

SCADA 2

Maciej Cebula
Piotr Merynda
Maciej Podsiadło

Spis treści

1	Wstęp	2
1.1	Opis projektu	2
1.2	Założenia	2
2	Model Systemu	3
2.1	System zbiorników	3
2.2	Regulatory	4
2.2.1	Regulacja poziomu wody w zbiorniku 3	4
2.2.2	Regulacja poziomu koncentratu w zbiorniku 2	4
2.2.3	Regulacja dawkowania wody i koncentratu do zbiornika mieszającego	4
2.2.4	Regulacja dawkowania gotowej mieszanki ze zbiornika mieszającego	4
3	Serwer HTTP	5
3.1	Opis działania	5
3.2	Uruchomienie serwera	5
4	Baza danych	6
4.1	Wstęp	6
4.2	Struktura bazy	6
4.3	Opis dostępu do bazy	7
4.4	Serwer OPC	8
4.4.1	Wprowadzenie	8
4.4.2	MatrikonOPC	8
4.4.3	Logowanie OPC - MySQL	8
5	Aplikacja	9
5.1	Opis działania aplikacji	9
5.2	Opis okien	9
5.3	Alarmy	10
5.4	Regulatory	10
5.5	Wygląd aplikacji	10
5.6	Uruchomienie aplikacji	14

Wstęp

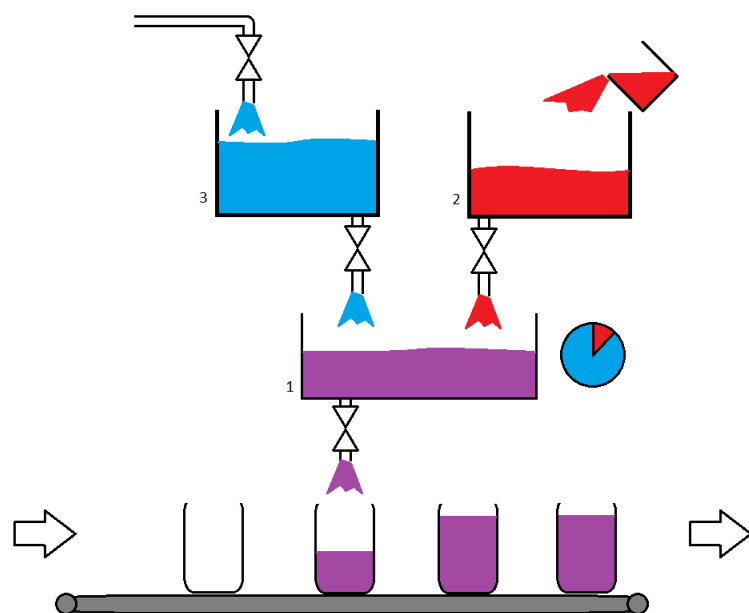
1.1 Opis projektu

1.2 Założenia

Model Systemu

2.1 System zbiorników

Przedmiotem rozważań jest system 3 połączonych zbiorników. Zbiornik 3 służy jako rezerwu-
ar wody. W zbiorniku 2 znajduje się koncentrat wykorzystywany do produkcji napoju. Oba
te zbiorniki umieszczone są nad zbiornikiem 1, w którym woda miesza się z koncentratem
tworząc gotowy produkt. Pomiedzy zbiornikami 3 i 1 oraz 2 i 1 znajdują się zawory pozwała-
jące na regulację przepływu cieczy pomiędzy nimi. Poglądowy schemat procesu znajduje się
na rysunku 2.1.

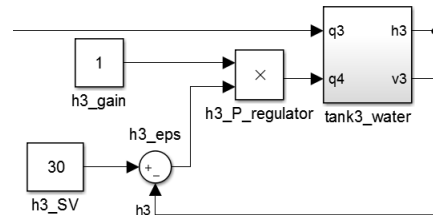


Rys. 2.1: Schemat procesu.

2.2 Regulatory

2.2.1 Regulacja poziomu wody w zbiorniku 3

Regulacja poziomu wody w zbiorniku 3 odbywa się poprzez regulator proporcjonalny sterujący stopniem otwarcia zaworu q_4 . Wartość sterowania wyliczana jest na podstawie uchybu pomiędzy wartością zadaną $h3_SV$ a wysokością poziomu wody w zbiorniku $h3$.



Rys. 2.2: Schemat regulatora poziomu cieczy $h3$.

2.2.2 Regulacja poziomu koncentratu w zbiorniku 2

Regulacja poziomu koncentratu w zbiorniku 2 odbywa się poprzez dolewanie dodatkowych porcji substancji przez pracownika po zgłoszeniu przez system komunikatu o niskim poziomie cieczy w zbiorniku. Aby zamodelować działanie pracownika, który posiada pewną zwłokę w wykonywaniu działań oraz potrzebuje czasu na przemieszczenie się z dodatkową porcją koncentratu, użyto maszyny stanów. Określono czas reakcji pracownika na komunikat systemowy, czas potrzebny do zabrania kolejnej porcji, oraz czas potrzebny na uzupełnienie koncentratu.

2.2.3 Regulacja dawkowania wody i koncentratu do zbiornika mieszającego

Regulacja przepływu wody oraz koncentratu do zbiornika mieszającego odbywa się na podstawie pomiaru wysokości cieczy w zbiorniku 1 oraz na podstawie pomiaru stężenia koncentratu w gotowym produkcie. Regulacja odbywa się w taki sposób aby utrzymać zadany poziom w zbiorniku oraz zadane stężenie gotowego produktu.

2.2.4 Regulacja dawkowania gotowej mieszanki ze zbiornika mieszającego

Regulator odpowiedzialny za napełnianie pojemników gotową mieszanką działa poprzez maksymalne otwarcie wypływu ze zbiornika 1 aż do całkowitego napełnienia się naczynia. Po całkowitym napełnieniu zawór zostaje zamknięty aż do nadejścia kolejnego pojemnika do napełnienia.

Serwer HTTP

3.1 Opis działania

W celu umożliwienia komunikacji aplikacji wizualizującej z bazą danych napisano w języku *JAVA* serwer HTTP. Dane pomiędzy serwerem i aplikacją wymieniane są za pomocą protokołu HTTP, wykorzystując w tym celu zapytania typu *GET* i *POST*. Wszystkie żądania są utożsamiane z odpowiednim adresem url i odpowiednio przetwarzane po stronie serwerowej. Aby zapewnić jak największą responsywność aplikacji oraz synchroniczne odświeżanie danych każdy request po stronie serwera przetwarzany jest w osobnym wątku.

Do napisania serwera wykorzystano następujące biblioteki *JAV-y*:

1. **RXJava** - biblioteka zapewniająca mechanizmy asynchronicznego przetwarzania danych [4],
2. **JDBI** - biblioteka zapewniająca połączenie z bazą danych [3],
3. **AKKA-HTTP** - framework do implementacji serwera HTTP [2],
4. **Javax mail** - biblioteka do wysyłania emaili.

3.2 Uruchomienie serwera

W momencie pisania sprawozdania w repozytorium nie udostępniono jeszcze skompilowanej wersji flików źródłowych. Dlatego też w celu uruchomienia aplikacji na komputerze wymagane jest:

1. zainstalowanie Jav-y w wersji 8
2. zainstalowanie dowolnego środowiska programistycznego *Intellij*, *Eclipse itp.*
3. skompilowanie i uruchomienie całego projektu dostępnego pod adresem <https://github.com/maciekc/SCADA2>

Baza danych

4.1 Wstęp

Aby móc wizualizować przebieg pracy systemu wszystkie najważniejsze dane z punktu widzenia automatyki są logowane w bazie danych. Do tego celu wykorzystano bazę danych typu *MYSQL* i środowisko *MYSQL Workbench*, zainstalowane na jednym z komputerów, którym stworzono całą strukturę przedstawioną na rysunku 4.1. Dostęp poszczególnych węzłów systemu do bazy danych zapewniony jest poprzez połączenie wszystkich komputerów w lokalną sieć.

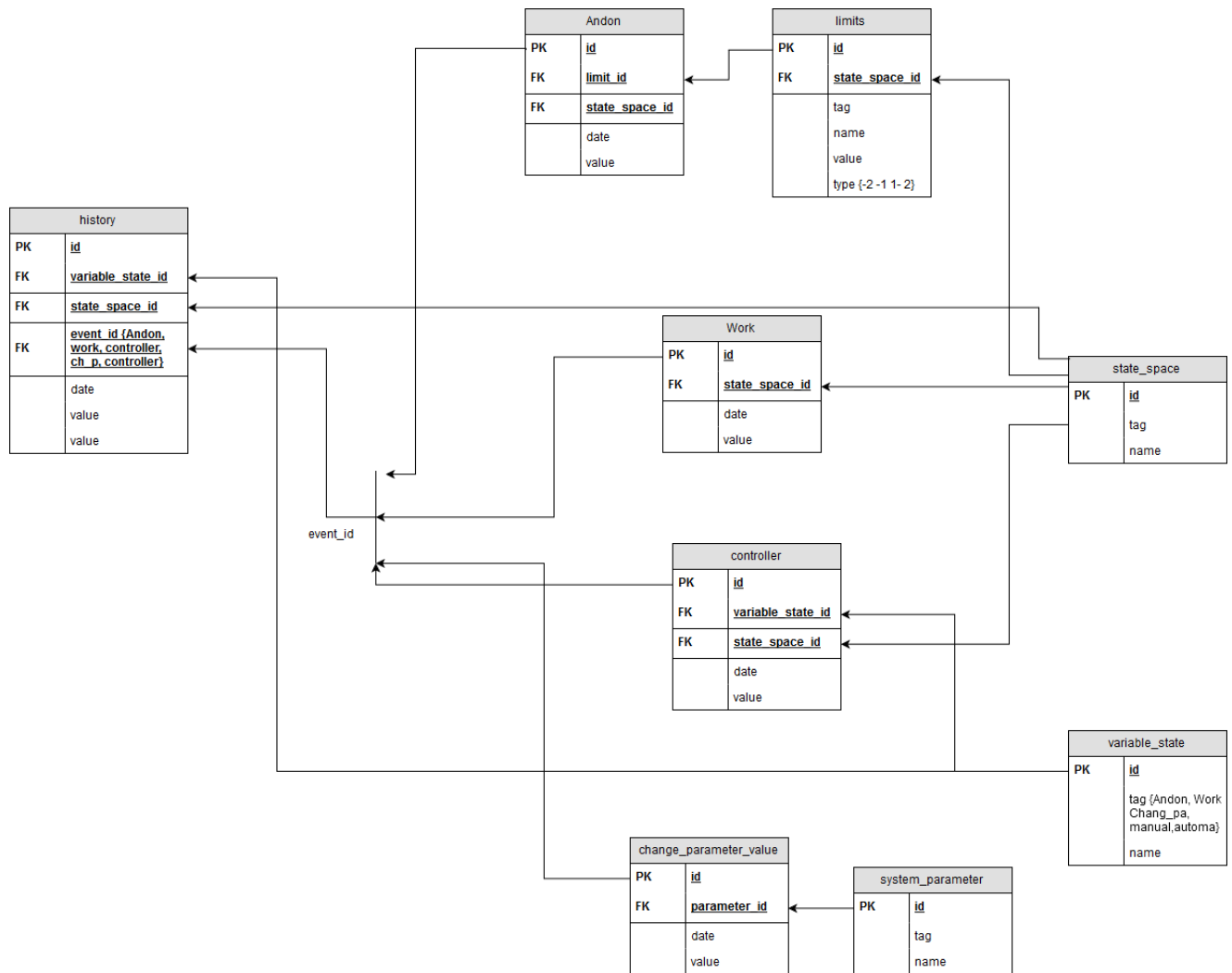
4.2 Struktura bazy

Zaprezentowana na rysunku baza danych na strukturę relacyjną aby zapewnić możliwość ewentualnej rozbudowy systemu o inne składowe. Z racji na to, że baza służy głównie do logowania pracy systemu podzielono ją na następujące części:

1. główną tabelę **history**, w której logowane są wszystkie zdarzenia w kolejności chronologicznej,
2. tabelę **Andon**, w której zapisywane są wszystkie sytuacje alarmowe,
3. tabelę **Work**, do której logowany jest stan normalnej pracy systemu,
4. tabelę **change_parameter**, w której zapisywane są wszelkie zmiany wartości parametrów systemu,
5. tabelę **controller**, która służy do logowania pracy regulatorów.

Dodatkowo:

1. w tabeli **state_space** zdefiniowano wszystkie zmienne stanu występujące w systemie,
2. tabela **variable_state** definiuje wszystkie możliwe stany danej zmiennej np. Work, Andon, Change parameter itd.
3. tabela **limits** definiuje poszczególne limity i przypisuje je do odpowiednich zmiennych stanu z tabeli **state_space**,
4. w tabeli **system_parameter** zdefiniowane są pozostałe parametry systemu takie jak: nastawy poszczególnych regulatorów, wartości zadane itp.



Rys. 4.1: Struktura bazy danych.

4.3 Opis dostępu do bazy

Dane dotyczące parametrów systemu w poszczególnych chwilach czasu są logowane przez serwer OPC i następnie prezentowane w aplikacji.

4.4 Serwer OPC

4.4.1 Wprowadzenie

Systemy typu PLC – SCADA są powszechnie wdrażane poczynając od przemysłu chemicznego, a kończąc na automatyce budynków. Stanowią pewną normę w nowoczesnych zakładach oraz fabrykach. W związku z bogatą ofertą producentów aparatury automatyzacji powstał problem komunikacji pomiędzy różnymi komponentami. Firmy promowały swoje rodzime protokoły przemysłowe, co wymuszało na końcowych użytkownikach stosowanie sprzętu pochodzącego od tego samego producenta w obrębie całego obiektu. Przełomem okazało się wprowadzenie otwartego rozwiązania – standardu OPC. OPC jest standardem umożliwiającym komunikację pomiędzy sterownikami (najczęściej PLC) a oprogramowaniem SCADA. Bazuje na modelu klient-serwer, przy czym strona serwera zaimplementowana jest w oprogramowaniu dostarczonym przez producentów sterowników, natomiast klientem - aplikacja wykorzystująca udostępniane dane.

4.4.2 MatrikonOPC

Dla celów zintegrowania omawianego systemu zastosowano testowy serwer udostępniony przez firmę Matrikon służący do celów niekomercyjnych. Dostawca oferuje serwer OPC (OPC Simulation Server), jak również oprogramowanie do zarządzania odczytywanymi danymi (OPC Explorer).

4.4.3 Logowanie OPC - MySQL

Kolejny element stworzonego wielopoziomowego systemu sterowania stanowi aplikacja logująca aktualne dane pochodzące z serwera OPC do procesowej bazy danych. Aplikacja napisana została w języku C#. Do komunikacji z serwerem użyto open source'owej biblioteki TitaniumAS.Opc (<https://github.com/titanium-as/TitaniumAS.Opc.Client>). Serwer lokalizowany jest jedynie po nazwie, co, zgodnie z ideą OPC, znacząco przyspiesza proces integracji. Aplikacja odczytuje bieżące pomiary z serwera OPC z zadaną przez użytkownika częstotliwością, a następnie loguje je w bazie danych MySQL. Podstawową częstotliwością odpytywania jest 1 sekunda, tak jak to ma miejsce w większości systemów typu SCADA. Poszczególne pomiary identyfikowane są po tagach jakie zostały im nadane w momencie inicjalizacji w serwerze OPC. Zapis do bazy danych odbywa się według konwencji narzuconej w momencie jej zaprojektowania. Aplikacja zapewnia mechanizmy przechwytywania zgłaszanych błędów, w szczególności błędów połączenia z serwerem oraz MySQL. Użytkownik informowany jest o napotkanym błędzie. Program dokonuje niezbędnej konwersji sposobu zapisu liczb zmiennoprzecinkowych, zamienia separator ‘,’ na ‘.’ bez czego zapis do bazy danych nie byłby możliwy.

Warto nadmienić, że użytkownik operujący na systemie Windows nie musi instalować, żadnego dodatkowego oprogramowania w celu uruchomienia omawianej aplikacji logującej.

Aplikacja

5.1 Opis działania aplikacji

Głównym założeniem aplikacji wizualizującej pracę systemu było zaprezentowanie pracy systemu w sposób przejrzysty i uporządkowany. Dodatkowo aplikacja powinna umożliwiać zmianę wybranych parametrów systemu oraz zapewniać jednoczesny dostęp dla wielu użytkowników. Biorąc pod uwagę wymienione wymagania zdecydowano się na napisanie aplikacji webowej wykorzystując do tego celu framework *Angular 4* - <https://angular.io/>.

5.2 Opis okien

Aplikacja do wizualizacji pracy systemu podzielona jest na 5 ekranów.

1. **Schemat** - przedstawia schemat rozpatrywanej instalacji,
2. **Statystyki** - przedstawia najważniejsze dane opisujące aktualny stan instalacji takie jak: poziomu cieczy w każdym ze zbiorników, aktualne stężenie, a także przebiegi czasowe tych sygnałów zaprezentowane za pomocą wykresów,
3. **Obiekt** - prezentuje szczegółowe dane dotyczące obiektu regulacji oraz daje możliwość zmieniania limitów w każdym ze zbiorników.
4. **Regulator** - przedstawia przebiegi czasowe sygnałów sterowania, a także daje możliwość zmieniania nastaw regulatorów,
5. **Logi** - daje możliwość generowania raportów z wybranych stanów pracy systemu tzn: zwykła praca, stany alarmowe, zmiany wartości parametrów systemu.

Każdy z ekranów na składa się z trzech głównych części 5.1:

1. Pasek menu - umożliwia przechodzenie pomiędzy wszystkimi oknami aplikacji,
2. **Panel danych** - w jego obszarze wyświetlane są dane dotyczące poszczególnych funkcjonalności systemu,
3. **Pasek wiadomości** - Wyświetlane są na nim najważniejsze wiadomości dotyczące pracy systemu takie jak np. przekroczenie limitów.

5.3 Alarmy

Każdy zbiornik ma zdefiniowany zestaw czterech limitów poziomu cieczy tzn.

1. Limit minimalny krytyczny,
2. Limit minimalny,
3. Limit maksymalny,
4. Limit maksymalny krytyczny

Wartości wyżej wymienionych limitów mogą być monitorowane oraz zmieniane przy wykorzystaniu ekranu *Obiekty*. Każda zmiana wartości jest automatycznie zapisywana w bazie danych i propagowana do regulatora.

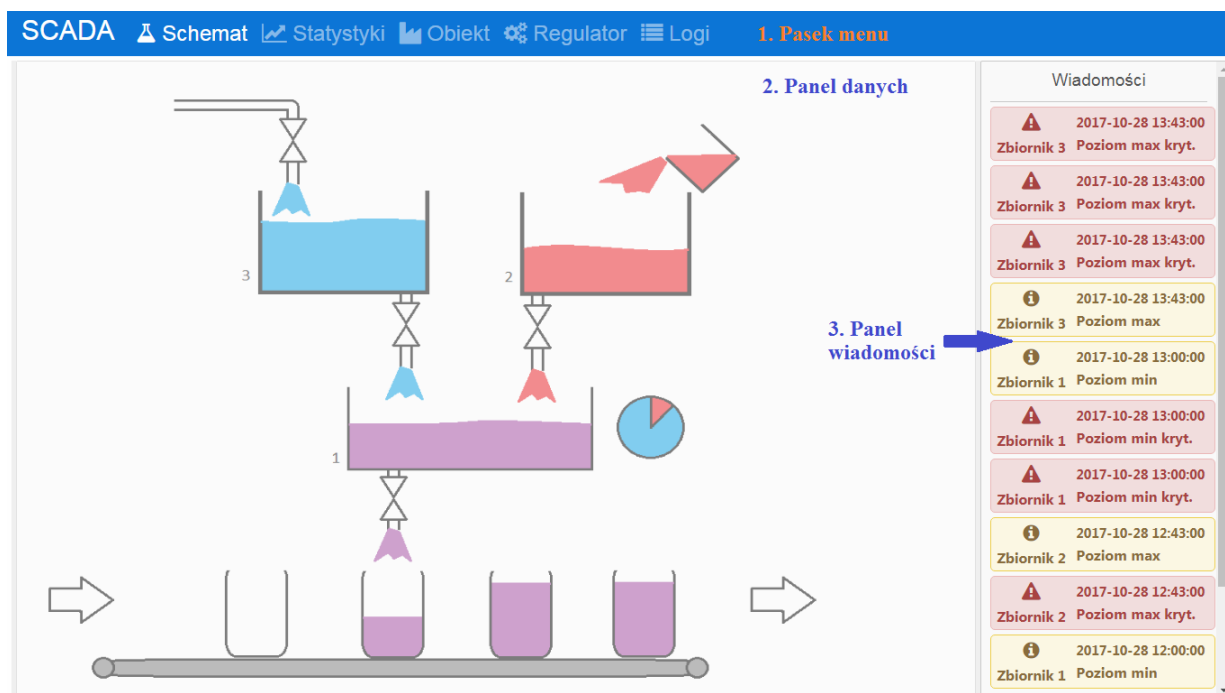
W aplikacji zaimplementowano również propagację alarmów. W momencie przekroczenia dowolnego ze zdefiniowanych limitów pod wskazany adres wysyłany jest email informujący o tym zdarzeniu.

5.4 Regulatory

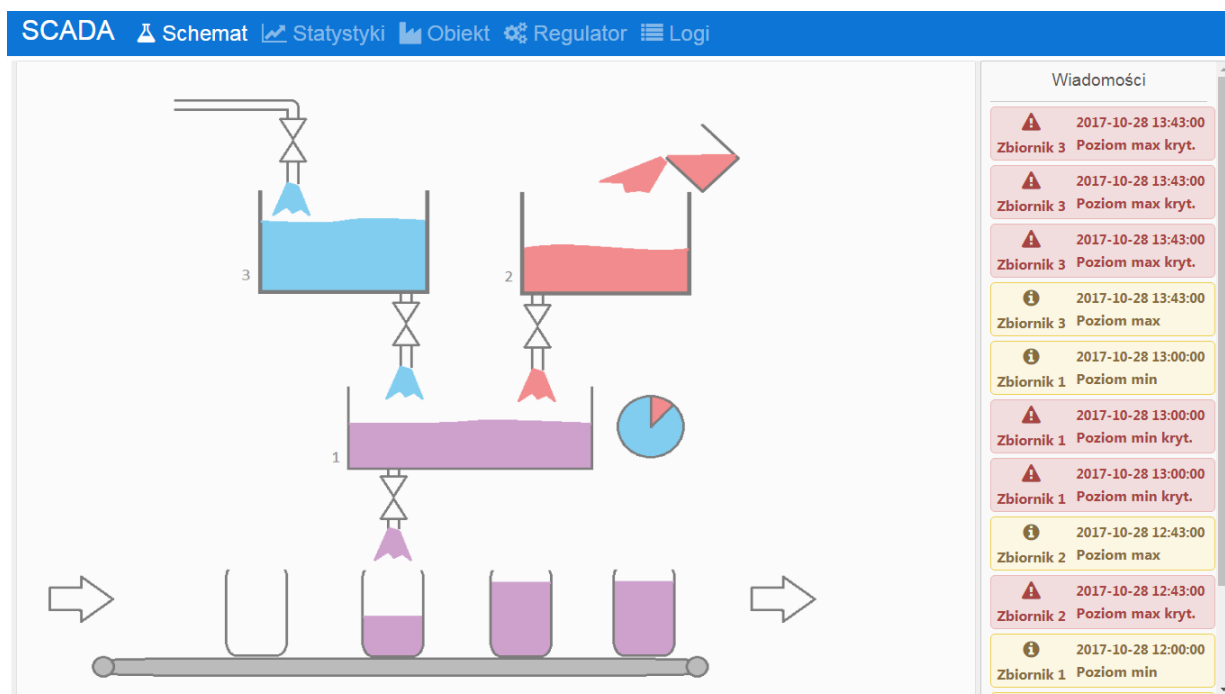
Aplikacja daje możliwość operatorowi na zdalną zmianę parametrów regulatorów za pomocą ekranu *Regulator*. Podobnie jak w przypadku alarmów każdorazowa zmiana dowolnego parametru jest propagowana do regulatora i zapisywana w bazie danych.

5.5 Wygląd aplikacji

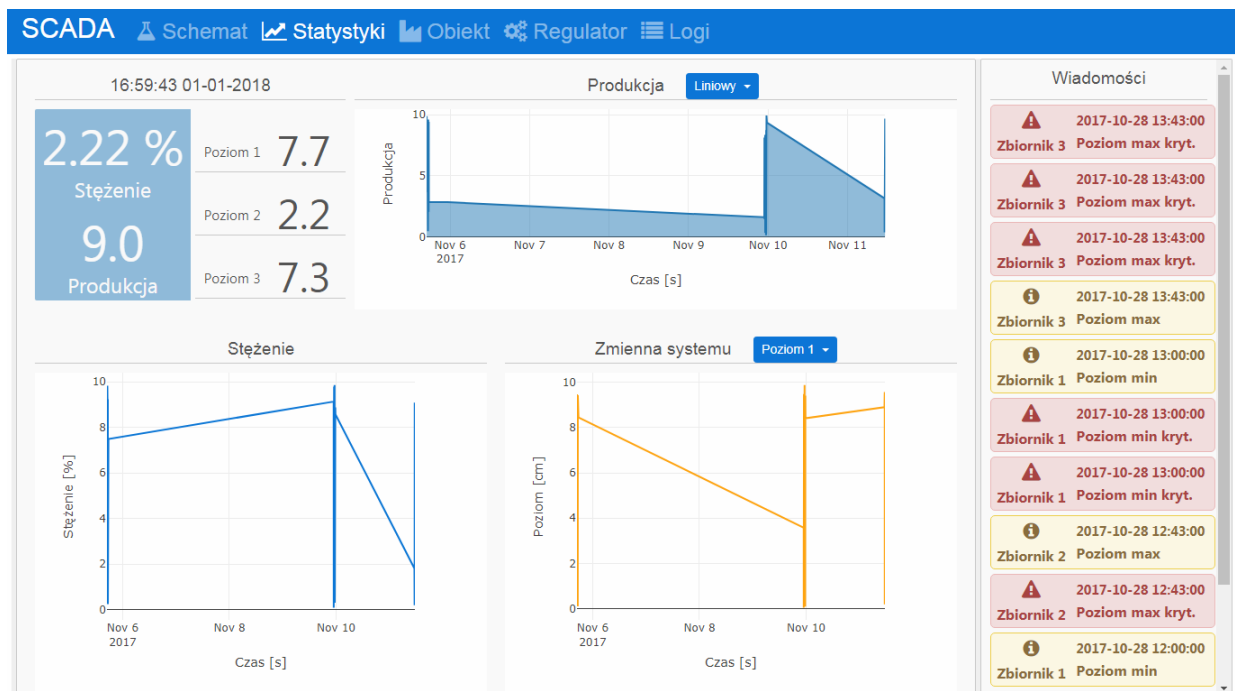
Na poniższych rysunkach przedstawiono screen-y z działania opisywanej aplikacji dla każdego z dostępnych ekranów.



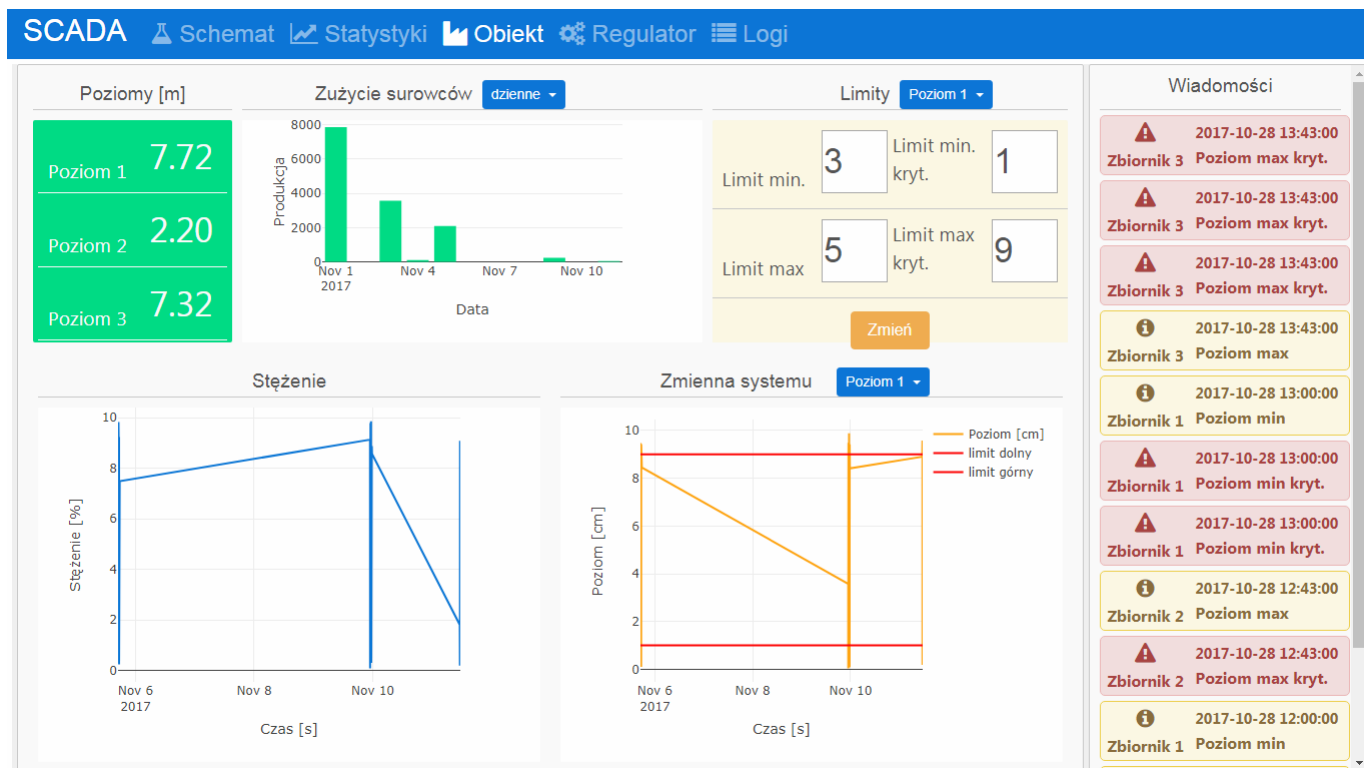
Rys. 5.1: Budowa ekranu.



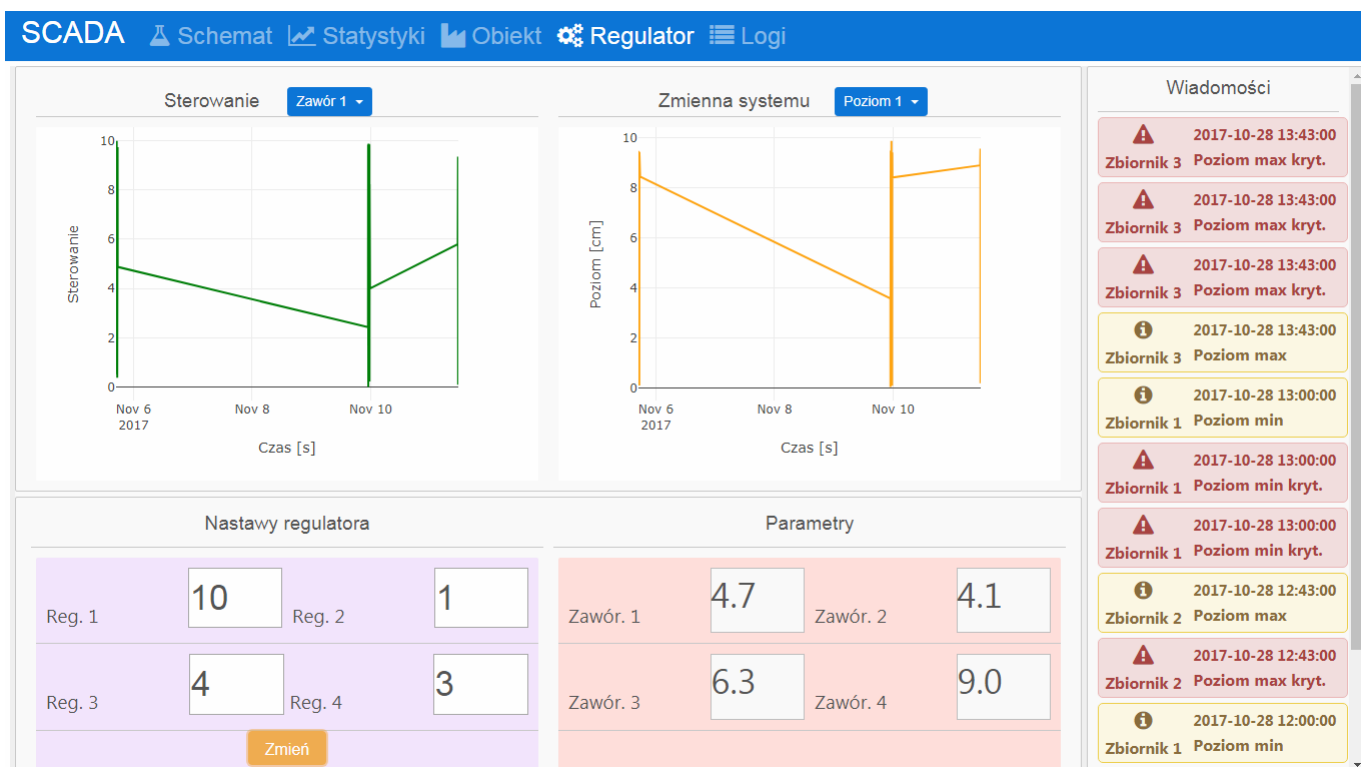
Rys. 5.2: Schemat systemu.



Rys. 5.3: Statystyki systemu.



Rys. 5.4: Ekran przedstawiający stan instalacji.



Rys. 5.5: Ekran przedstawiający pracę regulatorów.

SCADA

Schemat

Statystyki

Obiekt

Regulator

Logi

Typ raportu

Andon

Generuj

LP	Zmienna	Typ	Wartość	Data
12	Zbiornik 3	Poziom max kryt.	120.0	2017-10-28 13:43:00
11	Zbiornik 3	Poziom max kryt.	120.0	2017-10-28 13:43:00
10	Zbiornik 3	Poziom max kryt.	120.0	2017-10-28 13:43:00
9	Zbiornik 3	Poziom max	120.0	2017-10-28 13:43:00
8	Zbiornik 1	Poziom min	25.00	2017-10-28 13:00:00
7	Zbiornik 1	Poziom min kryt.	25.00	2017-10-28 13:00:00
6	Zbiornik 1	Poziom min kryt.	25.00	2017-10-28 13:00:00
5	Zbiornik 2	Poziom max	100.0	2017-10-28 12:43:00
4	Zbiornik 2	Poziom max kryt.	100.0	2017-10-28 12:43:00
3	Zbiornik 1	Poziom min	25.00	2017-10-28 12:00:00
2	Zbiornik 2	Poziom max	100.0	2017-10-28 12:43:00
1	Zbiornik 1	Poziom min	25.00	2017-10-28 12:00:00

Wiadomości

2017-10-28 13:43:00

Zbiornik 3 Poziom max kryt.

2017-10-28 13:43:00

Zbiornik 3 Poziom max kryt.

2017-10-28 13:43:00

Zbiornik 3 Poziom max kryt.

2017-10-28 13:43:00

Zbiornik 3 Poziom max

2017-10-28 13:00:00

Zbiornik 1 Poziom min

2017-10-28 13:00:00

Zbiornik 1 Poziom min kryt.

2017-10-28 13:00:00

Zbiornik 1 Poziom min kryt.

2017-10-28 12:43:00

Zbiornik 2 Poziom max

2017-10-28 12:43:00

Zbiornik 2 Poziom max kryt.

2017-10-28 12:00:00

Zbiornik 1 Poziom min

Rys. 5.6: Ekran do generowania raportów.

5.6 Uruchomienie aplikacji

Do uruchomienie aplikacji należy:

1. na wybranym komputerze zainstalować serwer *Node.js* - <https://nodejs.org/en/>
2. pobrać kod źródłowy projektu ze strony <https://github.com/maciekc/SCADA2>
3. w konsoli systemu *Windows* należy przejść do głównego folderu projektu *SCADA-app* i wywołać procedurę **npm start**
4. od tej pory aplikacja będzie dostępna pod adresem *www.localhost:4200*.

Bibliografia

- [1] <https://angular.io/docs> dokumanetacja framework-u Angular
- [2] <https://doc.akka.io/docs/akka-http/current/index.html> dokumentacja framework-u Akka-http.
- [3] <http://jdbi.org/> - dokumentacja biblioteki JDBI
- [4] <http://reactivex.io/> - dokumentacja biblioteki RXJava.