

UNIVERSITÀ DEGLI STUDI DI TORINO



CORSO DI LAUREA TRIENNALE IN INFORMATICA

DIPARTIMENTO DI INFORMATICA

RELAZIONE DI STAGE

## Sviluppo web app per monitoraggio stato referti

**Relatore:**  
Prof.ssa  
Susanna Donatelli

**Candidato:**  
Alberto Campini  
M:885128

Anno Accademico 2020/2021

16 luglio 2021

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Contesto Aziendale . . . . .	5
1.2	Proposta di progetto . . . . .	7
<b>2</b>	<b>Analisi preliminare</b>	<b>9</b>
2.1	Situazione iniziale . . . . .	9
2.2	Necessità riscontrate . . . . .	11
2.3	Fase di ideazione . . . . .	12
2.3.1	Richieste Principali . . . . .	13
2.4	Casi d'uso . . . . .	14
2.4.1	Attori . . . . .	14
2.4.2	Visualizzare le statistiche . . . . .	14
2.4.3	Consultare la tabella referti . . . . .	16
<b>3</b>	<b>Sviluppo</b>	<b>18</b>
3.1	Approccio e metodologie utilizzate . . . . .	19
3.2	Iterazioni . . . . .	19
3.3	Esperienza utente . . . . .	27
3.4	Problemi affrontati . . . . .	28
<b>4</b>	<b>Tenologie utilizzate</b>	<b>30</b>
4.1	ASP.Net . . . . .	30
4.1.1	DevExpress . . . . .	31
4.2	Database relazionale Oracle . . . . .	31
4.2.1	Schema relazionale . . . . .	33
4.3	HTML5 . . . . .	34
4.4	CSS . . . . .	35
4.4.1	Bootstrap . . . . .	36
4.5	JavaScript . . . . .	36
4.5.1	JQuery . . . . .	37
4.6	LADP . . . . .	37
4.7	Servizi Windows . . . . .	38
4.8	Caratteristiche e punti di forza . . . . .	39
<b>5</b>	<b>Proposte Alternative</b>	<b>40</b>
5.1	Pattern architetturale MVC . . . . .	40
5.2	Punti di forza . . . . .	43
5.3	confronto con webForm . . . . .	44

<b>6 Conclusioni</b>	<b>45</b>
6.1 Obiettivi raggiunti . . . . .	45
6.2 Possibili ampliamenti . . . . .	47
<b>7 Sitografia</b>	<b>48</b>

## **Abstract**

L'informatizzazione oggi giorno è ovunque, anche nel contesto sanitario pubblico: questo rende più semplice e immediato il trattamento dei dati, la loro gestione e relativa manipolazione, ma al contempo comporta una attenzione aggiuntiva alla loro gestione.

Il focus del mio stage è stato proprio quello di creare una piattaforma web che permette il monitoraggio e la gestione dello stato referti relativi all'ASL di Asti. Tramite questa web app gli operatori possono visualizzare statistiche, effettuare estrazioni e ricerche di documentazioni sanitarie e manipolare servizi lato server.

Attraverso la ricerca su singoli referti o gruppi, è possibile individuare problematiche precise o tramite i dati forniti, malfunzionamenti interni al sistema che possono comportare un rallentamento nell'invio dei relativi referti. In questa relazione affronterò lo studio preliminare che ho effettuato prima e durante lo sviluppo, le problematiche affrontate e relative soluzioni, le tecnologie utilizzate e i possibili ampliamenti futuri per la piattaforma.

# 1 Introduzione

In questa relazione di stage andrò ad analizzare e approfondire la mia esperienza riguardante lo stage curricolare svolto durante il terzo anno universitario presso l'università degli studi di Torino, nel dipartimento di Informatica.

Il tirocinio è stato svolto presso l'azienda astigiana CO.D.E.Solution SRL durante un lockdown nazionale dovuto alla pandemia da Coronavirus SARS COV-2 che ha comportato lo svolgimento in modalità smartworking dell'intera esperienza.

Nella trattazione di questa relazione, cercherò di focalizzarmi principalmente sulla narrazione delle scelte che ho effettuato e conseguenti ragionamenti che mi hanno portato a sviluppare il sistema richiesto dal cliente all'azienda. Analizzerò sia la fase iniziale di ideazione e disciplina dei requisiti con annessi casi d'uso, sia il successivo sviluppo della piattaforma vera e propria, strutturato in iterazioni.

## 1.1 Contesto Aziendale

L'azienda in cui ho svolto il tirocinio curricolare della durata complessiva di 225 ore, di nome CO.d.e Solution S.R.L., si pone nel panorama lavorativo come azienda di sviluppo software specializzata in web application. I clienti principali sono le Agenzie sanitarie locali, nello specifico l'ASL di Asti, Genova e Aosta. Nel contesto sanitario si occupa dello sviluppo e della gestione dei sistemi per l'organizzazione interna di ASL e ospedali, relative integrazioni con enti terzi o Regione Piemonte per comunicare e ricevere fascicoli sanitari e dati, gestione di tutto il database ospedaliero contenente tutte le anagrafiche dei cittadini del territorio e relativi referti e documentazione medica.

Oltre ad offrire servizi alle Aziende Sanitarie, l'azienda ha diversi clienti sul territorio piemontese, sempre in abito pubblico, dove, anche in questo caso, sviluppa e gestisce piattaforme web, ma anche sistemi modello client server costruiti ad hoc per piattaforme windows.

Avendo come principali clienti realtà pubbliche che hanno bisogno di mantenimento e supporto costante nel tempo delle piattaforme, gli applicativi spesso diventano pesanti e di difficile gestione, questo di pari passo con i database su cui lavorano. La difficoltà principale del lavoro dell'azienda è la gestione per decine di anni di una piattaforma che per molteplici fattori non può rinascere o stravolgersi ad ogni piccola modifica richiesta, ma deve risultare duttile e di facile modificazione per poter implementare nel minor tempo possibile features richieste dal cliente.

L'azienda è strutturata in due team principali, gli sviluppatori che implementano praticamente le richieste dei vari clienti e il team di analisi dei requisiti, assistenza e supervisione di progetto. Quest'ultimo si occupa principalmente del rapporto con il cliente, andando ad effettuare lo studio di fattibilità, ad analizzare le richieste e mantenendo il rapporto durante lo sviluppo del sistema, per ricevere importanti feedback sulle funzionalità implementate. Inoltre, a progetto terminato, è il team ad occuparsi della presentazione, assistenza e supporto tecnico al cliente finale.

Inizialmente mi era stato proposto il ruolo di semplice sviluppatore, andando ad operare e basandomi sul lavoro di analisi svolto dal team dedicato in azienda. Confrontandomi con il mio tutor e manifestandogli il mio interesse ad approfondire anche lo studio e l'analisi dei requisiti del sistema per poter mettere in pratica ciò che ho imparato durante i corsi universitari, abbiamo deciso che la documentazione tecnica contenente la disciplina dei requisiti e relativi casi d'uso, sebbene già parzialmente preparata, non

mi sarebbe stata fornita, ma mediante i processi che ho studiato, l'avrei realizzata io passo passo, supervisionato dal team di analisi requisiti.

## 1.2 Proposta di progetto

La proposta di progetto nasce da una necessità riscontrata nel tempo da parte dell'assistenza tecnica dell'ASL di Asti. Nello specifico, si è sottolineata la difficoltà del monitoraggio dello stato dei referti medici, dell'andamento del processo di arrivo, relativo schedule e successivo invio a organi terzi. Essendoci un'enorme mole di documenti provenienti da realtà differenti sul territorio, era complicato sia per l'assistenza tecnica ASL fornire dettagli ai pazienti che riscontravano problemi nel consultare il loro referto, sia capire il reale stato del sistema che si occupa di prendere a carico i referti, processarli e inviarli a terzi, ad esempio alla Regione Piemonte.

Il focus dell'intero progetto proposto era di riuscire a creare un sistema molto semplice da utilizzare, ma che riuscisse a fornire tutti gli strumenti adeguati agli operatori che effettuano assistenza ai cittadini con problemi nella consultazione dei loro referti medici, ma anche la visione generale delle statistiche di referti ricevuti e referti inviati, con la possibilità di visualizzare periodi distinti. Gli operatori, infatti, per poter fornire supporto al cittadino, hanno bisogno di un modo agevole per ricercare i referti singoli o gruppi di referti e visualizzarne i dettagli in modo chiaro per capire dove sia il problema.

Oltre a questo, il sistema deve offrire la possibilità di estrarre le informazioni in dettaglio di ogni referto o gruppi di referti in formato xlsx, dando così la possibilità agli operatori di inviare le informazioni agli organi che le richiedono. Per ogni referto con errore bloccante, quindi che non permette al server di inviarlo alla sua destinazione, vi deve essere l'opzione per visualizzare l'errore o gli errori, esportarli, e consigli su come risolverli. Sempre per questa tipologia di referti, occorre un tasto che solo agli utenti amministratori permetta l'invio forzato, spostando quello bloccato al fondo della coda per ritentare l'inoltro: questo è pensato per dare la possibilità di risolvere un eventuale errore e di tentare l'invio immediatamente.

Il sistema sviluppato deve prevedere anche la possibilità di interagire con i servizi windows che sono in esecuzione sul server e che gestiscono tutto lo schedule dei processi, nello specifico deve poter visualizzare lo stato in real time per dare la possibilità all'utilizzatore dell'applicativo di capire se tutto sta funzionando correttamente, mentre per gli utenti amministratori necessita un tasto che, se premuto, spenga il servizio in modo sicuro, permettendo all'operatore di effettuare un riavvio totale.

Per evitare ritardi significativi dovuti ad un problema sul servizio windows che opera sull'elaborazione dei referti, è stato deciso di affiancare il

sistema ad un task che a ritmo regolare controllasse lo stato delle code dei referti in arrivo. Il numero elevato in una coda provoca rallentamenti significativi nel sistema che possono essere causati da problemi tecnici, come il thread interrotto, o riferiti a moltissimi referti arrivati di colpo. Quando la coda dei referti ricevuti in elaborazione diventa troppo lunga, viene notificata agli amministratori con una mail la problematica riscontrata, così possono sincerarsi dello stato dei thread ed eventualmente prendere dei provvedimenti per il rallentamento del sistema.

Dal cliente è stata richiesta anche la possibilità di predisporre il software per possibili ampliamenti futuri. Infatti una funzione non necessaria subito, ma che potrebbe dover essere implementata con il passare del tempo, sarebbe la priorità dinamica per tipologia di referto, una funzione che permette di assegnare priorità ad interi gruppi di documenti per poter inviare più velocemente quelli ritenuti più importanti.

Questa funzione è stata studiata data l'enorme quantità di certificati vaccinali che per via della campagna di vaccinazione che ha subito una grande accelerata, tendono a formare coda nel sistema. Sarebbe stato quindi utile poter assegnare dinamicamente priorità più alta ad esempio a questo tipo di documento per far defluire il carico.

Tutte queste funzionalità devono essere racchiuse in un sistema intuitivo e sempre disponibile, cercando di ottimizzare i tempi di ricerca, sia sui singoli referti sia sulle estrazioni di più giorni. Per questi motivi il sistema è stato richiesto sotto forma di Web Application costruita su server Windows, quello in cui lavorano i thread che gestiscono i referti.

## 2 Analisi preliminare

In questa sezione della relazione andrò ad analizzare tutta la fase preliminare che ho messo in pratica per sviluppare in modo corretto e coerente la piattaforma richiesta dal cliente. Per svolgere al meglio questa parte, ho messo in pratica ciò che ho studiato durante il corso di Sviluppo Applicazioni software nel corso del secondo semestre del terzo anno. Ho avuto un focus maggiore per quanto riguarda la creazione di Use case dettagliati per capire a fondo le necessità e le aspettative del committente della piattaforma.

Ho trovato infatti davvero utile un approccio iterativo e evolutivo perché sviluppare un sistema che ha a che fare con un settore specifico come quello medico non è cosa semplice, soprattutto perché molti elementi, essendo estremamente specifici, non si conoscono e può capitare di non accorgersi di dettagli che l'utente vorrebbe nella piattaforma, ma che lo sviluppatore ignora, ma grazie ad una buona disciplina dei requisiti e feedback costanti, questo non può succedere.

Per queste motivazioni la prima cosa che ho fatto è stato analizzare il contesto in cui si trovava attualmente l'ASL di Asti. Questo mi ha permesso di capire quali fossero prima tutte le problematiche e le difficoltà che hanno portato il cliente a richiedere lo sviluppo di questo sistema. Tale analisi mi ha anche permesso di capire quali fossero le azioni principali e su quali basare i casi d'uso dettagliati.

La mia idea sin dall'inizio è stata quella di avvicinarmi allo sviluppo in modo iterativo, andando a realizzare il sistema ricevendo feedback costanti dal cliente. In realtà non ho avuto rapporti diretti con il committente, ma ho avuto la fortuna di interfacciarmi con il team aziendale che si occupa proprio dello studio dei casi d'uso e delle richieste. Io, quindi, ponevo le domande ai componenti del team che mi rispondevano o domandavano a loro volta al cliente, creando in questo modo un dialogo indiretto ma efficace.

### 2.1 Situazione iniziale

Per capire quali fossero gli elementi più importanti della piattaforma, ho analizzato la situazione iniziale prima dell'introduzione del software.

- **Statistiche:** Non c'era un modo vero e proprio per capire se effettivamente si stesse formando coda in entrata o in uscita sul server. L'unico campanello d'allarme erano le molte segnalazioni di terze parti, o semplici pazienti in attesa dell'esito o la Regione Piemonte per

la comunicazione giornaliera degli esiti dei tamponi Covid. Quindi, per visualizzare il numero di referti in coda, era necessario eseguire un'interrogazione sul database per capire se ci fosse o meno un problema di congestione dei dati.

- **Assistenza Clienti:** l'assistenza non aveva un accesso semplice ai dettagli del referto di un utente e soprattutto non vi era un modo per capire se ci fosse un problema e nel caso, come risolverlo. Durante questa analisi si è osservato che il paziente, pur avendo correttamente pagato il ticket del referto, l'evento non venisse catturato correttamente e gestito dal processo sul server, di conseguenza nel database il ticket risultava non pagato. Per questo motivo il referto veniva scartato e reinserito in fondo alla coda, andando a riprocessarlo in un secondo momento. Questo errore comportava un enorme disagio all'utente che per visionare il proprio documento probabilmente avrebbe dovuto ripagare il ticket poiché diversamente il relativo referto sarebbe rimasto sempre in coda, in attesa di ricevere l'evento di pagamento. In questo sistema, contattando L'Assistenza interna, il dipendente di turno non aveva un modo per risolvere il problema se non segnalandolo all'assistenza tecnica che mediante una query sul database avrebbe aggiornato lo stato del pagamento sul relativo ticket.
- **Inoltro dati:** Spesso succede che servano riscontri di dati tra enti e aziende differenti, basti pensare ad una incongruenza del numero di tamponi effettuati o vaccinazioni fatte, in cui la Regione chiede chiarimenti o vuole verificare che i dati presenti nel sistema dell'ASL di Asti siano congrui a quelli precedentemente forniti alla Regione stessa. Prima della web application non vi era un modo diretto per estrarre in formato xlsx referti su interi periodi o di tipologia specifica. In questi casi si passava sempre dall'assistenza tecnica dell'azienda fornitrice del servizio che esportava il risultato dell'interrogazione e successivamente convertiva in excel le informazioni. Naturalmente questo approccio risultava poco efficiente sia per l'ASL in termini di tempistiche sia per l'azienda che doveva ogni volta eseguire query per ogni problema.
- **Gestione Attività:** Anche in questo caso non vi era un modo per capire se i processi sul server stessero svolgendo correttamente il loro compito, l'unica possibilità era navigare sul server windows e verificare sia che il servizio fosse in esecuzione che il rispettivo stato dei thread. Questo rappresentava un problema perché non avendo possibilità di monitorare lo stato del servizio in tempo reale e non conoscendo le

statistiche sulla dimensione delle code nel caso di un malfunzionamento, ci si sarebbe accorti del problema solo alla segnalazione di terzi di mancanza di documenti. Tale situazione ipoteticamente implicava un successivo rallentamento al sistema. In quel caso l'unica possibilità per procedere, era riavviare manualmente il servizio, spostare la coda congestionata e ripristinare il flusso di esecuzione, mandando la coda congestionata separatamente.

## 2.2 Necessità riscontrate

Alla luce di quanto mi è stato esposto dal committente, tenendo conto delle problematiche che hanno portato alla richiesta di sviluppare questo applicativo, mi sono reso conto che serviva una soluzione semplice, ma allo stesso tempo capace di permettere ad un operatore di eseguire tutte le operazioni necessarie nel modo più semplice ed intuitivo possibile, rese quasi impossibili dal precedente sistema operativo.

Le necessità principali riscontrate erano due, con caratteristiche differenti. In primo luogo serviva un'attività di monitoraggio passiva, in cui il sistema permettesse di avere una visione generale sul suo stato, riportando il numero delle code in tempo reale sul servizio e lo stato di quest'ultimo. Inoltre era necessario inserire anche la possibilità di ottenere statistiche in tempi differenti, come ad esempio visualizzare il numero di referti ricevuti in un dato periodo.

La seconda, con caratteristiche più attive, era un'attività di ricerca di referti, mediante filtri differenti applicabili anche in accoppiata. Il sistema doveva poi dare la possibilità sia di visualizzare tutti i dettagli che di eseguire tutte le funzioni sul singolo referto che precedentemente risultavano scomode o impraticabili da eseguire dall'operatore.

## 2.3 Fase di ideazione

Come prima cosa, avendo analizzato attentamente sia la proposta di progetto che la situazione attuale, ho iniziato ad immaginarmi la piattaforma, andando mentalmente a definire quali fossero le azioni principali e quali le tipologie d'utente che avrebbe interagito con essa.

Dal corso di Sviluppo e applicazioni software ho imparato che per la buona riuscita di un progetto è fondamentale

1. definire nei minimi dettagli ogni funzionalità della piattaforma,
2. dettagliare inizialmente le funzioni principali,
3. iniziare a svilupparle per ottenere feedback,
4. passare a quelle meno rilevanti.

Per queste ragioni la prima cosa che ho fatto è stato capire quali fossero i casi d'uso principali, dai quali avrei incominciato il mio sviluppo evolutivo. Per fare ciò ho inserito in una tabella le principali richieste, assegnando una priorità ad ognuna di esse; successivamente mi sono confrontato con un referente del team di analisi requisiti che ha preso le veci del cliente e al quale potevo porre domande sulle funzionalità.

### 2.3.1 Richieste Principali

<b>Richiesta</b>	<b>Caso d'uso</b>	<b>Priorità</b>
Visualizzare i referti filtrati	Consultare la tabella referti	Alta
Accedere al dettaglio dei singoli referti	Consultare la tabella referti	Media
Avere statistiche sullo stato delle code sul server	Visualizzare le statistiche	Alta
Estrarre i referti in formato xlsx	Consultare la tabella referti	Media
Visualizzare lo stato del servizio e riavviarlo in modo sicuro	/	Media
Possibilità di forzare l'invio o il pagamento di un referto in errore	Consultare la tabella referti	Alta
Accedere al dettaglio della tipologia dei documenti presenti nella coda	Visualizzare le statistiche	Media
Ricevere notifiche via mail in caso di accodamento	/	Bassa

## 2.4 Casi d'uso

Analizzando in dettaglio la proposta di progetto e confrontandomi con il team di analisi requisiti è emersa la necessità di redigere due casi d'uso, uno per ogni pagina della piattaforma e tipologia di interazione che può avere l'utente che interagisce con il sistema.

Ognuno di essi esprime in forma scritta cosa si aspetta di fare un utente che interagisce con la piattaforma; successivamente, partendo da quello, ho individuato i relativi casi d'uso dettagliati.

### 2.4.1 Attori

Per potermi approcciare in modo corretto alla costruzione dei casi d'uso, ho analizzato quali potessero essere gli attori che avrebbero interagito con il sistema da sviluppare. Dallo studio preliminare ho identificato due attori principali:

- **Utente normale:** per questa tipologia sulla piattaforma sono abilitate solo funzioni passive, di mera visualizzazione e ricerca, senza poter operare direttamente sulle priorità dei referti e relativi dettagli, con la possibilità di visualizzare lo stato del processo lato server aggiornato in tempo reale, ma non interagirci.
- **Amministratore:** Quest'ultimo ha la possibilità di svolgere tutte le attività di un utente normale, ma anche di eseguire operazioni attive come ritentare un invio, andando a spostare in modo indiretto un referto da una coda all'altra, forzare un pagamento o interagire con il servizio in esecuzione sul server.

### 2.4.2 Visualizzare le statistiche

L'utente che ha effettuato l'accesso nel sistema e interagisce con la sezione dedicata alle statistiche, di default visualizza i dati riferiti al giorno stesso. Opzionalmente può modificare il periodo mediante una data di inizio periodo e una data di fine periodo. Sulla pagina visualizza, inoltre, il numero relativo ai referti in coda presenti nel sistema, suddivisi per tipologia; opzionalmente può visualizzare i dettagli delle singole code suddivise per tipologie di documento. Se desidera maggiori dettagli, può visualizzare i referti presenti in ogni coda.

# Visualizzare le statistiche

## Informazioni generali

Nome caso d'uso: Visualizzare le statistiche

Portata: Sistema

Livello: Obiettivo utente

Attore primario: Operatore

Parti Interessate: Operatore

Pre-condizioni: L'attore deve essere autenticato nel sistema

Garanzie di successo o post-condizioni: /

Scenario principale di successo

#	Attore	Sistema
1	Opzionalmente applico un filtro	Viene associato il filtro alla ricerca corrente
	<i>Ripete 2 finché non soddisfatto</i>	
2	Eseguo la ricerca dei referti per visualizzarne le statistiche	Fornisce le statistiche contenente tutti i referti che rispettano i filtri applicati
3	Opzionalmente consulto i dettagli di una categoria	Fornisce i dettagli del numero di referti della categoria corrente
	<i>Ripete 4 finché non soddisfatto</i>	
4	Opzionalmente esporto i risultati presenti in tabella	Fornisce il documento contenente la tabella mostrata
	<i>Se desidera torno al passo 1 oppure termina il caso d'uso</i>	

## Estensione 3a

#	Attore	Sistema
3a.1	Seleziono la descrizione descrizione dei documenti medi	Fornisce i dettagli dei documenti medi
3a.2	Opzionalmente consulto i dettagli di una categoria	Fornisce i dettagli del numero di referti della categoria corrente
	<i>Ripete da 4a.1 finché non soddisfatto</i>	

## Estensione 3b

#	Attore	Sistema
3b.1	Opzionalmente visualizzare i dettagli della categoria in tabella	Fornisce la tabella contenente tutti i referti che rientrano nella categoria
	<i>Termina il caso d'uso</i>	

### 2.4.3 Consultare la tabella referti

L'Utente che ha effettuato l'accesso alla piattaforma e visualizza la sezione di dettagli dei referti presenti nel sistema, di default trova solo il filtro sulle date e può selezionare i referti ricevuti con successo nella data indicata. L'utente inserisce il filtro sul tipo di documento e ricerca. Il sistema produce l'output dei referti che sono conformi ai filtri applicati. L'operatore visualizza i dettagli di errore di un referto e ne scarica il contenuto.

## Consultare la tabella referti

### Informazioni generali

**Nome caso d'uso:** Consultare la tabella Referti  
**Portata:** Sistema  
**Livello:** Obiettivo utente  
**Attore primario:** Operatore  
**Parti Interessate:** Operatore  
**Pre-condizioni:** L'attore deve essere autenticato nel sistema  
**Garanzie di successo o post-condizioni:** /  
**Scenario principale di successo**

#	Attore	Sistema
1	Opzionalmente riordino la tabella	Salva la preferenza dell'utente sulla struttura della tabella
2	Opzionalmente applico un filtro	Viene associato il filtro alla ricerca corrente
	<i>Ripete 2 finché non soddisfatto</i>	
3	Eseguo la ricerca dei referti	Fornisce la tabella contenente tutti i referti che rispettano i filtri applicati
4	Opzionalmente consulto i dettagli di un referto	Fornisce i dettagli del referto selezionato
	<i>Ripete 4 finché non soddisfatto</i>	
5	Opzionalmente esporto i risultati presenti in tabella	Fornisce il documento contenente la tabella mostrata
	<i>Se desidera torno al passo 1 oppure termina il caso d'uso</i>	

#### Estensione 4a

#	Attore	Sistema
4a.1	Opzionalmente consulto i dettagli dell'errore di un referto	Fornisce i dettagli dell'errore del referto selezionato
4a.2	Opzionalmente scarico l'errore mostrato in tabella	Fornisce il documento contenente l'xml dell'errore
	<i>Ripete da 4a.1 finché non soddisfatto</i>	

#### Estensione 4b

#	Attore	Sistema
4b.1	Opzionalmente consulto i dettagli dell'invio di un referto	Fornisce i dettagli dell'invio del referto selezionato
4b.2	Opzionalmente scarico l'invio mostrato in tabella	Fornisce il documento contenente l'xml dell'invio
	<i>Ripete da 4b.1 finché non soddisfatto</i>	

#### Estensione 4c

#	Attore	Sistema
4c.1	Forzo il pagamento di un referto	Imposta il valore di pagato a U

#### Estensione 4a.1a

#	Attore	Sistema
4a.1a.1	Forzo l'invio di un referto in errore	Sposta il referto in cima alla coda

#### Eccezione 4a.1a

#	Attore	Sistema
4a.1a.1	L'operatore non ha i permessi necessari per effettuare questa operazione	Il sistema rifiuta l'operazione perché i permessi non sono sufficienti

#### Eccezione 4c

#	Attore	Sistema
4c.1	L'operatore non ha i permessi necessari per effettuare questa operazione	Il sistema rifiuta l'operazione perché i permessi non sono sufficienti

### 3 Sviluppo

Dopo aver iniziato ad analizzare i casi d'uso dettagliati, nella disciplina dei requisiti ho cominciato a sviluppare concretamente la piattaforma. Questa fase dello stage è stata quella più impegnativa perché ho dovuto realizzare concretamente le richieste del cliente che ha commissionato il sistema per il monitoraggio referti.

In questa fase di lavoro ho sfruttato i primi giorni per ambientarmi con le tecnologie utilizzate durante lo sviluppo, dettagliate nella sezione 4, Come IDE ho utilizzato Visual studio 2019 che mi ha fornito molti strumenti di supporto che hanno reso più semplice l'implementazione dei vari requisiti richiesti. Io non lo avevo mai utilizzato precedentemente, ma dopo un breve periodo di ambientamento, posso dire di essermi trovato bene ed aver apprezzato alcune funzionalità come l'editor per le interrogazioni al database che mi hanno semplificato il lavoro.

Gli strumenti da utilizzare per lo sviluppo non sono stati scelti direttamente da me poiché mi sono dovuto adeguare alle tecnologie e al modo di lavorare dell'azienda, standardizzato da diverso tempo. Sulle metodologie invece ho avuto parecchia libertà, potendo sperimentare ed applicare ciò che ho studiato durante i corsi universitari.

### 3.1 Approccio e metodologie utilizzate

Ho deciso assieme al mio tutor aziendale di lavorare su piccole iterazioni di una settimana, andando ad effettuare una riunione il venerdì pomeriggio per aggiornarci sullo stato dello sviluppo e risolvere eventuali dubbi. Inoltre per avere un'opinione sul mio operato, avrei potuto interfacciarmi con il team di analisi dati e rapporto con il cliente che, in caso di necessità, avrebbe chiesto chiarimenti al committente del progetto.

L'idea era quella di suddividere il lavoro in 7 settimane totali, avvicinandomi inizialmente alle funzioni più importanti che avevano bisogno di maggiori feedback da parte del cliente, lasciando per ultime quelle meno rilevanti come ad esempio il login, che sono standard senza necessità di supervisione e approvazione.

La metodologia di sviluppo evolutiva strutturata ad iterazioni, l'ho scoperta, affrontata e imparata durante questo anno universitario e prima di questo stage non l'avevo mai utilizzata in un contesto extra universitario. Questo progetto è stato un'ottima opportunità per utilizzare sul campo nozioni imparate a lezione e durante la sezione conclusiva tratterò le mie conclusioni sull'esperienza.

### 3.2 Iterazioni

1. Durante la prima iterazione, ho incominciato ad avvicinarmi al caso d'uso che ho ritenuto centrale, ovvero "Consultare la tabella dei referti". Per inquadrare la situazione, sono partito dallo studio del Database su cui avrei dovuto andare a lavorare, andando a capire quali tabelle erano necessarie e come collegarle. Individuate le tabelle principali, mi sono concentrato sull'individuare quali fossero gli elementi su cui l'operatore poteva filtrare i vari referti. Analizzando il caso d'uso, la proposta di progetto e confrontandomi con il team di analisi requisiti, ho stilato la seguente lista:

Campo	Tabella	Descrizione
DataRicezione	FSE_PURELIST	La data in cui è arrivato il referto
TypeCode	FSE_PURELIST	Il codice identificativo della tipologia del documento relativo al referto
UUID	FSE_PURELIST	l'ID univoco del referto, univoco nel mondo
ID FSE	FSE_PURELIST	l'ID del referto di FSE
UniqueID	FSE_PURELIST	l'ID del referto utilizzato per il salvataggio nel repository
Nome	ANAGRAFICA_XT	Il nome del paziente a cui fa riferimento il referto
Cognome	ANAGRAFICA_XT	Il Cognome del paziente a cui fa riferimento il referto
DataNascita	ANAGRAFICA_XT	La data di nascita del paziente a cui fa riferimento il referto, utile per evitare omonimie

I filtri applicati dovevano produrre la creazione di una tabella con tutti i referti, in modo da soddisfare i requisiti richiesti. Successivamente ogni referto, se selezionato, avrebbe mostrato tutti i suoi dettagli. Durante la prima iterazione mi sono concentrato nel riuscire ad applicare correttamente sulla lista i filtri che dovevano essere cumulabili ed offrire quindi filtri multipli. Per fare ciò ho quindi implementato query responsive in base ai filtri inseriti dall'operatore.

**Filtri**

ID FSE: <input type="text"/>	Data Inizio: 06/05/2020 <input type="text"/>	Data Fine: 25/05/2021 <input type="text"/>
UUID: <input type="text"/>	UNIQUEID: <input type="text"/>	TYPECODE: <input type="text" value="seleziona"/>
Cognome: <input type="text"/>	Nome: <input type="text"/>	Data Nascita: <input type="text"/>

Al termine della settimana ho fatto la riunione con il mio tutor aziendale per mostrare i progressi e discutere delle scelte implementative utilizzate.

- Partendo dalle basi gettate durante la prima iterazione, durante questa seconda mi sono concentrato sul gestire l'output delle query filtrate dall'utente e come da richiesta del cliente, ho organizzato i risultati in una tabella, in modo che ogni record rappresentasse un referto.



su cui era possibile filtrare i referti, pertanto io ho dovuto aggiungere un tasto per dare la possibilità all'operatore collegato alla piattaforma, mediante drag and drop, di alterare la struttura della tabella di base, andando a togliere o aggiungere colonne per poter personalizzare la vista in base alle esigenze.

Ogni modifica risultava associata al profilo utente dell'operatore, così da avere sempre la tabella come era stata lasciata.

4. Alla luce dei testing fatti dal team di analisi requisiti assieme al cliente, mediante prove sul campo del lavoro sviluppato fino a quel momento, è emersa l'esigenza di ulteriori filtri da applicare a priori sui filtri stessi. Nello specifico, si è visto necessario poter filtrare gruppi di referti solo di una particolare categoria di relazioni cliniche che si trovano in uno stato ben distinto. I referti infatti possono essere divisi in 4 stati distinti:

- Stato: **Inviato** - il referto è stato ricevuto ed è stato inviato correttamente in data: `DataUltimoTentativo`
- Stato: **Non Inviato** - il referto è stato ricevuto, ma non è ancora stato processato.
- Stato: **Errore** - il referto è stato ricevuto, è stato processato, ma è stato riscontrato un errore che ne ha impedito l'invio, quindi si trova al fondo della coda per essere pronto a ritentare l'invio perché l'errore è risolvibile.
- Stato: **Bloccato** - il referto è stato ricevuto, è stato processato, ma è stato riscontrato un errore bloccante che ne ha impedito l'invio, e che non si può risolvere.

Oltre alla possibilità di selezionare solo referti appartenenti ad uno stato preciso, è stato chiesto anche di poter scegliere su quale data cercare, in quanto ogni referto in tabella è caratterizzato da diverse date relative alle varie azioni che compie. Infatti inizialmente avevamo previsto di filtrare in base alla data di ricezione della relazione clinica da parte del repository interno. Tuttavia analizzando e testando sul campo la piattaforma, si è riscontrato come fosse opportuno, per alcune estrazioni particolari, modificare la data su cui si ricercava, andando ad utilizzare ad esempio "data ultimo tentativo" o la data in cui era stato prodotto il referto. Ogni referto possiede tre differenti date che identificano ognuna un particolare momento della storia della relazione clinica, nello specifico abbiamo:

- Data: **UltimoTentativo** - indica la data di ultimo tentativo da parte del processo sul server di invio del referto; può indicare quindi la data di invio in assenza di errori oppure la data in cui si è verificato l'ultimo errore;
- Data: **Ricezione** - indica la data in cui il referto è stato effettivamente ricevuto dal repository interno e coincide con il momento in cui è stato inserito in una delle code, solitamente quella di elaborazione referto;
- Data: **Referto** - indica la data vera e propria in cui è stato creato il referto fisicamente, ad esempio il giorno e l'ora in cui viene processato il tampone e se ne conosce l'esito, successivamente il referto viene elaborato dall'ente terzo e inviato al Fascicolo sanitario Elettronico.

Alla luce di queste necessità, ho provveduto ad ampliare la tabella contenente i campi del database sul quale andare ad eseguire la ricerca.

<b>Campo</b>	<b>Tabella</b>	<b>Descrizione</b>
DataRicezione	FSE_PURELIST	La data in cui è arrivato il referto
DataUltimoTentativo	FSE_PURELIST	La data in cui è stato tentato l'ultimo invio del referto, coincide con la data di invio se lo stato è a 1
DataReferto	FSE_PURELIST	La data in cui il referto è stato creato
TypeCode	FSE_PURELIST	Il codice identificativo della tipologia del documento relativo al referto
UUID	FSE_PURELIST	l'ID univoco del referto, univoco nel mondo
ID FSE	FSE_PURELIST	l'ID del referto di FSE
UniqueID	FSE_PURELIST	l'ID del referto utilizzato per il salvataggio nel repository
Nome	ANAGRAFICA_XT	Il nome del paziente a cui fa riferimento il referto
Cognome	ANAGRAFICA_XT	Il Cognome del paziente a cui fa riferimento il referto
DataNascita	ANAGRAFICA_XT	La data di nascita del paziente a cui fa riferimento il referto, utile per evitare omonimie
<i>Avanzate</i>		
Stato	FSE_PURELIST	l'ID univoco del referto, univoco nel mondo
Date	/	Modificando questo filtro si modifica la data a cui fa riferimento il periodo selezionato scegliendo tra { DataRicezione, DataUltimoTenativo, DataReferto }

Ho integrato questi nuovi filtri in una sezione aggiuntiva denominata "avanzate" all'interno del container dei filtri.

- In questa quinta iterazione ho incominciato lo sviluppo del secondo caso d'uso: Visualizzare le statistiche.

Analizzando le richieste, ho suddiviso in due sezioni la pagina mostrando nel lato sinistro i documenti ricevuti dal sistema e a destra quelli in-

viati, così da avere subito sotto controllo lo stato del processo. Graficamente ho optato per realizzare le varie sezioni racchiuse in box colorati, verde per le statistiche di invii e ricezioni avvenuti con successo, rosso per gli errori riscontrati e giallo per i referti in pending. Ogni sezione possiede una breve descrizione che la identifica, successivamente vi è presente il numero che mostra il reale numero di referti presenti nella coda.

Anche per le statistiche ho implementato un filtro per migliorarne la visualizzazione. L'operatore può scegliere gli estremi di estrazione con una data inizio e una data fine, di default è impostata la data del giorno, ma per un complessivo, per esempio mensile, è necessario solamente modificare le date. Come da richiesta, ogni sezione, oltre al numero totale di referti presenti, possiede un dettaglio aggiuntivo che permette di visualizzare da quali documenti è composto il numero totale, utile per capire quali referti sono stati inviati e quali sono in coda. In questo periodo storico di pandemia, ad esempio, è risultato cruciale capire quanti risultati dei test molecolari COVID-19 fossero stati effettivamente inviati.

Si è pensato inoltre di dare all'operatore anche la possibilità di scegliere come visualizzare la descrizione dei documenti, se con classificazione ASL che ha una granularità maggiore o se con descrizione Fascicolo sanitario elettronico con meno categorie specifiche.

Ogni box presenta anche un bottone che permette di visualizzare in tabella i referti presenti nel periodo selezionato, andando così a rendere possibile la visualizzazione dei dettagli dei singoli referti ed eseguire operazione come il forza invio nella categoria dei referti in errore. Per realizzare questa parte ho sfruttato i filtri avanzati, andando a realizzare tramite essi la stessa query che mostra i risultati all'interno del box statistiche, facendo in modo che questo risultasse completamente trasparente all'utente, favorendo l'esperienza dell'utente stesso.

**Documenti inviati a FSE**

Il numero di Documenti inviati a FSE con successo nel periodo selezionato

17/06/2020 - 24/06/2021

Mostra in tabella

Descrizione tipo documento medio (FSE)

Maggiori dettagli +

**4723**

**Documenti inviati con errore**

Il numero di Documenti inviati a FSE ma con errore nel periodo selezionato. (Attualmente in coda per il rinvio)

17/06/2020 - 24/06/2021

Mostra in tabella

Descrizione tipo documento medio (FSE)

Maggiori dettagli +

**62**

**Documenti da inviare**

Il numero di Documenti da inviare a FSE nel periodo selezionato

17/06/2020 - 24/06/2021

Mostra in tabella

Descrizione tipo documento medio (FSE)

Maggiori dettagli -

DESCRIZIONE	NUMERO
Referto Laboratorio Analisi	872
Scheda Vaccinale	194
Certificato Vaccinale	213
Lettera di Dimissione Ospedaliera	9

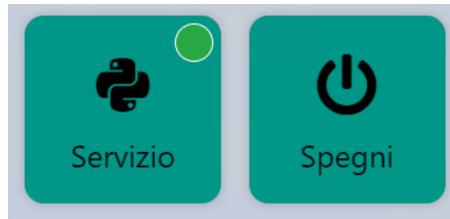
Pagina 3 di 3 (12 elementi) < 1 2 3 >

**1655**

Al termine della settimana, durante la riunione, ho mostrato i progressi del monitor e chiesto feedback sull'implementazione delle statistiche. La struttura è stata approvata, ma c'è stata la richiesta di un box aggiuntivo riguardante gli errori bloccanti e di uno riguardante la coda in uscita, da mostrare solo se di numero preoccupante. Quando la coda in uscita cresce a livelli sproporzionati, nel monitor 5000 referti per giorno di estrazione, è bene segnalarlo all'operatore perché potrebbe significare un malfunzionamento del sistema o un sovraccarico che comporta comunque rallentamenti nell'invio degli altri referti.

6. Nel corso della sesta iterazione ho aggiunto alla sezione statistiche, le sezioni richieste dal cliente, dopodiché mi sono concentrato sulla gestione del processo sul server che si occupa di shedulare i referti arrivati, elaborarli e successivamente inviarli. Come da richiesta iniziale,

il sistema sviluppato deve essere in grado in primis di fornire lo stato aggiornato e coerente del servizio in esecuzione e nel caso di operatore con privilegi sufficienti, dare la possibilità di eseguire un riavvio completo in totale sicurezza in modo user friendly.



Per riuscire a fare ciò ho iniziato creandomi un mio servizio placeholder che non esegue alcunché, per riuscire a vedere come intercettarne lo stato e come manipolarlo da interfaccia web.

Per quanto riguarda l'implementazione, ho inserito gli elementi all'interno della sidebar inserendo un riquadro per indicare lo stato in tempo reale e un bottone utilizzabile solo da utenti con livello adeguato per spegnere o accendere il servizio.

7. Come ultima iterazione ho lasciato la realizzazione dell'interfaccia di login. Per autenticarsi, all'operatore viene richiesto di inserire un username e una password. Per permettere l'autenticazione a tutte le tipologie di operatori, ho dovuto gestire il login andando a cercare in differenti server LDAP per verificare l'utenza, uno per il servizio interno all'ASL e uno per il supporto tecnico dell'azienda.

Sempre durante quest'ultima iterazione ho implementato il log delle azioni richiesto dal GDPR, nello specifico ogni azione, dal login alla ricerca, produce un log salvato sul database in una apposita tabella che permette di tenere traccia non solo di chi opera sul Monitor, ma anche a quali referti accede, quindi i relativi dati sensibili dei pazienti visualizzati e quali azioni compie.

### 3.3 Esperienza utente

Tutto il sistema segue una linea guida ben precisa, pensata per l'utente: facilitare chi interagisce con essa, permettendogli di eseguire tutte le operazioni con il minor numero di click. Ho deciso di suddividere la User Interface in due sezioni distinte:

- La sidebar: contiene le informazioni dell'utente loggato e il tasto di log out per poter terminare la sessione. Sempre nella sidebar ho inserito quattro bottoni che rappresentano le funzioni principali del software: visualizzare le statistiche, consultare la tabella dei referti, monitorare lo stato del servizio e interagire con esso. La sidebar può essere anche chiusa quando l'operatore vuole una visione completa della content section.
- La content section: in questa parte viene visualizzato il contenuto selezionato dall'operatore che sta utilizzando la piattaforma.

Tutta la web application è stata costruita per essere responsive e fruibile da ogni tipologia di dispositivo, anche se utilizzata prevalentemente da desktop. Per realizzare la UI sono partito da una bozza approvata dal cliente e durante lo sviluppo, iterazione dopo iterazione, attraverso i feedback, ho potuto modellare la grafica per renderla il più simile alle richieste. Una sezione che ha subito molte modifiche è stata la tabella in cui si visualizzano i referti. Inizialmente è stata pensata con colonne fisse e poche possibilità di personalizzazione, avente come unica interazione con l'utente, la possibilità di visualizzare i dettagli dei singoli referti. Successivamente invece dopo esplicita richiesta ho previsto la possibilità di modificare la tabella in tutte le sue colonne, così da rendere l'ambiente di lavoro più agevole per ogni operatore, le modifiche effettuate vengono salvate con un cookie sul browser in modo tale da offrire la medesima struttura all'operatore al successivo avvio del Monitor FSE. Oltre a questo ho aggiunto in ogni colonna selezionata la possibilità di eseguire ulteriori filtri oltre quelli che hanno mostrato i risultati, per offrire all'operatore una granularità maggiore nella ricerca.

### 3.4 Problemi affrontati

Durante lo sviluppo di questo progetto ho dovuto affrontare alcuni problemi dati generalmente dalle modifiche richieste dal cliente che ho gestito senza particolari difficoltà. Uno su tutti, però, ha creato in me apprensione: dovermi immergere in una realtà aziendale con metodologie di sviluppo consolidate e lontane dalla mia esperienza universitaria e personale. Inizialmente non poter utilizzare nessun pattern architetturale come ad esempio model view controller, mi ha messo in difficoltà perché non sapevo come creare una struttura solida, producendo codice riutilizzabile e pulito. Per questi motivi, soprattutto la prima iterazione è stata quasi più di adattamento al modo di lavorare dell'azienda per capire cosa fare e come. Mi è

stato chiesto di sviluppare la web app con ASP.Net con approccio WebForm, con backend in C# o vbNet e frontend in HTML5 e CSS3.

Io finora avevo utilizzato .Net solo nella sua versione MVC.

Un altra aspetto con cui ho dovuto prendere confidenza soprattutto all'inizio, è stato il data base con cui ho dovuto lavorare, quindi capire esattamente il significato delle tabelle e degli attributi che essendo del settore medico, da subito non mi erano del tutto chiari.

Il data base su cui ho lavorato era una porzione del database aziendale, nello specifico ho interagito con sette tabelle contenenti, però, molte decine di attributi ciascuna. Su di essi sono andato ad eseguire i filtri da mostrare in tabella, i dettagli dei referti e produrre le statistiche. E' stato difficile inoltre capire cosa era necessario e cosa, invece, poco significativo, soprattutto perché spesso i nomi degli attributi delle tabelle erano poco auto-esplicativi. Inoltre, data l'assenza di chiavi esterne nelle tabelle, era difficile capire anche come avvenivano le relazioni tra di esse. Soprattutto in questa fase, è stato fondamentale l'aiuto del mio tutor aziendale che mi ha permesso di capire il significato delle tabelle, degli attributi, quando usarli e come. Il data base, naturalmente era una versione, sandbox, creata appositamente per me al fine di poter oscurare per privacy i dati dei pazienti.

## 4 Tenologie utilizzate

Per realizzare la piattaforma ho utilizzato diverse tecnologie riguardanti l'ambito web. Molte di esse già le conoscevo e le padroneggiavo grazie all'università e ad esperienze personali, altre invece non le avevo mai utilizzate prima e grazie a questo progetto ho potuto capirle, comprenderne l'uso ed imparare ad applicarle.

Come standrd di sviluppo ho cercato di attenermi il più possibile a come lavorava l'azienda, ho ritenuto molto importante gestire al meglio i commenti sul codice, questo aspetto è sempre molto importante, ma lo è ancor di più quando si sviluppa qualcosa che nel tempo sarà sicuramente rimaneggiato e ampliato da qualche altro sviluppatore, ho cercato di rendere tutto il più chiaro possibile orientato alla modifica ed ampliamento futuro.

### 4.1 ASP.Net

Il nome ASP.NET deriva da ASP (Active Server Pages), la vecchia tecnologia per lo sviluppo web di Microsoft che fino a qualche anno fa rappresentava lo standard. Nonostante ciò esistono differenze sostanziali fra queste due tecnologie. Infatti ASP.NET si basa, come tutte le applicazioni della famiglia Microsoft .NET, sul CLR (Common Language Runtime).

Gli sviluppatori possono scrivere codice utilizzando uno qualsiasi dei linguaggi di alto livello supportati dal framework (libreria di classi) .NET, come, ad esempio, Visual Basic .Net, C# quello che ho usato io per il mio progetto di stage, J# (derivato da Java), ma anche, molti altri linguaggi open source, come Perl e Python, questo è possibile poiché il CLR è stato standardizzato presso ECMA (European Computer Manufacturers Association associazione fondata nel 1961 e dedicata alla standardizzazione nel settore informatico).

Le applicazioni ASP.NET risultano significativamente più veloci e performanti rispetto a quelle realizzate utilizzando altre tecnologie di scripting, in quanto l'intero codice del sito web è pre-compilato in pochi file gestiti da un server web.

ASP.NET nasce con l'idea di proporsi per semplificare la migrazione degli sviluppatori dalle applicazioni Windows alle applicazioni web, mettendoli in grado di generare pagine composte da tanti controlli widget, del tutto simili a quelli usati dall'interfaccia utente di Windows.

Un controllo web, come ad esempio un pulsante o un'etichetta (label), funzionano in modo del tutto simile a quello delle loro controparti Windows:

il codice ad essi associato può assegnare loro determinate proprietà ed eseguire azioni in risposta ad eventi ad essi associati (ad esempio alla pressione di un pulsante). La differenza sta nel fatto che, mentre i controlli Windows disegnano se stessi sullo schermo, i controlli web producono blocchi di codice HTML che vanno ad inserirsi automaticamente nella pagina che deve essere inviata al browser dell'utente finale.

#### 4.1.1 DevExpress

DevExpress è un prodotto software sviluppato dall'omonima azienda produttrice di software. Questo strumento mi è stato fornito direttamente dall'azienda perché utilizzato in tutti i loro progetti, il suo obiettivo è quello di semplificare lo sviluppo sia delle web app .NET che delle win application. Sostanzialmente è un insieme di dll che una volta aggiunta al progetto e referenziata nel file cshtml, aggiunge centinaia di controllo e strumenti già pronti da integrare all'interno del prodotto software che si sta sviluppando.

Nel mio caso ho utilizzato prevalentemente la gridview, una griglia per visualizzare dati predisposta appunto da devExpress che ho adattato per renderla simile alla richiesta fatta dal cliente. Oltre a questo ho anche utilizzato il timer, un oggetto che si occupa di chiamare una funzione lato server ogni tot tempo, questo è stato necessario per aggiornare lo stato in tempo reale dello stato del processo in esecuzione sul server.

## 4.2 Database relazionale Oracle

Come database ho utilizzato Oracle, è uno tra i più famosi software di database management system sviluppato da Oracle Corporation. Scritto in linguaggio C, che fa parte dei cosiddetti RDBMS (Relational DataBase Management System) ovvero di sistemi di database basati sul modello relazionale affermatosi come standard di riferimento dei database dell'ultimo decennio.

Ho utilizzato SQL, un linguaggio standardizzato per database basati sul modello relazionale (RDBMS), che permette attraverso serie di comandi di eseguire:

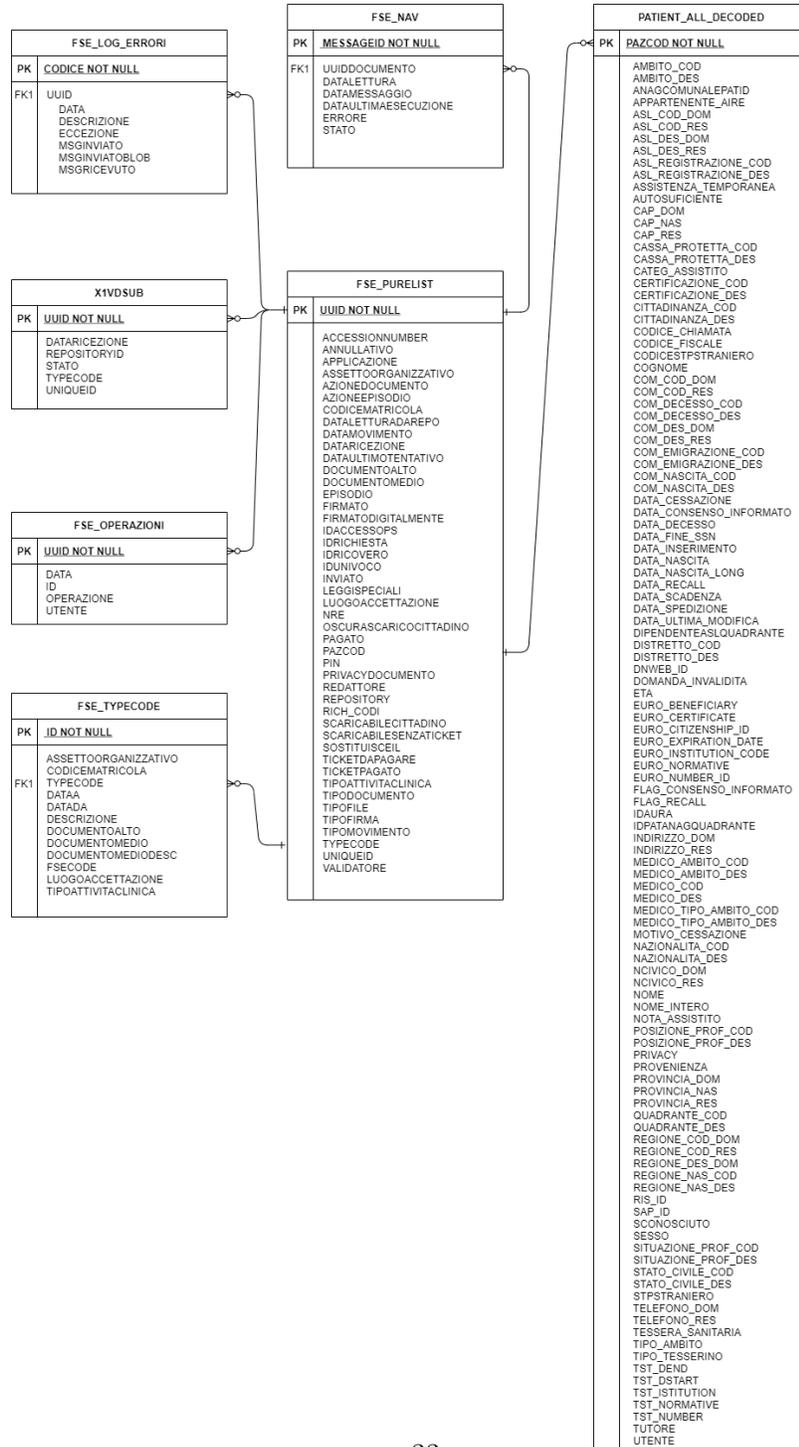
- DDL Data Definition Language: creare e modificare schemi di database
- DML Data Manipulation Language: inserire, modificare e gestire dati memorizzati
- DQL Data Query Language: interrogare i dati memorizzati

- DCL Data Control Language: creare e gestire strumenti di controllo e accesso ai dati

Come software per la gestione del database ho utilizzato Toad, sotto consiglio del mio tutor aziendale. Toad permette di creare e gestire database Oracle in maniera semplice e veloce. Tutte le operazioni vengono eseguite semplicemente utilizzando un'interfaccia grafica, si occupa Toad stesso a trasformare i comandi in comandi SQL.

Una funzione offerta dal software, che ho trovato molto utile, è stata la possibilità di ottimizzare le query, essendo cruciale avere interrogazione il più snelle possibile e questo strumento mi è venuto in soccorso in più di un'occasione. Per ottenere l'interrogazione ottimizzata è stato necessario inserire la query desiderata e anche quanto precisa si vuole l'ottimizzazione, a questo punto toad in automatico fa partire un processo che esegue ripetutamente la query modificandone la struttura, assegnando un punteggio ad ogni esecuzione fatta e infine fornendo l'interrogazione che ha fornito il punteggio più alto.

## 4.2.1 Schema relazionale



### 4.3 HTML5

HTML acronimo di HyperText Markup Language non possiamo definirlo come un vero e proprio linguaggio di programmazione bensì è più corretto inserirlo nella categoria dei linguaggi di markup. Nato per la formattazione, costruzione e impaginazione di documenti ipertestuali disponibili nel web 1.0, tuttavia oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web (definita interamente appunto dal markup) e la sua rappresentazione stilistica, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal W3C (World Wide Web Consortium). Nasce come derivato dall'SGML, un metalinguaggio finalizzato alla definizione di linguaggi utilizzabili per la stesura di documenti destinati alla trasmissione in formato elettronico. Oggi HTML ha raggiunto la sua quinta versione rilasciata sempre da W3C nell'ottobre 2014.

Il motivo principale che ha spinto il W3C e i suoi membri a sviluppare HTML5 è stata la necessità di fornire direttamente le funzionalità che in precedenza erano fruibili tramite estensioni proprietarie all'esterno dei browser, come il noto Adobe Flash oggi definitivamente obsoleto, e simili. Un secondo obiettivo di HTML 5 che gli sviluppatori si erano prefissati era quello di riuscire a garantire una maggiore compatibilità tra i diversi browser, indipendentemente dalla piattaforma software utilizzata rendendo trasparente il passaggio da uno all'altro all'utente.

L'HTML in definitiva è un linguaggio di formattazione che viene utilizzato per descrivere le modalità di impaginazione o visualizzazione grafica detta layout del contenuto, testuale e non, di una pagina web attraverso elementi chiamati tag di formattazione. Sebbene l'HTML supporti l'inserimento di script e oggetti esterni quali immagini o filmati, non è un linguaggio di programmazione: non prevedendo alcuna definizione di variabili, strutture dati, funzioni o strutture di controllo che possano realizzare programmi, il suo codice è in grado soltanto di strutturare e decorare dati testuali, inoltre non viene compilato, ma interpretato direttamente dal nostro browser.

Il linguaggio HTML, ha come scopo quello di gestire i contenuti associandone o specificandone allo stesso tempo la struttura grafica all'interno della pagina web da realizzare grazie all'utilizzo di tag diversi. Ogni tag (ad esempio `<h1>` o `<h2>`) specifica un diverso ruolo dei contenuti che esso contrassegna (quindi il tag `<h1>` definirà un'importanza maggiore del tag

`<h2>`), queste definizioni sono interne al browser da cui apriamo il contenuto. La formattazione consiste nell'inserimento nel testo di marcatori o etichette, detti tag, che descrivono caratteristiche come la funzione, il colore, le dimensioni, la posizione relativa all'interno della pagina. I browser che leggono il codice mostrano all'utente formattazioni predefinite per ogni tag che incontrano (così ad esempio i contenuti marcati con il tag `<h1>` avranno carattere 18pt e i contenuti marcati da `<p>` avranno carattere 12pt). Tuttavia questa formattazione è completamente sotto il controllo dell'utente, che può modificarla nelle Impostazioni del suo browser. Lo sviluppatore della pagina HTML può definire stili aggiuntivi per andare a sovrascrivere le regole di default per ogni elemento ad esempio rendendo i caratteri racchiusi dal tag `<h1>` di 24pt;

Nello sviluppo del progetto di stage HTML ha svolto un ruolo da protagonista essendo dato il linguaggio che ho utilizzato per la creazione dell'intera struttura della piattaforma web, nello specifico ho utilizzato HTML 5 unendolo a CSS e Script per ottenere un lavoro pulito.

## 4.4 CSS

Il linguaggio CSS viene usato per definire la formattazione o layout di documenti HTML, XHTML e XML, principalmente conosciuto nell'ambito web e relative pagine web. Le regole per comporre il CSS sono contenute in un insieme di direttive emanate a partire dal 1996 dal W3C che tuttora fornisce validatori e supporto tecnico agli sviluppatori web.

L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una programmazione più chiara, facile da utilizzare e che favorisce il riutilizzo del codice, sia per gli autori delle pagine stesse sia per gli utenti, garantendo contemporaneamente anche il riutilizzo di codice ed una sua più facile manutenzione.

I CSS introducono due concetti principale che danno allo sviluppatore la possibilità di organizzare meglio la struttura del progetto, le classi e gli id, la classe è un insieme di regole stilistiche caratterizzate da un nome preceduto da un `”.`”, a differenza degli id le classi sono riutilizzabili per tutto il progetto non dovendo così scrivere più volte le stesse regole per ottenere il medesimo risultato.

#### 4.4.1 Bootstrap

Durante il corso della mia esperienza sia universitaria che personale, ho imparato ad utilizzare Bootstrap, uno dei framework CSS più utilizzati ed imitati per la sua semplicità di utilizzo e ottima resa stilistica. Bootstrap può a buon diritto considerarsi lo standard de facto dei framework CSS per lo sviluppo di interfacce Web, trovando applicazione in ambiti come la creazione di template HTML preconfezionati ed i temi per i principali CMS, soprattutto in un'ottica responsiva, che di questi tempi è diventata uno standard. La sua nascita lo vede come un progetto interno a Twitter ad opera degli sviluppatori Mark Otto e Jacob Thornton ed oggi indipendente, Bootstrap può a buon diritto considerarsi il re dei framework per lo sviluppo di interfacce web.

A corroborare questo assunto una serie di dati:

- è il progetto più popolare sulla piattaforma GitHub, con oltre 58.000 star e più di 20.000 fork;
- secondo recenti statistiche tracce del codice di questo framework si ritrovano sull'1% dei siti web mondiali, una cifra che tradotta in termini assoluti si traduce in un numero senz'altro ragguardevole;
- gode di un supporto nella community degli sviluppatori che non ha pari rispetto agli front-end framework;
- chi lo usa può contare su una mole enorme di risorse e ausili di ogni tipo;

nel corso del tempo è diventato un vero e proprio standard come punto di partenza in ambiti come il web developing e sempre più spesso viene richiesta come conoscenza da avere per considerarsi un full stack developer. L'uscita della versione 4.0, con tutte le migliorie che porta con sé, non potrà che rafforzare questo scenario.

#### 4.5 JavaScript

JavaScript è un vero e proprio linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client (successivamente esteso poi anche al lato server con framework come Node.js) per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

Originariamente sviluppato da Brendan Eich della Netscape Communications con il nome di Mochan e successivamente di LiveScript, in seguito è stato rinominato "JavaScript" ed è stato formalizzato con una sintassi più vicina a quella del linguaggio Java di Sun Microsystems (che nel 2010 è stata acquistata da Oracle). Standardizzato per la prima volta il 1997 dalla ECMA con il nome ECMAScript, l'ultimo standard, di giugno 2017, è ECMA-262 Edition 8 ed è anche uno standard ISO (ISO/IEC 16262).

Le funzioni di script, utilizzati dunque nella logica di presentazione, possono essere opportunamente inserite in file HTML, in pagine JSP o in appositi file separati con estensione .js poi richiamati nella logica di business. Ultimamente il suo campo di utilizzo è stato esteso alle cosiddette Hybrid App (app ibride), con le quali è possibile creare app per più sistemi operativi utilizzando un unico codice sorgente basato appunto su JavaScript, HTML e CSS.

nel contesto del mio progetto javascript mi ha permesso di gestire la maggior parte degli eventi in modo asincrono e trasparente all'utente, uno su tutti l'aggiornamento in tempo reale dello stato del servizio a livello server

#### **4.5.1 JQuery**

jQuery nasce come una libreria JavaScript per applicazioni web, distribuita come software open source, distribuito sotto i termini della Licenza MIT. Nasce con l'obiettivo di semplificare la selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi DOM in pagine HTML, nonché semplificare l'uso di funzionalità AJAX, la gestione degli eventi e la manipolazione dei CSS.

Le sue caratteristiche permettono agli sviluppatori JavaScript di astrarre le interazioni a basso livello con i contenuti delle pagine HTML. L'approccio di tipo modulare che implementa jQuery consente la creazione semplificata di applicazioni web e contenuti dinamici versatili. Nel 2020, jQuery risulta la libreria JavaScript più utilizzata in Internet, ovvero è presente nel 74,4% dei primi 10 milioni di siti Internet più popolari secondo W3Techs, questo è dovuto alla sua immediatezza, semplicità d'uso e licenza gratuita.

#### **4.6 LDAP**

LDAP acronimo di Lightweight Directory Access Protocol è un protocollo standard, introdotto da Microsoft, utilizzato per l'interrogazione e la modifica dei servizi di directory, come ad esempio un elenco aziendale di email o

una rubrica telefonica, o più in generale qualsiasi raggruppamento di informazioni che può essere espresso come record di dati e organizzato in modo gerarchico. Nel contesto in cui l'ho utilizzato io permette di ottenere dato un utente in primis se e esiste e se la password inserita è corretta, successivamente la gestione della sua priorità in modo tale da permettere o meno azioni delicate e iterazioni dirette con lo stato del server e dei referti.

L'integrazione con questo protocollo è stato semplice e ha permesso di rendere il login della piattaforma coerente conforme con il resto dei servizi offerti internamente all'ASL di Asti.

#### **4.7 Servizi Windows**

Nello sviluppo ho dovuto andare a creare un'iterazione tra l'Interfaccia web e i servizi in esecuzione sul server per rendere possibile l'aggiornamento in tempo reale dello stato e la gestione di riavii manuali da parte degli operatori.

Un servizio di Windows è un programma eseguibile che svolge compiti specifici e che è progettato per non dover richiedere l'intervento dell'utente. Normalmente i servizi di Windows vengono avviati quando il sistema operativo Windows è avviato, e girano in background per tutto il tempo durante il quale gira Windows. Concettualmente sono simili ai daemon di Unix.

## 4.8 Caratteristiche e punti di forza

I punti di forza delle tecnologie utilizzate in particolare di ASP.NET, cuore del sistema che ho sviluppato, a mio modo di vedere sono le seguenti:

- Semplicità di utilizzo e della strutturazione del codice
- Buon possibilità di integrare il sistema con dll windows e servizi lato server
- Gestione della sicurezza embedded in .NET
- integrazione tra backend e frontend molto intuitiva

Come discuterò nella seguente sezione Webform a mio modo di vedere ha moltissimi lati positivi ed è orientato alla semplicità di implementazione, tuttavia per progetti di grandi dimensioni tende a creare confusione data dalla gestione non separata e strutturata delle interazioni tra backend e frontend. WebForm infatti creare queste pagine con estensione .cshtml, una sorta di frontend compilato assieme alla parte C# lato backend questo come inserito nei punti di forza è un grande plus a livello di sicurezza perché si occupa di molte cose .NET, di contro invece viola le regole di separazione e indipendenza tra parte visiva e logica applicativa.

## 5 Proposte Alternative

In questa sezione andrò ad analizzare e confrontare le tecnologie utilizzate per sviluppare la piattaforma e l'architettura MVC studiata all'università. Questo perché se avessi avuto completa libertà nella scelta delle tecnologie da utilizzare per lo sviluppo della piattaforma probabilmente il progetto sarebbe stato molto diverso ed avrei optato appunto per strutturare maggiormente il codice.

Io prediligo le web application strutturate con una netta separazione tra backend e frontend con chiamate asincrone gestite, ad esempio, attraverso ajax così da rendere completamente indipendenti le due parti. Questo purtroppo non è stato possibile attuarlo con ASP.NET con webform infatti questo approccio si basa sulla creazione di pagine che contengono sia codice html che C# compilato che comunica direttamente con il server andando a creare pagine contenenti sia struttura che logica applicativa.

### 5.1 Pattern architetturale MVC

Durante il corso di programmazione III, svolto nel primo semestre del terzo anno accademico, ho affrontato il pattern architetturale Model-view-controller sia a livello teorico che pratico, attraverso un progetto.

MVC è molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti e in applicazioni web in grado di separare nettamente la logica di presentazione dei dati dalla logica di business, questo pattern si posiziona nel livello logico o di business e di presentazione in una architettura multi-tier.

Il pattern MVC è composto da tre componenti differenti ognuno con un ruolo ben definito, il modello cattura il comportamento dell'applicazione in termini di dominio del problema, andando appunto a modellare generalmente attraverso classi la logica di base, tutto ciò indipendentemente dall'interfaccia utente.

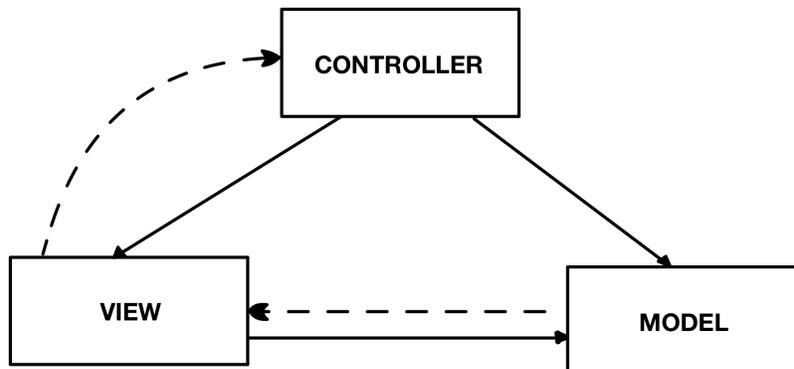
Oltre a questo il modello gestisce direttamente i dati, la logica e le regole dell'applicazione.

Il secondo componente invece è la vista, essa può essere una qualsiasi rappresentazione in output di informazioni, come un grafico o un diagramma. Questo è l'unico elemento visibile all'utente ed è quello con cui interagisce sia direttamente che indirettamente, naturalmente sono possibili viste multiple delle stesse informazioni, come ad esempio un grafico a barre per la gestione e la vista tabellare per l'amministrazione.

La terza ed ultima parte, il controller, accetta l'input e lo converte in comandi per il modello e/o la vista. Possiamo considerarlo come l'elemento che si pone in mezzo tra gli altri due componenti facendo li cooperare indirettamente per raggiungere l'obiettivo.

La struttura tipica quindi è caratterizzata dall'interazione tra componenti del pattern MVC andando a separare i compiti e le responsabilità fra i componenti software che interpretano tre ruoli principali:

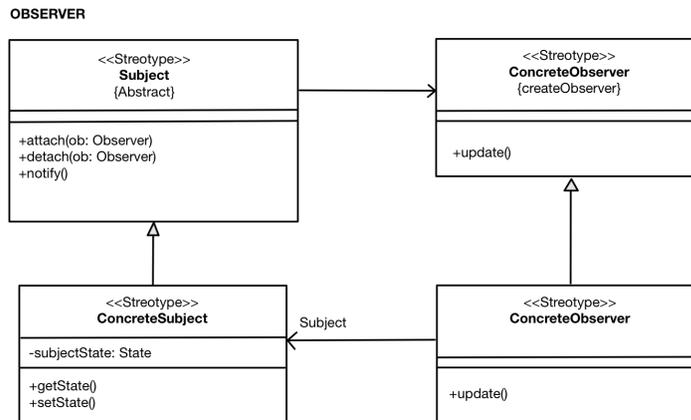
- il model fornisce i metodi per accedere ai dati utili all'applicazione;
- il view visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- il controller riceve i comandi dell'utente, in genere catturati attraverso il view, e li attua modificando lo stato degli altri due componenti.



Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata "logica di business"), a carico del controller e del model, e l'interfaccia utente a carico del view.

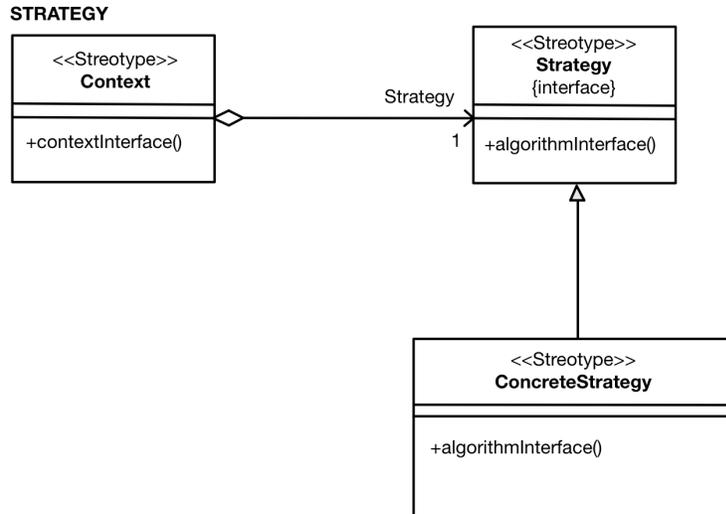
I dettagli delle interazioni fra questi tre oggetti software dipendono molto dalle tecnologie usate come il linguaggio di programmazione, le eventuali librerie, middleware e via dicendo, e dal tipo di applicazione. Nel mio caso, durante il corso di programmazione III, abbiamo affrontato il pattern MVC associato a java nello sviluppo di un client e di un server mail. Quasi sempre la relazione fra view e model è descrivibile anche come istanza del pattern architetturale Observer, implementato ad esempio in java

con la classe Observable e l'interfaccia Observer. Questo pattern è cruciale nell'implementazione dell'MVC perché permette di creare una dipendenza tra oggetti di tipo uno-a-molti: quando lo stato di un oggetto cambia, tale evento viene notificato a tutti gli oggetti dipendenti, essi vengono automaticamente aggiornati. Questo permette ad esempio di registrare un evento particolare ed associarlo al click di un bottone che verrà catturato dalla vista e gestito dal controller.



A volte, quando è necessario cambiare il comportamento standard degli elementi dell'applicazione a seconda delle circostanze, il controller implementa anche il pattern Strategy, esso si occupa del modo in cui un oggetto esegue un determinato compito: incapsula un algoritmo.

Un'implementazione diversa che realizza (fondamentalmente) la stessa cosa, in modo che un'implementazione possa sostituire l'altra a seconda della strategia richiesta. Questo pattern l'ho approfondito nel corso di Sviluppo Applicazioni Software, ne segue una rappresentazione grafica.



## 5.2 Punti di forza

Il pattern MVC oggi giorno è diventato un vero e proprio standard in moltissime realtà aziendali perché attraverso la sua netta distinzione garantisce una serie di caratteristiche fondamentali per progetti di grosse dimensioni, quali:

- le classi che formano l'applicativo possono essere più facilmente riutilizzate;
- l'applicativo è organizzato in parti semplici e comprensibili (ogni parte ha le sue specifiche finalità) favorendone l'ampliamento futuro;
- la modifica di una parte non coinvolge e non interferisce con le altre parti (maggiore flessibilità nella manutenzione del software).

Avendo potuto sperimentare durante il percorso universitario con un progetto itinerante questo approccio, mi sono reso conto di come, sebbene più verboso all'inizio nella definizione del model, permetta uno sviluppo più comodo, standardizzato e orientato al riutilizzo del codice, favorendo inoltre modifiche future con poco sforzo.

### 5.3 confronto con webForm

Sono due approcci molto diversi tra loro che a mio parere non sono completamente sovrapponibili, infatti la scelta di MVC secondo me è giustificata per progetti di grosse dimensioni in cui pagine (viste) differenti devono comunicare tra di esse scambiandosi informazioni, strutturando il codice e standardizzando gli elementi come appunto richiede l'architettura semplificata sia la stesura che i futuri ampliamenti. WebForm invece associa ad ogni pagina (vista) un codice lato server che gira durante l'esecuzione e l'interazione di quell'unica pagina.

Oltre a questo, un'altra differenza sostanziale tra le due tecnologie è l'iterazione tra la gestione delle request e delle rispettive response, in MVC viene gestita e delegata interamente al o ai controller che si occupano di elaborare la request e interagire con i model e viste opportune, in webform invece anche in questo caso viene delegato tutto al codice associato alla pagina che ha scatenato un evento.

## 6 Conclusioni

Questo stage è stato molto importante per la mia formazione, sia sotto un punto di vista professionale sia personale. Sin dall'inizio ho preferito fare un'esperienza fuori dall'università per potermi mettere alla prova nel mondo vero, rispetto alla proposta di un progetto interno in cui, come sempre avrei deciso io metodologie, tecnologie e modalità implementative senza scendere a compromessi. Nel mondo del lavoro mi è parso capire invece fondamentale la capacità adattativa, riuscire ad imparare in breve tempo qualcosa anche di mai visto prima.

Purtroppo vista la crisi sanitaria e il lockdown totale durante il mio periodo di stage, non ho potuto vivere l'ufficio e il mondo che ruota attorno all'azienda come avrei potuto fare andando in presenza; nonostante ciò posso ritenermi soddisfatto del rapporto creato tra me e il mio tutor aziendale. Una variabile anche molto importante data dalla possibilità di lavorare da casa è la comodità, per essere al lavoro è necessario accendere il computer e collegarsi alla VPN aziendale, questo mi ha permesso di recuperare molto tempo soprattutto nella pausa pranzo permettendomi di continuare a seguire i corsi universitari che naturalmente proseguivano.

Questo stage mi ha anche permesso di dimostrare le mie capacità sul campo. Mi sono sempre ritenuto abbastanza bravo in quello che faccio, ma senza mettersi alla prova in contesti del tutto nuovi, fuori dalla comfort zone, è difficile dimostrarlo in primis a se stessi. A sostegno di ciò, al termine dello stage, il mio tutor aziendale mi ha proposto di diventare suo collaboratore lavorando a progetto, permettendomi così di gestire meglio i tempi e di proseguire con la laurea magistrale. Questa proposta mi ha fatto sentire apprezzato, bravo nel mio lavoro e che fosse davvero soddisfatto della piattaforma che ho realizzato. Anche per questo motivo ci tengo a ringraziarlo.

In conclusione posso ritenermi pienamente soddisfatto di questa esperienza, che sebbene a livello accademico non mi abbia insegnato molto, al contrario mi ha fatto entrare nel mondo del lavoro e cominciare a capire molte cose su di esso .

### 6.1 Obiettivi raggiunti

Nella situazione iniziale, prima dello sviluppo del monitor FSE, le operazioni richieste risultavano difficili e macchinose; alla fine dello sviluppo del sistema tutte le richieste fatte dall'ASL di Asti in sede di proposta di progetto

e successive modifiche sono state rispettate, i feedback ricevuti sono stati utilissimi sia per poter modificare il lavoro sia per poter ampliare le richieste iniziali in base all'utilizzo sul campo della piattaforma stessa. Il software finale ha superato le mie aspettative iniziali: lavorare con tecnologie nuove in un contesto del tutto inesplorato, inizialmente mi aveva destabilizzato, ma fortunatamente grazie anche al mio tutor sono riuscito ad ambientarmi in un breve periodo e riuscire ad esprimere tutte le mie capacità sviluppate e coltivate sia all'università che con progetti personali. La piattaforma finale, nelle sue due sezioni risulta come segue:

**Tabella**

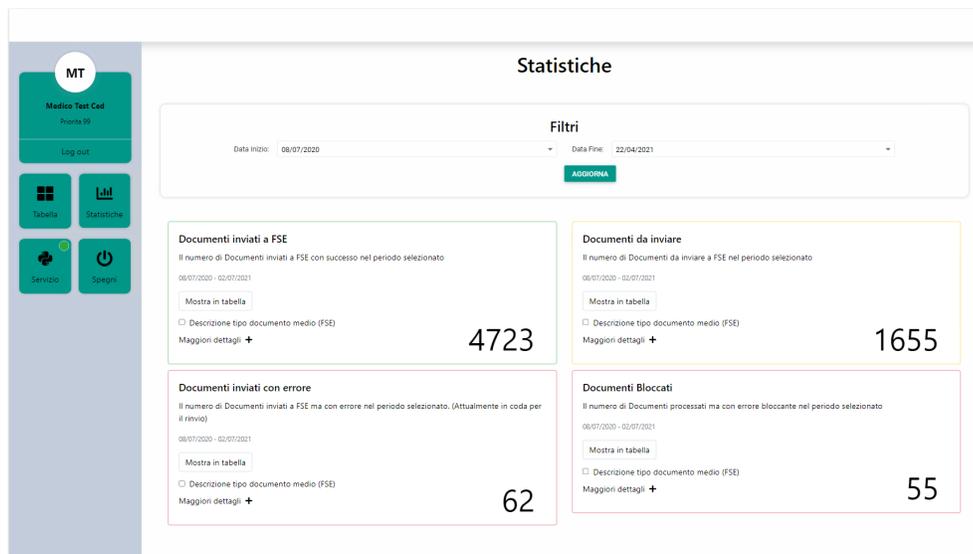
**Filtri**

ID FSE:  Data Inizio: 08/07/2020 Data Fine: 22/04/2021  
 UID:  UNIDUEID:  Descrizione: Referto Laboratorio Analisi  
 Cognome:  Nome:  Data Nascita:

Filtri Avanzati +  
 FILTRA

Personalizza Esporta come

DESCRIZIONE	DATAREFERTO	OSCURASCARICITTADINO	SCARICABILECITTADINO	EPISODIO	DATA INVIO FSE	STATO
> Referto Laboratorio Analisi	17/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare
> Referto Laboratorio Analisi	16/03/2021	N	0		17/03/2021 00:00:00	Da Inviare



## 6.2 Possibili ampliamenti

La piattaforma realizzata è stata sviluppata pensando anche al suo possibile ampliamento, infatti durante il periodo finale dello stage, quando la campagna vaccinale ha subito una forte accelerata, è nata l'idea di implementare in futuro una nuova sezione nel monitor per poter gestire ed assegnare dinamicamente delle priorità ai referti per poter gestire meglio le code. Questo perché data l'enorme mole di certificati vaccinali che vengono prodotti ogni giorno e inviati al repository per essere elaborati e successivamente inviati tendono a creare coda, ma soprattutto a rallentare referti più importanti. Per questo motivo è nata l'idea di poter assegnare dinamicamente priorità in base allo stato della cosa a tipologie di referti distinte così facendo si possono far defluire senza causare rallentamenti.

Oltre a questo, con il passare del tempo, se fosse necessario aggiungere altre statistiche per aumentare la granularità delle informazioni fornite, è possibile aggiungere semplicemente i box nella sezione apposita che sono già standardizzati con classi css per essere pronti all'uso così da permettere alla piattaforma di crescere ed evolvere nel tempo in base alle esigenze riscontrate dai vari operatori.

## 7 Sitografia

- [en.wikipedia.org](http://en.wikipedia.org), LDAP
- [dotnet.microsoft.com](http://dotnet.microsoft.com), LDAP
- [dotnet.microsoft.com](http://dotnet.microsoft.com), pattern MVC .NET
- [it.wikipedia.org](http://it.wikipedia.org), HTML5
- [en.wikipedia.org](http://en.wikipedia.org), JavaScript
- [stackoverflow.com](http://stackoverflow.com), MVC
- [c-sharpcorner.com](http://c-sharpcorner.com) WebForm and MVC
- [en.wikipedia.com](http://en.wikipedia.com) bootstrap