Impact Analysis

Vanessa Scovotto

Vincenzo Esposito

Giuseppe Valitutto

Gino Farisano

Indice

Aggiunte rispetto al precedente sistema

Gestione Paziente

1. Login(Client Utente)

Nel sistema precedente solo la segretaria poteva accedere al sistema ed effettuare tutte le operazioni che il sistema offriva, un utente non poteva loggarsi. Nel nuovo sistema l'utente attraverso un nome utente e una password inviatogli via email dalla segreteria può accedere al sistema.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Application Logic, Storage e Interface, in quanto:

- Al livello "Application Logic", si deve sviluppare del codice in più per poter rendere operativa la nuova funzionalità; in particolare si deve sviluppare un metodo, che permetta il login del paziente.
- Al livello "Interface": Apparirà una form di inserimento: nome utente e password

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - viene creata una form di inserimento in cui l'utente può inserire il nome utente e la password.	-Implementazione di una funzione che permetta l'autenticazione di un utente.	/
Modified	/	/	/
Eliminated	/	/	/

2. Registra Nuovo Paziente (Client Segretaria)

A differenza del sistema precedente viene aggiunta una nuova funzionalità per registrare un nuovo paziente da parte della segretaria.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Application Logic, Storage e Interface, in quanto:

- Al livello "Application Logic", si deve sviluppare del codice in più per poter rendere operativa la nuova funzionalità; in particolare si deve sviluppare un metodo, che permetta la registrazione di un nuovo paziente.
- Al livello "Interface", dalla schermata principale che si apre dopo avere effettuato il login da parte della segretaria comparirà un button "Registra Nuovo Paziente", cliccando su di esso apparirà una form di inserimento dei dati personali del paziente.
- Al livello "storage"; si crea una nuova tabella "Utente" in cui sono memorizzate le informazione del cliente che si è appena registrato al sistema.

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - inserimento nella pagina principale del sistema di un button “Registra Nuovo Paziente” tramite il quale si possono inserire attraverso la Form “ Registra Paziente” tutti i dati personali del paziente. Dopo aver inserito i dati la segreteria può effettuare o l’inserimento effettivo del paziente, attraverso il button “Inserisci” o resettare ciò che si è scritto nella form attraverso il button “resetta” o chiudere la finestra con il button “Esci”.	Implementazione di una funzione che controlla se la form di inserimento dati è piena, se tutti i dati sono stati inseriti correttamente, li raccoglie per memorizzarli nel database.	Nuove funzionalità: -è presente un gestore dell’evento(classe Registra Paziente) che interagirà con il database per poter aggiungere dati alla tabella “utente”, questa classe interagisce con il database tramite una chiamata al server.
Modified	/	/	/
Eliminated	/	/	/

3. Modifica Paziente(Client Segretaria)

Nel sistema precedente la segreteria non poteva modificare i dati di un paziente già registrato nel sistema.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello "Application Layer" si deve implementare una funzione che modifichi i dati di un paziente già registrato.
- Al livello "Interface" ; dalla schermata principale la segreteria attraverso il button "Modifica Paziente" accede ad una schermata in cui viene mostrata una lista contenente tutti i pazienti già registrati, cliccando su un paziente comparirà una form da cui si possono modificare i dati.
- Al livello Storage: viene aggiornata la tabella "Utente"

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - inserimento nella pagina principale del sistema di un button "Modifica Paziente" tramite il quale si aprirà una form "Lista Pazienti" che mostra una lista di tutti i pazienti del sistema già registrati precedentemente dalla segreteria . Selezionando un paziente si aprirà una nuova form "Modica Paziente" in cui sono modificabili solo alcuni campi. Dalla form "Modifica Paziente" la segreteria può o	-Implementazione di una funzione che non fa altro che modificare i dati di un cliente che sono stati inseriti precedentemente	/

	rendere effettive le modifiche attraverso il button “Modifica” o resettare i dati attraverso il button “Resetta” o uscire attraverso il button “Esci”.		
Modified	/	/	Viene aggiornata la tabella “utente”
Eliminated	/	/	/

3.1. Modifica Paziente(Client Utente)

Nel sistema precedente l’utente non era presente, quindi abbiamo aggiunto una nuova funzionalità per far si che l’utente possa modificare i suoi dati personali.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l’esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello “Application Layer” si deve implementare una funzione che permetta all’utente dello smartphone di accedere all’applicazione e modificare le sue credenziali.
- Al livello “Interface” ; comparirà una form in cui sono presenti tutti i dati dell’utente, l’utente può modificarli ad eccezione del nome, cognome e codice fiscale.
- Al livello “Storage” viene aggiornata la tabella “Utente”

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - inserimento nella home di un button “Gestione Utente” tramite il quale si aprirà una form in cui sono presenti tutti i dati di quel cliente, l’utente selezionando un campo può modificarlo. Dopo aver effettuato le modifiche cliccando sul button “invio dati” i dati vengono inviati al server e il server risponde con uno status html.	-Implementazione di una funzione che non fa altro che modificare i dati di un cliente che sono stati inseriti precedentemente.	/
Modified	/	/	Viene aggiorntata la tabella “Utente”
Eliminated	/	/	/

4. Storico Paziente(Client Segretaria).

Nel sistema precedente la segreteria non poteva avere informazioni riguardanti le visite dei pazienti.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l’esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello “Application Logic” si deve implementare una funzione che permetta la visualizzazione delle storico visite dei pazienti.
- Al livello “Interface” ; dalla schermata principale la segreteria attraverso il button “Storico Paziente” accede ad una schermata in cui viene mostrata una lista contenente tutte le informazioni personali del paziente e

selezionando un paziente comparirà una nuova lista che visualizza tutte le visite effettuate da quel paziente.

- Al livello “Storage” ; viene aggiunta una nuova tabella “Storico”.

	Interface Layer	**Application Logic Layer**	**Storage Layer**
New	Nuova interfaccia: - inserimento nella pagina principale del sistema di un button ”Storico Paziente” tramite il quale si aprirà la form “Lista Pazienti” che mostra una lista di tutti i pazienti del sistema già registrati precedentemente dalla segreteria . Selezionando un paziente si aprirà una nuova form “Storico di “nome e “cognome” del paziente selezionato in cui compare una lista di tutte le visite che quel paziente ha effettuato ed è presente unacheckbox “pagato” che indica l’avvenuto pagamento di un servizio da parte del paziente.	-Implementazione di una nuova funzione che permetta la visualizzazione delle visite effettuate da un cliente. -Implementazione di una nuova funzione che permette di settare il pagamento del servizio offerto al cliente.	Nuove Funzionalità: -E’ presente un gestore dell’evento (Classe VisualizzaStorico) che interagirà con la tabella “Prenotazione” e “Storico” .Le due taballe sono collegate tra di loro, ovviamente la tabella “Prenotazione” è collegata alla tabella “Utente” attraverso la chiave esterna “codiceFiscale” . Questa associazioni di chiavi mi permette di avere tutte le visite che uno specifico cliente ha effettuato.
Modified	/	/	/
Eliminated	/	/	/

4.1 Storico Paziente(Client Utente).

Nel sistema precedente non era implementata la funzionalità che permetteva all'utente di poter visualizzare lo storico delle sue visite.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello "Application Layer" si deve implementare una funzione che permetta la visualizzazione delle storico visite relative al cliente in uso dell'applicazione.
- Al livello "Interface" ; viene implementata una slider con cui l'utente cliccando su "Storico" comparirà una schermata con tutte le informazioni relative alle sue visite.
- Al livello "Storage" ; viene aggiunta una nuova tabella "Storico".

	Interface Layer	**Application Logic Layer**	**Storage Layer**
New	Nuova interfaccia: - Dopo aver cliccato "Gestione Utente" comparirà una slider in cui l'utente può visualizzare 3 cose: ▪ Dati Utente ▪ Storico ▪ Appuntamento Selezionando storico comparirà un'altra schermata in cui l'utente può inserire una data di inizio e una data di fine, cliccando sul button "Cerca" comparirà una lista con	-Implementazione di una nuova funzione che permetta la visualizzazione delle visite effettuate da un cliente.	Nuove Funzionalità: -E' presente un gestore dell'evento (Classe VisualizzaStorico) che interagirà con la tabella "Prenotazione" e "Storico" .Le due taballe sono collegate tra di loro, ovviamente la tabella

	tutte le visite effettuate dall'utente in quel periodo.		"Prenotazione" è collegata alla tabella "Utente" attraverso la chiave esterna "codiceFiscale" . Questa associazioni di chiavi mi permette di avere tutte le visite che uno specifico cliente ha effettuato.
Modified	/	/	/
Eliminated	/	/	/

5. Ricerca Paziente

Nel sistema precedente la segreteria non poteva effettuare una determinata ricerca su un paziente.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello "Application Layer" si deve implementare una funzione che permetta la ricerca di un paziente da parte della segretaria.
- Al livello "Interface" : la segretaria dopo aver inserito il nome o il cognome di un paziente cliccando sul button "Ricerca" verranno visualizzati i pazienti relativi a quel nome o cognome.

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: -Dalla schermata principale la segreteria cliccando o sul button “Modifica Paziente” o sul button “Storico Paziente” apparirà la form descritta precedentemente “Lista Pazienti” , da questa form la segretaria inserisce o il nome o il cognome di un paziente(non è necessario inserire il nome o il cognome per intero) e cliccando il button “Ricerca” apparirà una nuova form “Lista Pazienti” .	-Implementazione di una funzionalità che mi permetta di poter ricercare un paziente che ovviamente è già stato registrato nel sistema dalla segreteria.	/
Modified	/	/	/
Eliminated	/	/	/

6. Elimina Appuntamento

Nel sistema precedente la segreteria non poteva eliminare un appuntamento registrato nel sistema.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l’esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello “Application Layer” si deve implementare una funzione che elimini un appuntamento già registrato.

- Al livello “Interface” ; cliccando su un appuntamento comparirà una schermata di riepilogo della prenotazione, da tale schermata la segretaria può eliminare un appuntamento.
- Al livello “Storage” ; si va ad eliminare un appuntamento contenuto nella tabella “prenotazione” che già esisteva nel sistema precedente.

	Interface Layer	**Application Logic Layer**	**Storage Layer**
New	Nuova interfaccia: - dalla schermata principale con un doppio click sulla prenotazione apparirà la form “Gestione Prenotazione”; tale form conterrà un Button “Elimina”, cliccandoci su la prenotazione verrà eliminata.	Sviluppo di un metodo che prevede l’eliminazione di un appuntamento già presente nel sistema	
Modified	/	/	Implementazione: -E’ presente un gestore che interagirà con il database per permettere l’eliminazione di una prenotazione dalla tabella “Prenotazione”, ovviamente viene aggiornata anche la tabella “Storico” in quanto è collegata con la tabella “Prenotazione”.
Eliminated	/	/	/

7. Visualizza Appuntamento(Client Utente)

L'utente può visualizzare il suo prossimo appuntamento.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello "Application Layer" si deve implementare una funzione che permetta la visualizzazione degli appuntamenti suoi passati.
- Al livello "Interface" ; l'utente visualizzerà il prossimo appuntamento

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - Dopo aver cliccato "Gestione Utente" comparirà una slider in cui l'utente può visualizzare 3 cose: ▪ Dati Utente ▪ Storico ▪ Appuntamento Selezionando "Appuntamento" comparirà un'altra schermata in cui l'utente può visualizzare solo il prossimo appuntamento.	-Implementazione di una nuova funzione che permetta la visualizzazione del prossimo appuntamento del cliente specifico.	
Modified	/	/	/
Eliminated	/	/	/

8. Inserimento Appuntamento (Client Utente)

Inserimento di un appuntamento da parte dell'utente.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto su tutta l'architettura del sistema, in quanto:

- Al livello "Application Logic", si deve sviluppare del codice in più per poter far si che l'utente possa effettuare una prenotazione.
- Al livello "Interface",: comparirà una sorta di "Calendario" con tutti i dottori disponibili e gli orari liberi, dopo aver selezionato un orario l'utente deve scegliere solo il servizio di cui ha bisogno.
- Al livello "storage"; viene modificata la tabella "Calendario" che viene anche rinominata in "Prenotazione".

	Interface Layer	**Application Logic Layer**	**Storage Layer**
New	Nuova interfaccia: - dalla home l'utente selezionando "Gestione Prenotazione" gli apparirà una sorta di "Calendario" , selezionando un'ora evidenziata in verde(Per indicare che in quell'ora quel dottore è libero) le apparirà una listview da cui l'utente può selezionare il servizio che	Modifica funzionalità: - sviluppo di un metodo che permetta di inserire un appuntamento, una volta che l'utente ha effettuato la prenotazione, questi dati saranno inviati al server che risponderà con uno status html.	

	desidera.		
Modified	/	/	La tabella “Celendario” è stata rinominata con la tabella “Prenotazione” e alcuni campi sono stati modificati: - nella tabella precedente venivano memorizzati nome e cognome dell’utente ora invece abbiamo utilizzato una chiave esterna “codiceFiscale” che permette il collegamento con la tabella “Utente” in cui sono memorizzate le informazioni relative ad un paziente. - prima venivano memorizzati nome e cognome del personale ora invece abbiamo una chiave esterna “idPersonale” che permette il collegamento con la tabella “Personale” già esistente nel vecchio sistema.
Eliminated	/	/	

Modifiche rispetto al precedente sistema

Gestione Prenotazione

9. Inserimento Appuntamento

Inserimento di un appuntamento da parte della segretaria.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto su tutta l'architettura del sistema, in quanto:

- Al livello "Application Logic", si deve sviluppare del codice in più per poter rendere operativa la modifica alla funzionalità; in particolare si deve sviluppare un metodo che ,permetta di inserire una nuova prenotazione.
- Al livello "Interface", nella schermata principale apparirà una sorta di "calendario" in cui vengono mostrati tutti gli orari disponibili per il giorno che desidero effettuare la prenotazione.
- Al livello "storage"; viene modificata la tabella "Calendario" che viene anche rinominata in "Prenotazione".

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - dalla schermata principale la segretaria dopo aver selezionato l'orario in cui vuole fissare l'appuntamento e il dottore con cui vuole fissare l'appuntamento,	Modifica funzionalità: - sviluppo di un metodo che permetta di inserire un appuntamento, metodo interagirà.	

	comparirà la form “Prenotazione”. Da questa form la segreteria può scegliere il tipo di servizio e cliccando invio le comparirà una lista di tutti i clienti presenti nel sistema ; dopo di che cliccando sul Button “Prenota” la prenotazione viene effettuata. Dopo aver effettuato la prenotazione la riga del “calendario” si evidenzia.		
Modified	/	/	La tabella “Celendario” è stata rinominata con la tabella “Prenotazione” e alcuni campi sono stati modificati: - nella tabella precedente venivano memorizzati nome e cognome dell’utente ora invece abbiamo utilizzato una chiave esterna “codiceFiscale” che permette il collegamento con la tabella “Utente” in cui sono memorizzate le informazioni relative ad un

			paziente. - prima venivano memorizzati nome e cognome del personale ora invece abbiamo una chiave esterna "idPersonale" che permette il collegamento con la tabella "Personale" già esistente nel vecchio sistema.
Eliminated	/	/	

10. Ricerca Prenotazione

E' stata modificata la funzionalità "Ricerca Prenotazione" per renderla più user-friendly e più chiara per chi va ad utilizzare il sistema.

Tecnologie necessarie

Per scrivere questa funzionalità è stato utilizzato il linguaggio di programmazione Java, in particolare le tecnologie utilizzate sono: Web-services per l'esposizione del servizio, un DBMS per gestire e conservare i dati e JSON per la comunicazione.

Questa aggiunta ha impatto sul livello Storage, Application Layer ed Interface e in quanto:

- Al livello "Application Logic" si deve implementare una funzione che permetta alla segretaria di ricercare una prenotazione in maniera molto veloce e efficiente.
- Al livello "Interface" ;comparirà una tabella in cui sono visualizzati gli appuntamenti di una determinata data con i relativi dottori che hanno effettuato il servizio.

	Interface Layer	Application Logic Layer	Storage Layer
New	Nuova interfaccia: - sviluppo di un'interfaccia, che renda possibile la ricerca di una prenotazione attraverso la selezione di una data; dopo aver selezionata la data comparirà una sorta di "calendario" con tutte le prenotazioni presenti per quella data.	Modifica funzionalità: -implementazione di una funzionalità che permetta di ricercare una prenotazione .	/
Modified	/	/	
Eliminated	/	/	/