Projekt i implement. sys. web

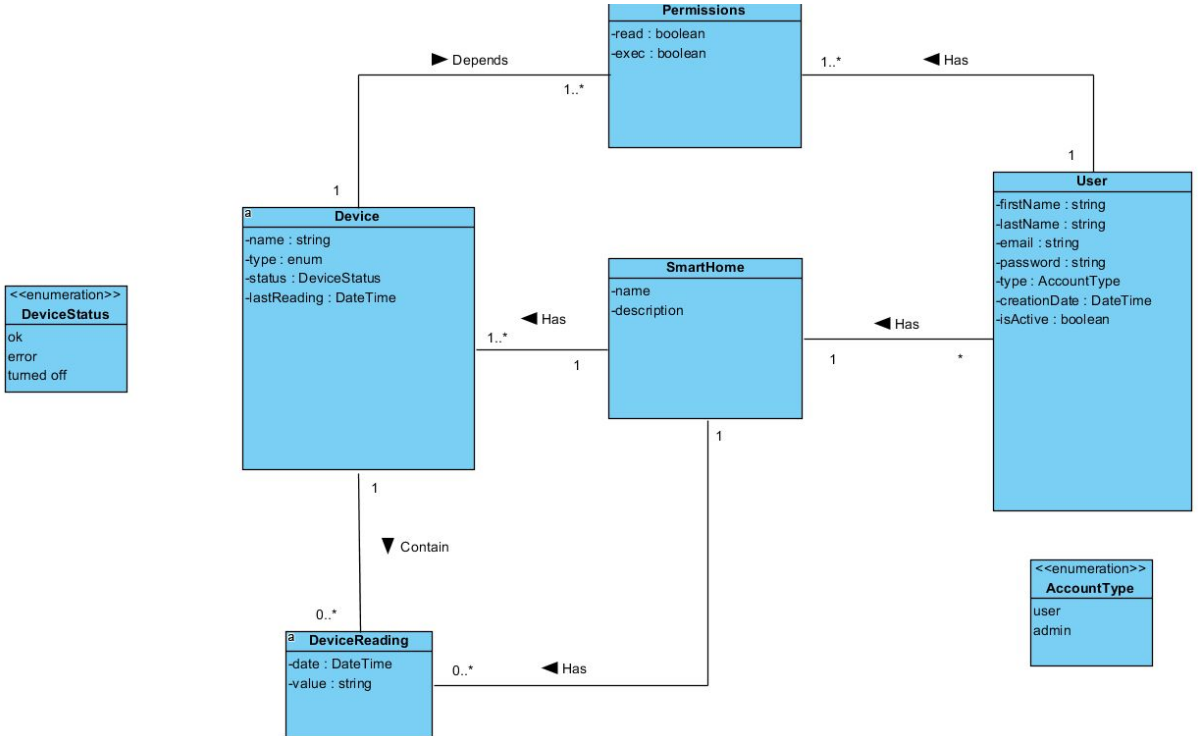
1. Cele projektu

Celem projektu jest stworzenie aplikacji webowej analizującej dane nabyte z systemu sensorów tworzących inteligentny dom. Projekt zawieraja część serwerową opartą o SpringBoot oraz JPA jak również o część Frontendową opartą o Angular.

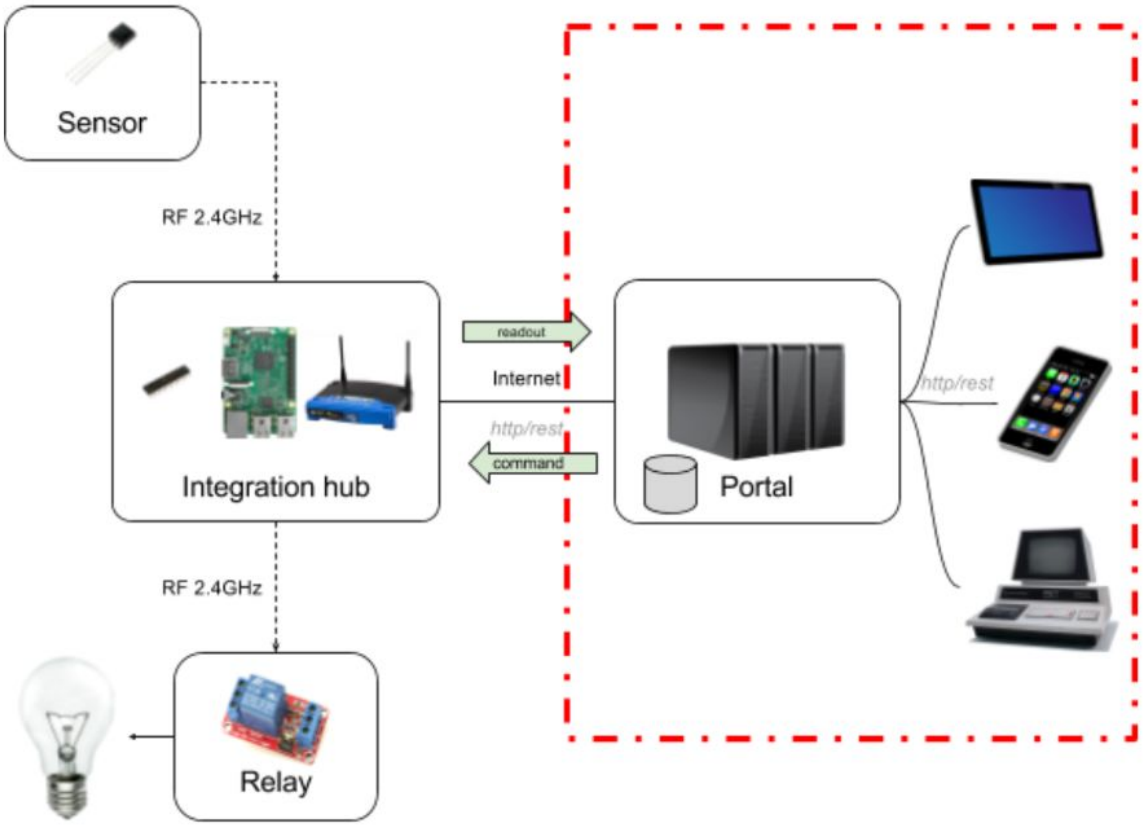
2. Model przypadków użycia

Materiał zostanie pokazany na zajęciach

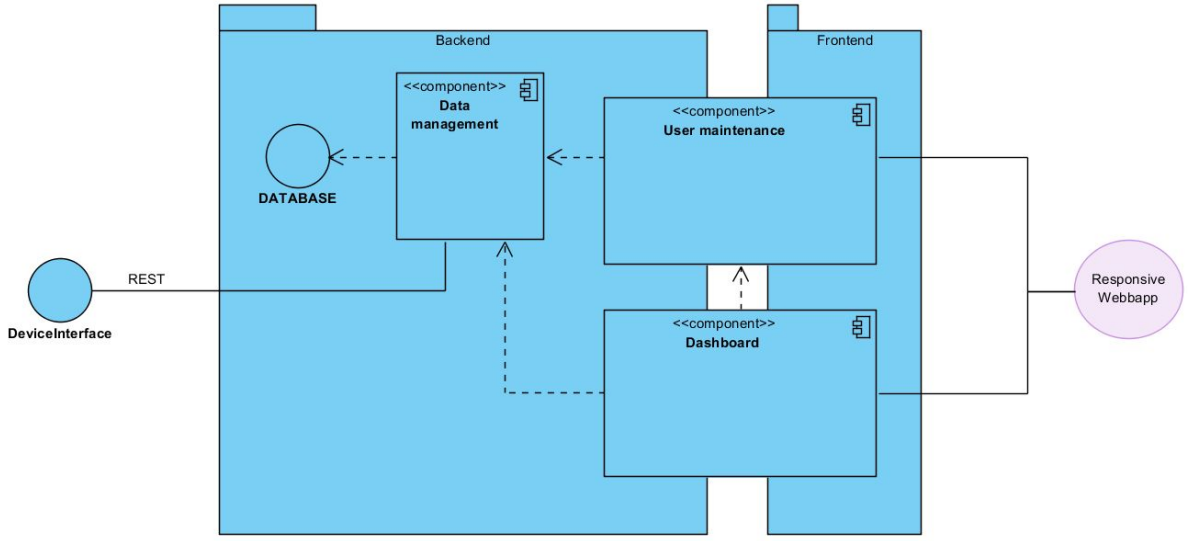
3. Model danych



4. Model architektury



Zakres projektu zaznaczono na czerwono. Aplikacja webowa kontaktuje się z urządzeniami za pomocą interfejsu REST. Ponadto udostępnia klienta webowego komunikującego się z systemem również w ten sam sposób.



Samą aplikację możemy podzielić na dwie części i trzy moduły. Pierwsza część odpowiada za logikę biznesową a druga za interfejs użytkownika.

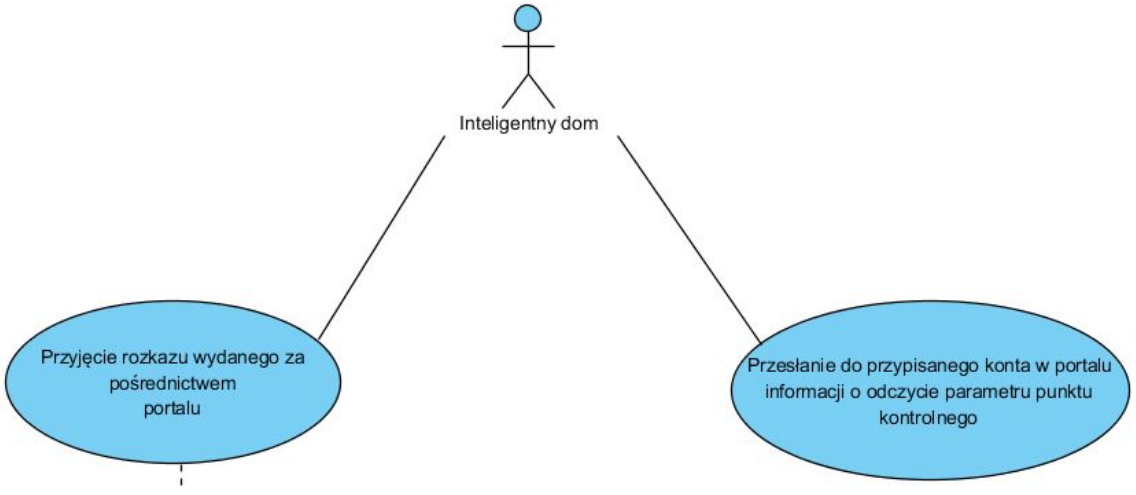
Wyróżniamy trzy moduły:

1. Zarządzanie kontami i użytkownikami - moduł administracyjny pozwalający zarządzać użytkownikami i nadawać im odpowiednie uprawnienia (operujący na warstwach backendu i frontendu)
2. Gromadzenie danych i komunikacja - moduł odpowiadający za przyjmowanie poleceń od użytkownika i komunikację z urządzeniami (operuje na warstwie backendu)
3. Publikacja danych oraz komend - moduł odpowiadający za przedstawienie danych użytkownikowi zgromadzonych przez moduł nr 2

Wszystkie moduły są zbudowane w ramach jednej aplikacji. Każdy moduł odpowiada osobnemu pakietowi. Backend i frontend wymagają postawienia osobnych serwerów.

5. Definicja interfejsów zewnętrznych

Komunikacja między aplikacją webową a urządzeniami będzie się odbywać poprzez interfejs REST. Zakładamy, że każde urządzenie będzie udostępniać API, dzięki któremu będziemy mogli się uzyskać odczyty lub wydać rozkaz.



Przykłady komunikacji aplikacji z urządzeniem można powiązać z dwoma przypadkami użycia, w których **<Inteligentny dom>** przyjmuje rozkaz od użytkownika i żąda jego wykonania oraz żąda odczytów - wszystko poprzez interfejs REST.

Żądanie odczytu - przykład protokołu komunikacyjnego:

HTTP Body	POST
URI template	/domotics/{portalId}/api/v1.0/readouts
Request Content Type	application/json

Request Body	{
	"Hub":{
	"Id":"3790f3c4-d79b-48a4-9299-c0860b395cea",
	"Name":"Universal meter",
	"Version":"1.0"
	},
	"Sensors":[
	{
	"SensorType":{
	"Id":"1",
	"Type":"Analog Temperature Sensor",
	"Model":"LM35DZ",
	"Unit":"Celsius"
	},
	"Readout":{
	"Time":"2017-08-17 09:46:52",
	"Value":"25.0"
	}
	},
	{
	"SensorType":{
	"Id":"2",
	"Type":"Humidity Sensor",
	"Model":"HR202L",
	"Unit":"Scalar"
	},
	"Readout":{
	"Time":"2017-08-17 09:43:12",
	"Value":"0.85"
	}
	}
	],
	"ControlPoints":[
	{
	"ControlPointType":{
	"Id":"3",
	"Type":"One Channel Relay",
	"Model":"SRD-05VDC-SL-C"
	},
	"Commands":[
	{
	"Id":"1",
	"Name":"Channel 0 On"
	},
	{
	"Id":"2",
	"Name":"Channel 0 Off"
	}
	],
	"State":"1",
	
	"URI":"http://127.0.0.1:8081/domotics/control/3790f3c4-d79b-48a4-9299-c0860b395cea/3"
	}
	]
	}

Response Content	n/a

Type	

Wykonanie rozkazu - przykład protokołu komunikacyjnego:

HTTP Body	POST
URI template	Przesłane w polu URI dla elementu sterowalnego.
Request Content Type	application/json

Request Body	{
	"CommandId":"1"
	}

Response Content Type	application/json

Response body	{
	"State":"1"
	}

6. Projekt interfejsu użytkownika

Makiety dodane zostały do repozytorium projektu