

SWEG

Software Engineering Group

swegsoftware@gmail.com

Specifica Tecnica

Informazioni sul documento

Redattori:

Verificatori:

Approvazione:

Destinatari: T. Vardanega R. Cardin Zero12

Versione: 1.0.0

Registro dei Cambiamenti - Changelog

<i>Versione</i>	<i>Data</i>	<i>Autore</i>	<i>Verificatore</i>	<i>Dettaglio</i>
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
	XXXX-XX-XX			
0.0.3	XXXX-XX-XX			
0.0.2	2023-04-10	Gabriel Rovesti	Andrea Meneghello	Scrittura sezione 2 e sottosezioni
0.0.1	2023-03-20	Davide Milan	Davide Sgrazzutti	Inizializzazione documento e creazione indici e struttura

Sommario

Sommario	2
Elenco delle immagini	2
Elenco delle tabelle	2
Elenco dei grafici	2
1 Introduzione	3
1.1 Scopo del documento	3
1.2 Scopo del prodotto	3
1.3 Glossario	3
1.4 Maturità e miglioramenti	3
1.5 Riferimenti	3
1.5.1 Riferimenti normativi	3
1.5.2 Riferimenti informativi	3
2 Tecnologie	3
2.1 Tecnologie per la codifica	4
2.2 Tecnologie per l'analisi del codice	5
3 Architettura	5
3.1 Introduzione	5
3.2 Pattern Architetturale	5
3.3 Design Patterns	5
4 Documentazione delle API	5
5 Punti di estensione	5

Elenco delle immagini

Elenco delle tabelle

Tabella 1 - Tabella di descrizione delle tecnologie

Tabella 2 - Tabella di descrizione degli strumenti di test

Elenco dei grafici

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è descrivere e motivare tutte le scelte architetturelle che sono state fatte nella fase di progettazione e codifica del prodotto.

Vengono quindi riportati i diagrammi dei React Components^G e dei package per descrivere le scelte di design pattern utilizzate per realizzare l'architettura finale del prodotto.

È poi presente una sezione dedicata ai requisiti che il gruppo è riuscito a soddisfare, così da fornire un'ampia visione sullo stato di avanzamento del lavoro.

1.2 Scopo del prodotto

Oggigiorno molti applicativi moderni, dalle pagine web alle app native per mobile, necessitano di essere multi-lingua in modo da essere disponibili in un mercato internazionale.

È quindi necessario poter realizzare e soprattutto gestire facilmente le traduzioni dei testi presenti nel prodotto in maniera sistematica e centralizzata.

Lo scopo del prodotto è quindi quello di creare una piattaforma web multi tenant^G, accessibile agli utenti traduttori di un'azienda, su cui si possa svolgere questo lavoro di scrittura di testi e traduzioni, e sviluppare delle API che permettano agli sviluppatori di implementare la visualizzazione dei testi nella lingua selezionata dall'utente del proprio prodotto.

1.3 Glossario

Al fine di evitare incomprensioni relative alla terminologia usata all'interno del documento, viene fornito un Glossario nel file apposito, tale da non avere terminologie ambigue nell'attività progettuale individuata e dandone una definizione precisa. Ogni termine avrà nel documento una lettera G come apice, per meglio evidenziare la loro appartenenza al documento indicato.

1.4 Maturità e miglioramenti

Il presente documento è redatto con un approccio incrementale, al fine di poter implementare facilmente cambiamenti nel corso del tempo a seconda di esigenze concordate bilateralmente tra membri del gruppo e proponente. Pertanto, non può essere considerato definitivo e completo in questa versione.

1.5 Riferimenti

1.5.1 Riferimenti normativi

1.5.2 Riferimenti informativi

2 Tecnologie

In questa sezione viene fornita una panoramica generale delle tecnologie utilizzate per la realizzazione del prodotto in questione. Vengono infatti descritte le tecnologie, gli strumenti e le librerie necessari per lo sviluppo, il test e la distribuzione del prodotto. In particolare, verranno trattate le tecnologie impiegate per la realizzazione del front-end e del back-end, nonché per la gestione del database e l'integrazione con altri servizi.

2.1 Tecnologie per la codifica

<i>Tecnologia</i>	<i>Descrizione</i>	<i>Versione</i>
<i>Linguaggi</i>		
HTML	Linguaggio di annotazione (markup) utilizzato per impostare la struttura delle singole pagine e definire gli elementi dell'interfaccia	5
CSS	Linguaggio utilizzato per la formattazione e la gestione dello stile degli elementi HTML	3
JavaScript	Linguaggio utilizzato per la gestione di eventi invocati dall'utente,	ECMAScript 2021
TypeScript ^G	Superset di JavaScript per utilizzare tipizzazione	5.0.x
<i>Librerie e framework^G</i>		
React ^G	Libreria grafica per facilitare lo sviluppo front-end gestendo modularmente le componenti grafiche, permettendo performance buone grazie all'efficacia della sua renderizzazione	18.0.x
Material UI ^G	Framework di componenti React preconfezionati per la creazione di interfacce utente gradevoli, funzionali e personalizzabili	4.1.x
<i>Strumenti e servizi</i>		
Node.js ^G	Ambiente di runtime open-source per l'esecuzione di codice JavaScript lato server tramite appositi script	19.0.x
NPM ^G	Gestore di pacchetti per il linguaggio JavaScript e l'ambiente di esecuzione Node.js	9.6.x
AWS Cognito ^G	Servizio di gestione delle identità ed autenticazione basato sui ruoli	2023-16-02
AWS DynamoDB ^G	Servizio di database non relazionale gestito in modo scalabile	2019-11-21

AWS Lambda ^G	Servizio di elaborazione serverless ^G di codice senza dover configurare/scalare l'infrastruttura server, eseguendo il codice in modo scalabile	2023-03-16
AWS API Gateway ^G	Servizio di gestione delle API ^G che permette di creare, pubblicare e proteggere e le stesse in modo sicuro, concentrandosi sulla logica di business	2023-04-06
Git ^G	Sistema di controllo versione distribuito utilizzato per la gestione del codice sorgente dal parte del gruppo di progetto	2.4.x

Tabella 1 - Tabella di descrizione delle tecnologie

2.2 Tecnologie per l'analisi del codice

<i>Tecnologia</i>	<i>Descrizione</i>	<i>Versione</i>
<i>Analisi statica</i>		
ESLint	Strumento JavaScript che aiuta a individuare gli errori di codice e le pratiche non ottimali	8.38.x
Prettier	Strumento di formattazione del codice che aiuta a mantenere uno stile di codifica coerente e leggibile	3.0.x
<i>Analisi dinamica</i>		
React Testing Library	Libreria di test integrata nativamente che consente di testare il comportamento dei componenti ^G React da una prospettiva degli utenti finali.	14.0.x
GitHub Actions	Servizio di CI/CD ^G per automatizzare il processo di build ^G , test e deploy ^G del progetto software	/

Tabella 2 - Tabella di descrizione degli strumenti di test

3 Architettura

3.1 Introduzione

ReactJS non impone una specifica struttura architetturale o un design architetturale specifico per realizzare un prodotto.

Per il Frontend si è quindi optato per non utilizzare un design pattern architetturale vero e proprio ma è stato deciso di utilizzare una struttura della directory ferrea per gestire i componenti e le funzioni di utility utilizzate nel software; sono stati inoltre definiti degli stili/design pattern di programmazione per realizzare i componenti lato Frontend.

3.2 Pattern Architetturale

3.3 Design Patterns

Per quanto riguarda i componenti sono stati utilizzati i seguenti pattern:
controlled components

props
presentational and container component patterns
hooks

4 Documentazione delle API

5 Punti di estensione