

POLITECHNIKA KRAKOWSKA im. T. KOŚCIUSZKI

WYDZIAŁ INŻYNIERII ELEKTRYCZNEJ I KOMPUTEROWEJ

Raport I z przedmiotu:

**Inżynieria programowania**

Temat projektu:

**Wypożyczalnia sprzętu muzycznego**

**CIEPIELA ADRIAN**

**WĄTOR MICHAŁ**

**Semestr 5**

**Rok akademicki 2016/2017**

# Wstęp

## Cel i zakres projektu:

Celem projektu jest zaprojektowanie i zaimplementowanie aplikacji desktopowej, która będzie umożliwiała obsługę i zarządzanie wypożyczalnią sprzętu muzycznego. Aplikacja będzie posiadała funkcjonalności odpowiadające za pracę zarówno administracji sklepu jak i klienta. Funkcjonalność oferowana przez aplikację będzie wystarczająca, aby klient był w stanie w prosty i intuicyjny sposób wypożyczyć sprzęt muzyczny z poziomu aplikacji, bez konieczności dodatkowego kontaktowania się z pracownikami wypożyczalni.

# Słownik danych

**Account** – konto w aplikacji przypisane do konkretnego uzytkownika, zapewniające funkcjonalność zależną od rodzaju konta (konto użytkownika, konto administratora)

**Admin** – osoba posiadająca uprawnienia administratora w aplikacji

**Client** – osoba korzystająca z aplikacji w celu wypożyczenia sprzętu muzycznego

* + id – klucz główny, unikatowy numer klienta, składający się z maksymalnie 10 cyfr
  + firstName – imię podane przez klienta, maksymalnie 20 liter
  + lastName – nazwisko podane przez klienta, maksymalnie 30 liter
  + address – adres klienta, maksymalnie 50 znaków

**Equipment** – sprzęt muzyczny przeznaczony do wypożyczenia

* + id – klucz główny, unikatowy numer instrumentu, składający się z maksymalnie 10 cyfr
  + type – typ instrumentu
  + brand – marka instrumentu
  + model – model instrumentu
  + yearOfProduction – rok produkcji instrumentu
  + pricePerDay – opłata dzienna za wypożyczenie instrumentu
  + description – opis instrumentu

**Order** – zamówienie, realizacja konkretnej rezerwacji

**Registration** – proces tworzenia konta w aplikacji

**Reservation** – rezerwacja sprzętu dokonywana przez klienta

* + id –klucz główny, unikatowy numer rezerwacji, składający się z maksymalnie 10 cyfr
  + startDate – data rozpoczęcia wypożyczenia
  + durationDays – długość wypożyczenia, składająca się z maksymalnie 3 cyfr
  + typeOfReception – typ utworzonego rachunku, składający się z maksymalnie 20 znaków
  + typeOfReturn – sposób zwrotu wypożyczonych instrumentów
  + paymentMethod – sposób płatności
  + ClientId – klucz obcy

# Analiza biznesowa

Przegląd stron internetowych krakowskich wypożyczalni sprzętu muzycznego pokazuje brak aplikacji, która umożliwiałaby dostęp do różnorodnego sprzętu w jednym miejscu. Lokalne wypożyczalnie dają możliwość dostępu jedynie do pewnego zestawu sprzętu. Na przykład:

* wypożyczalnia umożliwia dostęp tylko do sprzętu typowo estradowego/nagłośnieniowego. Brakuje możliwości wypożyczania samych instrumentów,
* wypożyczalnia umożliwia dostęp do mocno ograniczonego zestawu poszczególnych rodzajów sprzętu, np. tylko jedna perkusja, tylko jeden zestaw kolumn estradowych. Brakuje możliwości wybrania sprzętu dostosowanego ściśle do naszych potrzeb.

Drugim istotnym brakiem w aplikacjach krakowskich wypozyczalni jest brak interfejsu umożliwiającego dokonywanie zamówień. Strony te umożliwiają jedynie obejrzenie dostępnego asertymentu oraz zapoznanie się z jego cennikiem. Nie pozwalają jednak na dokonie samej akcji wypożyczenia sprzętu, ani dalszych związanych z tym rzeczy:

* wybór okresu wypożyczenia,
* ilości instrumentów,
* sposobu odbioru / zwrotu sprzętu.

Wszystkie te czynności zmuszeni jesteśmy ustalać poprzez kontakt telefoniczny lub mailowy. W dzisiejszych czasach, gdzie przyzwyczajeni jesteśmy, że zakupy możemy zrobić za pomocą kilku kliknięć myszką, takie braki w funkcjonalnościach są istotnymi niedogodnościami, które mogą mieć duży wpływ na prosperowanie tych usług na rynku.

# Analiza ryzyka oraz ograniczenia

Aplikacja do poprawnej pracy wymaga stałego połączenia z internetem co definiuje pierwsze i główne ograniczenie. Bez stałego dostępu do połączenia sieciowego, aplikacja nie będzie w stanie zapewnić swoich podstawowych funkcjonalności.

Istotną rzeczą jest też skuteczne rozgraniczenie funkcji administracji sklepu od funkcji z których korzysta klient. Dostęp do funkcji administracyjnych możliwy bedzie tylko po zalogowaniu się na odpowiednie, stworzone do tych celów konto. Osoba nie posiadająca dostepu do tego konta nie będzie mogła korzystać z żadnych dostarczanych przez nie funkcjonalności. Dzieki temu podejściu otrzymamy zabezpieczenie kluczowych danych (np. danych osobowych klientów) przed niepożądanym ujawnieniem oraz zablokujemy możliwość ich modyfikacji przez nieuprawnione do tego osoby.

Ponadto należy zabezpieczyć sie przed zmianami które są uprawnione ale niepoprawne. Nie jest możliwa całkowita kontrola nad tego typu błedami, natomiast można zapewnić, że dane zawsze są przetwarzana zgodnie z założeniami. Trzeba zadbać, aby wprowadzenie błędnych danych nigdy nie mogło spowodować zatrzymania działania aplikacji lub niestabilną jej pracę. Aby to osiągnąć należy uwzględnić w programie sprawdzanie zgodności typów oraz rozmiaru wszystkich wprowadzanych danych.

Aplikacja musi się dobrze skalować. Powinna zawsze, niezależnie od ilości jednocześnie używających jej osób działać płynnie. Aby to zapewnić serwer, którego używa aplikacja musi działać na sprzęcie, który umożliwi płynne działanie aplikacji przy różnych poziomach obciążenia serwera.

Należy dodatkowo zabezpieczać dane aplikacji wykonując ich regularną archiwizację. Działanie to powinno zapobiec możliwości całkowitej utraty swoich danych, lub czasowemu braku dostępu do nich w przypadku awarii serwera.

# Charakterystyka użytkowników:

## administracja sklepu

* + zajmuje się realizacją zamówień:

- przygotowanie sprzętu do wydania,

- kontrola dokonania opłaty przez klienta,

* + pracuje w magazynie,
  + program obsługuje myszką i klawiaturą,
  + posiada wiedzę na temat obsługi aplikacji,
  + posiada wiedzę na temat posiadanego sprzętu muzycznego,
  + udziela informacji na temat dostępnego sprzetu klientom,
  + dba o stan instrumentów.

## klient

* + wybiera interesujący go sprzęt muzyczny,
  + określa okres wypożyczenia,
  + dokonuje opłaty,
  + odbiera sprzęt,
  + po upływie czasu wypożyczenia, zwraca sprzęt.

# Główne funkcje produktu:

## a) funkcje obsługiwane przez administrację:

* + dodawanie sprzętu,
  + ustalanie dostępności sprzętu,
  + ustalanie cen.

## b) funkcje obsługiwane przez klienta:

* + przeglądanie dostępnego sprzętu,
  + zamówienie konkretnych pozycji,
  + określenie ilości wypożyczanego sprzętu,
  + wybór okresu na jaki wypożyczony będzie sprzęt,
  + wybór sposobu odbioru sprzętu,
  + wybór sposobu oddania sprzętu,
  + określenie sposobu płatności,
  + dokonywanie rezerwacji sprzętu.

# Wymagania funkcjonalne i niefunkcjonalne

**Wymagania funkcjonalne:**

* + wybór rodzaju sprzętu,
  + wybór ilości sprzętu
  + określanie długości czasu wypożycznia,
  + określenie terminu rozpoczęcia wypożyczenia
  + wyświetlanie informacji o sprzęcie,
  + wyświetlanie podsuwowania zamówienia,
  + możliwość uzyskania dodatkowych informacji (kontakt klienta z administracją wypożyczalni),
  + określenie sposobu płatności,
  + wprowadzanie nowych pozycji,
  + aktualizacja informacji o sprzęcie,
  + aktualizacja informacji o dostęponości sprzętu,
  + wprowadzanie i aktualizacja cen.

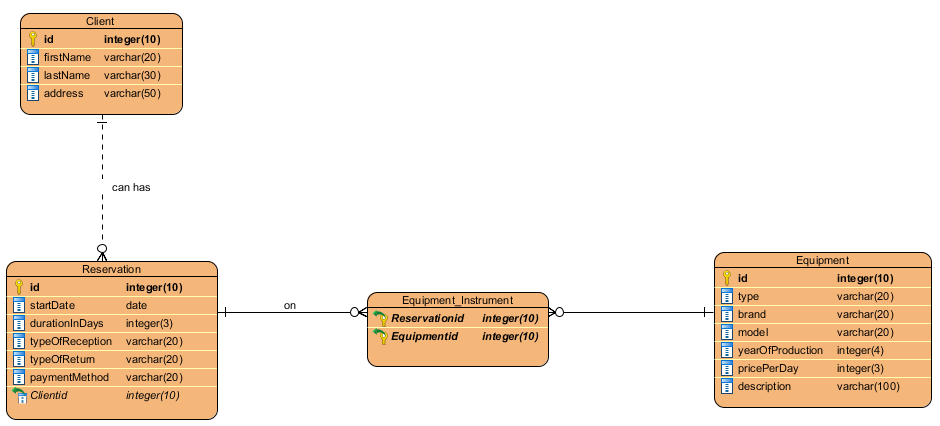
**Wymagania niefunkcjonalne:**

* + przydatność: wzrost efektywności i szybkości pracy wypożyczalni,
  + bezpieczęństwo: oddzielny panel aplikacji dla admina – zapezpieczenie konta przez hasło,
  + dostępność: system musi być dostępny bez przerwy   
    w godzinach pracy wypożyczalni, prace administracyjne będą odbywać się po godzinach funkcjonowania wypożyczalni,
  + wydajność: aplikacja działa sprawnie i bez długiego oczekiwania na reakcje nawet przy dużej ilości użytkowników,
  + skalowalność: system powinien łatwo dać się rozszerzać   
    i mieć możliwość zmiany oraz wprowadzanie nowych funkcji,
  + platforma: oprogramowanie - system Windows, Linux,
  + integracja z innymi systemami: aplikacja współpracuje   
    z systemem bazodanowym.

# Diagramy operacyjne

## Diagram ERD

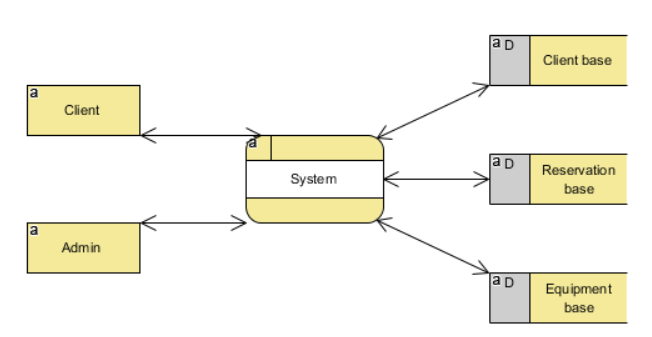
ERD (Entity-Relationship Diagram, diagram związków encji) przedstawia modelowanie danych w systemie.



## Diagram DFD

DFD (Data Flow Diagram, diagram przepływu danych) - graf, którego węzły są zwykle procesami, a łuki przepływami danych. Opisuje transformacje danych wewnątrz systemu

**Kontekstowy**



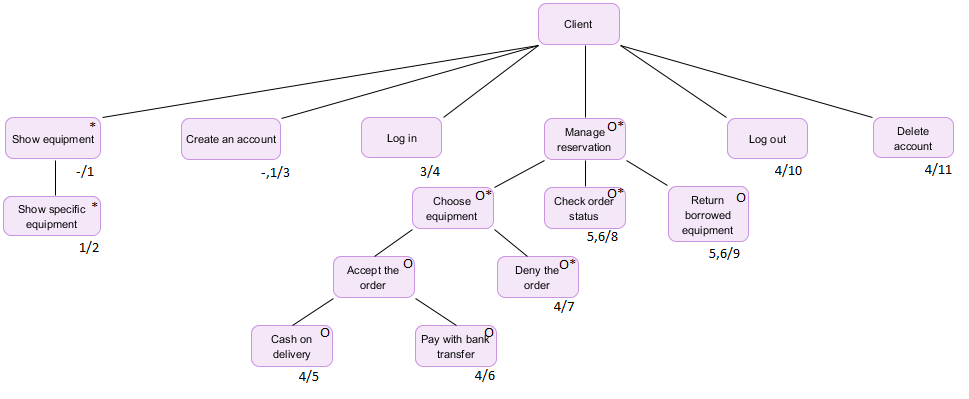
## Systemowy

### 

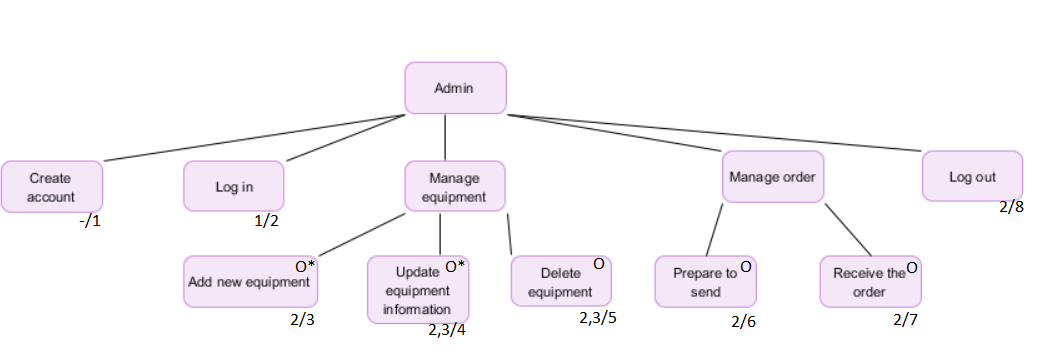
## Diagram ELH

ELH (Entity Life History, diagram historii życia obiektu) – ukazuje zmiany stanu danych w czasie.

## klient



## admin



## Diagram STD

STD (state diagram, diagram stanów) – pokazuje możliwe stany obiektu.

