

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



Sviluppo del frontend di un'applicazione web in React js

Tesi di laurea

Relatore

Prof. Paolo Baldan

Laureando

Mattia Episcopo

ANNO ACCADEMICO 2022-2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obbiettivi da raggiungere erano molteplici.

In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione ...

"I'm not crazy, my mother had me tested"

— Dott. Sheldon Lee Cooper

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Febbraio 2023

Mattia Episcopo

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'idea	1
1.3	Organizzazione del testo	2
2	Descrizione dello stage	3
2.1	Introduzione al progetto	3
2.2	Analisi preventiva dei rischi	3
2.3	Requisiti e obiettivi	4
2.4	Pianificazione	4
2.5	Strumenti	5
2.6	Prodotto ottenuto	7
3	Analisi dei requisiti	9
3.1	Casi d'uso	9
3.1.1	UC1 - Autenticazione	9
3.1.2	UC2 - Lettura dati da CD	11
3.1.3	UC3 - Visualizzazione dati ordinati per file	12
3.1.4	UC4 - Visualizzazione dati ordinati per procedimenti	13
3.1.5	UC5 - Upload procedimento	14
3.1.6	UC6 - Visualizzazione jobs	14
3.1.7	UC7 - Visualizzazione dei Marker relativi al file	15
3.1.8	UC8 - Visualizzazione degli Interventi relativi al file	17
3.2	Tracciamento dei requisiti	18
4	Progettazione	21
4.1	Tecnologie	21
4.1.1	Architettura dell'applicazione	22
4.1.2	Architettura frontend	23
4.2	Progettazione	23
4.3	Design Pattern utilizzati	23
5	Realizzazione e testing	25
5.1	Codifica	25
6	Conclusioni	27
6.1	Consuntivo finale	27
6.2	Raggiungimento degli obiettivi	28
6.3	Conoscenze acquisite	28

6.4 Valutazione finale	28
Bibliografia	31

Elenco delle figure

3.1	Descrizione grafica caso d'uso UC1	9
3.2	Descrizione grafica caso d'uso UC2	11
3.3	Descrizione grafica caso d'uso UC3	12
3.4	Descrizione grafica caso d'uso UC4	13
3.5	Descrizione grafica caso d'uso UC5	14
3.6	Descrizione grafica caso d'uso UC6	15
3.7	Descrizione grafica caso d'uso UC7	15
3.8	Descrizione grafica caso d'uso UC7	17

Elenco delle tabelle

Capitolo 1

Introduzione

Introduzione al contesto applicativo.

Esempio di utilizzo di un termine nel glossario
[Application Program Interface \(API\)](#).

Esempio di citazione in linea
site:agile-manifesto.

Esempio di citazione nel pie' di pagina
citazione¹

1.1 L'azienda

Synthema Artificial Intelligence (S.AI) è una start-up innovativa che si occupa di ricerca, progettazione, sviluppo, commercializzazione e manutenzione di prodotti e servizi innovativi ad alto valore tecnologico, basati sull'Internet of Things, la blockchain e su tecniche di Intelligenza Artificiale (in particolare reti neurali profonde), per l'analisi integrata e la comprensione di dati multimodali da fonti eterogenee (linguaggio naturale sia scritto che parlato, audio, immagini, video, dati generati da sensori) e per la gestione dei workflow, sia nel settore pubblico che privato.

1.2 L'idea

Si è resa necessaria per l'azienda, la costruzione di un'applicazione per il caricamento delle registrazioni audio, che vengono effettuate nelle aule di tribunale, verso un portale già esistente. Le registrazioni che vengono caricate, dispongono di vari metadati che possono essere usati per rendere il caricamento delle registrazioni stesse più veloce e semplice possibile.

¹womak:lean-thinking.

1.3 Organizzazione del testo

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Descrizione dello stage

Breve introduzione al capitolo

In questo capitolo viene descritto come si è pianificato e svolto lo stage presso SAI, introducendo il progetto, considerando gli obiettivi e i possibili rischi.

2.1 Introduzione al progetto

Lo stage svolto presso SAI aveva come scopo la creazione di un prototipo di piattaforma web/desktop per il caricamento delle registrazioni audio effettuate nelle aule di tribunale, con l'aiuto di alcuni metadata prodotti durante le medesime registrazioni. In particolare l'applicazione da sviluppare è divisa in due parti backend e frontend (spiegare meglio back e front) che comunicano tra loro mediante API rest (spiegare API). Lo stage si è focalizzato sullo sviluppo della parte frontend, avanzando di pari passo con lo sviluppo del backend realizzata da altri membri del team di sviluppo. Per la realizzazione del frontend si è scelto di utilizzare Javascript con i framework React e Redux.

2.2 Analisi preventiva dei rischi

Nella fase di analisi iniziali si sono individuati i principali rischi a cui si poteva andare incontro e si è proceduto a definire le possibili soluzioni per farne fronte.

1. Uso di nuove tecnologie

Descrizione: le tecnologie proposte per la gestione e lo sviluppo del progetto erano per lo più nuove o scarsamente conosciute..

Soluzione: è stato previsto un periodo iniziale di studio e formazione su queste tecnologie..

2. Modalità di lavoro smart working

Descrizione: lo stage è stato fatto completamente da remoto, e poteva portare ad una possibile mancanza di comunicazione e ad un'incertezza nelle attività da svolgere..

Soluzione: il tutor aziendale si è reso disponibile a vari meeting nelle prime fasi del progetto e nella formazione iniziale, e rimanendo a disposizione per altri meeting in caso di dubbi sulle attività da svolgere..

2.3 Requisiti e obiettivi

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- * O per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- * D per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- * F per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo. Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito.

Codice	Descrizione
O01	autenticazione mediante server remoto
O02	lettura dati da CD
O03	precompilazione di form con i dati caricati da CD
O04	editing dei dati del form
D01	upload dei dati verso i sistemi esterni
D02	test di unità esaustivi
F01	possibilità di ascoltare le registrazioni
F02	possibilità di modificare i dati mediante interazioni evolute (per es. drag-n-drop)
F03	realizzazione di un'applicazione desktop con Electron
F04	compilazione multiplatforma dell'applicazione desktop

2.4 Pianificazione

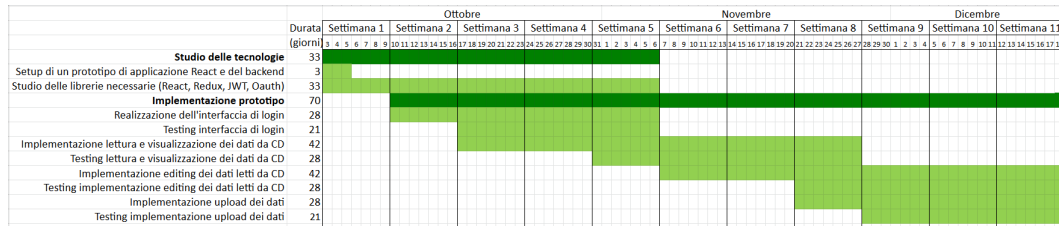
La pianificazione, in termini di quantità di ore, sarà distribuita in attività di studio e attività implementative.

Lo stage è stato strutturato in due fasi principali, la prima dedicata alle attività di studio e la seconda alle attività implementative. Di seguito vengono elencate le attività pianificate con la relativa quantità di ore stimata per ogni attività.

- * Studio delle tecnologie (80 ore)
 - Setup di un prototipo di applicazione React e del backend (16 ore);
 - Studio delle librerie necessarie (React, Redux, JWT, OAuth) (64 ore);
- * Implementazione prototipo (240 ore)
 - Realizzazione dell'interfaccia di login (30 ore);

- Implementazione lettura e visualizzazione dei dati da CD (70 ore);
- Implementazione editing dei dati letti da CD (70 ore);
- Implementazione upload dei dati (30 ore);
- Testing dei prodotti realizzati (40ore);

Per ogni attività riguardante la fase di implementazione del prototipo è stata prevista anche la relativa attività di testing. Il tutto viene rappresentato dal seguente diagramma di Gantt.



2.5 Strumenti

L'azienda usa un'infrastruttura già collaudata della quale fanno parte sistemi di versionamento del codice, strumenti per l'organizzazione del lavoro e per la comunicazione all'interno del gruppo di lavoro. Di seguito una panoramica dettagliata degli strumenti utilizzati per lavorare nel team di sviluppatori dell'azienda.

slack

È lo strumento che viene usato per la collaborazione aziendale, permette di vedere lo stato degli utenti (sviluppatori dell'azienda nel nostro caso) se disponibili o assenti al momento. Questa applicazione dà la possibilità di comunicare singolarmente con gli altri partecipanti oppure in piccoli gruppi divisi per progetto. Permette in oltre la condivisione dei file, molto utile nelle comunicazioni veloci.

Jira

Jira è lo strumento che l'azienda usa per l'organizzazione del lavoro. Permette di creare vari progetti, e per ogni progetto consente la creazione dei relativi ticket. In base alla metodologia di lavoro che si sceglie di utilizzare lo strumento permette di adottarla in tutto. Nel nostro caso la scelta aziendale è il metodo SCRUM e questo strumento consente la creazione degli sprint, è fornito di un backlog, ha una bacheca personalizzata per ogni utente che consente di vedere lo stato di avanzamento dei ticket di interesse. Inoltre consente di creare report sull'andamento degli sprint oppure su alcuni periodi, cosa molto utile per tracciare la guida nel miglioramento aziendale.

Gitlab

Sistema di versionamento dei file di codice usato dall'azienda. Questo strumento si basa su git e consente tutte le operazioni di un repository git. I file vengono condivisi tra tutti gli sviluppatori grazie a questo strumento, divisi in repository, uno per ogni

progetto sul quale l'azienda lavora. La linea guida aziendale per l'utilizzo del sistema GitLab si basa sul workflow denominato feature branching, questo sistema di lavoro prevede di lasciare il ramo principale (solitamente chiamato main) sempre pulito e con una versione funzionante e testata del prodotto, mentre invece quanto si vuole lavorare su una nuova feature si crea un nuovo ramo partendo da quello principale e una volta che questa feature sarà pronta per essere revisionata e integrata con quella principale si chiederà una merge request di questo feature branch sul branch principale. Il mantenimento dei file di codice sul repository si basa sulla regola aziendale 1 modifica - 1 commit - 1 file, ovvero si predilige che ogni branch che lo sviluppatore crea per sviluppare la relativa feature sia composto da commit che riguardano soltanto un file con la relativa modifica.

VS Code

È l'IDE () di lavoro utilizzato per sviluppare il locale. Questo strumento è completo di tutto quello che serve per lo sviluppo del prodotto. Esso infatti consente di sviluppare nei linguaggi che interessano il progetto, aiutando con vari plugin per il riconoscimento del codice, aiutando così lo sviluppatore a lavorare più velocemente e in modo più intuitivo. Inoltre è competamente integrato con i sistemi di versionamento, in particolare nel nostro caso con GitLab. Oltre a queste funzionalità consente di utilizzare plugin per la pulizia del codice, che settati in maniera corretta permettono di rimanere sempre fedeli alle regole aziendali automaticamente ad ogni salvataggio.

Jitsi

È lo strumento che l'azienda utilizza per le riunioni in videocall settimanali ma anche per quelle individuali che dovessero essere necessarie in qualsiasi momento.

SCRUM

SCRUM è il framework che l'azienda segue per il lavoro in gruppo sui vari progetti. Questo framework molto diffuso per lo sviluppo software in team si basa su sprint di durata breve che mirano al completamento dei task assegnati, il ripetersi temporale di questi sprint porta l'avanzamento del prodotto. Nel nostro caso particolare quella che segue l'azienda è una versione personalizzata di questo framework. Il funzionamento è spiegato di seguito.

- * Durata degli sprint: normalmente una settimana;
- * Meeting del lunedì:
 - ognuno da una rapida panoramica per tenere tutti aggiornati su quello che ha svolto nella settimana precedente;
 - si discutono e assegnano i task che si trovavano nel backlog per lo sprint seguente;
 - rapida retrospettiva per valutazioni critiche dello sprint appena terminato;
- * Stati dei task da svolgere:
 - da completare: nel corrente sprint ma nessuno ci sta ancora lavorando;

- in corso: qualcuno sta lavorando a questo task;
- pronto per la revisione: il task è stato completato, si attende che l'incaricato revisioni il codice e faccia il merge nel ramo principale;
- completato: il task è stato revisionato e integrato nel ramo principale;
- bloccato: per qualche ragione il task è bloccato, si scrive il motivo per il quale non può essere proseguito e se è necessario parlarne con qualcuno;
- da posticipare: il task deve essere riposto nel backlog e ripianificato per un altro sprint;

2.6 Prodotto ottenuto

Capitolo 3

Analisi dei requisiti

Breve introduzione al capitolo

3.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

3.1.1 UC1 - Autenticazione

- * **Identificativo:** UC1
- * **Nome:** autenticazione
- * **Descrizione grafica:**

Figura 3.1: Descrizione grafica caso d'uso UC1

- * **Attori**
 - *Primari:* utente non autorizzato
 - *Secondari:* Google o Faceebook (???)
- * **Precondizione:** l'utente non autenticato si trova sulla pagina di autenticazione.
- * **Postcondizione:** l'utente è autenticato.
- * **Scenario principale:** l'utente vuole effettuare il login all'applicazione.
- * **Scenario secondario:** l'utente non riesce ad autenticarsi a causa di un errore nella procedura. (**UC1.3**)

UC1.1 - Inserimento username

- * **Identificativo:** UC1.1
- * **Nome:** inserimento username
- * **Descrizione grafica:** (approfondita in UC1)
- * **Attori**
 - *Primari:* utente non autorizzato
- * **Precondizione:** l'utente ha a disposizione una username
- * **Postcondizione:** l'utente ha inserito la username.
- * **Scenario principale:** l'utente inserisce la username nell'apposito campo di input.
- * **Scenario secondario:** l'utente ha inserito una username non corretta che causa un errore. (UC1.3)

UC1.1.1 - Errore inserimento username

- * **Identificativo:** UC1.1.1
- * **Nome:** errore inserimento username
- * **Descrizione grafica:** (approfondita in UC1)
- * **Attori**
 - *Primari:* utente non autorizzato
- * **Precondizione:** la username inserita dall'utente non è corretta.
- * **Postcondizione:** l'errore viene mostrato all'utente.
- * **Scenario principale:** l'utente inserisce una username non corretta, il sistema segnala l'errore all'utente e mostra nuovamente la maschera di login.

UC1.2 - Inserimento password

- * **Identificativo:** UC1.1
- * **Nome:** inserimento password
- * **Descrizione grafica:** (approfondita in UC1)
- * **Attori**
 - *Primari:* utente non autorizzato
- * **Precondizione:** l'utente ha a disposizione una password
- * **Postcondizione:** l'utente ha inserito la password.
- * **Scenario principale:** l'utente inserisce la password nell'apposito campo di input.
- * **Scenario secondario:** l'utente ha inserito una password non corretta che causa un errore. (UC1.3)

UC1.2.1 - Errore inserimento password

- * **Identificativo:** UC1.2.1
- * **Nome:** errore inserimento password
- * **Descrizione grafica:** (approfondita in UC1)
- * **Attori**
 - *Primari:* utente non autorizzato
- * **Precondizione:** la password inserita dall'utente non è corretta.
- * **Postcondizione:** l'errore viene mostrato all'utente.
- * **Scenario principale:** l'utente inserisce una password non corretta, il sistema segnala l'errore all'utente e mostra nuovamente la maschera di login.

UC1.3 - Errore autenticazione

- * **Identificativo:** UC1.3
- * **Nome:** errore autenticazione
- * **Descrizione grafica:** (approfondita in UC1)
- * **Attori**
 - *Primari:* utente non autorizzato
- * **Precondizione:** il sistema di autenticazione riceve la richiesta da parte dell'utente.
- * **Postcondizione:** il sistema comunica all'utente l'errore avvenuto, viene riproposta la maschera di login.
- * **Scenario principale:** la richiesta di autenticazione non viene gestita dal sistema che comunica l'errore avvenuto all'utente e mostra nuovamente la maschera di login per un nuovo tentativo.

3.1.2 UC2 - Lettura dati da CD

- * **Identificativo:** UC2
- * **Nome:** lettura dati da CD
- * **Descrizione grafica:**

Figura 3.2: Descrizione grafica caso d'uso UC2

- * **Attori**
 - *Primari:* utente autorizzato

- * **Precondizione:** l'utente autorizzato si trova nella pagina per il caricamento dei dati.
- * **Postcondizione:** l'utente visualizza i dati che ha caricato.
- * **Scenario principale:** l'utente premendo sull'apposito bottone può caricare i file contenuti nel CD di interesse.
- * **Scenario secondario:** il sistema riscontra un errore nella procedura di caricamento dei dati. (**UC2.1**)

UC2.1 - Errore caricamento dati da CD

- * **Identificativo:** UC2.1
- * **Nome:** errore caricamento dati da CD
- * **Descrizione grafica:** (approfondita in UC2)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente ha tentato di caricare i dati.
- * **Postcondizione:** l'errore viene mostrato all'utente.
- * **Scenario principale:** il sistema non è riuscito a gestire la richiesta di caricamento dei dati da parte dell'utente.

3.1.3 UC3 - Visualizzazione dati ordinati per file

- * **Identificativo:** UC3
- * **Nome:** visualizzazione dati ordinati per file
- * **Descrizione grafica:**

Figura 3.3: Descrizione grafica caso d'uso UC3

- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente si trova sulla pagina per il caricamento dati, che ha già effettuato correttamente.
- * **Postcondizione:** l'utente visualizza i dati caricati sull'applicazione, ordinati per nome dei file.
- * **Scenario principale:** l'utente ha caricato correttamente i dati e questi vengono visualizzati ordinati per nome del file.
- * **Scenario secondario:** l'utente può riprodurre i file audio.

UC3.1 - Riproduzione audio file

- * **Identificativo:** UC3.1
- * **Nome:** riproduzione audio file
- * **Descrizione grafica:** (approfondita in UC3)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente si trova sulla pagina per il caricamento dati, che ha già effettuato correttamente.
- * **Postcondizione:** il file audio viene riprodotto.
- * **Scenario principale:** l'utente comanda il riproduttore audio con gli appositi comandi play/pause.

[Manca visualizzazione/modifica codici procedimenti da capire come gestire questo caso d'uso]

3.1.4 UC4 - Visualizzazione dati ordinati per procedimenti

- * **Identificativo:** UC4
- * **Nome:** visualizzazione dati ordinati per procedimenti
- * **Descrizione grafica:**

Figura 3.4: Descrizione grafica caso d'uso UC4

- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente si trova sulla pagina per il caricamento dati, che ha già effettuato correttamente.
- * **Postcondizione:** l'utente visualizza i dati caricati sull'applicazione, ordinati per procedimenti.
- * **Scenario principale:** l'utente ha caricato correttamente i dati e questi vengono visualizzati ordinati per procedimenti.

Figura 3.5: Descrizione grafica caso d'uso UC5

3.1.5 UC5 - Upload procedimento

- * **Identificativo:** UC5
- * **Nome:** upload procedimento
- * **Descrizione grafica:**
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente si trova sulla pagina di visualizzazione dei dati.
- * **Postcondizione:** l'utente ha caricato il procedimento desiderato e i relativi file.
- * **Scenario principale:** l'utente tramite l'apposito bottone per l'upload, carica il procedimento desiderato.
- * **Scenario secondario:** la procedura di upload procedimento non va a buon fine e restituisce un errore. (**UC5.1**)

UC5.1 - Errore upload procedimento

- * **Identificativo:** UC5.1
- * **Nome:** errore upload procedimento
- * **Descrizione grafica:** (approfondita in UC5)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** il sistema non ha gestito correttamente la richiesta di upload procedimento.
- * **Postcondizione:** l'errore viene visualizzato sull'applicazione.
- * **Scenario principale:** la richiesta di upload procedimento non va a buon fine e l'errore viene mostrato all'utente.

3.1.6 UC6 - Visualizzazione jobs

- * **Identificativo:** UC6
- * **Nome:** visualizzazione jobs
- * **Descrizione grafica:**
- * **Attori**

Figura 3.6: Descrizione grafica caso d'uso UC6

- *Primari*: utente autorizzato
- * **Precondizione**: l'utente si trova all'interno dell'applicazione e vuole visualizzare la lista dei jobs.
- * **Postcondizione**: l'utente visualizza la lista dei jobs.
- * **Scenario principale**: l'utente visualizza la lista dei jobs.
- * **Scenario secondario**: l'utente può visualizzare il dettaglio di un singolo job premendo sull'apposito link. (**UC6.1**)

UC6.1 - Visualizzazione dettaglio job

- * **Identificativo**: UC6.1
- * **Nome**: visualizzazione dettaglio job
- * **Descrizione grafica**: (approfondita in UC6)
- * **Attori**
 - *Primari*: utente autorizzato
- * **Precondizione**: l'utente ha premuto sull'apposito link.
- * **Postcondizione**: il dettaglio del job viene visualizzato.
- * **Scenario principale**: l'utente visualizza tutti i dettagli del singolo job.

3.1.7 UC7 - Visualizzazione dei Marker relativi al file

- * **Identificativo**: UC7
- * **Nome**: visualizzazione dei Marker relativi al file
- * **Descrizione grafica**:

Figura 3.7: Descrizione grafica caso d'uso UC7

- * **Attori**
 - *Primari*: utente autorizzato
- * **Precondizione**: l'utente si trova sulla pagina per il caricamento dati, che ha già effettuato correttamente.
- * **Postcondizione**: l'utente ha visualizzato i Marker relativi al file desiderato.
- * **Scenario principale**: l'utente premendo sull'apposito bottone visualizza i Marker relativi al file.
- * **Scenario secondario**: l'utente gestisce i Marker potendo modificare o rimuovere quelli esistenti oppure aggiungerne di nuovi.

UC7.1 - Aggiunta Marker

- * **Identificativo:** UC7.1
- * **Nome:** aggiunta Marker
- * **Descrizione grafica:** (approfondita in UC7)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente ha premuto il bottone per l'aggiunta di un Marker.
- * **Postcondizione:** l'utente visualizza il nuovo marker nell'elenco dei marker relativi al file.
- * **Scenario principale:** l'utente compila il form per l'aggiunta del nuovo marker e lo salva.

UC7.2 - Modifica Marker

- * **Identificativo:** UC7.2
- * **Nome:** modifica Marker
- * **Descrizione grafica:** (approfondita in UC7)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente ha premuto il bottone per la modifica di un Marker dall'elenco degli stessi.
- * **Postcondizione:** l'utente visualizza il marker modificato nell'elenco dei marker relativi al file.
- * **Scenario principale:** l'utente modifica i valori desiderati tramite il form per la modifica del marker e lo salva.

UC7.3 - Eliminazione Marker

- * **Identificativo:** UC7.3
- * **Nome:** eliminazione Marker
- * **Descrizione grafica:** (approfondita in UC7)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente ha premuto il bottone per l'eliminazione di un Marker dall'elenco degli stessi.
- * **Postcondizione:** l'utente visualizza l'elenco dei marker relativi al file senza il marker eliminato.
- * **Scenario principale:** l'utente elimina il marker scelto premendo il bottone per l'eliminazione.

3.1.8 UC8 - Visualizzazione degli Interventi relativi al file

- * **Identificativo:** UC8
- * **Nome:** visualizzazione degli Interventi relativi al file
- * **Descrizione grafica:**

Figura 3.8: Descrizione grafica caso d'uso UC7

- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente si trova sulla pagina per il caricamento dati, che ha già effettuato correttamente.
- * **Postcondizione:** l'utente ha visualizzato gli Interventi relativi al file desiderato.
- * **Scenario principale:** l'utente premendo sull'apposito bottone visualizza gli Interventi relativi al file.
- * **Scenario secondario:** l'utente gestisce gli Interventi potendo modificare o rimuovere quelli esistenti oppure aggiungerne di nuovi.

UC8.1 - Aggiunta Intervento

- * **Identificativo:** UC8.1
- * **Nome:** aggiunta Intervento
- * **Descrizione grafica:** (approfondita in UC8)
- * **Attori**
 - *Primari:* utente autorizzato
- * **Precondizione:** l'utente ha premuto il bottone per l'aggiunta di un Intervento.
- * **Postcondizione:** l'utente visualizza il nuovo Intervento nell'elenco degli Interventi relativi al file.
- * **Scenario principale:** l'utente compila il form per l'aggiunta del nuovo Intervento e lo salva.

UC8.2 - Modifica Intervento

- * **Identificativo:** UC7.2
- * **Nome:** modifica Intervento
- * **Descrizione grafica:** (approfondita in UC8)
- * **Attori**

– *Primari*: utente autorizzato

- * **Precondizione**: l'utente ha premuto il bottone per la modifica di un Intervento dall'elenco degli stessi.
- * **Postcondizione**: l'utente visualizza il Intervento modificato nell'elenco degli Interventi relativi al file.
- * **Scenario principale**: l'utente modifica i valori desiderati tramite il form per la modifica del Intervento e lo salva.

UC8.3 - Eliminazione Intervento

* **Identificativo**: UC8.3

* **Nome**: eliminazione Intervento

* **Descrizione grafica**: (approfondita in UC8)

* **Attori**

– *Primari*: utente autorizzato

- * **Precondizione**: l'utente ha premuto il bottone per l'eliminazione di un Intervento dall'elenco degli stessi.
- * **Postcondizione**: l'utente visualizza l'elenco degli Interventi relativi al file senza l'Intervento eliminato.
- * **Scenario principale**: l'utente elimina il Intervento scelto premendo il bottone per l'eliminazione.

3.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

F = facoltativo

Nelle tabelle ??, ?? e ?? sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Requisito	Descrizione	Use Case
RFO-1	autenticazione	UC1
RFO-2	lettura dati da CD	UC2
RFO-3	visualizzazione dati ordinati per file	UC3
RFO-4	visualizzazione dati ordinati per procedimenti	UC4
RFO-5	gestione dei Marker relativi al file	UC3
RFO-6	gestione degli Interventi relativi al file	UC3
RFD-1	upload procedimento	UC5
RFF-2	visualizzazione jobs	UC6
RQD-1	test di unità esaustivi	-
RFF-1	riproduzione audio file	UC3.1

Capitolo 4

Progettazione

Breve introduzione al capitolo

4.1 Tecnologie

Di seguito viene data una panoramica delle tecnologie utilizzate.

Javascript

JavaScript è un linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione applicazioni web. L'intera applicazione è stata scritta con questo linguaggio.

React

React è una libreria Javascript utilizzata per implementare interfacce utente (UI) lato frontend. React si basa sul concetto di component, idealmente è una libreria che permette di costruire i propri component come fossero degli elementi HTML del DOM per poi poterli riutilizzare nell'intera applicazione.

Redux & Redux RTK

Redux è un contenitore dello stato per le applicazioni Javascript. Viene usato per la gestione centralizzata dello stato delle applicazioni sviluppate in React Javascript. In particolare con la sua libreria Redux-Toolkit, permette una gestione dello stato semplice ed efficiente.

API rest

Descrizione Tecnologia 2

JSON

Markup usato per la pubblicazione dei dati dell'API.

MUI

Material UI è una libreria React open-source che permette di implementare i Google's Material Design. Essa comprende una collezione di componenti React precostruiti che possono essere facilmente adattati e messi in uso nella UI dell'applicazione.

React Router

React Router è la libreria standard per il routing in React. Questa libreria permette la navigazione tra le varie viste dell'applicazione, permette di gestire le URL, e mantenere la sincronizzazione tra URL e viste.

Docker

Docker è un sistema che permette di facilitare il deployment delle applicazioni.

airbnb js guidelines

(???) Descrizione Tecnologia 2

eslint

(???)

4.1.1 Architettura dell'applicazione

L'architettura dell'applicazione era già stata progettata prima dello stage e in piccola parte già esistente. L'intero sistema si basa sull'interazione tra backend e frontend mediante API rest, con l'aiuto del gestore di file minIO utilizzato per la memorizzazione dei file che vengono caricati dell'utente. Il sistema presenta quindi la formazione client-server dove nel server risiede la parte backend dell'applicativo mentre invece i vari client rappresentano il frontend dell'applicativo. [disegno generale di front end back end api minio ecc ecc]

Backend

Il backend è un'applicazione a se stante scritta interamente in python, con architettura MVC (Model-View-Controller). Questa applicazione espone delle URL come endpoint per le chiamate REST del client frontend.

Frontend

Il frontend dell'applicazione è la parte che è stata interamente implementata in oggetto allo stage, è l'interfaccia grafica presentata all'utente.

APIrest

Le API rest sono il tramite tra le due applicazioni, vengono gestite nella parte di frontend con redux-tolkit, e interrogano il backend sugli appositi endpoint, per poi lasciare al frontend l'interpretazione del risultato. Sono state utilizzate chiamate di tipo GET e POST. Inoltre una particolare chiamata di tipo PUT viene effettuata dal frontend per l'upload dei file su minIO.

MinIO**4.1.2 Architettura frontend**

disegno del frontend redux react come lavorano Il frontend dell'applicazione è stato progettato come una single page application, che mantiene separazione tra lo stato e la user interface.

React**Redux****React router****Mui****4.2 Progettazione**

viste implementate (jobs, proceeding view, file view), caricamento dati, upload dei dati, un po di come sono state pensate

In sede di progettazione del frontend si è deciso di realizzarlo in javascript, e in particolare usando la libreria react, sfruttando a pieno il riutilizzo del codice che questa libreria permette agevolmente costruendo componenti. Lo stato del frontend è stato gestito in modo indipendente utilizzando redux, e redux toolkit per sfruttare le API quando è necessario comunicare con il backend. Per il routing è stata utilizzata la libreria apposita react-router che permette con semplicità di gestire le url e i link all'interno dell'applicazione senza dover usare strumenti esterni al linguaggio. Graficamente invece, si è scelto di usare MUI per mantenere un interfaccia molto intuitiva, semplice da usare, responsive e con tanti elementi già pronti per essere integrati con React.

Redux permette di creare uno store, dove persistono tutti i dati che vogliamo gestire lato frontend, questo permette di non dover salvare i dati sul server e fare inutili e pesanti chiamate ogni volta che si necessita di rappresentare questi dati. Lo store nel nostro caso particolare è stato usato per memorizzare lato frontend i "dati caricati dall'utente nella fase di caricamento dati da CD".

1) store 2) componenti 3) integrazione componenti 5) routing 4) viste principali

4.3 Design Pattern utilizzati

(?????) tipo observer template se react lo usa... c'è una forma di MVVM in react+redux... posso citare MVC del back end o non frega niente a nessuno?

Capitolo 5

Realizzazione e testing

Breve introduzione al capitolo

5.1 Codifica

Come ho agito in fase di codifica, cioè creazione dello store redux, creazione dell'api rtk, creazione dei vari componenti in react, integrazione dei componenti react, interazione tra componenti e stato/store test??? eslint-airbnb style guide utilizzate come standard aziendale versionamento in gitlab regola aziendale 1 modifica 1 file 1 commit (nella situazione ideale), new file and delete file tutti sullo stesso commit purchè tutti dello stesso tipo big commit iniziali o di modifiche più sostanziose 1 file 1 commit

Generalmente il flusso della codifica viene gestito in base allo sprint scrum, quindi lunedì retrospettiva su attività fatte, selezione delle attività da fare per lo script settimanale e si parte. il ticket passa dallo stato "da completare" allo stato "in corso" parlando di react che induce all'uso di componenti riutilizzabili, si è cercato di sfruttare il più possibile questo principio. creazione di un nuovo componente integrazione del componente nella vista / nelle viste creazione dello store redux creazione delle api redux-toolkit

flusso di lavoro in gitlab, con branch flow.. merge requeste per quando si deve approvare qualche modifica, ticket che passa in stato "da verificare" se necessita di altre modifiche torna "in corso" altrimenti va in "completato"

parallelismo tra flow di git usato e ticketing jira con metodo scrum documentazione nei ticket e nelle merge request di tutto quello che è stato fatto

Capitolo 6

Conclusioni

6.1 Consuntivo finale

tabella cronologica dello stage e valutazione rispetto al gantt iniziale esempio subsection

Settimana		Attività
1	dal 17/10/2022 al 25/10/2022	autenticazione mediante server remoto
2	dal 17/10/2022 al 25/10/2022	lettura dati da CD
3	dal 17/10/2022 al 25/10/2022	precompilazione di form con i dati caricati da CD
4	dal 17/10/2022 al 25/10/2022	editing dei dati del form
5	dal 17/10/2022 al 25/10/2022	upload dei dati verso i sistemi esterni
6	dal 17/10/2022 al 25/10/2022	test di unità esaustivi
7	dal 17/10/2022 al 25/10/2022	possibilità di ascoltare le registrazioni
8	dal 17/10/2022 al 25/10/2022	possibilità di modificare i dati mediante interazioni evolute (per es. drag-n-drop)
9	dal 17/10/2022 al 25/10/2022	realizzazione di un'applicazione desktop con Electron
10	dal 17/10/2022 al 25/10/2022	compilazione multiplatforma dell'applicazione desktop

Orario

Sviluppo

6.2 Raggiungimento degli obiettivi

tabella requisiti stage superati e non

Codice	Descrizione	Stato
O01	autenticazione mediante server remoto	soddisfatto
O02	lettura dati da CD	soddisfatto
O03	precompilazione di form con i dati caricati da CD	soddisfatto
O04	editing dei dati del form	soddisfatto
D01	upload dei dati verso i sistemi esterni	soddisfatto
D02	test di unità esaustivi	soddisfatto
F01	possibilità di ascoltare le registrazioni	soddisfatto
F02	possibilità di modificare i dati mediante interazioni evolute (per es. drag-n-drop)	non soddisfatto
F03	realizzazione di un'applicazione desktop con Electron	non soddisfatto
F04	compilazione multiplatforma dell'applicazione desktop	non soddisfatto

6.3 Conoscenze acquisite

dove ho acquisito conoscenze s git, programmazione js, vari react ecc ecc, metodologie di lavoro

6.4 Valutazione finale

cazzate riguardo quanto è utile lo stage, come mi sono trovato con azienda tutor ecc ecc, valutazione critica delle conoscenze acquisite delle tempistiche

Bibliografia