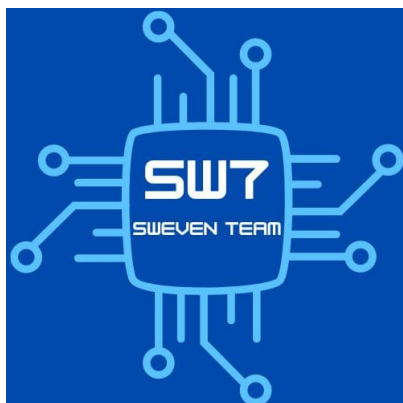


# SPECIFICA ARCHITETTURALE



**SWEVEN TEAM**  
swe7.team@gmail.com

## INFORMAZIONI SUL DOCUMENTO

<b>Versione</b>	0.0.0
<b>Uso</b>	Esterno
<b>Destinatari</b>	Gruppo Sweven Team Prof. Tullio Vardanega Prof. Riccardo Cardin Azienda Imola Informatica
<b>Stato</b>	in lavorazione
<b>Redattori</b>	
<b>Verificatori</b>	
<b>Approvatori</b>	

## Sintesi

Specifica architetturale e delle tecnologie per la realizzazione del *Chatbot<sub>G</sub>*.

## Diario delle modifiche

Versione	Data	Descrizione	Ruolo	Autore	Verificatore
	2022-08-22	Scrittura \$2.2 e \$2.3	Progettista	Irene Benetazzo	
	2022-08-09	Scrittura \$3	Progettista	Irene Benetazzo	
	2022-08-08	Scrittura \$1	Amministratore	Irene Benetazzo	
	2022-07-21	Creazione documento	Amministratore	Irene Benetazzo	

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del Documento . . . . .	4
1.2	Scopo del Capitolato . . . . .	4
1.3	Glossario . . . . .	4
1.4	Riferimenti . . . . .	4
1.4.1	Normativi . . . . .	4
1.4.2	Informativi . . . . .	4
<b>2</b>	<b>Architettura</b>	<b>5</b>
2.1	Client . . . . .	5
2.2	Server . . . . .	5
2.2.1	Chatterbot . . . . .	5
2.2.2	State . . . . .	5
2.2.3	Statement_State . . . . .	5
2.2.4	Statement . . . . .	5
2.2.5	LogicAdapter . . . . .	5
2.2.6	Request . . . . .	5
2.2.7	Login . . . . .	6
2.2.8	Logout . . . . .	6
2.2.9	Activity . . . . .	6
2.2.10	Gate . . . . .	6
2.2.11	Project_Creation . . . . .	6
2.2.12	Presence . . . . .	6
2.3	API Rest . . . . .	6
<b>3</b>	<b>Tecnologie</b>	<b>7</b>
3.1	Server . . . . .	7
3.1.1	Python . . . . .	7
3.1.1.1	Chatterbot . . . . .	7
3.1.2	API Rest Imola Informatica . . . . .	7
3.2	Client . . . . .	7
3.2.1	React . . . . .	7
3.2.2	HTML . . . . .	7
3.2.3	CSS . . . . .	7
3.2.4	Flask . . . . .	7
3.2.5	API AssemblyAI . . . . .	7

# 1 Introduzione

## 1.1 Scopo del Documento

La Specifica Architetturale ha lo scopo di descrivere le scelte architettureali e tecnologiche attuate per la realizzazione del *Chatbot<sub>G</sub>*.

## 1.2 Scopo del Capitolato

Lo scopo di tale progetto è quello di sviluppare un Chatbot che interfacciandosi con software aziendali spesso complessi e dispersivi, semplifichi i compiti che i dipendenti devono svolgere. In particolare vengono individuate le seguenti operazioni:

- Tracciamento della presenza in sede (**EMT<sub>G</sub>**)
- Rendiconto attività svolte quotidianamente (**EMT<sub>G</sub>**)
- Apertura del cancello aziendale (**MQTT<sub>G</sub>**)
- Creazione di una riunione in un servizio esterno
- Servizio di ricerca documentale (**CMIS<sub>G</sub>**)
- Creazione e tracciamento di bug (**Redmine<sub>G</sub>**)

## 1.3 Glossario

Per assicurare la massima fruibilità e leggibilità del documento, il team SWEven ha deciso di creare un documento denominato *Glossario* il cui scopo sarà quello di contenere le definizioni dei termini ambigui o specifici del progetto. Sarà possibile riconoscere i termini presenti al suo interno in quanto terminanti con la lettera *G* posta come pedice della parola stessa.

## 1.4 Riferimenti

### 1.4.1 Normativi

- Norme di Progetto *v1.0.0*

### 1.4.2 Informativi

- [Capitolato di appalto C1 - BOT4ME](#)
- [Slide del corso - Diagrammi dei casi d'uso](#)
- [Slide del corso - Diagrammi di sequenza](#)
- [Slide del corso - I pattern architetturali](#)

## 2 Architettura

L'architettura del prodotto è suddivisa tra Client e Server, inoltre si utilizzano le *API Rest<sub>G</sub>* messe a disposizione dall'azienda Imola Informatica. —IMG unica, mentre se è separata Client e Server vanno messe dentro relativa sezione —

### 2.1 Client

### 2.2 Server

#### 2.2.1 Chatterbot

classe della libreria esterna scritta in *Python<sub>G</sub>*

#### 2.2.2 State

classe che è la generalizzazione di tutti i vari stati e come dato privato si salva l'attuale stato corrente e pubblicamente dispone anche di un metodo per aggiungere informazioni necessarie per completare la richiesta in corso.

**State\_Null** sottoclasse aggiuntiva di *State* che comprende il caso di uno stato vuoto, una possibile causa può essere quando l'utente non è autenticato.

#### 2.2.3 Statement\_State

classe che fa da intermediario tra lo *State* e lo *Statement*. Come unico campo privato ha l'*Api-Key<sub>G</sub>* che certifica se l'utente è autenticato e la rende visibile pubblicamente tramite una funzione get.

#### 2.2.4 Statement

classe che viene fornita come dipendenza della libreria *Chatterbot<sub>G</sub>* e ha il compito di memorizzare i dati e viaggia tra il Client, *LogicAdapter* e Server.

#### 2.2.5 LogicAdapter

classe astratta base per tutte le classi adapter derivate, una per ogni funzionalità, dispone dei due metodi base di cui verrà fatto l'*overriding<sub>G</sub>*:

- **can\_process**: metodo booleano che controlla tutte le varie condizioni e se tutto okay fa procedere il metodo *process*.
- **process**: controlla ed elabora tutti i dati forniti così da produrre una risposta.

#### 2.2.6 Request

generalizzazione di tutte le varie *request* distinte per funzionalità. Interfaccia che riceve i dati pronti verificandone la completezza e in base all'*adapter* invia la richiesta *HTTP<sub>G</sub>* alle *API Rest<sub>G</sub>* di Imola per interagire con i loro servizi e soddisfare la richiesta dell'utente.

### 2.2.7 Login

### 2.2.8 Logout

### 2.2.9 Activity

classi *Adapter\_Activity*, *State\_Activity* e *Request\_Activity* per la funzionalità di consuntivare le ore dedicate ad un progetto compreso le eventuali ore di viaggio.

### 2.2.10 Gate

classi *Adapter\_Gate*, *State\_Gate* e *Request\_Gate* per la funzionalità di apertura cancello

### 2.2.11 Project\_Creation

classi *Adapter\_Project\_Creation*, *State\_Project\_Creation* e *Request\_Project\_Creation*

### 2.2.12 Presence

classi *Adapter\_Presence*, *State\_Presence* e *Request\_Presence* per la funzionalità di registrazione della presenza

## 2.3 API Rest

Un'API REST è un'interfaccia di programmazione delle applicazioni conforme ai vincoli dello stile architettuale REST, che consente l'interazione con servizi web RESTful.

Il termine REST è l'acronimo di REpresentational State Transfer. REST è un insieme di vincoli architeturali, non un protocollo né uno standard. Quando una richiesta client viene inviata tramite un'API RESTful, questa trasferisce al richiedente o all'endpoint uno stato rappresentativo della risorsa. L'informazione viene consegnata in HTTP in un formato JSON, HTML, Python o txt.

## 3 Tecnologie

### 3.1 Server

#### 3.1.1 Python

Linguaggio di programmazione ad alto livello, adatto alla programmazione orientata agli oggetti. E' stato utilizzato per sviluppare il back-end insieme alla libreria esterna Chatterbot.

**3.1.1.1 Chatterbot** Libreria esterna in *Python<sub>G</sub>* che utilizza algoritmi di intelligenza artificiale per trovare la migliore risposta per emulare il comportamento di un *chatbot<sub>G</sub>* nel server. Grazie alla sua flessibilità si sono implementati degli adapter che modellano e gestiscono le varie richieste dell'utente.

Durante l'esecuzione Chatterbot crea in automatico dei file dall'estensione *SQL<sub>G</sub>*

#### 3.1.2 API Rest Imola Informatica

L'azienda ha fornito delle *API Rest<sub>G</sub>* che permettono al *chatbot<sub>G</sub>* di interagire con i loro sistemi aziendali.

### 3.2 Client

#### 3.2.1 React

React è una libreria JavaScript per costruire l'interfaccia utente caratterizzata dal fatto che è dichiarativa, efficiente e flessibile. E' stato utilizzato per creare l'applicazione lato client.

#### 3.2.2 HTML

Linguaggio di markup, in standard W3C, per documenti visualizzabili attraverso un web browser

#### 3.2.3 CSS

Linguaggio di formattazione per i documenti HTML.

#### 3.2.4 Flask

Framework Python per lo sviluppo di applicazioni web. Flask contiene tutte le classi e le funzioni necessarie per la costruzione di una web app, e ha agevolato l'organizzazione e la gestione del *chatbot<sub>G</sub>*

#### 3.2.5 API AssemblyAI

L'API deve essere integrata con React e permette di tradurre automaticamente l'audio in testo.