



# RuneterraDB

Elaborato Basi di Dati

A.A 2022/2023

Pagliarani Martino

Matricola: 0001028990

[martino.pagliarani@studio.unibo.it](mailto:martino.pagliarani@studio.unibo.it)

Semproli Mattia

Matricola: 0001030793

[mattia.semproli2@studio.unibo.it](mailto:mattia.semproli2@studio.unibo.it)

# Indice

## **Analisi dei requisiti**

- ❖ **Intervista** **3**
- ❖ **Estrazione dei concetti principali** **4**

## **Progettazione Concettuale**

- ❖ **Schema scheletro** **7**
- ❖ **Schema finale** **9**

## **Progettazione Logica**

- ❖ **Stima del volume dei dati** **10**
- ❖ **Descrizione delle operazioni principali e stima della loro frequenza** **11**
- ❖ **Schemi di navigazione e tabelle degli accessi** **12**
- ❖ **Raffinamento dello schema** **18**
- ❖ **Analisi delle ridondanze** **19**
- ❖ **Traduzione di entità e associazioni in relazioni** **20**
- ❖ **Schema Relazione Finale** **21**
- ❖ **Traduzione delle operazioni in query SQL** **22**

## **Progettazione dell'Applicazione**

- ❖ **Descrizione dell'architettura dell'applicazione realizzata** **27**

# Analisi dei requisiti

Il progetto prevede la creazione di un sito web che contenga una wiki (archivio) con le statistiche e informazioni dei personaggi del videogioco online League of Legends, la possibilità di registrarsi come giocatore, di giocare ad un minigioco e poter osservare il proprio posizionamento nella classifica.

## Intervista

Il seguente testo rappresenta a grandi linee l'idea del progetto:

Il sito comprende una parte di login e register tramite la quale si può accedere come utente normale o utente amministratore, quest'ultimo è l'unico che può modificare le informazioni del database mentre l'utente normale può solo visualizzarle.

Il sito contiene una parte informativa di "wiki" che è accessibile anche agli utenti "guest", ovvero che non hanno eseguito l'accesso.

Accedendo si sblocca la possibilità di collegare il profilo user al proprio profilo "summoner" che può far parte di un "team" (composto da 5 giocatori). Ogni summoner può giocare al minigioco tramite il quale si completano delle missioni che hanno un nome, una descrizione e che danno un punteggio.

Ogni summoner ha la possibilità di vedere il proprio punteggio nella classifica dei summoner, invece la classifica dei team è composta dalla somma dei punteggi di tutti i summoner che appartengono ad un team.

La "wiki" è composta da due parti: la schermata di sinistra permette di selezionare da una lista, che contiene tutti i campioni, il personaggio di cui si vogliono sapere le informazioni quali nome, biografia, statistiche e ruoli che appaiono nella seconda parte ovvero la schermata di destra. Ogni campione possiede una regione di appartenenza a cui è legata la propria biografia e una serie comune di statistiche di gioco diverse per ognuno di essi.

Il minigioco, accessibile solamente dopo aver eseguito l'accesso, si chiama "Guess the Champion" il cui scopo è indovinare il nome del campione selezionato casualmente dal database tramite degli indizi che sono: il genere,

il ruolo, la specie, il tipo di risorsa, il tipo di attacco e la regione di appartenenza. Se corrispondono a quelli del campione da indovinare si colorano di verde, di rosso altrimenti. Tramite la classifica è possibile tenere traccia dei punti che hai ottenuto e del rank a cui sei arrivato.

## Estrazione dei concetti principali

Termine	Breve descrizione	Sinonimi Eventuali
Utente	Colui che si registra al sito e ne utilizza i contenuti (Guest se non è registrato).	User
Summoner	Account giocatore collegato univocamente ad un utente.	Player, giocatore
Team	Gruppo di massimo 5 summoner.	
Missione	Sfida che ogni summoner può completare.	Mission
Minigioco	Compone la parte videoludica del sito.	
Partita	Specifica partita giocata da un summoner nel minigioco	Game
Wiki	Compone la parte informativa del sito.	
Campione	Ogni personaggio presente nella wiki sui quali viene basato il minigioco.	Champion, personaggio
Biografia	Storia dei personaggi.	Lore
Statistiche di gioco	Statistiche di gioco dei personaggi.	Stats
Classifica	Classifica ordinata per punteggio dei team e dei summoner.	

Alla luce dell'intervista è possibile estrarre i concetti principali, riassunti nel seguente testo eliminandone le ambiguità:

L'**utente** può effettuare operazioni di login e register, ognuno di essi possiede un unico account **summoner** che viene creato al momento della registrazione.

Un account **summoner** possiede un nome e ha la possibilità di giocare ad un minigioco un numero indeterminato di volte, che però gli permette di ottenere punti tramite la vittoria solo per le prime 5 partite del giorno (minori i tentativi maggiori i punti ottenuti). Dopo un determinato numero progressivo di partite vinte si completano diverse missioni che assegnano ulteriori punti, tramite i quali il summoner viene assegnato al corretto rank.

Un **team** è composto da massimo 5 summoner, possiede un nome, l'acronimo e la somma dei punti dei componenti.

Una **missione** possiede una descrizione, un goal ovvero il traguardo necessario da raggiungere per ottenere i corrispettivi punti.

Il **minigioco** si chiama "Guess the Champion" ed è giocabile solo dopo aver effettuato l'accesso e consiste nell'indovinare un campione casuale aiutato dalle specifiche caratteristiche, una **partita** è caratterizzata dal numero di tentativi utilizzati per vincere, la data, il campione da indovinare e il nome del summoner che ha giocato.

La **wiki** è accessibile anche agli utenti che non hanno effettuato l'accesso.

Un **campione** possiede un identificatore univoco, un nome, il genere, il ruolo, la specie, il tipo di risorsa utilizzata, il tipo di attacco e la regione di appartenenza. Oltre a queste nella wiki vengono anche mostrate la **biografia** e le **statistiche di gioco** (come la velocità di movimento).

La **classifica** mostra ordinati per punteggio sia i summoner che i team.

Di seguito un elenco con le principali azioni:

1. Registrare un nuovo utente.
2. Creare il profilo summoner dopo la registrazione.
3. Accedere con i propri dati utente.
4. Selezionare un champion casualmente.
5. Effettuare un tentativo durante il minigioco.
6. Visualizzare la classifica dei summoner.
7. Aggiornare la classifica dei team.
8. Visualizzare la classifica dei team.
9. Selezionare uno specifico champion dalla wiki.
10. Inserire una partita giocata.
11. Aggiornare il punteggio di un summoner dopo una vittoria.
12. Aggiornare il rank di un summoner.
13. Aggiornare una missione completata da un summoner.
14. Aggiornare il numero di partite vinte da un summoner.
15. Aggiungere i punti bonus di una missione completata al punteggio del summoner.
16. Controllare il limite delle partite giocate in un giorno da un summoner.

# Progettazione Concettuale

## Schema scheletro

Le entità **BASE\_USER** e **ADMIN** sono la generalizzazione dell'entità **USER** che ha come identificatore uno username univoco. A questo proposito le due entità sono state collegate a user attraverso una gerarchia totale e disgiunta dal momento che tutti gli user devono appartenere ad una delle due categorie. Si può notare che uno user sia base sia admin, possiede un unico account **SUMMONER** di conseguenza la chiave primaria di quest'ultimo è la stessa dello user di riferimento.

Un summoner può giocare un numero indeterminato di **GAME** che hanno id univoco, questo porta al completamento di una o più **MISSION**, ognuna di esse è identica per ogni summoner, ma successivamente all'interno della relazione **COMPLETE**, ne sarà salvato il progresso individuale.

Un **TEAM** è composto da massimo 5 summoner ed è identificato da id univoco.

Ogni summoner in base al proprio score ottiene un **RANK** identificato da id univoco.

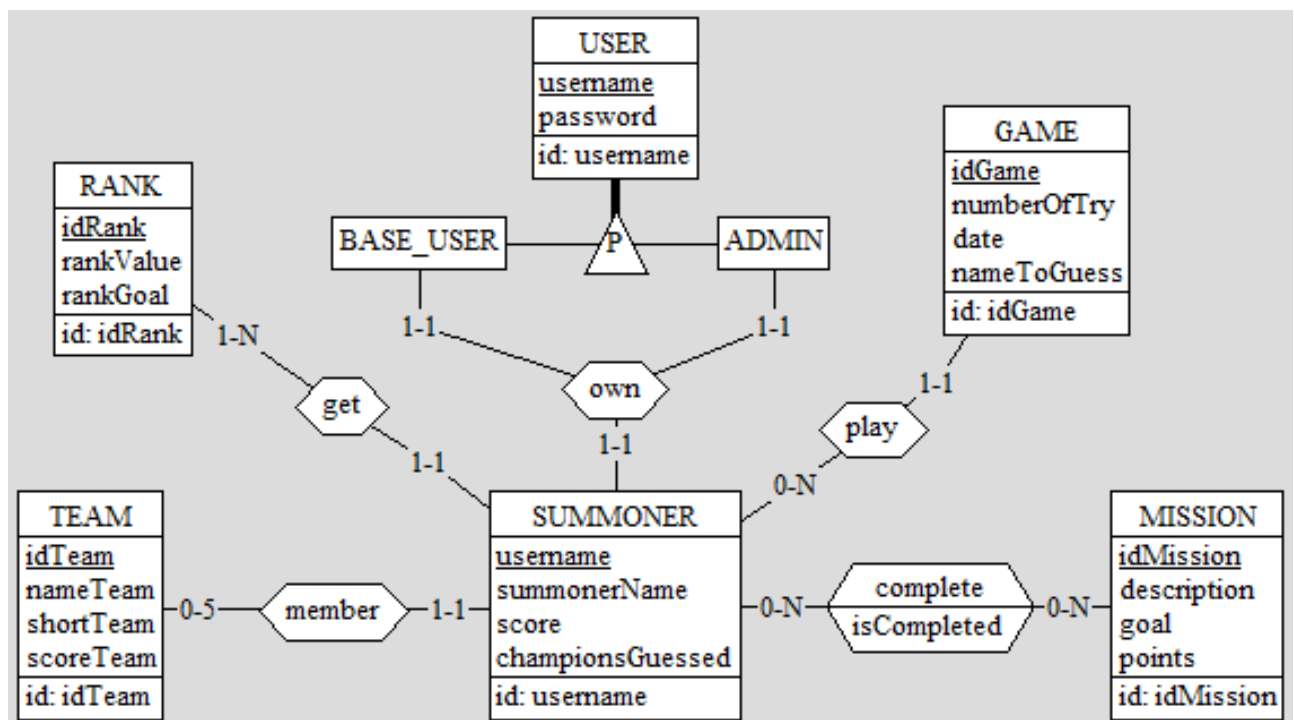


Figura 1.1 Schema concettuale della parte interagibile da un utente registrato.

Lo scopo di ogni game è indovinare un **CHAMPION** identificato univocamente da un id. La tabella **STATS** indica le statistiche di gioco di un determinato champion, essendo relazione univoca tra le due, stats come identificatore possiede l'id univoco del champion associato, vale la stessa cosa per l'entità **LORE**, dato che rappresenta la storia univoca di ogni champion. Ogni champion possiede altre caratteristiche: **REGION** di appartenenza, **GENDER** in cui si identifica, **SPECIES** a cui appartiene, **ROLE** in cui si posiziona, **RESOURCE** che utilizza, tipologia d'**ATTACCO**. Inizialmente avevamo optato di aggiungerle come attributi dell'entità champion, ma successivamente, dato che i loro valori avevano dominio predefinito, abbiamo deciso di separarle in più entità per agevolare modifiche future, eliminare eventuali ridondanze e per un corretto uso delle forme normali; tutte sono identificate da id univoco.

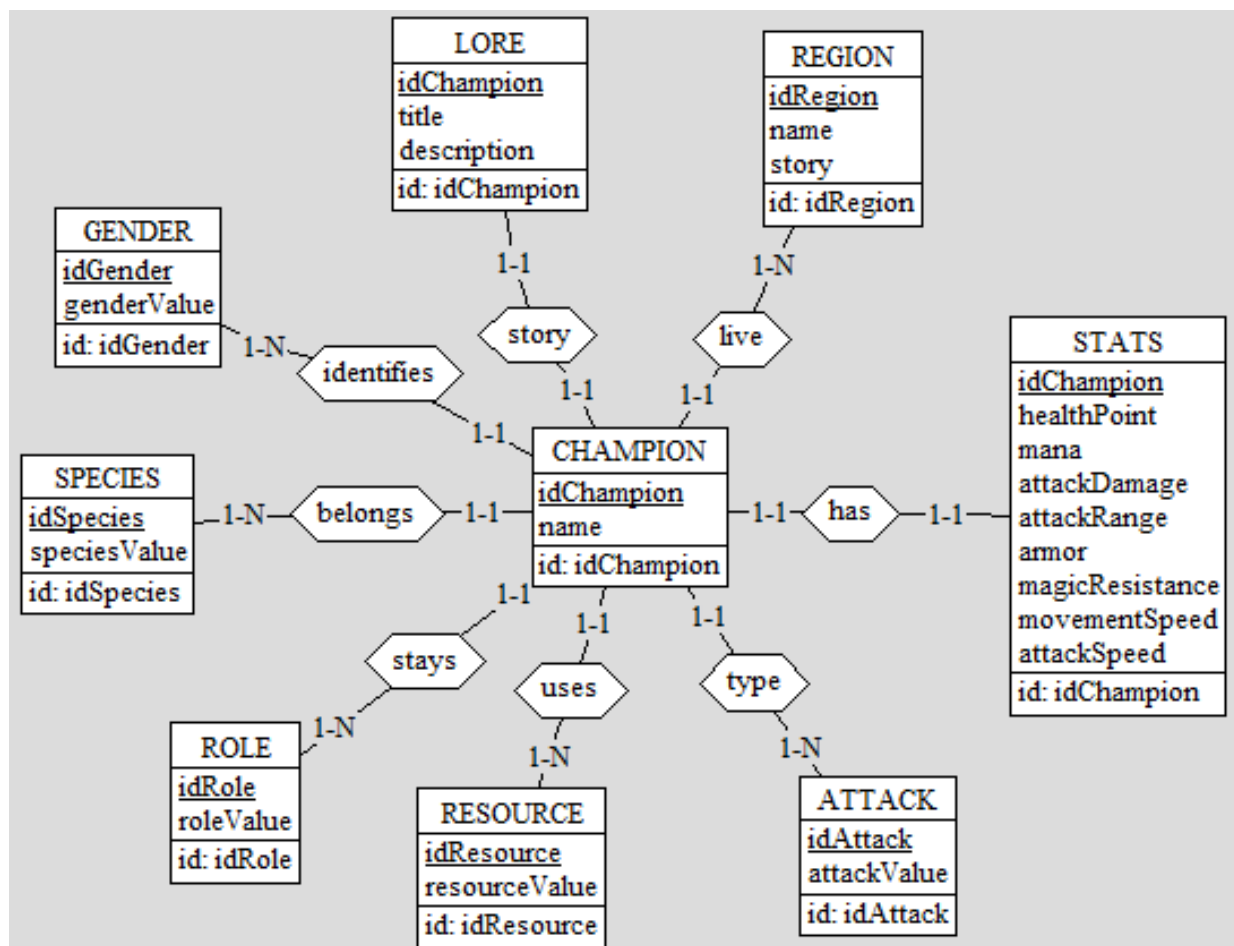


Figura 1.2 Schema concettuale della parte informativa.



## Schema finale

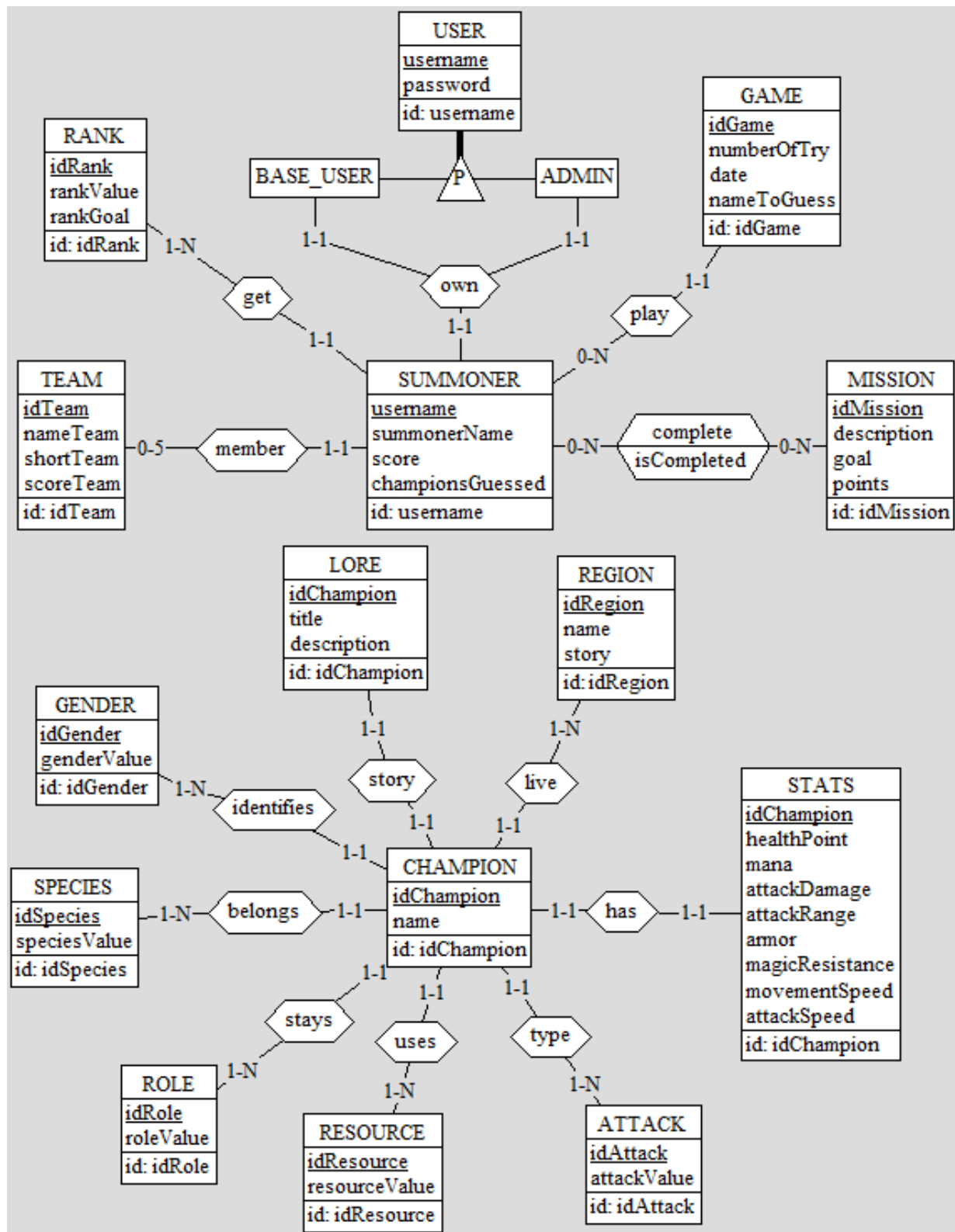


Figura 1.3 Schema concettuale finale.

# Progettazione Logica

## Stima del volume dei dati

Di seguito la tabella con la stima dei volumi di un mese:

Concetto	Costrutto	Volume	Concetto	Costrutto	Volume
User	E	1 000	Has	R	163
Own	R	1 000	Stats	E	163
Summoner	E	1 000			
			Type	R	2
Play	R	150 000	Attack	E	2
Game	E	150 000			
			Uses	R	13
Complete	R	9 000	Resource	E	13
Mission	E	9			
			Stays	R	5
Member	R	4	Role	E	5
Team	E	4			
			Belongs	R	28
Get	R	10	Species	E	28
Rank	E	10			
			Identifies	R	3
Champion	E	163	Gender	E	3
Live	R	14			
Region	E	14	Story	R	163
			Lore	E	163

## Descrizione delle operazioni principali e stima della loro frequenza

Codice	Operazione	Frequenza
1	Registrare un nuovo utente.	30 al giorno
2	Creare il profilo summoner dopo la registrazione.	30 al giorno
3	Accedere con i propri dati utente.	60 al giorno
4	Selezionare un champion casualmente.	200 al giorno
5	Effettuare un tentativo durante il minigioco.	1200 al giorno
6	Visualizzare la classifica dei summoner.	200 al giorno
7	Aggiornare la classifica dei team.	200 al giorno
8	Visualizzare la classifica dei team.	200 al giorno
9	Selezionare uno specifico champion dalla wiki.	600 al giorno
10	Inserire una partita giocata.	200 al giorno
11	Aggiornare il punteggio di un summoner dopo una vittoria.	200 al giorno
12	Aggiornare il rank di un summoner.	1 al giorno
13	Aggiornare una missione completata da un summoner.	1 al giorno
14	Aggiornare il numero di partite vinte da un summoner.	200 al giorno
15	Aggiungere i punti bonus di una missione completata al punteggio del summoner.	1 al giorno
16	Controllare il limite delle partite giocate in un giorno da un summoner.	200 al giorno

# Schemi di navigazione e tabelle degli accessi

Di seguito le tabelle degli accessi riguardanti le operazioni sopra elencate, le operazioni di scrittura vengono considerate di peso doppio rispetto a quello di lettura:

## OP 1 - Registrare un nuovo utente.

Per registrare un nuovo utente, bisogna salvarne i dati nell'entità corrispondente, presupponendo sia già stato controllato che non esista un utente uguale.

Concetto	Costrutto	Accessi	Tipo
User	E	1	S
Totale: 1S -> 60 al giorno			

## OP 2 - Creare il proprio profilo summoner dopo la registrazione.

Per la creazione del profilo summoner è necessario inserire i dati all'interno dell'entità, tra quei dati è presente anche il codice di un team.

Concetto	Costrutto	Accessi	Tipo
Team	E	1	L
Member	R	1	L
Summoner	E	1	S
Totale: 1S + 2L -> 120 al giorno			

## OP 3 - Accedere con i propri dati utente.

Per accedere con i propri dati utente basta controllare che sia corretti.

Concetto	Costrutto	Accessi	Tipo
User	E	1	L
Totale: 1L -> 30 al giorno			

#### OP 4 - Selezionare un champion casualmente.

Per selezionare un champion casualmente da indovinare nel minigioco, è necessario leggerlo dall'omonima entità, siccome nel gioco utilizziamo le caratteristiche come indizi, è necessario leggere anch'esse.

Concetto	Costrutto	Accessi	Tipo
Champion	E	1	L
Gender	E	1	L
Identifies	R	1	L
Role	E	1	L
Stays	R	1	L
Species	E	1	L
Belongs	R	1	L
Resource	E	1	L
Uses	R	1	L
Attack	E	1	L
Type	R	1	L
Region	E	1	L
Live	R	1	L
Totale: 13L -> 2600 al giorno			

**OP 5 - Effettuare un tentativo durante il minigioco.**

Effettuare un tentativo durante il minigioco richiede le stesse letture della selezione di un champion casuale.

**Totale: 13L -> 2 600 al giorno**

**OP 6 - Visualizzare la classifica dei summoner.**

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	L
Member	R	1	L
Team	E	1	L
Get	R	1	L
Rank	E	1	L

**Totale: 5L -> 1000 al giorno**

**OP 7 - Aggiornare la classifica dei team.**

Per aggiornare la classifica dei team è necessario aggiornare i team con la somma dei punti dei summoner componenti.

Concetto	Costrutto	Accessi	Tipo
Team	E	1	S
Member	R	4	L
Summoner	E	4	L

**Totale: 1S + 8L -> 2000 al giorno**

**OP 8 - Visualizzare la classifica dei team.**

Concetto	Costrutto	Accessi	Tipo
Team	E	1	L
<b>Totale: 1L -&gt; 200 al giorno</b>			

**OP 9 - Selezionare uno specifico champion dalla wiki.**

Per selezionare uno specifico champion nella wiki, occorre effettuare le stesse letture dell'operazione 5 che ci restituisce il champion richiesto con le caratteristiche principali, ma con aggiunta della biografia e delle statistiche.

Concetto	Costrutto	Accessi	Tipo
OP 5	...	...	...
Lore	E	1	L
Story	R	1	L
Stats	E	1	L
Has	R	1	L
Champion	E	2	L
<b>Totale: 19L -&gt; 11 400 al giorno</b>			

**OP 10 - Inserire una partita giocata.**

Concetto	Costrutto	Accessi	Tipo
Game	E	1	S
<b>Totale: 1S -&gt; 400 al giorno</b>			

**OP 11 - Aggiornare il punteggio di un summoner dopo una vittoria.**

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	S
Totale: 1S -> 400 al giorno			

**OP 12 - Aggiornare il rank di un summoner.**

Per aggiornare il rank di un summoner occorre controllare tramite il suo score a quale rank corrisponde.

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	S
Get	R	1	L
Rank	E	1	L
Totale: 1S + 2L -> 4 al giorno			

**OP 13 - Aggiornare una missione completata da un summoner.**

Supponiamo di aver già controllato che il summoner abbia completato missione, quindi occorre solo modificare l'attributo isCompleted della relazione complete sulla corretta missione.

Concetto	Costrutto	Accessi	Tipo
Complete	R	1	S
Complete	R	2	L
Totale: 1S + 2L -> 4 al giorno			



**OP 14 - Aggiornare il numero di partite vinte da un summoner.**

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	S
Totale: 1S -> 400 al giorno			

**OP 15 - Aggiungere i punti bonus di una missione completata al punteggio del summoner.**

Per aggiungere i punti bonus di una missione completata, occorre modificare lo score in summoner aggiungendo i punti ricavati dalla lettura della missione completata.

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	S
Complete	R	2	L
Mission	E	1	L
Totale: 1S + 3L -> 5 al giorno			

**OP 16 - Controllare il limite delle partite giocate in un giorno da un summoner.**

Occorre solo leggere da game quanti partite il summoner ha giocato nella data di oggi.

Concetto	Costrutto	Accessi	Tipo
Summoner	E	1	L
Totale: 1L -> 200 al giorno			

# Raffinamento dello schema

## Eliminazione delle gerarchie

Nello schema concettuale è presente una gerarchia totale e disgiunta tra BASE\_USER e ADMIN. È stato deciso di eliminarla tramite un collasso verso l'alto inserendo un attributo "isAdmin" nell'entità user, in quanto le due entità possono eseguire quasi tutte le stesse operazioni. Per ipotetiche operazioni con diverso livello di permessi, sarà sufficiente controllare l'attributo appena spiegato.

## Eliminazione degli attributi composti

Non è presente alcun attributo composto all'interno del nostro schema concettuale.

## Scelta delle chiavi primarie

Le chiavi primarie di ogni entità sono già spiegate ed evidenziate all'interno dello schema concettuale, ci teniamo a rammentare che la chiave primaria di LORE e STATS è la stessa del CHAMPION di riferimento dato che sono relazioni biunivoche, vale lo stesso ragionamento per USER e SUMMONER.

## Eliminazione degli identificatori esterni

Nello schema E/R precedentemente illustrato sono state rimosse le seguenti relazioni:

- ❖ Play, importando username in GAME.
- ❖ Complete, reificata importando username da SUMMONER e idMission da MISSION.
- ❖ Member, importando idTeam in SUMMONER.
- ❖ Get, importando idRank in SUMMONER.
- ❖ Type, importando idAttack in CHAMPION.
- ❖ Uses, importando idResource in CHAMPION.
- ❖ Stays, importando idRole in CHAMPION.
- ❖ Belongs, importando idSpecies in CHAMPION.
- ❖ Identifies, importando idGender in CHAMPION.
- ❖ Live, importando idRegion in CHAMPION.

## Analisi delle ridondanze

In team è stata aggiunta una ridondanza, scoreTeam, in quanto si potrebbe derivare dalla somma degli score dei summoner presenti nel team.

Di seguente è riportata il risparmio in termine di accessi.

### OP 8 - Visualizzare la classifica dei team.

Tramite la ridondanza per visualizzare la classifica dei team occorre solamente leggere dalla tabella team ordinata per attributo scoreTeam.

Concetto	Costrutto	Accessi	Tipo
Team	E	1	L
Totale: 1L -> 200 al giorno			

Senza la ridondanza invece, sarebbe necessario ricalcolare ogni volta la somma dei punteggi dei summoner del team.

Concetto	Costrutto	Accessi	Tipo
Team	E	1	L
Member	R	4	L
Summoner	E	4	L
Totale: 1S + 8L -> 2000 al giorno			

## Traduzione di entità e associazioni in relazioni

- ❖ **users**(username, password)
- ❖ **summoners**(username: users, summonerName, score, championsGuessed, rank: ranks, team: teams)
- ❖ **games**(idGame, numberOfTry, date, nameToGuess, username: summoners)
- ❖ **complete**(isCompleted, username: users, idMission: missions)
- ❖ **missions**(idMission, description, goal, points)
- ❖ **ranks**(idRank, rankValue, rankGoal)
- ❖ **teams**(idTeam, nameTeam, shortTeam, scoreTeam)
- ❖ **champions**(idChampion, name, gender: genders, role: roles, species: species, resource: resources, attack: attacks, region: regions)
- ❖ **stats**(idChampion: champions, healthPoint, mana, attackDamage, attackRange, armor, magicResistance, movementSpeed, attackSpeed)
- ❖ **attacks**(idAttack, attackValue)
- ❖ **resources**(idResource, resourceValue)
- ❖ **roles**(idRole, roleValue)
- ❖ **species**(idSpecies, speciesValue)
- ❖ **genders**(idGender, genderValue)
- ❖ **lores**(idChampion: champions, title, description)
- ❖ **regions**(idRegion, name, story)

# Schema Relazione Finale

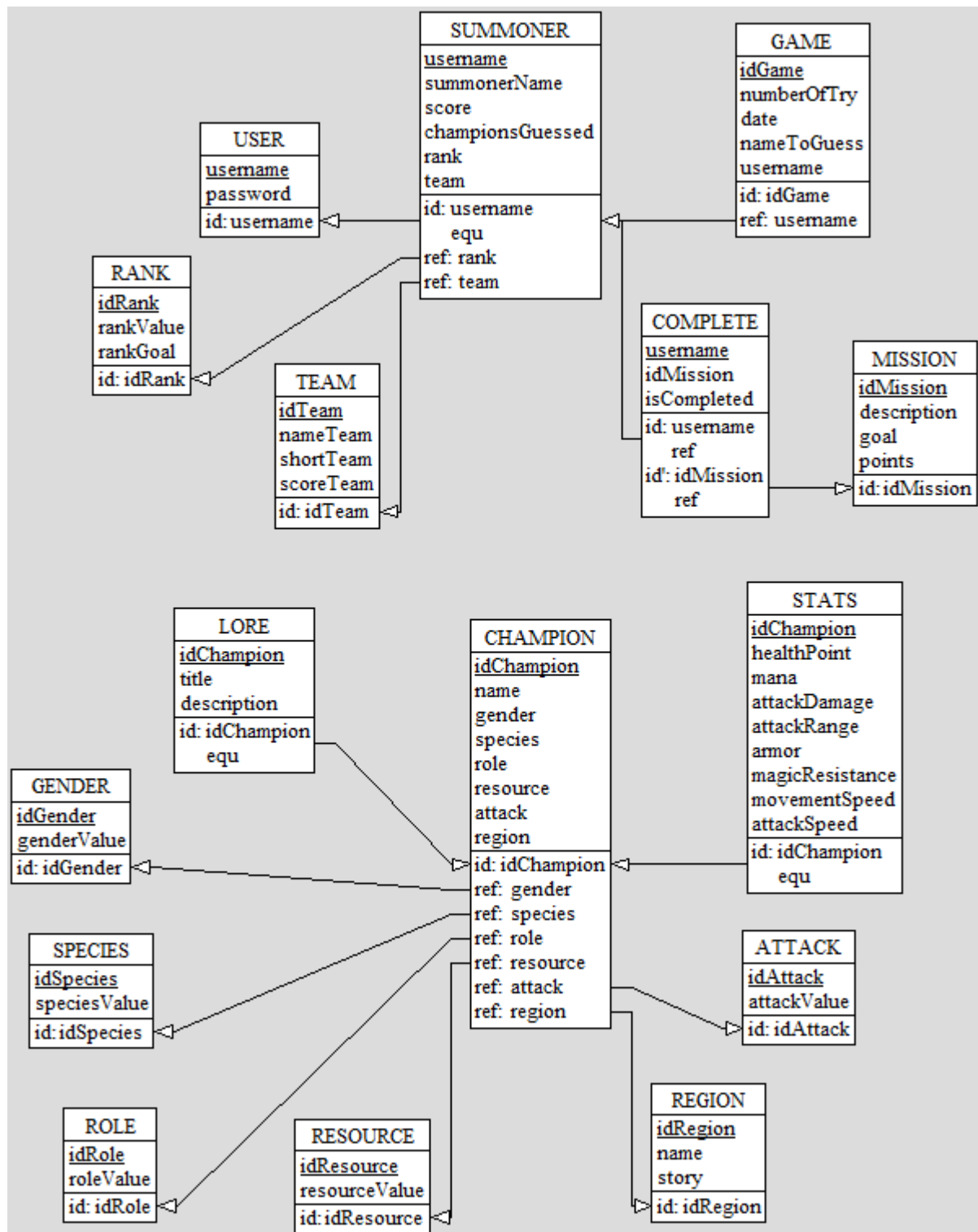


Figura 1.4 Schema relazionale finale.

# Traduzione delle operazioni in query SQL

## OP 1 - Registrare un nuovo utente.

```
INSERT INTO user(username,password)
VALUES (?, ?);
```

## OP 2 - Creare il proprio profilo summoner dopo la registrazione.

```
INSERT INTO summoner (username,summonerName,team)
VALUES (?, ?, (SELECT team.idTeam
                FROM team
                WHERE LOWER(REPLACE(team.nameTeam, ' ', '')) = ?));
```

## OP 3 - Accedere con i propri dati utente.

```
SELECT *
FROM user
WHERE username = ? AND password = ?;
```

## OP 4 - Selezionare un champion casualmente.

```
SELECT champion.name AS name,
       gender.genderValue AS gender,
       role.roleValue AS role,
       species.speciesValue AS species,
       resource.resourceValue AS resource,
       attack.attackValue AS attackType,
       region.name AS region
FROM champion
JOIN gender ON champion.gender = gender.idGender
JOIN role ON champion.role = role.idRole
JOIN species ON champion.species = species.idSpecies
JOIN resource ON champion.resource = resource.idResource
JOIN attack ON champion.attack = attack.idAttack
JOIN region ON champion.region = region.idRegion
WHERE champion.idChampion = ?;
```

### OP 5 - Effettuare un tentativo durante il minigioco.

Come era stato anticipato sopra nella sezione delle [tabelle degli accessi](#), la query di questa operazione è identica a quella della OP 4, la differenza risiede nel fatto che per il champion casuale, viene effettuato il SELECT con il WHERE sull'id champion, mentre in questo caso del tentativo, il SELECT viene effettuato con il WHERE sul nome del champion dato che ne richiediamo uno specifico.

```
SELECT champion.name AS name,
       gender.genderValue AS gender,
       role.roleValue AS role,
       species.speciesValue AS species,
       resource.resourceValue AS resource,
       attack.attackValue AS attackType,
       region.name AS region
FROM champion
JOIN gender ON champion.gender = gender.idGender
JOIN role ON champion.role = role.idRole
JOIN species ON champion.species = species.idSpecies
JOIN resource ON champion.resource = resource.idResource
JOIN attack ON champion.attack = attack.idAttack
JOIN region ON champion.region = region.idRegion
WHERE champion.name = ?;
```

### OP 6 - Visualizzare la classifica dei summoner.

```
SELECT summoner.summonerName AS name,
       summoner.score,
       summoner.championsGuessed AS guesses,
       team.shortTeam,
       `rank`.rankValue
FROM summoner
JOIN team ON summoner.team = team.idTeam
JOIN `rank` ON summoner.rank = `rank`.idRank
ORDER BY summoner.score DESC;
```

### OP 7 - Aggiornare la classifica dei team.

```
UPDATE team
JOIN (SELECT team.idTeam, SUM(summoner.score) AS tot
      FROM summoner
      JOIN team ON summoner.team = team.idTeam
      GROUP BY team.idTeam) AS res
ON team.idTeam = res.idTeam
SET team.scoreTeam = res.tot;
```

### OP 8 - Visualizzare la classifica dei team.

```
SELECT team.nameTeam AS team,
       team.shortTeam AS short,
       team.scoreTeam AS score
FROM team
ORDER BY team.scoreTeam DESC;
```

### OP 9 - Selezionare uno specifico champion dalla wiki.

Come specificato nella sezione delle [tabelle degli accessi](#), la prima parte è la identica della OP 5, ma in questo caso sono aggiunte le seguenti query:

```
SELECT healthPoint,
       mana,
       attackDamage,
       attackRange,
       armor,
       magicResistance,
       attackSpeed,
       movementSpeed
FROM stats
WHERE idChampion = (SELECT idChampion
                   FROM champion
                   WHERE name = ?);
```



```
SELECT title AS nickname,
       description AS bio
FROM lore
WHERE idChampion = (SELECT idChampion
                    FROM champion
                    WHERE name = ?);
```

#### OP 10 - Inserire una partita giocata.

```
INSERT INTO game (numberOfTry, date, nameToGuess, username)
VALUES (?, ?, ?, ?);
```

#### OP 11 - Aggiornare il punteggio di un summoner dopo una vittoria.

```
UPDATE summoner
SET championsGuessed = championsGuessed + 1
WHERE username = ?;
```

#### OP 12 - Aggiornare il rank di un summoner.

```
UPDATE summoner
SET summoner.rank = (SELECT MAX(`rank`.idRank)
                    FROM `rank`
                    WHERE `rank`.rankGoal <= summoner.score)
WHERE summoner.username = ?;
```

#### OP 13 - Aggiornare una missione completata da un summoner.

```
UPDATE complete
SET isCompleted = 1
WHERE idMission = (SELECT idMission
                  FROM (SELECT idMission
                        FROM complete
                        WHERE isCompleted = 0
                        AND username = ?
                        ORDER BY idMission
                        LIMIT 1) AS subquery)
AND username = ?;
```

**OP 14 - Aggiornare il numero di partite vinte da un summoner.**

```
UPDATE summoner
SET score = score + ?
WHERE username = ?;
```

**OP 15 - Aggiungere i punti bonus di una missione completata al punteggio del summoner.**

```
UPDATE summoner
SET score = score + (SELECT points
                      FROM mission
                      WHERE idMission = (SELECT MAX(idMission)
                                         FROM complete
                                         WHERE isCompleted = 1)
AND username = ?);
```

**OP 16 - Controllare il limite delle partite giocate in un giorno da un summoner.**

```
SELECT COUNT(game.date) AS numberOfGames
FROM game
WHERE username = ?
AND game.date = ?
GROUP BY game.date
```

# Progettazione dell'Applicazione

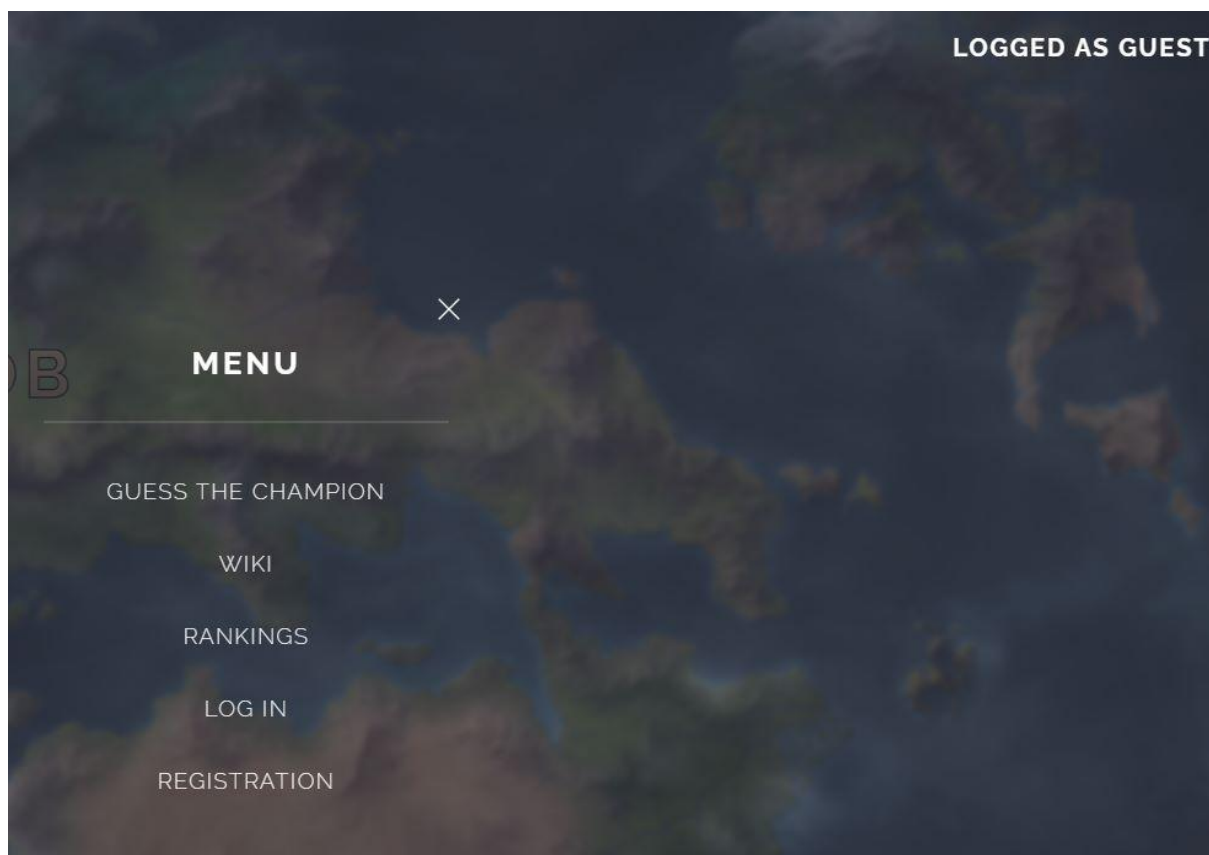
## Descrizione dell'architettura dell'applicazione realizzata

Per interfacciarsi al database è stato realizzato un sito web in linguaggio HTML, CSS e JAVASCRIPT per il front-end, PHP per la parte di back-end e soprattutto per l'interazione con il database, quest'ultimo risiede in locale e il DBMS usato è MySQL. Come appena accennato abbiamo deciso di dividere il front-end dal back-end: con il primo intendiamo la parte grafica del sito, tutto quello che l'utente vede e con la quale può interagire; con il secondo intendiamo tutta quella parte nascosta di applicazione che opera dietro le quinte della pagina web, si occupa della logica e gestione dei dati, riceve le richieste dal front-end in modo da modificare o recuperare i dati dal database.



*Figura 2.1 Schermata home utente guest.*

Questa schermata rappresenta la home del nostro sito, in alto a destra è possibile controllare se si è degli utenti guest oppure come vedremo di seguito se si ha eseguito l'accesso.



*Figura 2.2 Menù per un utente guest.*

In questa figura è mostrato il menù di un utente guest, in questo caso nella home, infatti sono presenti le opzioni di Log-In e di Registration.



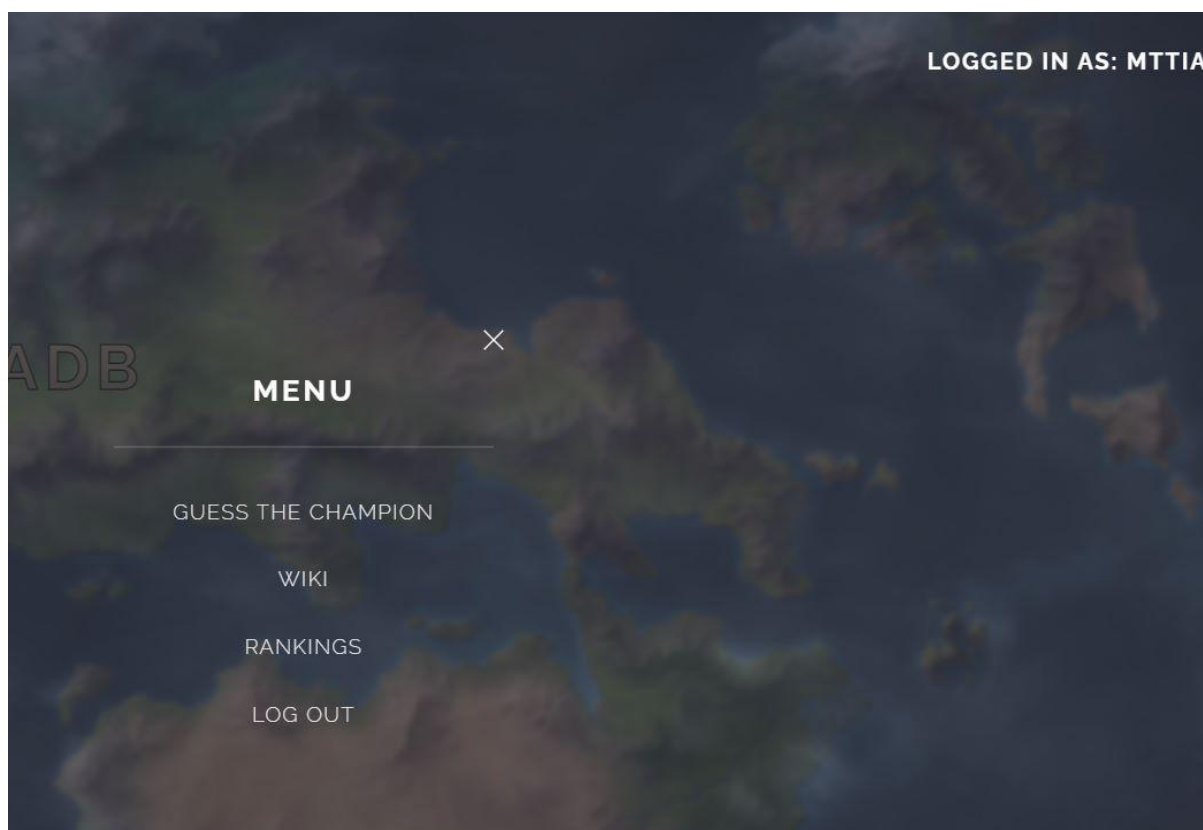
*Figura 2.3 Schermata di Log-In.*

In questa schermata l'utente ha la possibilità di accedere al sito con i propri dati, inserendo username e password negli appositi spazi, se dovessero rivelarsi corretti, verrà reindirizzato alla schermata home con l'accesso eseguito.



*Figura 2.4 Schermata home utente dopo l'accesso*

Questa schermata riprende la figura 2.1 dopo che un utente ha eseguito l'accesso correttamente.



*Figura 2.5 Menù per un utente dopo l'accesso.*

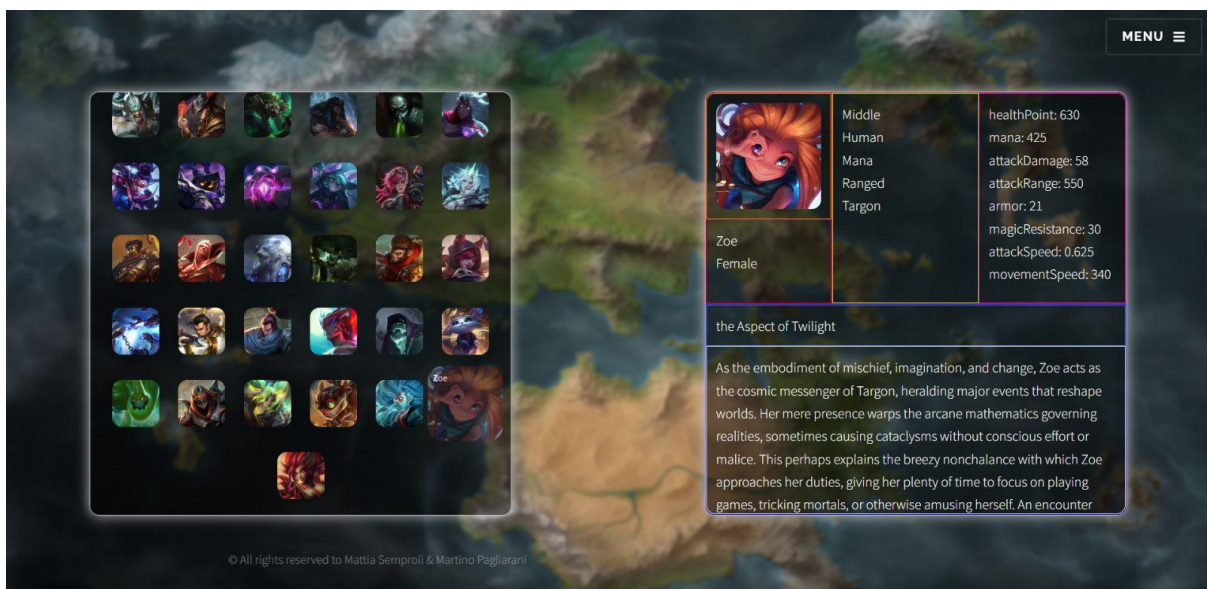
Come per la schermata home, questa figura riprende la figura 2.2, in questo caso avendo effettuato l'accesso, nel menù non sono presenti le opzioni di Log-In e Registration ma è disponibile quella per effettuare il Log-Out.





*Figura 2.6 Schermata raffigurante la wiki.*

In questa schermata viene mostrata la wiki che compone la parte informativa del sito, dove qualsiasi utente anche guest, all'interno della box a sinistra, ha la possibilità di scorrere tra le immagini dei champion e selezionare con il mouse quello ricercato.



*Figura 2.7 Schermata raffigurante la wiki con un champion selezionato.*

Questa figura riprende quella precedente dopo che l'utente ha selezionato uno specifico champion, che sarà mostrato nella box di destra. Qui l'utente ha la possibilità di leggere le informazioni richieste, come nome, statistiche e la biografia.



*Figura 2.8 Schermata di Registration.*

In questa schermata un nuovo utente ha la possibilità di creare il proprio account utente, inserendo uno username e una password, quest'ultima necessita di una conferma. In caso lo username inserito non dovesse già esistere e la password viene confermata correttamente, l'utente può procedere con la creazione del proprio profilo summoner.



*Figura 2.9 Schermata di creazione del profilo summoner.*

Questa schermata appare subito dopo la registrazione (figura 2.8), dove il nuovo utente deve creare il proprio profilo summoner. Sarà richiesto uno username e il team di cui si vuole fare parte.





*Figura 2.10 Schermata di gioco.*

In questa schermata l'utente che ha eseguito l'accesso ha la possibilità di giocare ad una partita del minigioco "Guess the Champion". All'interno del campo al centro è possibile inserire il nome di un champion per iniziare ad indovinare come vedremo successivamente.



*Figura 2.11 Schermata di gioco dopo alcuni tentativi.*

Dopo che vengono effettuati dei tentativi da parte dell'utente, qualora alcune caratteristiche del champion inserito dovessero corrispondere con alcune di quello da indovinare, saranno mostrate in verde, questo aiuterà l'utente ad avvicinarsi sempre più alla vittoria.



*Figura 2.12 Schermata di vittoria.*

In questa schermata viene mostrato all'utente un messaggio che gli indica di aver indovinato il champion corretto portando alla vittoria della partita.



*Figura 2.13 Schermata di vittoria dopo il termine delle partite a punti del giorno.*

Questa schermata riprende la figura 2.12 quando l'utente si ritrova nella situazione in cui ha raggiunto il limite di partite giornaliere che assegnano punti. Siccome è possibile guadagnarne solo con le prime cinque partite giornaliere, l'utente potrà continuare a giocare ma senza ottenere alcun punto.

Name	Score	Guesses	Team	Rank
mattia	44	10	T1	Iron
proli	5	1	C9	Unranked
ThePaglia02	3	1	T1	Unranked
Il Re	0	0	EDG	Unranked

Name	Acronym	Score
SK Telecom T1	T1	47
Clown 9	C9	5
G2 Esports	G2	0
Edward Gaming	EDG	0

© All rights reserved to Mattia Sempoli & Martino Pagliarani

*Figura 2.14 Schermata raffigurante le classifiche.*

In questa schermata un qualsiasi utente, può controllare sia la classifica dei summoner, sia quella dei team.