



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Informatica

A.a. 2023/2024

Sessione di laurea Aprile 2024

Estensione di una Piattaforma Web di e-voting tramite Riconoscimento Vocale

Relatori:

Luigi De Russis

Mirko Landolfo

Luca Monterzino

Candidato:

Salvatore Mallemaci

Ringraziamenti

Ringrazio il Prof. De Russis per avermi guidato durante la fase di realizzazione della tesi.

Ringrazio l'azienda Pro Logic Informatica per la fiducia concessa, un grazie in particolare va a Mirko e Luca per il loro prezioso supporto e contributo durante lo sviluppo del progetto e ai miei colleghi, i quali mi hanno fatto sentire parte del gruppo fin dal primo giorno.

Ringrazio mamma e papà, per il loro supporto e amore incondizionato in tutto quello che faccio, senza aver mai esercitato alcun tipo di pressione su di me, spero di continuare a renderli orgogliosi.

Ringrazio mio fratello Vincenzo, un riferimento costante nella mia vita fin da quando ne ho memoria, sempre pronto ad ascoltarmi, aria di casa, ad una porta di distanza.

Ringrazio i miei nonni, sia coloro che sono accanto a me, che quello che purtroppo non c'è più, sempre entusiasti di sentire novità riguardo al mio percorso, anche senza capirci più di tanto di quello che gli racconto.

Ringrazio i miei zii e cugini, sempre considerati come un'estensione naturale della famiglia di cui mi voglio circondare e con cui continuare a crescere.

Infine ringrazio i miei amici, quelli di una vita e quelli conosciuti lungo il cammino, per la loro capacità di distillare gli istanti durante questi anni, accompagnandoli un po' con la loro leggerezza.

Indice

Elenco delle figure	VI
1 Introduzione	1
1.1 Contesto	1
1.2 Obiettivo	2
1.3 Struttura della tesi	3
2 Studio del dominio	4
2.1 Tipologie di elezione	5
2.1.1 Europee	6
2.1.2 Politiche	7
2.1.3 Regionali	8
2.1.4 Comunali	9
2.1.5 Referendum	10
2.2 Raccolta dati in Eleweb	11
2.2.1 Affluenze	11
2.2.2 Raggruppamenti	13
2.2.3 Liste	13
2.2.4 Preferenze	14
3 Progettazione del sistema	16
3.1 Studio dell'interfaccia	16
3.2 Speech Recognition	19
3.3 Speech Synthesis	20
3.4 Backend	20
4 Implementazione Speech Recognition	21
4.1 Tecnologie utilizzate in Eleweb	21
4.1.1 PHP	21
4.1.2 HTML	22
4.1.3 Javascript	22

4.2	Scelta della tecnologia	22
4.3	Funzionamento Speech Recognition	23
5	Implementazione Speech Synthesis	29
5.1	Tecnologie utilizzate	29
5.2	Scelta della tecnologia	29
5.3	Funzionamento Speech Synthesis	30
5.4	Riepilogo	34
6	Prototipo	36
6.1	Integrazione dei moduli	36
6.2	Prototipo risultante	36
6.2.1	Operazioni preliminari	36
6.2.2	Conversazione utente - applicativo	38
7	Valutazione	52
7.1	Pianificazione	52
7.2	Attività	54
7.3	Svolgimento	55
7.4	Risultati	55
7.5	Possibili modifiche	57
8	Conclusioni	60
8.1	Sviluppi futuri	61
8.2	Best practices	61
A	Script per la valutazione del prototipo	63
A.1	Presentazione	63
A.2	Istruzioni	64
A.3	Attività	64
A.3.1	Attività 1: Elementi di interazione	64
A.3.2	Attività 2: Set up conversazione	64
A.3.3	Attività 3: Primo avvio conversazione	64
A.3.4	Attività 4: Conversazione “Affluenze”	65
A.3.5	Attività 5: Conversazione “Liste”	65
A.3.6	Attività 6: Interruzione conversazione	65
A.3.7	Attività 7: Riepilogo dati	65
A.4	Debriefing	66
B	Questionari SUS	67
	Bibliografia	73

Elenco delle figure

2.1	Homepage Eleweb - sezione Raccolta Dati	12
2.2	Eleweb - sezione Affluenze	12
2.3	Eleweb - sezione Raggruppamenti	13
2.4	Eleweb - sezione Raggruppamenti - schede	14
2.5	Eleweb - sezione Liste	14
2.6	Eleweb - sezione Preferenze	15
3.1	Prototipo Figma sezione Raggruppamenti	18
4.1	Avvio Speech Recognition	25
5.1	Elenco di selezione lingua	31
5.2	Slider di selezione voce e tono	32
5.3	Avvio Speech Synthesis	33
6.1	Gestione della voce sintetizzata	37
6.2	Interfaccia di inserimento dati sindaco sezione Raggruppamenti . . .	39
6.3	Interfaccia di inserimento dati schede sezione Raggruppamenti . . .	40
6.4	Inizio conversazione sezione Raggruppamenti	41
6.5	Registra la risposta sezione Raggruppamenti	43
6.6	Interfaccia di inserimento dati sezione Affluenze	46
6.7	Interfaccia di inserimento dati sezione Liste	47
6.8	Interfaccia di inserimento schede sezione Liste	47
6.9	Interfaccia di inserimento dati sezione Preferenze	48
6.10	Interfaccia di riepilogo dati sezione Preferenze	48
6.11	Riepilogo dei dati inseriti	50

Capitolo 1

Introduzione

1.1 Contesto

“L’Italia è una Repubblica democratica, fondata sul lavoro. La sovranità appartiene al popolo, che la esercita nelle forme e nei limiti della Costituzione”. Così asserisce l’articolo 1 della Costituzione Italiana [1], primo dei 139 articoli, i quali vanno a comporre la principale fonte del diritto della Repubblica Italiana, dalla quale dipendono gerarchicamente tutte le altre norme giuridiche dell’ordinamento dello Stato. È tramite il voto che si assicura al cittadino la possibilità di esprimere le proprie preferenze durante un’elezione, contribuendo così al funzionamento e alla vitalità del sistema elettorale. Il voto si qualifica dunque come un tassello fondamentale della democrazia, e in quanto tale è estremamente importante che i dati elettorali vengano raccolti correttamente, in maniera impeccabile e sicura, garantendo l’integrità del processo elettorale e preservando la fiducia dei cittadini nel sistema democratico.

È in questo contesto che si posiziona Eleweb [2], una piattaforma di e-voting e per la raccolta di dati scrutinati dai presidenti di seggio, sviluppata dall’azienda Pro Logic Informatica s.r.l. [3]. Eleweb permette di gestire ogni tipo di evento elettorale a livello nazionale (Camera e Senato o Europee) e locale (Elezioni Amministrative Regionali, Provinciali e Comunali), integrando l’interfaccia con il Ministero dell’Interno, tramite algoritmi di calcolo e di verifica per una corretta trasmissione dei dati elettorali.

Nella presente tesi è stato ideato e implementato un modulo di riconoscimento vocale delle informazioni contenute nelle “Comunicazioni del seggio agli uffici elettorali”, al fine di rendere la raccolta e la trasmissione dei dati più veloce e guidata, tramite di fatto una conversazione tra l’utente e l’applicativo, grazie anche a un modulo di sintesi vocale che lo accompagnerà nel processo di inserimento dati.

Al momento infatti, l’inserimento dei dati all’interno di Eleweb avviene tramite

una copiatura manuale a partire dai fonogrammi, ossia una serie di documenti cartacei, i quali contengono i dati ricavati a seguito dello spoglio della sezione elettorale, compilati direttamente dai presidenti di seggio.

Questa fase di copiatura spesso viene eseguita da due persone, una addetta alla dettatura dei dati contenuti all'interno dei fonogrammi, l'altra addetta alla trascrizione dei dati all'interno dell'applicativo Eleweb. In alcuni casi, le due persone non si trovano neanche fisicamente vicine, bensì comunicano telefonicamente.

L'introduzione di queste funzionalità potrebbe essere in grado dunque di semplificare questo processo di trascrizione dati coinvolgendo meno persone, andando a ridurre notevolmente il bisogno di coordinazione tra operatori e il rischio di errori derivanti da tale comunicazione remota, senza stravolgerne l'approccio operativo.

1.2 Obiettivo

L'obiettivo di questa tesi è dunque lo sviluppo di un'estensione della piattaforma Eleweb, all'interno della quale sarà possibile inserire i dati elettorali a voce, dove l'utente potrà intraprendere una vera e propria conversazione con l'applicativo. Per garantire ciò, sono stati pensati due moduli:

- Un modulo di Speech Recognition, il cui compito sarà quello di coprire le funzionalità di riconoscimento vocale, ossia riconoscere quanto detto dall'utente e trascriverne i dati all'interno della piattaforma.
- Un modulo di Speech Synthesis, il cui compito sarà quello di coprire le funzionalità di Text to Speech, guidando l'utente nel corso della conversazione tramite delle istruzioni vocali, per favorire l'inserimento e la validazione dei dati elettorali.

Prima di procedere con l'integrazione delle nuove funzionalità, sarà necessario condurre un'analisi dettagliata dell'interfaccia attuale per valutare le possibili sinergie e gli adattamenti necessari.

Sulla base delle informazioni ottenute, si procederà con la ricerca delle varie tecnologie disponibili sul mercato che possano soddisfare le esigenze richieste, optando possibilmente per delle soluzioni Open Source, in quanto tale approccio offrirebbe flessibilità, sicurezza e collaborazione, rispondendo alle necessità del progetto con costi ridotti e senza dipendenza da fornitori proprietari.

Per finire, si procederà con l'implementazione e l'integrazione dei moduli sopra citati, al fine di creare un prototipo funzionante.

1.3 Struttura della tesi

La tesi inizia con un'analisi generale del panorama attuale delle elezioni in Italia e di come esse vengano gestite dall'applicativo Eleweb. In seguito, viene delineata l'implementazione dei moduli di Riconoscimento Vocale e Speech Synthesis all'interno dell'applicazione, con particolare attenzione all'integrazione di tali componenti e alla definizione di un prototipo funzionante. Per finire, vi è la valutazione del prototipo, basata su test di usabilità svolti da cinque amministratori comunali.

Più nel dettaglio:

- Capitolo 2 “*Studio del dominio*”: quadro generale delle elezioni in Italia, tipologie di elezioni, funzionamento di Eleweb.
- Capitolo 3 “*Progettazione del sistema*”: studio dell'interfaccia, struttura dei moduli di Speech Recognition e di Speech Synthesis.
- Capitolo 4 “*Implementazione Speech Recogniton*”: tecnologie utilizzate per implementare il modulo di riconoscimento vocale.
- Capitolo 5 “*Implementazione Speech Synthesis*”: tecnologie utilizzate per implementare il modulo di sintesi vocale.
- Capitolo 6 “*Prototipo*”: integrazione dei due moduli, descrizione del risultato ottenuto.
- Capitolo 7 “*Valutazione*”: test di valutazione e usabilità del prodotto.
- Capitolo 8 “*Conclusioni*”: risultati ottenuti, punti di forza e limitazioni, best practices da adottare.

Capitolo 2

Studio del dominio

Per poter sviluppare al meglio la tecnologia richiesta, è prima necessario analizzare come funzioni il mondo degli eventi elettorali, sia in Eleweb, sia nei seggi veri e propri. A seconda del tipo di elezione, Eleweb ne gestisce la raccolta in 4 sub-eventi, ossia affluenze, raggruppamenti, liste e preferenze. Inoltre, a parità di sub-evento, a seconda dell'elezione possono esserci entità diverse che vengono coinvolte, ad esempio il sub-evento “raggruppamenti” si riferisce ai candidato sindaco nelle amministrative, mentre si riferisce ai raggruppamenti elettorali nel caso delle politiche.

Come anticipato nel capitolo introduttivo, al momento, l'inserimento dei dati all'interno di Eleweb avviene tramite una copiatura manuale a partire dai fonogrammi (a cui ci si riferirà tramite il nome usato comunemente “foglietti”), ossia una serie di documenti cartacei, i quali contengono i dati ricavati a seguito dello spoglio della sezione elettorale, compilati direttamente dai presidenti di seggio.

Il riconoscimento vocale nasce dunque dall'esigenza di fornire una soluzione più pratica e veloce per quanto riguarda l'inserimento dei dati all'interno di Eleweb, in grado di automatizzare questo processo di trascrizione dati e coinvolgere meno persone.

Per dare maggior contesto, si prenda in esame l'esempio del Comune di Torino: all'interno di questo comune ci sono più di un migliaio di sezioni. Se il comune dovesse assegnare una persona per sezione allo scopo di prendere i dati del foglietto, si avrebbe una cosa come 1000 dipendenti assunti solo per questo incarico, il che potrebbe risultare come uno spreco di denaro.

Per ovviare parzialmente a questo problema, i vari comuni hanno pensato di usufruire della collaborazione dei vigili, ognuno dei quali si occupa di raccogliere i foglietti da diverse sezioni, per poi consegnarle in un centro di raccolta, dove ci saranno i vari incaricati all'inserimento dei dati sulla piattaforma Eleweb. Così facendo, il numero di incaricati scenderà rispetto alla stima fatta in precedenza, ma ciascuno di essi si troverà a dover inserire una grande quantità di dati.

Un'altra soluzione adoperata attualmente è quella di farsi dettare i dati per via telefonica, in modo tale da comunicare i dati contenuti all'interno dei foglietti, senza dover passare necessariamente per un trasporto fisico.

Utilizzare una soluzione come quella del riconoscimento vocale, in combinazione ad un modulo di sintesi vocale, sarebbe dunque in grado di permettere di fatto una conversazione tra l'incaricato e la piattaforma Eleweb stessa, andando così a velocizzare l'operazione e richiedere meno personale, aumentandone l'efficienza.

All'interno di questo capitolo si farà una panoramica di quali sono le elezioni supportate in Eleweb, che tipo di dati vengono raccolti, come vengono configurate, ecc.

2.1 Tipologie di elezione

In Italia si distinguono ben 5 tipologie di elezioni.

A livello nazionale:

- Europee: si tengono ogni 5 anni e consentono ai cittadini dell'Unione Europea di eleggere i loro rappresentanti al Parlamento Europeo.
- Politiche: si tengono di solito ogni 5 anni per eleggere i membri del Parlamento Italiano, che è composto dalla Camera dei Deputati e dal Senato della Repubblica. Durante queste elezioni, i cittadini italiani votano per i partiti politici nazionali e i loro candidati, che competono per i seggi in entrambe le camere del Parlamento.
- Referendum: non c'è una periodicità predefinita per i referendum, ma possono essere convocati in diverse occasioni su varie questioni, come emendamenti costituzionali, leggi nazionali, regionali o locali, o anche su questioni di interesse internazionale. Durante un referendum, i cittadini vengono chiamati a votare "sì" o "no" su una specifica domanda posta dal governo o attraverso un'iniziativa popolare.

A livello locale:

- Regionali: si tengono di solito ogni 5 anni per eleggere i membri dei consigli regionali delle 20 regioni italiane. Durante queste elezioni, i cittadini votano per i partiti politici o le liste civiche che competono per i seggi nei consigli regionali.
- Provinciali: si svolgevano per eleggere i consiglieri provinciali e il presidente della provincia. Tuttavia, con la riforma costituzionale del 2014, molte funzioni delle province sono state trasferite ai comuni e alle regioni, riducendo l'importanza delle province stesse. Di conseguenza, le elezioni provinciali sono state

in larga parte sostituite dalle elezioni regionali e comunali. In alcune regioni, le province sono state addirittura abolite. Pertanto, al momento attuale, le elezioni provinciali non sono più un evento comune in Italia.

- **Comunali:** anche dette Amministrative, si svolgono ogni 5 anni per eleggere i rappresentanti delle amministrazioni locali, ovvero i sindaci e i consigli comunali.

Le Europee e le Politiche, essendo a livello nazionale, portano con sé una notevole quantità di dati da processare e, di conseguenza, un notevole stress sui server di Eleweb. Di contro, le Comunali e le Regionali sono organizzate in modo tale che la loro cadenza non coinvolga tutto il paese, perciò non tutte le regioni votano durante lo stesso anno/periodo (stesso discorso vale per i comuni).

Detto ciò, seguirà una descrizione più accurata di ciascuna tipologia di elezione e come esse vengono gestite all'interno di Eleweb.

2.1.1 Europee

Le elezioni europee sono un appuntamento cruciale per la democrazia dell'Unione Europea (UE). Durante queste elezioni, i cittadini dei paesi membri dell'Unione Europea eleggono i propri rappresentanti al Parlamento Europeo. I partiti politici nazionali o transnazionali presentano le proprie liste di candidati, e gli elettori possono esprimere il loro voto per determinare la composizione del Parlamento Europeo. Quest'ultimo gioca un ruolo fondamentale nella legislazione dell'UE, nell'approvazione del bilancio, nella supervisione della Commissione Europea e nella definizione delle politiche europee in una vasta gamma di settori, dall'economia alla giustizia, dall'ambiente alla politica estera. Le elezioni europee offrono agli elettori l'opportunità di influenzare il futuro dell'UE e di far sentire la propria voce su questioni che riguardano l'intera Unione.

Le votazioni per le elezioni europee [4] si svolgono in base alle leggi elettorali di ciascun paese membro dell'Unione Europea, ma generalmente seguono un processo simile:

- **Età:** Per poter votare alle elezioni europee, è necessario aver raggiunto l'età minima stabilita dal proprio paese di residenza. In genere, l'età minima per votare è di 18 anni, ma ci sono delle variazioni da un paese all'altro (ad esempio per votare in Austria l'età minima è di 16 anni). Ogni cittadino può votare una sola volta.
- **Numero di seggi:** Il numero di seggi assegnati a ciascun paese membro per il Parlamento Europeo è determinato in base alla sua popolazione. Attualmente, il Parlamento Europeo [5] conta ben 705 deputati, di cui 76 italiani, e rappresenta più di 450 milioni di persone.

- **Partiti e candidati:** I partiti politici o le liste civiche presentano le loro candidature per le elezioni europee. Gli elettori votano per i partiti o i candidati che preferiscono, scegliendo tra le liste presentate nel proprio paese. I partiti possono formare alleanze a livello europeo, e alcuni candidati possono essere inseriti in liste transnazionali. Il numero di deputati eletti da un partito politico è proporzionale al numero di voti che riceve.
- **Durata delle votazioni:** Le elezioni si svolgono durante un periodo di quattro giorni, da giovedì a domenica, con vari paesi che votano in date diverse all'interno dello stesso periodo. Questo consente ai cittadini di tutta l'Unione Europea di esprimere il loro voto durante un periodo di tempo prolungato. I cittadini dell'UE residenti in un altro paese dell'UE possono votare e candidarsi alle elezioni.

Durante il giorno delle elezioni, i cittadini si recano ai seggi elettorali designati nel loro paese di residenza e votano per il partito o il candidato prescelto. I voti vengono poi conteggiati e i risultati vengono annunciati una volta completato il processo di scrutinio in tutti i paesi membri.

Per capire meglio quali dati sono utili ai fini della raccolta all'interno di Eleweb, ci si focalizza ora sulla situazione in Italia: In base alla legge numero 18 del 1979 [6] (nota come "Legge elettorale per l'elezione dei rappresentanti italiani al Parlamento europeo") vengono definiti l'elezione dei rappresentanti italiani in parlamento, è possibile votare una sola tra le liste disponibili e il voto è valido solo per la lista, non per i candidati che possono essere invece selezionati attraverso il voto di preferenza, da 1 fino a 3 candidati (eccezione fatta per le liste di minoranze linguistiche, per le quali si può esprimere la preferenza per un unico candidato).

Per concludere, le informazioni necessarie al fine della raccolta dati per le elezioni europee in Eleweb riguardano le liste, le preferenze e le affluenze (parziali, inserite nella piattaforma durante i giorni in cui si tengono le elezioni, e finali, inserite al termine degli scrutini).

2.1.2 Politiche

Le elezioni politiche sono un processo attraverso il quale i cittadini di un paese eleggono i propri rappresentanti politici per far parte del governo nazionale o del Parlamento. Queste elezioni sono fondamentali per determinare la composizione del potere legislativo e/o esecutivo di una nazione.

Ecco come funzionano le politiche in Italia:

- **Età:** Per poter votare alle elezioni politiche è necessario aver compiuto 18 anni entro la data delle elezioni.

- Numero di seggi: Il Parlamento italiano [7] è bicamerale e comprende la Camera dei Deputati e il Senato della Repubblica. La Camera dei Deputati è composta da 630 deputati, di cui 12 sono eletti dalla circoscrizione estero. Il Senato è composto da 315 senatori, di cui 6 sono eletti dalla circoscrizione estero. I seggi sono assegnati ai partiti in base ai risultati delle elezioni proporzionali.
- Partiti e candidati: I partiti politici italiani e le coalizioni presentano le loro liste di candidati per le elezioni politiche. Gli elettori possono votare per una lista specifica di partito o coalizione, o possono esprimere il loro voto per un candidato specifico all'interno della lista.
- Registro elettorale: I cittadini italiani e gli altri cittadini dell'Unione Europea che risiedono legalmente in Italia devono essere iscritti nel registro elettorale del comune in cui risiedono per poter partecipare alle elezioni politiche. Se risiedono all'estero, possono essere iscritti nel registro elettorale per gli italiani residenti all'estero (AIRE) [8] e votare per corrispondenza o presso i consolati o le sedi diplomatiche.
- Giorni di voto: Le elezioni politiche in Italia si svolgono di solito in un singolo giorno, in conformità con il calendario stabilito per le elezioni in tutto il paese.

L'elezione avviene in contemporanea ma gli elettori ricevono schede separate per ciascuna camera e votano nei seggi elettorali. I voti vengono conteggiati separatamente per la Camera e per il Senato e i seggi vengono assegnati in base ai risultati proporzionali. Dunque nonostante il periodo sia lo stesso, le votazioni vengono viste come due eventi elettorali distinti su Eleweb. L'elettore può votare sia scegliendo la lista (dando così il voto al candidato a essa collegato per il collegio uninominale) sia votando il candidato uninominale senza esprimere esplicitamente la lista.

Per finire, le informazioni necessarie al fine della raccolta dati per le elezioni politiche in Eleweb riguardano le affluenze (sia parziali sia finali), raggruppamenti e liste.

2.1.3 Regionali

Le elezioni regionali sono elezioni che si tengono a livello di regione all'interno di uno stato o di un paese. Durante queste elezioni, gli elettori scelgono i rappresentanti regionali che li governeranno e prenderanno decisioni su questioni specifiche riguardanti la regione stessa, come l'istruzione, la sanità, l'urbanistica e altre politiche locali. I partiti politici e i candidati indipendenti si presentano alle elezioni per competere per i seggi regionali, con l'obiettivo di formare un governo regionale

e influenzarne le politiche. Trattandosi di elezioni che riguardano solo le singole regioni, ci sono delle eccezioni per quanto riguarda le modalità di votazione (si distinguono le regioni a statuto speciale a quelle a statuto ordinario). Ciò che accomuna tutto il paese è che la votazione avviene per poter decretare chi sarà il presidente della regione.

In generale, vigono le seguenti regole:

- Età: l'età minima per votare alle elezioni regionali è di 18 anni.
- Numero di seggi: dipende dalla dimensione della regione e dalla sua popolazione. Le regioni più grandi e popolate avranno più seggi rispetto a quelle più piccole. Il numero esatto di seggi può variare notevolmente.
- Partiti e candidati: Alle elezioni regionali, solitamente partecipano diversi partiti politici e anche candidati indipendenti. I partiti possono presentare liste di candidati, e gli elettori votano per il partito o per il candidato che preferiscono, a seconda del sistema elettorale in uso.
- Durata delle votazioni: le elezioni regionali solitamente durano un giorno, con i seggi elettorali aperti per un periodo di tempo stabilito, che di solito va da alcune ore del mattino fino a sera tardi.

Per concludere, le informazioni necessarie al fine della raccolta dati per le elezioni regionali in Eleweb riguardano le liste, le preferenze, i raggruppamenti (ossia chi sono i candidati presidente) e le affluenze parziali e totali.

2.1.4 Comunali

Le elezioni comunali (anche dette elezioni amministrative) decretano l'elezione del sindaco e del consiglio comunale annesso. C'è un rapporto 1:1 tra evento elettorale e comune, ciò significa che Eleweb deve essere in grado di gestire centinaia di elezioni.

Nonostante cambino da comune in comune, in generale vigono le seguenti regole:

- Età: l'età minima per votare alle elezioni comunali in Italia è di 18 anni.
- Numero di seggi: Il numero di seggi disponibili varia in base alla dimensione del comune, con un sistema proporzionale che assegna un numero di seggi proporzionato alla popolazione locale.
- Partiti e candidati: I partiti politici, le coalizioni o i candidati indipendenti presentano le loro liste di candidati per il consiglio comunale e il candidato per il ruolo di sindaco. Il numero di partiti e candidati dipende dalla situazione politica e dalla dimensione del comune.

- Durata delle votazioni: Le votazioni durano solitamente un giorno, con i seggi aperti per diverse ore durante la giornata elettorale. La durata esatta può variare a seconda delle decisioni delle autorità elettorali locali, ma di solito vanno dalle prime ore del mattino fino alla sera.

È possibile votare solo per una lista, senza esprimere preferenze per un candidato sindaco specifico; in tal caso, il voto viene automaticamente esteso al candidato sindaco associato a quella lista. Al contrario, se si vota solo per il candidato sindaco, il voto per la lista non viene conteggiato. È inoltre possibile esprimere un voto disgiunto. Se nessun candidato raggiunge il 50% dei voti al primo turno, si procede con il ballottaggio, nel quale si sfidano i due candidati con il maggior numero di voti. Tuttavia, nei comuni con meno di 15.000 abitanti, si applica un sistema elettorale a turno unico, senza ballottaggio.

Per concludere, le informazioni necessarie al fine della raccolta dati per le elezioni comunali in Eleweb riguardano le liste, le preferenze, i raggruppamenti e le affluenze parziali e totali. Nel caso di un possibile ballottaggio si hanno le affluenze parziali e totali e le liste (alle quali si fa riferimento come “candidati”).

2.1.5 Referendum

In Italia, il referendum [9] è uno strumento di democrazia diretta attraverso il quale i cittadini possono esprimere il proprio voto su questioni di interesse pubblico. Ci sono diversi tipi di referendum, tra cui:

- Referendum abrogativo: Consente ai cittadini di abrogare, cioè annullare, una legge o una parte di essa già approvata dal Parlamento. Per essere vincolante, il referendum abrogativo richiede una partecipazione di almeno il 50% degli aventi diritto al voto più uno.
- Referendum confermativo: Viene indetto per confermare o rigettare una legge o una decisione già approvata dal Parlamento. Anche in questo caso, per essere vincolante, richiede una partecipazione di almeno il 50% degli aventi diritto al voto più uno.
- Referendum consultivo: Viene indetto per chiedere l’opinione dei cittadini su una questione specifica, ma la sua esito non è vincolante per le istituzioni.

Il procedimento per l’indizione di un referendum è disciplinato dalla Costituzione italiana e dalla legge elettorale. Solitamente, il referendum è indetto su iniziativa popolare, raccolta tramite firme, oppure può essere deciso dal Parlamento o dal Presidente della Repubblica.

La partecipazione ai referendum è obbligatoria e ogni cittadino avente diritto al voto può esprimere la propria opinione. I risultati dei referendum vincolanti sono considerati vincolanti per il legislatore e devono essere attuati.

Indipendentemente dal tipo di referendum, le informazioni utili per la raccolta dati all'interno di Eleweb sono sempre le stesse, ossia le affluenze (le quali assumono un ruolo predominante in questo caso essendo che per validare un referendum in Italia, è necessario che abbia raggiunto il quorum del 50%) e le liste (che nel caso del referendum sono 2: la lista "sì" e la lista "no").

Per finire, è possibile riassumere in una tabella i sub-eventi presenti in Eleweb a seconda del tipo di elezione:

ELEZIONE	AFFLUENZE	RAGGRUPP.	LISTE	PREFERENZE
COMUNALI	X	X	X	X
REGIONALI	X	X	X	X
SENATO	X	X	X	
CAMERA	X	X	X	
EUROPEE	X		X	X
BALLOTTAGGIO	X		X	
REFERENDUM	X		X	

Tabella 2.1: sub-eventi in Eleweb per tipologia di elezione

2.2 Raccolta dati in Eleweb

All'interno di questa sezione ci si soffermerà su come vengono gestite le informazioni trattate finora. L'effettiva raccolta dati all'interno di Eleweb è raggiungibile tramite l'apposita sezione, chiamata per l'appunto "Raccolta Dati".

Si tratta dunque del modulo con cui l'utente si interfacerà e, di conseguenza, soggetto di trattamento di questa tesi, all'interno del quale verranno implementate le funzioni di riconoscimento e sintesi vocale.

L'interfaccia di Eleweb è stata pensata andando a rispettare nella maniera più fedele possibile il modo di operare degli utenti che devono andare a trascrivere i dati a partire dai foglietti.

Seguirà una panoramica su ciascuno dei sub-eventi presenti all'interno della piattaforma.

2.2.1 Affluenze

Come visto nella tabella 2.1, questo sub-evento è presente in tutte le tipologie di elezione.

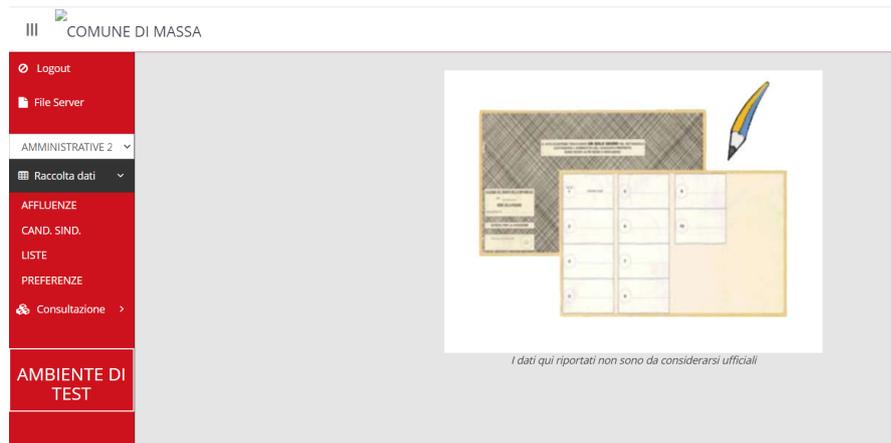


Figura 2.1: Homepage Eleweb - sezione Raccolta Dati

Interagendo con questa sezione, l'utente può inserire i dati relativi all'affluenza man mano che essi vengono comunicati dal seggio, con la restrizione che questi dati devono essere raccolti e comunicati entro 15 minuti dall'orario concordato di raccolta.

È possibile avere diverse fasce orarie, le quali vanno configurate nell'applicativo in modo da poterci interagire. Una volta scelta l'affluenza (ossia la fascia oraria), l'utente sarà invitato a scegliere il numero di sezione relativo, dopodiché appariranno i form per poter inserire i dati.

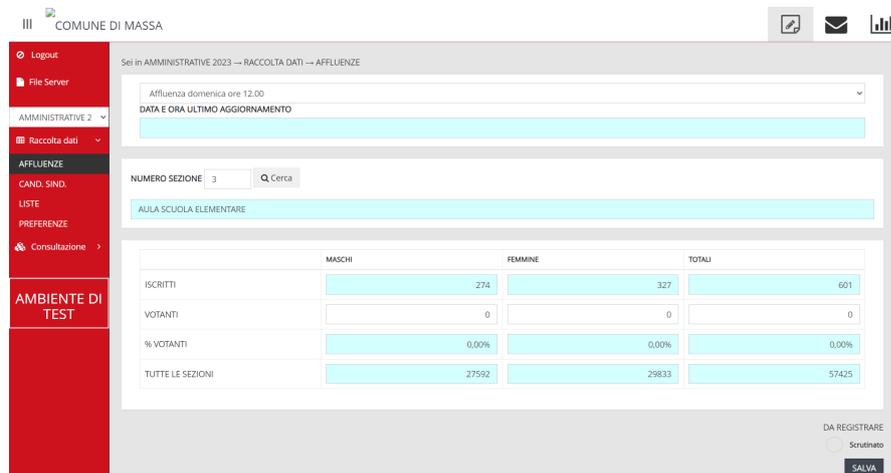


Figura 2.2: Eleweb - sezione Affluenze

Come citato prima, si distinguono affluenze parziali e finali: se per per l'affluenza totale c'è sempre una distinzione in maschi, femmine e totali, nel caso dell'affluenza

parziale potrebbe esserci direttamente solo il numero totale. Questa caratteristica è configurabile nell'applicativo ma, per quanto riguarda l'implementazione del modulo di riconoscimento vocale, si è assunto il caso più esaustivo, ossia la divisione in maschi, femmine e totali.

2.2.2 Raggruppamenti

Come visto nella tabella 2.1, questo sub-evento non è presente in tutte le tipologie di elezione, di conseguenza è stato configurato l'applicativo in modo tale che esso compaia solo se previsto dall'elezione.

Interagendo con questa sezione, l'utente può inserire i dati relativi ai candidati sindaco: più precisamente si distinguono i voti assegnati e i voti espressi al solo candidato, per poi vedere un riepilogo dei dati totali (A: totale voti validi, B: totale voti solo sindaco), i quali sono calcolati automaticamente dall'applicativo.

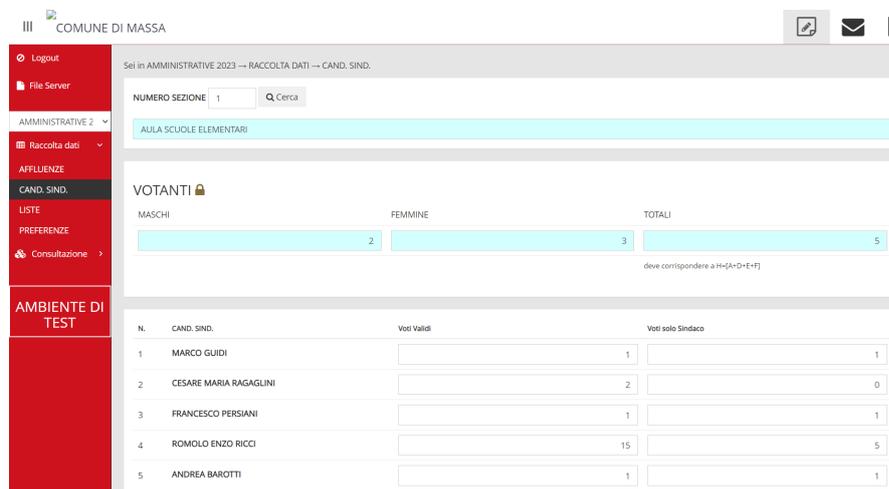


Figura 2.3: Eleweb - sezione Raggruppamenti

Al fondo di questo inserimento, è prevista una sezione con degli ulteriori form, ossia: schede bianche (D), schede nulle (E), schede contestate (F) (campi da compilare manualmente), in totale (ossia la somma delle schede citate fino ad ora) e totale ($G = [A + D + E + F]$), ossia la somma tra il totale dei voti validi e il totale delle schede). Anche in questo caso è possibile configurare l'applicativo a seconda delle esigenze del comune.

2.2.3 Liste

Come visto nella tabella 2.1, questo sub-evento è presente in tutte le tipologie di elezione.

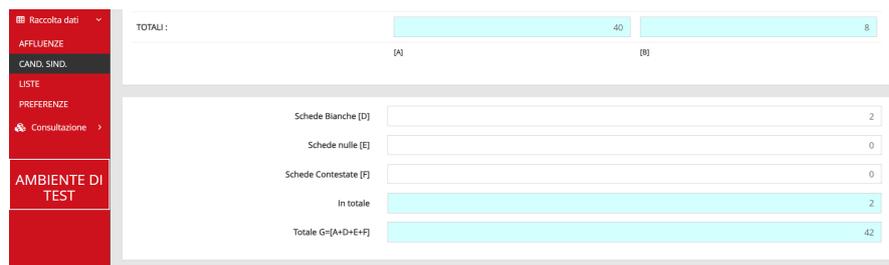


Figura 2.4: Eleweb - sezione Raggruppamenti - schede

Interagendo con questa sezione, l'utente può inserire i dati relativi alle liste candidate. Ha una struttura molto simile a quella delle liste, con l'eccezione che si inseriscono direttamente i voti validi per la lista e la sezione delle schede è compilata del tutto in automatico.

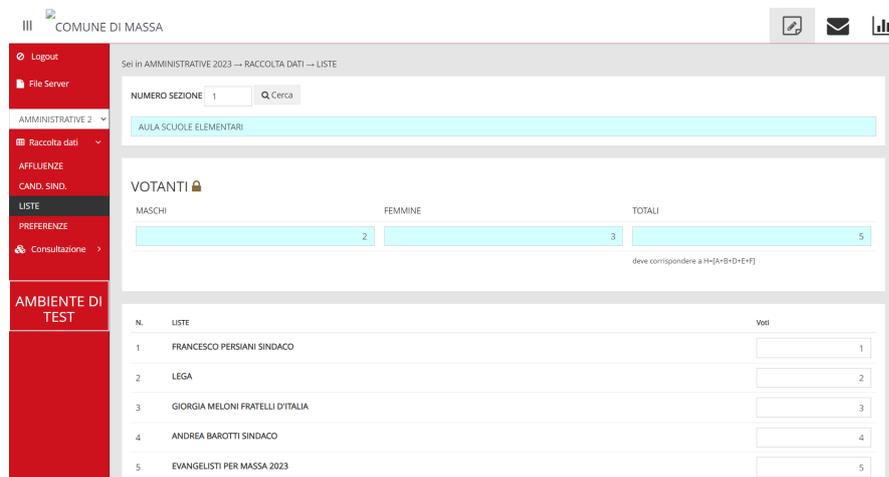


Figura 2.5: Eleweb - sezione Liste

2.2.4 Preferenze

Come visto nella tabella 2.1, questo sub-evento non è presente in tutte le tipologie di elezione, di conseguenza è stato configurato l'applicativo in modo tale che esso compaia solo se previsto dall'elezione.

Si tratta del sub-evento più ostico da gestire, in quanto caratterizzato da una grande mole di dati. In termini di dati a disposizione funziona esattamente come le liste, a differenza che al posto della lista c'è il nominativo del candidato.

Per finire, è possibile riassumere in una tabella i sub-le operazioni presenti in Eleweb a seconda del tipo di elezione:

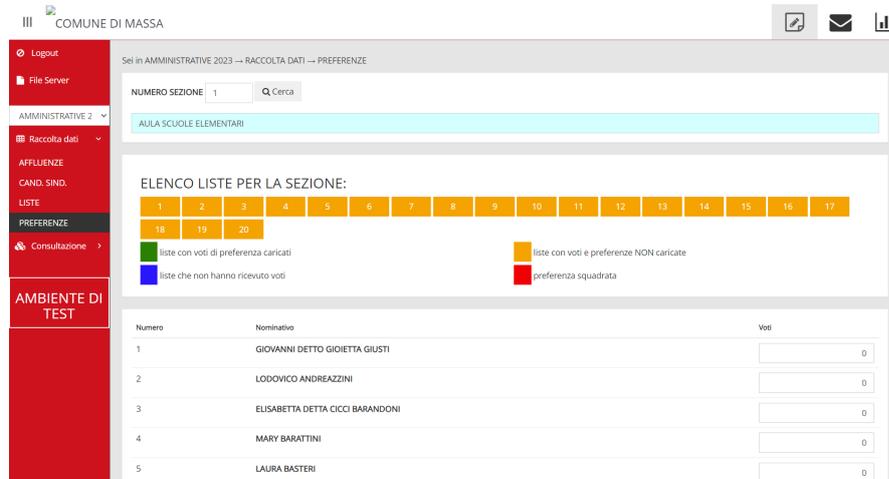


Figura 2.6: Eleweb - sezione Preferenze

ELEZIONE	Affluenza	Raggruppamenti
COMUNALI	Affluenze	Candidati Sindaco
REGIONALI	Affluenze	Candidati Presidente
SENATO	Affluenze	Coalizioni (read only)
CAMERA	Affluenze	Coalizioni (read only)
EUROPEE	Affluenze	
BALLOTTAGGIO	Affluenze	
REFERENDUM	Affluenze	

Tabella 2.2: sub-eventi in Eleweb per tipologia di elezione (Affluenze - Raggrup.)

ELEZIONE	Liste	Candidati
COMUNALI	Liste	Voti di Preferenza
REGIONALI	Liste	Voti di Preferenza
SENATO	Liste - Candidati	
CAMERA	Liste - Candidati	
EUROPEE	Liste	Voti di Preferenza
BALLOTTAGGIO	Candidati	
REFERENDUM	Sì/No	

Tabella 2.3: sub-eventi in Eleweb per tipologia di elezione (Liste - Candidati)

Capitolo 3

Progettazione del sistema

Una volta studiato il dominio e compreso il funzionamento di Eleweb, si passerà adesso allo studio e alla progettazione del sistema di speech recognition e text to speech.

Per fare ciò si è partiti dallo studio dell'interfaccia con cui l'utente interagisce, per potersi integrare nel modo più omogeneo possibile, per poi passare allo studio, scelta e implementazione della tecnologia con cui realizzare i moduli.

Si è deciso di saltare, come fase preliminare, lo studio dei foglietti in quanto si è optato di agire direttamente sulla piattaforma preesistente, senza introdurre operazioni antecedenti.

3.1 Studio dell'interfaccia

Per lo studio dell'interfaccia, si prendano come punti di riferimento visivo le figure presentate all'interno del capitolo precedente (*Affluenze 2.2, Raggruppamenti 2.3, Liste 2.5, Preferenze 2.6*).

Per semplicità, si prende come sezione di riferimento quella dei Raggruppamenti (figure 2.3 e 2.4), poiché si tratta della sezione più ricca di informazioni. Di fatto si può dire che questa sezione inglobi le altre in termini di operazioni consentite e informazioni coinvolte.

È possibile riconoscere un pattern di colori all'interno dell'applicativo, per quanto concerne i form:

- Bianco (#FFFFFF): è possibile inserire manualmente il dato all'interno del campo dati.
- Azzurro chiaro (#CCFFFF): non è possibile inserire manualmente il dato all'interno del campo dati, in quanto auto generato dal sistema.

Per prima cosa si richiede all'utente di inserire il numero di sezione (ne verrà dedotta automaticamente dal database la sede fisica in cui risiede la sezione), a seconda del quale verranno caricate tutte le informazioni riguardante la tipologia di elezione corrente.

Segue un riepilogo dei votanti relative all'ultima affluenza presso la stessa sezione, i cui dati vengono inseriti tramite la schermata vista in 2.2.

Da questo punto in poi inizia l'interazione vera e propria, dove l'utente può inserire i voti validi e i voti solo sindaco relativo al candidato corrente e, man mano che questi form vengono compilati, il counter dei totali viene automaticamente incrementato. Stesso discorso vale per la sezione delle schede.

Una volta che tutti i dati sono stati inseriti, è possibile spuntare il flag "Scrutinato", grazie al quale si segnala che i dati verranno contati e registrati nel sistema.

In definitiva, premendo il tasto "Apri foglio coll." si può vedere un riepilogo dei dati inseriti sulla schermata corrente, mentre premendo il tasto "Salva" verranno registrati i dati all'interno del database interno dell'applicativo.

Lo studio dei requisiti e le interviste annesse sono state gestite e realizzate da Pro Logic Informatica in un momento precedente all'inizio di questo progetto, di conseguenza le informazioni su quali siano i bisogni dell'utente sono stati forniti direttamente dall'azienda.

Da qui è partito lo studio di come integrare le funzionalità di riconoscimento e sintesi vocale, provando ad alterare il meno possibile l'usabilità del prodotto per gli utenti che sono soliti utilizzarlo, replicando dunque un comportamento che sia il più fedele possibile (essendo Eleweb un applicativo già esistenti e utilizzato da diversi comuni italiani, l'attenzione in questo progetto verterà maggiormente sul comparto UX rispetto a quello UI).

Di seguito il prototipo ideato.

1. Menu a tendina, tramite il quale si può selezionare il tipo di voce sintetizzata:
 - Ciascun utente potrebbe avere delle preferenze riguardo al tono, al genere, allo stile di voce. Offrire una varietà di voci consente agli utenti di scegliere quella che meglio si adatti alle loro preferenze personali.
2. Slider, tramite il quale si gestisce lo speech rate della voce sintetizzata:
 - Ritmo di ascolto preferito: consentire agli utenti di regolare la velocità della voce permette loro di adattare l'esperienza di ascolto alle proprie preferenze personali.
 - Grado di comprensione: alcuni utenti potrebbero avere difficoltà a seguire una voce sintetizzata troppo veloce o troppo lenta. La possibilità di regolare la velocità consente loro di trovare un ritmo più adatto alla loro comprensione.

Sei in AMMINISTRATIVE 2023 - RACCOLTA DATI - CAND. SIND.

NUMERO SEZIONE 2

AULA SCUOLA ELEMENTARE

VOTANTI

MASCHI 3 FEMMINE 3 TOTALI 6

Google Italiano, it - IT

Velocità

Avvia la conversazione Stop

N.	CAND. SIND.	Voti Validi	Voti solo Sindaco
1	MARCO GUIDI	<input type="text" value="1"/>	<input type="text" value="0"/>
2	CESARE MARIA RAGAGLINI	<input type="text" value="0"/>	<input type="text" value="0"/>
3	FRANCESCO PERSIANI	<input type="text" value="0"/>	<input type="text" value="0"/>
4	ROMOLO ENZO RICCI	<input type="text" value="0"/>	<input type="text" value="0"/>
5	ANDREA BAROTTI	<input type="text" value="0"/>	<input type="text" value="0"/>
6	MARCO LENZONI	<input type="text" value="0"/>	<input type="text" value="0"/>
7	DANIELA BENNATI	<input type="text" value="0"/>	<input type="text" value="0"/>
8	GUIDO MUSSI	<input type="text" value="0"/>	<input type="text" value="0"/>
TOTALI:		<input type="text" value="1"/>	<input type="text" value="0"/>

Recap

Figura 3.1: Prototipo Figma sezione Raggruppamenti

- **Efficienza:** in alcune situazioni più critiche, come l’assistenza vocale guidata all’inserimento dei dati, l’utente potrebbe voler rallentare il ritmo della conversazione per poter prestare maggiore attenzione. Di contro, in situazioni più rilassate come ad esempio un riepilogo dei dati, potrebbe permettersi di aumentare la velocità, soprattutto in presenza di una grande mole di dati.
3. Bottone “Avvia la conversazione”, tramite il quale si gestisce l’avvio della conversazione:
 - **Semplicità d’uso:** la presenza di un tasto semplifica il processo per gli utenti, eliminando la necessità di cercare altre opzioni per iniziare la conversazione.
 - **Controllo dell’utente:** premendo il tasto, l’utente può avere il controllo totale sull’inizio della conversazione, dando la possibilità di dettare il ritmo e avviare una volta pronto.
 4. Bottone “Stop”, tramite il quale si gestisce l’interruzione della conversazione:

- Controllo dell'utente: premendo il tasto, l'utente può avere il controllo sull'interruzione della conversazione.
 - Ottimizzazione di tempo: da la possibilità all'utente di interrompere la conversazione, permettendogli di poter scegliere, piuttosto che obbligarlo a far "scadere" la conversazione (es: timeout).
 - Introduzione nuovo "colore": il colore #CCCCCC indica che il bottone selezionato è momentaneamente disabilitato.
5. Bottone "Recap", tramite il quale si invoca un riepilogo sintetizzato dei dati attualmente inseriti nella schermata:
- Capacità di sintesi: si offre la possibilità di richiedere all'utente un riassunto, grazie al quale avere una panoramica dei dati inseriti fino a quel momento.
 - Accessibilità: permette all'utente di poter avere un riepilogo dei dati inseriti senza dover necessariamente guardare lo schermo.

(Stesso discorso si può fare per la sezione di interfaccia relative alle schede).

Una volta ideata l'interfaccia, si passa ora a valutare i moduli richiesti, che poi dovrebbero essere in grado di interfacciarsi tra di loro, simulando di fatto una conversazione tra l'utente finale e l'applicativo.

3.2 Speech Recognition

Come anticipato precedentemente, l'obiettivo di questo progetto è implementare una soluzione di riconoscimento vocale che possa aiutare e assistere l'utente nell'operazione di inserimento dati all'interno dell'applicativo Eleweb. Perciò, piuttosto che inserire manualmente i dati, si è pensato di offrirgli la possibilità di inserire questi dati a voce.

L'idea è dunque quella di collegare il tasto "Avvia la conversazione" visto nel prototipo (figura 3.1) a delle chiamate API che permettano di catturare la voce dell'utente e trasformarla in dati numerici, andando a riempire i vari form relativi alle votazioni selezionate.

Per poter implementare queste funzionalità, seguirà nel capitolo 4, uno studio delle tecnologie presenti, una valutazione dei pro e dei contro di ciascuna e, per finire, il codice e l'implementazione.

3.3 Speech Synthesis

Se il modulo di Speech Recognition [10] consente all'utente di poter inserire vocalmente i dati, il modulo di Speech Synthesis [11] permette all'utente la ricezione di istruzioni durante le operazioni.

Questo gioca un ruolo fondamentale, poiché senza istruzioni l'esperienza di riconoscimento vocale peggiorerebbe drasticamente, in quanto mancherebbe una guida detti i ritmi (quando è il momento di parlare/ascoltare, quando il tutto ha inizio/fine), che tipo di dato ci si aspetta, relativo a quale entità (in questo caso candidato sindaco, lista, scheda, ecc.).

Per poter implementare queste funzionalità, seguirà nel capitolo 5, uno studio delle tecnologie presenti, una valutazione dei pro e dei contro di ciascuna e, per finire, il codice e l'implementazione.

3.4 Backend

Dopo aver valutato la situazione, si è deciso di non passare per un backend, in quanto si ha la possibilità di operare direttamente sui form, andando di fatto a sostituire l'operazione di scrittura manuale tramite tastiera con quella di inserimento vocale, sfruttando i controlli già presenti lato client offerti da Eleweb, alleggerendo il carico computazionale che potrebbe gravare lato server.

Capitolo 4

Implementazione Speech Recognition

In questo capitolo si farà un'analisi del modulo di Speech Recognition all'interno dell'applicativo Eleweb, a partire dalla tecnologia scelta per poi trattare il funzionamento logico del modulo.

4.1 Tecnologie utilizzate in Eleweb

La web app di Eleweb è stata realizzata utilizzando PHP, HTML e Javascript. Di conseguenza, la scelta delle tecnologie è stata imposta, al fine di potersi integrare nel modo più semplice e fedele possibile, in quanto il contrario avrebbe richiesto di stravolgere il codice.

4.1.1 PHP

PHP [12] è un linguaggio di scripting open-source ampiamente utilizzato per lo sviluppo web. È progettato per generare pagine web dinamiche e interagire con database. Grazie alla sua flessibilità e alla vasta libreria di funzioni, PHP è popolare tra gli sviluppatori per la creazione di siti web complessi e applicazioni web. È in grado di integrarsi facilmente con HTML e CSS, consentendo dunque la creazione di contenuti dinamici e interattivi. Grazie anche alla sua comunità attiva e un continuo sviluppo, PHP rimane ancora oggi una scelta affidabile per la programmazione web.

4.1.2 HTML

HTML [13], acronimo di HyperText Markup Language, è il linguaggio standard per la creazione e la strutturazione delle pagine web. Basato su un sistema di tag, HTML consente di definire la struttura e il contenuto di una pagina web, includendo elementi come testo, immagini, link e form. La sua semplicità e flessibilità lo rendono un linguaggio accessibile a tutti, mentre la sua integrazione con CSS e JavaScript consente la creazione di layout e interazioni complesse.

4.1.3 Javascript

JavaScript [14] è un linguaggio di programmazione ad alto livello, interpretato e orientato agli oggetti, comunemente utilizzato per rendere le pagine web interattive e dinamiche. È ampiamente supportato dai browser moderni e consente agli sviluppatori di aggiungere funzionalità avanzate come animazioni, manipolazione del DOM e comunicazione asincrona con il server. La sua ampia adozione e la vasta gamma di librerie e framework disponibili lo rendono uno strumento essenziale per la programmazione web.

4.2 Scelta della tecnologia

Una volta presa visione dell'applicativo e delle relative tecnologie con cui è stato realizzato, si è svolta una fase di ricerca su framework e librerie che potessero offrire le funzionalità richieste, in maniera compatibile ai linguaggi utilizzati a priori.

Inizialmente si è posta l'attenzione su librerie scritte in Python, in quanto si tratta di uno dei migliori linguaggi di programmazione quando si parla di potenza e ricchezza di librerie specializzate in un campo come questo (librerie come "SpeechRecognition", "PocketSphinx" e "Google Cloud Speech-to-Text API").

Ma, poiché l'intero applicativo è stato scritto ed ideato usando PHP, HTML e JS, si è preferito concentrarsi sulla ricerca di strumenti che potessero adattarsi più naturalmente col codice pregresso, optando dunque per "Web Speech API", ossia un'interfaccia che può essere utilizzata attraverso il codice JavaScript all'interno di un'applicazione web per consentire funzionalità di riconoscimento vocale tramite un controller chiamato per l'appunto "SpeechRecognition".

La Web Speech API [15] è una specifica standardizzata aperta, il che significa che il suo sviluppo e la sua implementazione sono guidati da organizzazioni di standardizzazione come il World Wide Web Consortium [16] (W3C). Essa consente alle applicazioni web di accedere alle funzionalità di riconoscimento e sintesi vocale (la quale verrà trattata nel capitolo successivo) direttamente nel browser. Questa API offre due principali funzionalità: il riconoscimento vocale (speech-to-text) e la sintesi vocale (text-to-speech). Utilizzando JavaScript, gli sviluppatori possono

creare applicazioni web interattive che consentono agli utenti di interagire con il contenuto attraverso la voce.

Le motivazioni principali per cui si è scelta questa tecnologia sono:

- **Integrazione:** consente direttamente alle applicazioni web di accedere direttamente nel browser alle funzionalità richieste.
- **Standard aperto:** sviluppo e implementazione guidata da un team di sviluppo, disponibile gratuitamente per gli sviluppatori che desiderano integrarla nelle proprie applicazioni web.
- **Completezza:** offre entrambe le tecnologie richieste da questo progetto di tesi.

Il controller `SpeechRecognition` [17] è una parte chiave della Web Speech API che gestisce il riconoscimento vocale. Esso fornisce metodi per avviare e interrompere l'ascolto del microfono, nonché per elaborare e interpretare il flusso audio in testo utilizzabile dall'applicazione. Attraverso questo controller, è possibile configurare opzioni come la lingua e la grammatica da riconoscere, così come gestire gli eventi associati al processo di riconoscimento vocale, come il rilevamento delle parole e gli errori di trascrizione.

4.3 Funzionamento Speech Recognition

Il modulo di riconoscimento vocale (`SpeechRecognition`) offerto dalla Web Speech API consente alle applicazioni web di catturare l'input audio da un microfono dell'utente e di convertirlo in testo utilizzabile. Qui in seguito si analizzeranno le caratteristiche principali:

- **Inizializzazione:** Per utilizzare il modulo di riconoscimento vocale, è necessario creare un'istanza di un oggetto `SpeechRecognition`.

```
1 let speechRecognition = window.webkitSpeechRecognition ||
  window.SpeechRecognition;
2 if (speechRecognition) {
3   window.recognition = new speechRecognition();
4 }
```

Nella prima riga di codice si dichiara una variabile `speechRecognition` e si assegna ad essa il valore dell'oggetto `webkitSpeechRecognition` se disponibile, altrimenti si assegna il valore dell'oggetto `SpeechRecognition`. Questo approccio è utile per garantire la compatibilità con diverse implementazioni della Web Speech API nei browser.

Infatti, alcune versioni di browser (come Chrome, quella utilizzata durante lo sviluppo) utilizzano ancora il prefisso `webkit` per alcune funzionalità sperimentali, mentre altre (come Firefox) implementano la Web Speech API senza prefissi. Pertanto, utilizzando questo codice, si cerca prima l'implementazione specifica del browser e, se non disponibile, si utilizza la versione standard.

Una volta controllato che l'oggetto `SpeechRecognition` sia disponibile, ossia che il browser utilizzato supporti questa funzionalità, si istanzia un nuovo oggetto `SpeechRecognition` e lo si assegna alla proprietà `recognition` dell'oggetto globale `window`. Questo significa che la variabile `recognition` sarà accessibile globalmente in tutto il codice JavaScript dell'applicazione web, consentendo un facile accesso all'oggetto `SpeechRecognition` da qualsiasi parte del codice, il che sarà poi necessario per poter regolare correttamente il timing durante l'interazione col modulo di sintesi vocale.

- Configurazione delle opzioni: È possibile configurare diverse opzioni, come la lingua da riconoscere o i limiti di tempo per il riconoscimento.

```
1 recognition.lang = "it-IT";
2 recognition.maxAlternatives = 5;
3 // recognition.interimResults = true;
4 recognition.continuos = true;
```

- Questa riga imposta la lingua del riconoscimento vocale su italiano (codice lingua “it-IT”). Ciò indica all'API di riconoscimento vocale di ascoltare e riconoscere la lingua italiana.
- Questa riga imposta il numero massimo di alternative di trascrizione che l'API restituirà per ogni risultato di riconoscimento vocale. In questo caso, è impostato su 5, il che significa che l'API restituirà fino a 5 possibili trascrizioni per ogni risultato.
- Questa riga abilita il riconoscimento parziale, che significa che l'API restituirà risultati parziali mentre l'utente parla, anziché attendere che l'utente finisca di parlare prima di restituire il risultato completo. Tuttavia, si è preferito disabilitare questa opzione, poiché rendeva complicato portare avanti una conversazione simulata.
- Questa riga abilita il riconoscimento vocale continuo, il che significa che l'API continuerà ad ascoltare l'audio dall'utente e a riconoscerlo finché non si rende conto che quest'ultimo ha finito di parlare. Questa opzione risulta utile in questo caso per ottenere un riconoscimento vocale continuo, per far sì che sia possibili captare dei comandi vocali in tempo reale.

- Avvio del riconoscimento: Una volta configurato, è possibile avviare il riconoscimento vocale chiamando il metodo `start()` sull'oggetto `SpeechRecognition`. A questo punto, il browser inizia ad ascoltare l'audio proveniente dal microfono dell'utente.

```

1 recognition.addEventListener('start', () => {
2   vocalButton.classList.remove("myButtonOutline");
3   vocalButton.classList.add("myButtonOutlineRec");
4   vocalButton.textContent = 'Registra la tua risposta!';
5   gif.style.display = 'none';
6   gifRec.style.display = 'block';
7 })

```

In questo caso, all'interno dell'handler dell'evento “start”, si è gestito l'aggiornamento dell'interfaccia grafica:

- Si cambia la classe CSS associata al bottone “Avvia la conversazione”, poiché in base ad essa si capisce in che “stato” della conversazione ci si trova. (*myButtonOutlineRec* -> classe CSS che specifica che è il momento di registrare)
- Viene aggiornato il testo del bottone in “Registra la tua risposta!”, facendo così capire all'utente che è il momento di parlare per poter inserire i dati vocalmente.
- Compare una gif a schermo per poter dare anche un input visivo.



Figura 4.1: Avvio Speech Recognition

- Fine del riconoscimento: Questo evento viene emesso quando il riconoscimento vocale è completato, cioè quando l'utente smette di parlare o il riconoscimento viene interrotto. È utile per eseguire azioni di pulizia o salvataggio dopo che il processo di riconoscimento è terminato.

```

1 recognition.addEventListener('end', () => {
2   vocalButton.classList.add("myButtonOutline");
3   vocalButton.classList.remove("myButtonOutlineRec");
4   gifRec.style.display = 'none';
5 })

```

Anche in questo caso, all'interno dell'handler dell'evento "end", si è gestito l'aggiornamento dell'interfaccia grafica, eseguendo l'operazione complementare a quella di start:

- Si cambia la classe CSS associata al bottone "Avvia la conversazione", poiché in base ad essa si capisce in che "stato" della conversazione ci si trova. (*myButtonOutline* -> classe CSS che specifica che è il momento di ascoltare o di stallo)
 - La gif presente in precedenza viene rimossa, suggerendo all'utente che non è più il momento di parlare.
- Ascolto dell'audio: Durante il riconoscimento, il browser acquisisce l'input audio dal microfono dell'utente e lo invia al motore di riconoscimento vocale integrato.
 - Durante il processo di riconoscimento vocale utilizzando la Web Speech API, il browser acquisisce l'input audio dal microfono dell'utente utilizzando i dispositivi hardware del sistema, come la scheda audio. Questo input audio è costituito dalle onde sonore prodotte dalla voce dell'utente mentre parla.
 - Una volta acquisito, l'audio viene inviato al motore di riconoscimento vocale integrato nel browser. Questo motore di riconoscimento vocale è responsabile dell'elaborazione dell'audio e della conversione delle onde sonore in testo utilizzabile. Esso utilizza algoritmi avanzati di elaborazione del segnale e di apprendimento automatico per interpretare l'audio e identificare le parole pronunciate dall'utente.
 - Durante questo processo, il motore di riconoscimento vocale confronta l'audio ricevuto con i modelli linguistici e acustici memorizzati nel sistema per determinare quali parole sono state pronunciate. È in grado di riconoscere parole in una varietà di lingue e dialetti, a seconda delle impostazioni specifiche del riconoscimento vocale.
 - Una volta completata l'elaborazione, il motore di riconoscimento vocale restituisce il testo riconosciuto all'applicazione web sotto forma di stringa, consentendo all'applicazione di utilizzare e elaborare il testo riconosciuto secondo le necessità. Questo testo può essere quindi utilizzato per eseguire azioni come il controllo di comandi vocali, la trascrizione di discorsi o qualsiasi altra elaborazione basata sul testo.
 - Riconoscimento del testo: Il motore di riconoscimento vocale elabora l'audio e tenta di convertirlo in testo utilizzabile. Una volta completato il processo, il testo riconosciuto viene restituito all'applicazione web sotto forma di stringa.

- Gestione dei risultati: Una volta ottenuto il testo riconosciuto sotto forma di stringa, è possibile manipolarlo a proprio piacimento. In questo contesto, essendo che si ha a che fare con dei numeri, è necessario eseguire i controlli necessari, verificando sia il contenuto sia l'attendibilità di quest'ultimo.

```
1 for (let i = 0; i < event.results[0].length; i++){
2   let result = event.results[0][i].transcript;
3   let confidence = event.results[0][i].confidence;
4   if(Number.isInteger(Number(result)) || Number.isInteger(
   Number(utils.convertStringToNumber(result.toLowerCase()))))
5     {
6       ...
7       inputSearch.value = result;
8     }
9     ...
10 }
```

- In questo snippet di codice vi è una parte della gestione degli eventi result di un oggetto SpeechRecognition. Qui, event.results[0] rappresenta un array di risultati del riconoscimento vocale restituiti dall'API.
 - All'interno del ciclo for, ogni iterazione rappresenta un singolo risultato del riconoscimento vocale, in quanto per ogni espressione dettata si richiedono 5 alternative come visto precedentemente. Per ciascun risultato, il testo riconosciuto è accessibile tramite event.results[0][i].transcript e il livello di confidenza associato è accessibile tramite event.results[0][i].confidence.
 - Segue la verifica sul testo riconosciuto, ossia se esso può essere convertito in un numero. Per farlo si verificano due condizioni, se il testo risultante è di tipo numerico (es: 4, 5, ecc), oppure se è una stringa numerica (es: quattro, cinque, ecc.) e a seconda del tipo e del grado di confidence ottenuto, viene gestito al fine di poterlo associare al form di inserimento dati.
- Gestione degli eventi: Il modulo di riconoscimento vocale emette vari eventi durante il processo. L'applicazione dunque può registrarsi per questi eventi per controllare il flusso del riconoscimento e gestire i risultati in tempo reale. Oltre agli eventi di "start" e "end" visti in precedenza, segue l'evento di "result", ossia l'evento che viene emesso quando sono disponibili risultati parziali o finali del riconoscimento. É all'interno di questo handler che si gestiscono le principali operazioni di memorizzazione del risultato con conseguente aggiornamento dell'interfaccia grafica, manipolandone gli elementi css relativi (aggiornamento dei bottoni, (dis)abilitazione dei form, colorazione a seconda della confidence), nonché la ricezione dei comandi da parte dell'altro modulo.

Seguiranno maggiori spiegazioni all'interno del capitolo 6, in cui si vedranno i due moduli comunicare tra di loro, permettendo la conversazione utente - applicativo.

Capitolo 5

Implementazione Speech Synthesis

In questo capitolo si farà un'analisi del modulo di Speech Synthesis all'interno dell'applicativo Eleweb, a partire dalla tecnologia scelta per poi trattare il funzionamento logico del modulo.

5.1 Tecnologie utilizzate

Come citato nel capitolo precedente, la web app di Eleweb è stata realizzata utilizzando PHP, HTML e Javascript. Perciò la scelta delle tecnologie è stata imposta, al fine di potersi integrare nel modo più semplice e fedele possibile, in quanto il contrario avrebbe richiesto di stravolgere il codice.

5.2 Scelta della tecnologia

Come già detto nel capitolo precedente, si è optato per “Web Speech API”, ossia un'interfaccia che può essere utilizzata attraverso il codice JavaScript all'interno di un'applicazione web per consentire anche le funzionalità di sintesi vocale tramite un controller chiamato per l'appunto “SpeechSynthesis”.

Il controller SpeechSynthesis [18] è una componente fondamentale della Web Speech API che gestisce la sintesi vocale, consentendo alle applicazioni web di convertire il testo in parlato. Attraverso JavaScript, è possibile utilizzare il controller SpeechSynthesis per creare applicazioni web accessibili e interattive, offrendo agli utenti la possibilità di ascoltare il contenuto della pagina tramite voce. Il controller fornisce metodi per la gestione delle voci disponibili, la configurazione delle opzioni di sintesi, come il tono e la velocità, nonché per il controllo della riproduzione, come

l'avvio, la pausa e l'arresto della sintesi vocale. Inoltre, è possibile registrarsi per eventi associati alla sintesi vocale, come l'inizio e la fine della riproduzione, per offrire un'esperienza utente più ricca e dinamica. In sintesi, il controller `SpeechSynthesis` della Web Speech API apre la porta a una vasta gamma di applicazioni web accessibili e interattive, arricchendo l'esperienza degli utenti attraverso la sintesi vocale.

5.3 Funzionamento Speech Synthesis

Il modulo di sintesi vocale (`SpeechSynthesis`) della Web Speech API consente alle applicazioni web di convertire il testo in voce (text to speech) sfruttando le risorse offerte direttamente dal browser su cui stanno girando. Qui in seguito si analizzeranno le caratteristiche principali:

- Inizializzazione: Per utilizzare il modulo di sintesi vocale, è necessario creare un'istanza di un oggetto `speechSynthesis`.

```
1 const synth = window.speechSynthesis;
```

In pratica, `speechSynthesis` è un'interfaccia globale disponibile nei browser moderni che fornisce metodi e funzionalità per la sintesi vocale direttamente all'interno delle applicazioni web. Si assegna questo oggetto ad una costante "synth", in modo tale da poter utilizzare più facilmente le funzionalità di sintesi vocale offerte dall'API all'interno del codice, come ad esempio avviare la sintesi di un testo specifico o gestire le opzioni di sintesi.

- Definizione del testo da sintetizzare: È ora necessario passare all'oggetto `speechSynthesis` il contenuto di ciò che deve leggere e trasformare in voce. Per farlo si crea un oggetto specifico, ossia un `SpeechSynthesisUtterance`.

```
1 const speakText = new SpeechSynthesisUtterance('Qual è il  
   numero dei votanti ${param}?');  
2 /* In questo caso: param = "maschi" | "femmine" | "totali" */
```

`SpeechSynthesisUtterance` è un'ulteriore interfaccia fornita dalla Web Speech API che rappresenta un pezzo di testo da sintetizzare. Consente di definire le caratteristiche della sintesi vocale, come il testo da sintetizzare, la lingua della voce, il volume, il tono e la velocità.

- Lingua della voce: È possibile impostare la lingua con cui parlerà il sintetizzatore tramite la proprietà "lang" dell'oggetto `SpeechSynthesisUtterance`.

Per farlo è prima necessario estrarre dal browser le lingue disponibili, invocando il metodo `getVoices()` sull'oggetto `SpeechSynthesisUtterance`.

```

1 voices = synth.getVoices();
2
3 // Loop through voices and create an option for each one
4 voices.forEach(voice => {
5     if (voice.lang == "it-IT" || voice.lang == "it_IT"){
6         // Create option element
7         const option = document.createElement('option');
8         // Fill option with voice and language
9         option.textContent = voice.name + ', ' + voice.lang;
10
11        // Set needed option attributes
12        option.setAttribute('data-lang', voice.lang);
13        option.setAttribute('data-name', voice.name);
14        option.setAttribute('voice', voice);
15        voiceSelect.appendChild(option);
16    }
17 });

```

Invocando questo metodo si ottengono tutte le lingue supportate dal browser corrente, perciò è necessario filtrare per lingua di interesse (italiano in questo caso). Una volta estratte le lingue, è necessario formattarle correttamente in modo tale da figurare in un elenco a tendina in cui sarà possibile selezionare la voce preferita.

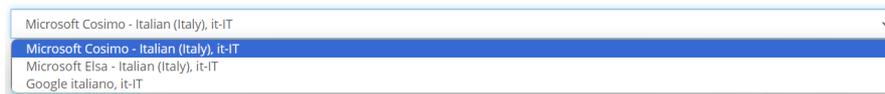


Figura 5.1: Elenco di selezione lingua

Il formato è il seguente: “voice.name, voice.lang” (es: Microsoft Elsa - Italian (Italy), it-IT).

```

1 const selectedVoice = voiceSelect.value.split(',') [0];
2 const selectedLang = voiceSelect.value.split(',')[1].trim().
  replace(/\\s/g, '');
3 // Loop through voices
4 voices.forEach(voice => {
5     if (voice.name == selectedVoice || voice.name.includes(
6         selectedVoice)) {
7         speakText.voice = voice;
8         speakText.lang = selectedLang;
9         return

```

```

9 }
10 });

```

Dall'elenco HTML si estraggono gli elementi “voce” e “lingua”, i quali vengono assegnati all'istanza `speakText`.

- Velocità e tono della voce: è possibile settare (facoltativamente) queste opzioni per personalizzare la natura della conversazione.

```

1 // Rate value change
2 rate.addEventListener('change', () => (rateValue.textContent =
   rate.value));
3
4 // Pitch value change
5 pitch.addEventListener('change', () => (pitchValue.textContent
   = pitch.value));
6
7 // Set pitch and rate
8 speakText.rate = rate.value;
9 speakText.pitch = pitch.value;

```

Innanzitutto si gestiscono gli eventi di cambiamento degli elementi HTML “rate” e “pitch”, registrando degli event listener attraverso i quali è possibile catturare il risultato proveniente dagli slider associati.



Figura 5.2: Slider di selezione voce e tono

Una volta catturati questi valori, si settano sull'istanza `SpeechSynthesisUtterance` corrispondente.

- Avvio della sintesi vocale: Una volta configurato l'oggetto, è possibile avviare la sintesi vocale vera e propria, invocando il metodo `speak()` sull'istanza di tipo `SpeechSynthesis`.

```

1 if(aspectRatio >= pcAspectRatioRange.min && aspectRatio <=
   pcAspectRatioRange.max) {
2   // dispositivo: computer
3   navigator.mediaDevices.getUserMedia({ audio: true })
4   .then(function(stream) {
5     synth.speak(speakText);
6

```

```

7   speakText.onStart = e => {
8       console.log('Start speaking...');
9       gif.style.display = 'block';
10      gifRec.style.display = 'none';
11  }
12  ...
13 } else {
14     // dispositivo: smartphone o tablet
15     ...
16 }

```

Prima di tutto, si individua su che dispositivo sta girando l'applicativo: questo check viene effettuato in quanto la cattura del microfono funziona in maniera differente a seconda del device. Su smartphone la cattura del microfono è istantanea e reattiva, su computer invece, per far sì che il microfono sia subito pronto a reagire agli input dell'utente, è necessario accedervi prima di eseguire tutte le operazioni (tramite la `getUserMedia`).

Detto ciò, ha inizio la conversazione vera e propria, quindi viene riprodotto il comando vocale che guida l'utente all'inserimento dei dati (in questo esempio, "Qual è il numero dei votanti maschi?"), nonché l'aggiornamento dell'interfaccia grafica.

Così come nel caso dell'avvio dell'istanza di speech recognition, si dà un ulteriore input all'utente che è quello visivo, tramite la comparsa di una gif che suggerisca l'operazione da fare, così come l'aggiornamento del campo di testo del bottone di conversazione e l'aggiornamento di stato del bottone di stop.



Figura 5.3: Avvio Speech Synthesis

- Esecuzione della sintesi vocale: Il browser utilizza il motore di sintesi vocale integrato per elaborare il testo fornito e generare un'uscita audio corrispondente al parlato. Google Chrome (browser utilizzato durante lo sviluppo di questo progetto) utilizza il suo motore di sintesi vocale integrato, noto come "ChromeVox". Questo motore di sintesi vocale è progettato per fornire una sintesi vocale di alta qualità e una migliore esperienza utente per gli utenti con disabilità visive, consentendo loro di navigare nel web e accedere ai contenuti tramite il parlato. ChromeVox supporta diverse lingue e voci, ed è in grado di gestire opzioni avanzate di sintesi vocale come la velocità, il ritmo e il tono della voce.

Gestione degli eventi: Come visto con il modulo di Speech Recognition, è possibile registrarsi per gli eventi emessi dall'oggetto SpeechSynthesis per monitorare lo stato della sintesi vocale e gestire le azioni aggiuntive, come la pausa, la ripresa o l'interruzione della sintesi.

```
1 speakText.onend = e => {
2   console.log('Done speaking...');
3   recognition.start();
4   gif.style.display = 'none';
5   gifRec.style.display = 'none';
6   ...
7 }
```

Anche in questo caso, all'interno dell'handler dell'evento “onend”, si è gestito l'aggiornamento dell'interfaccia grafica, eseguendo l'operazione complementare a quella di “onstart”.

5.4 Riepilogo

Per concludere questi due capitoli circa la progettazione dei due moduli, segue un riepilogo di quanto visto fino ad ora:

- **Speech Recognition (Riconoscimento Vocale):** Il modulo di riconoscimento vocale della Web Speech API permette la cattura dell'input audio proveniente dall'utente e convertirlo in testo utilizzabile. Durante il processo, il browser acquisisce l'audio dal microfono dell'utente e lo invia al motore di riconoscimento vocale integrato. Questo motore elabora l'audio e tenta di convertirlo in testo utilizzabile. Durante il riconoscimento, vengono emessi vari eventi come “speechstart” quando inizia il riconoscimento, “speechend” quando finisce e “result” quando sono disponibili risultati parziali o finali. L'applicazione può registrarsi per questi eventi per controllare il flusso del riconoscimento e gestire i risultati in tempo reale.
- **Speech Synthesis (Sintesi Vocale):** Il modulo di sintesi vocale della Web Speech API permetterà la conversione del testo in parlato direttamente nel browser. Durante il processo, l'applicazione fornisce il testo da sintetizzare e opzioni aggiuntive come la lingua, la velocità e il ritmo della voce. Il browser utilizza il motore di sintesi vocale integrato per elaborare il testo fornito e generare un'uscita audio corrispondente al parlato.

Entrambi i moduli funzionano tramite l'utilizzo di listener e handler di eventi per controllare il flusso dell'operazione e gestire i risultati in tempo reale. Nella

prossima sezione del capitolo, si esaminerà come far interagire i due moduli per creare applicazioni web interattive che combinano il riconoscimento vocale con la sintesi vocale, gestendo in maniera efficace e consona tutti gli eventi emessi dai moduli di speech recognition e speech synthesis. Ciò implica la creazione di un sistema robusto di gestione degli eventi che permetta all'applicazione di rispondere in modo appropriato alle varie fasi del processo di riconoscimento e sintesi vocale, al fine di offrire un'esperienza utente fluida e intuitiva.

Capitolo 6

Prototipo

6.1 Integrazione dei moduli

L'obiettivo che ci si prefissa adesso è dunque quello di integrare i due moduli, farli “interagire” tra di loro, in modo tale da poter creare una conversazione tra l'utente e l'applicativo.

Come detto nei precedenti capitoli, `SpeechRecognition` e `SpeechSynthesis` fanno entrambi parte di `Web Speech API`, dunque si prestano molto bene per poter essere utilizzati insieme, grazie anche al fatto che non è necessario utilizzare librerie esterne o plug-in aggiuntivi.

6.2 Prototipo risultante

6.2.1 Operazioni preliminari

Prima di poter interagire con l'applicativo per l'inserimento dei dati elettorali, ci sono alcune operazioni preliminari per poter preparare l'interfaccia e renderla operativa.

Caricamento delle lingue: è stata ideata la funzione `caricaLingue()`, il cui compito è quello di ottenere le lingue supportate dal browser su cui gira l'applicativo, per poi popolare il menu a tendina di selezione lingua. Tale funzione viene invocata nel momento in cui si seleziona la sezione dell'evento elettorale relativo:

```
1 function onclickNrsez() {
2     utils.checkCondition(utils.isInt(this.form.cdafflu.value), '
3     Selezionare 1\'affluenza.');
```

```
3     return this.caricamentoAffluenza(true)
4     .then( () => {
5         utils.gotoFocus.bind(null, this.form, 'afflu_tm'),
6         utils.gotoFocus.bind(null, this.form, 'nrsez'),
```

```

7         caricaLingue ()
8     });
9 }

```

Lo scopo di questa funzione è quello di andare ad estrarre per poi caricare i dati dell'evento elettorale selezionato (in questo snippet di esempio, le affluenze), riempiendo l'interfaccia con cui l'utente dovrà interagire. Lato interfaccia, sono stati aggiunti gli elementi HTML relativi alla selezione voce, velocità e tonalità.

```

1 <div class="form-group">
2   <label for="rate">Velocità</label>
3   <div id="rate-value" class="badge badge-primary float-right"
4     >1</div>
5   <input type="range" id="rate" class="slider" min="0.5" max="2"
6     value="1" step="0.1">
7 </div>
8 <div class="nascondi">
9   <label for="pitch">Tono di voce</label>
10  <div id="pitch-value" class="badge badge-primary float-right"
11    >1</div>
12  <input type="range" id="pitch" class="slider" min="0" max="1.5
13    " value="1" step="0.1">
14 </div>
15 <div class="form-group">
16   <select id="voice-select" class="form-control form-control-lg"
17     ></select>
18 </div>

```

Il codice si traduce dunque nella parte di schermata con cui l'utente interagirà per quanto concerne la gestione della voce sintetizzata:

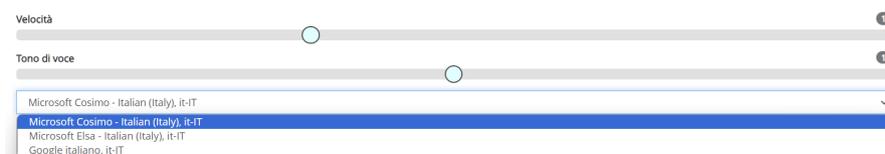


Figura 6.1: Gestione della voce sintetizzata

A questo punto, i vari elementi HTML vengono catturati tramite una querySelector basandosi sull'id associato:

```

1 const voiceSelect = document.querySelector('#voice-select');
2 const rate = document.querySelector('#rate');
3 const rateValue = document.querySelector('#rate-value');
4 const pitch = document.querySelector('#pitch');

```

```
5 | const pitchValue = document.querySelector('#pitch-value');
```

Una volta catturati gli elementi e definita un'istanza per ciascuno, è possibile interagirci e manipolarne il valore, così come discusso nel capitolo precedente, tramite i vari handler. L'utente potrà regolare a proprio piacimento la velocità con cui verrà eseguita la sintesi vocale, il tono e la tipologia di voce.

6.2.2 Conversazione utente - applicativo

Come prima cosa, è stato necessario fornire all'utente gli elementi necessari con cui interagire per poter portare avanti una conversazione con l'applicativo, in modo tale da poter sfruttare le funzionalità proposte.

- Si è dunque replicata l'interfaccia ideata, di cui si è già discusso all'interno del paragrafo 3.1, introducendo, oltre agli elementi per la gestione della voce sintetizzata, i bottoni di "Avvia Conversazione", "Stop" e "Recap".

```
1 <div class="panel panel-default">
2   <div class="panel-body" id="panel_schede">
3     <div id="divSchedeButton" class="button-container">
4       <button id="schedeButton" class="myButtonOutline"
5         onclick="divraggrup.avviaSchede()">
6         <i id="avviaSchedeIcon"></i>&nbsp;Avvia la
7         conversazione
8         </button>
9       <div class="gif-container">
10        
12        
15      </div>
16      <button id="vocalStopSchede" class="
17      myButtonOutlineDisabled" onclick="divraggrup.
18      stopConversazione()" disabled>
19        <i class="stopDisabledSchede">Stop</i>
20      </button>
21    </div>
22    <div class="form-horizontal">
23      <?php echo elaboraCampiRaggrup($view);?>
24    </div>
25    <button id="recapSchedeButton" class="myButtonOutline"
26      onclick="divraggrup.recapDatiSchede()">Recap</button>
27  </div>
```

Il codice si traduce dunque nella parte di schermata con cui l'utente interagirà per quanto concerne la conversazione e dunque l'inserimento dei dati:

N.	CAND. SIND.	Voti Validi	Voti solo Sindaco
1	MARCO GUIDI	1	1
2	CESARE MARIA RAGAGLINI	2	0
3	FRANCESCO PERSIANI	1	1
4	ROMOLO ENZO RICCI	15	5
5	ANDREA BAROTTI	1	1
6	MARCO LENZONI	5	0
7	DANIELA BENNATI	12	0
8	GUIDO MUSSI	3	0
TOTALI:		40	8

Figura 6.2: Interfaccia di inserimento dati sindaco sezione Raggruppamenti

- Lo stesso ragionamento vale per quanto riguarda la parte di schermata dedicata alle schede, ove previsto dall'evento elettorale sotto esame:

```

1 <div class="panel-body" id="panel_schede">
2   <div id="divSchedeButton" class="button-container">
3     <button id="schedeButton" class="myButtonOutline"
4       onclick="divraggrup.avviaSchede()">
5       <i id="avviaSchedeIcon"></i>&nbsp;Avvia la
6       conversazione
7     </button>
8     <div class="gif-container">
9       
11      
13    </div>
14    <button id="vocalStopSchede" class="
15    myButtonOutlineDisabled" onclick="divraggrup.
16    stopConversazione()" disabled>
17      <i class="stopDisabledSchede">Stop</i>
18    </button>
19  </div>
20  <div class="form-horizontal">
21    <?php echo elaboraCampiRaggrup($view);?>
22  </div>

```

```

17 <button id="recapSchedeButton" class="myButtonOutline"
18   onclick="divraggrup.recapDatiSchede()">Recap</button>
</div>

```

Ottenendo così l'ultima parte dell'interfaccia con cui l'utente andrà ad interagire:

Figura 6.3: Interfaccia di inserimento dati schede sezione Raggruppamenti

- Cliccando il bottone “Avvia la conversazione” che si vede nella figura 6.2, avrà inizio l’interazione con l’applicativo:
 - Viene catturata la risorsa “microfono” del dispositivo su cui gira l’applicativo.
 - Il testo all’interno del bottone cambia da “Avvia la conversazione” a “Conversazione in corso”.
 - Il bottone “Avvia la conversazione” diventa non più cliccabile.
 - Il bottone “Stop” viene abilitato, diventando cliccabile.
 - Il bottone “Recap” viene disabilitato.
 - I vari form relativi ai dati vengono disabilitati, in modo tale che durante la conversazione non sia possibile introdurre manualmente (inserimento tramite tastiera) i dati.

Questa scelta è stata presa per i seguenti motivi:

- * Chiarezza nell’interfaccia utente: disabilitando i form quando non sono in uso, si fornisce agli utenti un’indicazione chiara su quale modalità di inserimento dati sia attualmente attiva. Questo può ridurre la confusione e migliorare l’esperienza utente complessiva.
- * Prevenzione degli errori: separare le modalità di inserimento dati può aiutare a prevenire errori e ridurre i conflitti. Questo fa sì che l’utente utilizzi contemporaneamente il riconoscimento vocale e l’inserimento manuale tramite tastiera, riducendo la possibilità di errori o confusione.

- * Focus sull'input corrente: disabilitando i form non utilizzati, si può concentrare l'attenzione dell'utente sull'input corrente. Ciò può aumentare la precisione e l'efficienza, specialmente quando si utilizza il riconoscimento vocale, consentendo all'utente di concentrarsi esclusivamente sull'input vocale senza distrazioni.
- Viene resa visibile una gif che segnala all'utente che la conversazione ha avuto inizio. Più nel dettaglio, ogni qual volta comparirà questa animazione all'interno dell'applicativo, l'utente sarà in ascolto di istruzioni da parte dell'applicativo stesso su che operazione eseguire e che di che dati si tratta.

N.	CAND. SIND.	Voti validi	Voti solo Sindaco
1	MARCO GUIDI	1	1
2	CESARE MARIA RAGAGLINI	2	0
3	FRANCESCO PERSIANI	1	1
4	ROMOLO ENZO RICCI	15	5
5	ANDREA BAROTTI	1	1
6	MARCO LENZONI	5	0
7	DANIELA BENNATI	12	0
8	GUIDO MUSSI	3	0
TOTALI:		40	8

Figura 6.4: Inizio conversazione sezione Raggruppamenti

- La schermata riportata nella figura 6.4 fa riferimento al sub-evento elettorale “Raggruppamenti”, dove dunque si chiede all'utente di inserire, per ciascun candidato sindaco, i voti validi e i voti espressi al solo sindaco.

Al fine di guidare l'utente nell'inserimento dei dati, le istruzioni vocali saranno: “*Candidato sindaco numero #, voti validi?*” e “*Candidato sindaco numero #, voti solo sindaco?*”, dove # corrisponde al numero associato al candidato sindaco (ad esempio: #1 Marco Guidi, #2 Cesare Maria Ragaglini, ...).

Si è deciso di far riferimento ai vari candidati sindaco tramite numero piuttosto che tramite nome per diversi motivi:

- * Semplicità: utilizzando numeri per identificare i candidati, si semplifica il processo di selezione per l'utente. È più diretto rispetto alla lettura di nomi e cognomi più lunghi, riducendo il rischio di errori di pronuncia o comprensione.

- * Risparmio di tempo: poiché i numeri sono più concisi dei nomi, l'utente può completare l'inserimento dei dati più rapidamente, specialmente se ci sono molti candidati su cui dover iterare.
 - * Universalità: i numeri sono universalmente compresi e facilmente identificabili, indipendentemente dalla lingua o dalla cultura dell'utente. Ciò rende il sistema più accessibile e utilizzabile da parte di una vasta gamma di utenti.
 - * Minore carico cognitivo: Riducendo la quantità di informazioni da elaborare (in questo caso, nomi e cognomi), si riduce il carico cognitivo sull'utente, facilitando la comprensione e la decisione.
 - * Scalabilità: se la lista dei candidati sindaco è soggetta a cambiamenti o aggiornamenti, utilizzare numeri per identificarli rende il sistema più scalabile, in quanto non è necessario modificare il testo del sistema di sintesi vocale ogni volta che si aggiungono o rimuovono candidati.
- È possibile notare nella figura 6.4, una riga azzurra in corrispondenza della prima riga (quella riferita a #1 Marco Guidi). Ciò sta ad indicare che la conversazione avrà inizio da quella riga, dunque è possibile cliccare sulla riga (e quindi sul candidato sindaco) da cui si vuole partire per l'inserimento dei dati tramite riconoscimento vocale.

Le motivazioni alla base dell'introduzione di questa ulteriore feature sono:

- * Navigazione rapida: gli utenti possono passare rapidamente da un candidato all'altro senza dover aspettare necessariamente di dover inserire i dati anche per tutti i candidati precedenti. Questo risparmia tempo e rende l'interazione più efficiente, in quanto è possibile che l'utente voglia inserire i dati in più passaggi oppure che voglia interrompere la conversazione e ripartire da dove aveva finito.
 - * Precisione: essendo in grado di selezionare direttamente il candidato desiderato, si riduce il rischio di errori o fraintendimenti nell'identificazione del candidato.
 - * Esperienza utente migliorata: offrendo un'interfaccia che consente agli utenti di iniziare la conversazione da qualsiasi candidato con un semplice clic, si migliora l'esperienza complessiva dell'utente, rendendo l'applicazione più intuitiva e facile da usare.
 - * Personalizzazione: gli utenti possono iniziare la conversazione esplorando i candidati in qualsiasi ordine desiderino, in base ai propri interessi o alle proprie preferenze. Ciò fornisce un maggiore controllo e personalizzazione dell'esperienza.
- Una volta ricevuta l'istruzione vocale da parte dell'applicativo, seguirà un ulteriore aggiornamento della schermata:

- Il testo all’interno del bottone cambia da *“Conversazione in corso”* a *“Registra la tua risposta!”*.
- Viene emesso un audio per indicare all’utente che è il momento di parlare e registrare la propria risposta.
- Al posto della gif precedente ne compare un’altra, la quale indica all’utente che è il momento di parlare e registrare la propria risposta.

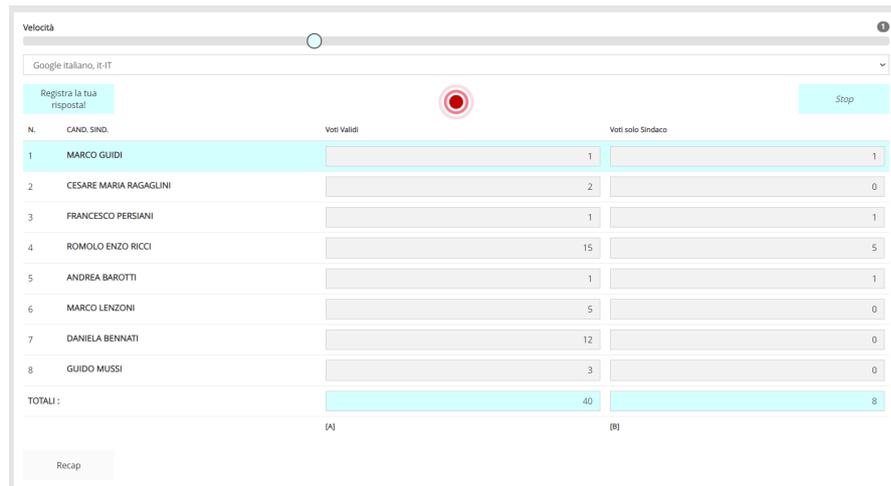


Figura 6.5: Registra la risposta sezione Raggruppamenti

- A questo punto della conversazione, si apre una finestra temporale (la finestra temporale dura circa 10 secondi) entro la quale l’utente è chiamato a pronunciare il numero relativo all’informazione richiesta.

A seconda della risposta data dall’utente si possono aprire diverse strade:

- L’utente non parla: se l’utente non parla o pronuncia qualcosa che viene riconosciuto dall’applicativo come “silenzio”, la conversazione viene interrotta e la schermata tornerà nello stato iniziale, così come descritto nella figura 6.2.
- L’utente pronuncia qualcosa che non viene percepito come un numero: nel momento in cui l’applicativo capta dell’informazione, ma non viene recepita come numero, invita l’utente a reinserire il dato, senza interrompere la conversazione. Perciò l’utente sentirà la frase *“Ho capito #, non è un numero”* accompagnata dagli stessi elementi visivi e sonori descritti in precedenza. A questo punto verrà ripetuta l’istruzione e la conversazione tornerà nello stato precedente alla risposta data.

– L’utente pronuncia un numero: nel caso in cui l’utente dovesse pronunciare qualcosa che viene recepito come numero dall’applicativo, si aprono tre percorsi distinti, i quali si basano su una proprietà dell’oggetto “*SpeechRecognitionResult*” (l’oggetto “risposta” che si ottiene), ossia la *confidence*. La *confidence* è una proprietà appunto che restituisce una stima numerica (valori tra 0 e 1) di quanto sia sicuro che il sistema di riconoscimento vocale abbia recepito correttamente quanto detto dall’utente. A seconda del valore della *confidence*:

- * *confidence* ≥ 0.8 : il dato numerico viene inserito automaticamente all’interno del form a cui fa riferimento e viene emesso un suono che possa far capire all’utente che il dato è stato inserito correttamente. Da un punto di vista grafico, non ci sono differenze tra un numero inserito manualmente o un numero inserito tramite riconoscimento vocale con una *confidence* alta (≥ 0.8 per l’appunto).

- * *confidence* ≥ 0.5 and *confidence* < 0.8 : il dato numerico non viene inserito in automatico all’interno del form, bensì l’applicativo richiederà all’utente una conferma circa il numero che ha recepito. Perciò seguirà un’ulteriore istruzione vocale, ossia “*Hai detto #, confermi o rifiuti?*”. A questo punto sarà nuovamente il turno dell’utente, il quale potrà rispondere:

- “*Confermo*”: dopo aver recepito tale comando, il dato numerico viene inserito all’interno del form corrispondente, emettendo un suono che faccia capire all’utente che il dato è stato inserito correttamente. Da un punto di vista grafico, il contorno del form verrà colorato di arancione (codice esadecimale #FFA500).

Si è optato per una soluzione del genere al fine di fornire un feedback visivo immediato all’utente sul livello di attendibilità del dato inserito, il che può essere utile in una fase di riepilogo e controllo dei dati, prima di confermarli.

N.	CAND. SIND.	Voti Validi	Voti solo Sindaco
1	MARCO GUIDI	16	1

- “*Rifiuto*”: dopo aver recepito tale comando, il dato numerico verrà scartato e l’applicativo inviterà l’utente a reinserire il dato, senza interrompere la conversazione. A questo punto verrà ripetuta l’istruzione e la conversazione tornerà nello stato precedente alla risposta data.

- Tutto ciò che è diverso dai comandi precedenti verrà riconosciuto come comando non valido. L'applicativo inviterà l'utente a reinserire il dato, senza interrompere la conversazione. Perciò l'utente sentirà la frase *“Ho capito #, non è un comando valido”* accompagnata dagli stessi elementi visivi e sonori descritti in precedenza. A questo punto verrà ripetuta l'istruzione e la conversazione tornerà nello stato precedente alla risposta data.
Si è deciso di usare comandi come “Confermo” e “Rifiuto” (piuttosto delle opzioni più scontate “Sì” e “No”) in quanto ci si è resi conto che il modulo di Speech Recognition fa più fatica a riconoscere parole molto brevi (composte ad esempio da due o tre lettere).
- * *confidence < 0.5*: l'applicativo inviterà l'utente a reinserire il dato, senza interrompere la conversazione. Perciò l'utente sentirà la frase *“Non ho capito, ripeti per favore”* accompagnata dagli stessi elementi visivi e sonori descritti in precedenza. A questo punto verrà ripetuta l'istruzione e la conversazione tornerà nello stato precedente alla risposta data.

Questi valori di confidence sono stati studiati a seguito di diversi test, dove si andava a vedere quale fosse il valore ottenuto a seconda delle parole pronunciate (sia numeri, sia comandi riconoscibili).

Inizialmente si era pensato di non introdurre dei comandi di conferma oppure di usarli a prescindere dall'attendibilità del risultato. Anche qui, facendo diversi test, si è trovato questo compromesso: si chiede conferma solo se la confidence è relativamente bassa, in modo tale da non rallentare troppo la conversazione, la quale potrebbe risultare sì più affidabile nel riconoscimento, ma sicuramente meno efficiente da un punto di vista di tempo richiesto per poter inserire i dati, pensando soprattutto ai sub-eventi elettorali con molti candidati.

- Una volta inserito con successo un dato, il processo visto fino ad ora viene ripetuto, iterando sulle varie righe, a differenza che, se non viene interrotta la conversazione (tramite clic del bottone “Stop” oppure tramite silenzio), non sarà necessario cliccare il bottone “Avvia la conversazione”, bensì essa continuerà in automatico fino a che tutti i dati richiesti non verranno inseriti.

Dunque seguirà un loop di operazioni, dove si alterneranno le fasi di “istruzione” da parte dell'applicativo e fasi di “registrazione” da parte dell'utente, accompagnate dai vari elementi audio e video citati precedentemente.

Ogni qual volta si finirà con l'inserimento dei dati relativi ad una singola riga, si passerà a quella successiva, con il conseguente aggiornamento automatico dei

counter dei voti totali a fine sezione e della riga corrente tramite la colorazione in azzurro (colore #CCFFFF).

Nel caso dei “Raggruppamenti”, si è visto che per ogni riga è necessario inserire due dati (voti validi e voti relativi al solo candidato sindaco), dunque l’operazione va ripetuta due volte per riga, con gli stessi accorgimenti. Inoltre in questo sub-evento è previsto anche l’inserimento dei dati relativi alle schede (figura 6.3): trattandosi di soli tre dati da inserire (schede bianche, schede nulle e schede contestate) si è deciso di imporre all’utente di inserire tutti e tre i dati ogni volta, mettendo a disposizione un tasto apposito per avviare la conversazione.

Per quanto riguarda invece invece gli altri sub-eventi elettorali la situazione è leggermente diversa:

- Caso delle “Affluenze”: si tratta del caso più semplice, in quanto vi è un’unica “riga” di dati da inserire, dunque non sarà necessario iterare né tra le righe, né tra le colonne:

	MASCHI	FEMMINE	TOTALI
ISCRITTI	560	609	1169
VOTANTI	2	2	4
% VOTANTI	0,36%	0,33%	0,34%
TUTTE LE SEZIONI	27592	29833	57425

Figura 6.6: Interfaccia di inserimento dati sezione Affluenze

Infatti si è scelto che, nel caso delle “Affluenze”, non sarà possibile scegliere da che dato partire per l’inserimento, in quanto per ogni affluenza ci sono solamente tre dati da inserire per volta (voti da parte dei votanti maschi, voti da parte dei votanti femmine, voti totali dei votanti). Si è dunque preferito dover imporre all’utente l’inserimento di tutti e tre i dati, trattandosi di un numero veramente esiguo, in modo tale da ridurre al minimo la possibilità di errore.

- Caso delle “Liste”: in questo caso vi è solo un dato per riga (i voti relativi alla lista), di conseguenza valgono gli stessi accorgimenti fatti per i “Raggruppamenti”. Sarà dunque possibile scegliere da che riga partire, ogni volta che una riga verrà completata il counter dei voti totali

verrà aggiornato e si passerà alla riga successiva, fino all'esaurimento dell'iterazione.

N.	LISTE	Voti
1	FRANCESCO PERSIANI SINDACO	1
2	LEGA	2
3	GIORGIA MELONI FRATELLI D'ITALIA	3
4	ANDREA BAROTTI SINDACO	4
5	EVANGELISTI PER MASSA 2023	5
6	MOVIMENTO 5 STELLE 2050	6
7	CESARE RAGAGLINI SINDACO	7
8	LIBERALI E RIFORMISTI N PSI	8
9	FORZA ITALIA	9
10	PD PARTITO DEMOCRATICO	10

Figura 6.7: Interfaccia di inserimento dati sezione Liste

Per quanto riguarda invece la sezione delle schede, essa sarà accessibile in sola lettura, in quanto i dati vengono raccolti dal sub-evento elettorale precedente, dunque non sarà necessario inserire alcun dato. Di conseguenza, sia il bottone “Avvia la conversazione”, sia il bottone “Stop” risultano disabilitati. Si è scelto comunque di metterli al fine di mantenere una coerenza grafica col resto delle pagine.

Schede Bianche [D]	2
Schede nulle [E]	0
Schede Contestate [F]	0
In totale	2
Voti solo Cand. [B]	8
Totale [A+B+D+E+F]	124

Figura 6.8: Interfaccia di inserimento schede sezione Liste

- Caso delle “Preferenze”: in questo caso valgono le stesse considerazioni fatte per le “Liste”, in quanto anche per le “Preferenze” si ha un unico dato da inserire per riga (voti per ciascuna preferenza), mentre la sezione schede è sostituita da una sorta di sezione riepilogo (contenente il numero dei voti lista, il numero di preferenze esprimibili e il numero massimo di preferenze), la quale, anche in questo caso, sarà accessibile in sola lettura, dunque non sarà necessario inserire alcun dato. Di conseguenza,

sia il bottone “Avvia la conversazione”, sia il bottone “Stop” risultano disabilitati. Si è scelto comunque di metterli al fine di mantenere una coerenza grafica col resto delle pagine.

Numero	Nominativo	Voti
1	GIOVANNI DETTO GIOIETTA GIUSTI	0
2	LODOVICO ANDREAZZINI	0
3	ELISABETTA DETTA CICCÌ BARANDONI	0
4	MARY BARATTINI	0
5	LAURA BASTERI	0
6	PIER LUIGI BASTERI	0
7	LUIGI BERNACCA	0
8	FABRIANO BERTI	0
9	PIERPAOLO BERTILORENZI	0
10	MONICA BERTONERI	0

Figura 6.9: Interfaccia di inserimento dati sezione Preferenze

Voti Lista	1
Preferenze esprimibili	2
Numero massimo di preferenze	2

Figura 6.10: Interfaccia di riepilogo dati sezione Preferenze

- Cliccando il bottone “Recap” verrà riprodotto un riepilogo, il quale verrà pronunciato dal sintetizzatore vocale, accompagnato dalla sua gif corrispondente: tale funzionalità è stata pensata a supporto dell’utente, nel momento in cui quest’ultimo volesse controllare i dati prima di salvarli. Più precisamente:
 - Il bottone “Recap” è abilitato (ossia è cliccabile, caratterizzato graficamente dal colore azzurro e dal puntatore a forma di mano quando si trascina il mouse su di esso) nel momento in cui non è attiva alcuna conversazione tra l’utente e l’applicativo. Altrimenti infatti, il tasto verrà disabilitato, assumendo il classico colore grigio che suggerirà all’utente che quella funzionalità non è al momento disponibile.
 - Nel caso in cui non sia stato ancora inserito nessun dato, cliccando il bottone “Recap” verrà riprodotto un messaggio di avviso per l’utente:
 - * Nel caso delle “Affluenze” i possibili messaggi sono:

- “*Devi prima inserire tutti i dati*”, nel caso in cui non siano stati inseriti sia i votanti maschi, sia i votanti femmine, sia i votanti totali.
- “*Votanti maschi mancanti*”, nel caso in cui non siano stati inseriti i dati relativi ai votanti maschi, ma femmine e totali sì.
- “*Votanti femmine mancanti*”, nel caso in cui non siano stati inseriti i dati relativi ai votanti femmine, ma maschi e totali sì.
- “*Votanti totali mancanti*”, nel caso in cui non siano stati inseriti i dati relativi ai votanti totali, ma maschi e femmine sì.
- * Nel caso dei “Raggruppamenti” i possibili messaggi sono:
 - “*Voti validi e solo sindaco mancanti*”, nel caso in cui non siano stati inseriti sia i voti validi sia i voti solo sindaco.
 - “*Voti validi mancanti*”, nel caso in cui non siano stati inseriti i dati relativi ai voti validi, ma voti solo sindaco sì.
 - “*Voti solo sindaco mancanti*”, nel caso in cui non siano stati inseriti i dati relativi ai voti solo sindaco, ma voti validi sì.
 - Non è stato pensato un messaggio di avviso nel caso delle schede in quanto si è assunto che potrebbe verificarsi lo scenario in cui effettivamente ci siano 0 schede bianche, 0 schede nulle e 0 schede contestate.
- * Nel caso delle “Liste”: avendo a che fare con un unico tipo di dato (voti relativi alle liste) è stato pensato un solo messaggio di avviso, “*Voti mancanti*”.
- * Nel caso delle “Preferenze”: come sopra, avendo a che fare con un unico tipo di dato (voti relativi alle liste) è stato pensato un solo messaggio di avviso, “*Voti mancanti*”.
- Nel caso in cui siano stati già inseriti effettivamente dei dati (ossia nel caso in cui il numero totale dei dati inseriti è diverso da 0), seguirà un riepilogo dei dati:
 - * Nel caso delle “Affluenze”: “*# votanti maschi, # votanti femmine, # votanti totali*”.
 - * Nel caso dei “Raggruppamenti”: “*Candidato sindaco numero 1 voti validi #, voti solo sindaco #; Candidato sindaco numero 2, voti validi #, voti solo sindaco #; ...*”.
 - * Nel caso delle “Liste”: “*Lista numero 1 voti #, lista numero 2 voti #, ...*”.
 - * Nel caso delle “Preferenze”: “*Candidato numero 1 voti #, candidato numero 2 voti #, ...*”.

A prescindere dal numero di voti già inseriti all'interno della piattaforma, dopo aver cliccato il bottone "Recap" seguirà un aggiornamento dell'interfaccia:

- * Il testo all'interno del bottone cambia da "Avvia la conversazione" a "Recap in corso".
- * Il bottone "Avvia la conversazione" diventa non più cliccabile.
- * Il bottone "Stop" viene abilitato, diventando cliccabile.
- * Il bottone "Recap" diventa non più cliccabile.
- * I vari form relativi ai dati vengono disabilitati, in modo tale che durante il riepilogo non sia possibile modificare i dati.

N.	CAND. SIND.	Voti Validi	Voti solo Sindaco
1	MARCO GUIDI	1	1
2	CESARE MARIA RAGAGLINI	2	0
3	FRANCESCO PERSIANI	1	1
4	ROMOLO ENZO RICCI	15	5
5	ANDREA BAROTTI	1	1
6	MARCO LENZONI	5	0
7	DANIELA BENNATI	12	0
8	GUIDO MUSSI	3	0
TOTALI:		40	8

Figura 6.11: Riepilogo dei dati inseriti

- Cliccando il bottone "Stop", indipendentemente che esso venga cliccato per interrompere le istruzioni vocali per l'inserimento dei dati/schede o per interrompere l'inserimento dei dati vero e proprio, seguirà il seguente aggiornamento:
 - Viene resettato il timeout di riconoscimento del silenzio.
 - Scompaiono eventuali gif a schermo (sia quella di istruzione, sia quella di inserimento).
 - Il bottone "Avvia la conversazione" viene resettato, diventando cliccabile. (stesso discorso vale per il bottone corrispondente per la sezione delle schede, ove presente.)
 - Il bottone "Recap" viene resettato, diventando cliccabile. (stesso discorso vale per il bottone corrispondente per la sezione delle schede, ove presente.)

- Il bottone “Stop” viene disabilitato, diventando non più cliccabile. (stesso discorso vale per il bottone corrispondente per la sezione delle schede, ove presente.)
- Si interrompe il riconoscimento vocale.
- Si interrompe la sintesi vocale.
- I vari form relativi ai dati vengono riabilitati.

Cliccando il bottone “Stop”, si ritorna dunque allo stato iniziale, così come descritto nella figura 6.2.

Capitolo 7

Valutazione

A conclusione di questa tesi, si è deciso di eseguire dei test di valutazione e di usabilità del prototipo realizzato, di cui ampiamente discusso nel capitolo precedente.

Un test di usabilità è un metodo di valutazione dei prodotti, come siti web, applicazioni software o dispositivi, che mira a valutare quanto sia facile e intuitivo utilizzare il prodotto da parte degli utenti. Durante un test di usabilità, ai partecipanti vengono assegnati compiti specifici da completare utilizzando il prodotto, mentre gli osservatori registrano le azioni degli utenti, le loro reazioni e le difficoltà incontrate. L'obiettivo principale di un test di usabilità è identificare eventuali problemi di usabilità nel design del prodotto e raccogliere feedback utili per migliorarlo. Questo tipo di test può coinvolgere una varietà di metodi e tecniche, come interviste, questionari, analisi delle metriche di utilizzo e osservazioni dirette degli utenti durante l'interazione con il prodotto.

Per poter svolgere questo test è stata necessaria dunque la partecipazione di alcuni utenti che abbiano dimestichezza nel dominio dell'applicativo su cui si è realizzata questa estensione, ossia dei moderatori e degli osservatori, il cui compito sarà quello di monitorare e fornire le informazioni all'utente durante lo svolgimento.

7.1 Pianificazione

Per condurre i vari test, è stato elaborato uno script per la valutazione del prototipo.

Poiché il prodotto è destinato all'uso da parte degli amministratori comunali, il test è stato condotto specificamente con questa categoria di utenti.

Si è deciso di testare il prototipo con cinque persone diverse. Il numero scelto non è casuale, bensì si basa su numerosi studi che dimostrano come bastino cinque persone per riuscire a individuare la maggior parte delle problematiche nascoste in un prototipo.

Tra questi studi si distingue quello di Jakob Nielsen [19], secondo il quale cinque utenti sono sufficienti per individuare la maggior parte dei problemi di usabilità in un prodotto per diverse ragioni:

- Legge di resoconto asintotico: Nielsen ha osservato che l'incremento del numero di utenti testati porta a un ritorno decrescente nell'individuare nuovi problemi di usabilità. Ciò significa che mentre i primi utenti possono rilevare una vasta gamma di problemi, ogni utente aggiuntivo tenderà a identificare problemi simili a quelli già scoperti.
- Ripetibilità dei problemi: molti problemi di usabilità sono ripetitivi e si verificano in più utenti. Pertanto, una volta che un problema viene identificato da alcuni utenti, è probabile che venga riscontrato anche dagli altri.
- Economicità e rapidità: condurre test con un numero ridotto di utenti è più economico e richiede meno tempo rispetto a test con un campione più ampio. Questo consente di ottenere risultati utili in modo più efficiente.
- Priorità dei problemi: identificare anche solo alcuni problemi principali può essere sufficiente per migliorare notevolmente l'usabilità del prodotto. Concentrandosi sui problemi più critici individuati dai cinque utenti, è possibile apportare miglioramenti significativi con relativamente pochi sforzi.

I test si sono svolti online tramite video-chiamata Zoom, in quanto gli amministratori non facevano parte degli stessi comuni italiani.

Al fine di favorire lo svolgimento dei test, i vari partecipanti sono stati affiancati da un facilitatore, il cui compito era quello di descrivere le attività e le azioni da svolgere, oltre che prendere nota dei loro comportamenti e ragionamenti (gli utenti sono stati invitati a esprimere ad alta voce i loro pensieri durante il test).

Nel caso in cui un partecipante avesse incontrato difficoltà nell'eseguire un'azione, il facilitatore sarebbe intervenuto per fornire assistenza e aiuto.

Nello script ideato del test, sono state delineate sette attività (che sono poi riassunte nella tabella 7.1), ciascuna delle quali sarà associata a una metrica specifica per calcolare il successo o il fallimento del test in modo matematico. Dopo il completamento del test, a ciascun partecipante è stato somministrato un questionario SUS (System Usability Scale) [20], seguito da alcune domande aggiuntive mirate all'individuazione di eventuali problemi riscontrati, criticità e/o funzionalità da poter introdurre.

In breve, il questionario SUS è uno strumento ampiamente utilizzato per valutare la percezione degli utenti sulla usabilità di un sistema, un prodotto o un'interfaccia utente. È composto da una serie di domande standardizzate che misurano la percezione dell'usabilità da parte degli utenti attraverso valutazioni soggettive. Il questionario SUS di solito include domande riguardanti vari aspetti dell'usabilità,

come l'apprendimento dell'utente, l'efficienza nell'utilizzo del sistema, la facilità di memorizzazione delle funzioni e la soddisfazione generale dell'utente con l'esperienza d'uso. Le risposte degli utenti vengono quindi analizzate per ottenere un punteggio complessivo che indica il livello di usabilità del sistema o del prodotto.

7.2 Attività

Al fine di valutare il prototipo, è stato pensato un approccio basato su diverse attività:

- **Attività 1:** come prima attività, viene richiesto inizialmente all'utente di selezionare i vari sub-eventi elettorali. L'ambiente viene predisposto in modo tale che il login all'interno della piattaforma sia già stato effettuato, così come la selezione del comune interessato. All'interno di questa attività si vuole far dare una panoramica all'utente di come sono state realizzate le varie schermate. Questa attività viene considerata "completata con successo" se l'utente è in grado di individuare gli elementi con cui dovrà interagire per gestire la conversazione (senza ancora interagirci attivamente).
- **Attività 2:** in questa attività viene richiesto all'utente di interagire con gli elementi "preliminari alla conversazione", ossia gli slider di "velocità" e "tono di voce" e con il menu a tendina di "selezione voce".
- **Attività 3:** in questa attività viene richiesto all'utente di dare inizio alla conversazione (per semplicità gli si chiede di iniziare dal sub-evento elettorale più semplice, ossia quello delle "Affluenze"). Qui si terrà d'occhio se l'utente inizierà a inserire i dati o lascerà scadere il timeout di riconoscimento.
- **Attività 4:** in questa attività viene richiesto esplicitamente all'utente di portare avanti la conversazione (sempre nel caso delle "Affluenze"). Qui si terrà d'occhio se l'utente riuscirà a inserire tutti e tre i dati necessari (maschi, femmine e totali) e come reagiranno qualora l'applicativo dovesse chiedere conferma della risposta.
- **Attività 5:** in questa attività ci si sposta su un altro sub-evento elettorale, ossia quelle delle "Liste" (si sceglie questo sub-evento in quanto un compromesso di difficoltà tra quello delle "Affluenze", il più semplice, e quello dei "Raggruppamenti", il più difficile e in quanto presente la feature di scorrimento tra le righe). A questo punto viene chiesto all'utente di portare avanti la conversazione. Qui si terrà d'occhio se l'utente noterà la colorazione delle righe durante l'iterazione.

- **Attività 6:** in questa attività viene richiesto all'utente di interrompere la conversazione mentre quest'ultima è in atto (qui si terrà conto se l'utente cliccherà il bottone "Stop" o lascerà scadere il timeout di riconoscimento). A questo punto si chiederà di partire da una riga in particolare (ad esempio a partire dalla riga n. 3), per vedere se verrà spontaneo cliccare sulla riga.
- **Attività 7:** come attività finale, viene richiesto all'utente quello di eseguire un riepilogo dei dati tramite il bottone "Recap". Qui si noterà se all'utente verrà la curiosità di aumentare la velocità della voce, trattandosi di un'operazione molto lunga, oppure se verranno riprodotti dei messaggi di avviso nel caso in cui dovessero mancare dei dati.

7.3 Svolgimento

Per condurre il test, sono stati utilizzati diversi strumenti: un computer contenente il prototipo funzionante in locale, un altro computer per il partecipante, e Zoom come mezzo di comunicazione.

Per poter dare pieno accesso del prototipo al partecipante, è stato creato un link attraverso il quale potervi accedere tramite browser (Google Chrome), per poi richiederne la condivisione dello schermo per poter seguire e monitorare le operazioni passo passo.

Il test non è stato registrato, ma le azioni e le osservazioni vocali dei partecipanti sono state annotate manualmente su carta: è stato chiesto ad essi di ragionare ad alta voce, in modo tale da poter conoscere il *modus operandi* dietro le azioni e decisioni di ciascun utente.

I partecipanti, provenienti da cinque comuni diversi (che rimarranno anonimi per motivi di privacy), sono qui identificati con i codici da P01 a P05. Tutti i partecipanti hanno una buona familiarità sia con il mondo web sia con il dominio sotto esame e comprendono la sfida che il prototipo si propone di affrontare.

Le attività selezionate per lo svolgimento di questo test sono mirate a coprire e valutare tutte le funzionalità dei due moduli, quello di riconoscimento vocale e di sintesi vocale.

7.4 Risultati

- **Attività 1:** tutti i partecipanti sono riusciti a individuare i vari elementi di interazione.
- **Attività 2:** anche questa attività, essendo molto semplice, è stata portata a buon fine da tutti i partecipanti. Durante questa fase, i partecipanti P02 e P05 hanno chiesto di poter verificare le conseguenze delle operazioni fatte

(slider velocità e tono, voce selezionata), mentre il partecipante P01 ha cliccato spontaneamente sul bottone “Recap” per verificarle autonomamente.

- **Attività 3:** in questa attività 4 partecipanti su 5 hanno risposto alle istruzioni date dall’applicativo dopo l’emissione del segnale acustico, a differenza del partecipante P03, il quale ha lasciato scadere il timeout del riconoscimento vocale, notando il reset dell’interfaccia.
- **Attività 4:** A questo punto, tutti e 5 i partecipanti sono stati in grado di portare a termine la conversazione (inserire dunque i 3 dati richiesti). Per alcuni dei partecipanti sono stati necessari più tentativi (per un massimo di 3 nel caso del partecipante P04). A un certo punto della conversazione, a tutti i partecipanti è stata chiesta la conferma dell’inserimento di almeno un dato, ad eccezione del partecipante P02, al quale è stato poi introdotta questa caratteristica per dare un quadro completo, allineato a quello degli altri partecipanti.
- **Attività 5:** questa è l’attività più “complessa” tra quelle previste. Solo 2 partecipanti su 5 sono riusciti a portare avanti la conversazione fino alla fine senza interruzioni. Un partecipante ha interrotto la conversazione, notando però la caratteristica della riga selezionata, mentre gli altri due partecipanti non si sono accorti di questa funzionalità.
- **Attività 6:** tutti e 5 i partecipanti sono stati in grado di interrompere la conversazione tramite il clic sul bottone “Stop” senza che gli venisse specificato. Allo stesso modo sono stati in grado di riprendere la conversazione a partire dalla riga richiesta. Anche chi non si era accorto della possibilità di cliccare sulle righe, è riuscito ad eseguire l’operazione richiesta durante questa fase.
- **Attività 7:** durante questa attività finale, 2 candidati su 5 hanno interrotto il riepilogo dei dati per poter aumentare la velocità della sintesi vocale (il sub-evento “Liste” del test prevedeva 20 liste candidate), un candidato ha modificato la velocità mentre il riepilogo era in corso (non notando dunque alcuna differenza), mentre i restanti 2 candidati hanno ascoltato l’intero riepilogo senza lamentare particolari.

Nel complesso, i risultati dei test sono stati positivi. Tutti i partecipanti, chi con più, chi con meno difficoltà, sono riusciti a portare a termine tutte le funzionalità previste dall’applicativo.

7.5 Possibili modifiche

In base ai risultati dei test e ai feedback dei partecipanti, emerge che l'applicativo è generalmente semplice e intuitivo da utilizzare. Tuttavia, ci sono ancora alcune modifiche che potrebbero essere apportate.

Di seguito alcuni dei suggerimenti proposti dai partecipanti:

- L'introduzione di un segnale acustico nel momento in cui scade il timeout per il riconoscimento vocale, il quale può suggerire più facilmente all'utente che è il caso di riavviare la conversazione.
- Lo slider per la gestione del tono di voce è stato per lo più ritenuto superfluo, dunque sotto suggerimento dei partecipanti potrebbe essere rimosso.
- Sarebbe utile, durante il riepilogo dei dati, colorare iterazione dopo iterazione la riga sotto analisi, in modo tale da dare all'utente un riferimento visivo durante l'ascolto, il che potrebbe facilitare il confronto dei dati inseriti nella piattaforma e quelli sotto mano.
- Sarebbe utile, durante il riepilogo dei dati, permettere di selezionare la riga da cui far partire l'operazione, come già avviene durante l'inserimento dei dati.
- Sempre durante la fase di riepilogo dei dati, sarebbe più comodo far sì che sia possibile modificare la velocità della voce sintetizzata senza dover necessariamente interrompere.

Titolo	Descrizione	Criteri di successo	Metrica
Attività 1	Individuazione degli elementi di interazione	Il partecipante riesce a individuare gli slider, il menu di selezione della voce, i bottoni di gestione conversazione	Successo (1) o fallimento (0)
Attività 2	Interazione preliminare con gli elementi di set up della conversazione	Il partecipante interagisce con gli slider di velocità e tono di voce, col menu di selezione della voce	1 punto per ogni interazione, per un massimo di 3
Attività 3	Primo avvio di conversazione	Il partecipante da inizio alla conversazione, inserendo i primi dati	Successo (1) o fallimento (0)
Attività 4	Conversazione nel sub-evento "Affluenze"	Il partecipante riesce a portare a termine la conversazione nel sub-evento "Affluenze"	1 punto per ogni dato inserito, per un massimo di 3
Attività 5	Conversazione nel sub-evento "Liste"	Il partecipante riesce a portare a termine la conversazione nel sub-evento "Liste", individuando la possibilità di riprendere da un punto specifico	Successo (1) o fallimento (0)
Attività 6	Interruzione della conversazione	Il partecipante riesce ad interrompere la conversazione nel sub-evento "Liste"	Successo (1) o fallimento (0)
Attività 7	Riepilogo dei dati inseriti	Il partecipante riesce ad eseguire il riepilogo dei dati nel sub-evento "Liste", modificando la velocità	Successo (1) o fallimento (0)

Tabella 7.1: Attività per la valutazione

Partecipante	A1	A2	A3	A4	A5	A6	A7	Successo medio partecipante
P01	1	3/3	1	3/3	1	1	0	≈ 85%
P02	1	3/3	1	3/3	1	1	1	100%
P03	1	3/3	0	3/3	0	1	1	≈ 70%
P04	1	3/3	1	3/3	1	1	1	100%
P05	1	3/3	1	3/3	0	1	0	≈ 70%
Successo medio attività	100%	100%	80%	100%	60%	100%	40%	

Tabella 7.2: Tabella dei risultati: esiti delle attività per la valutazione

Capitolo 8

Conclusioni

L'obiettivo del progetto di tesi era quello di creare un'estensione dell'applicativo "Eleweb", all'interno della quale integrare le funzionalità di riconoscimento vocale e di sintesi vocale.

Si è partiti dunque dallo studio del dominio, per comprendere appieno il contesto in cui si trova e le esigenze delle amministrazioni comunali italiane. Si è dunque effettuata un'analisi su come Eleweb gestisca già la raccolta dei dati e su quali sub-eventi supporti.

Sulla base delle informazioni raccolte, è iniziata la fase di progettazione del sistema, in cui si è dapprima ideato un prototipo di interfaccia che potesse ospitare le funzionalità richieste, per poi andare alla ricerca delle tecnologie che potessero soddisfare i requisiti identificati, optando in fine per "Web Speech API", un'interfaccia di programmazione delle applicazioni che consente ai browser web di accedere alle funzionalità di sintesi vocale e di riconoscimento vocale.

Si sono dapprima analizzati individualmente i due moduli scelti, "Speech Recognition" e "Speech Synthesis", studiandone l'implementazione, per poi integrarli all'interno di un prototipo finale.

Si è così ottenuto un prodotto in grado di permettere l'inserimento dei dati elettorali all'interno dell'applicativo tramite input vocale dell'utente, il quale sarà guidato nel corso della conversazione tramite istruzioni fornite dal modulo di sintesi vocale.

Sono stati condotti infine dei test di usabilità del prototipo realizzato, sottoponendoli a cinque diversi amministratori comunali. Ai partecipanti è stato chiesto di utilizzare la piattaforma e di valutarne le problematiche e i punti di forza, al fine di intervenire dove fossero emerse criticità. Nel complesso, le interviste hanno avuto un riscontro positivo, con modifiche previste in base alle esigenze specifiche di ciascun comune e eventuali integrazioni di funzionalità. Inoltre, è stato somministrato un questionario SUS a tutti i partecipanti, che ha ottenuto una valutazione media di 89, indicando un risultato favorevole.

Si può concludere affermando che è stata sviluppata un'estensione dell'applicativo in grado di semplificare e velocizzare uno dei punti critici che caratterizza un evento elettorale, ossia quello di riportare i dati dai fonogrammi (i cosiddetti “foglietti”) alla piattaforma Eleweb, dalla carta al digitale.

8.1 Sviluppi futuri

In seguito alla conclusione del progetto di tesi, tale estensione di Eleweb è stata proposta in via esplorativa al Comune di Torino.

In vista di una presentazione ufficiale, è stata proposta l'idea di utilizzare in combinazione alle tecniche di riconoscimento e sintesi vocale, anche un supporto OCR, in modo tale da poter “intersecare” i dati ottenuti dalle due tecnologie. Con una soluzione del genere, sarebbe possibile abbandonare quasi definitivamente i “foglietti” fisici, in quanto si lavorerebbe solo con documenti digitali: in questo modo l'input di riferimento sarebbe la foto dei fonogrammi, i quali non dovrebbero più essere trasportati fisicamente di sede in sede, da cui verrebbero estratti i dati da inserire vocalmente all'interno della piattaforma.

8.2 Best practices

In seguito ai vari di usabilità eseguiti per valutare il prototipo realizzato, sono emersi degli accorgimenti e best practices che potrebbero essere adottati per sfruttare al meglio l'applicativo. Tali accorgimenti verranno testati a loro volta, in modo tale da comunicarli eventualmente ai comuni italiani che potrebbero essere interessati alle funzionalità offerte.

I suggerimenti in questione sono:

- Si è notato che, in quanto ogni dispositivo ha un microfono diverso, alcuni sono più efficienti e maggiormente recettivi di altri. In particolare, si è osservato che i microfoni dei dispositivi mobili (quali smartphone e tablet) tendono ad essere migliori in termini di performance.
- L'uso delle cuffie o degli auricolari può migliorare significativamente il riconoscimento vocale per svariati motivi, tra cui la riduzione del rumore ambientale (cuffie o auricolari possono ridurre il rumore di fondo, consentendo al microfono di concentrarsi maggiormente sulla voce dell'utente), migliore qualità dell'audio (le cuffie di solito offrono una qualità audio superiore rispetto ai microfoni integrati nei dispositivi, il che può aiutare il sistema di riconoscimento vocale a distinguere meglio i suoni e a interpretare più accuratamente le parole pronunciate) e un migliore posizionamento del microfono (le cuffie spesso includono un microfono integrato che è posizionato più vicino alla

bocca dell'utente rispetto ai microfoni dei dispositivi. Questo posizionamento ottimizzato può catturare la voce con maggiore chiarezza e ridurre il rischio di rumore di fondo).

- Come anticipato nel corso della trattazione, si è notato come il modulo di riconoscimento faccia più fatica a riconoscere parole caratterizzate da poche lettere (“sì”, “no”, “uno”, “due”, ecc.). Mirando il problema ad un contesto numerico, potrebbe essere più efficiente ai fini del riconoscimento vocale, anteporre la parola “zero” prima dei numeri aventi poche lettere. Ad esempio, “zero zero”, “zero uno”, “zero due” (invece di semplicemente “zero”, “uno”, “due”) e via dicendo. Ciò non ha conseguenze sul numero riconosciuto (“zero uno” viene interpretato come “1”, ecc.), ma fornisce una chiarezza aggiuntiva e riduce il rischio di ambiguità nel processo di riconoscimento vocale.

Appendice A

Script per la valutazione del prototipo

A.1 Presentazione

Buongiorno! Innanzitutto ti ringrazio per aver accettato di partecipare a questo studio. Come prima cosa, permettimi di darti una breve infarinatura su quale sarà il contesto in cui andrà utilizzato il prototipo. Il progetto che ho sviluppato ha come obiettivo quello di proporre una soluzione innovativa e guidata al fine di favorire l'inserimento dei dati raccolti durante l'evento elettorale da parte delle amministrazioni comunali italiane. Per garantire ciò, sono stati pensati due moduli: un modulo di Speech Recognition, il cui compito è quello di coprire le funzionalità di riconoscimento vocale, riconoscere dunque quanto detto dall'utente e trascriverne i dati all'interno della piattaforma, e un modulo di Speech Synthesis, il cui compito è quello di coprire le funzionalità di Text to Speech, guidando l'utente nel corso della conversazione tramite delle istruzioni vocali, per favorire l'inserimento e la validazione dei dati elettorali.

Una volta identificate queste funzionalità, ho cercato di metterle insieme, in modo tale da riuscire a creare una conversazione fluida e intuitiva tra l'utente e il sistema, al fine di garantire una comunicazione efficace.

Durante il test, ti verranno fornite istruzioni sui compiti da svolgere, ma sei libero di interagire con l'applicazione a tuo piacimento, senza alcuna pressione dall'intervistatore. Inoltre, sarà disponibile un facilitatore per aiutarti nel caso tu incontri difficoltà durante l'esecuzione dei compiti assegnati, permettendoti di richiedere assistenza quando necessario.

A.2 Istruzioni

L'applicativo che ti andrò adesso a presentare è ancora un prototipo, quindi potresti notare delle imperfezioni, ma tutte le funzionalità sono state sviluppate e dovrebbero funzionare correttamente. Detto ciò, ti chiedo cortesemente di aprire il link che ti ho fornito in chat Zoom, per poi condividere il tuo schermo appena sei pronto, in modo tale da poterti seguire passo passo. Se possibile, ti invito a indossare delle cuffie (o degli auricolari) per migliorare la qualità dell'audio durante la nostra sessione.

Ti informo che la chiamata non verrà registrata, ma ti chiedo di parlare a voce alta in modo che io possa seguire i tuoi ragionamenti e opinioni.

Al termine della sessione, ti inviterò a compilare un questionario SUS, un questionario standard composto da dieci domande a risposta chiusa, per valutare l'usabilità del sistema. Ti prego di rispondere con la massima sincerità.

Siamo pronti per iniziare?

A.3 Attività

A.3.1 Attività 1: Elementi di interazione

Per questa prima attività, vorrei che tu dessi un'occhiata alle varie schermate presenti sulla piattaforma, andando a giocare un po' con la selezione dei sub-eventi elettorali, cercando di individuare gli elementi con cui potenzialmente potrai andare a interagire.

Al termine, passare alla prossima attività.

A.3.2 Attività 2: Set up conversazione

Come seconda attività, ti chiedo di interagire con gli elementi preliminari alla conversazione, mi riferisco agli slider con cui puoi impostare la velocità e il tono della conversazione, e con il menu a tendina tramite il quale puoi selezionare il tipo di voce da utilizzare.

Al termine, passare alla prossima attività.

A.3.3 Attività 3: Primo avvio conversazione

Ti chiedo adesso di andare nel sub-evento "Affluenze". Si tratta del sub-evento più semplice da gestire, per cui ha senso partire da questo. Fatto ciò, prova a dare inizio alla conversazione.

Vedere se il partecipante inizierà a inserire i dati dopo aver ricevuto l'istruzione da parte dell'applicativo o se lascerà scadere il timeout di riconoscimento.

Al termine, passare alla prossima attività.

A.3.4 Attività 4: Conversazione “Affluenze”

Adesso ti chiedo di provare a portare avanti una conversazione. Avrai notato che dopo aver cliccato “Avvia la conversazione”, si sentiranno delle istruzioni su che tipo di operazioni eseguire. Ti chiedo dunque di seguire queste istruzioni e provare a inserire i dati relativi ai votanti maschi, femmine e totali.

Vedere se il partecipante riuscirà a inserire correttamente tutti e tre i dati richiesti dalle “Affluenze”.

Al termine, passare alla prossima attività.

A.3.5 Attività 5: Conversazione “Liste”

Ci spostiamo ora su un altro sub-evento, quello delle “Liste”. Ho deciso di scegliere questo in quanto un compromesso di difficoltà tra quello visto precedentemente e uno più complicato come può essere quello dei “Raggruppamenti”. L’obiettivo da perseguire in questa attività è lo stesso visto prima, ti chiedo perciò di eseguire le stesse operazioni, ovviamente su un contesto leggermente variato.

Vedere se il partecipante riuscirà a portare avanti la conversazione trattandosi di parecchie righe da riempire. Stare attenti anche a vedere se noterà la variazione di colore delle righe mentre si procede con l’iterazione.

Al termine, passare alla prossima attività.

A.3.6 Attività 6: Interruzione conversazione

Adesso ti chiedo, così come ti sei ritrovato a termine dell’attività precedente, avvia la conversazione e interrompila quando meglio preferisci. Una volta fatto ciò, riparti dalle terza riga e riesegui la conversazione.

Vedere se il partecipante interromperà la conversazione tramite il bottone “Stop” o se lascerà scadere il timer. Inoltre notare la confidenza con cui cliccherà sulla riga richiesta.

Al termine, passare alla prossima attività.

A.3.7 Attività 7: Riepilogo dati

Come ultima attività, ti chiedo semplicemente di eseguire il riepilogo dei dati, cliccando sul bottone “Recap”.

Vedere se il partecipante andrà a modificare le impostazioni del riepilogo (velocità, tipo di voce) trattandosi di un’operazione potenzialmente lunga o se aspetterà la fine senza interagire.

A.4 Debriefing

Siamo giunti alla conclusione di questo test, grazie per aver eseguito le attività richieste. Come anticipato all'inizio, ti chiedo la cortesia di compilare un questionario (SUS) e rispondere a qualche domanda.

Condivisione e compilazione del questionario SUS, al termine del quale verranno, ove possibile, effettuate alcune domande.

1. Ti è piaciuto questo test?
2. Credi che dovrebbero essere apportate modifiche alle funzionalità attuali dell'applicativo appena mostrato?
3. Pensi che ci siano delle ulteriori funzionalità che dovrebbero essere integrate?
4. Trovi intuitiva la modalità con cui viene intrapresa la conversazione durante l'inserimento dei dati? Cambieresti qualcosa?
5. Hai bisogno di chiarimenti o hai delle domande riguardo al questionario?

Ti ringrazio molto per il tuo contributo. I tuoi commenti e il test che hai completato saranno estremamente utili per lo sviluppo del prodotto finale. Buona giornata!

Appendice B

Questionari SUS

All'interno di questa appendice sono stati riportati i questionari SUS che sono stati proposti e compilati dai cinque partecipanti selezionati per svolgere il test di usabilità relativo al prototipo realizzato. Per ognuno di essi è stato calcolato un punteggio finale, rappresentativo della valutazione complessiva dell'usabilità del prototipo.

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo applicativo frequentemente	4	3
2.	Ho trovato l'applicativo inutilmente complesso	1	4
3.	Ho trovato l'applicativo molto semplice da usare	4	3
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'applicativo	2	3
5.	Ho trovato le varie funzionalità dell'applicativo bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità dell'applicativo	2	3
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare l'applicativo facilmente	5	4
8.	Ho trovato l'applicativo molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare l'applicativo	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'applicativo	1	4
Valutazione totale			90

Tabella B.1: Valutazione SUS per partecipante P01

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo applicativo frequentemente	5	4
2.	Ho trovato l'applicativo inutilmente complesso	1	4
3.	Ho trovato l'applicativo molto semplice da usare	5	4
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'applicativo	1	4
5.	Ho trovato le varie funzionalità dell'applicativo bene integrate	4	3
6.	Ho trovato incoerenze tra le varie funzionalità dell'applicativo	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare l'applicativo facilmente	4	3
8.	Ho trovato l'applicativo molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare l'applicativo	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'applicativo	1	4
Valutazione totale			95

Tabella B.2: Valutazione SUS per partecipante P02

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo applicativo frequentemente	3	2
2.	Ho trovato l'applicativo inutilmente complesso	1	4
3.	Ho trovato l'applicativo molto semplice da usare	3	2
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'applicativo	2	3
5.	Ho trovato le varie funzionalità dell'applicativo bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità dell'applicativo	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare l'applicativo facilmente	3	2
8.	Ho trovato l'applicativo molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare l'applicativo	4	3
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'applicativo	1	4
Valutazione totale			80

Tabella B.3: Valutazione SUS per partecipante P03

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo applicativo frequentemente	4	3
2.	Ho trovato l'applicativo inutilmente complesso	1	4
3.	Ho trovato l'applicativo molto semplice da usare	5	4
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'applicativo	1	4
5.	Ho trovato le varie funzionalità dell'applicativo bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità dell'applicativo	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare l'applicativo facilmente	4	3
8.	Ho trovato l'applicativo molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare l'applicativo	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'applicativo	1	4
Valutazione totale			95

Tabella B.4: Valutazione SUS per partecipante P04

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo applicativo frequentemente	4	3
2.	Ho trovato l'applicativo inutilmente complesso	1	4
3.	Ho trovato l'applicativo molto semplice da usare	4	3
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare l'applicativo	2	3
5.	Ho trovato le varie funzionalità dell'applicativo bene integrate	4	3
6.	Ho trovato incoerenze tra le varie funzionalità dell'applicativo	2	3
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare l'applicativo facilmente	5	4
8.	Ho trovato l'applicativo molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare l'applicativo	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio l'applicativo	1	4
Valutazione totale			85

Tabella B.5: Valutazione SUS per partecipante P05

Bibliografia

- [1] *La Costituzione Italiana*. URL: <https://www.senato.it/istituzione/la-costituzione> (cit. a p. 1).
- [2] *Eleweb*. URL: <https://www.pro-logic.it/eleweb-elezioni/> (cit. a p. 1).
- [3] *Pro Logic Informatica*. URL: <https://www.pro-logic.it/> (cit. a p. 1).
- [4] *Elezioni europee*. URL: <https://elections.europa.eu/it/how-elections-work/> (cit. a p. 6).
- [5] *Il Parlamento Europeo*. URL: <https://www.affarieuropei.gov.it/it/istituzioni-europee/quadro-istituzionale/parlamento-europeo> (cit. a p. 6).
- [6] *Legge elettorale per l'elezione dei rappresentanti italiani al Parlamento europeo*. URL: <https://www.parlamento.it/parlam/leggi/790181.htm> (cit. a p. 7).
- [7] *Il Parlamento Italiano*. URL: <https://www.parlamento.it/> (cit. a p. 8).
- [8] *AIRE*. URL: https://www.esteri.it/it/servizi-consolari-e-visti/italiani-all-estero/aire_0/ (cit. a p. 8).
- [9] *Il Referendum*. URL: <https://www.treccani.it/enciclopedia/referendum/> (cit. a p. 10).
- [10] *What is Speech Recognition?* URL: <https://www.ibm.com/topics/speech-recognition> (cit. a p. 20).
- [11] *What is Speech Synthesis?* URL: <https://websitevoice.com/blog/what-is-speech-synthesis/> (cit. a p. 20).
- [12] *PHP*. URL: <https://www.php.net/> (cit. a p. 21).
- [13] *HTML*. URL: <https://www.html.it/guide/guida-html/> (cit. a p. 22).
- [14] *Javascript*. URL: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (cit. a p. 22).
- [15] *Web Speech API*. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API (cit. a p. 22).

- [16] *W3C*. URL: <https://www.w3.org/> (cit. a p. 22).
- [17] *Web Speech API: Speech Recognition*. URL: <https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition> (cit. a p. 23).
- [18] *Web Speech API: Speech Synthesis*. URL: <https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis> (cit. a p. 29).
- [19] *Why You Only Need to Test with 5 Users*. URL: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (cit. a p. 53).
- [20] *Questionario SUS*. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (cit. a p. 53).