|  |  |
| --- | --- |
|  | **AKADEMIA GÓRNICZO-HUTNICZA** |

Dokumentacja do projektu

**Stock Management System**

z przedmiotu

**Języki programowania**

Elektronika, I rok

*Łukasz Piasecki*

Wtorek 12:50

prowadzący: Rafał Frączek

28.05.2020

# Opis projektu

Oprogramowanie ma za zadanie monitorowanie poziomu zapasów, zamówień, dostaw. Celem programu jest usystematyzowanie i ułatwienie pracy z zasobami, oraz wszczęcia alarmu w przypadku przekorczenia limitów. Może być wykorzystywany aby uniknąć nadmiaru produktów i przestojów w firmie.

# Project description

The software is designed to monitor the level of stocks, orders, deliveries. The purpose of the program is to systematize and facilitate work with resources, and to raise an alarm if limits are exceeded. It can be used to avoid excess products and downtime in the company.

# Zanim włączysz Stock Management System

Program korzysta z systemu do zarządzania bazami danych MySQL, tym samym przed rozpoczęciem użytkowania programu konieczne jest zainstalowanie MySQL Connector/NET. Konieczne jest posiadanie platformy Visual Studio:

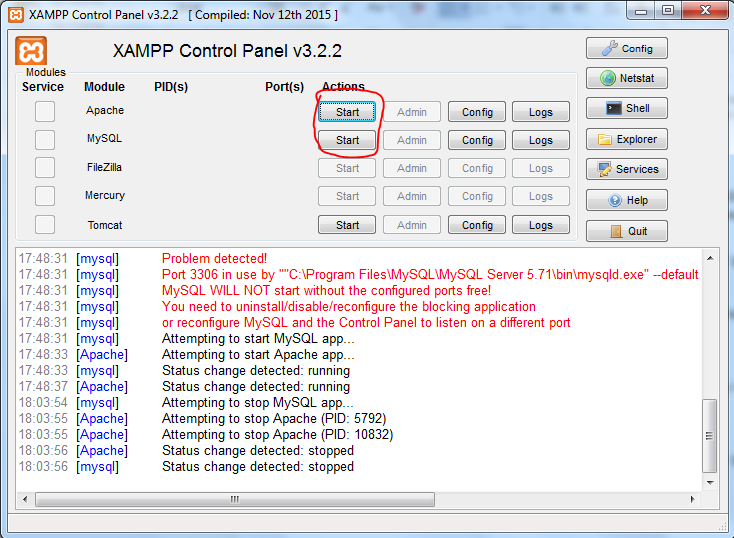
1. Jeśli posiadasz Connector / NET 6.6 lub starszy, należy usunąć go przed zainstalowaniem MySQL.
2. Instalacja MySQL

* Pobierz i uruchom [instalator MySQL](https://dev.mysql.com/downloads/installer/) . (W przypadku trudności zobacz [Instalator MySQL dla systemu Windows .](https://dev.mysql.com/doc/refman/8.0/en/mysql-installer.html)
* Postępuj zgodnie z poleceniami instalatora.

1. Program korzysta za bazy danych zlokalizowanych na dysku lokalnym, więc do obsługi i kontroli potrzebujesz oprogramowania XAMPP:

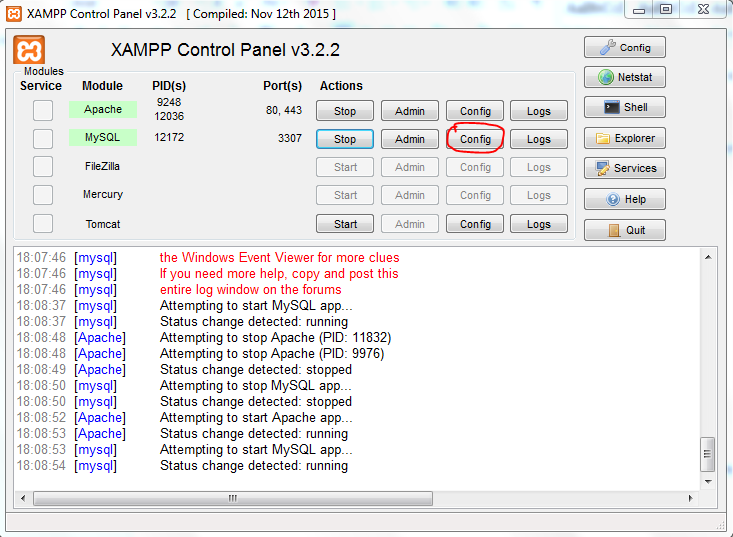
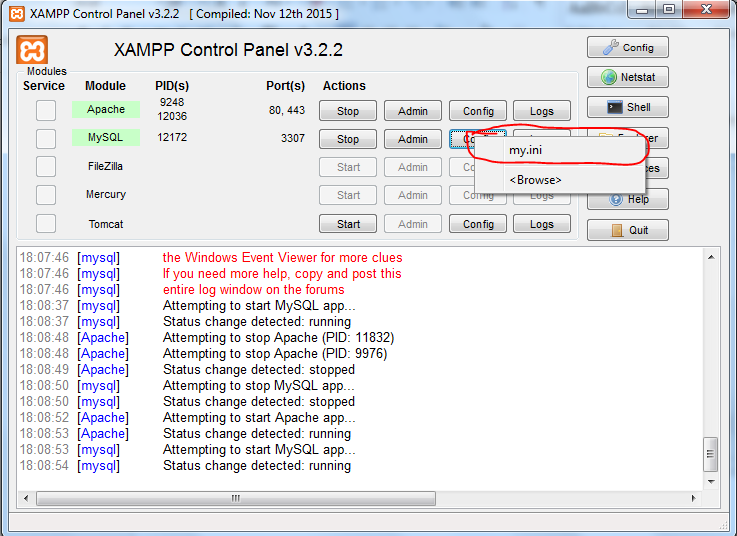
* Pobierz i zainstaluj [pakiet XAMPP](https://www.apachefriends.org/pl/download.html).

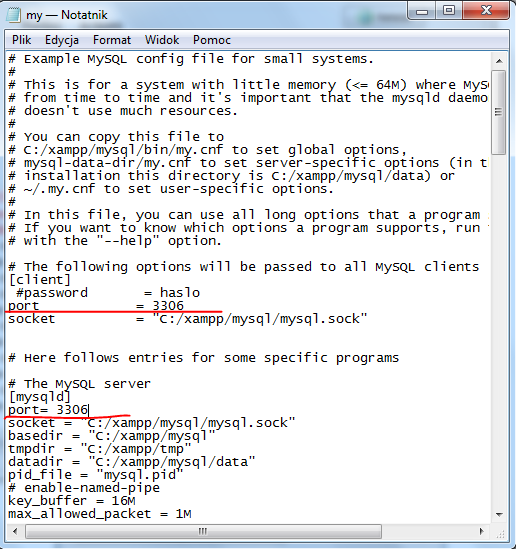
1. Uruchom xampp-control.exe, który znajdziesz w folderze głównym, w którym zainstalowno pakiet XAMPP.
   * Uruchom server Apache oraz usługę MySQL



* + W przypadku problemów z połączeniem z lokalną bazą danych zmień port na 3307

Klikamy Config 🡪 my.ini 🡪 zmień wartość 3306 na 3307.



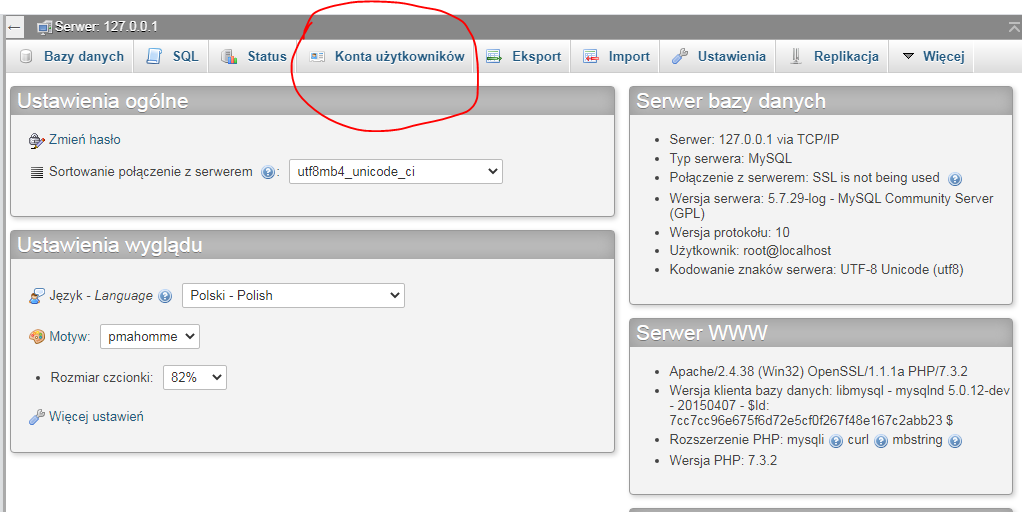
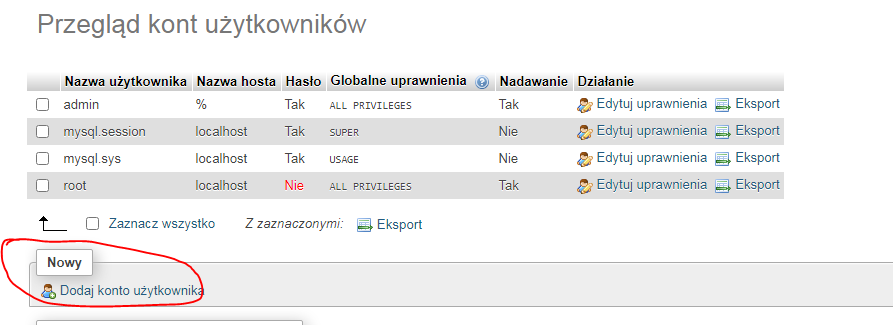
* + Kliknąć przycisk „Admin” w oknie XAMPP Control Panel lub wejść na [server lokalny](http://localhost/phpmyadmin/).

Login: root

Hasło: (nie wpisuj nic)

* + Tworzymy nowe konto urzytkownika:

Klikamy „konta użytkowników” -> „dodaj konto urzytkownika”

* + Tworzymy urzytkownika:

Nazwa urzytkownika : admin

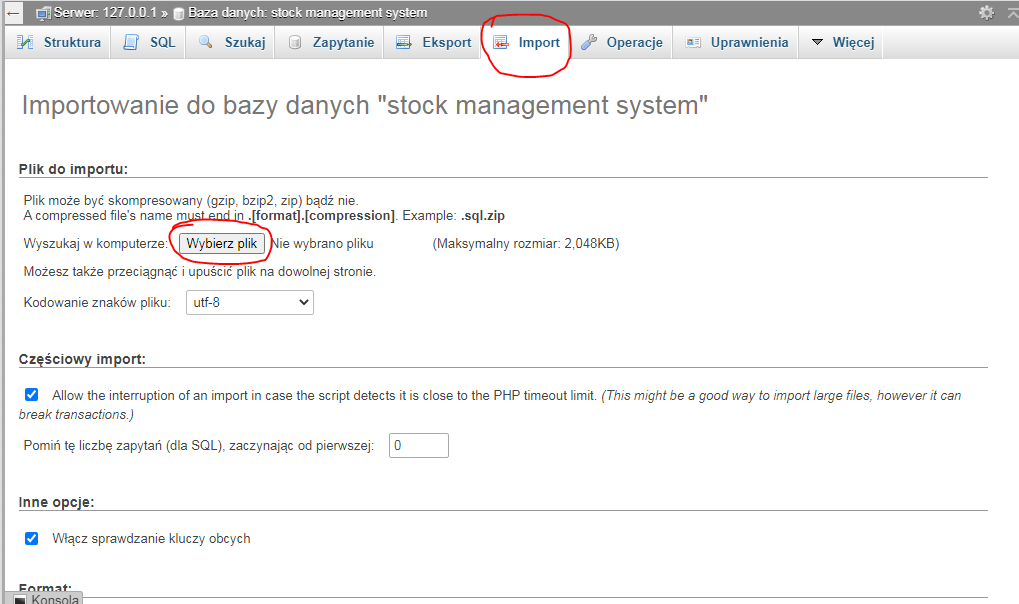
Hasło: admin

Globalne urawnienia: Zaznacz wszystko

Klikamy przycisk „Wykonaj” w prawym dolny rogu na końcu strony

4. Importowanie bazy danych.

* Wybierz zakładkę Import, a następnie „Wybierz plik” i znajdź plik stock\_management\_system.sql



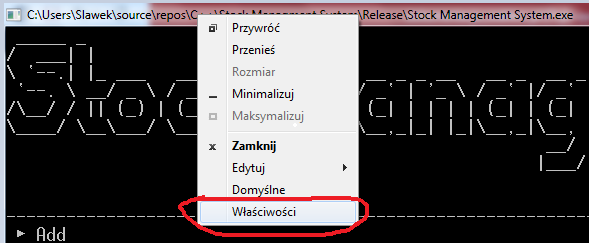
5.Włącz Stock Management System:

# Instrukcja użytkownika

* Po włączeniu programu zaloguj się wpisując login i hasło.

**Login: admin**

**Hasło: admin**

* Upewnij się, że program właczył się w pełnej rozdzielczości ekranu. W przeciwnym razie zmień ustawienia konsoli 

1. Ekran główny

Ekran główny składa się z dwóch kluczowych elementów.



**Ostrzeżenia** – W tym miejscu wyświetlane są wszystkie komunikaty o przekroczeniu ustalonych poziomów zaopatrzenia. Ostrzeżenie składa się z adresu ID którego dotyczy oraz komunikatu. Zakresy limitów można zmienić w zakładce **Settings**.

**Główne menu** – Jest to gówna część programu, odpowiadająca za nawigację i wybór poszczególnych funkcji. Nawigacja odbywa się za pomocą klawiszy:

* + - W górę – *strzałka w górę*
    - W dół – *strzałka w dół*
    - Wybranie opcji - *Enter*
    - Powrót do głównego menu – *Escape*

**Add** – Funkcja odpowiedzialna za dodawanie nowych rekordów w bazie danych oraz za zwiększanie już istniejących.

* + - W przypadku tworzenia nowego rekordu musimy podać nazwę, ilość i rok ważności produktu.
    - Przy dodawaniu nowego rekordu numer ID jest automatycznie inkrementowany.
    - Każdy numer ID jest unikatowy i nigdy się nie powtórzy.
    - Aby zwiększyć ilość już istniejącego towaru wystarczy podać nazwę lub numer ID oraz ilość.
    - Aby wyjść wystarczy wpisać „exit”.
    - W przypadku wpisania błędnych danych pojawi się komunikat, a następnie program wróci do menu głównego.

**Subtract** - Funkcja odpowiedzialna za odejmowanie ilości produktów w bazie danych

* + - Aby zmniejszyć ilość już istniejącego towaru wystarczy podać nazwę lub numer ID oraz ilość.
    - W przypadku gdy ilość elementów które chcemy odjąć jest większa od ilości elementów w bazie, zostanie wyświetlony odpowiedni komunikat, a proukt zostanie całkowicie usunięty.
    - Aby wyjść wystarczy wpisać „exit”.
    - W przypadku wpisania błędnych danych pojawi się komunikat, a następnie program wróci do menu głównego.

**Find** - Funkcja odpowiedzialna za znalezienie i wyświetlenie ilości produktu o który zapytamy.

* + - Aby znaleźć szukany produkt wystarczy podać nazwę lub numer ID.
    - Aby wyjść wystarczy wpisać „exit”.
    - W przypadku wpisania błędnych danych pojawi się komunikat, a następnie program wróci do menu głównego.

**Delete** - Funkcja odpowiedzialna za całkowite usuwanie produktu z bazy danych.

* + - Aby usunąć produkt wystarczy podać nazwę lub numer ID.
    - Aby wyjść wystarczy wpisać „exit”.
    - W przypadku wpisania błędnych danych pojawi się komunikat, a następnie program wróci do menu głównego.

**Statistics** - Funkcja odpowiedzialna za wyświetlenie całej bazy danych.

* + - Nawigacja taka sama jak w menu głównym.
    - Program posiada możliwość rozbudowy, poprzez dodanie funkcji po wciśnięciu przycisku *enter* (np. opis produktu, informację kto i kiedy dodał, możliwość manipulacji ilością produktu w oknie statistics). Obecnie po wciśnięciu program oczekuje na wciśnięcie dowolnego klawisza po czym wraca do okna statistics.
    - Aby wyjść wystarczy wcisnąć klawisz *Esc*.

**Settings** - Funkcja w której możemy zmieniać ustawienia programu. Aktualnie posiada dwie opcje. Zmianę granicy minimum i maksimum ostrzegawczego.

* + - Nawigacja taka sama jak w menu głównym.
    - Po wciśnięciu klawisza *Enter* program przechodzi do opcji zmiany granicy ostrzegawczej.
    - Za pomocą strzałek wybieramy którą cyfrę chcemy zmienić.
    - Cyfry zmieniamy za pomocą klawiszy z cyframi.
    - Za pomocą klawisza *Enter* akceptujemy i zapisujemy zmiany.
    - W czeasie edycji klawisz *Esc* odpowiada za anulowanie zmian.
    - Aby wyjść wystarczy wcisnąć klawisz *Esc*.
    - Aby zmiany zostały wprowadzone musimy ponownie uruchomić program.

**Exit** - Funkcja odpowiedzialna za zamknięcie programu.

# Kompilacja

Do uruchomienia programu wystarczy standardowa kompilacja. Program bezproblemowo uruchamia się w systemie Windows.

# Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

* *Data\_Base.h*, *Data\_Base.cpp*– deklaracja oraz implementacja klasy Data\_Base,
* *Stock Management System.cpp –* klasa główna programu,
* *Menu.h*, *Menu.cpp*– deklaracja oraz implementacja klasy Menu,
* *Global\_Function.h*, *Global\_Function.cpp*– deklaracja oraz implementacja funkcji globalnych,

# Zależności

W projekcie wykorzystano następujące dodatkowe biblioteki:

* MySQL Connector – zapewnia komunikację z serwerami MySQL
* MySQL – silnik bazodanowy SQL.

Zarówno dla MySQL jak i dla MySQL Connector skorzytać [z insalatora MySQL](https://dev.mysql.com/downloads/installer/).

# Opis klas

W projekcie utworzono następujące klasy:

* Data\_Base – reprezentuje bazę danych.
  + Data\_Base(string adress, string base\_name); -Inicjalizacja klasy
  + void Initialization(); - funkcja odpowiedzialna za logowanie i łączenie z bazą SQL
  + void Add(string name, string quantity); - funkcja odpowiedzialna za dodawanie oraz tworzenie nowych towarów w bazie danych.
  + void Subtract(string column, string id\_or\_name, int quantity);- funkcja odpowiedzialna za odejmowanie ilości towarów w bazie danych.
  + void Edit(string column, string value);- funkcja odpowiedzialna za edycję elementów bazy
  + void Delete(string column, string id\_or\_name); - funkcja odpowiedzialna za usuwanie elementów w bazie.
  + void PrintOneRow(int i); - wypisuje odpowiednio jeden wiersz z bazy danych.
  + int is\_exist(string model);- zwraca ilość elementów dla których odpowienia kolumna jest równa *model*. W praktyce sprawdza, czy dany model istnieje.
  + int count\_row(); zwraca ilość wierszy(elementów) w bazie danych.
  + vector<string> show\_all\_id(string column); - zwraca vector wszystkich adresów ID z bazy danych.
* Menu – klasa odpwiedzialna za obsługę wszystkich okien menu.
  + Menu(int stock\_min, int stock\_max); - Inicjalizacja klasy
  + void SetStock(int stock\_min, int stock\_max); - funkcja odpowiedzialna za przypisanie zmiennym prywatnym wartości.
  + int GetStock\_min(); - zwraca wartość stock\_min
  + int GetStock\_max(); - zwraca wartość stock\_max
  + int MainMenu(Data\_Base base); - funkcja odpowiedzialna za obsługę głównego menu.
  + int StatisticsMenu(int number\_row, Data\_Base base); - funkcja odpowiedzialna za obsługę Statistics menu.
  + int SettingsMenu(int number\_row, Data\_Base base); - funkcja odpowiedzialna za obsługę Settings menu.
* Global\_Function – klasa zawierająca funkcje statyczne
  + static int StringToInt(string s1); - zamienia zmienną string na int.
  + static string IntToString(int i1); - zamienia zmienną int na string.
  + static bool is\_number(string text); - zwraca prawda jeśli string jest liczbą.
  + static void Clean(); - funkcja odpowiedzialna za czyszczenie ekranu i rysowanie tytułu.
  + static void GoToxy(const int x, const int y); - funkcja odpowiedzialna za przesuwanie kursora we wskazane miejsce

# Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

* Settings.txt – plik zawierający ustawienia programu.
  + każda kolejna linijka to osobna zmienna odpowiedzialna za poszczególne parametry.
* stock\_management\_system.sql – plik sql zawierający bazę danych, którą trzeba zainportować do servera lokalnego.

# Dalszy rozwój i ulepszenia

* Utworzenie dla programu interfejsu graficznego znacznie poprawiłoby nawigację.
* Rozszerzenie programu o historię modyfikacji.
* Automatyczne przypisywanie aktualnej daty jako kolejną kolumnę dla każdego produktu nazwaną „czas dodania”.
* Modyfikacja programu, aby tworzyć nowe konta użytkownika, bazy, katalogi z poziomu programu.
* Ulepszenie autoryzacji i sprawdzania hasła(obecne hasło można znaleźć w kodzie).
* Zakrywanie hasła podczas wpisywania.
* Dodanie kolejnych kont użytkowników z różnymi uprawnieniami.
* Dodanie kolejnych baz danych („magazynów”).
* Dodanie możliwości „zamawiania” przez urzytkowników produktów, czyli automatycznego tworzenia listy zamówień.
* Wykorzystać internetową zamiast lokalnej bazy danych.
* Umożliwienie korzystania z programu na kilku urządzeniach.
* Dodanie możliwości skanowania kodu kreskowego, tym samym umożliwienie automatycznego aktualizowania bazy danych.

# Inne

Brak.