tej podstawie wypisują zadania pracownikom. Na koniec okresu rozliczeniowego każdy pracownik jest rozliczany ze swojej pracy godzinowo oraz ilościowo.

Na podstawie założeń które zostały dostarczone przez klienta, można przyjać że system będzie podzielony na dwie części:

* Rejestracja czasu pracy
* Realizacja zadań produkcyjnych

### Podstawowe pojęcia związane z zadaniem projektowym

* ***Pracownik*** – osoba zatrudniona w przedsiębiorstwie
* ***Kierownik*** – osoba zatrudniona w przedsiębiorstwie zarządzająca pracownikami
* ***Kadrowa*** – osoba zatrudniona w przedsiębiorstwie zajmująca się obsługą kadrowo-płacową pracowników
* ***Klient*** – osoba składająca zamówienie.
* ***Zamówienie*** - czynność związana zebraniem potrzeb klienta zawierająca listę produktów do wytworzenia
* ***Harmonogram*** – obiekt opisujący tydzień pracujący pracownika
* ***Dniówka*** – obiekt opisujący dzień pracujący pracownika
* ***Zadanie*** – obiekt zawierający elementy do wyprodukowania
* ***Norma* –** wyznaczony czas pracy na dzień pracujący

### Szczegóły opisu

#### **Pracownik** - potrzeba przechowywania: nazwisko, imię, wiek, adres zamieszkania, telefon, płeć, uprawnienia systemowe, przynależność do jednostki organizacyjnej.

#### **Kierownik** – potrzeba przechowywania: nazwisko, imię, wiek, adres zamieszkania, telefon, płeć, uprawnienia systemowe, przynależność do jednostki organizacyjnej.

#### **Kadrowa** – potrzeba przechowywania: nazwisko, imię, wiek, adres zamieszkania, telefon, płeć, uprawnienia systemowe, przynależność do jednostki organizacyjnej.

#### **Klient** – potrzeba przechowywania: Nazwa firmy, Adres, Miasto, NIP, Regon, Telefon1, Telefon 2, Osoba do kontaktów1, Osoba do kontaktów2

#### **Zamówienie** – potrzeba przechowywania: nazwę zamawiającego, datę wprowadzenia, składniki oferty, datę ukończenia, termin realizacji,osobę wprowadzającą.

#### **Harmonogram** – potrzeba przechowywania: Nazwa, dni tygodnia z przypisanym typem dniówki.

#### **Dniówka** – potrzeba przechowywania: Godzina wejścia, Godzina wyjścia, Długość przerwy, Ile można przyjść wcześniej, Ile można wyjść wcześniej, Ile można przyjść później, Ile można wyjść później, przejście przez 00:00, zezwolenie na nadgodziny

**Uwagi:**

1. Dniówka może zaczynać się jednego dnia a kończyć następnego.

#### **Zadanie** - potrzeba przechowywania: nadany numer zadania, numer zamówienia, czynność, imię i nazwisko pracownika który się zajmuje zadaniem, informacja czy zakończono realizację, notatka do zadania.

### RCP(rejestracja czasu pracy):

Do systemu będą wprowadzani pracownicy. Każdy pracownik będzie miał przypisane godziny pracy na pracujący dzień tygodnia. Po przyjściu do pracy, pracownik będzie wciskał przycisk Start/Stop/Przerwa początek//koniec/Wyjście służbowe/Powrót służbowy.

Na podstawie transakcji będzie budowany raport który będzie liczył ilość przepracowanych godzin, ilość absencji, łączny czas przerw.

System będzie umożliwiał definiowanie warunków przychodzenia do pracy i wychodzenia z pracy. Całość będzie się opierać na harmonogramie zbudowanym z typów dniówek.

#### **Dniówka**

Dniówka jest to okres czasu mający swój początek o dowolnej godzinie i kończący się o dowolnej godzinie. Dniówka nie musi zaczynać się i kończyć tego samego dnia.

Dniówka będzie miała swoją nazwę, godzinę rozpoczęcia i zakończenia pracy, długość przysługującej przerwy liczoną w minutach i zezwolnienie na nadgodziny. Nadgodziny będą definiowane na cały czas obowiązywania dniówki.

Oprócz tego każda dniówka będzie miała możliwość wskazania jaka jest dopuszczalna najwcześniejsza i najpóźniejsza godzina rozpoczęcia pracy oraz jaka jest najwcześniejsza i najpóźniejsza godzina zakończenia pracy. Ma to na celu eliminowanie doliczania sobie czasu do nadgodzin oraz wcześniejszego niż planowane opuszczenia stanowiska.

#### **Harmonogram**

Harmonogram jest to wykaz dniówek na każdy dzień tygodnia. Jeśli na dany dzień nie ma wpisanej dniówki to znaczy że jest to dzień wolny. Żeby ustalić harmonogram musi być zdefiniowana przynajmniej jedna dniówka.

Harmonogram będzie przypisywany do pracownika na określony czas co sprawi że program będzie miał elastyczny sposób budowania schematu pracy pracowników.

#### **Algorytmy rozliczania**

Z uwagi na różne podejścia pracodawców do rozliczania pracowników zaimplementowane będą sposoby liczenia czasu pracy. System będzie posiadał trzy algorytmy rozliczania czasu pracy:

1. Liczy czas w godzinach pracy między Start/Stop oraz Wyjście służbowe/Powrót z Wyj. Służb. Uwzględni przy tym narzucone normy przyjścia, wyjścia i czasu przerwy.
2. Liczy czas między pierwszym wejściem (dzień roku, godzina, minuta) a ostatnim wyjściem (dzień roku, godzina, minuta) nie uwzględniając tego co działo się pomiędzy.
3. Liczy czas między Start/Stop bez patrzenia na normy.

### RZP(Realizacja zadań produkcyjnych) :

System ma na celu zwiększenie wydajności pracy pracowników. W tym celu czas pracy będzie podzielony na zadania produkcyjne które należy wykonać.

Zadanie produkcyjne będzie budowane automatycznie na podstawie zamówienia i produktów które zawiera. Każdy produkt będzie rozbity na elementy, a zrobienie elementu będzie zadaniem produkcyjnym. Następnie na podstawie kompetencji, zadanie będzie przydzielane pracownikom.

Przykład. Zamówienie na 3 stoły kuchenne->Po wprowadzeniu generują się 2 zadania produkcyjne->Zrobienie 3 blatów stołowych oraz Zrobienie 12 nóg stołowych.-> Jedno zadanie idzie do pracownika robiącego blaty a drugie do robiącego nogi.

Rozpoczęcie pracy nad danym zadaniem rozpoczyna się odznaczeniem w systemie że jest w trakcie realizacji. Jeśli zadania przypisane do zamówienia zostaną zakończone, zamówienie zmienia status na zrealizowane. Zakończenie pracy również będzie odznaczane w systemie.

System będzie umożliwiał raportowanie z wykonanych zadań w określonym przedziale czasowym. Oprócz tego będzie umożliwiał raportowanie:

1. Realizowanych aktualnie zadań

2. Średnich czasów wykonania poszczególnych zadań – zestawień dla danego pracownika w porównaniu z resztą na tym samym stanowisku

3. Porównania z poprzednimi okresami.

#### **Zamówienie**

Zamówienie jest to polecenie na wykonanie jakiegoś towaru z asortymentu dostępnego w ofercie. Zamówienie będzie tworzone przez wyznaczonego do tego pracownika przedsiębiorstwa. Będzie zawierać nazwę zamawiającego, wybrane produkty, datę zamówienia oraz osobę która je zleciła. Jedno zamówienie może dotyczyć jednego zamawiającego i może zawierać wiele produktów.

#### **Produkty**

Produkt to rzecz lub czynności dobro i usługa, które można nabyć, użytkować lub skonsumować. Produkty będą definiowane i wprowadzane do systemu przez pracowników przedsiębiorstwa.

Każdy produkt będzie posiadał nazwę oraz listę elementów z których się składa. Na przykład:

Zakładamy, że produktem będzie stół do kuchni. Stół składać się będzie z elementów – jeden blat, cztery nogi

#### **Elementy**

Element lub elementy są to części z których składa się produkt.

Element będzie posiadał swoją nazwę oraz typ. Typ będzie potrzebny do uporządkowania kompetencji dla pracowników. Pracownicy produkcyjni będą mieli wpisane typy elementów które potrafią wykonać.

#### **Zadanie produkcyjne**

Zadanie produkcyjne jest to zestaw czynności które musi wykonać pracownik żeby wykonać element produktu.

Zadanie produkcyjne będzie przydzielane pracownikom na podstawie kompetencji i będzie miało rozliczany czas poświęcony na realizację.

#### **Zaznaczanie czasu pracy przy zadaniu produkcyjnym** -potrzeba przechowywania informacji: numer zadania, imię i nazwisko pracownika, typ transakcji, data i czas.

**Uwagi**:

1. W czasie pracy nad zadaniem, dozwolone typy transakcji to start i stop.

#### **Zaznaczanie czasu pracy w przedsiębiorstwie** – potrzeba przechowania informacji: imię i nazwisko pracownika, data i czas, typ transakcji.

1. dozwolone typy transakcji to rozpoczęcie czasu pracy, zakończenie czasu pracy, początek przerwy, koniec przerwy, początek wyjścia służbowego, koniec wyjścia służbowego.

## Przegląd produktów rynkowych

Na rynku jest kilka produktów które częściowo spełniają oczekiwania klienta. Większość rozwiązań skupia się na rozliczaniu czasu pracy pracownika…….

DO DOPRACOWANIA PÓŹNIEJ

## Wymagania dotyczące systemu

Wymagania dotyczące systemu możemy ogólnie podzielić na dwa typy: funkcjonalne i pozafunkcjonalne. Opisują one jak system ma działać, jakie ma mieć funkcje użytkowe oraz jakie są oczekiwania jakościowe od systemu.

### Wymagania funkcjonalne

Na podstawie opisu problemu i wymagań możemy zdefiniować następujące wymagania funkcjonalne:

* Logowanie(PU1)
* Dodawanie danych pracowników(PU2)
* Usuwanie danych pracowników (PU3)
* Modyfikacja danych pracowników(PU4)
* Przeglądanie listy pracowników(PU5)
* Dodawanie typów obowiązków pracowników(PU6)
* Usuwanie typów obowiązków pracowników(PU7)
* Modyfikowanie typów obowiązków pracowników(PU8)
* Przeglądanie typów obowiązków(PU9)
* Przydzielanie typów obowiązków pracownikom(PU10)
* Odbieranie typów obowiązków pracownikom(PU11)
* Dodawanie harmonogramów pracowników(PU12)
* Usuwanie harmonogramów pracowników(PU13)
* Modyfikacja harmonogramów pracowników(PU14)
* Przeglądanie harmonogramów(PU15)
* Dodawanie zadań produkcyjnych(PU16)
* Usuwanie zadań produkcyjnych(PU17)
* Przeglądanie zadań produkcyjnych(PU18)
* Nadawanie uprawnień do programu(PU19)
* Odbieranie uprawnień do programu(PU20)
* Przeglądanie uprawnień do programu(PU21)
* Dodawanie nowych struktur przedsiębiorstwa(PU22)
* Usuwanie struktur przedsiębiorstwa(PU23)
* Modyfikacja nazw struktur przedsiębiorstwa(PU24)
* Przeglądanie struktur przedsiębiorstwa(PU25)
* Tworzenie kopii zapasowej(PU26)
* Przeliczanie czasu pracy pracowników ogólnie(PU27)
* Przeliczanie czasu pracy pracowników poświęconego na wykonywanie zadań(PU28)
* Dodawanie transakcji czasu pracy(PU29)
* Usuwanie transakcji czasu pracy(PU30)
* Modyfikowanie transakcji czasu pracy(PU31)
* Przeglądanie transakcji czasu pracy(PU32)
* Generowanie okresowych raportów czasowych(PU33)
* Drukowanie raportów czasowych(PU34)
* Dodawanie dniówki(PU35)
* Usuwanie dniówki (PU36)
* Modyfikowanie dniówki(PU37)
* Przeglądanie dniówek(PU38)

### Wymagania Pozafunkcjonalne – dopiszę więcej, na razie nie mam pomysłów

Wymagania pozafunkcjonalne czyli inaczej jakościowe podobnie jak funkcjonalne zostały określone na podstawie analizy wstępnej, jak również określone przez zamawiającego.   
Wymagania:

Tabela Wymagania funkcjonalne nr 1

|  |  |
| --- | --- |
| Identyfikator | WPF1 |
| Opis | System będzie działać w oparciu o bazę danych MS SQL 2014 lub wyższą |
| Priorytet | Wysoki |

Tabela Wymagania funkcjonalne nr 2

|  |  |
| --- | --- |
| Identyfikator | WPF2 |
| Opis | System będzie działać w środowisku .NET, systemie operacyjnym Windows 7 i wyższym. |
| Priorytet | Wysoki |

Tabela Wymagania funkcjonalne nr 3

|  |  |
| --- | --- |
| Identyfikator | WPF3 |
| Opis | System będzie przechowywał hasła do systemu w postaci zaszyfrowanej |
| Priorytet | Wysoki |

Tabela Wymagania funkcjonalne nr 4

|  |  |
| --- | --- |
| Identyfikator | WPF4 |
| Opis | System będzie przechowywał historię logowań |
| Priorytet | Niski |

Tabela Wymagania funkcjonalne nr 5

|  |  |
| --- | --- |
| Identyfikator | WPF5 |
| Opis | Czas wygenerowania raportu czasu pracy dla jednego pracownika nie przekroczy 30 sekund przy założeniu że raport będzie obejmował maksymalnie 3 miesiące. |
| Priorytet | Średni |

Tabela Wymagania funkcjonalne nr 6

|  |  |
| --- | --- |
| Identyfikator | WPF6 |
| Opis | System będzie podawać czas z dokładnością do minut |
| Priorytet | Wysoki |

Tabela Wymagania funkcjonalne nr 7

|  |  |
| --- | --- |
| Identyfikator | WPF7 |
| Opis | Raporty czasu pracy będą przechowywane przynajmniej przez rok od momentu utworzenia |
| Priorytet | Niski |

Tabela Wymagania funkcjonalne nr 8

|  |  |
| --- | --- |
| Identyfikator | WPF8 |
| Opis | System musi zachowywać w sposób ciągły wewnętrzną integralność, bez względu na:   * czynności serwisowe; * inne czynności wykonywane przez użytkownika; * awarię komponentów systemu. |
| Priorytet | Wysoki |

### Aktorzy

Po analizie wstępnej zdefiniowano że system będzie obsługiwany przez aktorów biorących udział w działaniu programu:

* Administrator
* Kadrowa
* Pracownik
* Kierownik
* Szef

Aktorzy będą mieli dostęp do poszczególnych elementów programu zgodnie z poniższą tabelką uprawnień:

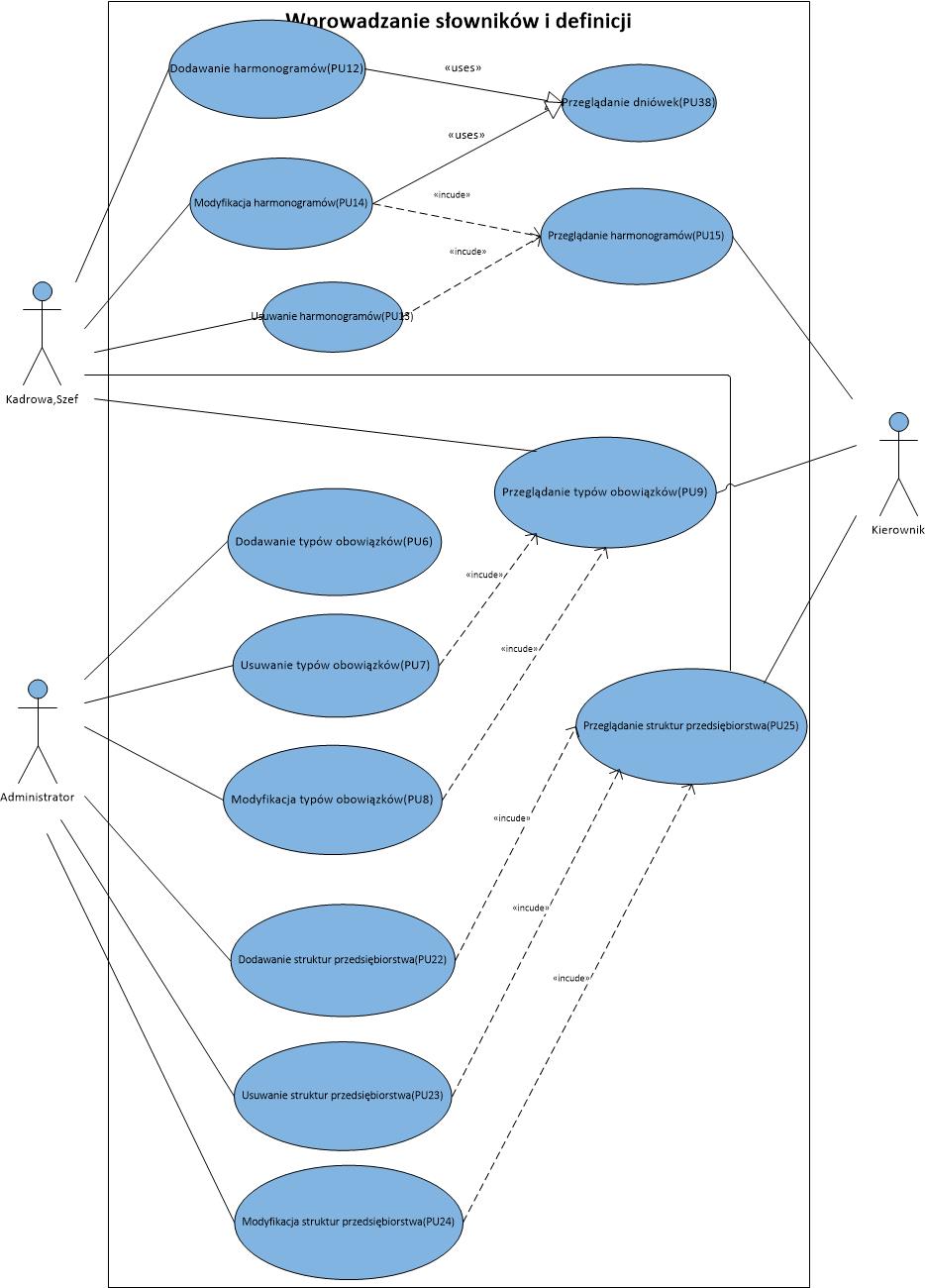
Tabela Uprawnienia aktorów do elementów systemu

|  | Administrator | Kadrowa | Pracownik | Kierownik | Szef |
| --- | --- | --- | --- | --- | --- |
| PU1 | • | • | • | • | • |
| PU2 |  | • |  |  | • |
| PU3 |  | • |  |  | • |
| PU4 | • | • |  | • | • |
| PU5 | • | • |  | • | • |
| PU6 | • |  |  |  |  |
| PU7 | • |  |  |  |  |
| PU8 | • |  |  |  |  |
| PU9 | • | • |  | • | • |
| PU10 |  | • |  | • | • |
| PU11 |  | • |  | • | • |
| PU12 |  | • |  | • | • |
| PU13 |  | • |  | • | • |
| PU14 |  | • | • | • | • |
| PU15 | • | • |  | • | • |
| PU16 | • |  |  |  |  |
| PU17 | • |  |  |  |  |
| PU18 | • | • |  | • | • |
| PU19 | • |  |  |  |  |
| PU20 | • |  |  |  |  |
| PU21 | • |  |  |  | • |
| PU22 | • |  |  |  |  |
| PU23 | • |  |  |  |  |
| PU24 | • | • |  |  | • |
| PU25 | • | • |  | • | • |
| PU26 | • |  |  |  |  |
| PU27 |  | • | • | • | • |
| PU28 |  | • | • | • | • |
| PU29 |  | • | • | • |  |
| PU30 |  | • |  | • |  |
| PU31 |  | • |  | • | • |
| PU32 | • | • | • | • | • |
| PU33 |  | • |  | • | • |
| PU34 |  | • |  |  | • |
| PU35 | • |  |  |  |  |
| PU36 | • |  |  |  |  |
| PU37 | • |  |  |  |  |

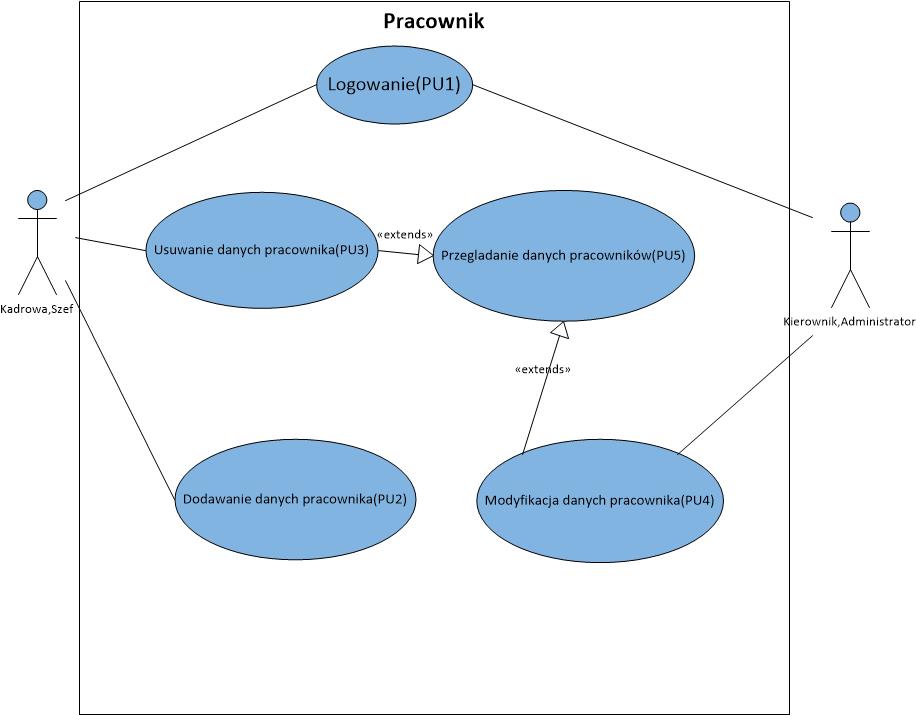
## Modelowanie

W podrozdziale przedstawiono projekt graficzny systemu. Do jego wykonania wykorzystano i przedstawiono poniżej wybrane (ze względu na ograniczenia objętościowe Pracy Dyplomowej) diagramy języka UML. Do tworzenia diagramów został użyty program Microsoft Visio.

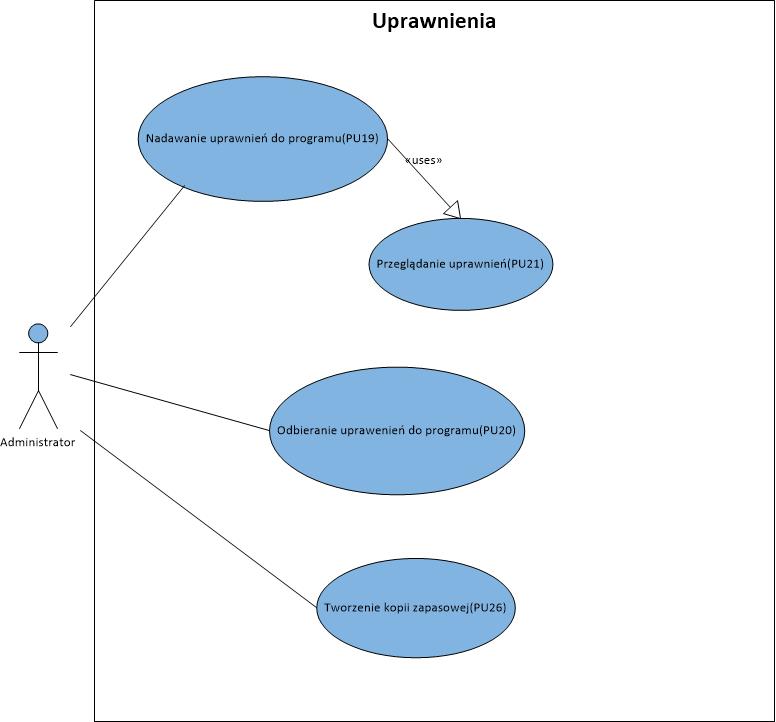
### Diagramy przypadków użycia



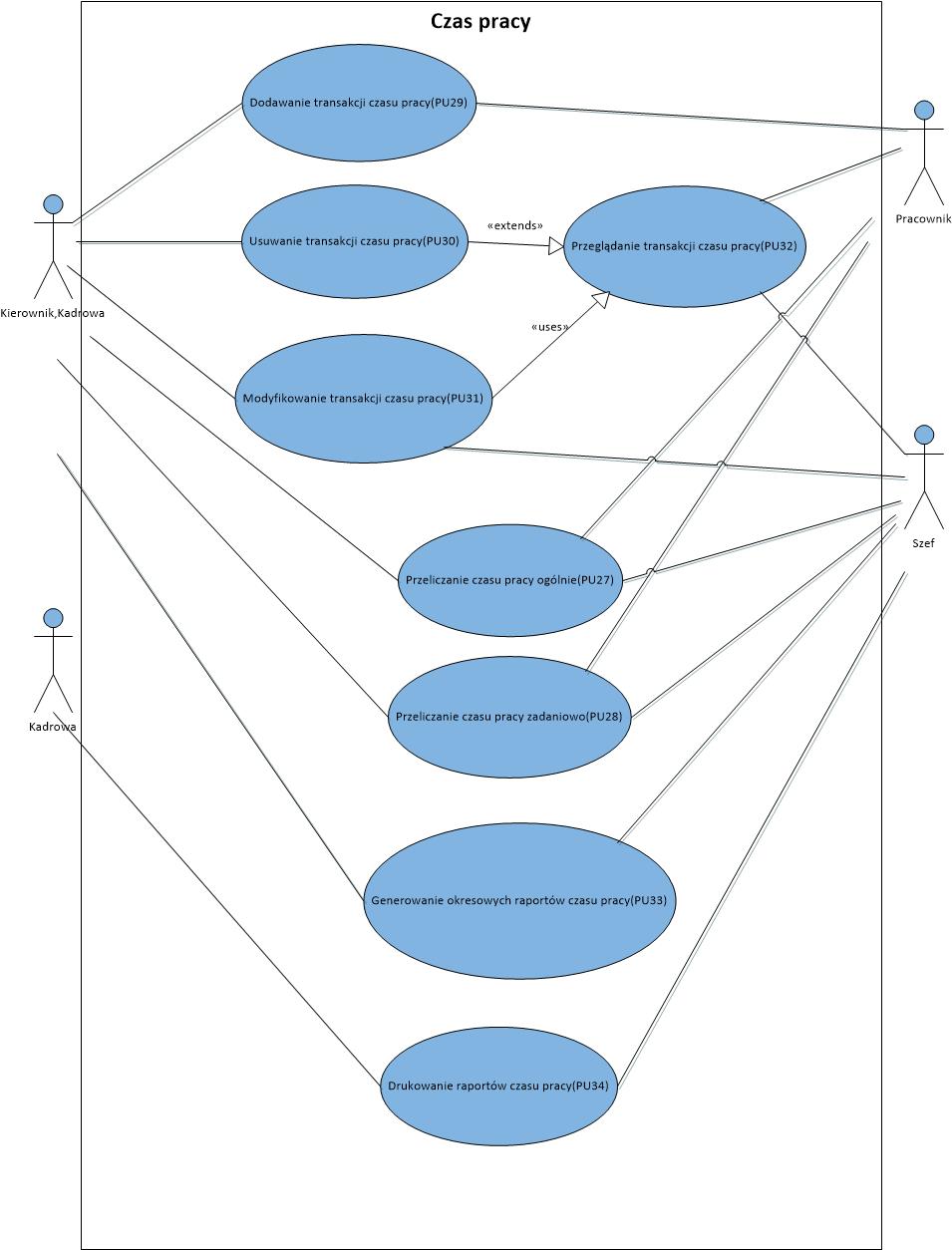
Rysunek Przypadki użycia; źródło: opracowanie własne



Rysunek Przypadki użycia; źródło: opracowanie własne



Rysunek Przypadki użycia; źródło: opracowanie własne



Rysunek Przypadki użycia; źródło: opracowanie własne

Tabela Przypadki użycia- PU1:Logowanie

|  |
| --- |
| **Przypadek użycia:** PU1:Logowanie |
| **Aktorzy:** Administrator, Kadrowa, Szef, Kierownik |
| **Opis:** Przypadek użycia dotyczy logowania do systemu |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlone jest okno logowania zawierające pola:  * Login * Hasło   Przyciski:   * Zaloguj * Anuluj  1. System czeka na aktywność użytkownika |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez wpisanie danych logowania |
| **Scenariusz główny** |
| 1. Użytkownik wpisuje login i hasło do systemu i wybiera opcje Zaloguj 2. System sprawdza czy login i hasło do siebie pasują 3. System potwierdza poprawność danych logowania przez wyświetlenie ekranu głównego interfejsu użytkownika   4. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 1.A Użytkownik podał niewłaściwe login lub hasło  1.A.1 System podaje informację o błędnych danych logowania i pozostałą ilość prób wpisania  1.A.2 System wyświetla okno logowania  1.B Użytkownik podał niewłaściwe dane logowania powyżej dozwolonej ilości  1.B.1 System podaje informację o wykorzystanej ilości prób i kończy działanie  1.B.2 System kończy działanie |

Tabela Przypadki użycia- PU2 : Dodawanie danych pracownika

|  |
| --- |
| **Przypadek użycia:** PU2 : Dodawanie danych pracownika |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy dodawania danych nowego pracownika |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę pracowników 2. Na formularzu przyciski:  * nowy * edytuj * usuń   Formularz zawiera również pola z danymi pracownika:   * imię * nazwisko * data urodzenia * pesel * ulica * miasto * kod pocztowy * wykształcenie  1. Użytkownik wybiera opcję wprowadzenia nowego pracownika 2. System umożliwia wpisanie danych nowego pracownika 3. Użytkownik wprowadza wymagane dane   6. Użytkownik akceptuje wprowadzone dane przez wybranie opcji zapisu  7. System zapisuje dane  8. System wyświetla informację o zapisanych danych  9. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 3.A Użytkownik chce anulować wprowadzanie nowego pracownika  3.A.1 Użytkownik wybiera opcję anuluj  3.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia- PU3 : Usuwanie danych pracownika

|  |
| --- |
| **Przypadek użycia:** PU3 : Usuwanie danych pracownika |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy usuwania danych pracownika |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę pracowników 2. Na formularzu przyciski:  * nowy * edytuj * usuń  1. Użytkownik wybiera opcję usuwania 2. System wyświetla potwierdzenie usunięcia wybranego pracownika 3. Użytkownik akceptuje usunięcie   6. System usuwa dane  7. System wyświetla informację o usunięciu danych  8. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować usuwanie harmonogramu  4.A.1 Użytkownik nie potwierdza usunięcia pracownika  4.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia- PU4:Modyfikacja danych pracownika

|  |
| --- |
| **Przypadek użycia:** PU4:Modyfikacja danych pracownika |
| **Aktorzy:** Kadrowa, Szef, Kierownik, Administrator |
| **Opis:** Przypadek użycia dotyczy modyfikacji danych wybranego pracownika |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej, szefa, kierownika lub administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę pracowników 2. Na formularzu przyciski:  * nowy * edytuj * usuń   Formularz zawiera również pola z danymi pracownika:   * imię * nazwisko * data urodzenia * pesel * ulica * miasto * kod pocztowy * wykształcenie  1. Użytkownik wybiera opcję wprowadzenia edycji 2. System wyświetla formularz z danymi pracownika które były już wprowadzone i umożliwia ich edycję. 3. Użytkownik wprowadza nowe dane i wybiera opcję zapisu. 4. System zapisuje dane 5. System wyświetla informację o zapisaniu danych 6. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować modyfikację danych pracownika  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia – PU06:Dodawanie typów obowiązków

|  |
| --- |
| **Przypadek użycia:** PU06:Dodawanie typów obowiązków |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy dodawania do listy typu obowiązku |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. Użytkownik wybiera opcję wprowadzenia nowego typu obowiązku 2. Na ekranie pojawia się okno z polami:  * Nazwa, * Symbol obowiązku   Oprócz tego pojawiają się też przycisk Zapisz oraz Anuluj  3. Użytkownik wprowadza wymagane dane  4. Użytkownik akceptuje wprowadzone dane przez wciśnięcie przycisku Zapisz  5. System zapisuje dane  6. System wyświetla informację o zapisanych danych  7. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować wprowadzanie nowego typu obowiązku  4.A.1 Administrator wybiera przycisk anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia – PU07:Usuwanie typów obowiązków

|  |
| --- |
| **Przypadek użycia:** PU07: Usuwanie typów obowiązków |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy usuwania z listy typów obowiązków |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. Użytkownik zaznacza na liście typ obowiązku  2. System wyświetla przycisk usuń  3. Użytkownik wciska przycisk usuń  3. System wyświetla potwierdzenie usunięcia  4. Użytkownik akceptuje usunięcie typu obowiązku  5. System usuwa dane  6. System wyświetla informację o usunięciu danych  7. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować usuwanie typu obowiązku  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU08:Modyfikacja typów obowiązków

|  |
| --- |
| **Przypadek użycia:** PU08:Modyfikacja typów obowiązków |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy modyfikacji typu obowiązku |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. Użytkownik wybiera typ obowiązku 2. System wyświetla dane obowiązku oraz przycisk edytuj   3. Użytkownik wybiera opcję edytuj  4. System umożliwia edycję danych i wyświetla przycisk zapisz  5. Użytkownik wprowadza zmiany w wybranych polach i naciska przycisk zapisz  6. System zapisuje dane  7. System wyświetla informację o zapisanych danych  8. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować modyfikację typu obowiązku  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia- PU12 : Dodawanie harmonogramu

|  |
| --- |
| **Przypadek użycia:** PU12 : Dodawanie harmonogramu |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy dodawania do listy typu obowiązku |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę harmonogramów 2. Na formularzu zakładki znajdują się również pola:  * Dni tygodnia z listą wybieraną na której pojawiają się zdefiniowane wcześniej dniówki * Nazwa * Przyciski nowy i edytuj  1. Użytkownik wybiera opcję wprowadzenia nowego harmonogramu pracy 2. System umożliwia wpisanie nazwy harmonogramu oraz wybór dniówek z listy na każdy dzień tygodnia 3. Użytkownik wprowadza wymagane dane i wybiera opcję zapisu   6. Użytkownik akceptuje wprowadzone dane  7. System zapisuje dane  8. System wyświetla informację o zapisanych danych  9. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować wprowadzanie nowego harmonogramu  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU13 : Usuwanie harmonogramów

|  |
| --- |
| **Przypadek użycia:** PU13 : Usuwanie harmonogramów |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy usuwania harmonogramów z listy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę harmonogramów 2. Na formularzu, po zaznaczeniu harmonogramu pojawia się opcja usuń 3. Użytkownik wybiera opcję usuwania 4. System wyświetla potwierdzenie usunięcia wybranego harmonogramu 5. Użytkownik akceptuje usunięcie   6. System usuwa dane  7. System wyświetla informację o usunięciu danych  8. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować usuwanie harmonogramu  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia- PU14:Modyfikacja harmonogramów

|  |
| --- |
| **Przypadek użycia:** PU14:Modyfikacja harmonogramów |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy modyfikacji zdefiniowanego wcześniej harmonogramu |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę harmonogramów 2. Na formularzu zakładki znajdują się również pola:  * Dni tygodnia z listą wybieraną na której pojawiają się zdefiniowane wcześniej dniówki * Nazwa * Przyciski nowy i edytuj  1. Użytkownik wybiera opcję wprowadzenia edycji 2. System wyświetla formularz z danymi harmonogramu które były już wprowadzone i umożliwia ich edycję. 3. Użytkownik wprowadza nowe dane i wybiera opcję zapisu.   6. System zapisuje dane  7. System wyświetla informację o zapisaniu danych  8. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować modyfikację harmonogramu  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU19:Nadawanie uprawnień do programu

|  |
| --- |
| **Przypadek użycia:** PU19:Nadawanie uprawnień do programu |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy nadawania uprawnień do programu |
| **Warunki początkowe** |
| 1. System ma na stałe wprowadzonego użytkownika z uprawnieniami administratora, którego nie można usunąć. 2. Na ekranie wyświetlony jest graficzny interfejs użytkownika 3. System czeka na aktywność użytkownika 4. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę pracowników i ich uprawnienia do programu. Żeby nadać uprawnienia użytkownik musi być już dodany do systemu. 2. Użytkownik wybiera pracownika z listy 3. System wyświetla przycisk nadaj uprawnienia 4. Użytkownik wybiera opcję nadawania uprawnień 5. System wyświetla okno z formularzem do wprowadzania danych logowania:  * Login * Hasło * Powtórz hasło   Z przyciskami Zapisz oraz Anuluj   1. Użytkownik wprowadza dane logowania i wybiera opcję zapisu 2. System sprawdza zgodność wprowadzonych haseł 3. System wyświetla informację o zapisaniu danych 4. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 6.A Użytkownik chce anulować modyfikację danych pracownika  6.A.1 Użytkownik wybiera opcję anuluj  6.A.2 System wyświetla główne menu programu  6.B Użytkownik błędnie wprowadził powtórzone hasło  6.B.1 System wyświetla informację o błędzie  6.B.2 System kasuje dane z pól z hasłami  6.B.3 Użytkownik musi ponownie wprowadzić hasła |

Tabela Przypadek użycia- PU20:Odbieranie uprawnień do programu

|  |
| --- |
| **Przypadek użycia:** PU20:Odbieranie uprawnień do programu |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy odbierania uprawnień do programu |
| **Warunki początkowe** |
| 1. System ma na stałe wprowadzonego użytkownika z uprawnieniami administratora, którego nie można usunąć. 2. Na ekranie wyświetlony jest graficzny interfejs użytkownika 3. System czeka na aktywność użytkownika 4. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla zakładkę na której można zobaczyć listę pracowników i ich uprawnienia do programu. 2. Użytkownik wybiera pracownika z listy 3. System wyświetla przycisk odbierz uprawnienia 4. Użytkownik wybiera opcję odbierania uprawnień 5. System wyświetla okno z potwierdzeniem odebrania uprawnień z przyciskami tak i nie 6. Użytkownik potwierdza działanie. 7. System usuwa uprawnienia pracownikowi 8. System wyświetla informację o zapisaniu danych 9. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 5.A Użytkownik chce anulować odbieranie uprawnień pracownika  5.A.1 Użytkownik wybiera opcję nie podczas potwierdzania odbierania uprawnień  5.A.2 System wyświetla menu uprawnień programu |

Tabela Przypadek użycia- PU22:Dodawanie struktur przedsiębiorstwa

|  |
| --- |
| **Przypadek użycia:** PU22:Dodawanie struktur przedsiębiorstwa |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy dodawania do listy typu obowiązku |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla strukturę przedsiębiorstwa w postaci drzewka 2. System wyświetla opcję dodawania nowego elementu struktury 3. Użytkownik wybiera opcję wprowadzenia nowego elementu struktury 4. System wyświetla pola do wypełnienia:  * Nazwa * Wybór jednostki nadrzędnej w postaci listy wybieranej   Ponad to pojawiają się przyciski zapisz oraz anuluj  5. Użytkownik wprowadza wymagane dane i wybiera opcję zapisz  6. System zapisuje dane  7. System wyświetla informację o zapisanych danych |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować wprowadzanie nowego elementu struktury  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU23: Usuwanie struktur przedsiębiorstwa

|  |
| --- |
| **Przypadek użycia:** PU23: Usuwanie struktur przedsiębiorstwa |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy usuwania z listy typu obowiązku |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla strukturę przedsiębiorstwa w postaci drzewka  2. Użytkownik zaznacza na liście element struktury  3. System wyświetla opcję usuwania  4. Użytkownik wybiera opcję usuń  5. System wyświetla potwierdzenie usunięcia elementu struktury  6. Użytkownik akceptuje usunięcie.  7. System usuwa dane  8. System wyświetla informację o usunięciu danych i aktualizuje strukturę przedsiębiorstwa  9. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować usuwanie elementu struktury  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU24:Modyfikacja struktur przedsiębiorstwa

|  |
| --- |
| **Przypadek użycia:** PU24:Modyfikacja struktur przedsiębiorstwa |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy modyfikacji struktury przedsiębiorstwa |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla strukturę przedsiębiorstwa w postaci drzewka  2. Użytkownik zaznacza na liście element struktury  3. System wyświetla opcję edycji w postaci przycisku  4. Użytkownik wybiera opcję edycji  5. System wyświetla aktualną nazwę struktury oraz ewentualną strukturę nadrzędną oraz przyciski zapisz i anuluj  6. Użytkownik wprowadza wymagane zmiany  7. Użytkownik akceptuje wprowadzone zmiany wybierając opcję zapisu.  8. System zapisuje dane  9. System wyświetla informację o zapisanych danych  10. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 4.A Użytkownik chce anulować modyfikację elementu struktury  4.A.1 Użytkownik wybiera opcję anuluj  4.A.2 System wyświetla główne menu programu |

Tabela Przypadek użycia- PU26:Tworzenie kopii zapasowej

|  |
| --- |
| **Przypadek użycia:** PU26:Tworzenie kopii zapasowej |
| **Aktorzy:** Administrator |
| **Opis:** Przypadek użycia dotyczy tworzenia kopii zapasowej programu |
| **Warunki początkowe** |
| 1. Posiada funkcję tworzenia kopii zapasowej 2. Do tworzenia kopii zapasowej jest zrobiona specjalna zakładka w systemie 3. System zapisuje kopię programu w postaci spakowanej do formatu ZIP 4. Użytkownik jest zalogowany do systemu z uprawnieniami administratora 5. System jest otwarty na zakładce utwórz kopię zapasową |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. Użytkownik ma do wyboru dwie opcje:  * Kopia zapasowa bazy danych * Kopia zapasowa bazy danych i programu  1. Użytkownik wybiera kopię zapasową bazy danych i programu 2. System wyświetla okno dialogowe w którym użytkownik może wskazać miejsce zapisu pliku z kopią zapasową 3. Użytkownik wybiera miejsce na dysku, nadaje nazwę dla pliku i wybiera opcję zapisu 4. System tworzy backup bazy danych oraz kopiuje wszystkie pliki programu do pliku tymczasowego, a następnie tworzy z nich plik w formacie ZIP i nadaje mu nazwę którą wskazał użytkownik. 5. System wyświetla informację o udanym zapisie 6. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 1.A Użytkownik wybiera opcję kopii zapasowej bazy danych  1.A.1 System wyświetla okno dialogowe w którym użytkownik może wskazać miejsce zapisu pliku z kopią zapasową  1.A.2 Użytkownik wybiera miejsce na dysku, nadaje nazwę dla pliku i wybiera opcję zapisu  1.A.3 System tworzy backup bazy danych, a następnie tworzy z niego plik w formacie ZIP i nadaje mu nazwę którą wskazał użytkownik.  1.A.4System wyświetla informację o udanym zapisie  6.A System nie może zapisać danych  6.A.1 System informuje o niepowodzeniu akcji tworzenia kopii zapasowej  6.A.2 System zapisuje informacje o przyczynie wystąpienia problemów z zapisem w pliku z logiem błędów. |

Tabela Przypadki użycia- PU27:Przeliczanie czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU27:Przeliczanie czasu pracy |
| **Aktorzy:** Pracownik, Kierownik, Kadrowa |
| **Opis:** Przypadek użycia dotyczy przeliczania czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce czas pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej, kierownika lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla listę pracowników w postaci drzewka 2. Użytkownik wybiera pracownika lub dział przedsiębiorstwa 3. System pokazuje opcję przeliczania czasu pracy wraz z zakresem czasowym od- do 4. Użytkownik wprowadza zakres w którym ma być obliczony czas pracy i wybiera opcję przelicz 5. System przelicza czas pracy w zadanym okresie czasowym 6. W zakładce raport czasu pracy pojawiają się transakcje z opisem oraz sumy czasu pracy o czym system informuje użytkownika. 7. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 6.A Użytkownik wybrał opcję przeliczania czasu pracy dla działu  6.A.1 W zakładce raport czasu pracy pojawiają się transakcje z opisem oraz sumy czasu pracy dla wszystkich pracowników działu o czym system informuje użytkownika  6.A.2 Przypadek użycia się kończy |

Tabela Przypadki użycia- PU28: Przeliczanie czasu pracy zadaniowo

|  |
| --- |
| **Przypadek użycia:** PU28: Przeliczanie czasu pracy zadaniowo |
| **Aktorzy:** Pracownik, Kierownik, Kadrowa, Administrator |
| **Opis:** Przypadek użycia dotyczy dodawania transakcji czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce czas pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami pracownika, kadrowej, kierownika lub administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla listę pracowników w postaci drzewka 2. Użytkownik wybiera pracownika lub dział przedsiębiorstwa 3. System pokazuje opcję przeliczania czasu pracy zadaniowo wraz z zakresem czasowym od - do 4. Użytkownik wprowadza zakres w którym ma być obliczony czas pracy i wybiera opcję przelicz 5. System przelicza czas pracy przeznaczonym na zadania w zadanym okresie czasowym 6. W zakładce raport czasu pracy pojawiają się transakcje z opisem oraz sumy czasu pracy o czym system informuje użytkownika. 7. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 6.A Użytkownik wybrał opcję przeliczania czasu pracy zadaniowo dla działu  6.A.1 W zakładce raport czasu pracy pojawiają się transakcje z opisem oraz sumy czasu pracy dla wszystkich pracowników działu o czym system informuje użytkownika  6.A.2 Przypadek użycia się kończy |

Tabela Przypadki użycia- PU29:Dodawanie transakcji czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU29:Dodawanie transakcji czasu pracy |
| **Aktorzy:** Pracownik, Kierownik, Kadrowa, Administrator |
| **Opis:** Przypadek użycia dotyczy dodawania transakcji czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce czas pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami pracownika, kadrowej, kierownika lub administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla przyciski służące do dodawania transakcji:  * Czas start * Czas stop  1. Użytkownik wybiera odpowiedni rodzaj transakcji 2. System zapisuje transakcję 3. System wyświetla informację o zapisanej transakcji 4. Przypadek użycia się kończy |
| **Rozszerzenia** |
| Brak rozszerzeń |

Tabela Przypadki użycia- PU30:Usuwanie transakcji czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU30:Usuwanie transakcji czasu pracy |
| **Aktorzy:** Kierownik, Kadrowa |
| **Opis:** Przypadek użycia dotyczy usuwania transakcji czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce czas pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej, kierownika |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla listę pracowników w postaci drzewka 2. Użytkownik wybiera pracownika 3. System wyświetla transakcje pracownika oraz przyciski:  * Dodaj * Usuń * Modyfikuj   4. Użytkownik zaznacza transakcję i wybiera opcję usuń  5. System wyświetla komunikat z potwierdzeniem usunięcia transakcji  6. Użytkownik potwierdza usunięcie transakcji  7. System wyświetla informację o usunięciu danych  8. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 6.A Użytkownik chce anulować usuwanie transakcji  6.A.1 Użytkownik wybiera opcję anuluj  6.A.2 System wyświetla główne menu programu |

Tabela Przypadki użycia- PU31:Modyfikacja transakcji czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU31:Modyfikacja transakcji czasu pracy |
| **Aktorzy:** Kierownik, Kadrowa |
| **Opis:** Przypadek użycia dotyczy modyfikacji transakcji czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce czas pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej, kierownika lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla listę pracowników w postaci drzewka 2. Użytkownik wybiera pracownika 3. System wyświetla transakcje pracownika oraz przyciski:  * Dodaj * Usuń * Modyfikuj  1. Użytkownik zaznacza transakcję i wybiera opcję modyfikuj 2. System wyświetla transakcję w nowym oknie z możliwością edycji i zapisu 3. Użytkownik wprowadza zmiany i wybiera opcję zapisu 4. System zapisuje dane i wyświetla informacje o zapisie 5. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 6.A Użytkownik chce anulować modyfikacje transakcji  6.A.1 Użytkownik wybiera opcję anuluj  6.A.2 System wyświetla główne menu programu |

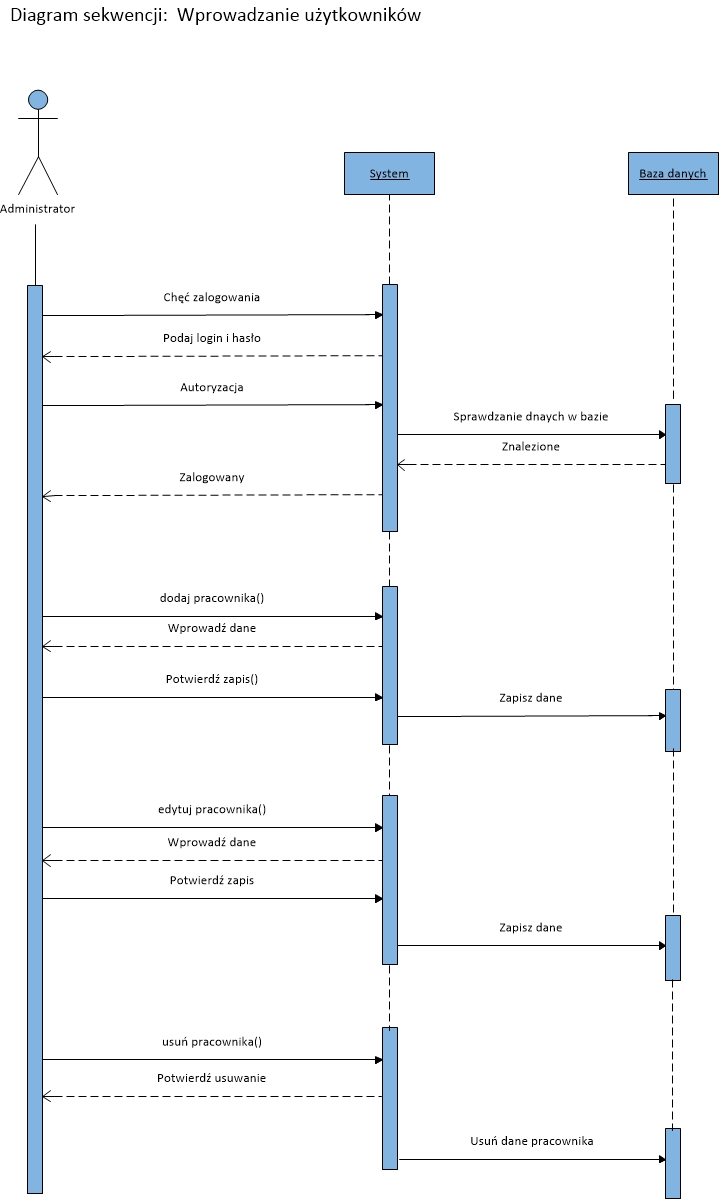
Tabela Przypadki użycia- PU33:Generowanie okresowych raportów czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU33:Generowanie okresowych raportów czasu pracy |
| **Aktorzy:** Pracownik, Kierownik, Kadrowa, Administrator |
| **Opis:** Przypadek użycia dotyczy generowania okresowych raportów czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce raporty 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami pracownika, kadrowej, kierownika lub administratora |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla listę pracowników w postaci drzewka 2. Użytkownik wybiera pracownika lub dział przedsiębiorstwa 3. System wyświetla opcję generowania raportu 4. Użytkownik wybiera opcję generowania raportu 5. System otwiera okno w którym pokazuje się raport czasu pracy 6. Przypadek użycia się kończy |
| **Rozszerzenia** |
| Brak |

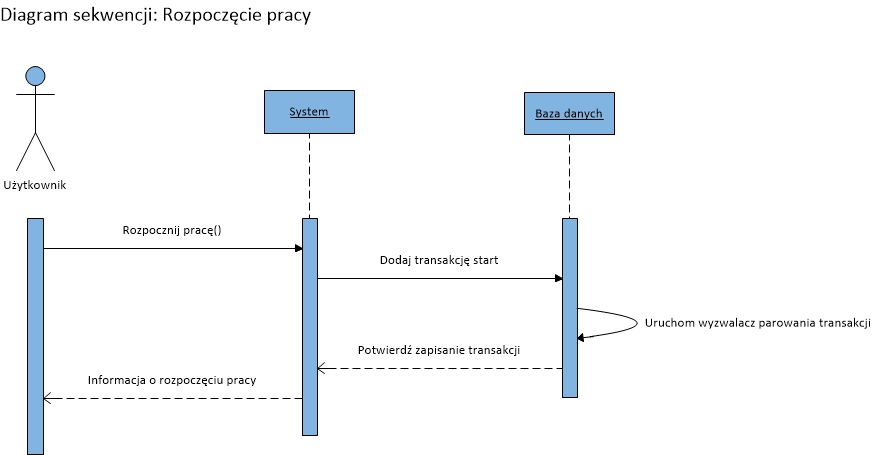
Tabela Przypadki użycia- PU34:Drukowanie raportów czasu pracy

|  |
| --- |
| **Przypadek użycia:** PU34:Drukowanie raportów czasu pracy |
| **Aktorzy:** Kadrowa, Szef |
| **Opis:** Przypadek użycia dotyczy dodawania transakcji czasu pracy |
| **Warunki początkowe** |
| 1. Na ekranie wyświetlony jest graficzny interfejs użytkownika na zakładce raporty, otwartym oknem z wygenerowanym raportem czasu pracy 2. System czeka na aktywność użytkownika 3. Użytkownik jest zalogowany do systemu z uprawnieniami kadrowej lub szefa |
| **Zdarzenie inicjujące** |
| Użytkownik rozpoczyna akcję poprzez żądanie za pośrednictwem GUI |
| **Scenariusz główny** |
| 1. System wyświetla okno z wygenerowanym wcześniej raportem oraz przyciski:  * Drukuj * Anuluj  1. Użytkownik wybiera opcję drukuj 2. System wyświetla okno z dostępnymi drukarkami, ustawieniami drukowania i przyciskami:  * Drukuj * Anuluj  1. Użytkownik wybiera opcję drukuj   5. System drukuje raport  6. Przypadek użycia się kończy |
| **Rozszerzenia** |
| 2.A Użytkownik chce anulować drukowanie raportu  2.A.1 Użytkownik wybiera opcję anuluj  2.A.2 System wyświetla okno z zakładką czas pracy  3.A Użytkownik chce anulować drukowanie raportu  3.A.1 Użytkownik wybiera opcję anuluj  3.A.2 System wyświetla okno z zakładką czas pracy |

### Diagramy sekwencji

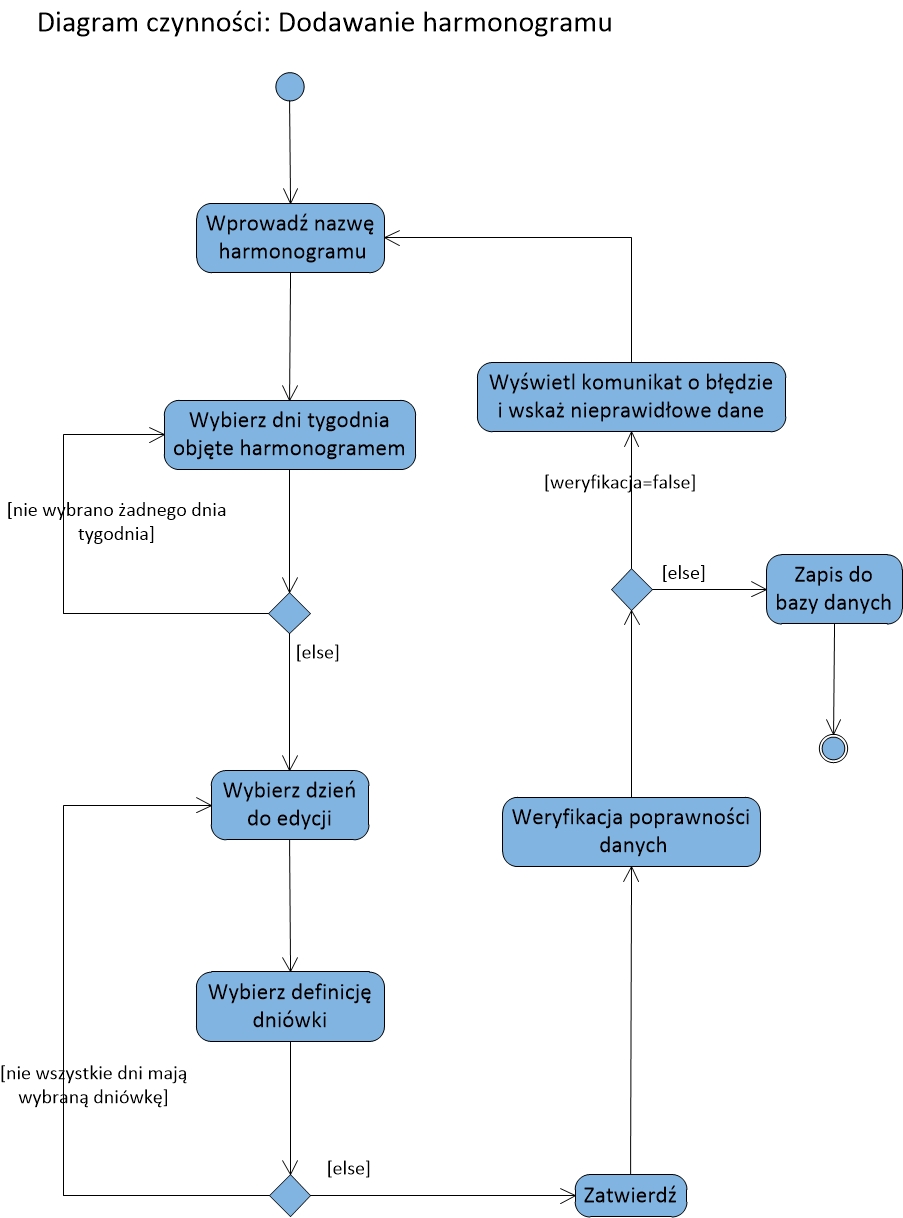


Rysunek Diagram sekwencji - wprowadzanie użytkowników; źródło: opracowanie własne

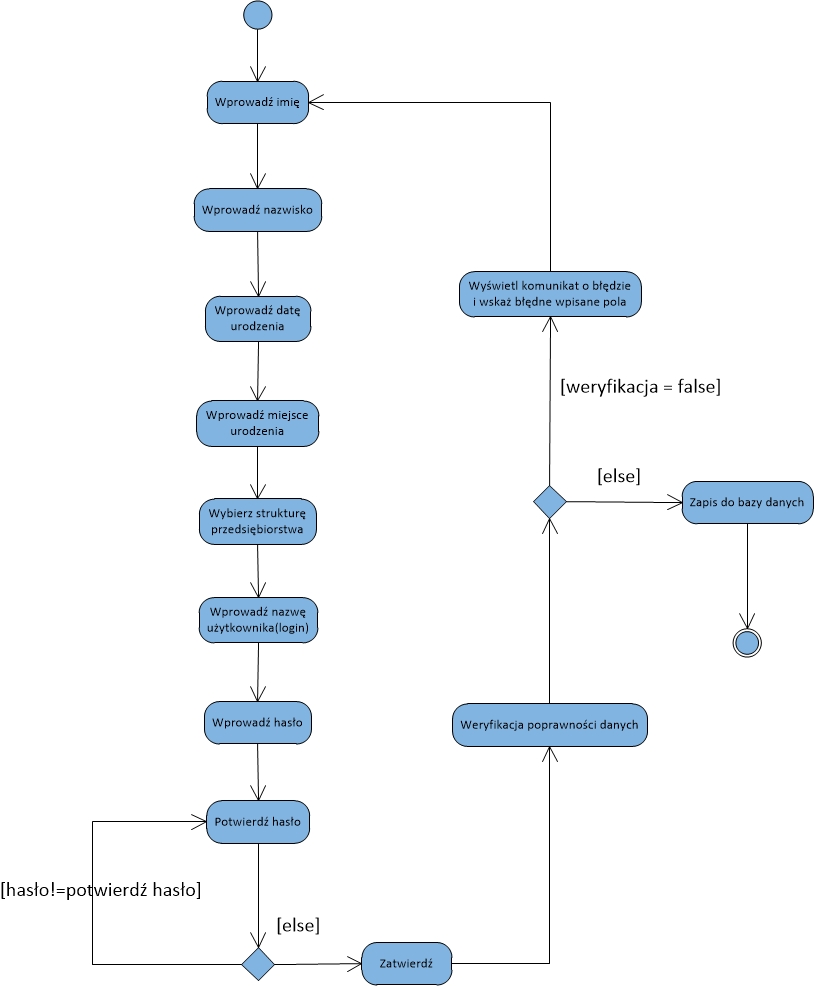


Rysunek Diagram sekwencji - rozpoczęcie pracy; źródło: opracowanie własne

### Diagram czynności



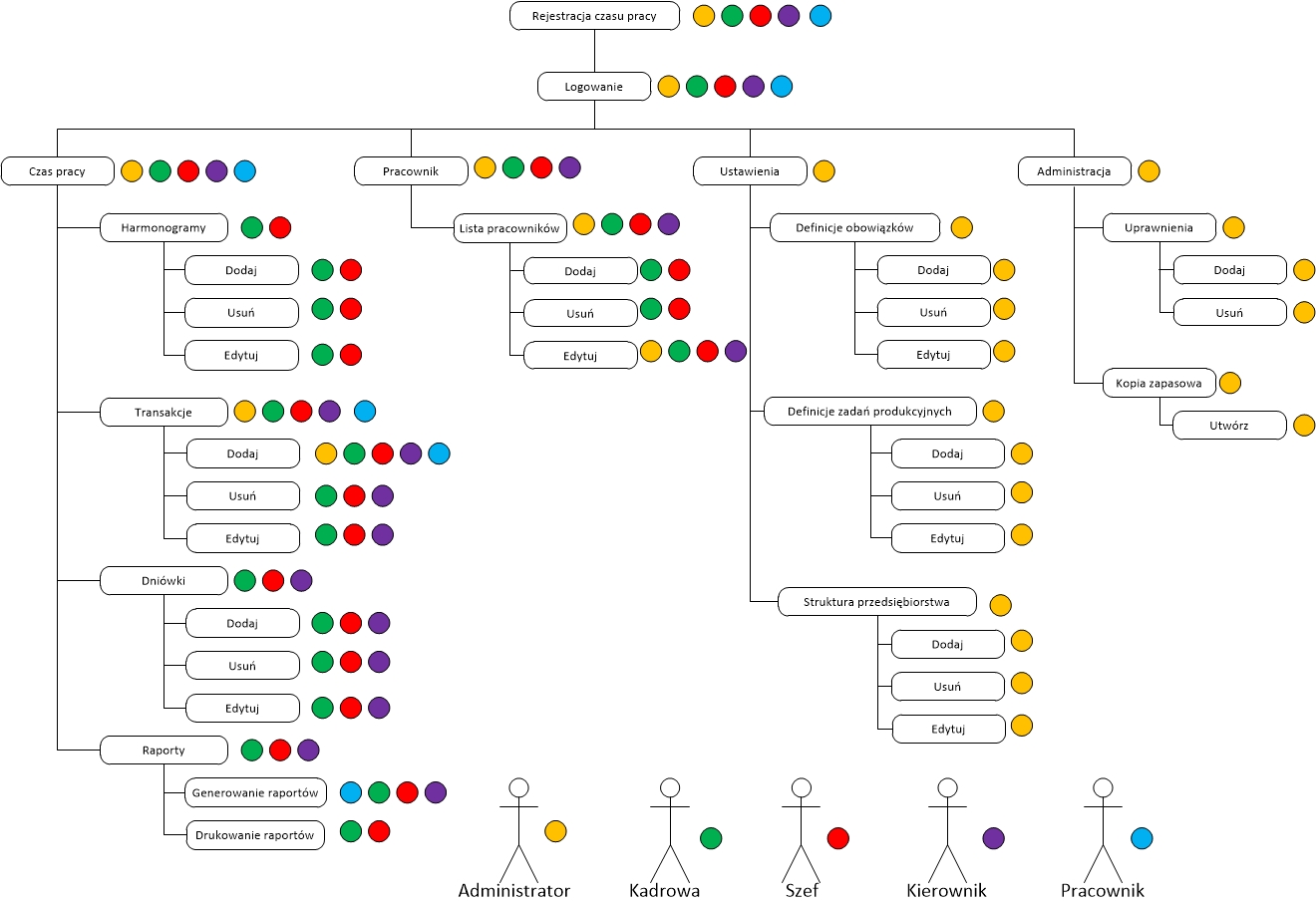
Rysunek Diagram czynności – dodawanie harmonogramu ; źródło: opracowanie własne



Rysunek - Diagram czynności - rejestracja pracownika; źródło: materiały seminaryjne J. Stankiewicz

### Diagram hierarchii funkcji

Diagram hierarchii funkcji nie jest standardowym diagramem języka UML przy podejściu obiektowym, ale ułatwi utworzenie struktury systemu.

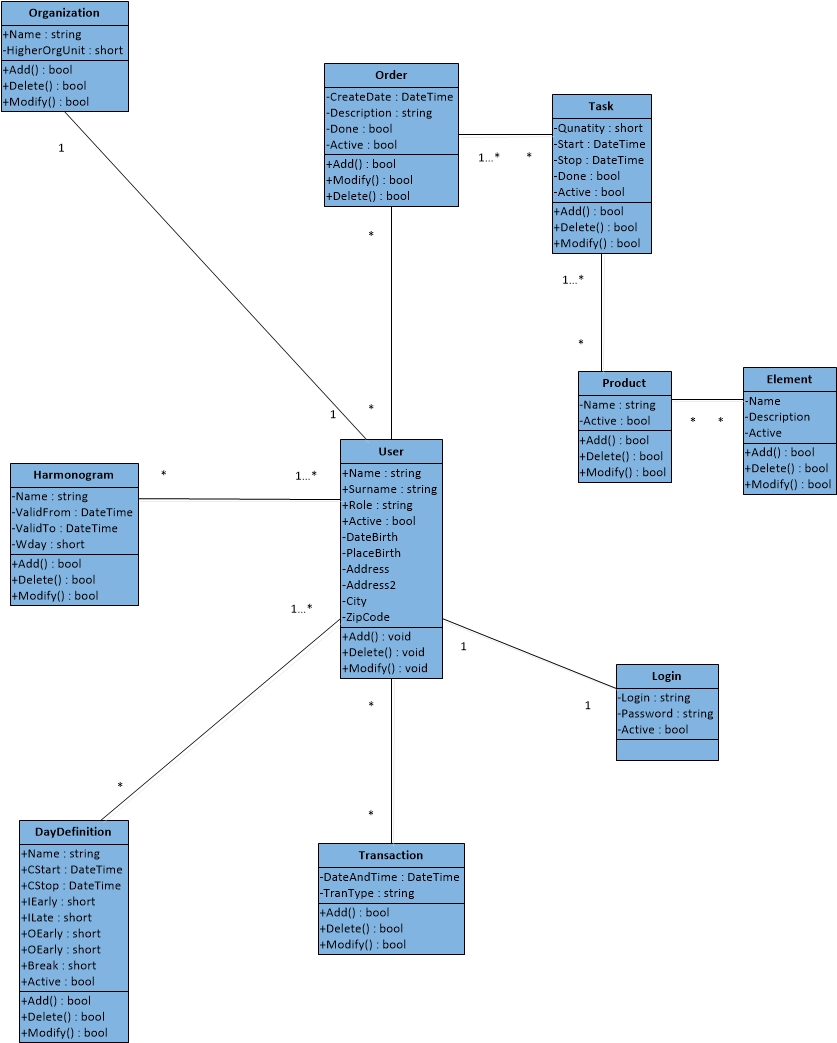


Rysunek Diagram hierarchii funkcji; źródło: opracowanie własne

### 

### Diagram klas

Na podstawie opisu programu i diagramów ustalono następujący diagram klas:



Rysunek Diagram klas; źródło: materiał własny

## Wybór architektury

Na podstawie przeprowadzonej analizy zdecydowano, że system będzie stworzony w oparciu o środowisko .NET firmy Microsoft ponieważ posiada on wszystkie niezbędne mechanizmy i narzędzia umożliwiające zrealizowanie projektu.

Wybrano następujące narzędzia i technologie:

**Microsoft Visual Studio 2015 Enterprise** – jest to zintegrowane środowisko programistyczne obsługujące kilka języków programowania, służące do tworzenia aplikacji konsolowych, jak i aplikacji z graficznym interfejsem użytkownika. Środowisko posiada wbudowany mechanizm IntelliSense który na bieżąco weryfikuje poprawność pisanego kodu.

**Microsoft SQL Server 2016** – jest to system zarządzania bazą danych. Charakterystycznym elemntem produktu jest język zapytań T-SQL który jest rozwinięciem standardu ANSI/ISO.

**Język programowania C# w wersji 5.0** – język ten został utworzony przez firmę Microsoft i jest wspierany przez narzędzie Visual Studio.

**Technologia Entity Framework 6.2** – jest to technologia która umożliwia zarządzanie bazą danych z poziomu kodu aplikacji bez konieczności tworzenia skryptów SQL. Ogólnie technologia opiera się na 3 podejściach:

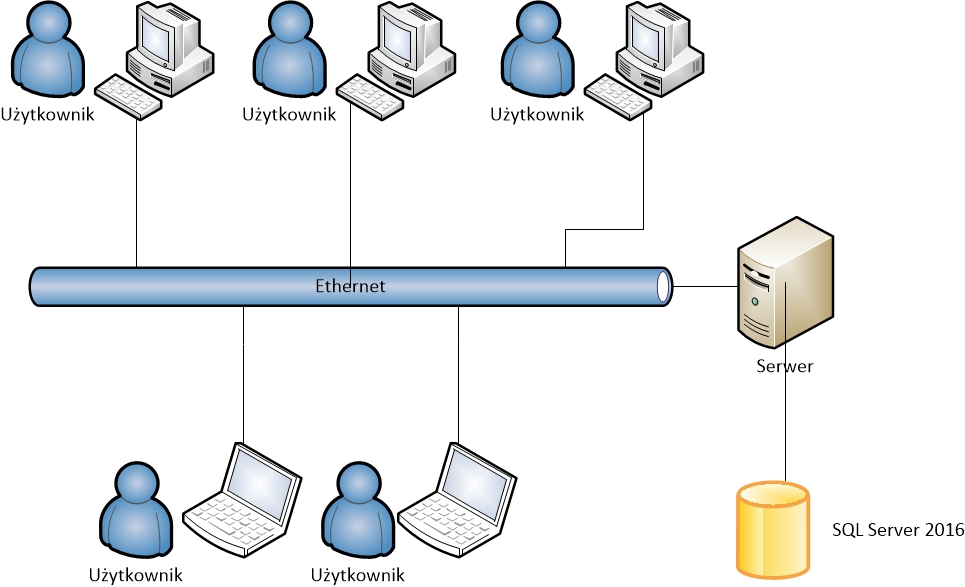
- database first – wykorzystywany zazwyczaj kiedy już jest działająca baza danych i korzystają z niej inne aplikacje,

- code first- tworzona jest baza danych od zera w kodzie aplikacji,

- model first – tworzony jest sam model bazy wraz z relacjami a następnie struktura konwertowana na kod w aplikacji.

Z uwagi na to że model bazy wraz z relacjami został już utworzony, wykorzystane zostanie podejście model- first.

System będzie zbudowany w oparciu o architekturę typu klient-serwer. Jest to asymetryczna architektura oprogramowania rozdzielająca niektóre funkcjonalności, tak aby zwiększyć elastyczność i ułatwić wprowadzenie zmian w każdej z części. W praktyce wygląda to tak że serwer zapewnia usługi dla klientów, którzy komunikują się z serwerem wysyłając żądanie (request).



Rysunek Architektura klient-serwer; źródło: opracowanie własne

# Projektowanie

Rozdział zawiera projekt bazy danych.

## Projekt bazy danych



Rysunek - Diagram struktury bazy danych; źródło: opracowanie własne

### Tabele

W tym rozdziale zostały opisane tabele bazy danych. Każda tabela bazy danych ma dokładnie opisane pole wraz z przeznaczeniem. Schemat połączeń jest widoczny na załączonym powyżej diagramie.

Tabela - Tabela genLogin - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genLogin | | | |
| **Opis** | Tabela przechowuje dane potrzebne do logowania | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| Id | int | - | NIE | Klucz główny |
| IdUser | int | - | NIE | Identyfikator użytkownika. Klucz obcy |
| Login | nvarchar | 20 | NIE | Nazwa użytkownika |
| Password | nvarchar | 20 | NIE | Hasło |
| Try | smallint | - | NIE | Ilość prób logowania |

Tabela - Tabela genUsers - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genUsers | | | |
| **Opis** | Tabela przechowuje podstawowe dane o użytkownikach | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int | - | NIE | Klucz główny |
| Name | nvarchar | 50 | TAK | Przechowuje imię pracownika |
| Surname | nvarchar | 50 | TAK | Przechowuje nazwisko pracownika |
| IdRole | int | - | NIE | Identyfikator uprawnienia |
| Active | int | - | NIE | Przechowuje informacje czy użytkownik jest widoczny/usunięty |

Tabela - Tabela genUserDet - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genUserDet | | | |
| **Opis** | Tabela przechowuje rozszerzone dane o użytkownikach | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int | - | NIE | Identyfikator użytkownika. Klucz obcy |
| DateBirth | datetime | - | TAK | Przechowuje datę urodzin |
| PlaceBirth | nvarchar | 255 | TAK | Przechowuje miejsce urodzenia |
| Adrress | nvarchar | 255 | TAK | Przechowuje adres |
| Adrress2 | nvarchar | 255 | TAK | Przechowuje adres |
| City | nvarchar | 255 | TAK | Przechowuje miasto z adresu |
| ZipCode | nvarchar | 10 | TAK | Przechowuje kod pocztowy |

Tabela - Tabela OrgUnitPriv - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genOrgUnitPriv | | | |
| **Opis** | Tabela przechowuje dane o uprawnieniach do struktur przedsiębiorstwa | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int | - | NIE | Identyfikator użytkownika. Klucz obcy |
| IdOrgUnit | int | - | NIE | Identyfikator struktury przedsiębiorstwa. Klucz obcy |
| Active | bit | - | TAK | Informacja czy uprawnienie jest aktualne |

Tabela - Tabela genOrgUsers - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genOrgUsers | | | |
| **Opis** | Tabela przechowuje dane o | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| Lp | int | - | NIE | Licznik wierszy |
| IdUser | int | - | NIE | Identyfikator użytkownika. Klucz obcy |
| IdOrgUnit | int | - | NIE | Identyfikator struktury przedsiębiorstwa. Klucz obcy |

Tabela - Tabela genRoles - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | genRoles | | | |
| **Opis** | Tabela przechowuje dane o uprawnieniach | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdRole | int | - | NIE | Klucz główny |
| Name | nvarchar | 30 | NIE | Nazwa uprawnienia |
| Active | smallint | - | TAK | Informacja czy uprawnienie jest aktywne |

Tabela - Tabela Calendar - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | Calendar | | | |
| **Opis** | Tabela pomocnicza zawierające dane o kalendarzu | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| DateKey | int | - | NIE | Pole opisuje dane o dniach kalendarza |
| Date | date | 10 | NIE | Pole opisuje dane o dniach kalendarza |
| Day | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| DaySuffix | char | 2 | NIE | Pole opisuje dane o dniach kalendarza |
| Weekday | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| WeekDayName | varchar | 10 | NIE | Pole opisuje dane o dniach kalendarza |
| IsWeekend | bit | - | NIE | Pole opisuje dane o dniach kalendarza |
| IsHoliday | bit | - | NIE | Pole opisuje dane o dniach kalendarza |
| HolidayText | varchar | 64 | TAK | Pole opisuje dane o dniach kalendarza |
| DOWInMonth | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| DayOfYear | smallint | - | NIE | Pole opisuje dane o dniach kalendarza |
| WeekOfMonth | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| WeekOfYear | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| ISOWeekOfYear | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| Month | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| MonthName | varchar | 10 | NIE | Pole opisuje dane o dniach kalendarza |
| Quarter | tinyint | - | NIE | Pole opisuje dane o dniach kalendarza |
| QuarterName | varchar | 6 | NIE | Pole opisuje dane o dniach kalendarza |
| Year | int | - | NIE | Pole opisuje dane o dniach kalendarza |
| MMYYYY | char | 6 | NIE | Pole opisuje dane o dniach kalendarza |
| MonthYear | char | 7 | NIE | Pole opisuje dane o dniach kalendarza |
| FirstDayOfMonth | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| LastDayOfMonth | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| FirstDayOfQuarter | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| LastDayOfQuarter | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| FirstDayOfYear | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| LastDayOfYear | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| FirstDayOfNextMonth | date | - | NIE | Pole opisuje dane o dniach kalendarza |
| FirstDayOfNextYear | date | - | NIE | Pole opisuje dane o dniach kalendarza |

Tabela - Tabela rcpTranType - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpTranType | | | |
| **Opis** | Tabela przechowuje dane o typach transakcji | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdTranType | int |  | NIE | Klucz główny. Identyfiaktor transakcji |
| Name | nvarchar | 20 | TAK | Opis transakcji |

Tabela - Tabela rcpTrancaction - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpTransaction | | | |
| **Opis** | Tabela przechowuje dane o tranckcjach czasu pracy | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdTrans | bigint |  | NIE | Klucz główny. Licznik transakcji |
| DateAndTime | datetime |  | NIE | Czas i data transakcji |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| IdTransType | int |  | NIE | Klucz obcy. Typ transakcji |

Tabela - Tabela rcpPairsOnB- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpPairsOnB | | | |
| **Opis** | Tabela przechowuje dane o parach transakcji początek/koniec przerwy | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| Start | datetime |  | TAK | Data i czas początku przerwy |
| Stop | datetime |  | TAK | Data i czas końca przerwy |

Tabela - Tabela rcpPairsIO - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpPairsIO | | | |
| **Opis** | Tabela przechowuje dane o parach transakcji początek/koniec pracy | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| Start | datetime |  | TAK | Data i czas początku pracy |
| Stop | datetime |  | TAK | Data i czas końca pracy |

Tabela - Tabela rcpPairsBRK - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpPairsBRK | | | |
| **Opis** | Tabela przechowuje dane o parach transakcji …. | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| Start | datetime |  | TAK | Data i czas początku pracy |
| Stop | datetime |  | TAK | Data i czas końca pracy |

Tabela - Tabela rcpHarmoDef- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpHarmoDef | | | |
| **Opis** | Tabela przechowuje dane o definicjach dni w harmonogramie | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdHarmo | int |  | NIE | Klucz obcy. Identyfikator harmonogramu |
| WDay | int |  | NIE | Informacja o dniu tygodnia |
| IdDay | int |  | NIE | Identyfikator dnia |

Tabela - Tabela rcpHarmo- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpHarmo | | | |
| **Opis** | Tabela przechowuje podstawowe dane o harmonogramie | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdHarmo | int |  | NIE | Klucz główny.Identyfikator harmonogramu |
| HarmoName | nvarchar | 30 | TAK | Nazwa harmonogramu |

Tabela - Tabela rcpDay- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpDay | | | |
| **Opis** | Tabela przechowuje definicję dniówki / dnia pracy | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdDay | int |  | NIE | Klucz główny. Identyfikator dnia |
| Name | nvarchar | 30 | TAK | Nazwa dniówki |
| CStart | time | 7 | TAK | Czas rozpoczęcia pracy |
| CStop | time | 7 | TAK | Czas zakończenia pracy |
| IEarly | smallint |  | TAK | Informacja w minutach, ile można przyjść wcześniej do pracy |
| ILate | smallint |  | TAK | Informacja w minutach, ile można przyjść później do pracy |
| OEarly | smallint |  | TAK | Informacja w minutach, ile można wyjść wcześniej z pracy |
| OLate | smallint |  | TAK | Informacja w minutach, ile można wyjść później z pracy |
| [Break] | smallint |  | TAK | Informacja w minutach, ile czasu przerwy przysługuje |
| Active | bit |  | TAK | Informacja czy dniówka jest aktywna |

Tabela - Tabela zadTypes - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadTypes | | | |
| **Opis** | Tabela przechowuje dane o typach zadań produkcyjnych | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdType | int |  | NIE | Klucz główny. Identyfikator typu |
| TypeName | nvarchar | 100 | NIE | Nazwa zadania prodkcyjnego |
| Active | bit |  | NIE | Informacja czy zadanie jest aktywne |

Tabela - Tabela zadTaskProduction- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadTaskProduction | | | |
| **Opis** | Tabela przechowuje dane o wykonywanym zadaniu produkcyjnym | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdTask | int |  | TAK | Klucz główny. Identyfikator zadania |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| Start | datetime |  | NIE | Data i czas rozpoczęcia pracy przy zadaniu |
| Stop | datetime |  | TAK | Data i czas zakończenia pracy przy zadaniu |
| Active | bit |  | TAK | Informacja czy zadanie zostało zakończone |

Tabela - Tabela zadTaskList - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadTaskList | | | |
| **Opis** | Tabela przechowuje szczegółowe dane o zadaniu produkcyjnym | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdTask | int |  | NIE | Klucz główny. Identyfikator zadania |
| IdOrder | int |  | NIE | Klucz obcy. Identyfikator zamówienia |
| IdElement | int |  | NIE | Identyfikator elementu |
| Quantity | int |  | NIE | Ilość elemntów |
| Active | bit |  | TAK | Informacja czy zadanie jest aktywne |
| Done | bit |  | TAK | Informacja czy zadanie jest skończone |

Tabela - Tabela zadProduct - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadProduct | | | |
| **Opis** | Tabela przechowuje dane o produktach | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdProduct | int |  | NIE | Klucz główny. Identyfikator produktu. |
| Name | nvarchar | 150 | NIE | Nazwa produktu |
| Active | bit |  | NIE | Informacja czy produkt jest w ofercie |

Tabela - Tabela zadOrderProduct - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadOrderProduct | | | |
| **Opis** | Tabela przechowuje dane o złożonych zamówieniach produkcyjnych na produkt | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| Id | int |  | NIE | Klucz główny. Identyfikator zamówienia |
| IdOrder | int |  | NIE | Klucz obcy. Identyfikator zamówienia |
| IdProduct | int |  | NIE | Klucz obcy. Identyfikator produktu |
| Active | bit |  | TAK | Informacja czy zamówienie jest aktualne |

Tabela - Tabela zadOrder - źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadOrder | | | |
| **Opis** | Tabela przechowuje dane o zamówieniach | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdOrder | int |  | NIE | Klucz główny. Identyfikator zamówienia |
| CreateDate | datetime |  | NIE | Data utworzenia |
| IdUser | int |  | NIE | Identyfikator użytkownika składającego zamówienie |
| Description | nvarchar | 250 | NIE | Opis zamówienia |
| Done | bit |  | TAK | Informacja czy skończone |
| Active | bit |  | TAK | Informacja czy aktualne |

Tabela - Tabela zadElement- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | zadElement | | | |
| **Opis** | Tabela przechowuje dane o elementach produktów | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdElement | int |  | NIE | Klucz główny. Identyfikator elementu |
| Name | nvarchar | 150 | NIE | Nazwa elementu |
| Description | nvarchar | 200 | NIE | Opis elementu |
| IdProduct | int |  | NIE | Klucz obcy. Identyfikator produktu |
| IdType | int |  | NIE | Klucz obcy. Identyfikator typu |
| Active | bit |  | NIE | Informacja czy aktualne |

Tabela - Tabela rcpUserHarmo- źródło: opracowanie własne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa tabeli** | rcpUserHarmo | | | |
| **Opis** | Tabela przechowuje dane o harmonogramach użytkowników | | | |
| **Nazwa pola** | **Typ** | **Rozmiar** | **Czy puste** | **Opis pola** |
| IdUser | int |  | NIE | Klucz obcy. Identyfikator użytkownika |
| IdHarmo | int |  | NIE | Klucz obcy. Identyfikator harmonogramu |
| ValidFrom | datetime |  | NIE | Data ważności od |
| ValidTo | datetime |  | NIE | Data ważności do |

### Widoki

Baza nie posiada utworzonych widoków.

### Procedury składowane

Baza posiada aktualnie następujące procedury składowane:

### Procedury wyzwalane

Baza posiada procedury wyzwalane:

### Funkcje skalarne/tabularne

W bazie nie stosowano funkcji skalarnych oraz funkcji tabelarycznych.

## Projekt interfejsu

### Ogólna koncepcja interfejsu

### Interfejs głównego formularza

### Interfejs pozostałych formularzy

### System pomocy i komunikatów

# Dokumentacja systemu

Rozdział ten zawiera opis systemu od strony użytkowania i administracji. Pierwsza część jest to instrukcja instalacji programu czyli krok po kroku należy wykonać żeby go uruchomić. W kolejnej części zawarty jest opis wraz z instrukcją obsługi.

## Opis instalacji systemu

Aby zainstalować program na wybranym komputerze, należy w pierwszej kolejności zainstalować serwer bazy danych MS SQL w wersji 2014.Instalacja serwera powinna się odbyć na maszynie typu serwer ponieważ zazwyczaj jest on dostępny dla wszystkich komputerów w domenie i jest zawsze włączony.

Kolejnym krokiem jest uruchomienie pliku instalacyjnego załączonego przez twórcę systemu. Po uruchomieniu pojawi się następujące okienko:

SCREEN

Następnie program instalacyjny przeprowadzi przez proces instalacji pytając na poszczególnych etapach o wybranie wymaganych opcji.

Po zakończeniu procesu instalacji we wskazanym miejscu na dysku pojawi się plik wykonywalny RCPSystem.exe oraz plik konfiguracyjny RCPSystem.exe.Config. W pliku konfiguracyjnym należy wpisać w zaznaczonym miejscu nazwę serwera SQL oraz bazę danych do której ma się podłączyć aplikacja.

SCREEN

Należy pamiętać o tym że baza powinna być pusta ponieważ cała struktura zostanie utworzona przy pierwszym uruchomieniu. Autentykacja SQL odbywa się w trybie domenowym, tak że nie wymaga wpisywania haseł tylko odbywa się na podstawie uprawnień nadanych przez administratora.

## Dokumentacja użytkowa

W przypadku wykonania wszystkich czynności wynikających z opisu instalacji, po uruchomieniu programu powinno pojawić się okno logowania:

SCREEN

Domyślne login i hasło to odpowiednio admin:admin. Login i hasło są przypisane do użytkownika o Id=1 który ma jednocześnie uprawnienia do wszystkich elementów aplikacji. Program nie pozwoli usunąć użytkownika domyślnego jeśli nie będzie wprowadzonego przynajmniej jednego użytkownika z uprawnieniami administratora.

Po poprawnym zalogowaniu uruchomi się program oraz automatycznie moduł użytkownika:

SCREEN

W górnym menu programu znajduje się lista dostępnych modułów programu, zgodnie z nadanymi uprawnieniami.

### Moduł Słowniki

Pracę z programem należy rozpocząć od wprowadzenia danych słownikowych. W tym celu należy wybrać ikonę SCREEN.

SCREEN

Moduł ten posiada pięć zakładek. Każda z zakładek odpowiada za inną funkcję aplikacji.

1. Struktura przedsiębiorstwa – odpowiada za budowanie struktury przedsiębiorstwa która ułatwia pracę z programem.

SCREEN

W tym miejscu można wprowadzić niezależny dział, lub dział podlegający po inny. Jeśli wprowadzamy dział niezależny należy wpisać jego nazwę w pole niezależny i wybrać przycisk Dodaj znajdujący się obok. Jeśli wprowadzamy dział podlegający, należy wpisać nazwę w polu podlegający i wybrać dział nadrzędny z listy rozwijanej. Akcję kończymy wybierając przycisk Dodaj obok listy rozwijanej.

1. Produkty – w założeniu przedsiębiorstwo posiada listę produktów które wytwarza.

SCREEN

Aby dodać produkt do listy należy wybrać przycisk Dodaj. Otworzy się okno dialogowe w którym wpisujemy nazwę produktu i krótki opis.

SCREEN

Po wpisaniu nowego produktu pojawi się on na liście produktów. Następnym krokiem jest dodanie składników które należy utworzyć, żeby produkt był kompletny. W tym celu wybieramy opcję Zarządzaj składnikami. Pojawi się okno:

SCREEN

Zaczynając pracę z programem lista elementów będzie pusta. Po wprowadzeniu nowego elementu, zostanie on dodany do listy. Jeśli kolejny produkt składa się z któregoś z wpisanych wcześniej elementów, można go wybrać z listy. Przy wprowadzaniu elementu należy podać nazwę, symbol oraz ilość.

1. Elementy – jest to miejsce gdzie jest lista wpisanych elementów w procesie dodawania produktów.

SCREEN

Lista ma charakter poglądowy i umożliwia usuwanie elementów jeśli produkt zmienił skład i dany element nie będzie już produkowany. Aby usunąć element należy zaznaczyć go na liście i wybrać przycisk Usuń

1. Słownik zadań produkcyjnych - umożliwia tworzenie zadania produkcyjnego dotyczącego konkretnego elementu:

SCREEN

Po wprowadzeniu nazwy zadania i wybrania przycisku dodaj pojawi się okno dialogowe z listą elementów które nie mają przypisanego zadania produkcyjnego:

SCREEN

Po wybraniu elementu, zostanie utworzone zadanie produkcyjne z przypisanym elementem.

1. Słownik obowiązków – służy do kategoryzacji obowiązków pracowników. Jego celem jest ograniczenie wyśwetlania zadań produkcyjnych do tych które są przypisane pracownikowi.

SCREEN

Wprowadzenie nowego obowiązku odbywa się przez wpisanie nazwy w pole tekstowe i wybranie opcji Zapisz. Po zapisaniu obowiązek pojawi się na liście. Kolejnym etapem jest przypisanie do obowiązku odpowiednich zadań produkcyjnych. W tym celu należy wybrać przycisk Zarządzaj obowiązkami. Pojawi się okno dialogowe:

SCREEN

Aby dodać obowiązek należy zaznaczyć go na liście po lewej stronie i przy użyciu przycisku >> przerzucić na lewą stronę. W odwrotny sposób usuwamy obowiązki z listy.

### Moduł Użytkownicy

Moduł użytkownicy wygląda następująco:

SCREEN

Po lewej stronie znajduje się struktura przedsiębiorstwa w postaci drzewka. W każdym dziale znajdują się podlegający pracownicy. Po wybraniu pracownika na drzewku jego dane pojawiają się na formularzu.

Dodanie nowego pracownika odbywa się przez wybranie przycisku Dodaj. Wszystkie pola formularza zostaną wyczyszczone i umożliwią wpisanie nowych danych.

SCREEN

Przypisanie użytkownika do struktury odbywa się przez wybranie z listy rozwijanej odpowiedniego działu. Obowiązki dodajemy z listy rozwijanej używając przycisku Dodaj.

SCREEN

Usuwanie odbywa się zaznaczając obowiązek na liście i wybieranie przycisku Usuń.

### Dniówki

Dniówka jest to definicja pojedynczego dnia pracy. Każda dniówka posiada zdefiniowany moment rozpoczęcia i zakończenia pracy.

SCREEN

Program umożliwia definiowanie spóźnień i przyjść/wyjść wcześniejszych a także określa długość przysługującej przerwy.

Definiowanie odbywa rozpoczyna się od wybrania przycisku Dodaj. Pola formularza zostaną wyczyszczone co umożliwia wprowadzenie nazwy i definicji nowej dniówki. Po ukończeniu wybieramy przycisk Zapisz. Poprawnie dodana dniówka powinna pojawić się na liście.

### Harmonogramy

Harmonogramy służą do określenia jak pracownik ma przychodzić do pracy. Harmonogram składa się z dni tygodnia i wybranej na dany dzień rodzaju dniówki. Jeśli jakiś dzień jest dniem wolnym od pracy, można zaznaczyć że jest on wolny używając checkbox`a Wolne.

SCREEN

Definiowanie polega na wprowadzeniu nazwy harmonogramu i wybraniu dniówki z listy rozwijanej przypisanej do danego dnia tygodnia. Dane zapisujemy wybierając przycisk Zapisz zmiany.

Poprawnie zdefiniowany harmonogram pojawi się na liście po lewej stronie.

### Transakcje

Dla każdego pracownika jest to najważniejszy moduł ponieważ odpowiada on za logowanie czasu pracy. Moduł ten dzieli się na dwie części. Pierwsza część to zaznaczenie rozpoczęcia pracy ogólnie.

SCREEN

Pracownik ma do wyboru rozpoczęcie i zakończenie pracy, oraz rozpoczęcie i zakończenie przerwy. Każda transakcja jest rejestrowana w bazie danych i wykazywana później na raportach.

Druga część odpowiada za czas pracy przy wybranym zadaniu produkcyjnym.

SCREEN

Lista zadań produkcyjnych ładowana jest zgodnie z przypisanymi obowiązkami które ma pracownik na swojej liście. Następnie pracownik ma możliwość wybrania zadania na liście i rozpoczęcia pracy nad nim przez wybranie przycisku Start. Wybrane zadanie zostanie dodane do jego listy zadań i odznaczona zostanie data i godzina rozpoczęcia. Jak pracownik zakończy pracę nad zadaniem wybiera opcję Zakończ zadanie i odznaczona zostanie godzina zakończenia pracy.

Czas pracy uwzględniony zostanie później w raportach czasu pracy.

TODO:

### Konfiguracja

### Zamówienia

### Raporty czasu pracy

## Testowanie systemu

# Zakończenie

## Podsumowanie

## Wnioski

## Możliwości dalszego rozwoju

# Dodatek A: Słownik terminów użytych w pracy

# Dodatek B: Wykaz rysunków

# Dodatek C: Słownik tabel

# Dodatek D: Wykaz literatury

# Bibliografia