*swegsoftware@gmail.com*

Specifica Tecnica

Informazioni sul documento

Redattori: Davide M. Gabriel R.

Verificatori: Andrea M. Davide S.

Approvazione:

Destinatari: T. Vardanega R. Cardin Zero12

Versione: 1.0.0

Uso: Esterno

Registro dei Cambiamenti - Changelog

| *Versione* | *Data* | *Autore* | *Verificatore* | *Dettaglio* |
| --- | --- | --- | --- | --- |
| 1.0.0 | xxxx-xx-xx |  |  |  |
| 0.9.0 | xxxx-xx-xx |  |  |  |
| 0.8.0 | xxxx-xx-xx |  |  |  |
| 0.7.0 | xxxx-xx-xx |  |  |  |
| 0.6.0 | xxxx-xx-xx |  |  |  |
| 0.5.0 | xxxx-xx-xx |  |  |  |
| 0.4.0 | 2023-04-19 | Marco Bernardi | Davide Sgrazzutti | Scrittura sezione 4 |
| 0.3.0 | 2023-04-15 | Davide Milan | Andrea Meneghello | Creazione indici sezioni 3,4 e 5 e prima scrittura sezione 3 |
| 0.2.0 | 2023-04-10 | Gabriel Rovesti | Andrea Meneghello | Scrittura sezione 2 e sottosezioni |
| 0.1.0 | 2023-03-20 | Davide Milan | Davide Sgrazzutti | Inizializzazione documento e creazione indici e struttura |

# 

# Sommario

[**Sommario 2**](#_heading=h.l4bpjcpynuc)

[Elenco delle immagini 2](#_heading=h.72azuiwwxoe2)

[Elenco delle tabelle 2](#_heading=h.bkqthipu5hlb)

[Elenco dei grafici 3](#_heading=h.kheyc37k7hbt)

[**1 Introduzione 4**](#_heading=h.6nupdkus4pit)

[1.1 Scopo del documento 4](#_heading=h.khetm49b31ay)

[1.2 Scopo del prodotto 4](#_heading=h.5x4ck03wcvls)

[1.3 Glossario 4](#_heading=h.7t5g28vwsv5s)

[1.4 Maturità e miglioramenti 4](#_heading=h.mrcxk6l7pag4)

[1.5 Riferimenti 5](#_heading=h.vdfv13qc6so3)

[1.5.1 Riferimenti normativi 5](#_heading=h.vkpl4vutxn5p)

[1.5.2 Riferimenti informativi 5](#_heading=h.g5mk7e2q8xse)

[**2 Tecnologie 5**](#_heading=h.nmhe5s6t2x5p)

[2.1 Tecnologie per la codifica 5](#_heading=h.bodm3v2asc8x)

[2.2 Tecnologie per l’analisi del codice 6](#_heading=h.qybjyjm2bles)

[**3 Architettura Frontend 6**](#_heading=h.feni4ihkng5j)

[3.1 Introduzione 6](#_heading=h.sbj9msd6n6w9)

[3.2 Diagrammi delle Classi 7](#_heading=h.ngxtnvi38vbu)

[3.2.1 User 7](#_heading=h.r6m8rzjqu0l6)

[3.2.2 Admin 7](#_heading=h.h8ew8zpbn0in)

[3.2.3 SuperAdmin 7](#_heading=h.6r7c1asstzkd)

[3.3 Design Patterns 7](#_heading=h.g49ftjcxsx1f)

[Compound Components (per tutti gli elementi lista, quindi un padre che contiene i listitem) 7](#_heading=h.9fd6f9cbo8el)

[**4 Architettura Backend 7**](#_heading=h.gtb2vfqwhgcc)

[4.1 Introduzione 7](#_heading=h.7bi8bc95g0zy)

[4.2 Diagrammi delle Classi 7](#_heading=h.t5ad0w5z6mah)

[4.3 Design Patterns 7](#_heading=h.wlrr2d6fqc5r)

## Elenco delle immagini

## Elenco delle tabelle

[Tabella 1 - Tabella di descrizione delle tecnologie](#bookmark=id.3aqdvq924s2h)

[Tabella 2 - Tabella di descrizione degli strumenti di test](#bookmark=id.edb9rzlvy9rg)

## Elenco dei grafici

# 

# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo di questo documento è descrivere e motivare tutte le scelte architetturali che sono state fatte nella fase di progettazione e codifica del prodotto.  
Vengono quindi riportati i diagrammi dei React ComponentsG 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 tenantG , 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

| *Tecnologia* | *Descrizione* | *Versione* |
| --- | --- | --- |
| *Linguaggi* | | |
| 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 |
| TypeScriptG | Superset di JavaScript per utilizzare tipizzazione | 5.0.x |
| *Librerie e frameworkG* | | |
| ReactG | 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 UIG | Framework di componenti React preconfezionati per la creazione di interfacce utente gradevoli, funzionali e personalizzabili | 4.1.x |
| *Strumenti e servizi* | | |
| Node.jsG | Ambiente di runtime open-source per l'esecuzione di codice JavaScript lato server tramite appositi script | 19.0.x |
| NPMG | Gestore di pacchetti per il linguaggio JavaScript e l'ambiente di esecuzione Node.js | 9.6.x |
| AWS CognitoG | Servizio di gestione delle identità ed autenticazione basato sui ruoli | 2023-16-02 |
| AWS DynamoDBG | Servizio di database non relazionale gestito in modo scalabile | 2019-11-21 |
| AWS LambdaG | Servizio di elaborazione serverlessG di codice senza dover configurare/scalare l’infrastruttura server, eseguendo il codice in modo scalabile | 2023-03-16 |
| AWS API GatewayG | Servizio di gestione delle APIG che permette di creare, pubblicare e proteggere e le stesse in modo sicuro, concentrandosi sulla logica di business | 2023-04-06 |
| GitG | 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

| *Tecnologia* | *Descrizione* | *Versione* |
| --- | --- | --- |
| *Analisi statica* | | |
| 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 |
| *Analisi dinamica* | | |
| React Testing Library | Libreria di test integrata nativamente che consente di testare il comportamento dei componentiG React da una prospettiva degli utenti finali. | 14.0.x |
| GitHub Actions | Servizio di CI/CDG per automatizzare il processo di buildG, test e deployG del progetto software | / |

Tabella 2 - Tabella di descrizione degli strumenti di test

# 3 Architettura Frontend

## 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 Diagrammi delle Classi

### 3.2.1 User

### 3.2.2 Admin

### 3.2.3 SuperAdmin

## 3.3 Design Patterns

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

props  
presentational and container component patterns

Hooks Pattern

Provider Pattern (per il componente di autenticazione)

Singleton

Conditional rendering (a seconda che sia user - admin - superadmin)

## Compound Components (per tutti gli elementi lista, quindi un padre che contiene i listitem)

# 4 Architettura Backend

## 4.1 Introduzione

Per il backend del servizio, è stato scelto di adottare il Serverless Architecture Pattern.  
Questo modello di cloud computing permette di sviluppare e gestire l’applicazione senza dover gestire direttamente i server.  
Il seguente pattern permette di avere un’elevata scalabilità, alta disponibilità, e bassi costi senza preoccuparsi dell’infrastruttura sottostante, grazie ad esso sarà infatti possibile scalare verso l’alto o verso il basso a seconda delle esigenze del servizio.

## 4.2 Schema base di dati

## 

## 4.3 Design Patterns

* **Function-as-a-Service (FaaS):** consente di scrivere funzioni senza la necessità di gestire l'infrastruttura sottostante;
* **Backend-as-a-Service (BaaS):** consente di utilizzare servizi cloud per gestire funzionalità comuni del backend, come l'autenticazione degli utenti, lo storage dei dati e la gestione delle notifiche push;
* **Event-driven computing:** consente di scrivere codice che viene eseguito in risposta a eventi specifici, come ad esempio una richiesta API;
* **API Gateway:** consente di creare un punto di ingresso centralizzato per l'accesso alle funzioni serverless.

## 4.4 Documentazione API