

# JGuesser

Specifica dei requisiti e design del sistema  
(UML use-case, tabelle strutturate, diagramma di contesto  
e diagramma delle componenti)



**UNIVERSITÀ  
DI TRENTO**

Dipartimento di Ingegneria e Scienza dell'Informazione

Gruppo: T24

Membri: Lorenzo D'Ambrosio, Andrea Goldoni, Marco Murru

## Indice

<b>1</b>	<b>Requisiti Funzionali</b>	<b>4</b>
1.1	Compendio degli alfabeti . . . . .	4
1.2	Registrazione . . . . .	5
1.3	Login con credenziali . . . . .	6
1.4	Login con google . . . . .	7
1.5	Logout . . . . .	8
1.6	Visualizzazione dati personali . . . . .	8
1.7	Modifica e-mail . . . . .	9
1.8	Modifica password . . . . .	10
1.9	Recupero nome utente e password . . . . .	11
1.10	Profilo Premium . . . . .	12
1.11	Quiz tipo 1 . . . . .	13
1.12	Quiz tipo 2 . . . . .	14
1.13	Quiz tipo 3 . . . . .	15
1.14	Quiz tipo 4 . . . . .	16
1.15	Single-player . . . . .	17
1.16	Multiplayer . . . . .	18
1.17	Risultato score . . . . .	19
1.18	Termina sessione di gioco . . . . .	20
1.19	Visualizzazione statistiche . . . . .	20
1.20	Visualizzazione classifica . . . . .	21
1.21	Ricerca nella classifica . . . . .	22
<b>2</b>	<b>Requisiti Non Funzionali</b>	<b>23</b>
2.1	Privacy . . . . .	23
2.2	Sicurezza . . . . .	23
2.3	Efficienza . . . . .	24
2.4	Portabilità . . . . .	24
2.5	Scalabilità . . . . .	25
2.6	Usabilità . . . . .	25
<b>3</b>	<b>Analisi Del Contesto</b>	<b>26</b>
3.1	Diagramma di contesto . . . . .	26
3.2	Descrizione interazioni attori e sistemi esterni . . . . .	27
3.2.1	Guest . . . . .	27
3.2.2	Logged User . . . . .	28
3.2.3	Premium User . . . . .	29
3.2.4	Google Identity Service . . . . .	29
3.2.5	Paypal . . . . .	29
3.2.6	reCaptcha . . . . .	30
3.2.7	Nodemailer . . . . .	30
3.2.8	Nyckel . . . . .	30
3.2.9	MongoDB . . . . .	30

<b>4</b>	<b>Analisi Dei Componenti</b>	<b>32</b>
4.1	Definizione dei componenti . . . . .	32
4.1.1	Pagina registrazione . . . . .	32
4.1.2	Gestore registrazione . . . . .	33
4.1.3	Validatore reCAPTCHA . . . . .	34
4.1.4	Interfaccia Nodemailer . . . . .	34
4.1.5	Pagina login . . . . .	35
4.1.6	Autenticazione . . . . .	35
4.1.7	Autenticazione con Google . . . . .	36
4.1.8	Pagina dati personali . . . . .	36
4.1.9	Pagina recupero username e password . . . . .	37
4.1.10	Gestore credenziali . . . . .	38
4.1.11	Pagina pagamento . . . . .	39
4.1.12	Gestore pagamenti . . . . .	39
4.1.13	Gestore upgrade profilo . . . . .	40
4.1.14	Pagina statistiche . . . . .	40
4.1.15	Pagina classifica . . . . .	40
4.1.16	Gestore dati utente . . . . .	41
4.1.17	Pagina single-player . . . . .	42
4.1.18	Quiz maker . . . . .	42
4.1.19	Pagina quiz . . . . .	43
4.1.20	Sistema interpretazione disegno . . . . .	44
4.1.21	Pagina multi-player . . . . .	45
4.1.22	Sistema di ricerca giocatori . . . . .	45
4.1.23	Gestore invio inviti di sfida . . . . .	46
4.1.24	Notifica di sfida . . . . .	46
4.1.25	Gestore giocatori disponibili . . . . .	47
4.1.26	Gestore sessione di gioco . . . . .	47
4.1.27	Pagina info . . . . .	48
4.2	Diagramma dei componenti . . . . .	49

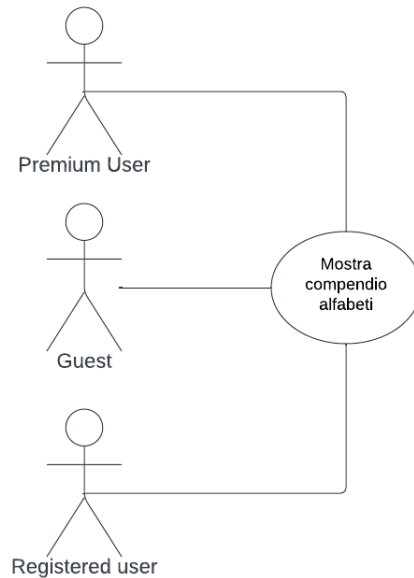
## Scopo del documento

Nel precedente documento sono stati descritti gli obbiettivi e i requisiti del sistema, utilizzando il solo linguaggio naturale. In questo documento è riportata la specifica dei requisiti dell'applicazione. Si procede quindi con una descrizione dei requisiti più specifica, che necessita di essere più formale. Per questo motivo ora i requisiti funzionali saranno descritti utilizzando il linguaggio UML use-case, mentre i requisiti non funzionali saranno delineati utilizzando delle tabelle strutturate. Inoltre, tenendo conto di tali requisiti, in questa sezione, viene presentato il design del sistema con l'utilizzo di diagrammi di contesto e dei componenti.

# 1 Requisiti Funzionali

In questo capitolo vengono ripresi i requisiti funzionali precedentemente descritti in linguaggio naturale e vengono riscritti in linguaggio formale UML use-case diagram. Si va inoltre più nello specifico, mostrando quali attori sono coinvolti in ciascuna funzione e illustrando il ruolo che svolgono all'interno di essa.

## 1.1 Compendio degli alfabeti



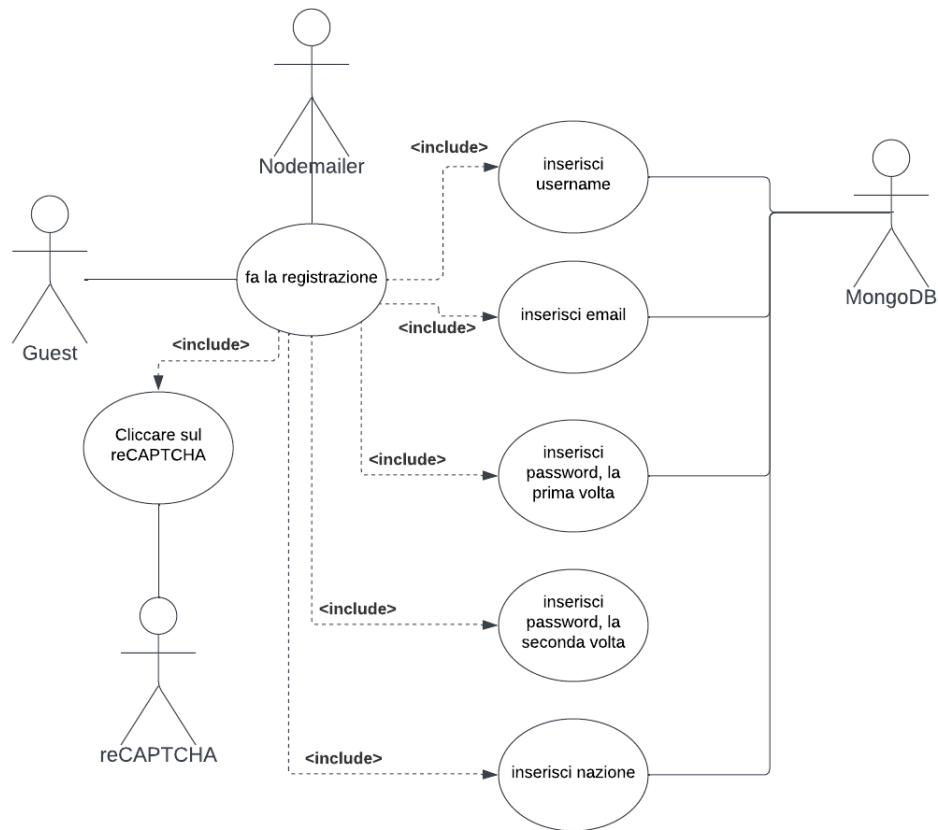
**Titolo:** mostrare compendio alfabeti

**Riassunto:** questo use-case spiega come i diversi tipi di utenti possono accedere al compendio degli alfabeti.

**Descrizione:**

1. L'utente, che sia autenticato o non autenticato che si trova nella pagina principale dell'applicazione, clicca sul pulsante 'info'.
2. Il sistema porterà l'utente in una nuova pagina in cui gli sarà mostrata una descrizione dei tre alfabeti utilizzati per la nostra applicazione. Oltre a ciò gli saranno resi disponibili due tabelle e una lista. Le tabelle conterranno tutti i simboli degli alfabeti Hiragana e Katakana, mentre la lista conterrà tutti i simboli dell'alfabeto Kanji che sono stati utilizzati, specificatamente all'interno dell'applicazione.

## 1.2 Registrazione



**Titolo:** registrazione

**Riassunto:** questo use-case spiega come un utente anonimo (guest), può registrarsi.

**Descrizione:**

1. L'utente che si trova nella pagina di registrazione, inserisce in degli appositi campi le seguenti informazioni: username, e-mail, password, conferma password e nazione.
2. L'utente clicca sul reCAPTCHA e poi sul pulsante 'create account', per confermare che si tratta di un essere umano e per confermare i dati precedentemente inseriti. [ Eccezione 1 ]
3. Il sistema attraverso delle chiamate al database MongoDB, memorizza username, e-mail, password e nazione, come un nuovo utente. [ Eccezione 2 ]
4. L'applicazione invia all'utente un'e-mail di conferma dell'avvenuta registrazione. [ Eccezione 3 ]
5. L'utente è finalmente stato registrato e ritorna alla pagina principale dell'applicazione, con login già effettuato.

### 1.3 Login con credenziali

---

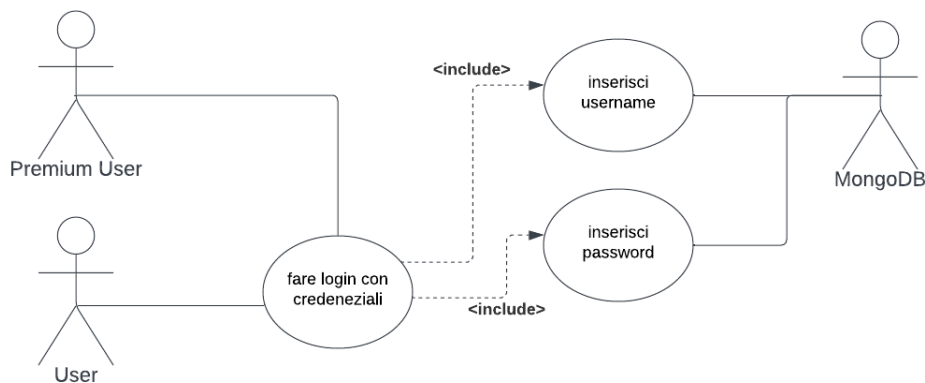
**Eccezioni:**

[ Eccezione 1 ] nel caso in cui l'utente si è dimenticato di inserire qualche campo, di cliccare sul reCAPTCHA, i campi password e conferma password non coincidono, allora un messaggio di errore viene mostrato a video.

[ Eccezione 2 ] nel caso in cui esista già un utente con quello specifico username o e-mail, non si procede e sarà visualizzato un messaggio di errore.

[ Eccezione 3 ] l'e-mail inserita non esiste e allora viene stampato a video un messaggio di errore.

### 1.3 Login con credenziali



**Titolo:** login con credenziali

**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore, può fare il login all'interno dell'applicazione usando le credenziali inserite nella fase di registrazione.

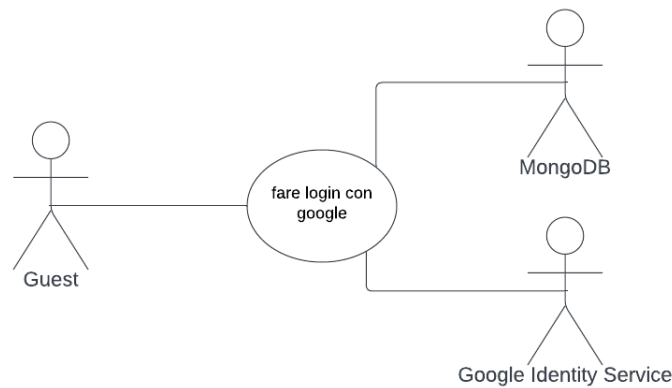
**Descrizione:**

1. L'utente che si trova nella pagina di login, inserisce l'username e la password relativi al proprio account e clicca sul pulsante 'login'.
2. L'applicazione interagisce con il sistema esterno MongoDB, per verificare che le credenziali inserite siano valide. [ Eccezione 1 ]
3. Viene eseguito il login e viene fatto ritornare l'utente alla pagina principale dell'applicazione.

**Eccezioni:**

[ Eccezione 1 ] se l'username inserito non esiste oppure la password non è corretta, non si procede e sarà visualizzato un messaggio all'utente con scritto username o password errata.

## 1.4 Login con google



**Titolo:** login con google

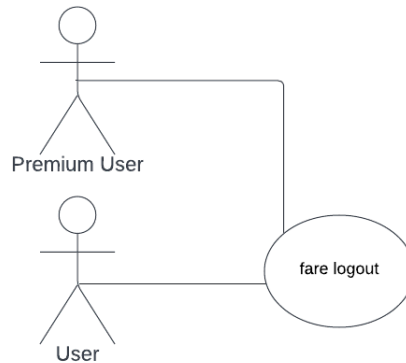
**Riassunto:** questo use-case spiega come un utente non registrato, può fare il login all'interno dell'applicazione usando il proprio account google.

**Descrizione:**

1. L'utente che si trova nella pagina di login, clicca sul pulsante che gli permetterà di effettuare il login con google.
2. L'applicazione grazie alle API di google interagirà con il Google Identity Service e si aprirà una finestra che permetterà all'utente di effettuare il login tramite credenziali (e-mail e password) di google.
3. Dopo che l'utente avrà effettuato l'autenticazione con le proprie credenziali google, il sistema interagirà con il database MongoDB per creare un entry all'interno database, nel caso in cui non fosse già presente, per ricordarsi che l'utente si è registrato/loggato con account google.
4. Viene fatto ritornare l'utente alla pagina principale dell'applicazione.



## 1.5 Logout



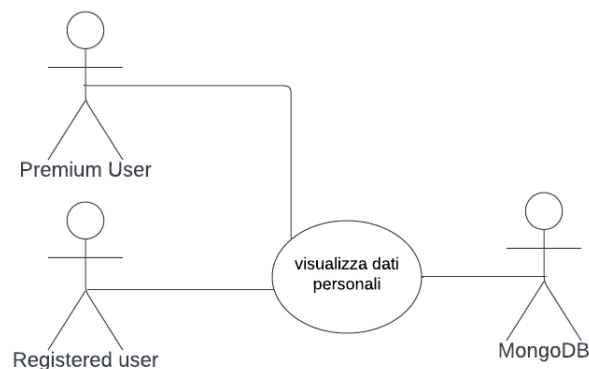
**Titolo:** logout dall'applicazione

**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore può uscire dal proprio account.

**Descrizione:**

1. L'utente che si trova nella pagina di principale dell'applicazione, clicca sul icona del proprio profilo e seleziona la voce 'logout'.
2. Il sistema fa uscire l'utente dal proprio profilo e lo trasferisce nella pagina principale come utente guest.

## 1.6 Visualizzazione dati personali



**Titolo:** visualizzazione dati personali

**Riassunto:** questo use-case mostra come un utente registrato o di livello superiore può visualizzare i propri dati personali inseriti in fase di registrazione.

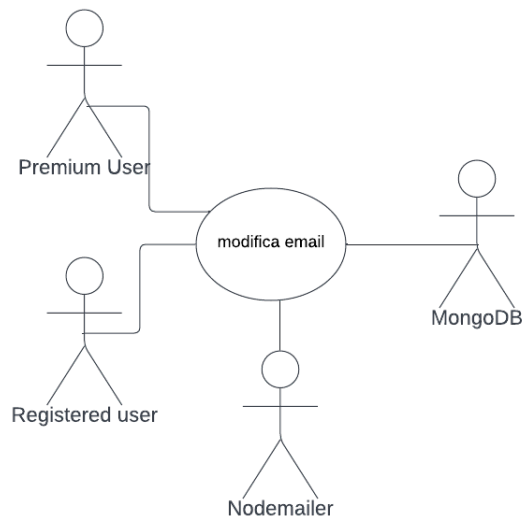
**Descrizione:**

1. L'utente che si trova nella pagina principale dell'applicazione, clicca sul icona del proprio profilo e seleziona la voce 'my account'.

1.7 Modifica e-mail

---

2. L'applicazione interagirà con il sistema esterno MongoDB per recuperare i dati personali dell'utente.
3. Il sistema porterà l'utente nella pagina della propria area personale, dove potrà visualizzare username, e-mail, password e nazione.

**1.7 Modifica e-mail****Titolo:** modifica e-mail**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore può modificare la propria e-mail**Descrizione:**

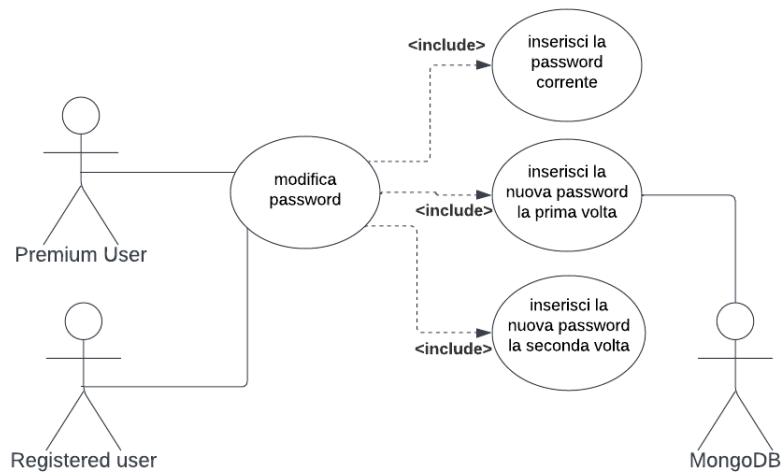
1. L'utente che si trova nella pagina in cui può visualizzare le proprie informazioni personali, decide di modificare la propria e-mail, che è presente in un apposito 'box'. Dopo aver modificato l'e-mail l'utente clicca sul pulsante 'confirm' per confermare la nuova e-mail.
2. L'applicazione interagisce con il sistema esterno Nodemailer per verificare che l'e-mail inserita sia un e-mail effettivamente esistente e quindi valida. [ Eccezione 1 ]
3. Il sistema interagisce con il database esterno MongoDB per sostituire l'e-mail precedente dell'utente con la nuova e-mail. [ Eccezione 2 ]

**Eccezioni:**

[ Eccezione 1 ] l'e-mail è inesistente e quindi viene stampato un messaggio di errore all'utente.

[ Eccezione 2 ] l'e-mail che si è cercato di inserire è già registrata per un altro account, allora viene stampato a video un messaggio di errore.

## 1.8 Modifica password



**Titolo:** modifica password

**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore può modificare la propria password

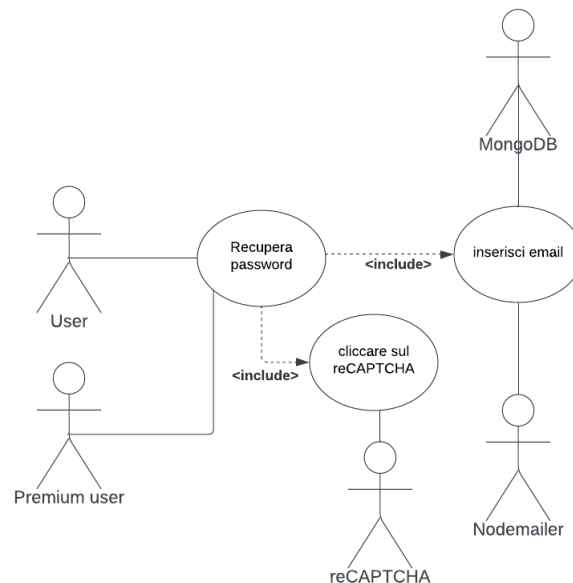
**Descrizione:**

1. L'utente che si trova nella pagina in cui può visualizzare le proprie informazioni personali, decide di modificare la propria password. La modifica della password avviene inserendo in un form: la password precedente, la nuova password una volta e la nuova password una seconda volta. Infine il tutto viene confermato attraverso il pulsante 'confirm'. [ Eccezione 1 ]
2. L'applicazione interagisce con il database esterno MongoDB, il quale modifica la password dell'utente.

**Eccezioni:**

[ Eccezione 1 ] alcuni campi sono mancanti oppure i due campi dove viene inserita la nuova password non coincidono e allora viene stampato a video un messaggio di errore.

## 1.9 Recupero nome utente e password



**Titolo:** recupero username e password

**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore può recuperare la propria password e username, nel caso se li fosse dimenticati

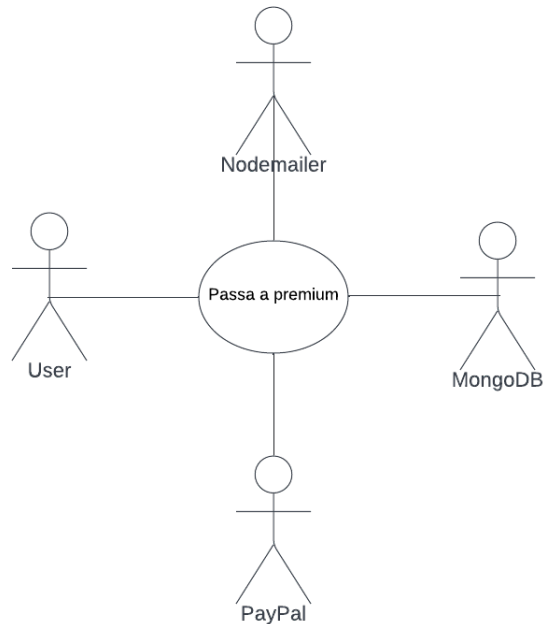
**Descrizione:**

1. L'utente che si trova nella pagina di login, clicca sulla scritta 'Forgot password?'. L'applicazione porta l'utente in una pagina dove dovrà inserire la propria e-mail.
2. Inserita l'e-mail l'utente clicca sul reCAPTCHA e sul pulsante 'Confirm reset password'.
3. Il sistema genererà una password temporanea, si interfaccia con il database MongoDB, che modificherà la password dell'utente a cui è associata l'e-mail precedentemente inserita. [ Eccezione 1 ]
4. L'applicazione si interfacerà con il sistema esterno Nodemailer per inviare l'e-mail che conterrà l'username dell'utente con la nuova password.

**Eccezioni:**

[ Eccezione 1 ] nel caso in cui non esista nessun utente con associata quella specifica e-mail, viene mostrato un messaggio a video che dice all'utente che l'e-mail inserita non è valida.

### 1.10 Profilo Premium



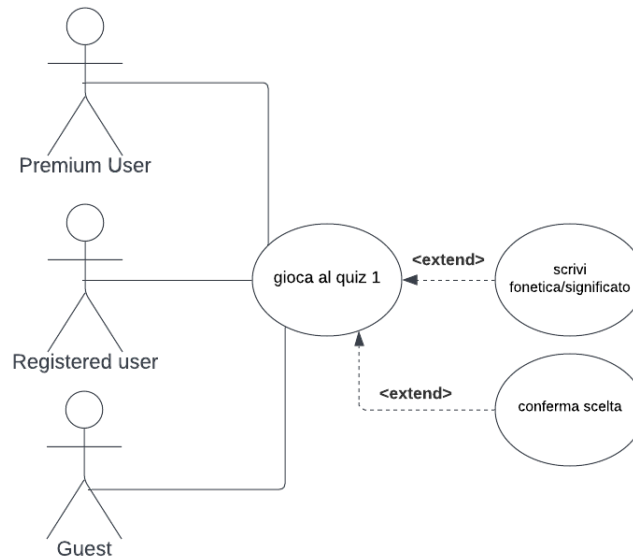
**Titolo:** acquisto versione premium

**Riassunto:** questo use-case spiega come un utente registrato può upgradare il suo profilo a premium.

**Descrizione:**

1. L'utente registrato che si trova nella pagina principale dell'applicazione, clicca sul icona del proprio profilo e seleziona la voce 'upgrade profile'.
2. L'applicazione interagisce con il servizio di pagamento Paypal, con il quale verrà effettuata la transazione di 2.99€, all'indirizzo e-mail lorenzo.dambro@gmail.com.
3. L'applicazione interagisce con il database MongoDB e aggiorna i privilegi dell'utente, registrandolo come utente premium.
4. Il sistema invia un e-mail all'utente che lo ringrazia dell'acquisto e che lo avverte che il suo profilo è stato upgradato.

### 1.11 Quiz tipo 1



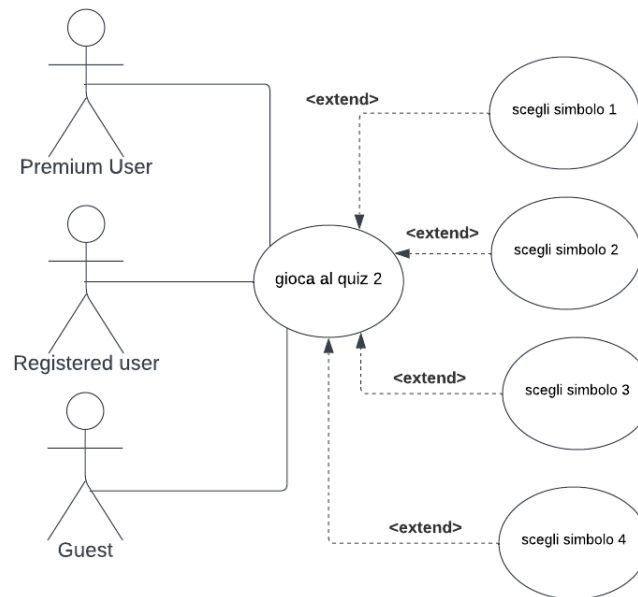
**Titolo:** Quiz di tipo 1

**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può giocare al quiz di tipo 1.

**Descrizione:**

1. L'utente che sta giocando al quiz di tipo 1 si trova davanti al simbolo di uno degli alfabeti scelti dall'utente, ad una casella in cui inserire la fonetica/significato del simbolo e ad un pulsante di conferma, per verificare la correttezza della propria risposta.
2. L'utente risponde con la fonetica/significato corrispondente, clicca sul pulsante di conferma e il sistema procederà in automatica con la correzione.
3. L'applicazione rimane in pausa due secondi, per mostrare il risultato della correzione all'utente e poi il sistema procederà con il quiz successivo, oppure tornando alla schermata principale dell'applicazione se non ci sono altri quiz.

## 1.12 Quiz tipo 2



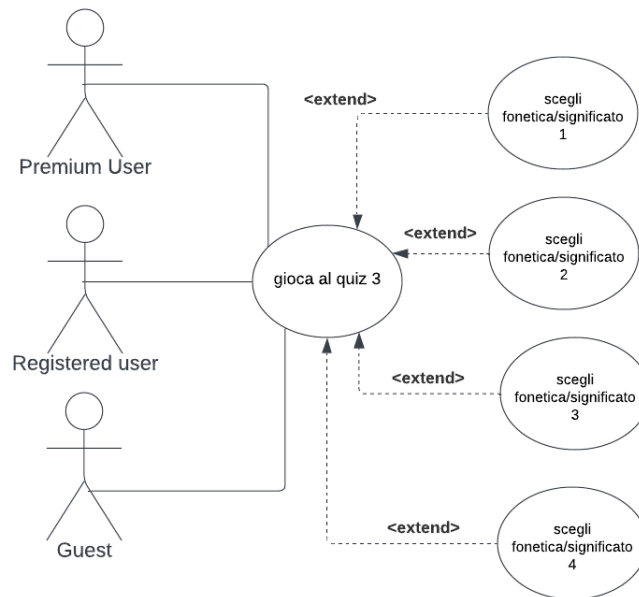
**Titolo:** Quiz di tipo 2

**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può giocare al quiz di tipo 2.

**Descrizione:**

1. L'utente che sta giocando al quiz di tipo 2 si trova davanti alla fonetica/significato di un simbolo di uno degli alfabeti scelti dall'utente e a 4 simboli casuali.
2. L'utente clicca sul simbolo che è associato alla fonetica/significato corretto.
3. L'applicazione rimane in pausa due secondi, per mostrare il risultato della correzione all'utente e poi il sistema procederà con il quiz successivo, oppure tornando alla schermata principale dell'applicazione se non ci sono altri quiz.

### 1.13 Quiz tipo 3



**Titolo:** Quiz di tipo 3

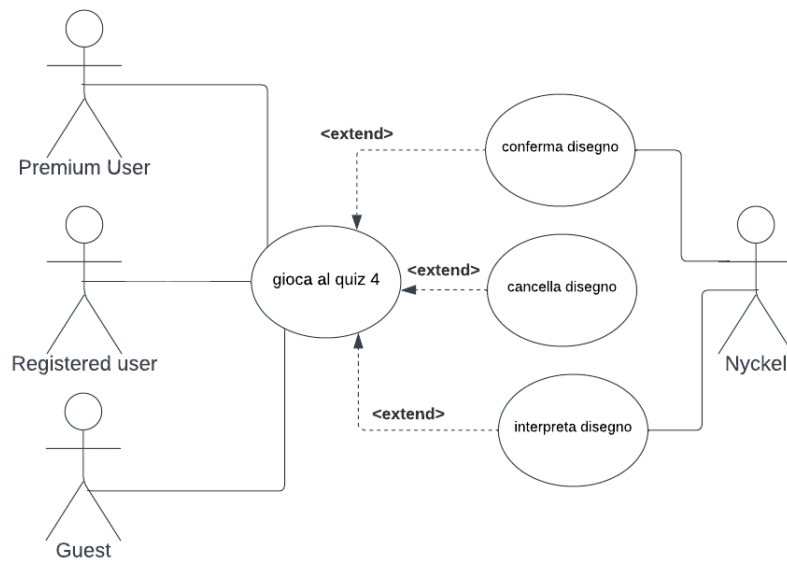
**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può giocare al quiz di tipo 3.

**Descrizione:**

1. L'utente che sta giocando al quiz di tipo 3 si trova davanti ad un simbolo di uno degli alfabeti scelti dall'utente e a 4 fonetiche/significati casuali.
2. L'utente deve cliccare sulla fonetica/significato che si associa correttamente al simbolo.
3. L'applicazione rimane in pausa due secondi, per mostrare il risultato della correzione all'utente e poi il sistema procederà con il quiz successivo, oppure tornando alla schermata principale dell'applicazione se non ci sono altri quiz.



## 1.14 Quiz tipo 4



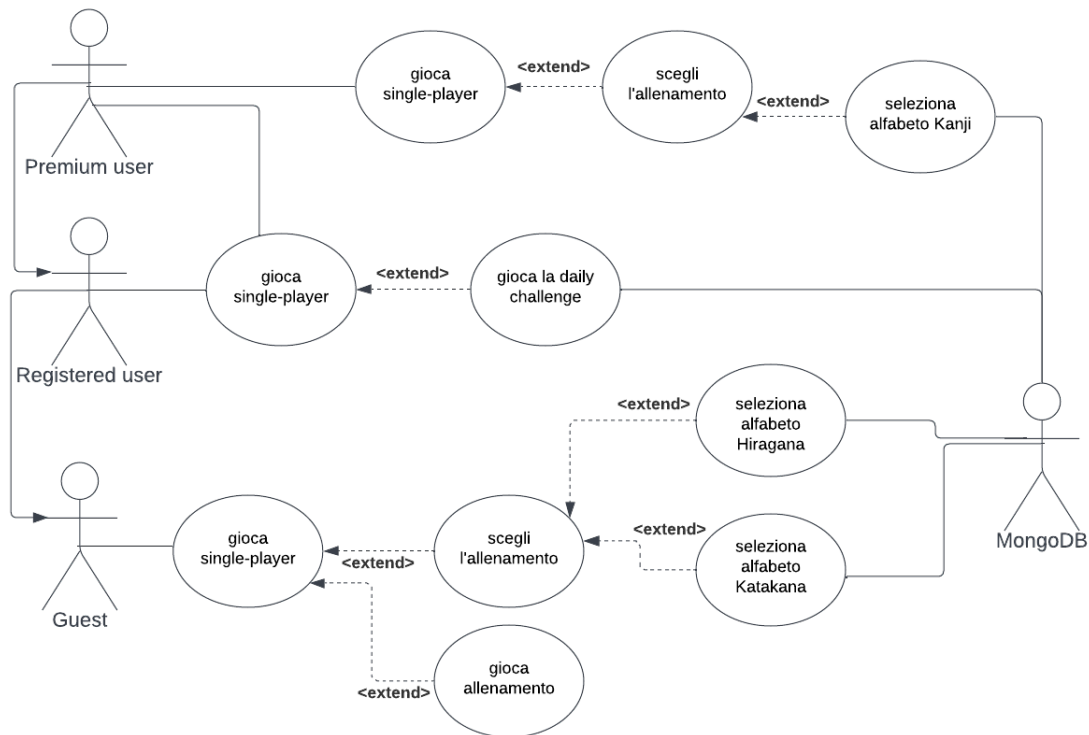
**Titolo:** Quiz di tipo 4

**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può giocare al quiz di tipo 4.

**Descrizione:**

1. L'utente che sta giocando al quiz di tipo 4 si trova davanti alla fonetica/significato (per alfabeti Hirigana e Katakana) o solo al significato (per alfabeti Katakana), ad uno spazio vuoto in cui può disegnare il simbolo e a tre pulsanti 'confirm', 'interpret' e 'cancel'.
2. L'utente disegna il simbolo a cui è associata la fonetica/significato corrispondente, clicca sul pulsante 'interpret' per verificare che il simbolo sia stato interpretato correttamente dal sistema e infine clicca sul pulsante 'confirm' per confermare quanto disegnato. Quando l'utente clicca sul pulsante 'interpret' o 'confirm' il sistema esterno Nyckel sarà utilizzato per classificare quanto disegnato e quindi per verificare il disegno. Nel caso l'utente volesse ridisegnare il simbolo perchè si rende conto che ha sbagliato oppure il sistema l'ha interpretato in maniera errata allora, può farlo cliccando sul pulsante 'cancel', che cancella quanto precedentemente disegnato.
3. L'applicazione rimane in pausa due secondi, per mostrare il risultato della correzione all'utente e poi il sistema procederà con il quiz successivo, oppure tornando alla schermata principale dell'applicazione se non ci sono altri quiz.

## 1.15 Single-player



**Titolo:** single player

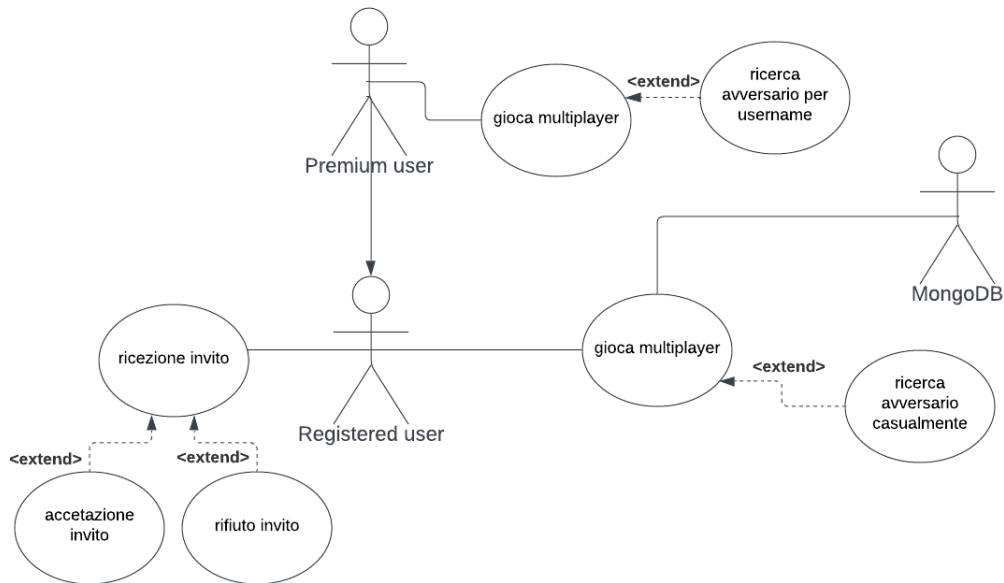
**Riassunto:** questo gruppo di use-case spiega come come l'utente guest o di livello superiore può giocare in modalità single-player.

**Descrizione:**

1. L'utente che si trova nella schermata principale dell'applicazione clicca sul pulsante 'Single-player'.
2. Si apre una finestra dove l'utente avrà due scelte, effettuare un allenamento con gli alfabeti selezionati cliccando sul pulsante 'Training' oppure provare la sfida giornaliera cliccando sul pulsante 'Daily challenge' (se non l'ha già effettuata per quello specifico giorno). [ Eccezione 1 ]
3. Effettuate le scelte l'applicazione interagirà con il sistema esterno MongoDB, per ricevere la lista dei quiz da proporre all'utente.
4. A questo punto il gioco inizierà e il sistema porterà l'utente nella pagina dove sarà visualizzato il primo quiz.

[ Eccezione 1 ] se l'utente che ha deciso di giocare alla modalità 'training' non ha selezionato nessun alfabeto, allora viene mostrato a video un messaggio di errore, che fa notare all'utente questo problema.

## 1.16 Multiplayer



**Titolo:** multiplayer

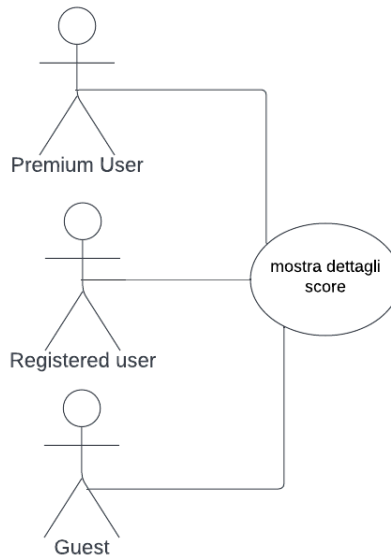
**Riassunto:** questo gruppo di use-case spiega come come l'utente registrato o di livello superiore può giocare in modalità multiplayer.

**Descrizione:**

1. L'utente che si trova nella schermata principale dell'applicazione clicca sul pulsante 'Multiplayer'.
2. Si apre una finestra dove l'utente avrà due scelte, effettuare una ricerca dell'avversario in maniera casuale oppure sfidare un avversario specifico inserendo il suo username.
3. Effettuata la scelta l'applicazione interagirà con il sistema esterno MongoDB, per ricevere la lista dei quiz da proporre ad entrambi gli utenti. [ Eccezione 1 ]
4. Se un utente ha effettuato la ricerca di un avversario per username. L'utente ricercato riceverà una notifica di invito che potrà accettare o rifiutare.

[ Eccezione 1 ] se si è ricercato un avversario per username, ma il giocatore ricercato non esiste viene mostrato a video un messaggio di errore.

### 1.17 Risultato score



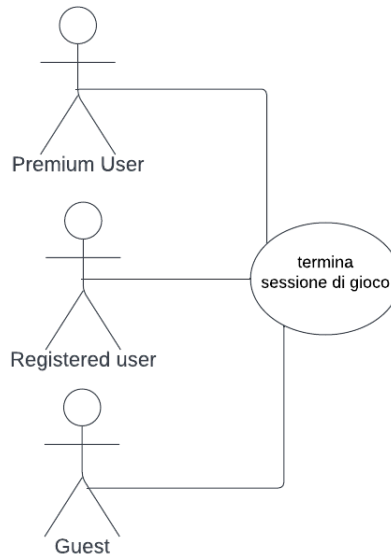
**Titolo:** mostra dettagli score

**Riassunto:** questo use-case spiega come l'utente può visualizzare una serie di informazioni aggiuntive sui quiz fatti, durante una specifica sessione di gioco.

**Descrizione:**

1. L'utente che attualmente si trova nella schermata di uno dei 4 tipi di quiz, clicca sul pulsante 'score' in alto a destra.
2. Si apre una finestra, in cui compaiono le seguenti informazioni:
  - Punteggio totale;
  - Numero di vittorie, divise tra sfide giornaliere e ranked;
  - Numero di sconfitte, divise tra sfide giornaliere e ranked;
  - Rapporto vittorie e sconfitte, divise tra sfide giornaliere e ranked.

### 1.18 Termina sessione di gioco



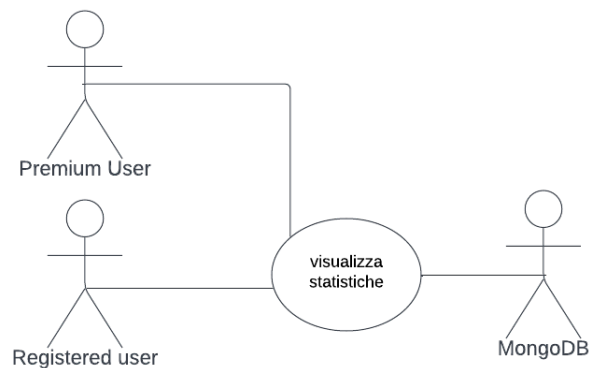
**Titolo:** termina sessione di gioco

**Riassunto:** questo use-case spiega come l'utente può terminare la sessione di gioco attuale.

**Descrizione:**

1. L'utente che attualmente si trova nella schermata di uno dei 4 tipi di quiz, clicca sul pulsante 'home' in alto a sinistra.
2. Il sistema riporta l'utente nella pagina principale dell'applicazione.

### 1.19 Visualizzazione statistiche

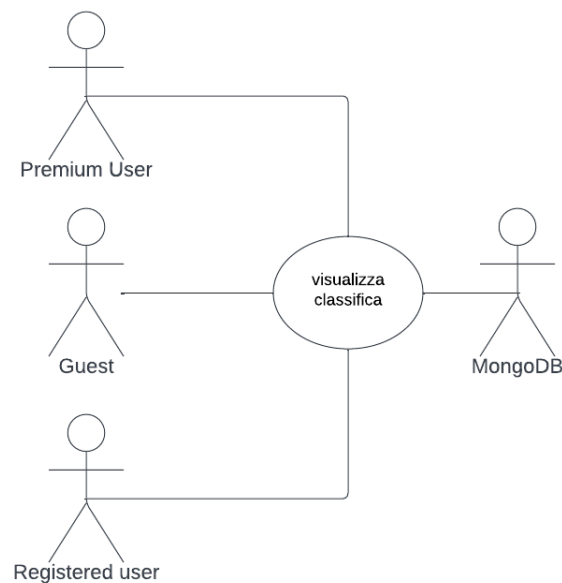


**Titolo:** visualizza statistiche

**Riassunto:** questo use-case spiega come un utente registrato o di livello superiore può visualizzare le proprie statistiche personali.

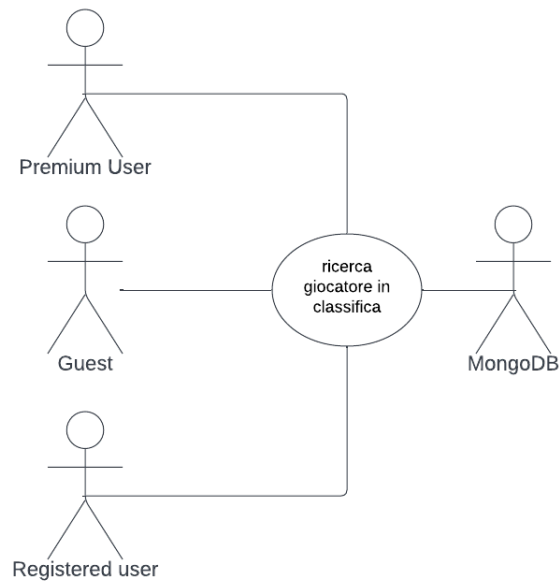
**Descrizione:**

1. L'utente che si trova nella pagina principale dell'applicazione, clicca sul icona del proprio profilo e seleziona la voce 'my statistics'.
2. L'applicazione interagirà con il database esterno MongoDB, per recuperare tutte le statistiche di quello specifico utente e porterà l'utente in una pagina dove potrà visualizzarle. Le statistiche includono: numero partite giocate, punteggio totale, numero di vittorie (giornaliere/ranked), numero di sconfitte (giornaliere/ranked) e rapporto tra vittorie e sconfitte (giornaliere/ranked).

**1.20 Visualizzazione classifica****Titolo:** visualizzazione classifica**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può visualizzare la classifica dei giocatori.**Descrizione:**

1. L'utente che si trova nella pagina principale dell'applicazione, clicca sul pulsante "World Ranking".
2. L'applicazione interagirà con il database esterno MongoDB, per recuperare le informazioni dei primi 100 giocatori e trasferirà l'utente in una nuova pagina, dove gli verrà mostrata la classifica. Le informazioni mostrate saranno: posizione in classifica, nazionalità, username, punteggio totale e partite giocate.

## 1.21 Ricerca nella classifica



**Titolo:** ricerca nella classifica

**Riassunto:** questo use-case spiega come un utente di qualsiasi livello può effettuare la ricerca di un altro giocatore all'interno della classifica

**Descrizione:**

1. L'utente che si trova nella schermata "World Ranking" clicca sulla barra di ricerca e digita il nome del giocatore da ricercare.
2. L'applicazione interagirà in real time con il database esterno MongoDB per filtrare tutte le statistiche del nome giocatore che l'utente sta digitando.

## 2 Requisiti Non Funzionali

In questo capitolo sono stati ripresi i requisiti non funzionali e sono state costruite delle tabelle strutturate, nella quale sono state inserite le proprietà che deve avere il sistema, una descrizione di queste proprietà e un valore di misura per capire quando il requisito non funzionale è stato soddisfatto oppure no.

### 2.1 Privacy

Proprietà	Descrizione	Misura
Regolamento per la protezione dei dati (GDPR)	Il Regolamento Generale sulla Protezione dei Dati in sigla RGPD (o GDPR in inglese General Data Protection Regulation), ufficialmente regolamento (UE) n. 2016/679, è un regolamento dell'Unione europea in materia di trattamento dei dati personali e di privacy.	I dati personali dell'utente (email, password) non vengono condivisi con terze parti.

### 2.2 Sicurezza

Proprietà	Descrizione	Misura
Connessione sicura	Lo scambio di dati tra client e server deve avvenire in maniera criptata.	La trasmissione dati avviene mediante il protocollo https.
Caratteristiche password	Le password utilizzate all'interno del sistema devono essere delle strong password.	Le password utilizzate per gli account devono rispettare le seguenti caratteristiche: minimo otto caratteri, minimo due numeri, minimo un carattere speciale ed inoltre devono esservi presenti sia lettere minuscole che maiuscole.
Memorizzazione password tramite cifratura	Le password verranno memorizzate all'interno del database in maniera cifrata.	Le password vengono cifrate attraverso la funzione hash crittografica SHA256.



## 2.3 Efficienza

## 2.3 Efficienza

Proprietà	Descrizione	Misura
Tempo di risposta del sistema per la somministrazione dei quiz	Il tempo massimo entro il quale il sistema deve somministrare il quiz successivo, dopo che si ha già risposto al quiz precedente.	Il lasso di tempo non è superiore a tre secondi.
Tempo allestimento partita online	Il lasso di tempo che intercorre tra il matching con un avversario e la ricezione del primo quiz da parte dei due utenti online.	Il lasso di tempo non è superiore a dieci secondi.
Tempo massimo di caricamento della classifica	Il tempo massimo entro il quale, nel momento in cui un utente accede alla classifica globale, vengono visualizzati i primi cento giocatori.	Il lasso di tempo non è superiore a tre secondi.
Tempo massimo di ricerca di un giocatore nella classifica	Il tempo massimo entro il quale, nel momento in cui viene effettuata la ricerca di un giocatore all'interno della classifica, viene visualizzato (se presente) l'utente cercato.	Il lasso di tempo non è superiore a due secondi.

## 2.4 Portabilità

Proprietà	Descrizione	Misura
Compatibilità con browser differenti	L'applicazione dovrà poter essere utilizzata e quindi funzionare correttamente sul browser Google Chrome, Mozilla Firefox e Opera.	Le versioni dei browser dalle quale l'applicazione li supporta sono, dalla 107.0.5304.87 per Google Chrome, dalla 106.0.2 per Mozilla Firefox e dalla 92.0.4561.21 per Opera.
Visualizzazione dell'interfaccia su dispositivi differenti	L'interfaccia deve essere visualizzata in maniera adeguata sia su desktop, ma anche su dispositivi mobili.	Il range di pixel entro il quale l'applicazione dovrà essere visualizzata in maniera adeguata deve essere 412x914 px - 1920x1080 px

## 2.5 Scalabilità

---

### 2.5 Scalabilità

Proprietà	Descrizione	Misura
Elaborazione con un numero crescente di utenti	Capacità del sistema di gestire un numero crescente di utenti in simultanea.	Garantita fino a 10.000 utenti in simultanea.
Memorizzazione dei dati con un numero crescente di utenti	Capacità del sistema di gestire i dati generati da un alto numero di utenti.	Garantita fino a 10.000 utenti.

### 2.6 Usabilità

Proprietà	Descrizione	Misura
Usabilità	L'interfaccia dell'applicazione dovrà essere facile ed intuitiva.	Preso un utente medio che non conosce l'applicazione, questo è in grado di comprenderne le funzionalità principali in 5-10 minuti.

### 3 Analisi Del Contesto

Nei capitoli precedenti del documento ci siamo concentrati molto sui requisiti funzionali e non funzionali e di come si possano specificare utilizzando un linguaggio UML-based per i primi e delle tabelle strutturate per i secondi.

In questo capitolo e in quelli successivi ci concentreremo invece sulla fase di design e delle interazione del sistema JGuesser con i vari sistemi esterni. Il tutto verrà mostrato in una prima battuta con il diagramma di contesto.

Il diagramma di contesto è un diagramma in cui vengono identificati un insieme di sistemi esterni che si appoggiano / utilizzano il sistema e in cui compaiono i vari flussi d'informazione che questi sistemi si scambiano.

I sistemi esterni si possono dividere in:

- Sistemi superiori: che utilizzano il nostro sistema;
- Sistemi subordinati: utilizzati dal nostro sistema;
- Sistemi paritari: che hanno un rapporto paritario con il nostro sistema.
- Attori: soggetti che utilizzano il sistema, effettuando principalmente richieste.

#### 3.1 Diagramma di contesto

In questa sezione è presente il diagramma di contesto dell'applicazione JGuesser.

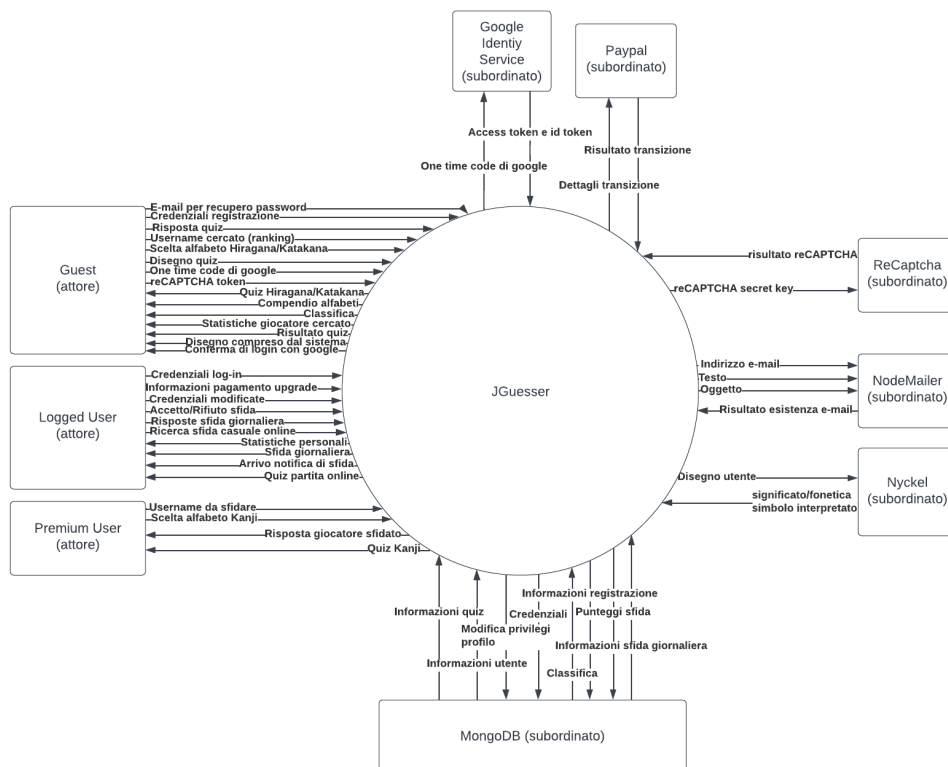


Figura 1: Diagramma di contesto dell'applicazione JGuesser

### 3.2 Descrizione interazioni attori e sistemi esterni

In questa sezione vengono descritti i diversi flussi d'informazione per ogni attore o sistema esterno.

#### 3.2.1 Guest

Guest è l'attore che rappresenta l'utente che non è loggato all'interno del sistema. L'utente guest invia una serie di dati in ingresso al sistema JGuesser:

- E-mail per recupero password: flusso che rappresenta l'invio di un'email di uno degli account registrati all'interno dell'applicazione JGuesser.
- Credenziali registrazione: flusso che rappresenta l'invio dei dati che l'utente utilizza per registrarsi (username, e-mail, password e nazione).
- Risposta quiz: flusso che rappresenta l'invio della risposta da parte dell'utente ad uno specifico quiz.
- Username cercato (ranking): flusso che rappresenta l'invio del username del giocatore che si vuole ricercare in classifica.
- Scelta alfabeto Hiragana/Katakana: flusso che rappresenta l'invio degli alfabeti selezionati dall'utente nel caso in cui questo abbia scelto come modalità di gioco 'training' del single-player.
- Disegno quiz: flusso che rappresenta l'invio dell'immagine del simbolo disegnato dall'utente nel caso in cui questo abbia giocato al quiz di tipo 4.
- One time code di google: flusso che rappresenta l'invio al sistema del one time code, ovvero di quel codice che l'utente guest ottiene dialogando con il sistema esterno google identity service. Viene inviato per ottenere una conferma di avvenuto login con google da parte dall'applicazione JGuesser.
- reCAPTCHA token: flusso che rappresenta l'invio del reCAPTCHA token, ovvero di quel token che l'utente guest ottiene dialogando con il sistema esterno reCAPTCHA. Questo dato viene inviato all'applicazione, perchè si vuole ottenere una conferma dall'applicazione che l'utente abbia cliccato effettivamente sul reCAPTCHA per verificarsi.

e riceve in ingresso le seguenti informazioni:

- Quiz Hiragana/Katakana: flusso che rappresenta la ricezione dei quiz di ogni tipologia, limitati agli alfabeti Hiragana e Katakana.
- Compendio alfabeti: flusso che rappresenta la ricezione della pagina che conterrà il compendio degli alfabeti.
- Classifica: flusso che rappresenta la ricezione dei dati della classifica, che verranno mostrati a video all'utente.
- Statistiche giocatore cercato: flusso che rappresenta la ricezione dei soli dati della classifica specifici a un solo giocatore scelto dall'utente. In questo modo l'utente potrà visualizzare soltanto i dati dell'utente che gli interessano.

- Risultato quiz: flusso che rappresenta la ricezione del risultato del quiz, per cui l'utente ha dato una risposta. Serve per far capire all'utente se ha risposto correttamente oppure no.
- Disegno compreso dal sistema: flusso che rappresenta la ricezione della fonetica/significato a cui è associato il disegno che il sistema esterno Nyckel ha interpretato. Questo flusso di dati prende in considerazione sia il risultato del quiz di tipo 4, ma anche la possibile richiesta di interpretazione da parte dell'utente.
- Conferma di login con google: flusso che rappresenta la ricezione di un dato che confermi all'utente che il login attraverso google è avvenuto con successo e che è stato aggiornato il proprio stato.

### 3.2.2 Logged User

Logged user è l'attore che rappresenta l'utente loggato all'interno del sistema. L'utente loggato può fare tutto quello che può fare l'utente guest e quindi eredita tutti i suoi flussi di dati in ingresso e in uscita. All'interno del diagramma di contesto per questo attore, sono stati individuati i seguenti flussi di dati in uscita diretti verso il sistema JGuesser:

- Credenziali log-in: flusso che rappresenta l'invio delle credenziali (username e password) per effettuare il login.
- Informazioni pagamento upgrade: flusso che rappresenta l'invio delle informazioni di pagamento. Con questo flusso si vuole far sapere al sistema che si ha intenzione di effettuare il pagamento della quota per effettuare l'upgrade del proprio profilo a premium.
- Credenziali modificate: flusso che rappresenta l'invio delle credenziali modificate (e-mail e password). Queste devono essere inviate quando l'utente che si trova nella propria area personale ha deciso di modificare la propria e-mail o password con una nuova.
- Accetto/Rifiuto sfida: flusso che rappresenta l'invio del messaggio di accetto o di rifiuto di una sfida.
- Risposte sfida giornaliera: flusso che rappresenta l'invio delle risposte con punteggi alle sfide giornaliere.
- Ricerca sfida casuale online: flusso che rappresenta l'invio del username del giocatore a cui si intende inviare l'invito di sfida.

e i seguenti flussi di dati in ingresso:

- Statistiche personali: flusso che rappresenta la ricezione di tutte le statistiche personali dell'utente.
- Sfida giornaliera: flusso che rappresenta la ricezione dei quiz di sfida giornaliera.
- Arrivo notifica di sfida: flusso che rappresenta la ricezione della notifica di sfida. La notifica viene ricevuta quando l'utente viene sfidato personalmente da un utente premium.

- Quiz partita online: flusso che rappresenta la ricezione del quiz di una partita online. Differisce dal Quiz Hiragana/Katakana, perchè su questo tipo di flusso possono viaggiare anche quiz di tipo Kanji, che nel single-player può scegliere soltanto il premium user.

### 3.2.3 Premium User

Il premium user è l'attore che rappresenta l'utente che ha effettuato l'aggiornamento del proprio account a premium. Il premium user eredita tutti i flussi di dati da parte del logged user. Per il premium user sono stati identificati i seguenti flussi di dati in uscita:

- Username da sfidare: flusso che rappresenta l'invio del username che il premium user vuole sfidare (multiplayer).
- Scelta alfabeto Kanji: flusso che rappresenta l'invio di un'informazione che dica al sistema che l'utente ha scelto l'alfabeto Kanji nella modalità 'training' del single-player.

e i seguenti flussi di dati in ingresso:

- Risposta giocatore sfidato: flusso che rappresenta la ricezione di un'informazione che dica all'utente sfidante, se questo invito è stato accettato oppure no.
- Quiz Kanji: flusso che rappresenta la ricezione dei quiz Kanji durante il training.

### 3.2.4 Google Identity Service

Google Identity Service è il sistema esterno che permetterà all'utente guest di effettuare l'accesso con le proprie credenziali google. Questo attore dovrà comunicare con l'applicazione JGuesser e con l'attore guest per verificare e quindi di conseguenza confermare l'accesso di un determinato utente tramite google. Nel diagramma di contesto sono stati individuati due flussi di dati. Un flusso di dati in ingresso al sistema Google Identity Service e un flusso di dati in uscita. Il flusso di dati in ingresso è rappresentato dal One time code che è un dato che l'utente avrà precedentemente inviato all'applicativo JGuesser e che il sistema dovrà inviare al sistema esterno per ricevere gli altri dati dell'utente. Ottenuti gli altri dati dell'utente, che sono l'access token e l'id token attraverso il flusso di uscita, l'applicativo potrà confermare il login con google. Questo sistema esterno compare nel RF1.4.

### 3.2.5 Paypal

Paypal è il sistema esterno con cui l'applicativo JGuesser dovrà interfacciarsi per permettere all'utente registrato di effettuare il pagamento di 2.99€ all'indirizzo email lorenzo.dambro@gmail.com. Per il seguente sistema sono stati individuati due flussi di dati. Un flusso di dati in ingresso e un flusso di dati in uscita. Il flusso di dati in ingresso rappresenta i dettagli della transazione che saranno forniti dall'applicazione. Il flusso di dati in uscita invece rappresenta il risultato della transazione e quindi verrà restituito al sistema JGuesser un valore, che gli permetterà di capire se la transazione è andata a buon fine oppure se ci sono stati degli errori. Questo sistema compare e serve a soddisfare il RF1.10.

### 3.2.6 reCaptcha

ReCAPTCHA è il sistema esterno con cui l'applicazione JGuesser si dovrà interfacciare per ottenere la conferma che l'utente non è un bot. In particolare questo tipo di sistema sarà coinvolto quando l'utente dovrà registrarsi e quando l'utente vorrà resettare la propria password. Nel diagramma di contesto sono stati identificati due flussi, uno in ingresso al sistema reCAPTCHA e l'altro in uscita. Il flusso d'ingresso è rappresentato dalla chiave segreta del sito, che sarà inviata al sistema reCAPTCHA, che dovrà rispondere con una verifica di successo/fallimento del token. Questa risposta è stata rappresentata nel diagramma attraverso l'unico flusso di uscita presente.

Questo sistema esterno è presente nei RF1.2 e RF1.9.

### 3.2.7 Nodemailer

Nodemailer è il sistema esterno che si occuperà di inviare tutte le e-mail che l'applicativo JGuesser necessita di inviare e che verificherà ogni singola e-mail che gli utenti inseriranno. All'interno del diagramma di contesto sono stati identificati quattro flussi di dati di cui tre in ingresso, che rappresentano:

- e-mail;
- oggetto;
- testo.

e uno in uscita che rappresenta un dato che permetterà di capire se l'e-mail fornita è un'e-mail valida e quindi effettivamente esistente oppure no. Il sistema JGuesser necessita di inviare e-mail nei seguenti 3 casi:

- un utente si è registrato e quindi viene inviata un e-mail di conferma avvenuta della registrazione. Vedere RF1.2.
- un utente ha dimenticato le proprie credenziali e quindi deve essere inviata un e-mail con le nuove credenziali. Vedere RF1.9.
- un utente ha effettuato l'upgrade del proprio profilo a premium e quindi viene inviata un e-mail di ringraziamento all'utente. Vedere RF1.10.

### 3.2.8 Nyckel

Nyckel è il sistema esterno che attraverso una serie di API, riuscirà a classificare i simboli dei diversi alfabeti, che l'utente andrà a disegnare. Nel diagramma di contesto sono stati specificati due flussi, uno in ingresso a Nyckel e l'altro in uscita. Quello in ingresso rappresenta il disegno che l'utente potrà disegnare nel quiz di tipo 4. Quello in uscita invece rappresenta la risposta e quindi la fonetica/significato del simbolo che Nyckel ha classificato.

Questo sistema esterno entra in gioco nel RF1.14.

### 3.2.9 MongoDB

MongoDB è il sistema esterno, che verrà utilizzato all'interno dell'applicazione per memorizzare tutte le informazioni riguardanti gli utenti, sia informazioni personali che statistiche. Oltre a ciò si utilizzerà per memorizzare tutte le informazioni riguardanti i simboli dei diversi alfabeti giapponesi. Nel diagramma di contesto sono stati identificati una serie di flussi di dati in ingresso:

### 3.2 *Descrizione interazioni attori e sistemi esterni*

---

- Informazioni registrazione: flusso che rappresenta la ricezione dal sistema delle informazioni di registrazione (username, password, nazione).
- Credenziali: flusso che rappresenta la ricezione delle credenziali utenti (username/email, password).
- Modifica privilegi profilo: flusso che rappresenta la ricezione di un dato che fa capire al database che deve aggiornare i privilegi dell'utente a premium.
- Punteggi sfida: flusso che rappresenta la ricezione dei punteggi di ogni singola sfida online e di ogni singola sfida giornaliera che l'utente gioca.

e una serie di flussi di dati in uscita:

- Informazioni utente: flusso che rappresenta l'invio di tutte le informazioni utente, che servono per soddisfare diversi RF.
- Informazioni quiz: flusso che rappresenta l'invio delle informazioni dei quiz che poi verranno inviata all'utente per visualizzare e svolgere il quiz.
- Classifica: flusso che rappresenta l'invio delle informazioni della classifica, che l'utente ha richiesto di visualizzare.
- Informazioni sfida giornaliera: flusso che rappresenta l'invio delle informazioni dei quiz giornalieri.

Questo sistema esterno è stato utilizzato per soddisfare moltissimi requisiti funzionali, quasi tutti.



## 4 Analisi Dei Componenti

In questo capitolo viene descritta l'architettura interna del sistema JGuesser. La presentazione dell'architettura interna avviene individuando le diverse componenti, che costituiscono il sistema, e specificando i flussi di dati in ingresso e in uscita individuati all'interno del diagramma di contesto. L'identificazione quindi delle componenti, l'interazione che essi hanno tra di loro e quindi i dati che devono scambiarsi e la specificazione dei flussi in entrata ed in uscita al sistema, permettono quindi di definire il diagramma delle componenti.

### 4.1 Definizione dei componenti

In questa sezione vengono definiti i componenti che formeranno il sistema JGuesser. In particolare ne viene motivata la loro esistenza, attraverso dei riferimenti ai requisiti funzionali e vengono spiegate le interfacce fornite o richieste.

#### 4.1.1 Pagina registrazione

**Descrizione:** questo componente si occupa di permettere all'utente di inserire le diverse informazioni per registrarsi e di inviare al componente validatore reCAPTCHA il proprio reCAPTCHA token. Oltre a ciò si occupa di verificare che il formato dei dati sia quello corretto. Questo componente è stato creato per rispondere al requisito funzionale 1.2.

**Interfaccia richiesta – Info registrazione:** per informazioni di registrazione si intendono:

- Username
- E-mail
- Password
- Conferma password
- Nazione

**Interfaccia richiesta – reCAPTCHA token:** cliccando sul reCAPTCHA presente nella pagina di registrazione, l'utente genera una chiave, che dovrà essere validato dal sistema esterno reCAPTCHA.

**Interfaccia fornita – reCAPTCHA token:** la chiave del reCAPTCHA deve essere inviata alla componente validatore reCAPTCHA, che si occuperà di effettuare la validazione.

**Interfaccia fornita – Info registrazione:** vengono inoltrare le informazioni di registrazione alla componente gestore credenziali.

**Interfaccia richiesta – Check registrazione:** flag di registrazione, che vale 'false' nel caso in cui si sia verificato qualche errore durante la registrazione. Serve a mostrare eventuali messaggi d'errore.

#### 4.1 Definizione dei componenti

---

##### 4.1.2 Gestore registrazione

**Descrizione:** dato il RF1.2, l'applicazione deve fornire all'utente la possibilità di registrarsi. Per questo motivo è stato definito un componente che si occupa di verificare e attuare la registrazione. Questa componente si occupa quindi di controllare che l'utente abbia inserito tutti i dati necessari e che questi rispettino alcuni vincoli. Oltre a ciò attuerà la registrazione interrogando il database e restituirà un check di conferma/errore alla componente di pagina di registrazione.

**Interfaccia fornita – Check registrazione:** flag di registrazione che sarà generato dalla componente gestore registrazione. Questo flag varrà 'true' se il Check reCAPTCHA e il Check e-mail sono anche essi positivi e se valgono una serie di controlli effettuati all'interno di questa componente, es: non esiste alcun utente registrato con la stessa e-mail inserita dall'utente.

**Interfaccia fornita – e-mail:** dato che verrà fornito all'interfaccia Nodemailer, che si occuperà di comunicare con il sistema esterno Nodemailer per verificare che l'e-mail sia una e-mail esistente e per sapere a quale indirizzo deve inviare l'e-mail di avvenuta registrazione, nel caso l'operazione di registrazione vada a buon fine.

**Interfaccia fornita – Richiesta inserimento info registrazione:** viene inviata la richiesta d'inserimento dei dati di registrazione al sistema esterno MongoDB.

**Interfaccia richiesta – Info registrazione:** la componente gestione registrazione riceve tutte le informazioni di registrazione che sono state raccolte dalla pagina di registrazione.

**Interfaccia richiesta – Check e-mail:** valore che il gestore della registrazione riceverà dalla componente interfaccia Nodemailer e che serve a capire se l'e-mail inserita dall'utente è un e-mail valida oppure no.

**Interfaccia richiesta – Check inserimento dati:** verifica da parte del sistema esterno MongoDB, che l'utente è stato registrato correttamente. Potrebbe verificarsi che per l'e-mail che l'utente ha specificato sia già esistente un profilo registrato.

**Interfaccia richiesta – Credenziali google per registrazione:** insiemi di dati che si ricevono dalla componente autenticazione con google che servono a registrare l'utente all'interno del database nel caso in cui sia la prima volta che sta effettuando l'accesso con google.

**Interfaccia richiesta – Check validazione reCAPTCHA:** valore che verrà ricevuto dalla componente validatore reCAPTCHA e che servirà alla componente gestore registrazione per capire se l'utente ha cliccato sul reCAPTCHA oppure no e quindi si è validato correttamente come non bot.

#### 4.1.3 Validatore reCAPTCHA

**Descrizione:** questa componente si occupa di verificare il reCAPTCHA token che l'utente invia al server. Il server deve infatti potersi interfacciare con il sistema esterno reCAPTCHA per capire se l'utente ha confermato correttamente la sua identità oppure no. Questo componente permette di realizzare i RF1.9 e RF1.2.

**Interfaccia richiesta – reCAPTCHA token:** la componente validatore reCAPTCHA riceve dall'utente il reCAPTCHA token che dovrà essere validato.

**Interfaccia richiesta – Check reCAPTCHA:** il sistema esterno reCAPTCHA restituisce alla componente validatore reCAPTCHA un dato, che permette di capire se la validazione è avvenuta con successo oppure no.

**Interfaccia fornita – reCAPTCHA secret key:** il server invia al sistema esterno reCAPTCHA il proprio reCAPTCHA secret key, che servirà per poter verificare il reCAPTCHA token.

**Interfaccia fornita – Check reCAPTCHA:** il dato che permette di capire la validità del reCAPTCHA viene inoltrato alle componenti: gestore registrazione e gestore credenziali a secondo di chi lo sta utilizzando in quel momento.

#### 4.1.4 Interfaccia Nodemailer

**Descrizione:** questa componente è quella che si interfaccia con il sistema esterno Nodemailer. È stata realizzata un'unica componente, perché il sistema di invio e-mail è un sistema che cambia frequentemente ed è utile incapsulare tutte le interazioni in un'unica componente. Questa componente risponde ai requisiti funzionali RF1.2, RF1.10 e RF1.9.

**Interfaccia richiesta – Check e-mail:** l'interfaccia Nodemailer riceverà dal sistema esterno un dato che li permetterà di capire se l'e-mail inserita dall'utente è un e-mail esistente e quindi valida oppure no.

**Interfaccia richiesta – e-mail:** l'interfaccia Nodemailer può ricevere un e-mail grazie al quale si dovranno eseguire delle operazioni da parte delle componenti: gestore registrazione, gestore credenziali e gestore upgrade profilo.

**Interfaccia fornita – Richiesta verifica e-mail:** viene fornita al sistema esterno Nodemailer l'email di cui verificare l'esistenza.

**Interfaccia fornita – Check e-mail:** il dato se l'e-mail è valida oppure no viene inoltrato alla componente di gestore registrazione.

**Interfaccia fornita – Invio e-mail conferma registrazione:** viene inviata a Nodemailer una richiesta di inviare un e-mail all'utente che si è appena registrato, in cui viene avvisato di ciò.

#### 4.1 Definizione dei componenti

---

**Interfaccia fornita – Invio e-mail reset credenziali:** viene inviata a Nodemailer una richiesta di inviare un e-mail all'utente, che ha richiesto di resettare la propria password. L'utente riceverà quindi una e-mail con all'interno il proprio username e una nuova password generata casualmente.

**Interfaccia fornita – Invio e-mail ringraziamento upgrade profilo:** viene inviata a Nodemailer una richiesta di inviare un e-mail di ringraziamento all'utente, che ha appena effettuato l'upgrade del proprio profilo alla versione premium.

##### 4.1.5 Pagina login

**Descrizione:** questa componente rappresenta l'interfaccia di login del sistema JGuesser. Ha quindi il compito di raccogliere le informazioni di login, verificare errori di formati delle credenziali e mostrare all'utente eventuali messaggi di errore che si possono verificare durante questa fase. Questa componente contribuisce a soddisfare i requisiti funzionali RF1.4 e RF1.3.

**Interfaccia richiesta – One time code di google:** nel caso in cui l'utente abbia scelto di effettuare il login con google, otterrà un one time code che invierà alla componente di autenticazione con google.

**Interfaccia richiesta – Credenziali:** l'utente inserisce nella pagina le proprie credenziali, per effettuare l'accesso.

**Interfaccia richiesta – Check login google:** la pagina di login riceve dalla componente autenticazione con google un dato, che permetterà di capire se l'accesso con google è avvenuto con successo oppure no.

**Interfaccia richiesta – Check login:** la pagina di login riceve dalla componente autenticazione un dato, che permetterà di capire se il login tramite credenziali è avvenuto con successo oppure no.

**Interfaccia fornita – One time code di google:** viene inviato il one time code ottenuto dall'utente alla componente autenticazione con google.

**Interfaccia fornita – Credenziali:** vengono inviate le credenziali di login alla componente autenticazione.

##### 4.1.6 Autenticazione

**Descrizione:** questa componente si occuperà di verificare l'autenticazione tramite le credenziali inserite dall'utente. Questa componente risponde al requisito funzionale RF1.3.

**Interfaccia richiesta – Credenziali:** riceve le credenziali dalla componente pagina login.

#### 4.1 Definizione dei componenti

---

**Interfaccia richiesta – Password utente:** viene richiesto all'utente di inserire la propria password, che verrà poi verificata con quella associata all'username inserito all'interno del database.

**Interfaccia fornita – Check login:** dato che viene restituito alla pagina di login, che indica se l'autenticazione ha avuto successo oppure no.

**Interfaccia fornita – Richiesta password utente:** la componente interagirà con il sistema esterno MongoDB per inviare una richiesta di password dell'username inserito dall'utente.

##### 4.1.7 Autenticazione con Google

**Descrizione:** questa componente serve a gestire le interazioni con il sistema esterno google identity service e ad effettuare il login con google. Questa componente risponde al requisito funzionale RF1.4.

**Interfaccia richiesta – Info google:** questa componente interagirà con il sistema esterno google identity service, che fornirà:

- access token;
- id token;
- e-mail.

**Interfaccia richiesta – One time code di google:** riceve il one time code di google dalla pagina di login e quindi dall'utente.

**Interfaccia fornita – Credenziali google per registrazione:** nel caso in cui sia la prima volta che un utente effettua l'accesso con google, le informazioni che vengono restituite serviranno per registrare un account per l'utente.

**Interfaccia fornita – One time code di google:** si interagisce con il google identity service fornendogli il one time code.

**Interfaccia fornita – Check login google:** si restituisce alla pagina di login un dato, che permetterà di capire se l'accesso con account google è avvenuto con successo oppure no.

##### 4.1.8 Pagina dati personali

**Descrizione:** questo componente si occupa di permettere la modifica e la visualizzazione dei dati personali da parte dell'utente. I dati personali visualizzabili saranno: nazionalità, username e indirizzo mail. I dati modificabili saranno: indirizzo mail e password, per la modifica della prima sarà necessario inserire una mail, esistente e non associata ad un ulteriore account, sostitutiva. Per la modifica della password sarà

#### 4.1 Definizione dei componenti

---

necessario inserire la password attuale e la nuova password (che dovrà rispettare le stesse condizioni della fase di registrazione) per 2 volte. All'interno di questa componente verrà effettuato un controllo di uguaglianza tra le 2 nuove password inserite e, in caso di esito positivo del confronto, verrà inviata la prima di queste due più la vecchia password alla componente Gestore credenziali. Questo componente è stato creato per rispondere ai requisiti funzionali RF1.6, RF1.7 e RF1.8.

**Interfaccia richiesta – Nuove credenziali:** vengono richieste all'utente le nuove credenziali, ovvero il nuovo indirizzo e-mail in caso si voglia modificare quest'ultima, altrimenti la nuova password per 2 volte, sarà poi la componente stessa che verificherà la corrispondenza tra le due.

**Interfaccia richiesta – Vecchia password:** viene richiesto all'utente di inserire la vecchia password, utilizzata per permettere la modifica della stessa, tale password verrà poi inoltrata al Gestore credenziali, che si occuperà di accertarne la validità.

**Interfaccia richiesta – Check modifica credenziali:** viene richiesta una conferma da parte del Gestore credenziali dell'avvenuta modifica delle credenziali.

**Interfaccia richiesta – Dati utente:** vengono richiesti dal Gestore credenziali i dati utente da visualizzare nelle informazioni personali, tali dati saranno poi mostrati all'utente.

**Interfaccia fornita – Informazioni dati personali:** vengono mostrati all'utente i propri dati personali.

**Interfaccia fornita – Nuove credenziali:** viene fornita al Gestore credenziali la nuova credenziale inserita dall'utente, che potrà essere un nuovo indirizzo mail, o una nuova password, a seconda della richiesta effettuata dall'utente. Tale credenziale verrà inserita in seguito all'interno del database e successivamente, in caso di modifica indirizzo mail, verrà inviata a quest'ultimo una mail di conferma.

**Interfaccia fornita – Vecchia password:** viene fornita al Gestore credenziali la vecchia password dell'utente, inserita da quest'ultimo, per verificarne la corrispondenza ed eventualmente permettere la modifica.

##### 4.1.9 Pagina recupero username e password

**Descrizione:** questo componente si occupa di permettere il recupero di username e password, tali dati verranno inviati via mail. In caso di richiesta di username, verrà inviata all'indirizzo mail fornito una mail contenente l'username corrispondente a tale indirizzo. In caso di richiesta di password, verrà inviata all'indirizzo mail fornito una mail contenente una nuova password generata, che l'utente potrà visualizzare in chiaro e poi modificarla tramite l'apposita sezione. Prima di ogni richiesta di recupero sarà necessario l'autorizzazione reCAPTCHA. Questo componente è stato creato per rispondere al requisito funzionale RF1.9.

**Interfaccia richiesta – Check reset:** viene richiesta al Gestore credenziali una conferma dell'avvenuto invio di una mail contenente le credenziali richieste. Per far sì che ciò accada, lo stesso Gestore credenziali dovrà aver ricevuto la conferma del

#### 4.1 Definizione dei componenti

---

corretto reCAPTCHA e dell'effettiva corrispondenza tra l'indirizzo mail utilizzato per il recupero e quello associata ad uno degli account all'interno del database.

**Interfaccia richiesta – ReCAPTCHA chiave sito:** viene richiesta la validazione da parte dell'utente di un form reCAPTCHA, il risultato verrà inviato al Validatore reCAPTCHA, che si occuperà di verificarne la validità.

**Interfaccia richiesta – E-mail:** viene richiesto all'utente di inserire l'indirizzo mail associato all'account del quale si vogliono ottenere le credenziali.

**Interfaccia fornita – ReCAPTCHA chiave sito:** viene fornito il reCAPTCHA compilato dall'utente al Validatore reCAPTCHA, che si occuperà di verificarne la correttezza.

**Interfaccia fornita – E-mail:** viene inoltrato l'indirizzo mail inserito dall'utente al Gestore credenziali, il quale si occuperà di verificarne la corrispondenza.

##### 4.1.10 Gestore credenziali

**Descrizione:** questo componente si occupa di estrapolare e manipolare le credenziali. Inoltre, si occupa della gestione della comunicazione con Nodemailer (per inviare la mail di recupero password) e con il validatore reCaptcha. Questo componente è stato creato per rispondere ai requisiti funzionali RF1.6, RF1.7, RF1.8, RF1.9.

**Interfaccia richiesta – Dati utente:** vengono richiesti al database i dati di un determinato utente, che verranno poi inoltrati e mostrati all'utente stesso, oppure utilizzati per operazioni di verifica.

**Interfaccia richiesta – Check modifica credenziali:** viene richiesta al database una conferma di avvenuta modifica delle credenziali fornite dall'utente.

**Interfaccia richiesta – Vecchia password:** viene richiesta a Pagina dati personali (la quale la otterrà dall'utente stessi), la password corrente, che l'utente è intenzionato a sovrascrivere. All'interno di questa componente sarà poi possibile verificare che tale indirizzo corrisponda a uno di quelli presenti sul database, ed in caso affermativo effettuare la sovrascrittura.

**Interfaccia richiesta – Nuove credenziali:** viene richiesta alla Pagina dati personali la nuova credenziale inserita dall'utente, in caso si tratti di un nuovo indirizzo mail, verrà verificato che quest'ultimo sia un indirizzo esistente e che non sia già associato ad un account esistente ed in caso negativo verrà sovrascritto l'indirizzo all'interno del database. In caso si tratti di una nuova password, verrà, dopo la verifica di corrispondenza della vecchia password, sovrascritta all'interno del database quest'ultima in favore della nuova.

**Interfaccia richiesta – E-mail:** viene richiesto alla Pagina recupero username e password l'indirizzo mail, fornito dall'utente stesso, al quale inviare la mail con il recupero delle credenziali, tale indirizzo, dopo averne verificato l'esistenza all'interno del database (e quindi l'associazione ad un determinato account) verrà poi inoltrato all'interfaccia Nodemailer, che si occuperà della spedizione della mail.

#### 4.1 Definizione dei componenti

---

**Interfaccia richiesta – Check validazione reCAPTCHA:** viene richiesta al Validatore reCAPTCHA la conferma della corretta validazione del reCAPTCHA, in caso positivo, sarà possibile procedere con il recupero username e password, segnalando l'esito alla Pagina recupero username e password.

**Interfaccia fornita – Richiesta dati utente:** viene effettuata una richiesta al database, finalizzata all'ottenimento dei dati di un determinato utente, i quali verranno poi utilizzati per alcune verifiche o inoltrati direttamente a quest'ultimo.

**Interfaccia fornita – Richiesta modifica credenziali:** viene effettuata sul database una richiesta di modifica credenziali, con annesse le nuove credenziali fornite dall'utente che si vogliono sovrascrivere.

**Interfaccia fornita – Dati utente:** vengono forniti i dati utente, risultanti dall'interrogazione al database all'interfaccia Pagina dati personali, che si occuperà di farli visualizzare all'utente stesso.

**Interfaccia fornita – Check modifica credenziali:** viene fornito alla Pagina dati personali l'esito dell'avvenuta modifica delle credenziali.

**Interfaccia fornita – Check reset:** viene fornita alla Pagina recupero username e password la conferma dell'invio della mail contenente le credenziali richieste dall'utente.

**Interfaccia fornita – e-mail:** viene fornito all'Interfaccia Nodemailer l'indirizzo mail, inserito dall'utente, al quale inviare la mail di recupero delle credenziali richieste dallo stesso.

##### 4.1.11 Pagina pagamento

**Descrizione:** questo componente si occupa di permettere il pagamento, finalizzato all'upgrade del profilo, tramite Paypal. Questo componente è stato creato per rispondere al requisito funzionale RF1.10.

**Interfaccia richiesta – Effettua pagamento:** viene richiesto dall'utente di effettuare il pagamento. Una volta terminata l'operazione, l'esito della stessa viene comunicato al Gestore pagamenti.

**Interfaccia fornita – Informazioni pagamento:** viene fornito il risultato della transazione effettuata dall'utente al Gestore pagamenti, che si occuperà di proseguire le operazioni finalizzate all'upgrade del profilo.

##### 4.1.12 Gestore pagamenti

**Descrizione:** questo componente si occupa di gestire la transazione finalizzata all'upgrade del profilo da user a premium user. Questo componente è stato creato per rispondere al requisito funzionale RF1.10.



#### 4.1 Definizione dei componenti

---

**Interfaccia richiesta – Informazioni pagamento:** vengono richieste le informazioni sull'avvenuto o meno pagamento.

**Interfaccia richiesta – Risposta pagamento:** viene richiesto a Paypal l'esito della transazione effettuata dall'utente. Tale risultato sarà utilizzato, in caso di esito positivo, per permettere al Gestore upgrade profilo di effettuare l'upgrade del profilo.

**Interfaccia fornita – Informazioni pagamento:** vengono fornite a Paypal le informazioni fornite dall'utente relative alla transazione.

**Interfaccia fornita – Risposta pagamento:** viene fornito il risultato della transazione effettuata dall'utente ed approvata da Paypal. In caso di esito positivo, il Gestore upgrade profilo si occuperà di aggiornare l'account da user a premium user.

##### 4.1.13 Gestore upgrade profilo

**Descrizione:** questo componente si occupa di far variare un account da user a premium user. Questo componente è stato creato per rispondere al requisito funzionale RF1.10.

**Interfaccia richiesta – Risposta pagamento:** viene richiesto l'esito del pagamento effettuato dall'utente ed approvato da Paypal. In caso di esito positivo si effettuerà l'upgrade del profilo.

**Interfaccia fornita – E-mail:** viene fornito all'interfaccia Nodemailer un indirizzo al quale inviare una e-mail di ringraziamento per aver effettuato l'upgrade del profilo.

**Interfaccia fornita – Upgrade profilo:** viene aggiornato nel database l'utente, in modo da farlo diventare un utente premium.

##### 4.1.14 Pagina statistiche

**Descrizione:** questo componente si occupa di mostrare all'utente registrato o premium una sezione riportante le proprie statistiche. Per statistiche si intendono: numero di partite, punteggio totale, numero di vittorie e di sconfitte (giornaliere/classificate) e rapporto tra quest'ultime. Questo componente è stato creato per rispondere al requisito funzionale RF1.19.

**Interfaccia richiesta – Statistiche utente:** vengono richieste al gestore dati utente le statistiche che verranno poi mostrate all'utente stesso.

**Interfaccia fornita – Statistiche utente:** vengono fornite all'utente le proprie statistiche, ricevute dal gestore dati utente.

##### 4.1.15 Pagina classifica

**Descrizione:** questo componente si occupa di mostrare la classifica dei primi 100 giocatori, con la possibilità di filtrare e visualizzare utenti specifici, grazie ad un apposito

#### 4.1 Definizione dei componenti

---

campo che permette il filtraggio in base all'username. Nella classifica saranno mostrate le seguenti informazioni: posizione in classifica, nazionalità, username, punteggio totale e partite giocate. Questo componente è stato creato per rispondere ai requisiti funzionali RF1.21 e RF1.21.

**Interfaccia richiesta – Username:** viene richiesto all'utente l'username dell'utente per il quale si vuole filtrare la visualizzazione della classifica.

**Interfaccia richiesta – Informazioni giocatori:** vengono richieste al Gestore dati utente le informazioni dei giocatori, eventualmente filtrati per username, che verranno visualizzati dall'utente in classifica.

**Interfaccia fornita – Username:** viene fornito al Gestore dati utente l'username dell'utente per il quale si vogliono visualizzare le informazioni nella classifica, username che verrà fornito dall'utente stesso.

**Interfaccia fornita – Informazioni giocatori:** vengono fornite all'utente le informazioni relative ai giocatori, filtrati per username o meno, ottenute dal Gestore dati utente.

##### 4.1.16 Gestore dati utente

**Descrizione:** questo componente si occupa di fornire determinati dati di determinati utenti e di aggiornare le statistiche degli utenti. I dati possono essere le statistiche di un giocatore oppure i dati di un giocatore. Nelle prime possiamo trovare numero di partite, punteggio totale, numero di vittorie e di sconfitte (giornaliere/classificate) e rapporto tra quest'ultime. Nelle seconde possiamo trovare i seguenti dati: posizione in classifica, nazionalità, username, punteggio totale e partite giocate. Questi dati possono riguardare un singolo utente, nel caso di statistiche personali o di filtraggio in classifica per utente, o riguardanti un blocco delle prime 100 persone in classifica. Questo componente è stato creato per rispondere ai requisiti funzionali RF1.21, RF1.20 e RF1.19.

**Interfaccia richiesta – Username:** viene richiesto dalla componente Pagina classifica l'username, inserito dall'utente, del quale si vogliono visualizzare le informazioni in classifica.

**Interfaccia richiesta – Dati primi 100 giocatori:** vengono richiesti dal database i dati dei primi 100 giocatori, che verranno poi visualizzati in classifica.

**Interfaccia richiesta – Dati utente:** vengono richiesti i dati di un utente specifico, dati che verranno utilizzati o all'interno della Pagina classifica o della Pagina statistiche.

**Interfaccia fornita – Statistiche utente:** vengono fornite alla Pagina statistiche le statistiche relative all'utente che le ha richieste, le quali saranno poi mostrate all'utente stesso.

**Interfaccia fornita – Richiesta dati utenti:** viene fornita una richiesta al database per ottenere i dati di un determinato utente, che saranno poi smistati in base

#### 4.1 Definizione dei componenti

---

al componente che le richiede.

**Interfaccia fornita – Richiesta dati primi 100 giocatori:** viene fornita una richiesta al database per ottenere i dati dei primi 100 giocatori, che verranno poi visualizzati in classifica.

**Interfaccia fornita – Informazioni giocatori:** vengono fornite alla Pagina classifica le informazioni dell'utente selezionato, che verranno poi visualizzati in classifica.

**interfaccia fornita - Aggiornamento statistiche utente:** vengono aggiornate nel database le statistiche di un utente in base all'esito della sfida ricevuto dalla componente Pagina quiz.

##### 4.1.17 Pagina single-player

**Descrizione:** questo componente si occupa di fornire all'utente la possibilità di scegliere l'alfabeto e la modalità con cui giocare. Inoltre, tale componente deve occuparsi di prendere le risposte dell'utente e fornire le soluzioni ai quiz che ha completato.

**Interfaccia richiesta - Alfabeti selezionati:** cliccando sulla scelta multipla degli alfabeti, l'utente seleziona gli alfabeti con i quali intende giocare.

**Interfaccia richiesta - Tipo modalità:** cliccando su una delle due scelte (training o sfida giornaliera) l'utente seleziona la modalità con cui giocare.

**Interfaccia fornita - Alfabeti selezionati:** la scelta dell'alfabeto viene inoltrata alla componente quiz maker.

**Interfaccia fornita - Tipo di modalità:** la scelta della modalità viene inoltrata alla componente quiz maker.

##### 4.1.18 Quiz maker

**Descrizione:** questo componente si occupa di generare randomicamente i quiz in base alle scelte dell'utente, chiedendo le informazioni necessarie al database, per poi inviare la lista di quiz creata alla componente pagina quiz.

**Interfaccia richiesta - Alfabeti selezionati:** riceve dalla componente pagina single-player la scelta dell'alfabeto effettuata dall'utente, e in base ad essa stabilisce con che query interrogare il database.

**Interfaccia richiesta - Tipo di modalità:** riceve dalla componente pagina single-player la scelta della modalità, grazie ad essa decide quanti quiz generare.

**Interfaccia richiesta - Richiesta quiz sfida online:** riceve dalla componente gestore sessione di gioco la richiesta di generare i quiz per una sfida online (2 Kanji + 4 Katakana + 4 Hiragana).

#### 4.1 Definizione dei componenti

---

**Interfaccia richiesta - Info simboli:** le informazioni richieste variano a seconda del tipo di quiz che deve essere generato:

- nel caso di quiz tipo 2 e tipo 3 vengono richieste le informazioni di 4 simboli, che sono tutti e quattro dello stesso alfabeto, questo alfabeto però potrebbe variare a seconda della scelta dell'utente;
- nel caso del quiz tipo 1 e tipo 4 vengono richieste le informazioni solamente di un simbolo, anche qui l'alfabeto da cui viene scelto dipende dalle scelte dell'utente.

**Interfaccia fornita - Richiesta info simboli:** vengono richieste al database le informazioni necessarie alla creazione dei quiz, queste informazioni variano congruentemente con quelle descritte nell'interfaccia precedente.

**Interfaccia fornita - Lista quiz con info per correzione:** viene messa a disposizione della componente pagina quiz la lista dei quiz generati, insieme ad essi vengono anche trasmesse le informazioni per poter verificare la correttezza delle risposte dell'utente, queste informazioni variano a seconda del quiz:

- nel caso dei quiz di tipo 2 e 3 viene esplicitata quale è la scelta corretta tra quelle disponibili;
- nel caso dei quiz di tipo 1 e 4 viene fornita la pronuncia (Katakana/Hiragana) o il significato (Kanji) sotto forma di stringa, che ci si aspetta che l'utente scriva come risposta.

##### 4.1.19 Pagina quiz

**Descrizione:** questo componente fornisce all'utente la pagina con i vari quiz possibili, inoltre deve anche ricevere le risposte dell'utente per i vari quiz e verificarne la correttezza, dandogli un feedback.

**Interfaccia richiesta - Risposta utente:** la risposta dell'utente varia a seconda del quiz fornitogli, essa si può suddividere in:

- selezione di una delle scelte multiple fornite (quiz tipo 2 e tipo 3);
- stringa corrispondente o al suono o al significato del simbolo visualizzato a schermo, a seconda dell'alfabeto a cui fa riferimento (quiz tipo 1);
- un disegno del simbolo fornito (tramite pronuncia o significato), il più possibile simile al simbolo stesso.

**Interfaccia richiesta - Lista quiz con info per correzione:** riceve dalla componente quiz maker una lista di quiz e le relative soluzioni, la lista può essere composta da 5 o da 30 quiz nel caso debba essere visualizzata una sfida giornaliera, oppure da 10 quiz nel caso dell'allenamento, una volta completati i 10 quiz ne saranno inviati altri 10 sempre dalla componente quiz maker, fino a quando il giocatore non decide di smettere di giocare.

**Interfaccia richiesta - Disegno interpretato:** dopo che l'utente avrà inviato il suo disegno o dopo che abbia chiesto al sistema di mostrargli l'interpretazione fatta di esso, viene richiesta alla componente sistema interpretazione disegno l'interpretazione

#### 4.1 Definizione dei componenti

---

del disegno, questa non è altro ”che una stringa corrispondente al significato o alla pronuncia del simbolo interpretato dal sistema tramite l’API Nyckel”.

**Interfaccia fornita - Quiz:** vengono mostrati all’utente i quiz, uno ad uno in sequenza.

**Interfaccia fornita - Dettagli score:** man mano che l’utente procede nella risoluzione dei quiz, gli viene mostrato il suo punteggio aggiornato.

**Interfaccia fornita - Risultato quiz:** dopo che l’utente ha dato la sua risposta, gli viene mostrata la soluzione al quiz, o nel caso del quiz tipo 4, sotto richiesta dell’utente, viene mostrata l’interpretazione del disegno fatta dal sistema per verificare che corrisponda con lo stesso simbolo che l’user ha cercato di disegnare.

**interfaccia fornita - Esito quiz:** nel caso venga fornita all’utente una sfida giornaliera o una sfida online, il punteggio dell’utente viene fornito alla componente aggiornamento statistiche utente.

**Interfaccia fornita - Disegno:** se all’utente viene fornito il quiz tipo 4, il disegno fatto dall’utente viene inoltrato alla componente sistema interpretazione disegno.

**Interfaccia fornita - Termine sfida:** nel caso venga fornita una sfida online all’utente, viene inviato un booleano alla componente gestore sessione di gioco quando uno dei due utenti coinvolti nella sfida avrà fornito le sue risposte a tutti i quiz della sfida.

**Interfaccia fornita - Esito sfida:** quando un utente finisce una sfida giornaliera o una sfida online, fornisce alla componente gestore dati utenti l’esito della sfida per aggiornare le statistiche.

##### 4.1.20 Sistema interpretazione disegno

**Descrizione:** questo componente fa da interfaccia per l’API Nyckel, ha quindi il ruolo di inviare a Nyckel il disegno fatto dall’utente e di inoltrare al sistema l’interpretazione fatta da Nyckel del disegno.

**Interfaccia richiesta - Interpretazione disegno:** l’interpretazione del disegno fatta da Nyckel viene inviata alla componente.

**Interfaccia richiesta - Disegno:** viene richiesta dalla componente pagina quiz il disegno dell’utente.

**interfaccia fornita - Disegno:** il disegno dell’utente viene inoltrata all’API Nyckel per essere interpretato.

**interfaccia fornita - Interpretazione disegno:** l’interpretazione del disegno viene inoltrata alla componente pagina quiz per essere visualizzata all’utente.

#### 4.1 Definizione dei componenti

---

##### 4.1.21 Pagina multi-player

**Descrizione:** questo componente fornisce all'utente la possibilità di scegliere di partecipare a una sfida online con un altro giocatore casuale oppure di sfidare un altro utente fornendo l'username di quest'ultimo.

**Interfaccia richiesta - Username:** l'utente premium scrive lo username dell'utente che intende sfidare.

**Interfaccia richiesta - Tipo di ricerca avversario:** l'utente sceglie se fare una sfida casuale o se sfidare uno giocatore specifico.

**Interfaccia richiesta - Check giocatore:** viene richiesta una conferma dell'esistenza del giocatore cercato, questo check viene fatto attraverso un booleano.

**Interfaccia richiesta - Risposta sfida:** viene richiesta la risposta (booleano) dell'utente cercato (se esiste), affinché possa essere notificata all'utente sfidante.

**Interfaccia fornita - Esito ricerca:** viene fornita all'utente sfidante una conferma dell'inizio della sfida, o del rifiuto di essa, a seconda della risposta dell'utente sfidato. Se invece lo username cercato non esiste, l'utente sfidante viene notificato di ciò.

**Interfaccia fornita - Username:** viene inoltrato l'username dell'utente sfidato al componente sistema di ricerca giocatori per verificare che l'utente esista.

**Interfaccia fornita - Tipo di ricerca avversario:** viene inoltrato il tipo di ricerca avversario fatta dall'utente al componente sistema ricerca giocatori per verificare se è necessario un check giocatore.

##### 4.1.22 Sistema di ricerca giocatori

**Descrizione:** questo componente si occupa di controllare che lo username inserito dall'utente sfidante sia valido facendo una query verso il database. Inoltra alla componente pagina multi-player il check-giocatore (booleano). Se lo username è valido, lo inoltra alla componente gestore invio invito di sfida. Se invece viene scelta l'opzione sfida casuale, inoltra la scelta alla componente gestore giocatori disponibili.

**Interfaccia richiesta - Username** lo username viene richiesto alla componente pagina multi-player.

**interfaccia richiesta - Tipo di ricerca avversario:** la modalità di ricerca avversario (per nome o casuale) viene inoltrata dalla componente pagina multi-player.

**Interfaccia richiesta - Check giocatore:** viene richiesto al database l'esistenza o meno dell'utente con l'username fornito dalla pagina multi-player.

**Interfaccia fornita - Richiesta di ricerca giocatore per username:** viene fatta una query al database per verificare che l'username inserito esista nel sistema, riceverà

#### 4.1 Definizione dei componenti

---

una risposta booleana chiamata *check giocatore*.

**Interfaccia fornita - Check giocatore:** il valore di *check giocatore* viene fornito alla componente pagina multi-player, per dare un feedback all'utente sfidante.

**Interfaccia fornita - Username:** viene inoltrato alla componente gestore invio inviti di sfida lo username dell'utente sfidato (se *check giocatore* è uguale a *true*) per notificarlo.

**Interfaccia fornita - Richiesta ricerca casuale:** se il tipo di sfida fatta dall'utente è la sfida casuale, questa informazione viene inoltrata alla componente gestore giocatori disponibili.

##### 4.1.23 Gestore invio inviti di sfida

**Descrizione:** questo componente si occupa di inviare all'utente sfidato una notifica di tale sfida, e di reindirizzare la risposta di quest'ultimo al fine di notificare l'utente sfidante dell'esito della sua richiesta. Se l'utente sfidato accetta la sfida, si occupa anche di inoltrare gli username dei due utenti alla componente gestore sessione di gioco per far avvenire la sfida.

**Interfaccia richiesta - Username:** viene richiesto lo username sotto forma di stringa per poter inviare a questo utente, se online, la richiesta di sfida.

**Interfaccia richiesta - Risposta sfida:** riceve dalla componente notifica di sfida la risposta dell'utente sfidato sotto forma di booleano.

**Interfaccia fornita - Username utente sfidante e sfidato:** se la risposta sfida ha come valore *true*, viene fornita alla componente gestione sessione di gioco la coppia di username dei due utenti che parteciperanno alla sfida.

**interfaccia fornita - Richiesta invio sfida:** viene inoltrata alla componente notifica di sfida la richiesta di invito affinché possa essere visualizzato il pop-up a schermo dell'utente sfidato.

##### 4.1.24 Notifica di sfida

**Descrizione:** questo componente si occupa di far visualizzare all'utente che ha ricevuto una richiesta di sfida la richiesta stessa, e di notificare la risposta di quest'ultimo alla pagina multi-player e al gestore invio inviti sfida.

**Interfaccia richiesta - Richiesta invio invito:** viene ricevuta la richiesta dell'invito dalla componente gestore invio inviti sfida, per essere notificata all'utente sfidato. In questa interfaccia sono anche presenti le informazioni che permettono alla componente di capire a che utente far apparire la richiesta di sfida.

**Interfaccia richiesta - Risposta sfida:** riceve dall'utente la risposta alla sfida (interpretata sotto forma di booleano), questa risposta dovrà poi essere inoltrata alle

#### 4.1 Definizione dei componenti

---

componenti gestore invio inviti sfida e pagina multi-player.

**Interfaccia fornita - Invito sfida:** fornisce all'utente una schermata di invito a una sfida, dove sono presenti lo username del giocatore sfidante, e due pulsanti per accettare o rifiutare la sfida.

**Interfaccia fornita - Risposta sfida:** la risposta dell'utente sfidato viene inoltrata alla componente gestore invio inviti di sfida per processarne l'esito.

##### 4.1.25 Gestore giocatori disponibili

**Descrizione:** questo componente si occupa di aggiornare lo stato nel database del giocatore che ha fatto richiesta di partecipare a una sfida online, dopo di che chiede al database tutti gli utenti che sono in ricerca di una partita casuale, decide un utente da accoppiare al giocatore che ha iniziato la richiesta di sfida casuale, la coppia degli username degli utenti verrà poi inviata alla componente gestore sessione di gioco per far iniziare la sfida.

**Interfaccia richiesta - Richiesta ricerca casuale:** riceve dalla componente sistema di ricerca giocatori la richiesta di una sfida casuale da parte di un utente, questa interfaccia contiene anche l'username del giocatore che ha fatto la richiesta, affinché possa essere accoppiato con un altro utente estratto casualmente tra quelli disponibili.

**Interfaccia richiesta - Giocatori disponibili:** viene ricevuta dal database la lista degli utenti che sono in stato di ricerca di una partita casuale.

**Interfaccia fornita - Aggiornamento stato giocatore:** viene richiesto al database di aggiornare lo stato del giocatore che ha richiesto di fare una sfida casuale.

**Interfaccia fornita - Richiesta giocatori disponibili:** richiede al database di inviargli tutti i giocatori che sono in stato di ricerca sfida online.

**Interfaccia fornita - Coppia username giocatori:** una volta identificato l'utente con cui far sfidare il giocatore che ha richiesto una sfida casuale, la coppia dei due username viene inviata alla componente gestore sessione di gioco per far iniziare la sfida.

##### 4.1.26 Gestore sessione di gioco

**Descrizione:** questo componente si occupa di avviare una sessione di sfida sincrona tra i due utenti forniti o dal gestore giocatori disponibili o dal gestore invio inviti di sfida, a seconda del tipo di sfida selezionato dai giocatori.

**Interfaccia richiesta - Coppia username giocatori:** riceve dalla componente gestore giocatori disponibili la coppia degli username dei due giocatori scelti casualmente per iniziare la sessione di sfida sincrona.

**Interfaccia richiesta - Coppia sfidante sfidato:** riceve dalla componente gestore



#### 4.1 Definizione dei componenti

---

invio inviti di sfida gli username dell'utente sfidato e dell'utente sfidante per iniziare la sessione di sfida sincrona.

**Interfaccia richiesta - Termina sfida:** una volta conclusa la sfida tra i due giocatori, riceve dalla componente pagina quiz la notifica di terminare la sessione di sfida sincrona tra i giocatori.

**Interfaccia fornita - Richiesta quiz sfida online:** una volta inizializzata la sessione di sfida, invia alla componente quiz maker una richiesta di generazione di quiz per la sfida online.

##### 4.1.27 Pagina info

**Descrizione:** questo componente riceve dal quiz maker le informazioni dei simboli di tutti gli alfabeti, dopodiché crea le tabelle dei vari alfabeti e fa visualizzare all'utente il compendio degli alfabeti.

**Interfaccia fornita - Compendio alfabeti:** mette a disposizione dell'utente la pagina del compendio degli alfabeti.

In questa sezione è presente il diagramma dei componenti dell'applicazione JGuesser.

