Student: IEREMIAȘ Viorel

Grupa: 244

Tema 3 – Serviciu web

Tematică: Agenda evenimente

# Cerință

*“Agenda: Intretinerea unei DB cu evenimente:*

*\* Adaugare / stergere / modificare eveniment*

*\* Lista cronologica a evenimentelor*

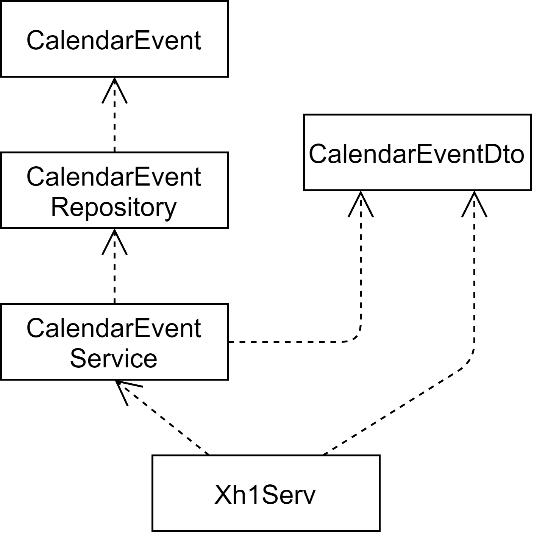
*\* Lista evenimente filtrata dupa parti din continut*

*Problema: Agenda, tehnologii:* *XML-RPC, serviciu: PHP, client: Python.”*

# Soluție propusă

## Server

In etapa de design a modelului de date se identifica nevoia unei singure entitati pentru reprezentarea evenimentelor. In implementare, aceasta se numeste **CalendarEvent** si are urmatoare campuri:

* + - * id – numeric, auto-incrementat, cu rol de cheie primara
* name – sir de caractere, de asemenea unic, cu lungimea maxima de 100 de caractere
* description – sir de caractere cu lungimea maxima de 500 de caractere
* scheduled\_time – structura de *date* in format ISO 8601

Arhitectura componentei Server este prezentata in Figura 1. **CalendarEventRepository** asigura legatura cu baza de data si opereaza cu obiecte de tip **CalendarEvent**, entitatea de business principala a aplicatiei.

Pentru a separa stratul de persistenta de cel de prezentare si din motive de siguranta, s-a definit entitatea **CalendarEventDto**, clasa care este folosita in comunicarea cu clientii. Aceasta contine aceleasi campuri ca entitatea pereche CalendarEvent, mai putin campul **id**, folosit intern de baza de date. Maparea intre cele doua este facuta de un set de metode statice din cele doua clase.

Figură 1. Arhitectura server

La nivelul de servicii, exista un **CalendarEventService**, careimplementeaza metodele expuse de catre serviciul web catre clienti. Implementarea este delegata spre nivelul de persistenta (clasa CalendarEventRepository in principal), insa la acest nivel se face translatarea intre CalendarEvent si CalendarEventDTO si invers. **Xh1Serv** instantiaza si porneste serviciul web *XML-RPC*, primeste cererile si le delega catre **CalendarEventService**.

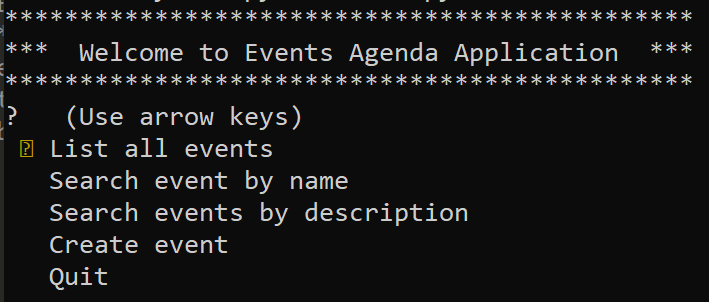
Serverul este implementat conform cerintei, in PHP. Persistenta datelor este realizata folosind o baza de date **MySQL.** Ca ORM se foloseste **MySQLi (MySQL Improved Extension)**, cel mai popular si folosit ORM pentru PHP. O data creata obiectul *Connection*, operatiile CRUD pe care de date sunt realizat ruland query-uri SQL native cu ajutorul conexiunii create.

## Clienti

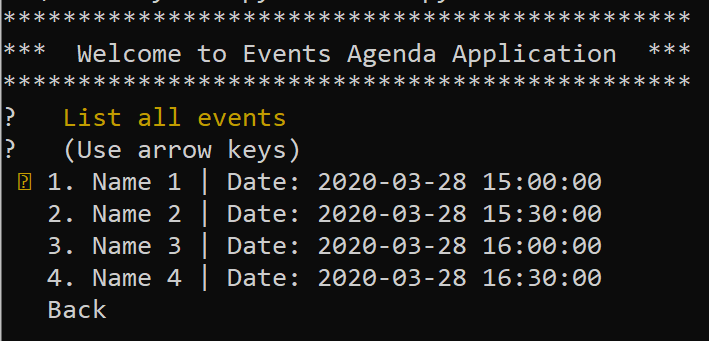
S-au implementat 2 clienti pentru serverul descris mai sus, unul de test, in PHP si unul in Python 3. Clientii folosesc partea de client a pachetelor *XML-*RPC disponibile in PHP si Python pentru a se conecta la serviciul web. Clientul Python este o aplicatie **CLI** (command-line interface) **interactiva** (permit navigarea cu ajutorul sagetilor). Clientul PHP afiseaza rezultatele operatiilor de baza ale serviciul pentru a putea verifica buna functionare a acestiua.

In pachetul **model** clientul Python contine specificatia clasei **CalendarEventDTO** si validatori pentru campurile entitatii (nume, descriere si data). Pachetul **cli** contine clase **enum** care contin optiunile afisate utilizatorului in diferite situatii in aplicatie. De asemenea, in acest pachet sunt definite metodele care proceseaza alegerile utilizatorului.

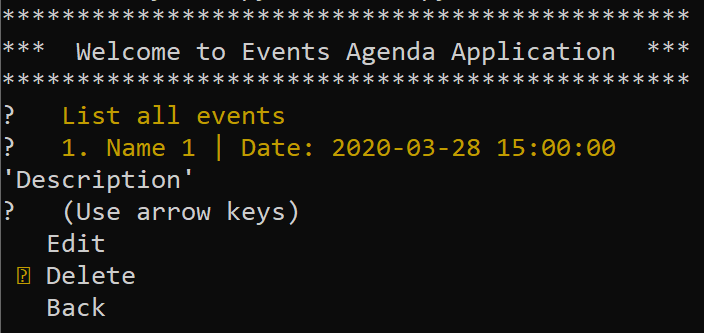
Implementarea interfetei interactive se bazeaza pe **PyInquirer**, un pachet care permite afisarea unei liste de optiuni, navigarea cu ajutorul sagetilor pentru selectarea unui optiuni si returnarea raspunsului.

 Clasa **Main** creeaza clientul *XML-RPC* si initializeaza interfata.

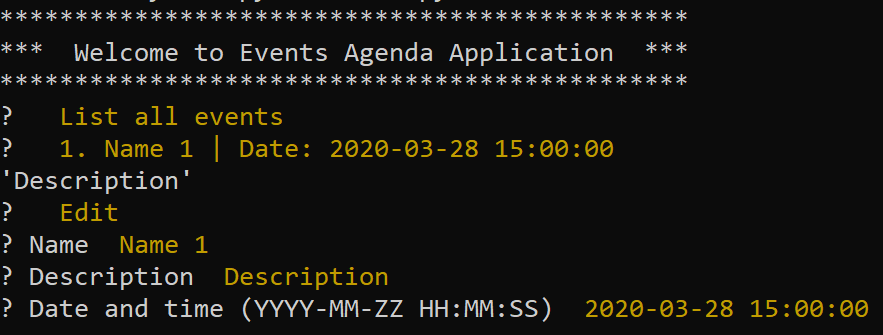
*Meniul principal*



*Lista evenimentelor ordonate cronologic*

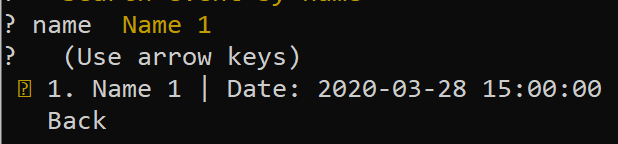


*Operatiunile disponibile pentru un eveniment*



*Formularul de editare a unui eveniment, cu valorile existente precompletate*

*(formularul de creare este identic, insa nu vine precompletat cu nicio valoare)*



*Cautarea unui eveniment dupa nume*

*(cautarea dupa cuvinte cheie este asemanatoare, insa nu la fel de stricta)*

# Instalare și rulare

1. Server

* Se copiaza continutul directorului Server in directorul htdocs al serverului web Apache
* Se porneste serverul web

1. Client Python

* Se ruleaza comanda pip install -r requirements.txt
* Se ruleaza comanda python main.py

1. Client PHP

* Se copiaza continutul directorului Server in directorul htdocs al serverului web Apache
* Se porneste serverul web
* Se acceaseaza in browser http://localhost/Xh1Clie.php