OpenIMS GUI

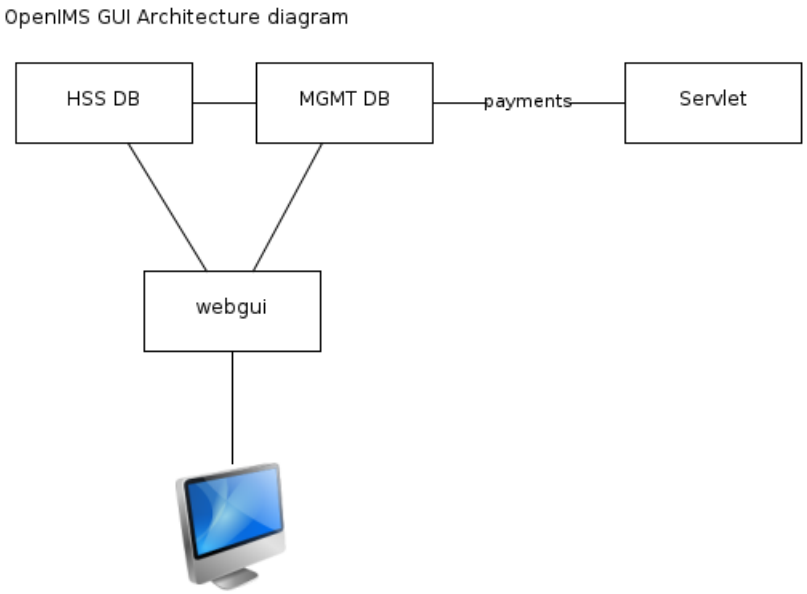
Marcin Wawrzyniak
Marek Pawłowski

Zawartość

Architektura

Użytkownik za pomocą przeglądarki internetowej korzysta z systemu **webgui**.

Webgui komunikuje się z bazą utworzoną na potrzeby **webgui**, oraz z bazą **OpenIMS** (tabela **impi**). Oprócz tego, **servlet** kolejnej grupy komunikuje się z tabelą **payments** bazy danych **webgui**



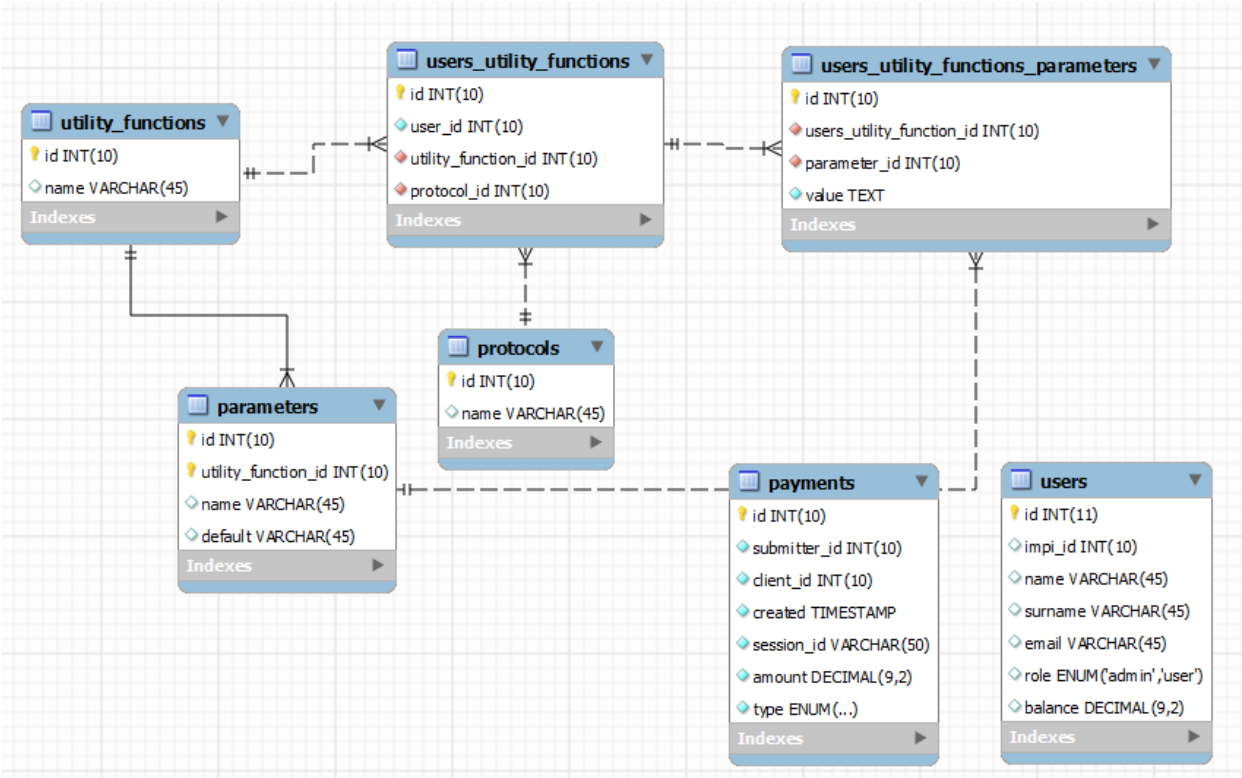
Technologie

PHP, CakePHP

System został napisany w języku PHP 5.3, korzystając z Frameworka CakePHP w wersji 2.2 .

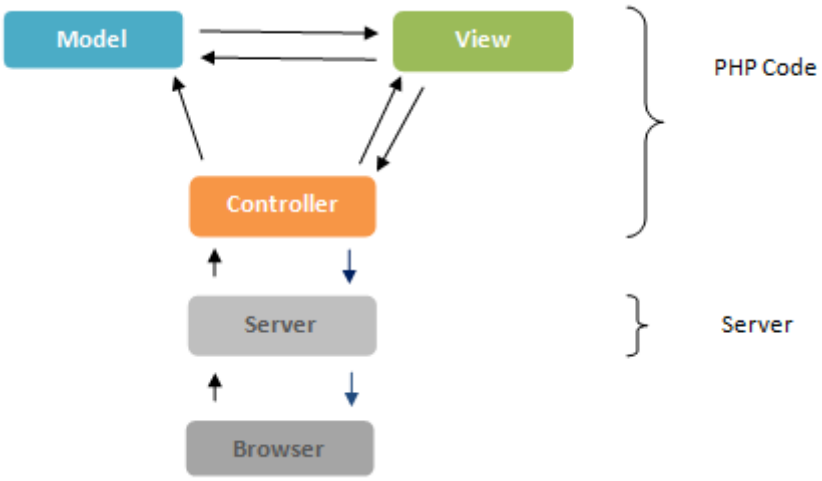
CakePHP narzuca wzorzec MVC, dzięki czemu mamy możliwośc uniknięcia „spaghetti code”.

Jako baza danych została użyta sprawdzona już konstrukcja, mianowicie MySQL w wersji 5.x.



MVC

MVC to wzorzec wykorzystywany przez CakePHP między innymi do organizacji kodu.



Modele

Kod w modelach odpowiada tabelom w bazie danych, oprócz tego zawiera reguły walidacyjne.

Wszystkie modele dziedziczą po **AppModel** udostępnianym z **CakePHP**.

Dodatkowym modelem jest model **Impi**, który jest modele z bazy danych **OpenIMS**.

Widoki

Widoki zawierają kod HTML (ewentualnie JSON), wraz z PHP użytym w roli systemu szablonów.

Kontrolery

Kontrolery **Parameters**, **UtilityFunctions** oraz **Protocols** umożliwiają CRUD odpowiadających modeli (Parameter, UtilityFunction, Protocol).

Dodatkowo kontroler **Users** umożliwia tworzenie oraz uwierzytelnianie użytkownika (korzystając z tabeli **impi**).

Kontroler **Payments** umożliwia wyświetlenie użytkownikowi billingu.

Flow

Punktem wejścia w CakePHP jest **index**.php oraz **bootstrap**.php, konfiguracje akcji można przeprowadzić w pliku **router**.php .

Plik ten mówi: „po wywołaniu tego adresu”, wywołaj „tą akcję w tym kontrolerze, z takimi parametrami”.

REST API

GET	http://openims.local/api/parameters/:id
DELETE	http://openims.local/api/parameters/:id
POST (PUT)	http://openims.local/api/parameter/:id
POST	http://openims.local/api/parameters
GET	http://openims.local/api/parameters
GET	http://openims.local/api/protocols/:id
DELETE	http://openims.local/api/protocols/:id
POST (PUT)	http://openims.local/api/protocol/:id
POST	http://openims.local/api/protocols
GET	http://openims.local/api/protocols
GET	http://openims.local/api/utility_functions/:id
DELETE	http://openims.local/api/utility_functions/:id
POST (PUT)	http://openims.local/api/utility_function/:id
POST	http://openims.local/api/utility_functions
GET	http://openims.local/api/utility_functions

GET	http://openims.local/api/users/bob@open-ims.test/payments	Lista płatnosci bob’a
GET	http://openims.local/api/users/bob@open-ims.test/utility_functions	Lista funkcji użytecznosci
GET	http://openims.local/api/users/bob@open-ims.test/utility_functions/1	Pobiera konkretna funkcje (dla usera)
GET	http://openims.local/api/users/bob@open-ims.test/parameters/1	Pobiera parametry dla funkcji 1 (dla usera)
GET	http://openims.local/api/users/bob@open-ims.test/protocols	Pobiera protokoły, przypisane funkcje i parametry dla usera

Przykład

parameter.json:

```
{
    "utility_function_id" : 2,
    "name" : "MyBitrate",
    "default": 128
}
```

zapytanie:

```
curl -i -X DELETE http://openims.local/api/parameter/12

curl -i -X POST --data @parameter.json http://openims.local/api/parameters

curl -i -X GET http://openims.local/api/users/bob@open-ims.test/payments
```

odpowiedź:

```
HTTP/1.0 200 OK
Date: Wed, 21 Nov 2012 12:33:56 GMT
Server: Apache/2.2.22 (Win32) PHP/5.3.13
X-Powered-By: PHP/5.3.13
Content-Length: 0
Connection: close
Content-Type: text/html
```

Tabela kodów odpowiedzi

404	Not found	Zasób nie znaleziony
200	OK	Operacja udana
403	Forbidden	Błąd podczas zapisu (np. błąd bazy danych)
400	Bad request	Złe zapytanie (np. błędny JSON)