

Proiect Baze de Date2

- *Platforma tip wiki pentru jocul League of Legends* -

Duinea Eduard
342C3

I. Descrierea temei

Tema aleasa presupune construirea unei platforme informative pentru jocul League of Legends. League of Legends este un joc online competitiv, cu ritm alert, care combina viteza si intensitatea unui joc RTS cu elemente RPG. Doua echipe de campioni puternici, fiecare cu un design si stil de joc unic, se lupta direct pe diverse campuri de batalie [1si in moduri de joc diferite. [1]

Avand in vedere popularitatea necontestabila a jocului, proiectul acesta isi propune sa ofera functionalitatea de ghid pentru jucatorii noi ce patrund in universul acestui joc amplu, cu multe elemente si caracteristici unice si interesante.

Pentru dezvoltarea aplicatiei alese, au fost necesare 3 componente: o baza de date mysql, un server implementat cu ajutorul micro framework-ului Flask, si template-uri html pentru randarea informatiilor distribuite clientului.

II. Descrierea bazei de date

1. Diagrama bazei de date

Diagrama bazei de date este disponibila pe pagina urmatoare. Este o descriere grafica a tuturor elementelor prezente in baza de date aferenta aplicatiei, evidentian-du-se de asemenea si legaturile intre tabele respective.

2. Structura tabelelor

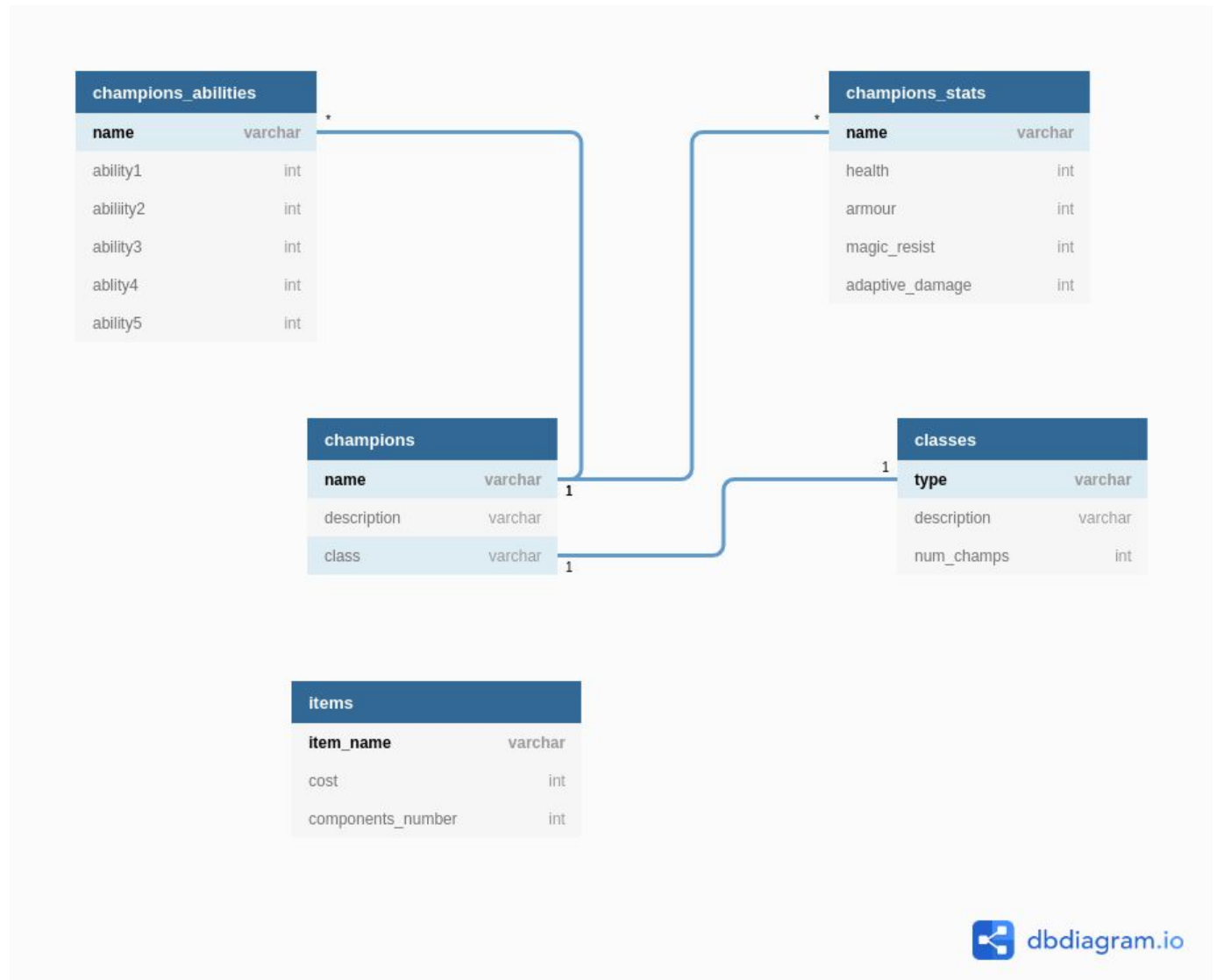
Tabelele evidentiate, in numar de 5, prezinta caracteristicile jocului League of Legends.

Tabela "champions" este tabela principala, aceasta ghidand la randul ei modul de dispunere al informatiei. Fiecare campion apartine uneia sau mai multor clase, de exemplu "mage" sau "assassin". Clasele sunt astfel reprezentate intr-o alta tabela, tinandu-se in acelasi timp o evidenta a numarului de campioni prin intermediul campului "num_champs". Luand in considerare acest lucru, definim o relatie de tipul one-to-one intre cele doua tabele mentionate, fiecare campion apartinand unei singure clase.

Campionii pot avea mai multe multe seturi de abilitati, de unde se explica si relatia one-to-many intre tabelele “champions” si “champions_abilities”.

“Champion_stats” este tabela care descrie capabilitatile fiecarui campion.

Tabela “items” descrie item-urile pe care campionii le pot cumpara in timpul meciului.



3. Descrierea constrangerilor de integritate

Fiecare campion are un nume unic, astfel ca am ales campul “name” ca fiind cheie primara pentru tabelele “champions”, “champions_stats” si “champions_abilities”.

Deoarece fiecare campion face parte dintr-o clasa unica, "type" este cheie primara in tabela "classes", si respectam o constrangere aferenta de cheie secundara pentru campul "class" din tabela "champions".

4. Descrierea procedurilor si functiilor

Procedurile si functiile stocate in interiorul bazei de date, au rolul de a oferi transparenta din punct de vedere al aplicatiei de backend (server-ul). Astfel, pe server, in aplicatia "go.py" din arhiva, se pot executa apeluri de proceduri si functii din interiorul bazei de date. Aceasta functionalitate permite modificarea functiilor / procedurilor stocate pe server-ul mysql, fara a fi necesara modificarea structurilor folosite in aplicatia server.

Functiile au ca scop oferirea unui mijloc de a realiza rapoarte pentru aplicatia noastra. Astfel, realizam totalul unor colectii de elemente, sau calculam date in functie de caracteristicile unui campion, folosind functii.

Procedurile ofera prilejul mentionat anterior, ci anume, pasarea transparenta a unor seturi de date catre aplicatia server.

Exemplu de functie:

```
DELIMITER $$
CREATE FUNCTION get_nr_items()
RETURNS int
BEGIN
    DECLARE num int;

    select count(*) into num from itemss;
    RETURN (num);
END$$
DELIMITER ;
```

Exemplu de procedura:

```
DELIMITER $$

CREATE PROCEDURE get_all_items()
BEGIN
    select * from itemss;
END $$

DELIMITER ;
```

Trigger-ul creat:

```
DELIMITER $$

CREATE TRIGGER before_itemss_delete
BEFORE DELETE
ON itemss FOR EACH ROW
BEGIN
    INSERT INTO itemss_archive(name, cost, components_number)
    VALUES(OLD.name, OLD.cost, OLD.components_number);
END$$

DELIMITER ;
```

III. Descrierea aplicatiei

Aplicatia este descrisa de catre cele doua componente importante: baza de date si server. Baza de date este reprezentata de catre un server mysql care ruleaza local, iar server-ul a fost construit folosind micro framework-ul Flask[2].

Astfel, aplicatia server a fost scrisa folosind limbajul de programare python. Componeta client care se conecteaza la server, este reprezentata prin intermediul browser-ului. Astfel, aplicatia server randeaza diferite fisiere html, in functie de input-ul oferit de client.

Diagrama de stari si fluxul de lucru

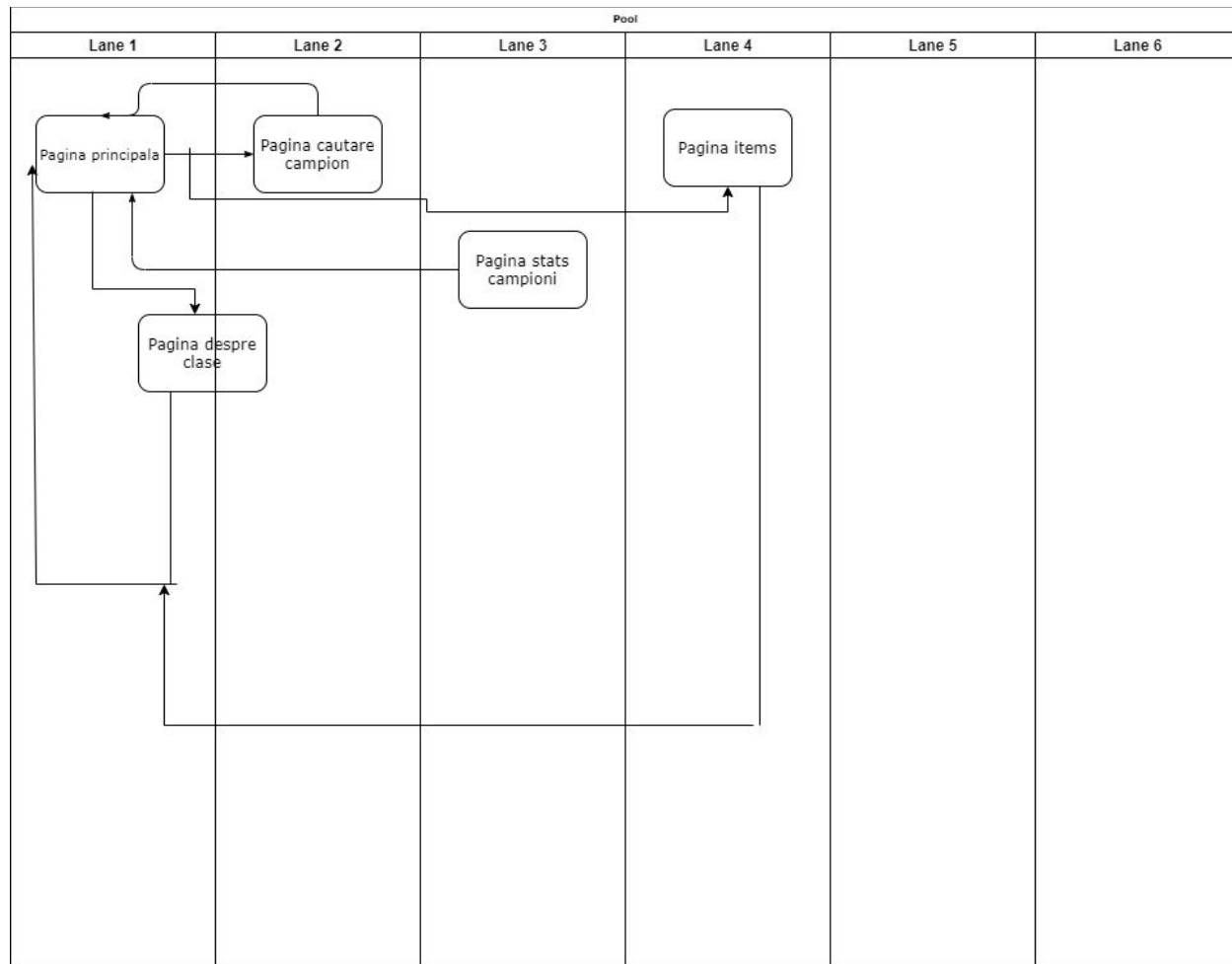


Diagrama de flux de mai sus prezinta toate posibilitatile care pot aparea folosind aplicatia. Este evidentiat faptul ca fiecare pagina are acces catre pagina principala, iar din pagina principala se poate accesa orice alta pagina.

Conexiunea la baza de date

Conexiunea la baza de date este descrisa in imaginea de mai jos. Conexiunea se face prin intermediul aplicatiei flask. Se realizeaza conexiunea la baza de date prin intermediul pachetului "mysql.connector", pasand ca parametru configurarile necesare pentru conectare. Obtinem astfel un obiect de tipul cursor, prin intermediul caruia interactionam cu baza de date. Putem scrie comenzi sql, saucel mai elegant se pot apela proceduri si functii stocate.

```
@app.route('/champs_raport', methods=['POST', 'GET'])
def process_raport_1():
    config = {
        'user': 'edi',
        'password': 'Password#6',
        'host': 'localhost ',
        'database': 'lol'
    }

    connection = mysql.connector.connect(**config)
    cursor = connection.cursor()

    cursor.execute('select * from champions_stats')
    results = [(n, a, b, c, d) for (n, a, b, c, d) in cursor]
```

IV. Vizualizare

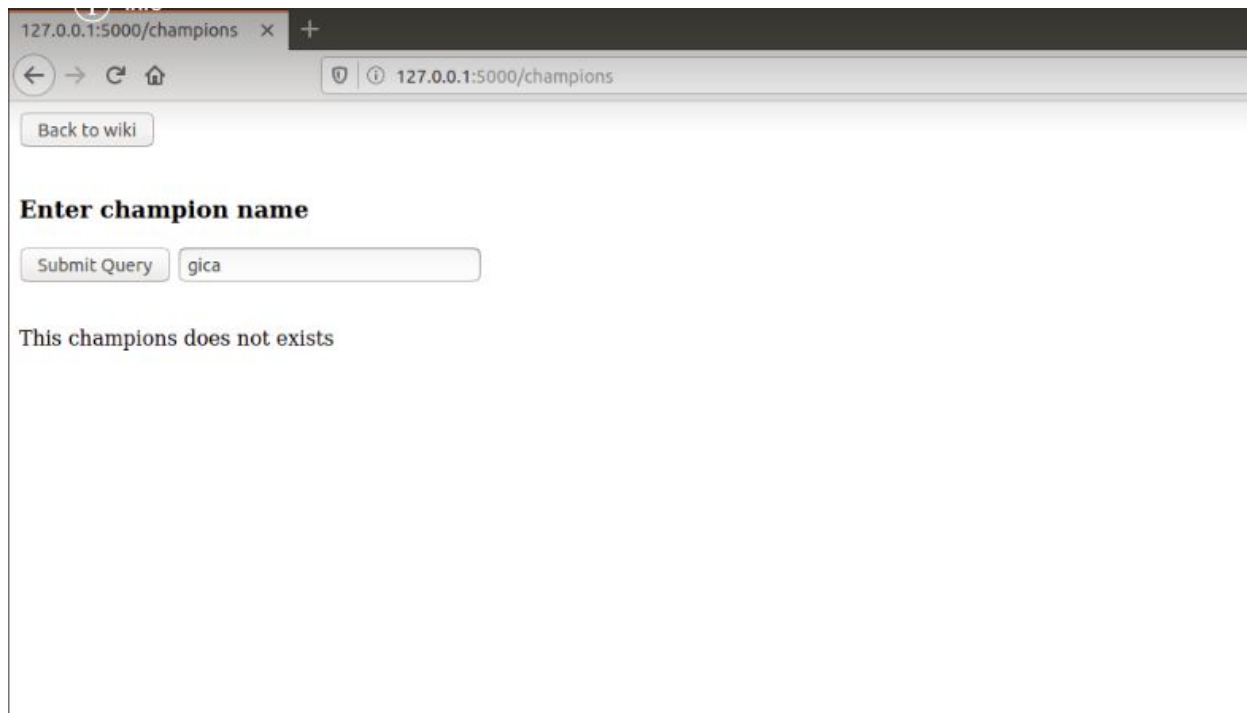
Pagina principala a aplicatiei

League of Legends WIKI

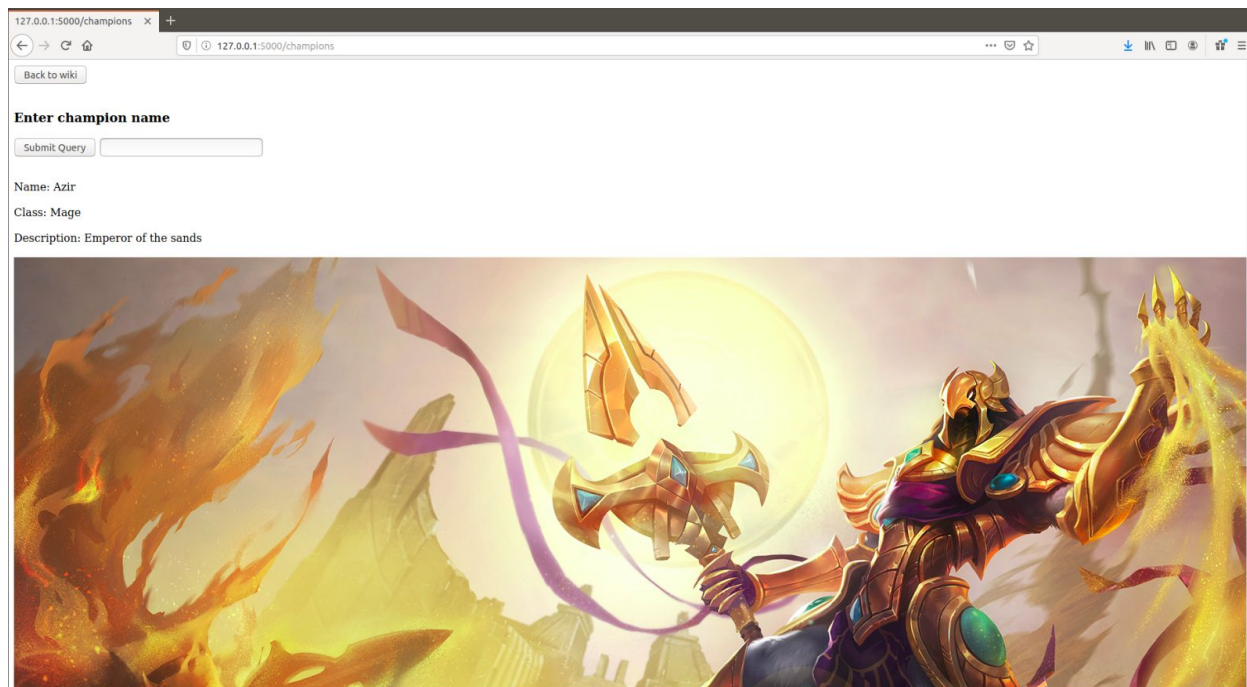
League of Legends is a multiplayer online battle arena video game developed and published by Riot Games. While the company and its first game are continuously growing, now it's the time to discover more about LoL's universe



Ecranul afisat in momentul in care se cauta un campion care nu se afla in baza de date.



Ecranul afisat in momentul gasirii unui campion in baza de date.



Ecranul afisat pentru a vedea stats-urile fiecarui campion si rapoartele corespunzatoare acestora.

127.0.0.1:5000/champs_raport X +

Back to wiki

Champion Azir

Health: 100
Armour: 20
Magic Resist: 30
Adaptive Damage: 40

Force power based on the above stats: 47.5

Champion Anivia

Health: 120
Armour: 15
Magic Resist: 25
Adaptive Damage: 40

Force power based on the above stats: 50.0

Champion Akali

Health: 110
Armour: 17
Magic Resist: 35
Adaptive Damage: 45

Force power based on the above stats: 51.75

Champion Bard

Health: 99
Armour: 20
Magic Resist: 22
Adaptive Damage: 40

Force power based on the above stats: 45.25

Champion Evelynn

Health: 8
Armour: 20
Magic Resist: 20
Adaptive Damage: 50

Force power based on the above stats: 24.5

Champion Irelia

Ecranul cu informatii despre item-urile din joc.

127.0.0.1:5000/items_raport X +

Bloodthirster 3500 4

Frostfang 500 0

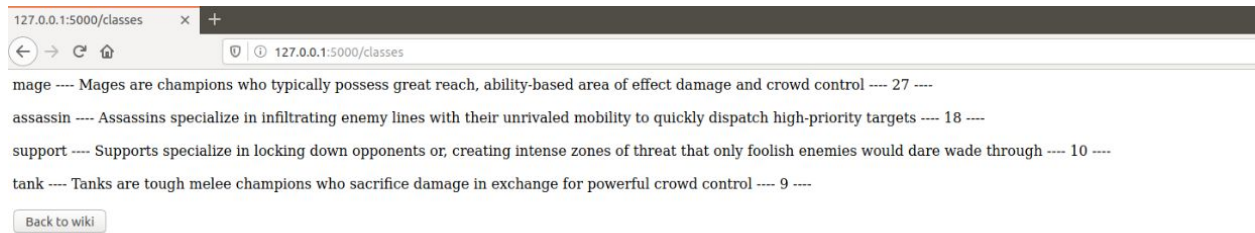
Guardian Angel 2800 3

Iceborn gauntlet 2700 2

Total: 4

Back to wiki

Ecranul cu informatii despre clasele din joc.



V. *Concluzie*

Am generat astfel o aplicatiei care ofera informatii despre jocul luat in considerare, folositoare jucatorilor noi. Atat partea de server, cat si baza de date au fost construite astfel incat sa fie usor de adaptat in momentul aparitiei de elemente noi din joc, ce trebuiesc adaugate in aplicatie.

VI. Bibliografie

- [1] : <https://eune.leagueoflegends.com/ro/game-info/get-started/what-is-lol/>
- [2]: <http://flask.palletsprojects.com/en/1.1.x/>
- [3]: <https://eune.leagueoflegends.com/en/> -> informatii despre joc
- [4]: https://docs.oracle.com/cd/E17952_01/index.html -> documentatie mysql
- [5]: <https://www.mysqltutorial.org> -> informatii despre proceduri si functii