**Hello Word**

**User Requirements Document**

**DIBRIS – Università di Genova. Scuola Politecnica, Corso di Ingegneria del Software 80154**

**DATA : 14/03/2018**

**VERSION: 1.0**

**Autori**

Massimo Narizzano

studenti Corso di SE

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versione** | **Data** | **Autore** | **Note** |
| 1.0 | 14/03/2018 | MN | Inizio |
|  |  |  |  |
|  |  |  |  |

[**​1​ Introduzione**](#_iha900e9ujmq) **2**

[​1.1​ Scopo del Documento](#_txu7jyxtu5yq) 2

[​1.2​ Ambito Applicativo del Documento](#_xyh9a6dur4gk) 2

[​1.3​ Definizioni e Acronimi](#_g2f3usokfwxn) 2

[​1.4​ Bibliografia](#_6b7hsbyzqros) 3

[​1.5​ Overview del documento](#_zds9ob1i8pkz) 3

# ​1​ Introduzione

## ​1.1​ Scopo del Documento

Questo documento è un esercizio proposto in aula durante il corso di Software Engineering(SE). Il documento si promette di descrivere le funzionalità di un tool dedicato alla pratica dell'insegnamento della lingua inglese.

## ​1.2​ Ambito Applicativo del Documento

Questo URD è un esercizio pratico svolto durante le ore di esercitazioni del corso di SE.

## ​1.3​ Definizioni e Acronimi

|  |  |
| --- | --- |
| **Acronimo-Nome** | **Definizione** |
| SE | Software Engineering |
| MN | Massimo Narizzano |
| HW | Hello Word |
|  |  |
|  |  |

## ​1.4​ Bibliografia

## ​1.5​ Overview del documento

Il documento è organizzato come segue: la prima parte del documento contiene una breve descrizione del problema che si vuole affrontare e una possibile soluzione; la seconda parte elenca le funzionalità aspettate dal sistema e la loro priorità di implementazione.

# ​2​ Descrizione Generale del Sistema

HelloWord è un sistema pensato per gestire l’apprendimento della lingua inglese per quelle classi che utilizzano il metodo chiamato Lexical Approach. Il sistema serve a far pratica agli studenti di una classe. In generale il sistema prepare un test, ovvero un insieme di domande, e le somministra all’utente. L’utente deve rispondere alle domande, e il sistema registra le risposte, confrontandole con la risposta corretta. Alla fine del test, il sistema deve dare un punteggio in base al numero di risposte giuste/errate. Il sistema dovrebbe anche tener traccia degli errori dell’utente, permettendo così di preparare test mirati a risolvere le incertezze dell’utente.

## ​2.1​Contesto e Motivazioni

HelloWord è un sistema pensato per gestire l’apprendimento della lingua inglese per quelle classi che utilizzano il metodo chiamato Lexical Approach. Questa metodologia di apprendimento si basa sullo studio di una wordlist", ovvero un vocabolario di parole che tutti gli utenti registrano su un quaderno. Un insegnante *gestisce* la wordlist, ovvero sceglie le parole che gli studenti devono trascrivere, trascrivendo anche la loro traduzione. Gli studenti a casa devono studiare queste parole, ed hanno come obiettivo quello di imparare a leggerle, scriverle e riconoscere il suono ad esse associato. L’insegnante, periodicamente, esegue delle verifiche in classe tramite test, dove viene presentata una serie di parole che lo studente deve tradurre. Questo tipo di metodologia però presenta diversi problemi: (i) richiede un aggiornamento della wordlist, che spesso gli studenti non riescono a mantenere perchè lenti nella scrittura, oppure semplicemente assenti; (ii) richiede un’esercitazione quasi quotidiana, e conseguentemente da parte del genitore è richiesto un impegno di tempo importante, dal preparare il test alla correzione; (iii) richiede da parte del docente un costante test degli studenti, questo per misurare effettivamente il grado di conoscenza degli studenti.

Necessariamente, vista la grande quantità di tempo necessaria per poterli fare e correggere, in questo approccio gli studenti faranno molti meno test di allenamento, e il docente farà molti meno check per verificare il grado di apprendimento dello studente. Come conseguenza la qualità della preparazione dello studente sarà meno alta e ne risentirà la qualità dell’apprendimento della lingua inglese.

## 

## 2.2​ Obiettivo del progetto

L’obiettivo del progetto è quello di fornire uno strumento informatico che organizzi gli studenti in classi e che

1. Permetta al docente di aggiornare la wordlist e notificare tutti gli studenti di questo aggiornamento. In questo modo lo studente ogni volta che accede al sistema può scaricarsi le nuove parole, aggiornando la wordlist, eliminando il problema della perdita dei vocaboli.
2. Permetta allo studente di creare un test in modo automatico, scegliendo diverse domande da un pool di domande possibili, e i vocaboli tra quelli della wordlist.
3. Permetta ad uno studente di eseguire un test da lui creato, e di registrarne le risposte, confrontandole con le risposte corrette.
4. Registri le risposte dello studente calcolando le statistiche come ad esempio le risposte sbagliate, le risposte giuste, le risposte quasi giuste, etc...
5. Permetta allo studente di creare test mirati utilizzando i dati statistici raccolti

Il sistema deve poter funzionare sia con la connessione attiva sia senza connessione, sia su un desktop che su un tablet.

## 2.3Utenti

1. Studente: utente base, appartiene ad una classe, può creare test e svolgere dei test.
2. Docente: gestisce una classe. Il suo compito primario è quello di aggiornare la wordlist
3. Genitore: Può controllare i progressi del figlio e può creare test personalizzati per il figlio. Inoltre ogni genitore può registrare una nuova login studente per ogni figlio, che poi può essere usata da lui per poter accedere all’applicazione. Il Genitore è responsabile dell’identità del figlio.

# 

# 

# 

# 3​ User Requirement

In questa sezione descriveremo i requisiti dello strumento chiamato HelloWord. In particolare ad ogni requisito daremo un id univoco e una priorità seguendo la seguente tabella:

|  |  |
| --- | --- |
| M | Mandatory. Requisito Obbligatorio. |
| D | Desiderable. Requisito che dovrebbe essere inserito nel sistema. |
| O | Optional. Una funzionalità marcata come optional può essere inserita nel sistema, a discrezione del manager del progetto. Ad esempio se il tempo di sviluppo è minore di quello previsto oppure se il costo di implementazione non è troppo alto. |
| E | future Enhancement. Questo requisito viene lasciato per la prossima release. |

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrizione** | **Priorità** |
| **1** | Registrazione. Al lancio il sistema deve permette di registrarsi tramite email. Due sono i ruoli ammessi durante la registrazione: Genitore o Docente. |  |
| **2** | Accesso: |  |
| **3** | Creare una login studente: un genitore può creare un nuovo studente compilando i campi correttamente. Lo studente creato deve avere una relazione di parentela con il genitore che lo ha creato. |  |
| **4** | Gestione Studente: Un genitore può GESTIRE un utente studente del sistema che lui stesso ha creato. In particolare oltre al profilo, un genitore deve gestire l’appartenenza ad una classe, mandando richieste oppure accettando richieste. Un genitore può anche rimuovere uno studente da una classe. |  |
| **5** | Visionare i risultati dei test degli studenti: Un Genitore può vedere tutti i test portati a compimento dagli studenti da lui creati. |  |
| **6** | Creare una classe. Un docente può creare una classe compilando correttamente i campi. Il creatore di una classe ne diventa amministratore. |  |
| **7** | Gestione di una classe. Un docente può gestire una classe aggiungendo, cancellando studenti, oppure mandando inviti a studenti/genitori. |  |
| **8** | Creare un Test per una classe. Un docente può creare un test e associarlo ad una classe. In particolare questo può essere usato per test iniziali oppure test intermedi per capire il grado di preparazione degli studenti. Il test può essere obbligatorio (come il test di ingresso) oppure opzionale. Un test obbligatorio blocca le funzionalità di uno studente fino a che il test non viene eseguito. |  |
| **9** | Statistiche della classe. Il sistema deve permettere di vedere le statistiche di una classe, come ad esempio i progressi dal test di ingresso, oppure il grado di conoscenza del vocabolario degli studenti, quali sono le parole più difficili…. |  |
| **10** | Memorizzazione in un database. Il sistema deve associare ad ogni parola inglese, la sua traduzione, un’immagine evocativa, il suono (speech) della parola. Tutte queste informazioni vanno registrate in un database. |  |
| **11** | Tipo di dati da memorizzare. Il sistema per ogni parola può memorizzare anche altri tipi di dato. Ad esempio i sinonimi, i contrari, le parole simili, i false friends, e altro. |  |
| **12** | Creare una Wordlist. Il sistema deve permettere di creare una nuova wordlist e di associarla ad una classe. Una classe in pratica ha un’unica wordlist, ma teoricamente ne può avere più di una. Le wordlist appena create saranno vuote. |  |
| **13** | Aggiornare la wordlist. La wordlist può essere aggiornata aggiungendo o eliminando parole. Le parole da aggiungere vanno scelte da una lista predefinita inserita nel sistema, ovvero l’intero vocabolario. |  |
| **14** | Creare un test. Chiunque può creare un test. Un test comprende un numero di domande stabilito a priori, diverse tipologie di domande (aperte, sonore…). Un utente crea un test definendo il numero di domande, per ogni tipo di domanda la quantità massima, e può scegliere se il test è personalizzato (alta probabilità di inserire parole meno conosciute, siano esse state poco utilizzate oppure sbagliate tante volte) o non personalizzato (tutte le parole hanno la stessa probabilità di essere scelte). Un test può essere un test di ripasso, oppure un test di learning. Un test di ripasso è un test in cui le parole associate alle domande possono essere scelte da tutta la wordlist, mentre nei test di learning le parole vengono scelte tra le ultime parole inserite (inserite nell’ultimo mese, semestre, settimana ….). E’ possibile fare un test che comprenda le due tipologie di test insieme. |  |
| **15** | Aggiornamento.Al lancio il sistema richiede il login, e se il login va a buon fine, aggiorna la wordlist se c’è connessione a internet. |  |
| **16** | Il sistema deve funzionare su un computer desktop. |  |
| **17** | Il sistema deve funzionare su un dispositivo mobile (tablet, telefono). |  |
| **18** | Quando il sistema aggiorna la wordlist, le nuove parole introdotte nel vocabolario sono marcate come nuove. |  |
| **19** | Learning delle nuove parole. Per le parole nuove il sistema deve prevedere una fase di learning guidata. Ad esempio mostrando prima la figura, poi scrivendo la parola associata alla figura (in inglese), e infine riprodurre lo speech associato alla parola. Questo tipo learning delle nuove parole può essere ripetuto per tutte le nuove parole. Il sistema alla fine della fase di inserimento delle nuove parole costruisce automaticamente un test per verificare che le nuove parole siano state apprese dallo studente. |  |
| **20** |  |  |
|  |  |  |
|  |  |  |