

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

1	Requisiti Funzionali	4
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
2	Requisiti Non Funzionali	23
2.1	Privacy	23
2.2	Sicurezza	23
2.3	Efficienza	24
2.4	Portabilità	24
2.5	Scalabilità	25
2.6	Usabilità	25
3	Analisi Del Contesto	26
3.1	Diagramma di contesto	26
3.2	Descrizione interazioni attori e sistemi esterni	26
3.2.1	Guest	26
3.2.2	Logged User	27
3.2.3	Premium User	28
3.2.4	Google Identity Service	28
3.2.5	Paypal	28
3.2.6	reCaptcha	28
3.2.7	Nodemailer	28
3.2.8	Nyckel	29
3.2.9	MongoDB	29

4	Analisi Dei Componenti	30
4.1	Definizione dei componenti	30
4.1.1	Pagina registrazione	30
4.1.2	Gestore registrazione	30
4.1.3	Validatore reCAPTCHA	32
4.1.4	Verificatore e-mail	32
4.1.5	Pagina login	32
4.1.6	Autenticazione	32
4.1.7	Autenticazione con Google	32
4.1.8	Pagina dati personali	32
4.1.9	Pagina recupero username e password	32
4.1.10	Gestore credenziali	32
4.1.11	Pagina pagamento	32
4.1.12	Gestore pagamenti	32
4.1.13	Aggiornamento privilegi profili	32
4.1.14	Pagina statistiche	32
4.1.15	Pagina classifica	32
4.1.16	Gestore statistiche	32
4.1.17	Pagina single-player	32
4.1.18	Quiz maker	32
4.1.19	Pagina quiz	32
4.1.20	Sistema interpretazione disegno	32
4.1.21	Pagina multiplayer	32
4.1.22	Sistema di ricerca giocatori	32
4.1.23	Gestore invio inviti di sfida	32
4.2	Diagramma dei componenti	32

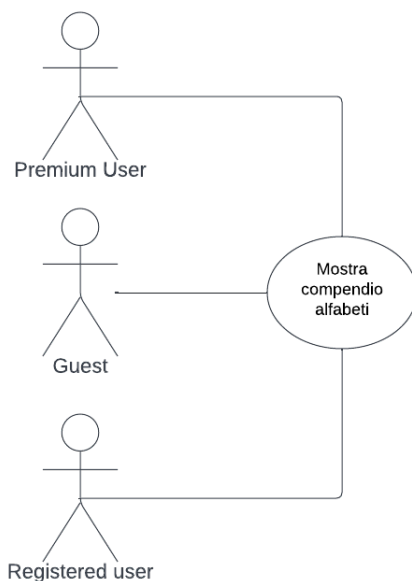
Scopo del documento

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



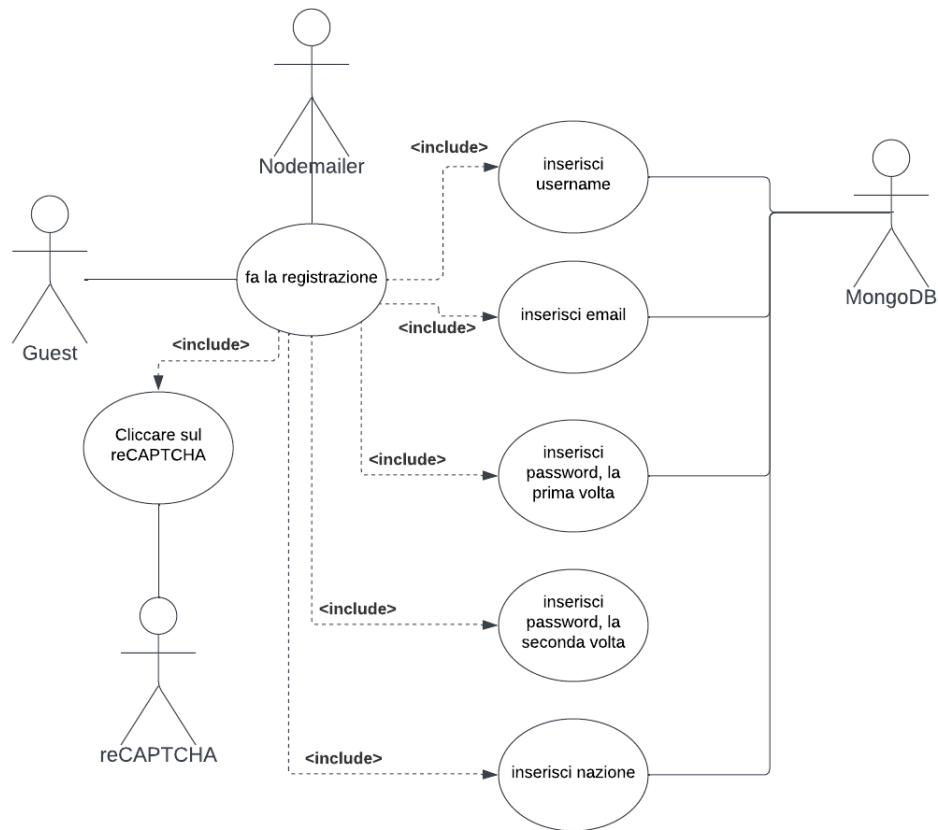
Titolo: mostrare compendio alfabeti

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

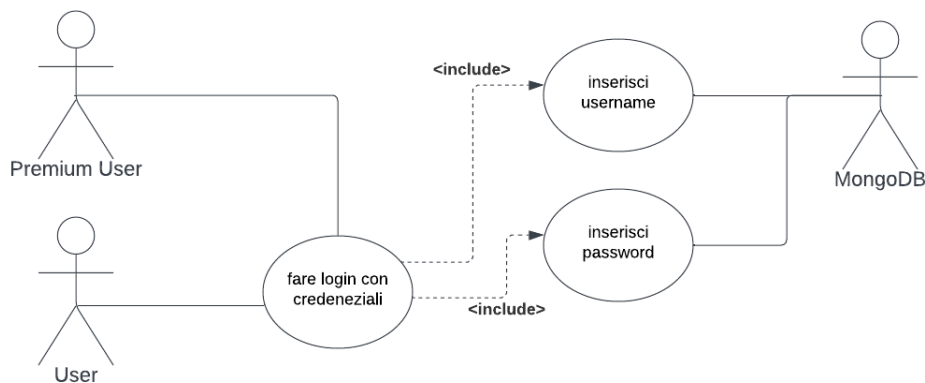
Eccezioni:

[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.

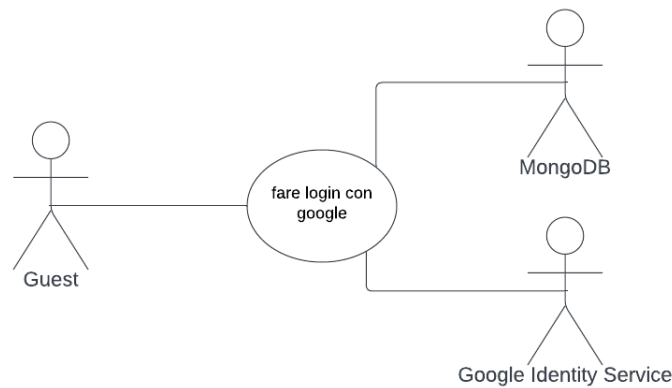
Descrizione:

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



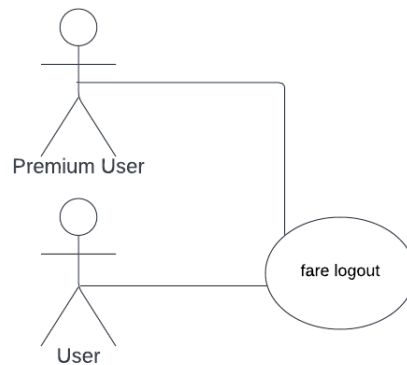
Titolo: login con google

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 li 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



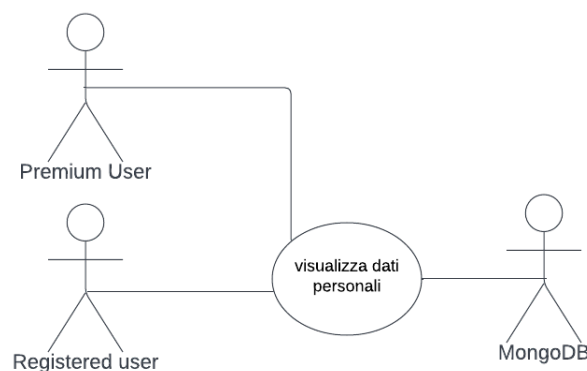
Titolo: logout dall'applicazione

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

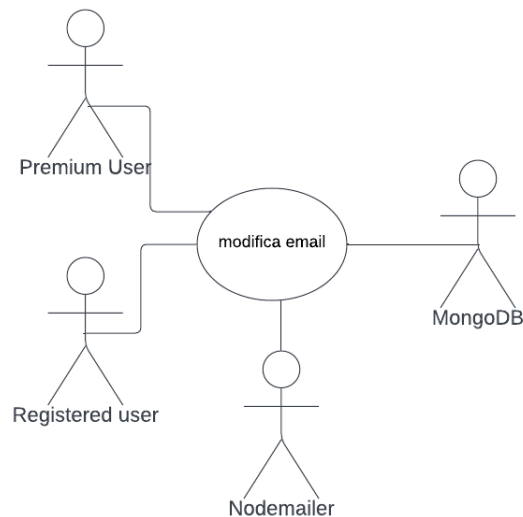
Riassunto: questo use-case mostra come un utente registrato o di livello superiore può visualizzare i propri dati personali inseriti in fase di registrazione.

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:**

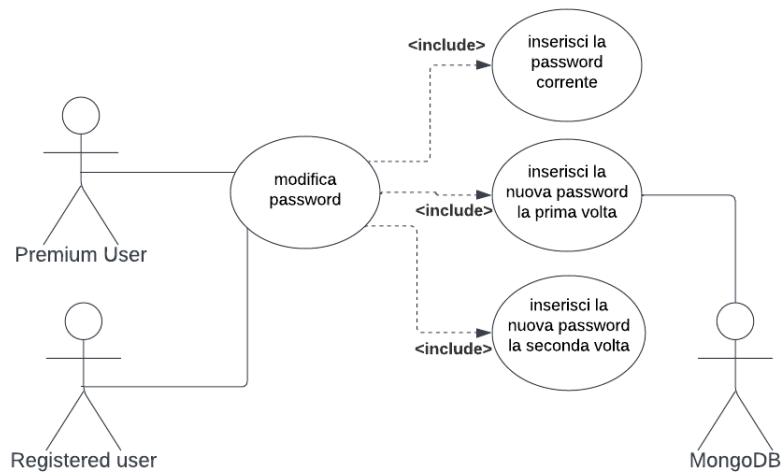
1. L'utente che si trova nella pagina in cui può visualizzare le proprie informazioni personali, decide di modificare la propria e-mail, che è presente in un apposito 'box'. Dopo aver modificato l'e-mail l'utente clicca sul pulsante 'confirm' per confermare la nuova e-mail.
2. L'applicazione interagisce 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

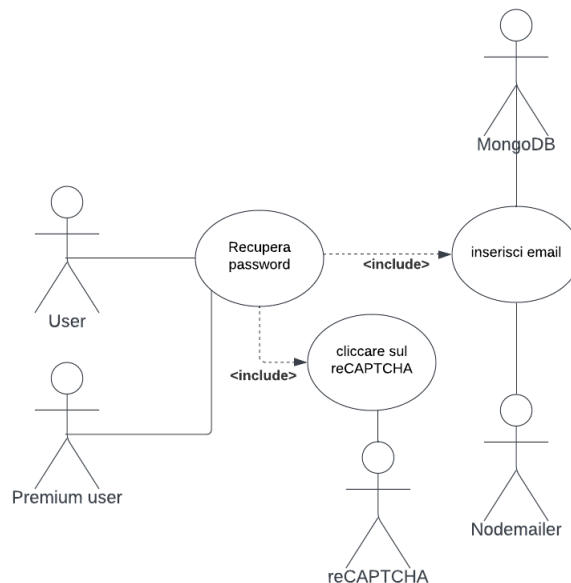
Descrizione:

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

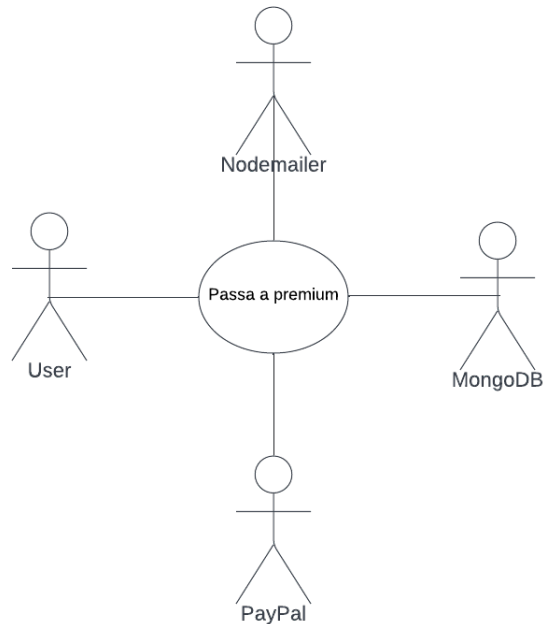
Descrizione:

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



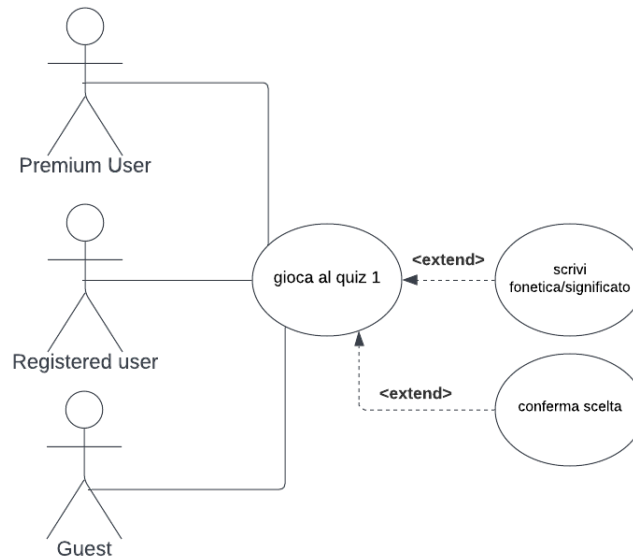
Titolo: acquisto versione premium

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



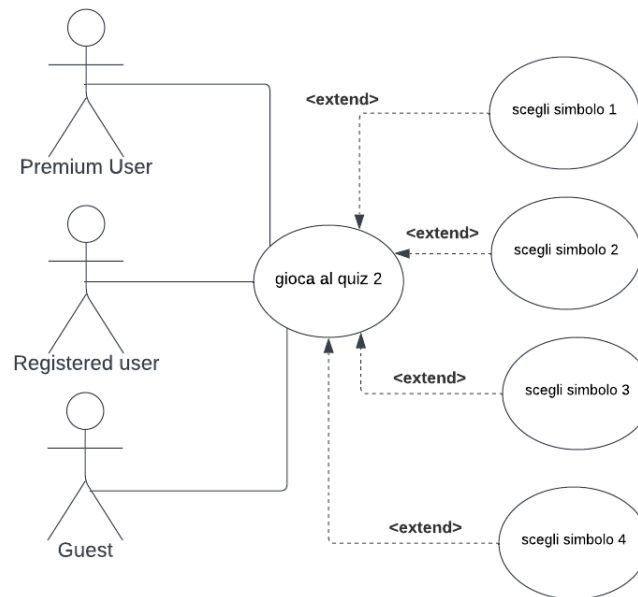
Titolo: Quiz di tipo 1

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



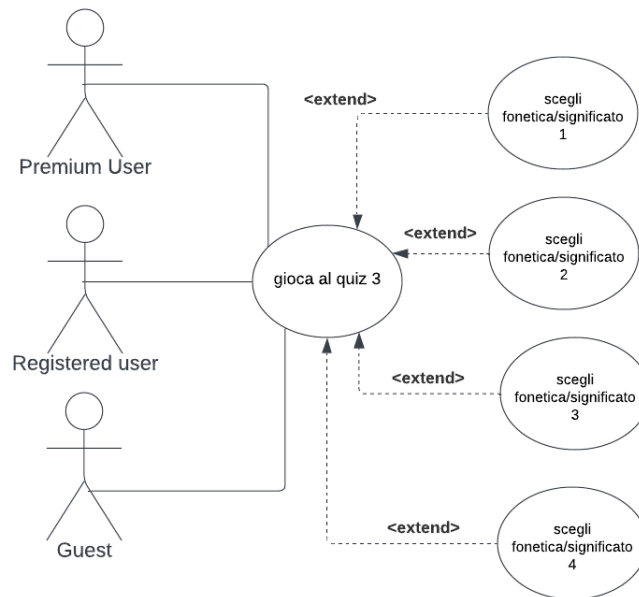
Titolo: Quiz di tipo 2

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



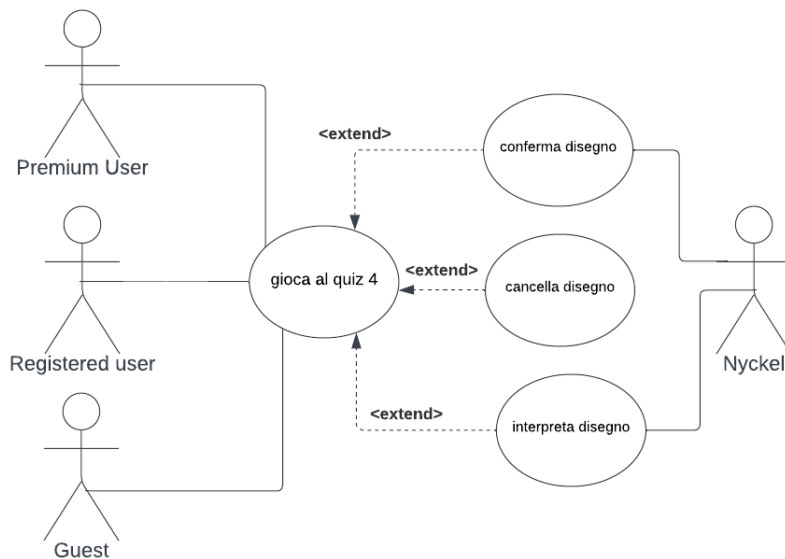
Titolo: Quiz di tipo 3

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



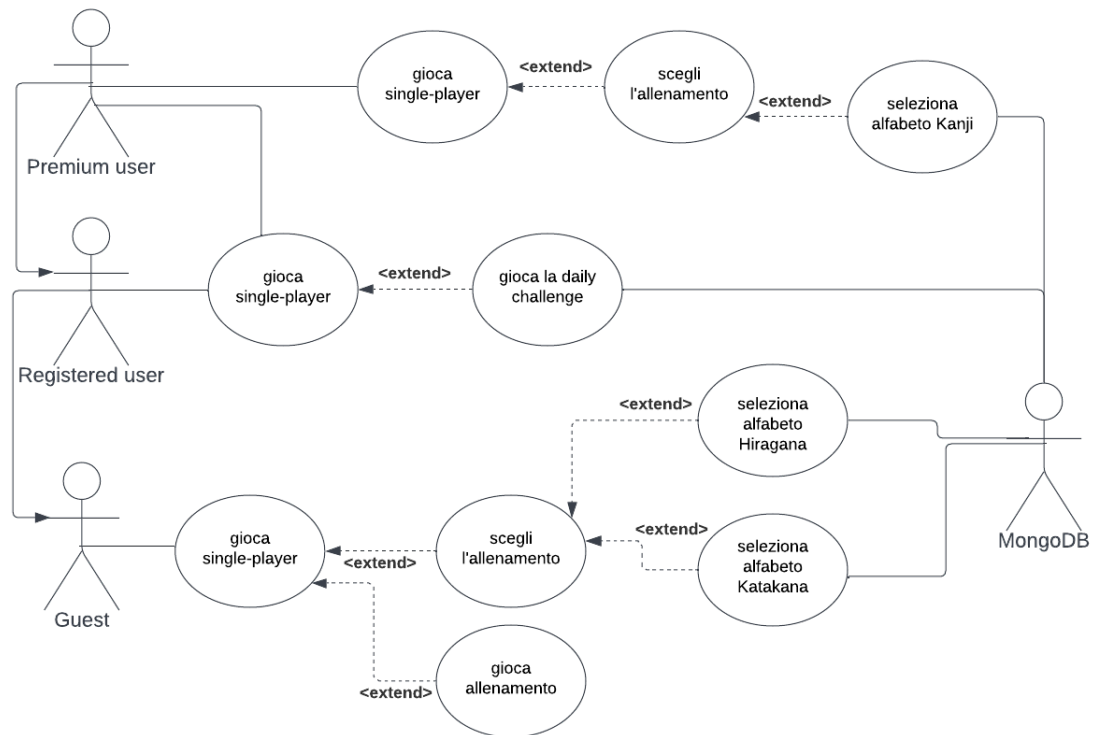
Titolo: Quiz di tipo 4

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

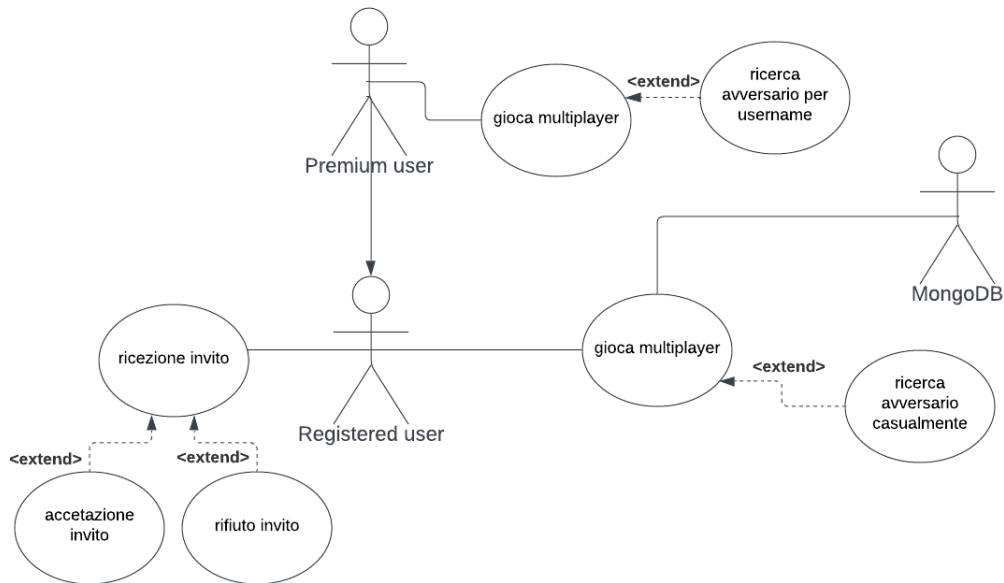
Riassunto: questo gruppo di use-case spiega come come l'utente guest o di livello superiore può giocare in modalità single-player.

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

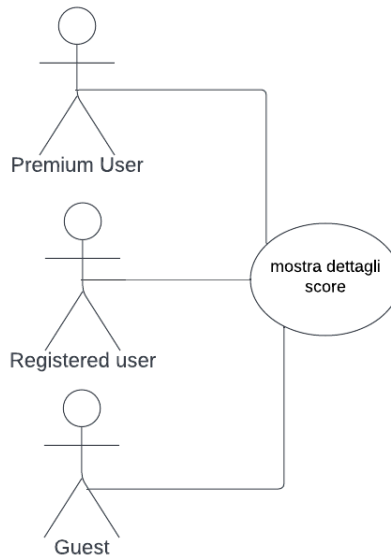
Riassunto: questo gruppo di use-case spiega come come l'utente registrato o di livello superiore può giocare in modalità multiplayer.

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



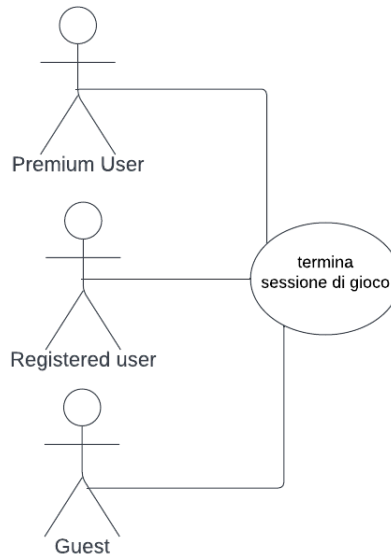
Titolo: mostra dettagli score

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



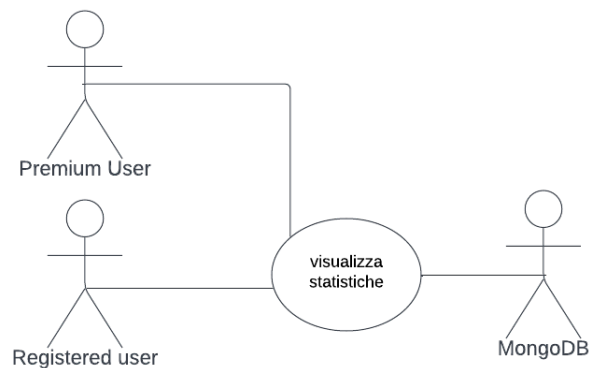
Titolo: termina sessione di gioco

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

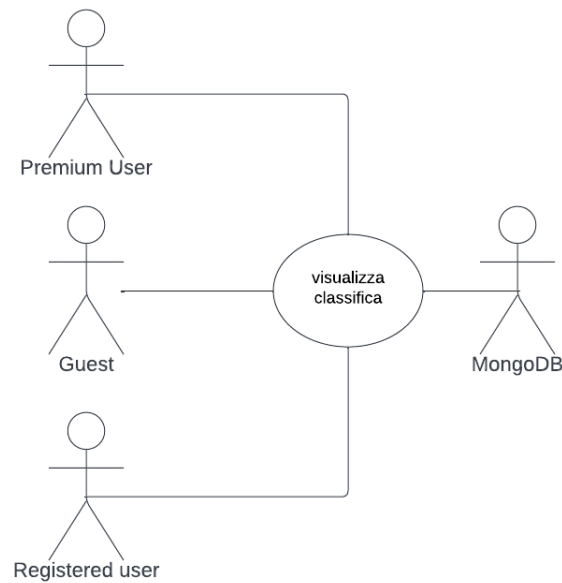


Titolo: visualizza statistiche

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

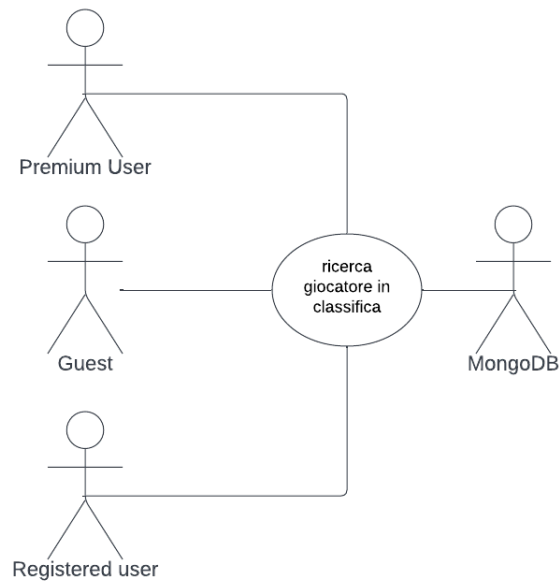
Descrizione:

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 quanto riguarda i requisiti funzionali e le tabelle strutturate per quanto riguarda i requisiti non funzionali.

In questo capitolo e in quelli successivi ci concentreremo invece sulla fase di design e delle interazione del sistema JGuesser con gli altri sistemi esterni. Questo verrà fatto 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 o utilizzano il sistema e in cui compaiono tutti i flussi d'informazione che questi sistemi si scambiano.

I sistemi esterni si possono dividere in:

- sistemi superiori: sistemi che utilizzano il nostro sistema;
- sistemi subordinati: sistemi utilizzati dal nostro sistema;
- sistemi paritari: sistemi che hanno un rapporto paritario con il nostro.
- 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.

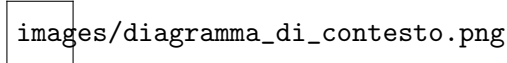


Figura 1: Diagramma di contesto dell'applicazione JGuesser

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 cosa può fare l'utente che non è loggato, all'interno del sistema. L'utente guest riceve una serie di dati in ingresso dal sistema JGuesser:

- Quiz Hiragana/Katakana:
- Compendio alfabeti:
- Classifica:
- Statistiche giocatore cercato:
- Risultato quiz:
- Disegno compreso dal sistema:

- Conferma di login con google:

e fornisce in uscita e quindi all'applicativo una serie di dati:

- E-mail per recupero password:
- Credenziali registrazione:
- Risposta quiz:
- Username cercato (ranking):
- Scelta alfabeto Hiragana/Katakana:
- Disegno quiz:
- One time code di google:
- reCAPTCHA token:

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 anche l'utente guest può fare 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 ingresso diretti verso l'utente loggato:

- Statistiche personali:
- Sfida giornaliera:
- Arrivo notifica di sfida:
- Quiz partita online:

e i seguenti flussi di dati in uscita diretti verso il sistema JGuesser:

- Credenziali log-in:
- Informazioni pagamento upgrade:
- Credenziali modificate:
- Accetto/Rifiuto sfida:
- Risposte sfida giornaliera:
- Ricerca sfida casuale online:

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 ingresso:

- Risposta giocatore sfidato:
- Quiz Kanji:

e i seguenti flussi di dati in uscita:

- Username da sfidare:
- Scelta alfabeto Kanji:

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 e al server JGuesser di confermarlo.

3.2.5 Paypal

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 robot. In particolare questo tipo di sistema sarà coinvolto quando l'utente dovrà registrarsi e quando l'utente vorrà resettare la propria password, perchè se l'è dimenticata. 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 l'utente inserirà. 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 all'interno. Vedere RF1.9.
- un utente ha effettuata 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

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 nel 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 che forniscono o che richiedono.

4.1.1 Pagina registrazione

Descrizione: questo componente si occupa di permettere all'utente di inserire le diverse informazioni per registrarsi. 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 chiave sito: cliccando sul reCAPTCHA presente nella pagina di registrazione, l'utente genera una chiave, che dovrà essere validato dal sistema esterno reCAPTCHA.

Interfaccia fornita – reCAPTCHA chiave sito: la chiave del reCAPTCHA deve essere inviata alla componente validatore reCAPTCHA, che si occuperà di validarla.

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.

4.1.2 Gestore registrazione

Motivazione: 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 verificare

4.1 Definizione dei componenti

e attuare la registrazione.

Descrizione:

Interfaccia fornita – Check registrazione:

Interfaccia fornita – e-mail:

Interfaccia fornita – Richiesta inserimento info registrazione:

Interfaccia fornita – Richiesta invio e-mail conferma registrazione:

Interfaccia richiesta – Info registrazione:

Interfaccia richiesta – Check e-mail:

Interfaccia richiesta – Check inserimento dati:

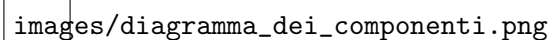
Interfaccia richiesta – Credenziali google per registrazione:

Interfaccia richiesta – Check validazione reCAPTCHA:

4.2 Diagramma dei componenti

- 4.1.3 Validatore reCAPTCHA
- 4.1.4 Verificatore e-mail
- 4.1.5 Pagina login
- 4.1.6 Autenticazione
- 4.1.7 Autenticazione con Google
- 4.1.8 Pagina dati personali
- 4.1.9 Pagina recupero username e password
- 4.1.10 Gestore credenziali
- 4.1.11 Pagina pagamento
- 4.1.12 Gestore pagamenti
- 4.1.13 Aggiornamento privilegi profili
- 4.1.14 Pagina statistiche
- 4.1.15 Pagina classifica
- 4.1.16 Gestore statistiche
- 4.1.17 Pagina single-player
- 4.1.18 Quiz maker
- 4.1.19 Pagina quiz
- 4.1.20 Sistema interpretazione disegno
- 4.1.21 Pagina multiplayer
- 4.1.22 Sistema di ricerca giocatori
- 4.1.23 Gestore invio inviti di sfida
- 4.2 Diagramma dei componenti

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



images/diagramma_dei_componenti.png

Figura 2: Diagramma dei componenti dell'applicazione JGuesser