

Projekt elektronicznego dziennika obecności w podstawówce

Spis treści:

1. Moduły w aplikacji
2. Dokładne zdefiniowanie problematyki projektu.
3. Ustalenie tzw. interesariuszy projektu (czyli grupy osób, które są potencjalnymi użytkownikami).
4. Ustalenie jakie są gromadzone dane w organizacji i kto z użytkowników wprowadza je do systemu.
5. Ustalenie rodzajów potrzebnych raportów, które mają być generowane (kto je generuje i kto z nich korzysta).
6. Ustalenie przepływu informacji w środowisku, w którym ma pracować system.
7. Ustalenie uprawnień (praw dostępu) poszczególnych grup użytkowników
8. Wymagania funkcjonalne
9. Wymagania niefunkcjonalne

1. Moduły w aplikacji

Moduły aplikacji będą w postaci zakładki. Każdy rodzaj konta, będzie miał dostęp do określonych zakładek. Możemy wyróżnić trzy widoki dla poszczególnych rodzajów użytkowników:

- a) Widok dyrektora placówki (administratora)
- b) Widok nauczyciela (zwykłego)
- c) Widok wychowawcy danej klasy (kierownik)

Zakładki :

(NAUCZYCIEL)

- sprawdzanie obecności
- zobacz swoje przedmioty (a w nich klasy wypisane)

(ADMINISTRATOR)

- dodaj użytkownika
- edytuj użytkownika

- usuń użytkownika
- dodaj przedmiot
- edytuj przedmiot
- usuń przedmiot
- powiąż uczeń - klasa
- powiąż wychowawca - klasa
- powiąż klasa - przedmioty
- powiąż nauczyciel - przedmiot
- generuj raport

(WYCHOWAWCA)

- zobacz nieobecności
- usprawiedliwianie nieobecności
- generuj raport (dot. klasy)
- wszystkie funkcjonalności nauczyciela

2. Dokładne zweryfikowanie problematyki projektu

Powstająca aplikacja ma na celu usprawnienie wymiany informacji wewnątrz szkoły. Aplikacja ma umożliwiać nauczycielom uzupełnianie informacji dotyczących nieobecności danego ucznia na danych zajęciach, a następnie wychowawca klasy, do której uczeń należy, może zweryfikować tę informację. Do wychowawcy należała będzie także rola powiadamiania rodziców odnośnie nieobecności ich dziecka w szkole. Aplikacja będzie w prosty sposób umożliwiać generowanie takich informacji, a następnie udostępnianie ich rodzicom (po wygenerowaniu doc lub pdf, np w formie maila). Wówczas, jeśli rodzic w jakikolwiek sposób zgłosi do wychowawcy chęć usprawiedliwienia danej nieobecności, to ten będzie posiadał możliwość jej zmiany na usprawiedliwioną.

Administrator systemu będzie posiadał możliwość tworzenia i aktualizowania bazy danych (uczniów, nauczycieli i wychowawców). Z poziomu administratora możliwe będzie także dodawanie harmonogramu zajęć konkretnych klas, powiązań pomiędzy klasą a wychowawcą, czy nauczycielem a przedmiotami - a tych z kolei z konkretnymi klasami.

Nauczyciel, który nie jest wychowawcą dostanie możliwość wpisania nieobecności ucznia na swoich zajęciach (uczeń jest przypisany do danej klasy, a każda klasa ma swojego wychowawcę).

Ponadto przewiduje się dodanie możliwości wpisywania ocen uczniom i zapisywanie ich w bazie danych.

Niezależnie od rodzaju użytkownika, w celu użytkowania systemu konieczne będzie zalogowanie się .

3. Ustalenie tzw. interesariuszy projektu (czyli grupy osób, które są potencjalnymi użytkownikami).

Potencjalnymi użytkownikami są pracownicy szkół podstawowych. Można ich podzielić na:

- dyrektora placówki (administratora)
- nauczyciela (zwykłego)
- wychowawcę danej klasy (kierownik)

4. Ustalenie jakie są gromadzone dane w organizacji i kto z użytkowników wprowadza je do systemu.

~~W systemie przechowywane będą imiona i nazwiska uczniów oraz pracowników szkoły. Wprowadzał je do systemu będzie administrator, w którego rolę wcieli się dyrektor placówki.~~

~~Dodatkowo do uczniów będą przypisane obecności na zajęciach, które będą wpisywane przez nauczycieli i weryfikowane przez wychowawców klas.~~

LEGENDA : - rodzaj gromadzonych danych - OSOBA WPROWADZAJĄCA

- Dane personalne uczniów i nauczycieli - ADMINISTRATOR
- Listę przedmiotów i przypisanie do nich nauczycieli - ADMINISTRATOR
- Lista klas (przypisanie uczniów do konkretnych klas) - ADMINISTRATOR
- Obecności/nieobecności - NAUCZYCIEL
- Zmiana statusu nieobecności (z nieusprawiedliwionej na usprawiedliwioną) - WYCHOWAWCA

Domyślnie dostęp do gromadzonych danych (wszystkich) posiada dyrektor placówki, jako jej administrator. Nauczyciel ma dostęp do danych, które wprowadza (nieobecności), ale także do imion i nazwisk uczniów(dane podstawowe). Oczywiście nauczyciel ma także podgląd prowadzonych przez siebie przedmiotów. Natomiast wychowawca danej klasy, ma dostęp do tych samych danych co nauczyciel, oraz dodatkowo jego dane o uczniach są uszczegółowione(tj. adres, nr telefonu rodziców, data urodzenia).

5. Ustalenie rodzajów potrzebnych raportów, które mają być generowane (kto je generuje i kto z nich korzysta).

Raporty może generować wychowawca dla każdego ucznia z osobna. Zawierać on będzie tygodniowe bądź miesięczne podsumowanie wszystkich obecności (i nieobecności) ucznia w wybranym okresie.

6. Ustalenie przepływu informacji w środowisku, w którym ma pracować system.

Pierwsze informacje do systemu -> bazę danych, będzie wprowadzał administrator. Następnie te informacje będą dostępne zarówno dla niego, jak i dla nauczycieli prowadzących dany przedmiot (jednak w ograniczonej formie - nauczyciel widzi tylko swoje przedmioty, bądź klasy w jakich naucza).

Nauczyciel ma możliwość zaznaczenia nieobecności danego ucznia, i ta informacja zapisana zostaje w systemie, aby mógł obejrzeć ją wychowawca danej klasy, i ewentualnie usprawiedliwić, co także widoczne będzie w dzienniku obecności (a także w generowanym raporcie).

7. Ustalenie uprawnień (praw dostępu) poszczególnych grup użytkowników

Admin: może wprowadzać nowych użytkowników (uczniów i nauczycieli). Dodatkowo może dodawać plan zajęć oraz ustalać wychowawców klas i przypisywać nauczycieli do przedmiotów.

Nauczyciel: ma podgląd do prowadzonych przez siebie przedmiotów i poszczególnym uczniom może zmieniać status obecności na swoich zajęciach.

Wychowawca: jego uprawnienia będą takie jak nauczyciela i dodatkowo zostaną rozszerzone o podgląd jego wychowanków, z możliwością usprawiedliwienia nieobecności.

8. Wymagania funkcjonalne

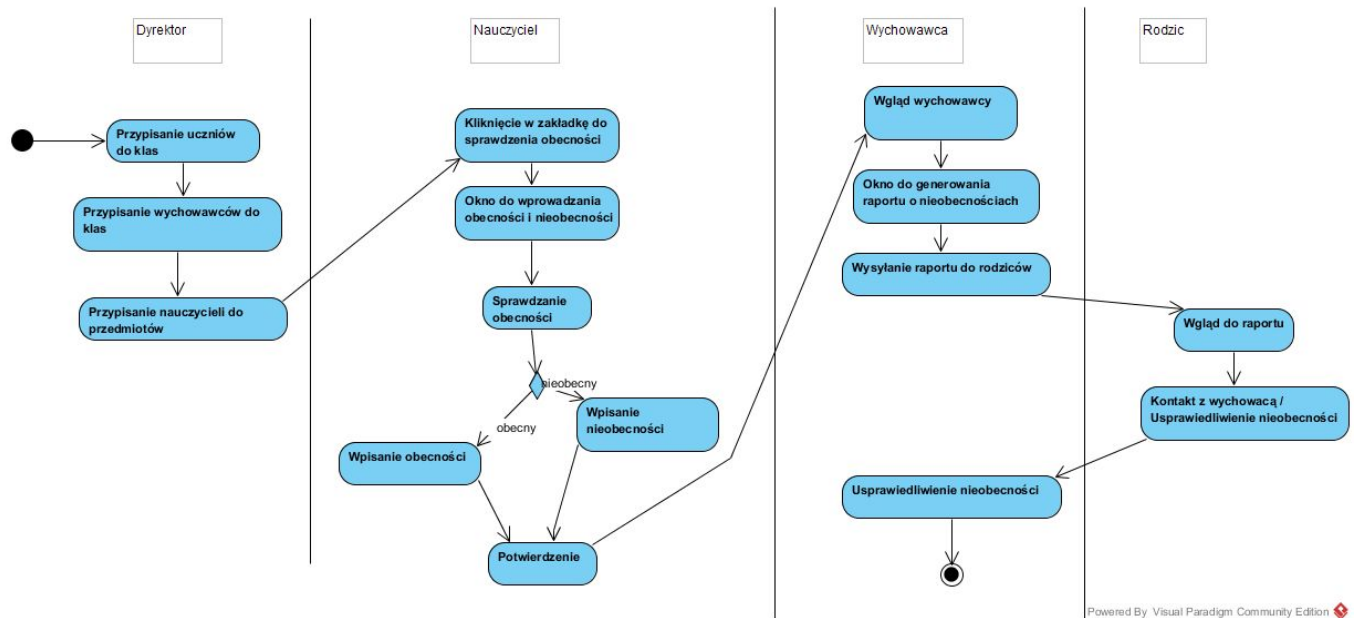
System ma pełnić funkcję dziennika obecności (i ocen), uczniów szkół podstawowych. Podstawowe funkcje konkretnych użytkowników:

- nauczyciele - mogą wpisywać obecności uczniów na zajęciach (oraz wystawiać im oceny). Funkcja wpisania tych danych będzie możliwa tylko podczas obecnie trwających zajęć, a każdy nauczyciel ma możliwość ustawienia obecności tylko dla przedmiotów i klas, których uczy.
- wychowawcy - są to zwykli nauczyciele, jednak mają dodatkową zakładkę, w której mogą zobaczyć podsumowanie obecności klasy, której są wychowawcami, a także wygenerować raporty dla każdego ucznia.
- dyrektor placówki (admin) - jego zadaniem będzie tworzenie nowych użytkowników, a także przypisywanie ich do klas i przedmiotów, które również będą przez niego tworzone. Oprócz tego admin ma dostęp do wszystkich danych w bazie danych.

9. Wymagania niefunkcjonalne

Aplikacja zostanie napisana w języku Java, za pomocą programu NetBeans. Do stworzenia wyglądu wykorzystana zostanie technologia JavaFX.

Program posiadać będzie relacyjną bazę danych MySQL, która będzie obsługiwana pakietem XAMPP.



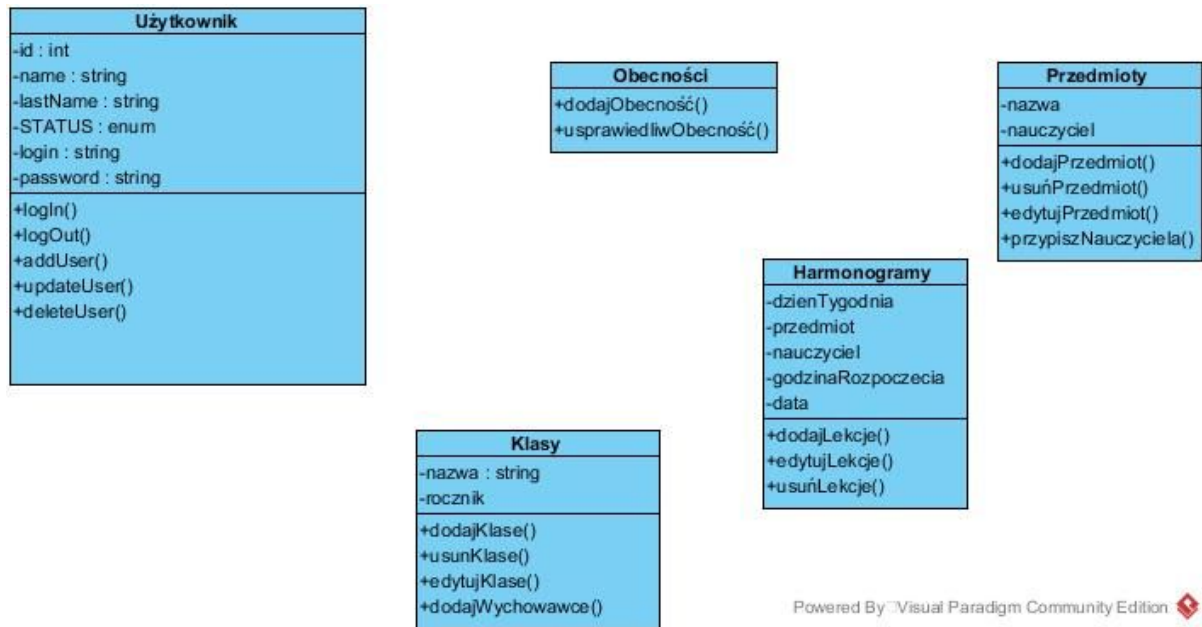
Aplikacja będzie działać na systemach Windows w wersjach: 7, 8, 8.1 oraz 10. Do uruchomienia potrzebny będzie dodatkowo pakiet Java Runtime Environment (JRE) przynajmniej w wersji 8.

Do samej obsługi programu potrzebna będzie klawiatura i myszka. Będzie nimi można manipulować interfejs graficzny i uzupełniać go o odpowiednie dane

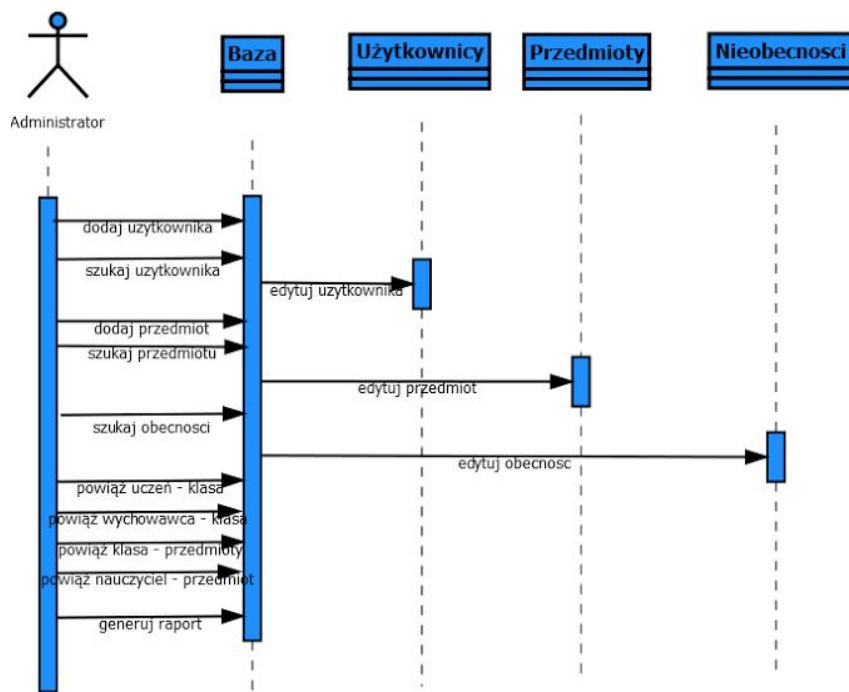
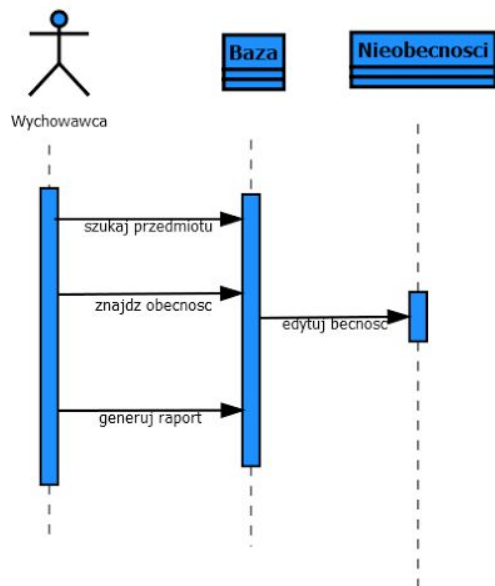
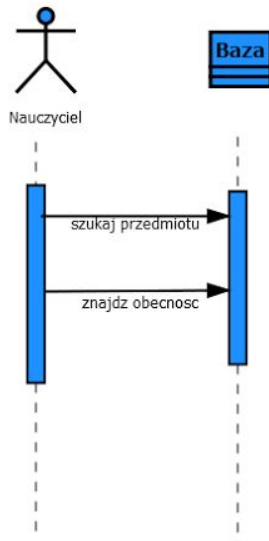
Visual Paradigm

[Diagram przypadków użycia](#) - w formie linku, bo tu nie widać ;)

Diagram klas :



-diagramy sekwencji (zdarzenia, komunikacja . kiedy są utworzone obiekty)



- diagramy klas (ogólne - nie to co jako klasa w programie)
- klasa użytkownik. klasa kierownik (bez pól, metod) - między nimi zależności
- diagram aktywności
- wykresy Gantta

diagram aktywności:

(zostaje)

schemat pokazujący dane wewnątrz aplikacji (ktoś klika- ma okienko)

algorytm kiedy pojawia się coś nowego, itd

klas:

cały program, okno, w oknie zakładki - a tam konkretne moduły

używać agregacji częściowej i całkowitej(który istnieje sam, który potrzebuje istnienia innych do życia)

struktura graficzna w klasach

przypadków użycia:

zamiast include , to extend

include jest wtedy kiedy w kodzie są wspólnie używane części(trzeba je wyciągnąć i tak to oznaczyć)

przy tworzeniu harmonogramu nie zaznaczać zależności z klasami

sekwencji:

system, baza danych (zamiast konkretnych tabel)

przy tworzeniu użytkownika obiekt dajemy niżej, żeby było widać że są one tworzone powiadomienia (nauczyciel, po środku system, na końcu rodzic - usprawiedliwienia)

git:

atlassian.com (getting started)

git init

clone

config

add

commit

stash(opcjonalnie)

status

log

checkout

.gitignore

remote

fetch

pull

push

branch

merge

revert

reset

clean

commit --amend ?