



Studenti: *Tornabene Alessio*
Musumeci Giorgia
Arena Giorgia

Sommario

Fase di Ideazione.....	4
Documento di Visione	4
Modello dei casi d'uso	Errore. Il segnalibro non è definito.
Documento di Visione	17
Regole di business	17
Specifiche Supplementari.....	18
Glossario.....	18
Analisi orientata agli oggetti	19
Introduzione	19
Modello di Dominio	20
SSD e Contratti	22
Contratti Operazioni UC1	24
Contratti Operazione UC2	24
Contratti Operazioni UC3	25
Contratti Operazioni UC4	25
Contratti Operazioni UC5	28
Contratti Operazioni UC6	28
Contratti Operazioni UC10.....	31
Contratti Operazioni UC11.....	31
Contratti Operazioni UC12.....	32
Contratti Operazioni UC9	34
Progettazione	35
Diagramma delle classi.....	35
Diagrammi di sequenza.....	38
UC1.....	38
UC2.....	39
UC3.....	40
UC4.....	41
UC5.....	42
UC6.....	42
UC10	45

UC11	45
UC12	47
UC3.....	48
UC10	48
UC9.....	48
Testing	51
Introduzione	51
Individuazione dei casi di test e testing unitario	51
Iterazione 1:.....	51
Iterazione 2:.....	53
Iterazione 3:.....	54
Iterazione 4:.....	55
Test di Sistema	57
Refactoring e conclusioni	58
Refactoring.....	58
Test di Accettazione	58

Fase di Ideazione

Documento di Visione

INTRODUZIONE

Si vuole realizzare un'applicazione per la gestione di un laboratorio di analisi cliniche, chiamata MedLab che permette di gestire le prenotazioni dei test delle varie sedi del laboratorio.

Il sistema permetterà di ottimizzare la gestione delle prenotazioni per i test di laboratorio, la pianificazione delle risorse, la registrazione dei pazienti, e la visualizzazione dei risultati dei test, migliorando l'esperienza sia del personale del laboratorio che dei pazienti.

OPPORTUNITÀ DI BUSINESS

Gestire un laboratorio di analisi utilizzando i sistemi cartacei o strumenti non integrati porta spesso alla generazione di errori, che possono rallentare le operazioni e compromettere la qualità del servizio offerto. MedLab è stato ideato per eliminare questi problemi, offrendo un sistema digitale e automatizzato che semplifica ogni aspetto.

Grazie a MedLab i laboratori possono organizzare e gestire in modo ottimale le prenotazioni dei test, garantendo che le risorse necessarie, siano utilizzate in modo efficiente. Inoltre, MedLab consente di creare e mantenere aggiornati i profili dei pazienti, inclusi i dati personali e lo storico, facilitando così l'accesso ad informazioni essenziali. Infine, l'app consente di consultare i risultati degli esami in maniera rapida e sicura, migliorando significativamente la comunicazione tra il laboratorio ed i pazienti.

DESCRIZIONE DELLE PARTI INTERESSATE

Gli utenti principali di MedLab sono:

- Amministratore: registra ed elimina i profili dei pazienti, gestisce le varie sedi e ne aggiunge di nuove, gestisce i profili del personale di laboratorio, inserisce tutti gli esami possibili dei vari laboratori e genera i report.
- Personale del laboratorio: inserisce i referti nei vari profili dei pazienti e verifica se sono prenotati ad un dato esame.
- Paziente: modifica il proprio profilo con dati personali e i dati medici, consulta il catalogo degli esami disponibili per la prenotazione, sceglie la sede di preferenza, visualizza i risultati dei test, visualizza le prenotazioni attive, fa delle recensioni.

DESCRIZIONE GENERALE DEL PRODOTTO

MedLab sarà accessibile sia dal personale del laboratorio, dall'amministratore e dai pazienti attraverso un'interfaccia semplice e intuitiva. Una volta effettuato l'accesso al proprio profilo, i pazienti potranno visualizzare direttamente gli esami prossimi prenotati, con i dettagli relativi alla data, all'orario e al tipo di analisi da effettuare, garantendo così un'esperienza semplice e organizzata.

Modello dei casi d'uso

REQUISITI

MedLab è un sistema software per la gestione di un laboratorio di analisi cliniche. Tale applicazione si occupa di:

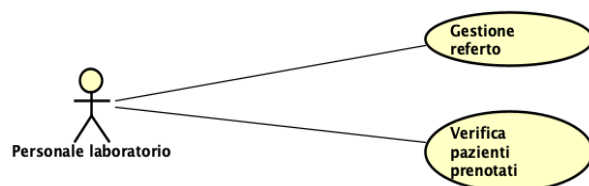
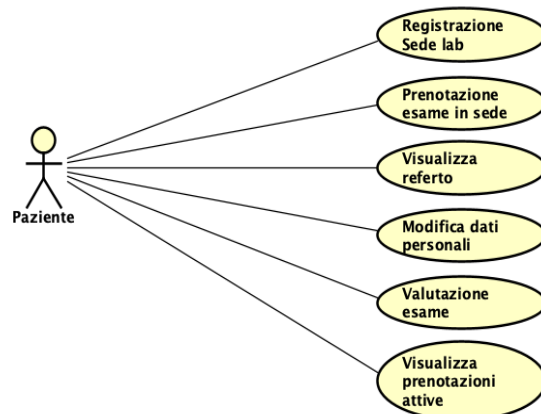
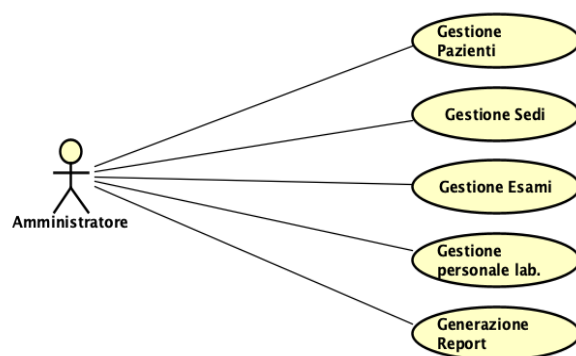
- Gestire i test di laboratorio (inserimento, modifica, rimozione e visualizzazione).
- Gestire la registrazione e il profilo dei pazienti.
- Pianificare e organizzare le prenotazioni dei test.
- Visualizzare i risultati dei test da parte dei pazienti.
- Generare report e statistiche per l'analisi delle attività del laboratorio.
- Gestire il personale.
- Gestire sedi del laboratorio.
- Gestire recensioni da parte dei pazienti.

OBIETTIVI E CASI D'USO

Analizzando i requisiti riportati, sono stati individuati gli attori principali e i loro obiettivi. Da queste informazioni sono stati ricavati i seguenti casi d'uso principali:

ATTORE	OBIETTIVO	CASO D'USO
Amministratore	Creare, modificare o eliminare il profilo di un paziente.	UC1: Gestione pazienti MedLab (CRUD)
Paziente	Una volta iscritto al sistema inserisce la sede di preferenza e può aggiornarla.	UC2: Registrazione sede laboratorio
Paziente	Una volta scelta la sede, prenota un dato esame.	UC3: Prenotazione esame in sede
Amministratore	Inserisci, aggiorna, elimina sede.	UC4: Gestione sedi (CRUD)
Personale laboratorio	Visualizza la lista dei pazienti prenotati per un dato esame e inserisci stato (completato o annullato).	UC5: Verifica pazienti prenotati
Personale laboratorio	Crea, modifica ed elimina i referti di un paziente che ha effettuato un dato esame.	UC6: Gestione referto (CRUD)
Paziente	Visualizza la lista dei referti associati agli esami.	UC7: Visualizza referto
Paziente	Modifica i propri dati personali.	UC8: Modifica dati personali
Paziente	Inserisce, modifica ed elimina le recensioni	UC9: Valutazione esame (recensione) (CRUD)

Amministratore	Crea, modifica ed elimina gli esami	UC10: Gestione esami (CRUD)
Amministratore	Crea, modifica ed elimina un personale nel sistema	UC11: Gestione personale laboratorio (CRUD)
Amministratore	Generare i report statistici sulle prenotazioni e sugli esami prenotati	UC12: Generazione report
Paziente	Visualizza le future prenotazioni	UC13: Visualizza prenotazioni attive



CASI D'USO

UC1: Gestione pazienti MedLab

Nome caso d'uso	UC1: Gestione pazienti MedLab (CRUD)
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Amministratore
Parti interessate e interessi	Amministratore: vuole registrare, modificare o eliminare i profili dei pazienti per una gestione organizzata del laboratorio. Il sistema deve mantenere l'integrità dei dati relativi ai pazienti, evitando incongruenze.
Pre-Condizioni	L'amministratore deve essere autenticato nel sistema MedLab.
Garanzia di successo	Il profilo del paziente è correttamente creato, aggiornato o eliminato.
Scenario principale di successo	<ol style="list-style-type: none">1. L'amministratore accede alla gestione dei pazienti.2. Sceglie l'azione desiderata "Aggiungi Paziente".3. Inserisce i dati necessari (nome, cognome, data di nascita, codice fiscale, sesso, malatoCronico).4. Il sistema verifica la validità dei dati.5. L'amministratore conferma l'operazione.6. Il sistema aggiorna tutti i dati con le modifiche.
Estensioni	*a. i dati inseriti sono incompleti o errati: il sistema notifica l'errore e richiede la correzione.
Requisiti speciali	Verifica automatica dell'unicità del codice fiscale.
Elenco delle varianti tecnologie e dei dati	
Frequenza di ripetizione	Ogni volta che si registrano nuovi pazienti.

UC2: Registrazione sede laboratorio

Nome caso d'uso	UC2: Registrazione sede laboratorio
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Paziente
Parti interessate e interessi	Il paziente vuole selezionare una sede tra quelle disponibili per effettuare esami. Il sistema deve mantenere l'associazione tra paziente e sede.
Pre-Condizioni	Il paziente è registrato e autenticato nel sistema.

Garanzia di successo	Il paziente effettua correttamente la scelta della sede di preferenza.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il paziente accede alla funzione "Registrazione nuova sede". 2. Visualizza l'elenco delle sedi disponibili. 3. Seleziona la sede desiderata. 4. Il sistema registra la scelta.
Estensioni	*a. Il paziente può sbagliare l'inserimento di una sede e annullare l'operazione -> la selezione non viene salvata.
Requisiti speciali	Associazione sicura tra paziente e sede.
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	In fase di prima registrazione o cambio sede.

UC3: Prenotazione esame in sede

Nome caso d'uso	UC3: Prenotazione esame in sede
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Paziente
Parti interessate e interessi	<p>Il paziente desidera prenotare un esame in una sede di laboratorio.</p> <p>Il sistema deve garantire la disponibilità e applicare la regola di business R8 (massimo 3 prenotazioni al giorno).</p>
Pre-Condizioni	Il paziente è registrato, autenticato e ha associato almeno una sede.
Garanzia di successo	Il paziente effettua una prenotazione con successo
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il paziente accede alla funzione "Prenotazione Esame". 2. Visualizza gli esami disponibili. 3. Seleziona sede, data e orario. 4. Il sistema verifica disponibilità e rispetto della regola di business massimo 3 prenotazioni. 5. Il paziente conferma. 6. Il sistema registra la prenotazione e genera conferma.
Estensioni	<p>*a. Il paziente sceglie un esame che per quella data o orario è non disponibile-> il sistema avvisa della non disponibilità e suggerisce alternative.</p> <p>*b. il paziente non completa la prenotazione-> il sistema annulla l'operazione dopo un tempo limite.</p> <p>*c. il paziente supera il limite massimo di prenotazioni per giorno -> il sistema impedisce l'operazione.</p>

Requisiti speciali	Applicazione della regola di business R8 gestione prenotazioni riservate tramite Decorator.
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Ogni volta che il paziente desidera effettuare un nuovo esame.

UC4: Gestione sedi (CRUD)

Nome caso d'uso	UC4: Gestione sedi
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Amministratore
Parti interessate e interessi	L'amministratore desidera inserire, modificare o eliminare sedi per la gestione efficiente delle risorse.
Pre-Condizioni	L'amministratore è autenticato nel sistema.
Garanzia di successo	Le informazioni sulle sedi vengono aggiornate correttamente nel sistema.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'amministratore accede alla funzione "Gestione sedi". 2. Seleziona l'azione "Aggiungi sede". 3. Inserisci i dati della sede. 4. Il sistema verifica i dati. 5. L'amministratore conferma. 6. La lista delle sedi viene aggiornata.
Estensioni	<p>*a. I dati inseriti sono incompleti o errati -> il sistema notifica l'errore e richiede la correzione.</p> <p>*b. si tenta di eliminare una sede con prenotazioni attive -> il sistema impedisce l'eliminazione e avvisa l'amministratore.</p>
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Ogni volta che nasce una nuova sede o subisce variazioni.

UC5: Verifica pazienti prenotati

Nome caso d'uso	UC5: Verifica pazienti prenotati
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Personale laboratorio

Parti interessate e interessi	Il personale di laboratorio vuole verificare l'elenco dei pazienti prenotati per un dato esame per organizzare meglio il lavoro.
Pre-Condizioni	Il personale del laboratorio è autenticato nel sistema. Si visualizzano solo gli esami associati alla data attuale.
Garanzia di successo	L'elenco dei pazienti prenotati per un determinato esame viene visualizzato e lo stato della prenotazione viene aggiornato correttamente.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il personale accede alla funzione "Inserisci Stato Prenotazione". 2. Visualizza la lista degli esami prenotati. 3. Seleziona la prenotazione di interesse. 4. Il sistema mostra i dettagli dei pazienti prenotati. 5. Il personale inserisce l'esito della prenotazione (completata o annullata) . 6. Il sistema aggiorna lo stato della prenotazione e genera un referto vuoto.
Estensioni	*a. non ci sono pazienti prenotati per l'esame selezionato: il sistema notifica che non ci sono prenotazioni per l'esame.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Quotidiana.

UC6: Gestione referto

Nome caso d'uso	UC6: Gestione referto
Portata	Applicazione MedLab
Livello	Obiettivo utente
Attore primario	Personale laboratorio
Parti interessate e interessi	Il personale laboratorio vuole inserire, aggiornare o eliminare i referti per garantire l'accuratezza delle informazioni cliniche.
Pre-Condizioni	Il personale del laboratorio è autenticato nel sistema. Il paziente ha completato un esame.
Garanzia di successo	Il referto è correttamente creato, aggiornato o eliminato.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il personale accede alla funzione "Gestione referti". 2. Visualizza i pazienti associati alla sede. 3. Seleziona il paziente. 4. Visualizza le sue prenotazioni confermate. 5. Seleziona una prenotazione. 6. Inserisce un referto.

	7. Il sistema verifica la correttezza dei dati inseriti. 8. Il sistema viene aggiornato.
Estensioni	*a. tentativo di aggiornare un referto già confermato -> il sistema blocca l'operazione.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Ogni volta che viene effettuato un esame.

UC7: Visualizza referto

Nome caso d'uso	UC7: Visualizza referto
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Paziente
Parti interessate e interessi	Paziente: vuole accedere ai propri referti in modo rapido e sicuro.
Pre-Condizioni	Il paziente è autenticato nel sistema e ha esami completati.
Garanzia di successo	Il paziente visualizza correttamente il proprio referto.
Scenario principale di successo	1. Il paziente accede alla funzione "Visualizza referti". 2. Il sistema recupera i dati dei referti e li visualizza.
Estensioni	*a. non ci sono referti disponibili -> il sistema informa il paziente che non ci sono referti disponibili.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Ogni volta che un paziente vuole consultare un referto.

UC8: Modifica Dati Personali

Nome caso d'uso	UC8: Modifica dati personali
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Paziente
Parti interessate e interessi	Il paziente desidera mantenere aggiornate le proprie informazioni per garantire l'efficacia della comunicazione e la precisione dei dati medici.

Pre-Condizioni	Il paziente è autenticato nel sistema.
Garanzia di successo	I dati vengono aggiornati e salvati nel sistema.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il paziente accede alla sezione “Modifica dati personali”. 2. Modifica i dati desiderati. 3. Il sistema valida i dati. 4. I dati vengono aggiornati.
Estensioni	*a. dati mancanti o errati -> il sistema segnala l'errore e chiede correzione.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	Possibilità di aggiornare la password.
Frequenza di ripetizione	Ogni qual volta il paziente ha necessità di aggiornare le proprie informazioni.

UC9: Valutazione Esame

Nome caso d'uso	UC9: Valutazione Esame
Portata	Applicazione MedLab
Livello	Obiettivo utente
Attore primario	Paziente
Parti interessate e interessi	<ul style="list-style-type: none"> • Il Paziente desidera lasciare una valutazione sull'esperienza vissuta durante l'esame, esprimendo il proprio gradimento o meno. • MedLab vuole raccogliere feedback dai pazienti per monitorare la qualità del servizio offerto. • L'Amministratore può utilizzare le recensioni come indicatore della qualità del servizio o per attivare eventuali miglioramenti.
Pre-Condizioni	Il Paziente è autenticato nel sistema MedLab e ha almeno una prenotazione completata associata a un referto.
Garanzia di successo	La recensione viene registrata correttamente e associata all'esame completo.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il paziente accede alla sezione “Inserisci recensioni”. 2. Il sistema mostra solo le sedi dove il paziente ha effettuato esami completati. 3. Il paziente seleziona una sede da recensire. 4. Il paziente inserisce una valutazione numerica (1-5 stelle) e un commento facoltativo. 5. Il sistema registra la recensione e la associa al paziente e alla sede. 6. Il sistema notifica l'amministratore attraverso il Pattern Observer.

Estensioni	<p>*a. Nessuna sede disponibile per la recensione -> il sistema mostra il messaggio “Non hai ancora completato esami in nessuna sede”.</p> <p>*b. Il paziente non inserisce una valutazione valida (manca il punteggio o fuori range) -> il sistema segnala l’errore e chiede di correggere l’input.</p> <p>*c. Errore durante il salvataggio della recensione -> il sistema mostra un messaggio di errore e suggerisce di riprovare.</p>
Requisiti speciali	Validazione del campo punteggio (obbligatorio da 1-5). Notifica automatica agli osservatori tramite Pattern Observer.
Elenco delle varianti tecnologie e dati	Supporto per commento testuale facoltativo. Gestione mappa recensioni associata agli ID.
Frequenza di ripetizione	Ogni volta che un paziente completa un esame, può inserire una recensione per la relativa sede.

UC10: Gestione Esami

Nome caso d’uso	UC10: Gestione Esami (CRUD)
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Amministratore
Parti interessate e interessi	<p>L’amministratore desidera inserire, modificare o rimuovere esami dal sistema per mantenerlo aggiornato.</p> <p>Il paziente desidera visualizzare solo esami correttamente descritti e disponibili.</p>
Pre-Condizioni	L’amministratore è autenticato nel sistema.
Garanzia di successo	L’esame viene creato, aggiornato o eliminato correttamente.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L’amministratore accede all’area “Gestione esami”. 2. Seleziona l’operazione di aggiunta esame. 3. Inserisce i dati richiesti (codice, data, orario, nome). 4. Il sistema verifica i dati ed esegue l’operazione.
Estensioni	*a. dati non validi o incompleti -> il sistema segnala l’errore e richiede correzioni.
Requisiti speciali	Il sistema deve essere in grado di gestire l’associazione degli esami alla sede.
Elenco delle varianti tecnologie e dati	Il sistema deve essere in grado di gestire diverse tipologie di esami ed orari.

Frequenza di ripetizione	Ogni qual volta l'amministratore vuole aggiungere degli esami.
--------------------------	--

UC11: Gestione Personale di Laboratorio

Nome caso d'uso	UC11: Gestione Personale di Laboratorio (CRUD)
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Amministratore
Parti interessate e interessi	L'amministratore vuole gestire in modo flessibile il personale del laboratorio. Il personale del laboratorio desidera che i propri dati siano corretti e aggiornati.
Pre-Condizioni	L'amministratore è autenticato nel sistema.
Garanzia di successo	L'amministratore inserisce i dati relativi al personale di laboratorio in una determinata sede.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'amministratore accede alla sezione "Aggiungi personale laboratorio". 2. Verifica se esiste altro personale di lab associato alla sede. 3. Inserisce la sede lavorativa associata al personale. 4. Compila i dati del personale. 5. Il sistema salva i dati.
Estensioni	*a. Tentativo di inserire un altro personale associato alla stessa sede -> il sistema avvisa e blocca l'operazione.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Al cambio del personale o per aggiornamenti di dati.

UC12: Generazione Report

Nome caso d'uso	UC12: Generazione Report
Portata	Applicazione MedLab
Livello	Obbiettivo utente
Attore primario	Amministratore
Parti interessate e interessi	L'amministratore desidera monitorare statistiche sulle prenotazioni per migliorare l'efficienza.
Pre-Condizioni	L'amministratore è autenticato nel sistema. I dati su prenotazioni sono presenti nel sistema.

Garanzia di successo	Il report viene generato e visualizzato correttamente.
Scenario principale di successo	L'amministratore accede all'area "Generazione Report". Seleziona il tipo di report (mensile, annuale, semestrale). Il sistema analizza i dati. Viene generato un report con dati aggregati.
Estensioni	*a. periodo temporale non valido -> il sistema chiede di reinserire i dati. *b. Nessun dato trovato -> il sistema ritorna statistiche vuote.
Requisiti speciali	
Elenco delle varianti tecnologie e dati	
Frequenza di ripetizione	Su base mensile, annuale e semestrale.

UC13: Visualizza Prenotazioni Attive

Nome caso d'uso	UC13: Visualizza Prenotazioni Attive
Portata	Applicazione MedLab
Livello	Obiettivo utente
Attore primario	Paziente
Parti interessate e interessi	Il Paziente vuole visualizzare l'elenco ordinato delle sue prenotazioni per esami, con data, orario e sede.
Pre-Condizioni	Il Paziente è autenticato nel sistema MedLab
Garanzia di successo	Il sistema mostra correttamente l'elenco delle prenotazioni del paziente con relative informazioni su esame, data, orario e sede.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il paziente seleziona l'opzione "Visualizza prenotazioni". 2. Il sistema verifica che il paziente sia autenticato. 3. Il sistema recupera le prenotazioni associate al paziente. 4. Il sistema ordina le prenotazioni in base agli esami. 5. Il sistema mostra l'elenco con nome esame, data, orario e sede.
Estensioni	<p>*a. Il paziente non è autenticato -> il sistema mostra il messaggio di errore "Errore: nessun paziente attualmente autenticato".</p> <p>*b. il paziente non ha prenotazioni -> il sistema mostra il messaggio "Nessuna prenotazione trovata".</p>

Requisiti speciali	<p>Ordinamento automatico delle prenotazioni tramite comparatore sugli esami.</p> <p>Visualizzazione semplice e leggibile delle informazioni essenziali.</p>
Elenco delle varianti tecnologie e dati	<p>Supporto per più esami, sedi e orari.</p> <p>Utilizzo di una mappa (Map <String,Prenotazione>) per gestire le prenotazioni del paziente.</p>
Frequenza di ripetizione	<p>Ogni volta che il paziente vuole consultare la propria cronologia o situazione delle prenotazioni.</p>

Documento di Visione

MedLab si propone di diventare la soluzione di riferimento per la gestione di un laboratorio di analisi cliniche, offrendo un sistema digitale avanzato. L'obiettivo è fornire una piattaforma completa, intuitiva e altamente efficiente, migliorando l'esperienza sia del personale del laboratorio che dei pazienti.

Per il documento di visione approfondito vedere il documento specifico.

Regole di business

Per il corretto utilizzo del sistema devono essere rispettate le seguenti regole di dominio:

ID	Regola	Modificabilità	Sorgente
R1	Per prenotare un test di laboratorio, il paziente deve essere registrato nel sistema.	Bassa	Politica interna del laboratorio
R2	Il personale del laboratorio può inserire, modificare o eliminare referti solo per i pazienti registrati.	Media	Normative sulla gestione dei dati sanitari
R3	Il paziente può prenotare solo test disponibili nella sede selezionata.	Bassa	Politica interna del laboratorio
R4	I risultati dei test saranno accessibili solo al paziente e al personale autorizzato.	Alta	Regolamenti sulla privacy e GDPR
R5	Il personale può visualizzare solo le prenotazioni relative alla propria sede.	Media	Politica interna del laboratorio
R6	Il paziente può avere più sedi di preferenza in cui effettuare una prenotazione.	Bassa	Politica interna del laboratorio
R7	Il personale di laboratorio può essere associato ad una e una sola sede.	Bassa	Politica interna del laboratorio
R8	Un paziente può avere un massimo di tre prenotazioni attive al giorno.	Media	Politica interna del laboratorio

Specifiche Supplementari

Usabilità

- L'interfaccia deve essere intuitiva e accessibile anche per utenti non esperti, inclusi pazienti di ogni fascia d'età.
- L'interazione con il sistema deve essere semplice e immediata, riducendo al minimo la necessità di formazione per il personale del laboratorio e gli amministratori.

Affidabilità e Performance

- Il software deve essere affidabile e garantire la conservazione dei dati anche in caso di guasti hardware o interruzioni di corrente.
- Il sistema deve garantire un'elevata disponibilità, con tempi di risposta rapidi per la gestione delle prenotazioni, la consultazione dei risultati e le operazioni amministrative.

Vincoli Hardware e Software

- Il software deve essere compatibile con qualsiasi sistema operativo purché sia installata la Java Virtual Machine (JVM).
- Il sistema deve garantire la sicurezza e la protezione dei dati personali e sanitari dei pazienti e del personale.
- Il software deve essere scalabile e modulare, in modo da adattarsi a laboratori di diverse dimensioni e consentire l'aggiunta di nuove funzionalità senza impattare sulle operazioni esistenti.

Glossario

- **MedLab:** Software dedicato alla gestione e all'ottimizzazione delle attività di un laboratorio di analisi cliniche, che semplifica l'amministrazione, la gestione delle prenotazioni e la consultazione dei risultati degli esami.
- **Amministratore di Sistema:** Responsabile della gestione delle attività del laboratorio, inclusa la registrazione di nuovi pazienti, la gestione del personale e delle sedi, la configurazione degli esami disponibili e la generazione di report.
- **Paziente:** Utente registrato su MedLab che può prenotare test di laboratorio, visualizzare i risultati delle analisi, modificare il proprio profilo e lasciare recensioni sui servizi ricevuti.
- **Personale del Laboratorio:** Professionisti sanitari che utilizzano MedLab per inserire e gestire i referti dei pazienti, verificare le prenotazioni e organizzare gli esami del laboratorio.
- **Esame di Laboratorio:** Esame clinico eseguibile presso una delle sedi del laboratorio. Ogni esame ha specifici requisiti e può essere prenotato attraverso MedLab.
- **Referto:** Documento digitale contenente i risultati di un esame di laboratorio. I referti vengono caricati dal personale del laboratorio e sono consultabili dai pazienti attraverso MedLab.
- **Prenotazione:** Procedura attraverso la quale un paziente sceglie una sede, una data e un esame da eseguire. La prenotazione è necessaria per accedere ai servizi del laboratorio.
- **Sede del Laboratorio:** Struttura fisica in cui vengono eseguiti gli esami di laboratorio. MedLab permette di gestire più sedi e assegnare esami specifici a ciascuna di esse.
- **Recensione:** Valutazione che il paziente può lasciare dopo aver effettuato un esame, contribuendo a migliorare la qualità del servizio del laboratorio.
- **Report Statistici:** Analisi generate dall'amministratore per monitorare le prenotazioni e l'efficienza operativa del laboratorio.
- **Team di Sviluppo e IT:** Responsabili della progettazione, implementazione e manutenzione del software MedLab, garantendo la sicurezza e la scalabilità del sistema.

Analisi orientata agli oggetti

Introduzione

L'implementazione dell'applicazione MedLab ha seguito l'approccio iterativo ed evolutivo proposto dal Unified Process (UP), articolandosi in quattro iterazioni.

Questo approccio ha consentito di sviluppare progressivamente il nucleo architetturale del sistema, affrontando di volta in volta i rischi più significativi.

Inoltre, è stata condotta un'analisi graduale dei requisiti, minimizzando l'impatto di eventuali errori di progettazione e implementazione.

Iterazione 1:

implementazione dei casi d'uso UC1: "Gestione pazienti", UC2: "Registrazione sede", UC3: "Prenotazione esame" e UC4: "Gestione Sedi".

In questa fase si è costruita l'infrastruttura di base, includendo la registrazione di pazienti, sedi e la possibilità per i pazienti di prenotare esami in laboratorio.

Iterazione 2:

implementazione dei casi d'uso UC5: "Verifica pazienti prenotati", UC6: "Gestione referto", UC7: "Visualizza referto".

In questa iterazione sono stati introdotti i Pattern GoF:

- State: per la gestione degli stati delle prenotazioni (inAttesa, Completato, Annullato), migliorando la modularità.
- Proxy: per controllare l'accesso alle funzionalità riservate solo al personale autorizzato (es. gestione referti).

L'integrazione di questi casi d'uso ha completato il flusso di gestione degli esami, ottimizzando la comunicazione tra personale di laboratorio e pazienti, e migliorando l'efficienza operativa.

Iterazione 3:

Implementazione dei casi d'uso UC8: "Modifica dati personali", UC10: "Gestione esami", UC11: "Gestione personale di laboratorio", UC12: "Generazione report".

Introduzione del Pattern GoF:

- Factory Method: per la creazione organizzata e scalabile di diversi tipi di report (mensile, semestrale, annuale), rendendo la generazione dei documenti flessibile ed estendibile.

Questa iterazione ha potenziato le funzionalità amministrative del sistema, consentendo una gestione completa di esami, personale e produzione di statistiche di sistema, migliorando la supervisione da parte dell'amministratore.

Iterazione 4:

durante la quarta ed ultima iterazione, il lavoro è focalizzato sull'implementazione dello scenario alternativo UC3 con applicazione della Regola di Business R8: Limitazione a un massimo di tre prenotazioni giornaliere per paziente. Implementati anche i casi d'uso UC9: "Valutazione esame" e UC13: "Visualizza prenotazioni attive del paziente".

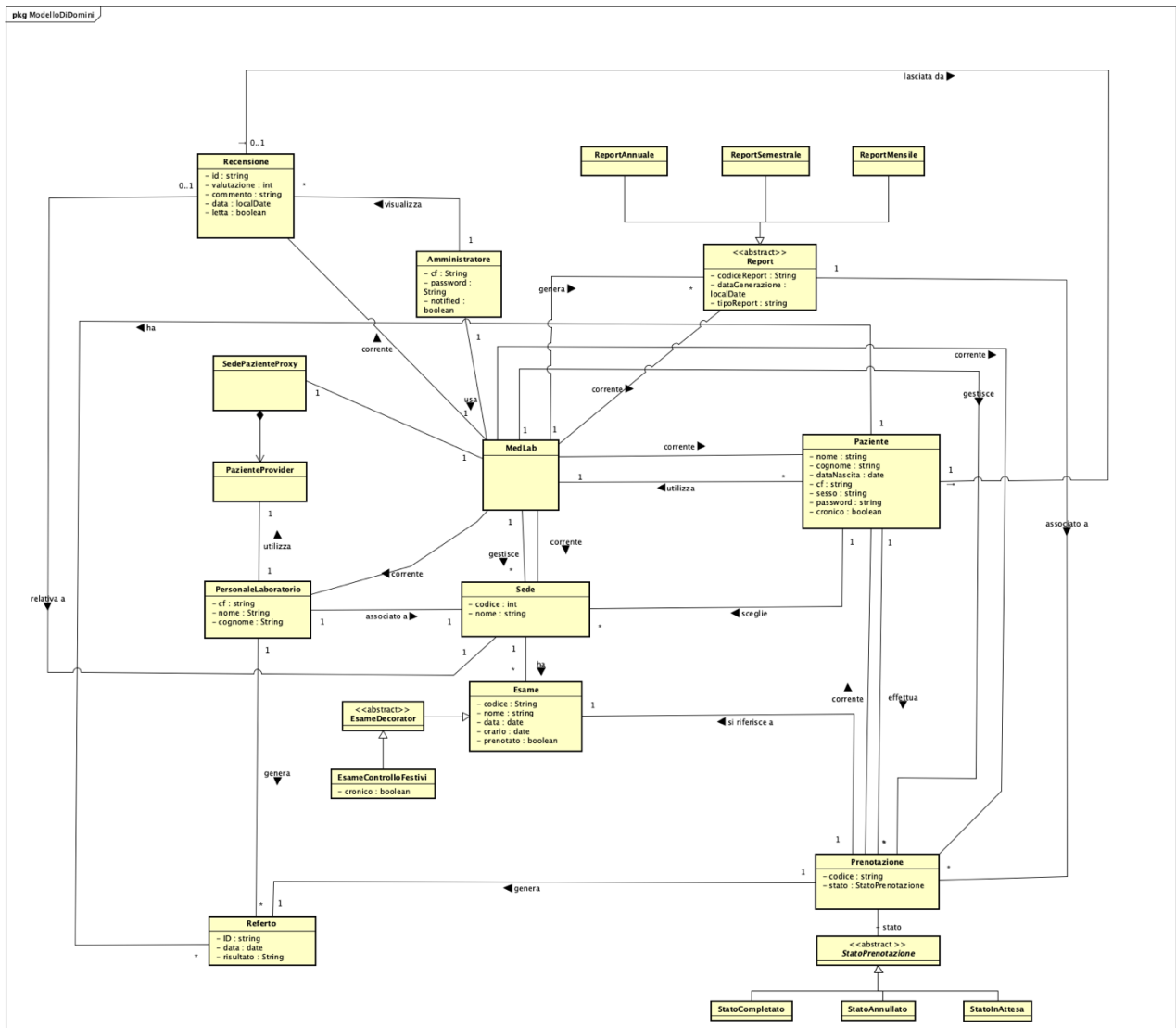
Introduzione di Pattern GoF:

- Decorator: per estendere le funzionalità della prenotazione (es. gestione priorità per pazienti con malattie croniche) senza alterare la struttura delle classi esistenti.

Questa iterazione ha reso il sistema più estendibile, reattivo e modulare, arricchendo l'esperienza utente e consolidando l'architettura di MedLab.

Modello di Dominio

Alla fine delle quattro iterazioni, il modello di dominio di MedLab si presenta completo e consolidato, e si configura come segue:



- **MedLab:** sistema principale che gestisce le funzionalità rappresenta l'applicazione o piattaforma stessa.
- **Amministratore:** utente con privilegi di gestione globale. Gestisce pazienti, sedi, personale e report.
- **Paziente:** prenota esami, modifica i propri dati, inserisce recensioni.
- **Sede:** struttura fisica in cui vengono effettuati gli esami.
- **Esame:** test diagnostico disponibile per la prenotazione.
- **Prenotazione:** rappresenta l'atto di prenotazione di un esame.
- **StatoPrenotazione:** classe astratta che rappresenta lo stato dinamico di una prenotazione.
- **Stato InAttesa:** implementazione concreta di StatoPrenotazione, gestisce le transizioni di stato.
- **Stato Completato:** implementazione concreta di StatoPrenotazione, gestisce le transizioni di stato.

- **Stato Annullato:** implementazione concreta di StatoPrenotazione, gestisce le transizioni di stato.
- **Referto:** documento clinico con i risultati di un esame.
- **Recensione:** feedback rilasciato dal paziente dopo l'esame.
- **Report:** documento aggregato che analizza statisticamente le prenotazioni.
- **Report Mensile:** rappresenta una sottoclasse di Report specificamente per la durata mensile.
- **Report Annuale:** rappresenta una sottoclasse di Report specificamente per la durata annuale.
- **Report Semestrale:** rappresenta una sottoclasse di Report specificamente per la durata semestrale.

SSD e Contratti

Nel contesto dell'analisi orientata agli oggetti, la fase successiva all'analisi dei casi d'uso è stata la creazione dei Diagrammi di Sequenza di Sistema (SSD), per rappresentare il flusso di eventi di input e output relativi a ciascun caso d'uso implementato durante ogni iterazione.

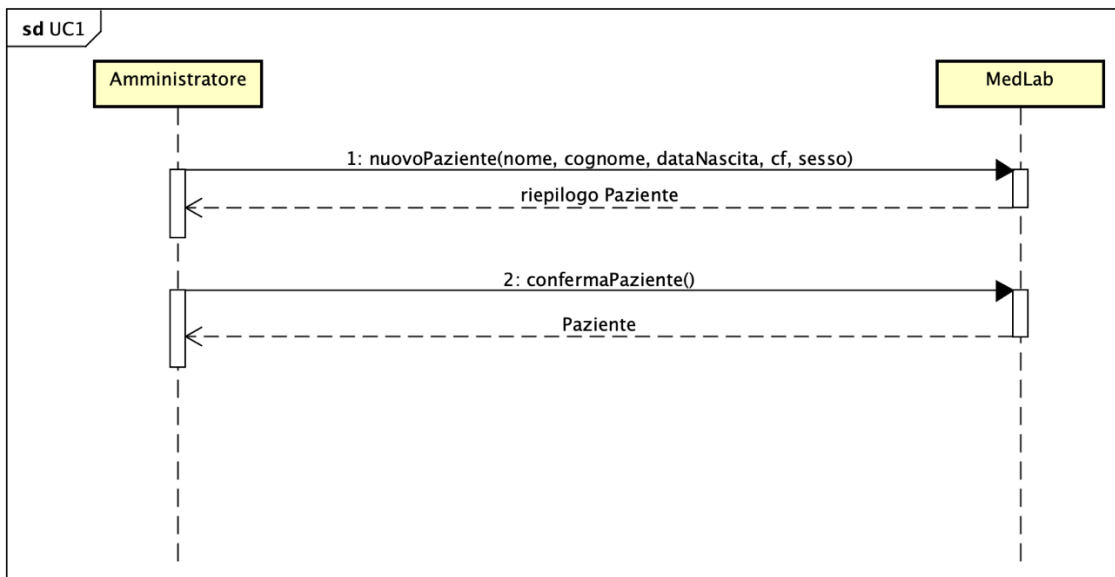
Questi SSD mostrano come il sistema MedLab risponde alle interazioni degli utenti (Pazienti, Personale di laboratorio e Amministratore).

Le operazioni rilevanti individuate nei diagrammi SSD sono state poi descritte mediante i Contratti delle Operazioni, che specificano le pre-condizioni, post-condizioni e gli effetti dell'operazione sul modello di dominio.

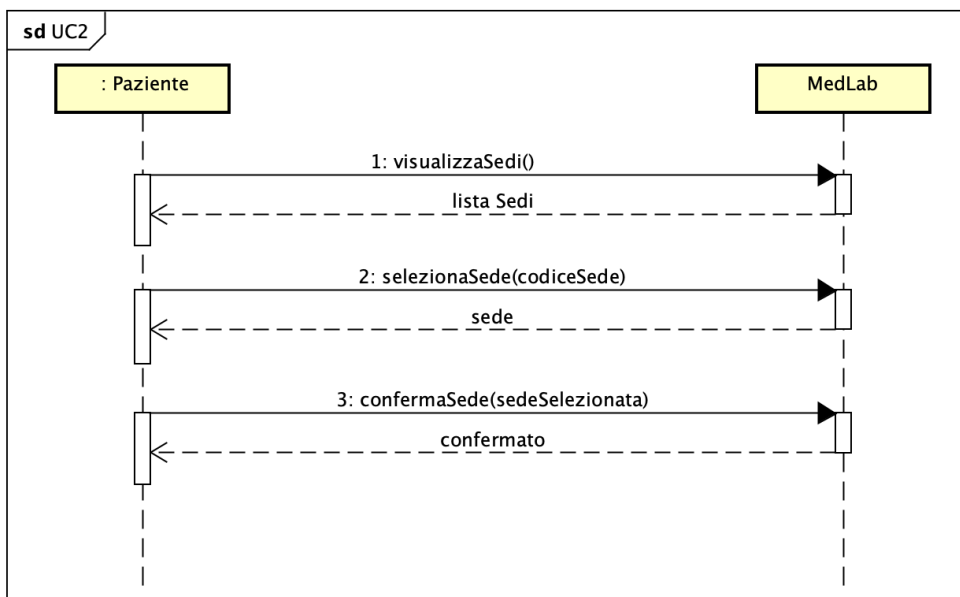
Questo approccio garantisce una visione chiara e dettagliata delle responsabilità e dei flussi ed una solida base per le fasi successive di progettazione e implementazione.

Iterazione 1:

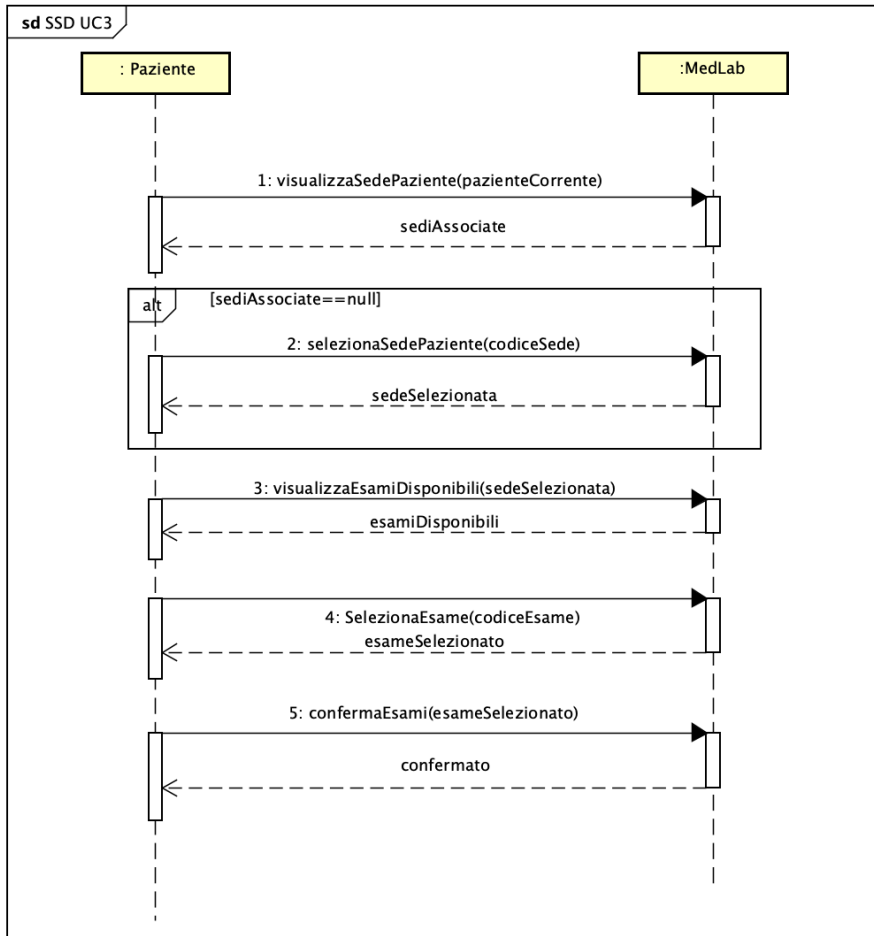
SSD UC1



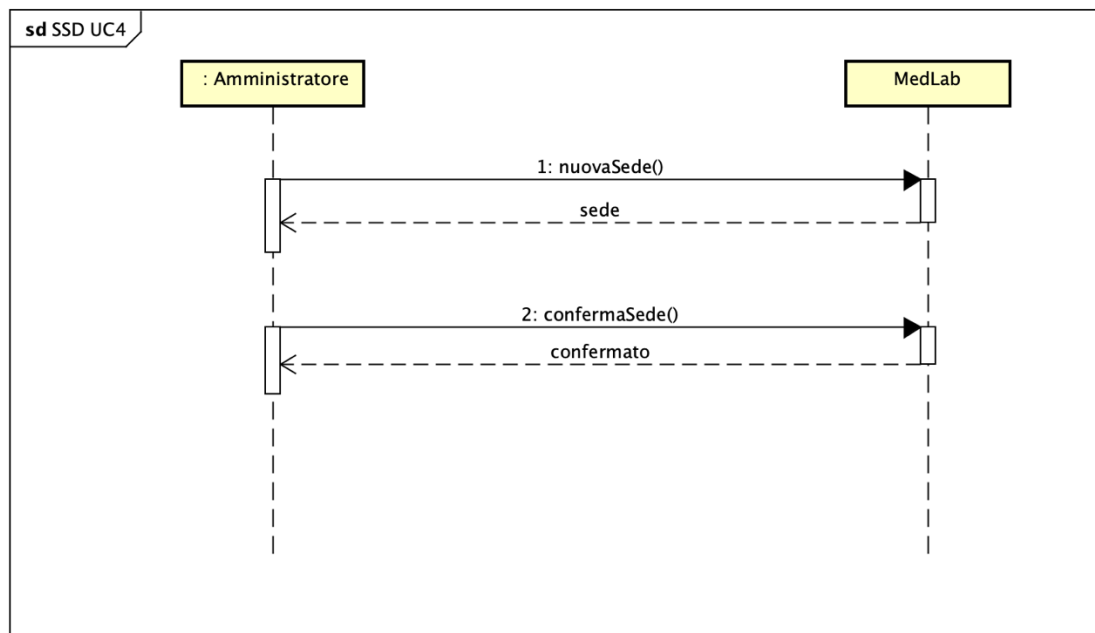
SSD UC2



SSD UC3



SSD UC4



Contratti Operazioni UC1

Contratto CO1: nuovoPaziente

Operazione	nuovoPaziente(nome: String, cognome: String, dataNascita: Date, cf: String, sesso: String)
Riferimenti	Caso d'Uso UC1: Gestione Pazienti
Pre-Condizioni	L'amministratore deve essere autenticato nel sistema Medlab. Non deve esistere già un paziente con lo stesso codice fiscale (cf)
Post-Condizioni	<ul style="list-style-type: none">È stata creata un'istanza di paziente di Paziente paziente è stato inizializzato con le informazioni inseritepaziente è stato associato a Medlab tramite l'associazione "corrente".

Contratto CO2: confermaPaziente

Operazione	confermaPaziente()
Riferimenti	Caso d'Uso UC1: Gestione Pazienti
Pre-Condizioni	È in corso la registrazione di un nuovo Paziente (esiste un'istanza pazienteCorrente)
Post-Condizioni	<ul style="list-style-type: none">L'istanza pazienteCorrente è stata aggiunta alla lista dei pazienti di MedLabÈ stata eliminata l'associazione "corrente" tra MedLab e l'istanza pazienteCorrente.Viene visualizzato un riepilogo delle informazioni inserite per il paziente.

Contratti Operazione UC2

Contratto CO1: selezionaSede

Operazione	selezionaSede(codSede: integer)
Riferimenti	Caso d'Uso UC2: Registrazione Sede Lab.
Pre-Condizioni	Esiste una lista di sedi disponibili nel sistema MedLab. La sede con il codSede specificato deve esistere nella lista delle sedi.
Post-Condizioni	<ul style="list-style-type: none">Viene restituita l'istanza sede di Sede corrispondente al codSede.Se il codSede non è valido, viene restituito null.

Contratto CO2: confermaSede

Operazione	confermaSede (sedeSelezionata: Sede)
Riferimenti	Caso d'Uso UC2: Registrazione Sede Lab.
Pre-Condizioni	È stato selezionato un paziente (pazienteCorrente non è null). È stata selezionata una sede valida (sedeSelezionata non è null)
Post-Condizioni	<ul style="list-style-type: none">L'istanza sedeSelezionata è stata associata al pazienteCorrente.Viene visualizzato un messaggio di conferma con i dettagli della sede e del paziente.

Contratti Operazioni UC3

Contratto CO1: selezionaSede

Operazione	selezionaSedePaziente(codSede: integer)
Riferimenti	Caso d'Uso UC2: Registrazione Sede Lab.
Pre-Condizioni	Esiste una lista di sedi disponibili nel sistema MedLab. La sede con il codSede specificato deve esistere nella lista delle sedi.
Post-Condizioni	<ul style="list-style-type: none">Viene restituita l'istanza sede di Sede corrispondente al codSede.Se il codSede non è valido, viene restituito null.

Contratto CO2: selezionaEsame

Operazione	selezionaEsame (sedeSelezionata:Sede, codiceEsame: String)
Riferimenti	Caso d'Uso UC3: Prenotazione esame in sede.
Pre-Condizioni	È stata selezionata una sede valida Esiste una lista di esami disponibili presso la sede.
Post-Condizioni	<ul style="list-style-type: none">Verifica che l'esame selezionato è ancora disponibile.Se è disponibile crea un'istanza di Prenotazione associata al paziente e all'esame.

Contratto CO3: confermaEsame

Operazione	confermaEsame()
Riferimenti	Caso d'Uso UC3: Prenotazione esame in sede.
Pre-Condizioni	È stata creata un'istanza di Prenotazione (Prenotazione non è null)
Post-Condizioni	<ul style="list-style-type: none">L'istanza di prenotazione corrente viene aggiunta alla mappa di Prenotazioni presenti nel sistema.L'attributo dell'istanza esameSelezionato è stata contrassegnata come prenotata.Viene visualizzato un messaggio di conferma con i dettagli della prenotazione.

Contratti Operazioni UC4

Contratto CO1: nuovaSede

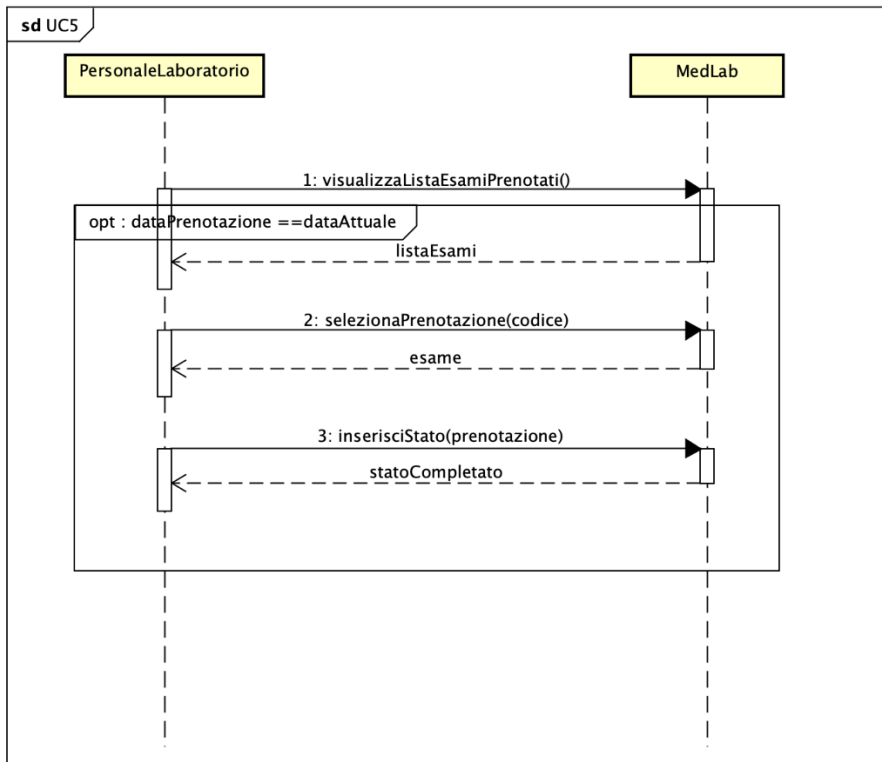
Operazione	nuovaSede (nome:String , codice:String)
Riferimenti	Caso d'Uso UC4: Gestione Sedi
Pre-Condizioni	L'amministratore è autenticato nel sistema MedLab. Non deve esistere già una sede con lo stesso codice.
Post-Condizioni	<ul style="list-style-type: none">È stata creata un'istanza di Sede con i parametri nome e codice.L'istanza sede è stata associata a MedLab tramite l'associazione 'corrente'.

Contratto CO2: confermaSede

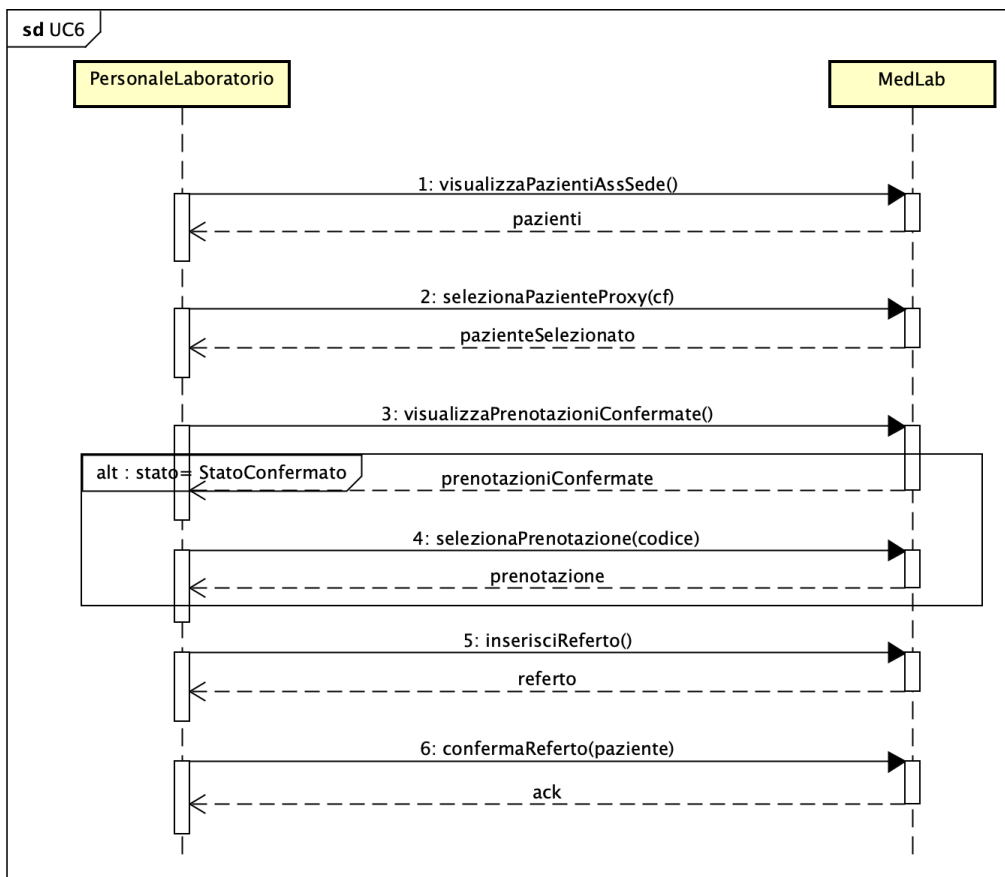
Operazione	confermaSede()
Riferimenti	Caso d'Uso UC4: Gestione Sedi
Pre-Condizioni	È in corso la registrazione di una nuova Sede (esiste un'istanza sedeCorrente).
Post-Condizioni	<ul style="list-style-type: none"> • L'istanza sedeCorrente è stata aggiunta alla lista delle sedi di MedLab. • È stata eliminata l'associazione 'corrente' tra MedLab e l'istanza sedeCorrente. • Viene visualizzato un riepilogo delle informazioni inserite per sede.

Iterazione 2:

SSD UC5



SSD UC6



Contratti Operazioni UC5

Contratto CO1: selezionaPrenotazione

Operazione	selezionaPrenotazione(codice)
Riferimenti	Caso d'Uso UC5: Verifica pazienti prenotati
Pre-Condizioni	<ul style="list-style-type: none">Il personale di laboratorio deve essere autenticato nel sistema Medlab.Esiste almeno un paziente associato alla sede del personale.Il parametro codice corrisponde al codice di una prenotazione.
Post-Condizioni	<ul style="list-style-type: none">Se il codice è valido, viene restituita l'istanza della prenotazione associata.Se il codice non è valido o la prenotazione non è trovata, viene restituito null.

Contratto CO2: inserisciStato

Operazione	inserisciStato(prenotazione)
Riferimenti	Caso d'Uso UC5: Verifica pazienti prenotati
Pre-Condizioni	<ul style="list-style-type: none">Il personale di laboratorio deve essere autenticato in MedLab.La prenotazione deve esistere nel sistema.Lo stato attuale della prenotazione deve essere "in attesa".
Post-Condizioni	<ul style="list-style-type: none">Lo stato della prenotazione viene aggiornato in base all'azione eseguita.Se lo stato è "completato" -> prenotazione avvenuta con successo, l'esame è stato effettuato e si può generare il referto.Se lo stato è "annullato" -> la prenotazione non è più valida.

Contratti Operazioni UC6

Contratto CO1: selezionaPazienteProxy

Operazione	selezionaPazienteProxy(cf)
Riferimenti	Caso d'Uso UC6: Gestione Referto
Pre-Condizioni	<ul style="list-style-type: none">Il personale di laboratorio deve essere autenticato nel sistema Medlab.Il paziente deve essere associato alla sede del personale di laboratorio.
Post-Condizioni	<ul style="list-style-type: none">Viene restituita l'istanza di paziente associato alla sede.Se il paziente non è trovato o non è associato alla sede, viene restituito null.

Contratto CO2: selezionaPrenotazione

Operazione	selezionaPrenotazione(codice)
Riferimenti	Caso d'Uso UC6: Gestione Referto

Pre-Condizioni	<ul style="list-style-type: none"> • Il personale di laboratorio deve essere autenticato nel sistema Medlab. • Il paziente deve essere associato alla sede del personale di laboratorio. • Il sistema mostra solo le prenotazioni con stato completato.
Post-Condizioni	<ul style="list-style-type: none"> • Se la prenotazione è trovata ed è in stato “completato”, viene restituita l’istanza. • Se il codice non è valido o la prenotazione non esiste, viene restituito null.

Contratto CO3: inserisciReferto

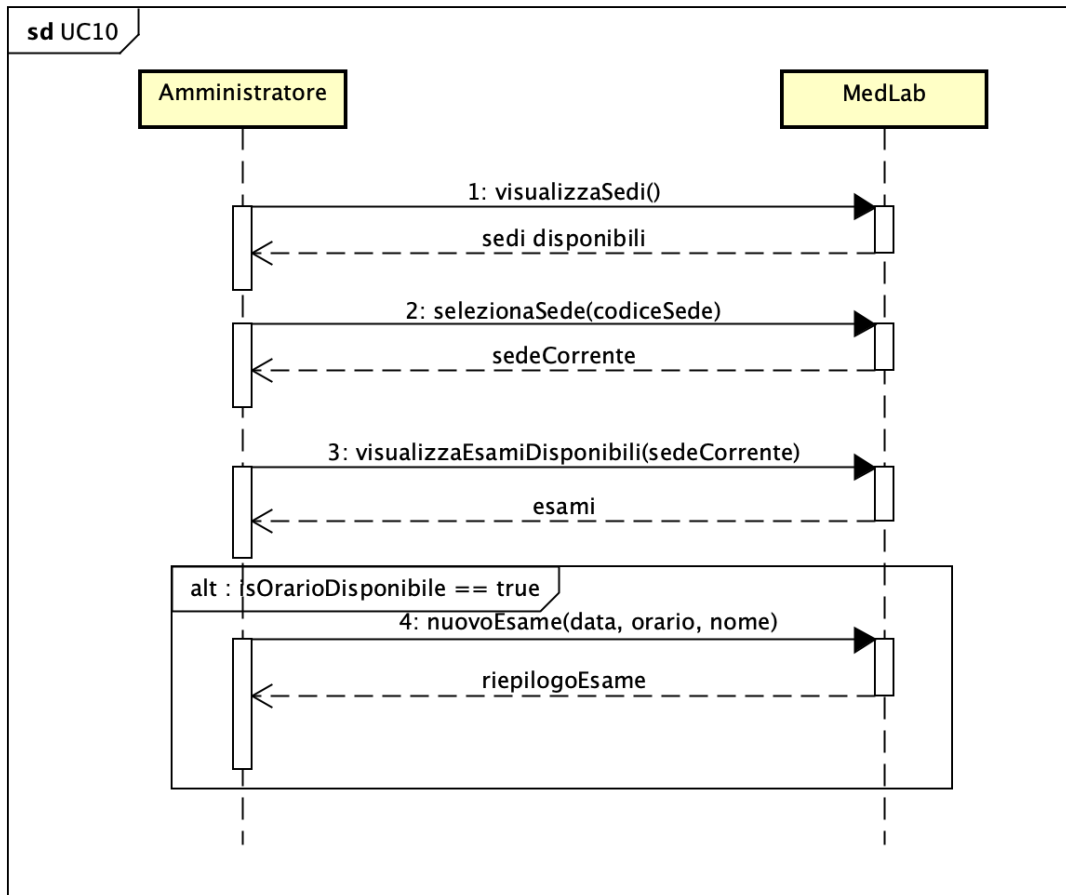
Operazione	InserisciReferto()
Riferimenti	Caso d’Uso UC6: Gestione Referto
Pre-Condizioni	<ul style="list-style-type: none"> • Il personale di laboratorio deve essere autenticato nel sistema MedLab. • Il paziente deve essere associato alla sede di laboratorio. • La prenotazione del paziente deve essere nello stato “completato”, e quindi ha già effettuato un esame.
Post-Condizioni	<ul style="list-style-type: none"> • Viene aggiornata la descrizione relativa al referto selezionato.

Contratto CO4: confermaReferto

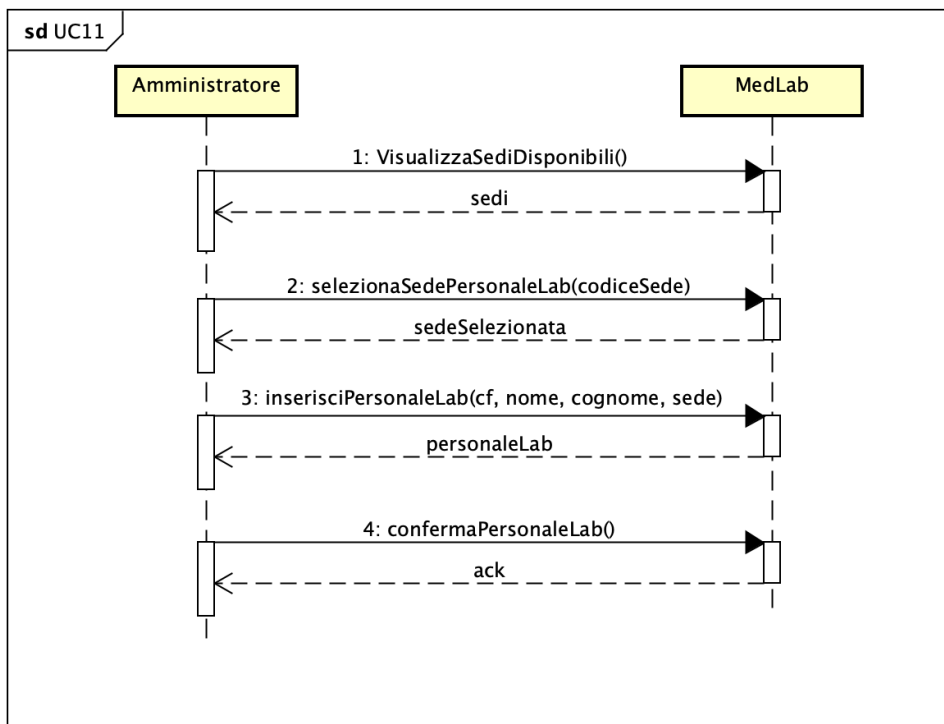
Operazione	confermaReferto(paziente)
Riferimenti	Caso d’Uso UC6: Gestione Referto
Pre-Condizioni	<ul style="list-style-type: none"> • Esiste un’istanza valida di refertoCorrente. • Il paziente è valido.
Post-Condizioni	<ul style="list-style-type: none"> • Il referto viene associato al paziente. • Viene salvato nella mappa dei referti correnti del paziente. • Il refertoCorrente viene azzerato.

Iterazione 3:

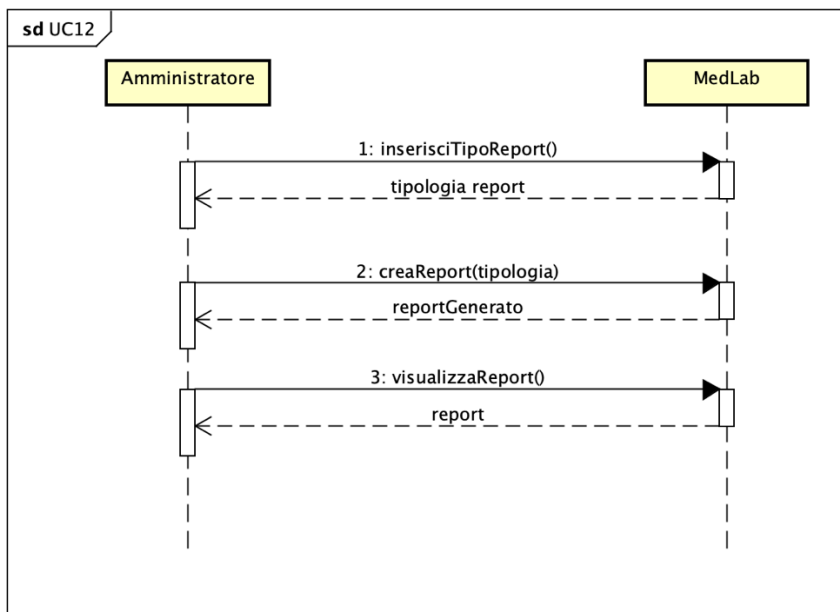
SSD UC10



SSD UC11



SSD UC12



Contratti Operazioni UC10

Contratto CO1: selezionaSede

Operazione	selezionaSede(codiceSede)
Riferimenti	Caso d'Uso UC10: Gestione Esami
Pre-Condizioni	<ul style="list-style-type: none"> Esistono sedi nel sistema.
Post-Condizioni	<ul style="list-style-type: none"> Se il codice è valido, restituisce la sede selezionata, altrimenti restituisce null.

Contratto CO2: nuovoEsame

Operazione	nuovoEsame (data, orario, nome)
Riferimenti	Caso d'Uso UC10: Gestione Esami
Pre-Condizioni	<ul style="list-style-type: none"> sedeCorrente è stata selezionata.
Post-Condizioni	<ul style="list-style-type: none"> Viene aggiunto un nuovo esame alla sede selezionata. sedeCorrente viene poi resettata a null.

Contratti Operazioni UC11

Contratto CO1: selezionaSedePersonaleLab

Operazione	selezionaSedePersonaleLab(codiceSede)
Riferimenti	Caso d'Uso UC11: Gestione Personale Laboratorio
Pre-Condizioni	<ul style="list-style-type: none"> La sede esiste nel sistema. Non deve esserci già personale assegnato a quella sede.
Post-Condizioni	<ul style="list-style-type: none"> Se la sede è valida e libera, viene restituita. Se occupata o non trovata, restituisce null.

Contratto CO2: inserisciPersonaleLab

Operazione	inserisciPersonaleLab (cf, nome, cognome, sede)
Riferimenti	Caso d'Uso UC11: Gestione Personale Laboratorio

Pre-Condizioni	<ul style="list-style-type: none"> I dati del nuovo personale sono validi.
Post-Condizioni	<ul style="list-style-type: none"> Viene creata un'istanza <code>PersonaleLaboratorio</code> e salvata come <code>personaleLaboratorioCorrente</code>.

Contratto CO3: confermaPersonaleLab

Operazione	confermaPersonaleLab
Riferimenti	Caso d'Uso UC11: Gestione Personale Laboratorio
Pre-Condizioni	<ul style="list-style-type: none"> Esiste un <code>personaleLaboratorioCorrente</code> da confermare.
Post-Condizioni	<ul style="list-style-type: none"> Il personale viene inserito nella mappa <code>personaleLaboratori</code>. <code>personaleLaboratorioCorrente</code> viene messo a null.

Contratti Operazioni UC12

Contratto CO1: inserisciTipoReport

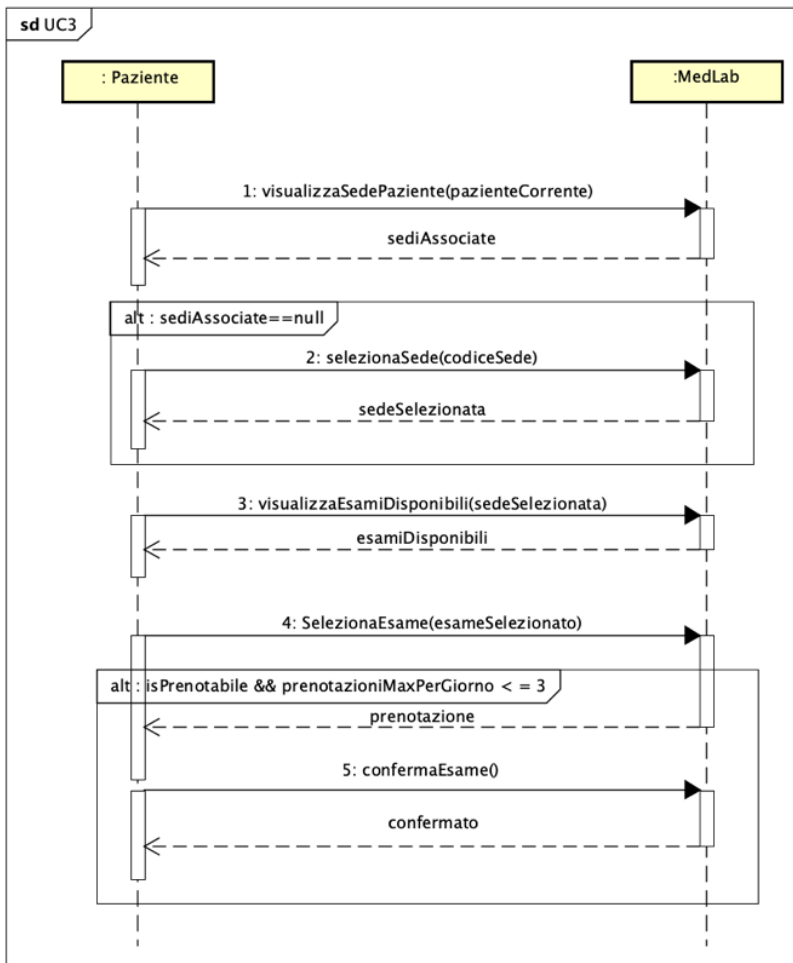
Operazione	inserisciTipoReport
Riferimenti	Caso d'Uso UC12: Generazione Report
Pre-Condizioni	<ul style="list-style-type: none"> L'amministratore è autenticato.
Post-Condizioni	<ul style="list-style-type: none"> Viene restituito il tipo di report selezionato (mensile, semestrale, annuale). Se inserito tipo invalido ritorna null.

Contratto CO2: creaReport

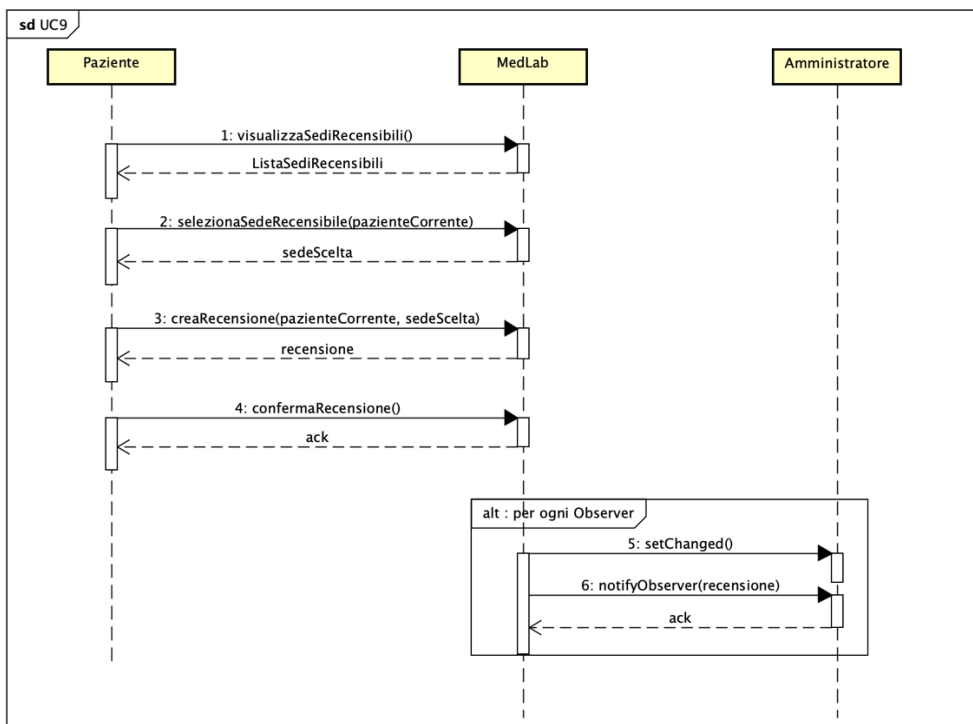
Operazione	creaReport (tipologia)
Riferimenti	Caso d'Uso UC12: Generazione Report
Pre-Condizioni	<ul style="list-style-type: none"> Il tipo di report è valido.
Post-Condizioni	<ul style="list-style-type: none"> Viene generato e salvato un report nella variabile <code>reportCorrente</code>.

Iterazione 4:

SSD UC3



SSD UC9



Contratti Operazioni UC9

Contratto CO1: visualizzaSediRecensibili

Operazione	visualizzaSediRecensibili
Riferimenti	Caso d'Uso UC9: Valutazione Esame
Pre-Condizioni	<ul style="list-style-type: none">Il paziente è autenticato nel sistema.Il paziente ha almeno una prenotazione con stato "Completato".
Post-Condizioni	<ul style="list-style-type: none">Viene restituita una lista di sedi in cui il paziente ha effettuato esami completati.Se non ci sono sedi valide, viene restituita una lista vuota.

Contratto CO2: selezionaSedeRecensibile

Operazione	selezionaSedeRecensibile (paziente)
Riferimenti	Caso d'Uso UC9: Valutazione Esame
Pre-Condizioni	<ul style="list-style-type: none">Il paziente è autenticato nel sistema.Esiste almeno una sede recensibile restituita da VisualizzaSediRecensibili().
Post-Condizioni	<ul style="list-style-type: none">Viene restituita l'istanza della sede selezionata dal paziente.Se la selezione è errata o annullata, viene restituito null.

Contratto CO3: creaRecensione

Operazione	creaRecensione(paziente, sedeSelezionata)
Riferimenti	Caso d'Uso UC9: Valutazione Esame
Pre-Condizioni	<ul style="list-style-type: none">Il paziente e la sede sono validi e non null.Il paziente ha effettuato un esame nella sede indicata.
Post-Condizioni	<ul style="list-style-type: none">Viene creata una nuova istanza Recensione contenente: paziente, sede, stelle, commento.La recensione è pronta per esser salvata.

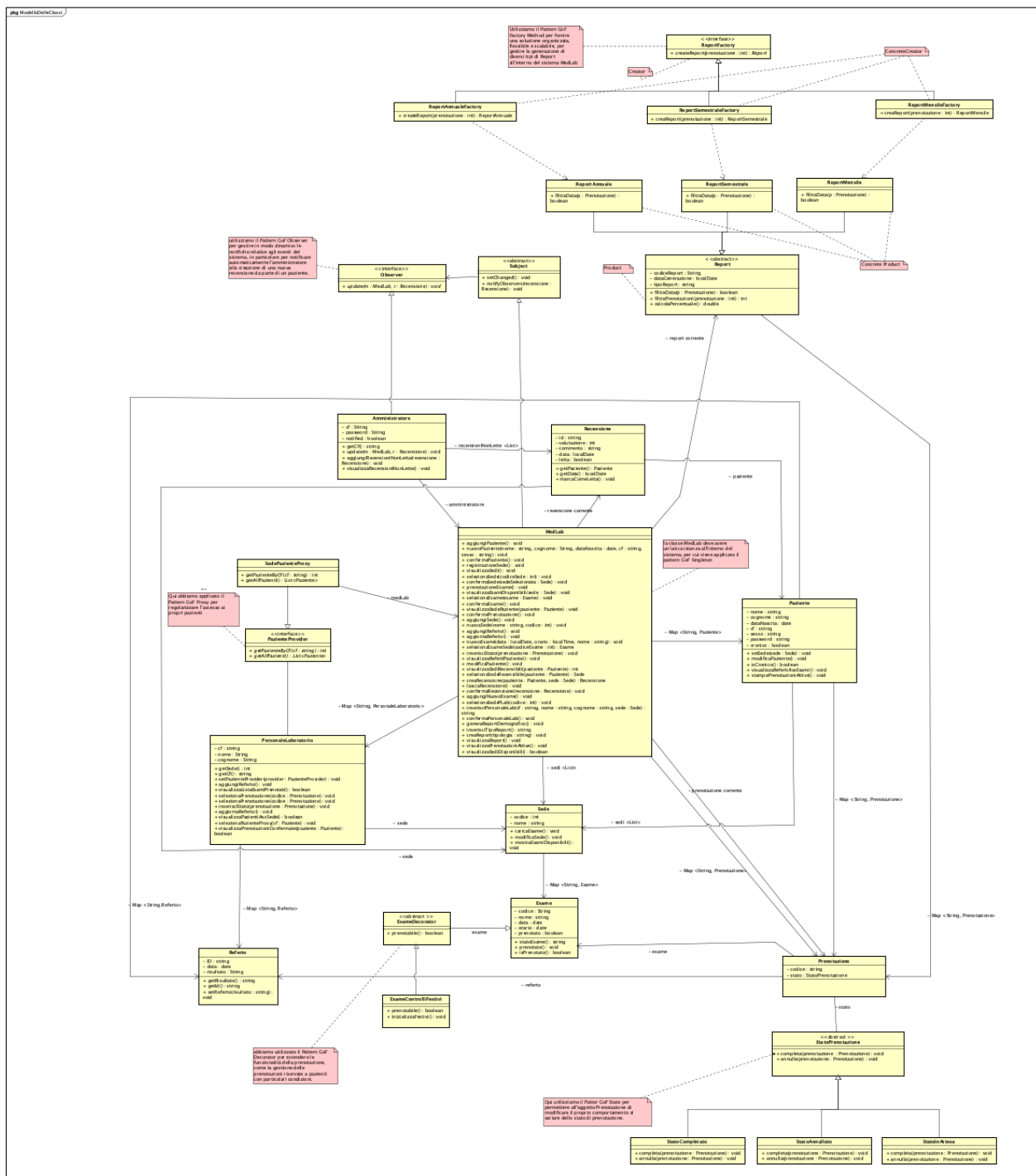
Contratto CO4: confermaRecensione

Operazione	confermaRecensione
Riferimenti	Caso d'Uso UC9: Valutazione Esame
Pre-Condizioni	<ul style="list-style-type: none">È stata creata un'istanza valida di Recensione.
Post-Condizioni	<ul style="list-style-type: none">La recensione viene salvata nel sistema.L'amministratore registrato viene notificato con notifyObserver.Il paziente riceve un messaggio di conferma dell'invio.

Progettazione

La fase di progettazione dell'applicazione MedLab è stata affrontata in modo strutturato e progressivo, tenendo conto sia degli aspetti dinamici (modellati tramite diagrammi di interazione) che di quelli statici (descritti con il diagramma delle classi).

Diagramma delle classi

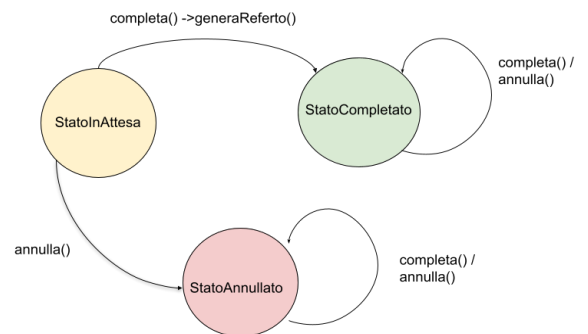


Un ruolo centrale nella progettazione è stato ricoperto dall'utilizzo di alcuni Pattern GoF, scelti per migliorare la flessibilità, la manutenibilità e la coesione del sistema. Ogni pattern è stato selezionato per rispondere a precise esigenze architetturelle emerse nel corso delle iterazioni. Di seguito vengono illustrati i principali pattern adottati e le motivazioni della loro integrazione.

Abbiamo deciso di utilizzare il **Pattern State** per la gestione dello stato delle prenotazioni.

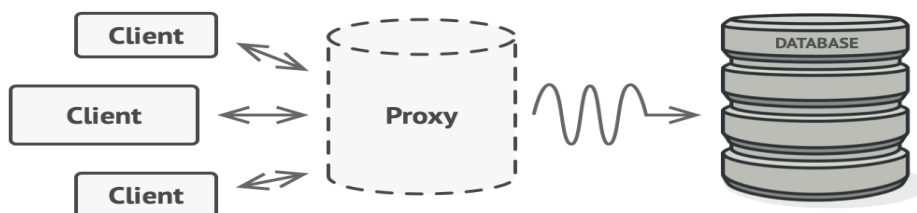
Questa scelta architetturale è motivata dai seguenti vantaggi:

- Consente all'oggetto Prenotazione di modificare dinamicamente il proprio comportamento in base allo stato corrente (StatoInAttesa, StatoCompletato, StatoAnnullato), evitando l'uso di strutture condizionali.
- Ogni stato implementa comportamenti specifici, rendendo il codice più leggibile, modulare e facilmente estendibile.
- Le transizioni di stato sono gestite internamente: ad esempio, la chiamata a completa() nello stato InAttesa imposta automaticamente lo stato a Completato e genera il referto associato.



Per il controllo degli accessi, abbiamo implementato il **Pattern Proxy**, in particolare per proteggere le funzionalità riservate al personale di laboratorio. I vantaggi ottenuti includono:

- Controllo centralizzato degli accessi, con blocco automatico delle operazioni se l'utente non è autorizzato.
- La sicurezza dei dati sensibili, grazie all'intermediazione del proxy che impedisce l'accesso a informazioni cliniche da parte di utenti non autorizzati.
- La separazione dei ruoli, garantendo che ogni utente possa accedere solo alle funzionalità a lui destinate.



Per la generazione dei report, abbiamo impiegato il **Pattern Factory Method**. Questa soluzione si è rivelata efficace per:

- Creare in modo dinamico report mensili, semestrali e annuali, grazie a factory dedicate (ReportMensileFactory, ReportAnnualeFactory, ReportSemestraleFactory).
- Garantire una facilità di estensione, così i nuovi tipi di report possono essere aggiunti creando nuove factory senza alterare quelle esistenti.
- Gestire eventuali complessità future nella generazione dei dati in modo modulare e isolato, mantenendo l'integrità del sistema.

Il **Pattern Decorator** è stato utilizzato nei casi d'uso UC3: "Prenotazione Esame" e UC10: "Gestione Esami" per estendere le funzionalità delle prenotazioni. I motivi della scelta includono:

- Estendibilità senza modificare le classi esistenti (Esame, Prenotazione), facilitando l'aggiunta di logiche come il controllo sui pazienti cronici o il rispetto della regola R8 (massimo tre prenotazioni giornaliere).
- Possibilità di combinare diversi decorator (es. per gestire i festivi e priorità) senza introdurre nuove sottoclassi.
- Ogni Decorator è responsabile di una singola funzionalità, garantendo alta coesione e semplicità di manutenzione.

Per la gestione della notifica automatica di eventi, come l'invio di una recensione da parte del paziente, abbiamo integrato il **Pattern Observer**. Questa scelta ha permesso di:

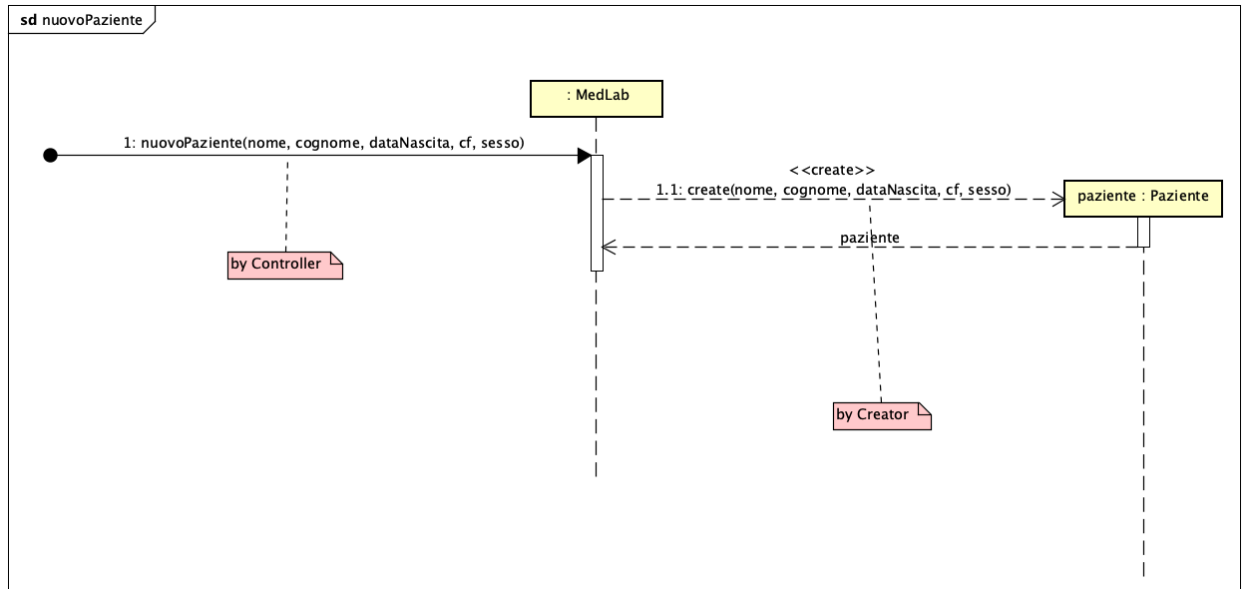
- Separare la logica di business dalla notifica, chi crea la recensione non ha bisogno di sapere chi deve essere avvisato.
- Inviare automaticamente aggiornamenti all'amministratore attraverso `notifyObserver(recensione)` ogni volta che una recensione viene inserita.
- Estendere facilmente il sistema in futuro per includere altri canali o destinatari delle notifiche.

Diagrammi di sequenza

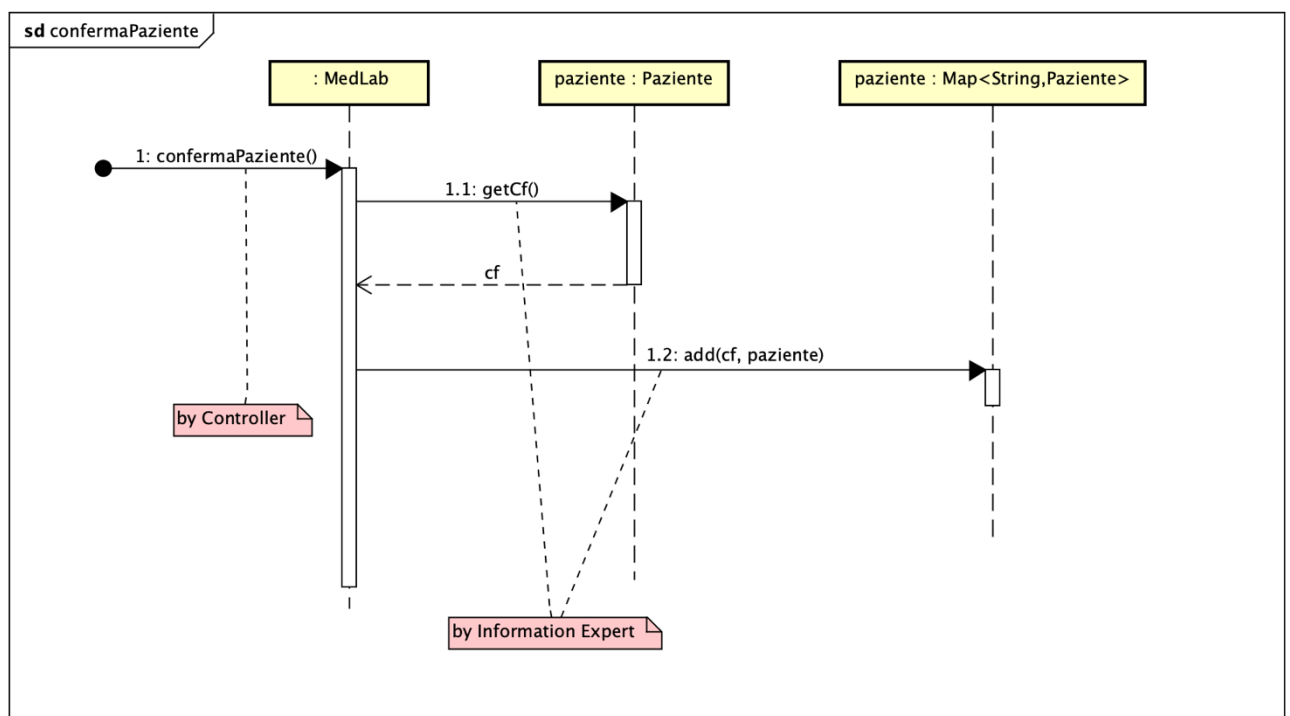
Iterazione 1:

UC1

- nuovoPaziente

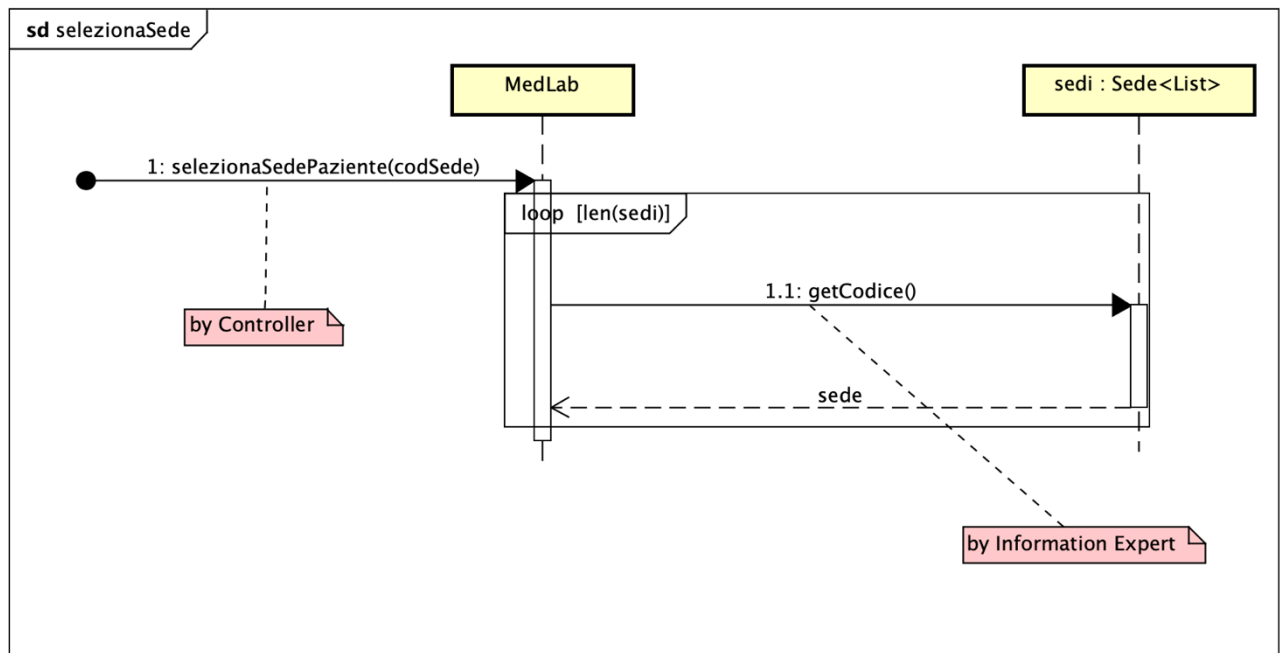


- confermaPaziente

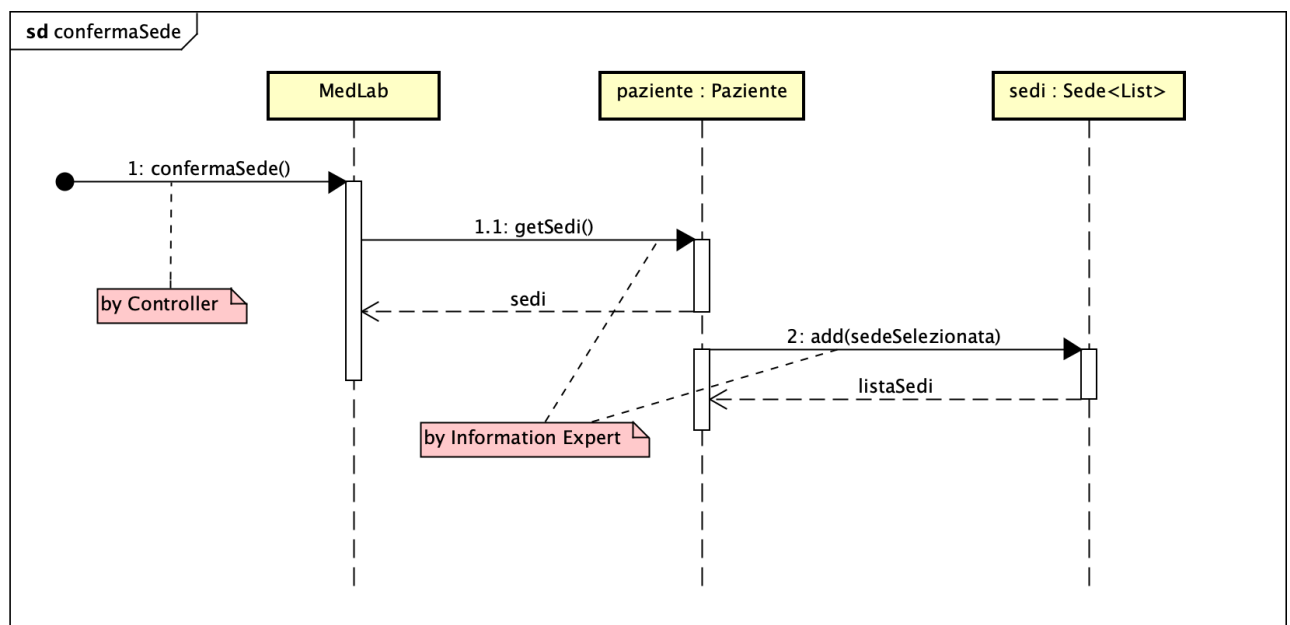


UC2

- SelezionaSedePaziente

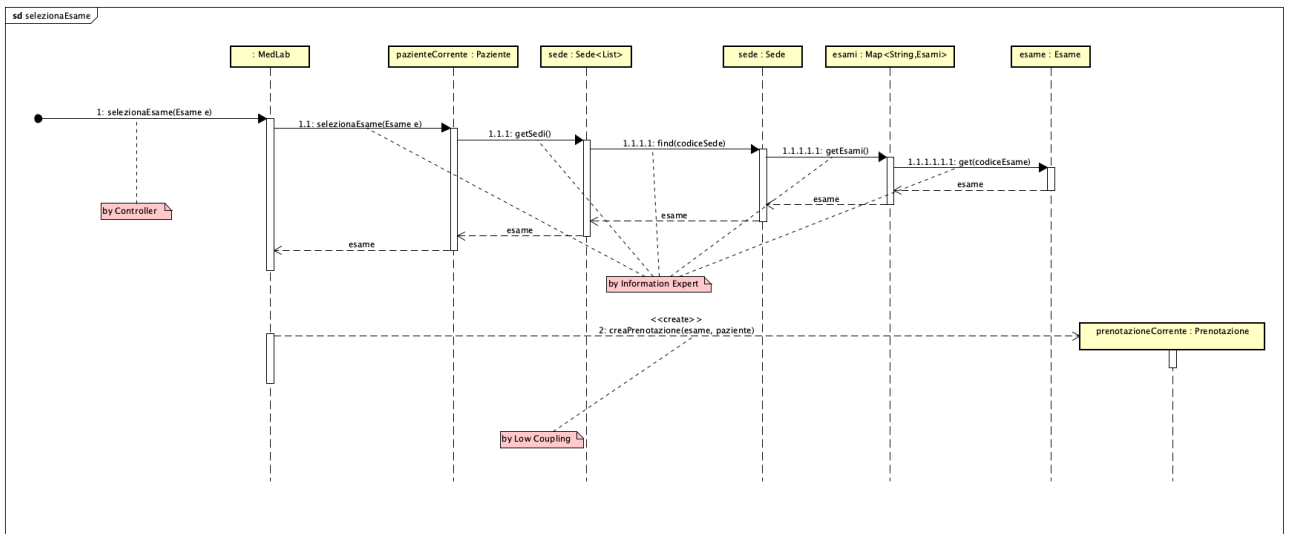


- confermaSede

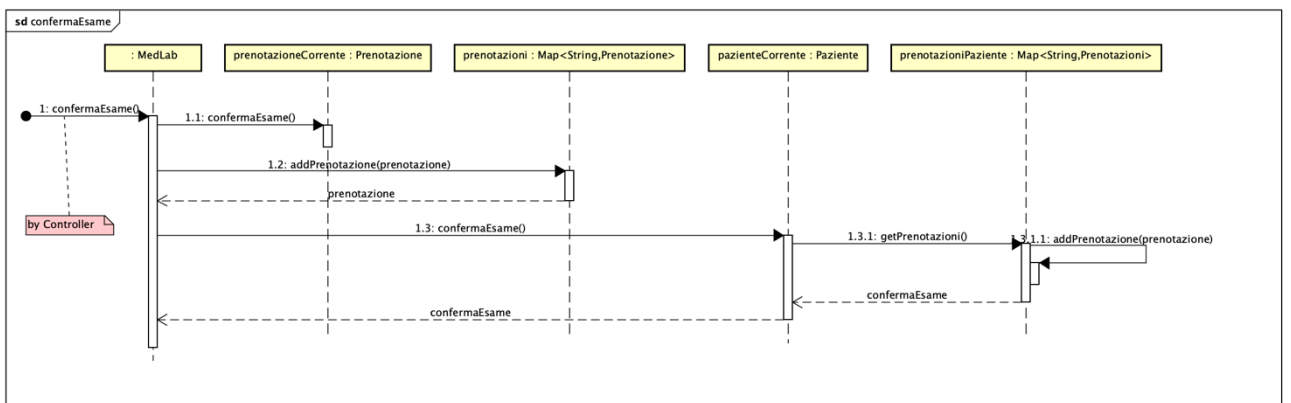


UC3

- selezionaEsame

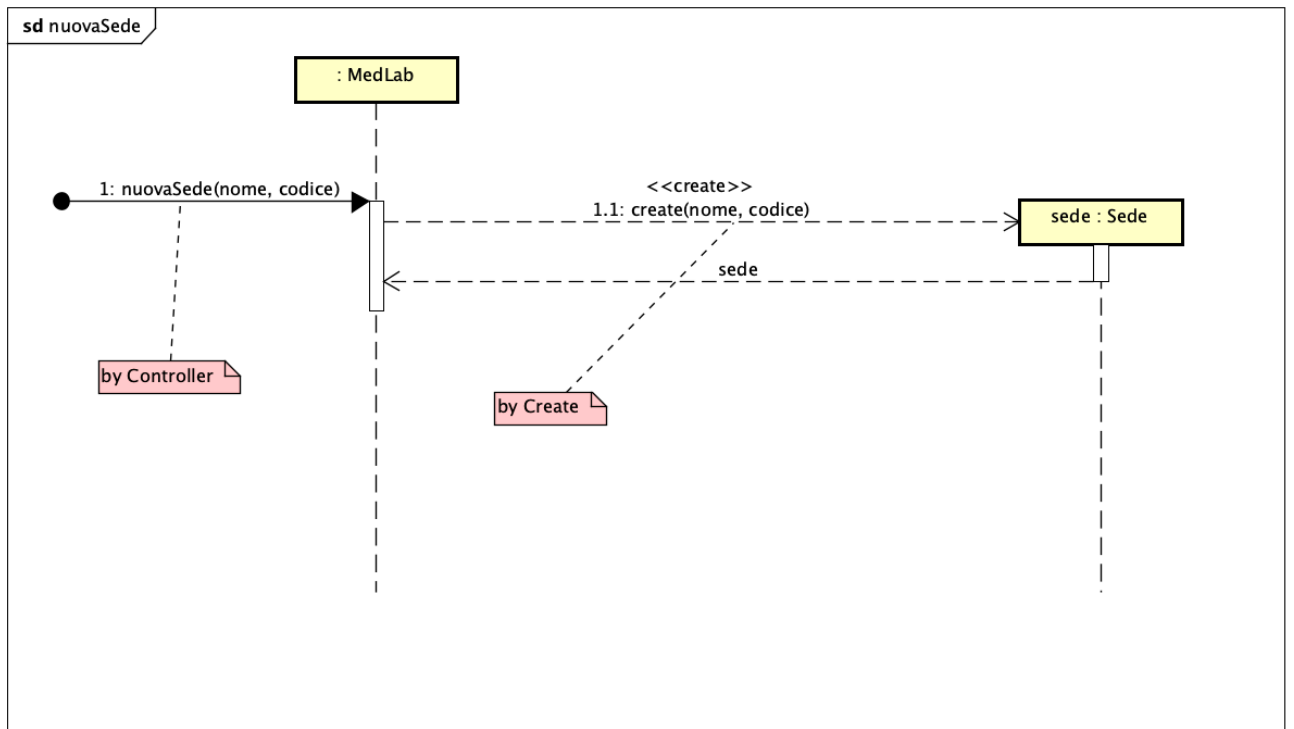


- ConfermaEsame

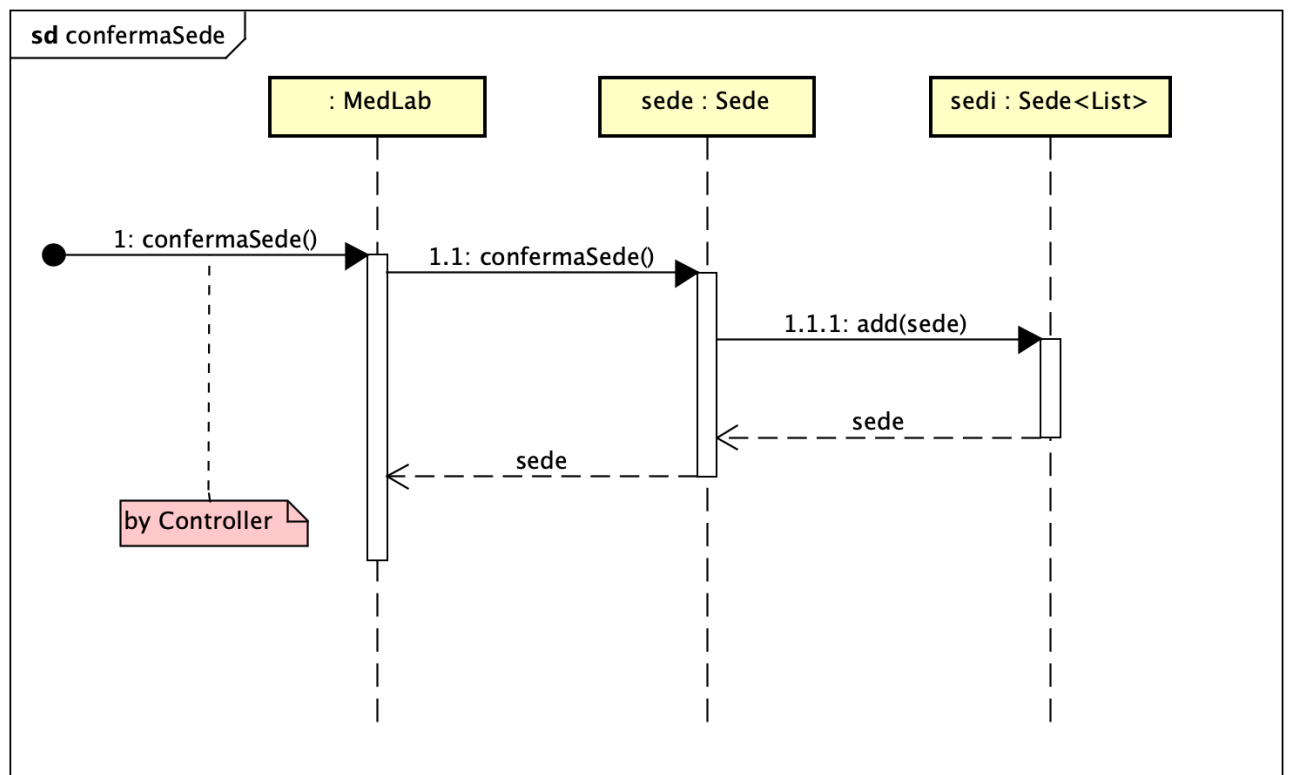


UC4

- nuovaSede



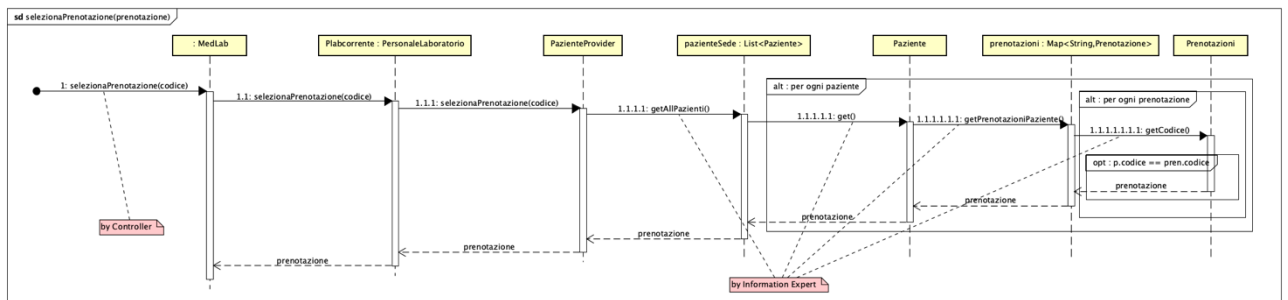
- confermaSede



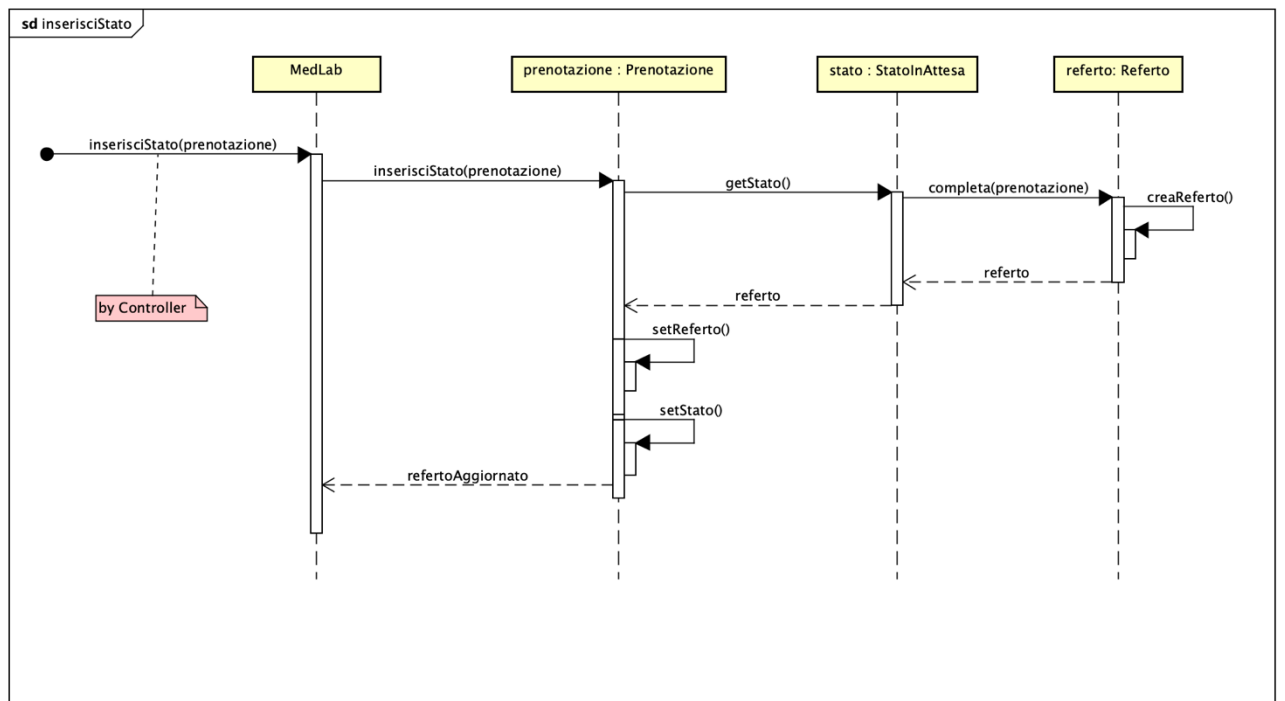
Iterazione 2:

UC5

- selezionaEsameSede

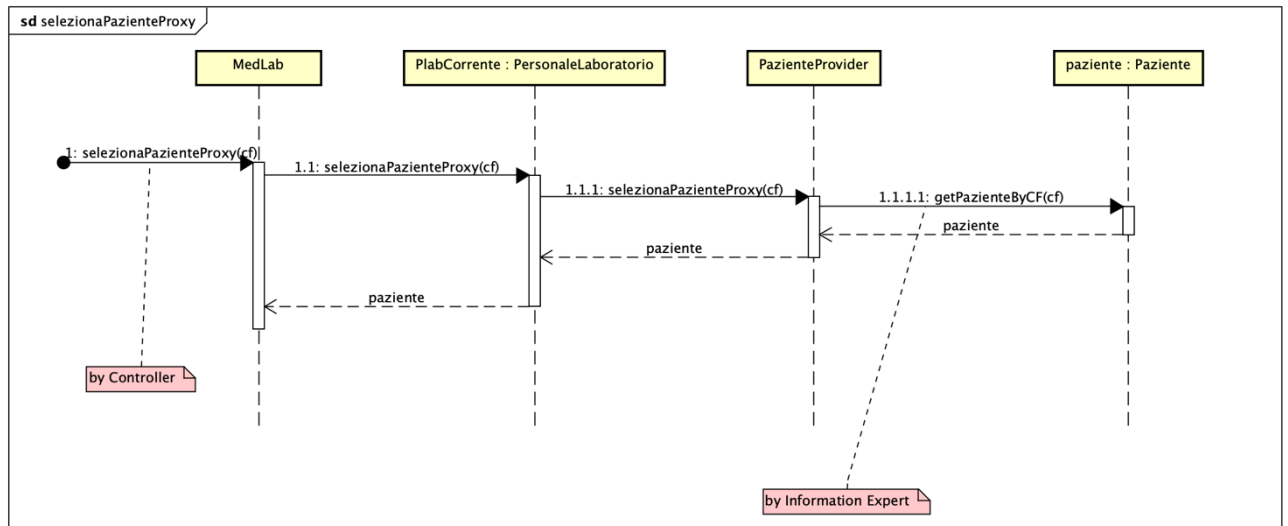


- inserisciStato

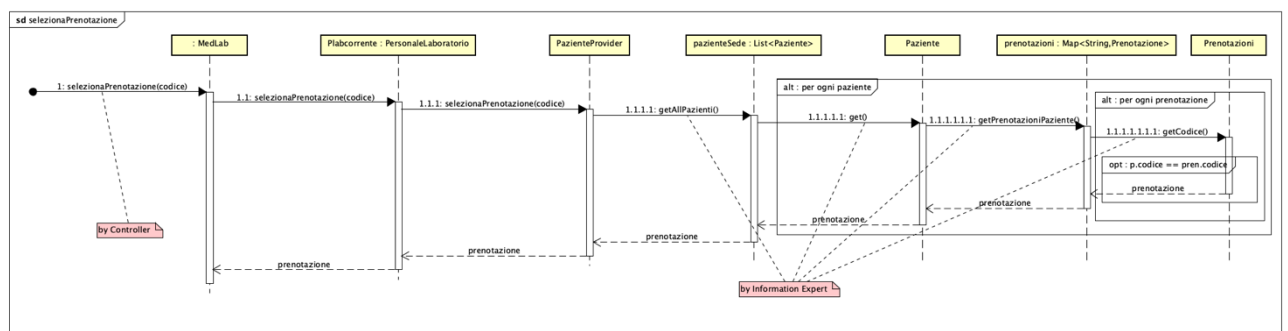


UC6

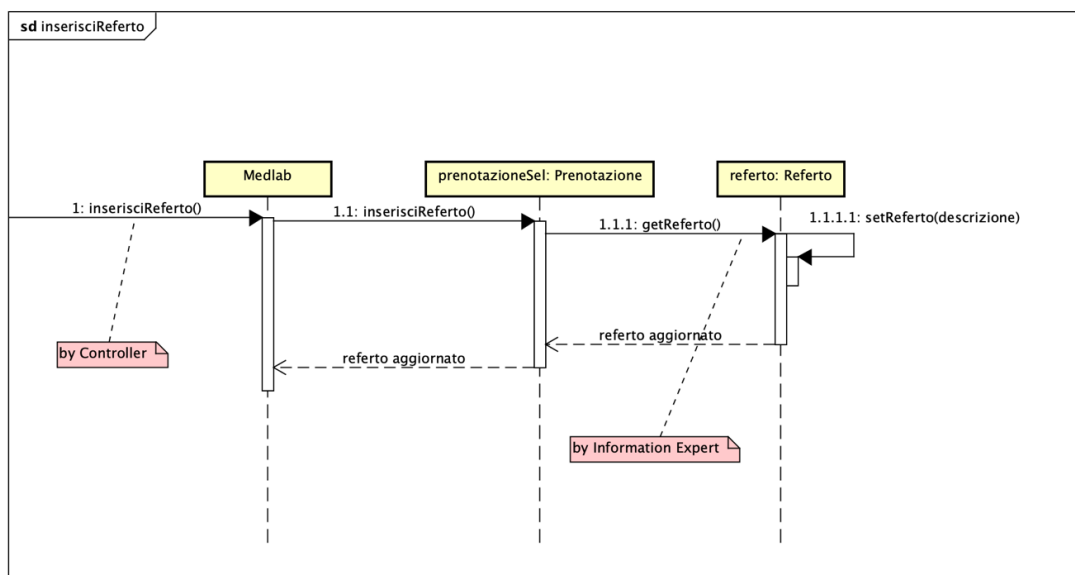
- selezionaPazienteProxy



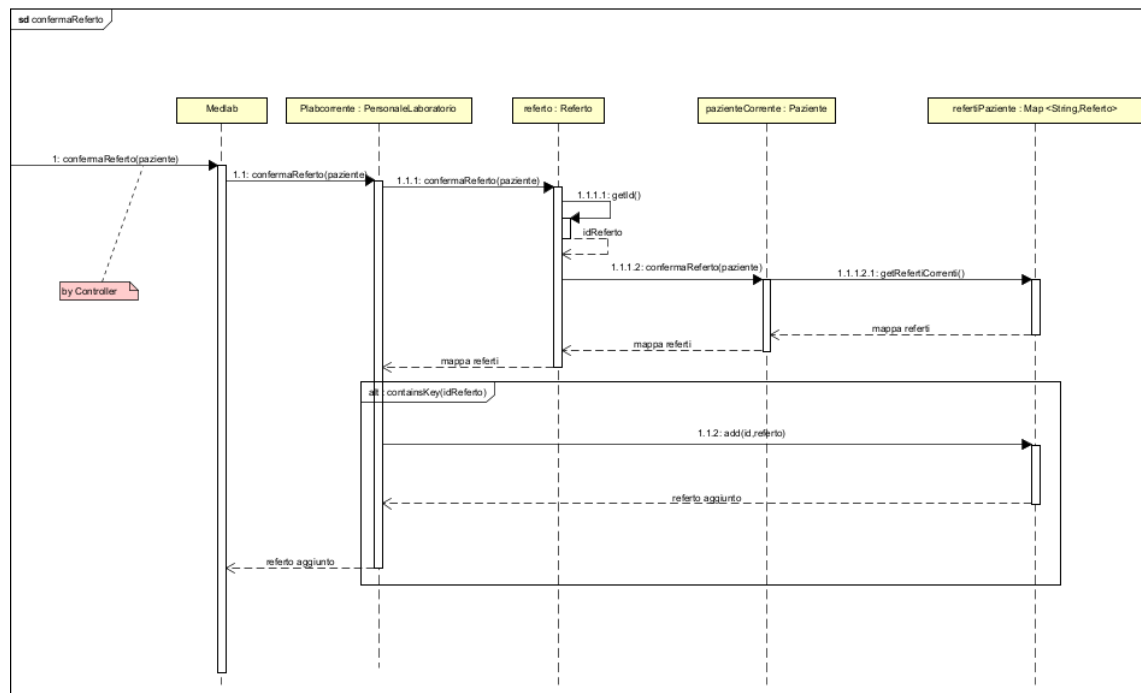
- selezionaPrenotazione



- inserisciReferto



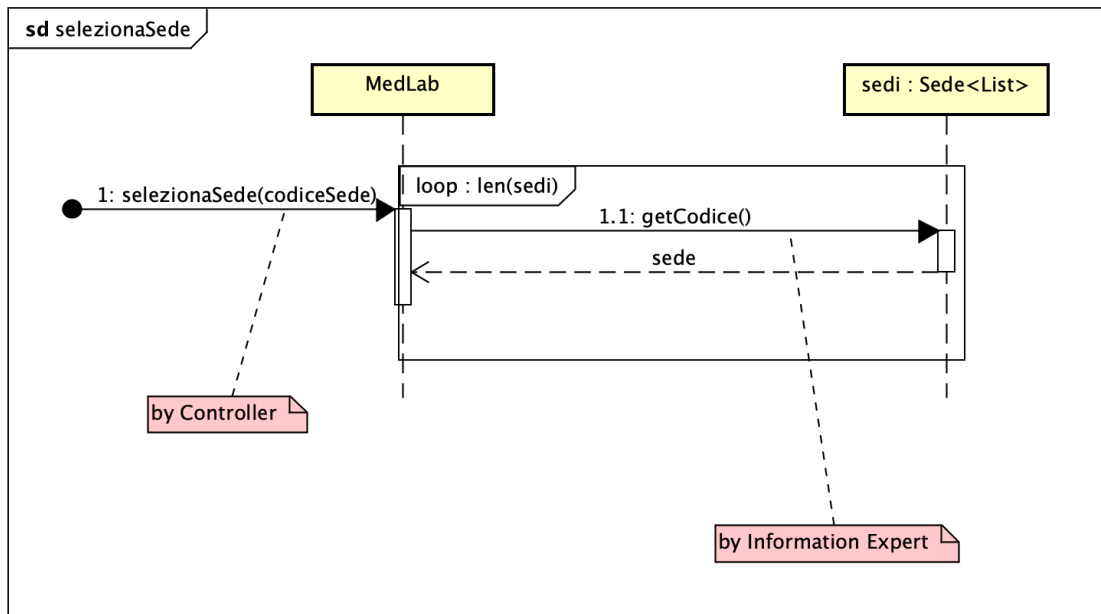
- confermaReferto



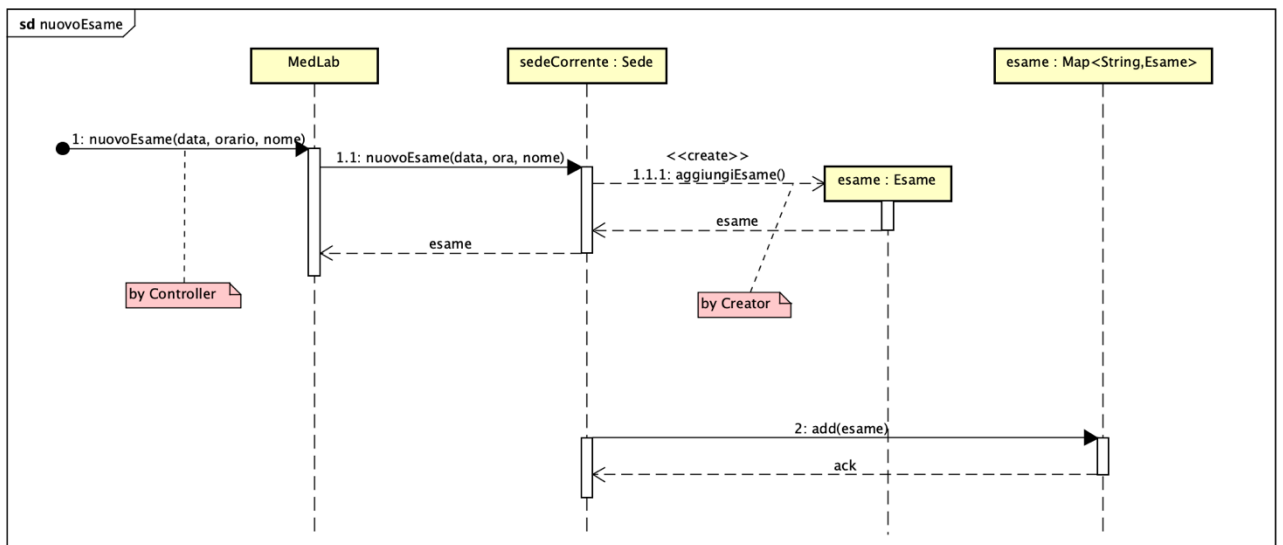
Iterazione 3:

UC10

- selezionaSede

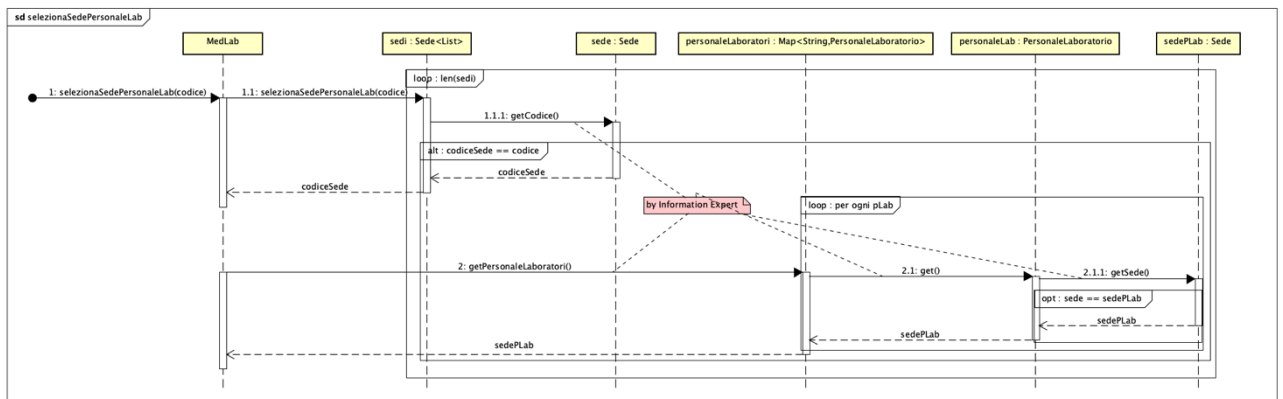


- nuovoEsame

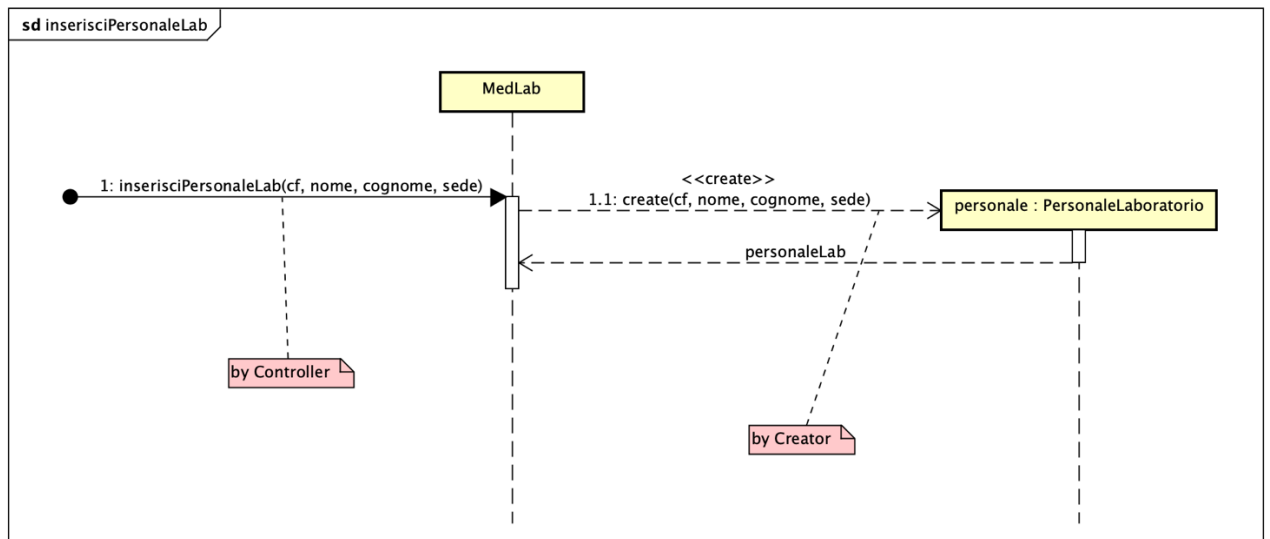


UC11

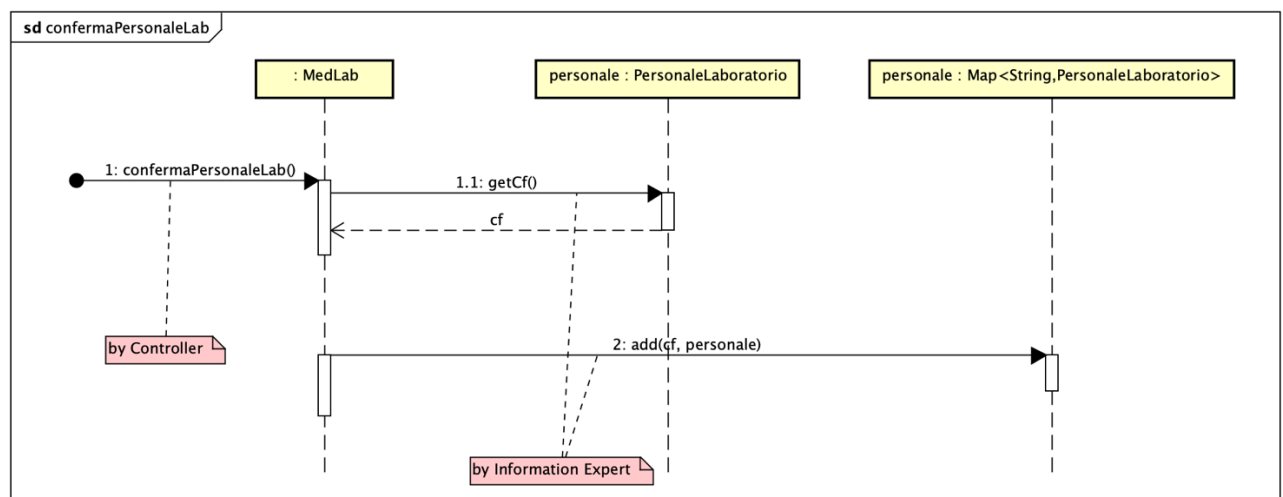
- selezionaSedePersonaleLab



- inserisciPersonaleLab

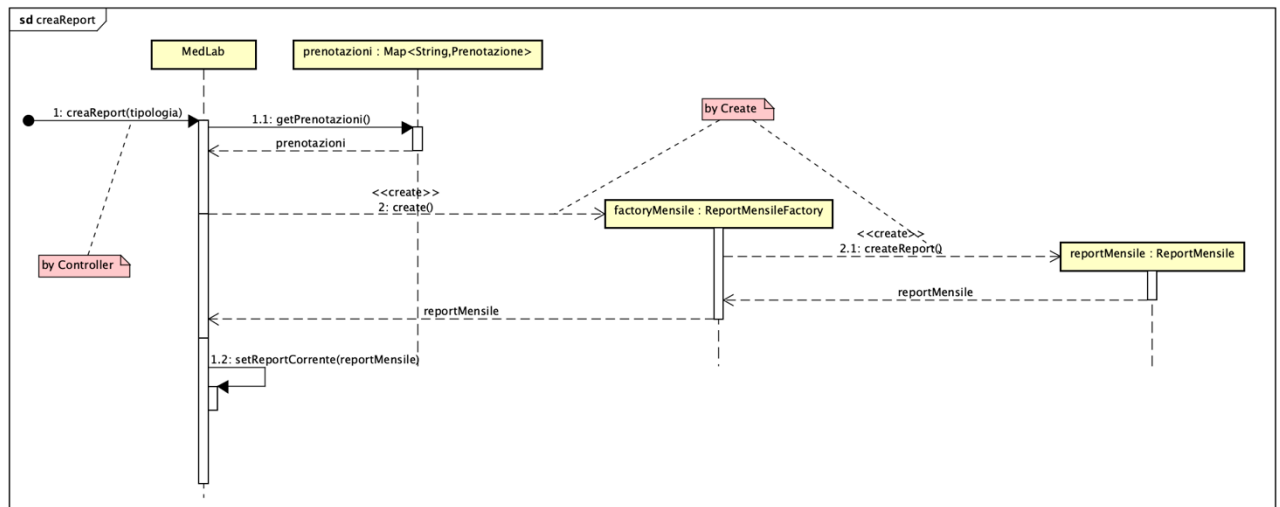


- confermaPersonaleLab



UC12

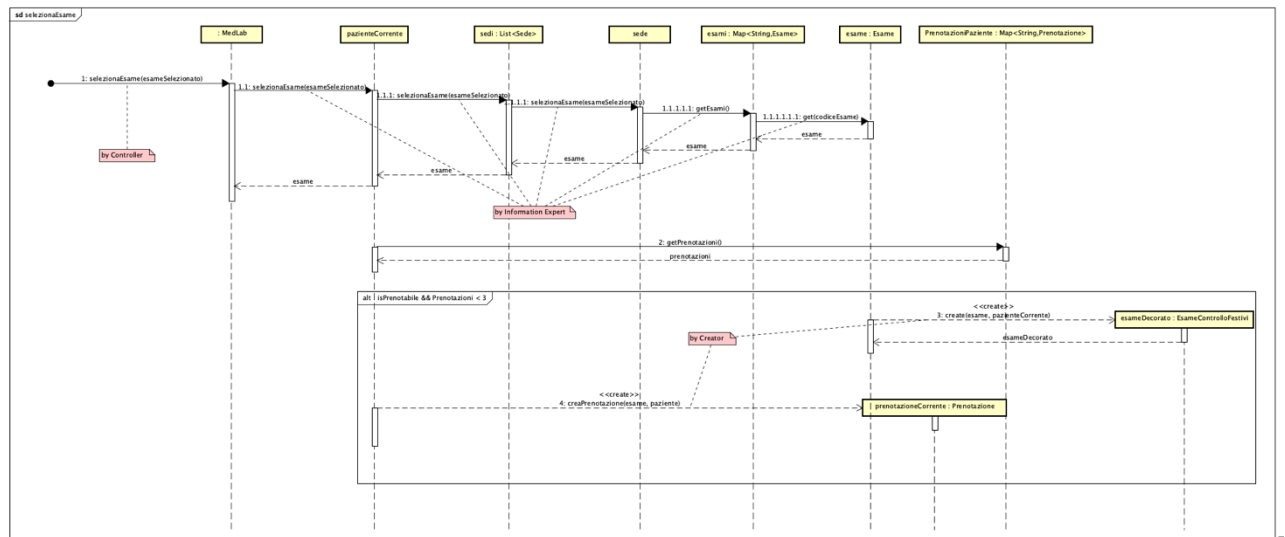
- creaReport



Lo stesso diagramma di comunicazione viene ripetuto anche nel caso in cui la tipologia di report è Semestrale ed Annuale.

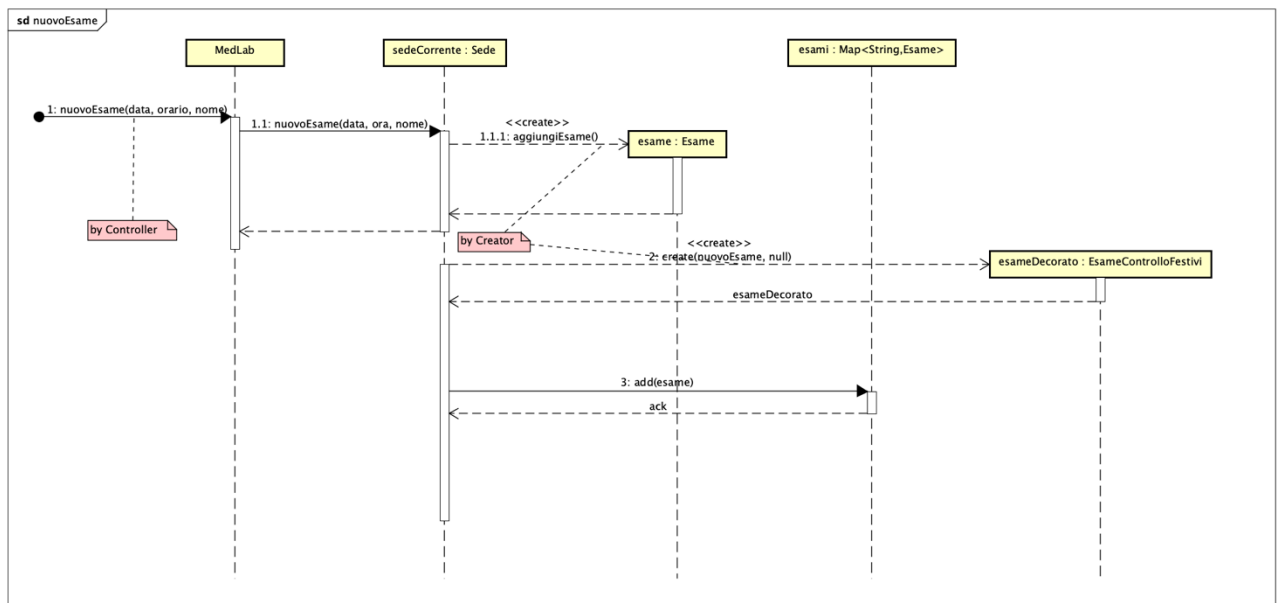
UC3

- `selezionaEsame`



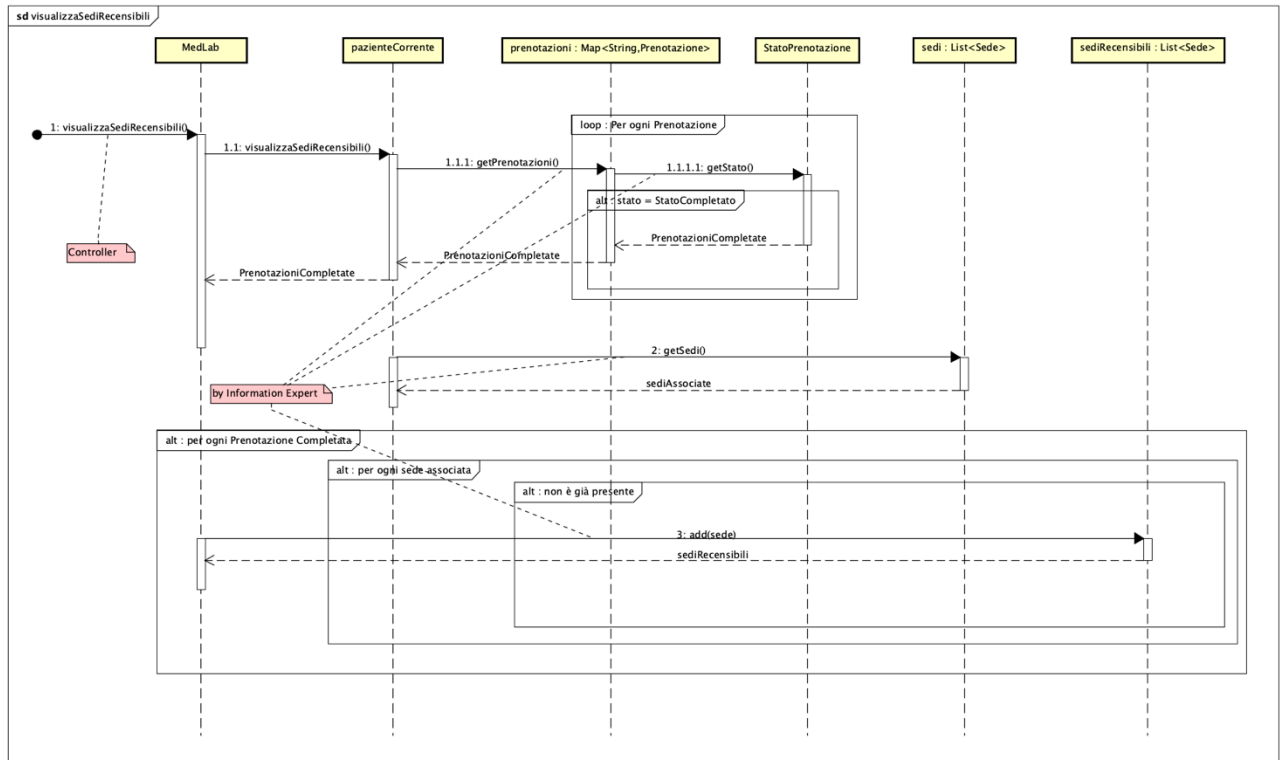
UC10

- nuovoEsame

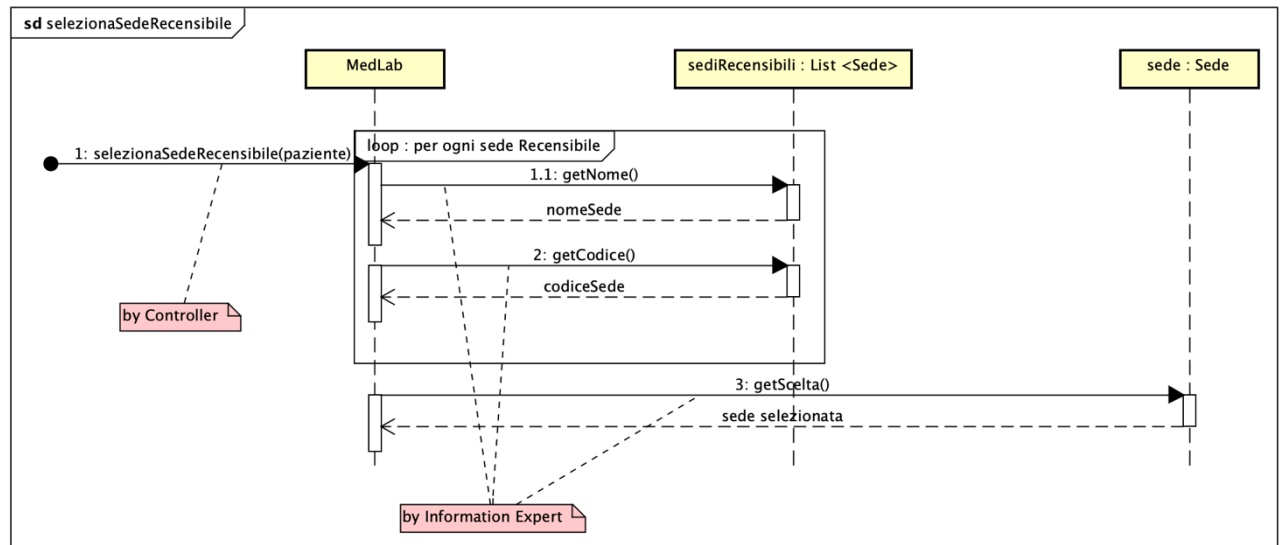


UC9

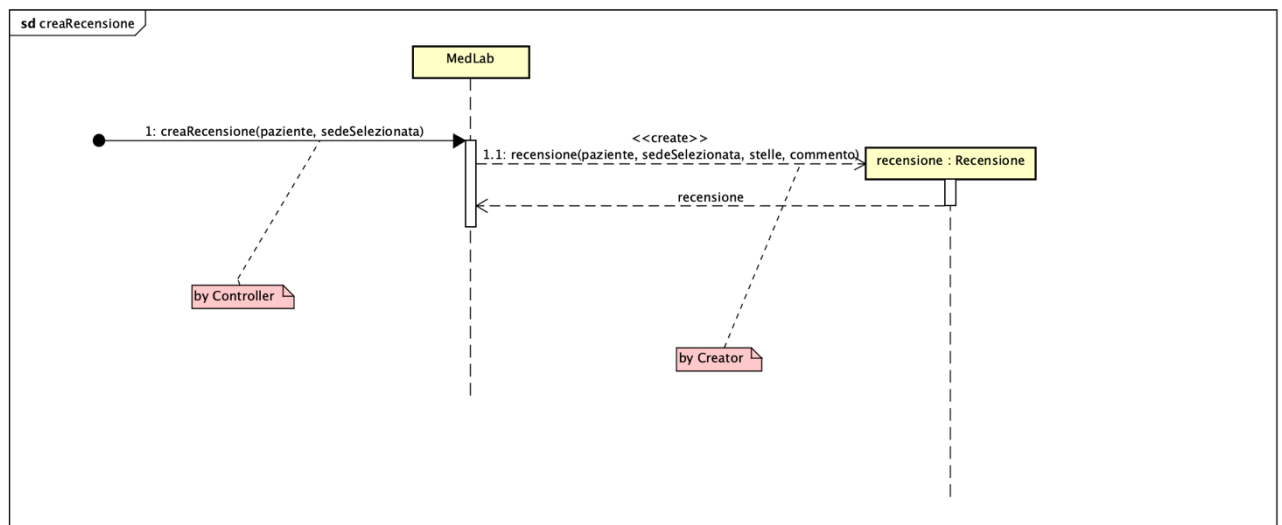
- visualizzaSediRecensibili



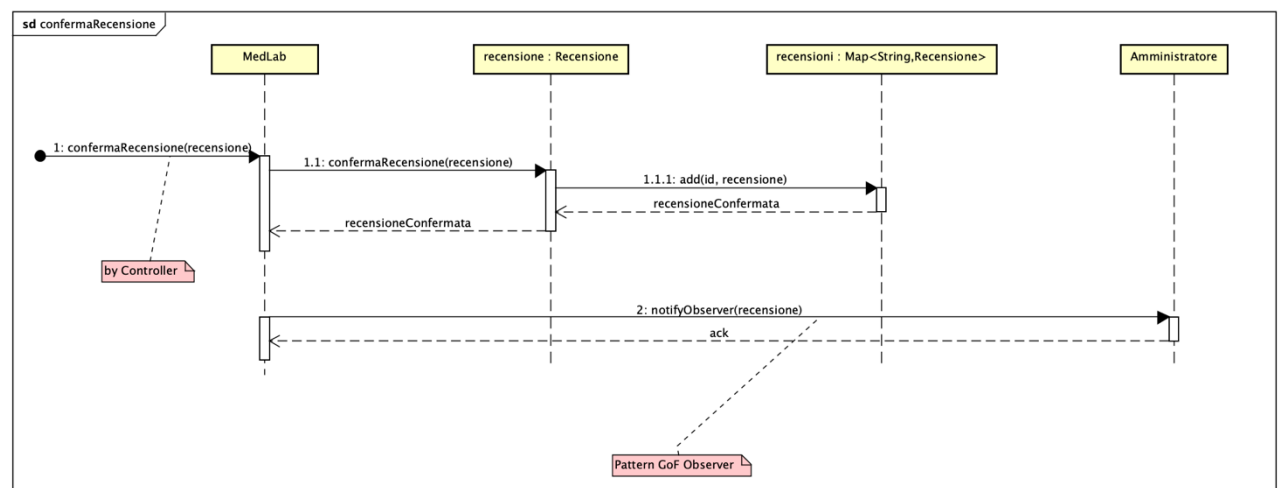
- selezionaSedeRecensibile



- creaRecensione



- confermaRecensione



Testing

Introduzione

Il testing rappresenta una fase fondamentale nello sviluppo del sistema MedLab, poiché garantisce che le funzionalità implementate rispondano ai requisiti richiesti. Testare il software in maniera sistematica ha permesso non solo di intercettare e correggere eventuali errori durante le varie fasi dello sviluppo, ma anche di costruire un'applicazione più solida e conforme ai requisiti definiti.

Un processo di testing efficace consente inoltre di ridurre i costi di manutenzione a lungo termine, prevenendo l'insorgenza di errori nelle fasi successive dello sviluppo.

Sebbene testare un'applicazione in modo esaustivo sia spesso impraticabile, a causa del numero elevato di combinazioni di input possibili, un piano di test ben strutturato consente di identificare tempestivamente comportamenti anomali o inattesi, migliorando la qualità generale del sistema.

Ogni fase del progetto ha previsto una verifica accurata delle nuove componenti introdotte, seguendo una strategia bottom-up, in cui si sono prima testate le singole unità di codice per poi integrare e validare il funzionamento complessivo del sistema.

In particolare, sono stati eseguiti test unitari per verificare in maniera isolata il corretto comportamento di ogni funzione rispetto ai valori attesi. Per questi test si è fatto uso del framework JUnit, che ha permesso di automatizzare la verifica di condizioni, stati, transizioni e dati specifici come, ad esempio, quelli relativi alle prenotazioni, ai referti o alla generazione dei report.

Parallelamente sono stati svolti anche test funzionali, focalizzati sull'interazione dell'utente con il sistema, in modalità "scatola nera". In questo caso, l'attenzione è stata rivolta alla verifica del corretto flusso delle operazioni nei casi d'uso principali (come UC3 "Prenotazione Esame", UC6 "Gestione Referti" o UC12 "Generazione Report"), simulando scenari realistici di utilizzo e valutando se gli output generati fossero coerenti con le attese.

Nel complesso, il processo di testing ha accompagnato ogni iterazione del progetto MedLab, contribuendo a costruire un sistema più robusto, sicuro e coerente agli obiettivi previsti. Attraverso la combinazione di test automatici e verifiche funzionali, è stato possibile migliorare l'affidabilità dell'applicazione e aumentare la fiducia nella sua operatività, soprattutto in un contesto delicato come quello della gestione di dati clinici e sanitari.

Individuazione dei casi di test e testing unitario

Il processo di testing è stato fondamentale nello sviluppo dell'applicazione MedLab, con l'obiettivo di garantire che il sistema funzionasse correttamente secondo le specifiche definite. Abbiamo adottato principalmente un approccio di testing Bottom-Up, implementando test unitari automatizzati tramite il framework JUnit. Questo ci ha consentito di testare e verificare con precisione ogni componente isolatamente prima della sua integrazione nel sistema completo.

Di seguito, riportiamo i test suddivisi per iterazione e funzionalità, descrivendo per ciascuno lo scopo e la realizzazione.

Iterazione 1:

testInit():

- scopo: inizializzare i dati necessari per i test, creando oggetti coerenti e riutilizzabili.
- Realizzazione: si crea un paziente, un personale di laboratorio e un esame, successivamente si inizializza una prenotazione con questi oggetti e una data specifica. Si verificano la correttezza dei dati inizializzati.

testNuovoPaziente():

- scopo: verificare la corretta creazione e inizializzazione di un nuovo paziente.
- realizzazione: si crea un nuovo paziente tramite il metodo nuovoPaziente, verificando che venga impostato correttamente come paziente e che i suoi attributi siano corretti.

testConfermaPaziente():

- scopo: verificare che il paziente venga correttamente inserito nella mappa dei pazienti e che l'attributo pazienteCorrente venga azzerato.
- Realizzazione: dopo la creazione di un paziente, si esegue confermaPaziente e si verifica che la mappa sia aggiornata e pazienteCorrente sia null.

testSelezionaPazienteEsistente():

- Scopo: verifica che un paziente possa essere selezionato correttamente tramite codice fiscale.
- Realizzazione: si seleziona un paziente già presente e si verifica che venga impostato correttamente come corrente.

testCodiceFiscaleDuplicato():

- Scopo: verificare che il sistema gestisca correttamente la duplicazione del codice fiscale.
- Realizzazione: si tenta di creare un paziente con CF già esistente, verificando il lancio di un'eccezione.

testRegistrazioneSedeValida():

- Scopo: verificare che una sede venga registrata solo se ha tutti i dati corretti.
- Realizzazione: si crea una sede con dati validi e si verifica che venga accettata senza errori.

testPrenotazioneEsameSuccesso():

- Scopo: verificare la prenotazione corretta di un esame.
- Realizzazione: si seleziona un esame disponibile, si conferma la prenotazione e si verifica l'associazione.

testPrenotazioneEsameGiaPrenotato():

- Scopo: verificare che non sia possibile prenotare un esame già prenotato.
- Realizzazione: si segna un esame come prenotato e si tenta di prenotarlo, aspettandosi un'eccezione.

testStatoInizialeEsame():

- Scopo: verificare lo stato iniziale corretto degli esami appena creati.
- Realizzazione: si crea un nuovo esame e si verifica che non sia prenotato e che il codice sia generato correttamente.

testNuovaSede()

- Scopo: verifica la creazione corretta di una nuova sede.
- Realizzazione: si crea una sede e si verifica che il nome e codice siano corretti e che la sede venga impostata come corrente.

testConfermaNuovaSede():

- Scopo: verificare l'inserimento della sede corrente nella lista delle sedi.
- Realizzazione: si esegue confermaSede dopo aver creato una sede e si verifica che venga aggiunta alla lista e sedeCorrente sia null.

Iterazione 2:

Classe PersonaleLabTest

testAggiornaReferto():

- Scopo: verificare che un operatore possa aggiornare correttamente un referto.
- Realizzazione: si simula l'inserimento di una descrizione tramite input e si verifica l'aggiornamento del referto.

testInserisciReferto():

- Scopo: verificare che l'inserimento di un referto da parte del personale consenta di acquisire correttamente il risultato da input.
- Realizzazione: viene simulato l'input da tastiera con una stringa "tutto ok", utilizzando uno InputStream. Si crea un referto e lo si imposta come corrente. Si invoca il metodo inserisciReferto che acquisisce il testo da input. Si verifica che il risultato del referto corrisponda alla stringa inserita. Infine, si ripristina il System.in.

testConfermaReferto():

- Scopo: verificare che il referto inserito venga correttamente associato al paziente e che il personale non mantenga più il riferimento temporaneo al referto dopo la conferma.
- Realizzazione: si crea un referto e si simula l'inserimento del risultato "Tutto ok". Dopo aver impostato il referto corrente, si invoca confermaReferto. Si verifica che il referto sia presente nei referti del paziente, che contenga il risultato corretto e che il riferimento temporaneo nel personale sia stato azzerato.

testInserisciStato():

- Scopo: verificare che lo stato dell'esame possa essere aggiornato correttamente.
- Realizzazione: si assegna uno stato a un esame e si controlla che venga modificato coerentemente.

testVisualizzaPazientiAssociatiAllaSede():

- Scopo: verificare che siano visualizzati correttamente i pazienti associati a una determinata sede.
- Realizzazione: si crea una sede con pazienti associati e si verifica che vengano restituiti correttamente.

testSelezionaPazienteEsistente():

- Scopo: verificare la selezione di un paziente esistente tramite codice fiscale.
- Realizzazione: si passa il codice fiscale corretto e si verifica che venga restituito il paziente atteso.

testVisualizzaPrenotazioniConfermate():

- Scopo: verificare che siano restituite correttamente solo le prenotazioni confermate.
- Realizzazione: si crea un set di prenotazioni confermate e non confermate e si verifica il filtraggio corretto.

testListaPazientiDellaSede():

- Scopo: verificare che venga restituita correttamente la lista dei pazienti della sede.
- Realizzazione: si verifica che il paziente precedentemente aggiunto sia presente nella lista restituito.

testSelezionaPrenotazioneEsistente():

- Scopo: verificare la selezione di una prenotazione esistente.
- Realizzazione: si passa il codice della prenotazione creata e si verifica che venga restituita correttamente.

classe PazienteTest

testVisualizzaReferti():

- Scopo: verificare che un paziente possa visualizzare correttamente i propri referti.
- Realizzazione: si aggiungono referti al paziente e si verifica che vengano restituiti correttamente alla richiesta.

classe PazienteProviderMock per testare funzionalità del proxy

- Scopo: simula il comportamento di un provider esterno per i pazienti.
- Realizzazione: si implementano i metodi getAllPazienti e getPazienteByCF per restituire risultati simulati durante i test.

Iterazione 3:

testNuovoEsameAggiunto():

- Scopo: verificare che l'aggiunta di un nuovo esame nella sede corrente aumenti correttamente il numero di esami disponibili e che, al termine dell'operazione, la sede corrente venga azzerata.
- Realizzazione: viene impostata la sede corrente su sedeTest e si registra il numero iniziale di esami. Si invoca il nuovoEsame con data, orario e descrizione dell'esame. Si verifica che il numero di esami sia aumentato di 1 e che la sede corrente sia stata reimpostata a null.

testInserisciPersonaleLab():

- Scopo: verificare che l'inserimento di un nuovo membro del personale di laboratorio imposti correttamente tutte le proprietà dell'oggetto in costruzione (codice fiscale, nome, cognome, sede).
- Realizzazione: viene chiamato il metodo inserisciPersonaleLab con dati di esempio. Successivamente si recupera il personale corrente e si verificano, tramite assertAllert, tutte le proprietà assegnate.

testConfermaPersonaleLab():

- Scopo: verificare che la conferma del personale inserito completi correttamente l'operazione, azzerando il riferimento temporaneo al personale e aggiungendo l'oggetto alla mappa del personale registrato.
- Realizzazione: dopo aver chiamato inserisciPersonaleLab, si invoca confermaPersonaleLab. Si verifica che il campo personaleLaboratorioCorrente sia stato impostato a null e che la mappa getPersonaleLaboratori contenga il nuovo personale identificato dal codice fiscale.

testGeneraReportMensile():

- Scopo: verificare che il sistema generi correttamente un report mensile contenente le prenotazioni e le statistiche di genere e stato cronico.
- Realizzazione: si creano due prenotazioni per pazienti di genere diverso, uno dei quali cronico. Si esegue la generazione del report mensile tramite creaReport("mensile") e si verifica il contenuto del report generato.

classe PazienteTest

testSetPassword():

- Scopo: verificare la modifica corretta della password da parte del paziente.
- Realizzazione: si imposta una nuova password e si controlla che venga memorizzata correttamente.

Iterazione 4:

testPrenotazioniMaxPerGiorno():

- Scopo: verificare che il sistema consenta la prenotazione finché il limite massimo giornaliero non è raggiunto.
- Realizzazione: si aggiungono due prenotazioni e si verifica che sia ancora possibile effettuare una terza.

testLasciaRecensione():

- Scopo: verificare che un paziente possa lasciare una recensione associata ad una prenotazione.
- Realizzazione: si inserisce una recensione e si controlla che venga associata correttamente alla prenotazione.

testVisualizzaSediRecensibili():

- Scopo: verificare che il sistema identifichi correttamente le sedi in cui il paziente ha completato esami e che, quindi, può recensire.
- Realizzazione: si assegnano due sedi, ciascuna con un esame completato associato al paziente. Si inseriscono queste prenotazioni nello storico del paziente. Si chiamano il metodo visualizzaSediRecensibili, e si verifica che entrambe le sedi vengano restituite come recensibili.

testSelezionaSedeRecensibile():

- Scopo: verificare che un paziente possa selezionare correttamente una sede in cui ha effettuato un esame completato.
- Realizzazione: viene simulata una prenotazione completata in una sede. Usando uno Scanner con input "1", si verifica che il metodo selezionaSedeRecensibile() restituisca la sede corretta selezionata dal paziente.

testCreaRecensioneDatiValidi():

- Scopo: assicurarsi che la creazione della recensione funzioni correttamente quando vengono forniti dati validi.
- Realizzazione: viene simulata l'esperienza del paziente che ha completato un esame in una sede. Tramite uno Scanner con input "5\nOttimoServizio", si crea la recensione e si verificano i dati inseriti (valutazione, commento, associazione a paziente e sede).

testConfermaRecensioneValida():

- Scopo: verificare che una recensione venga notificata correttamente all'amministratore dopo la conferma.
- Realizzazione: si crea una recensione, si aggiunge l'amministratore come observer, e invoca confermaRecensione. Si verifica che la recensione sia stata aggiunta alle recensioni non lette dell'amministratore.

classe AmministratoreTest():

testVisualizzaRecensioniNonLette():

- Scopo: verificare che l'amministratore possa visualizzare solo le recensioni non ancora lette.
- Realizzazione: si inseriscono recensioni con stato "letto" e "non letto" e si controlla il filtraggio.

testAggiungiRecensioneNonLetta():

- Scopo: verificare che una recensione venga inizialmente registrata come non letta.
- Realizzazione: si inserisce una nuova recensione e si verifica che lo stato iniziale sia "non letta".

testVisualizzaPrenotazioniAttive():

- Scopo: verificare che siano restituite solo le prenotazioni attive del paziente.
- Realizzazione: si crea un insieme di prenotazioni con vari stati e si controlla il filtraggio corretto.

testUpdate():

- Scopo: verificare che il metodo update dell'amministratore, richiamato da un osservatore (in stile Observer), aggiunga correttamente una nuova recensione all'elenco delle recensioni non lette.
- Realizzazione: si crea un oggetto Recensione e si simula una notifica invocando update dell'amministratore. Si verifica che la lista delle recensioni non lette contenga esattamente una recensione e che sia proprio quella passata in input.

sono stati aggiunti anche dei test per il login

classe AmministratoreTest

testVerificaPassword():

- Scopo: verificare il controllo corretto della password al momento del login dell'amministratore.
- Realizzazione: si imposta una password e si verifica che l'autenticazione fallisca o riesca correttamente in base all'input.

classi PazienteTest

testVerificaPasswordCorretta():

- Scopo: verificare che il paziente venga autenticato correttamente con la password giusta.
- Realizzazione: si verifica che il metodo restituisca true con la password corretta e false con quella errata.

Test di Sistema

Alla conclusione dello sviluppo dell'applicazione MedLab, è stata condotta una fase di test manuale con esito positivo. Questa fase ha previsto la verifica dell'interfaccia utente e delle funzionalità principali del sistema, con l'obiettivo di garantire coerenza con le specifiche progettuali, reattività nelle interazioni e integrità dei dati trattati.

Verifica dell'Interfaccia Utente

Obiettivo: garantisce che l'interfaccia grafica di MedLab fosse chiara, ben strutturata e facilmente utilizzabile da tutte le tipologie di utenti (Paziente, Personale di Laboratorio, Amministratore).

Attività: sono state esplorate le principali schermate dell'applicazione, come la prenotazione degli esami, la visualizzazione delle prenotazioni, la gestione del personale e la valutazione degli esami. È stata verificata la coerenza visiva e la facilità di utilizzo.

Test delle Funzionalità Principali

Obiettivo: verificare che le funzionalità principali previste dai casi d'uso (prenotazioni, conferme, valutazioni esami, notifiche, visualizzazioni) fossero correttamente implementate e soddisfacessero i requisiti.

Attività: sono stati eseguiti scenari d'uso realistici per ogni tipo di utente, ad esempio: un paziente che effettua una prenotazione rispettando il limite massimo giornaliero, il personale che valuta un esame inserisce i relativi referti.

Analisi della Coerenza dei Dati

Obiettivo: verificare che i dati mostrati all'utente siano corretti, aggiornati e coerenti con le informazioni persistenti nel sistema.

Attività: i dati presenti nelle interfacce sono stati confrontati con quelli contenuti nel database, per garantire che le operazioni di inserimento, aggiornamento e visualizzazione si comportassero come previsto, mantenendo l'integrità del sistema.

Valutazione delle Prestazioni

Obiettivo: garantisce che MedLab mantenesse un buon livello di reattività anche sotto carico.

Attività: sono stati simulati scenari con più prenotazioni contemporanee, gestione parallela di più utenti e operazioni ripetute per osservare i tempi di risposta e individuare eventuali problemi nelle prestazioni.

In conclusione, questa fase di test ha confermato che MedLab risponde alle specifiche progettuali, con un'interfaccia intuitiva, funzionalità affidabili e prestazioni soddisfacenti anche in condizioni di utilizzo intensivo. La valutazione globale della verifica manuale è stata positiva, rendendo l'applicazione pronta per la distribuzione.

Refactoring e conclusioni

Refactoring

Durante le fasi finali del progetto MedLab, è stato eseguito un refactoring dell'autenticazione utente per rendere l'accesso al sistema più chiaro, sicuro e funzionale. È stato introdotto un meccanismo che prevede l'inserimento di un codice fiscale e di una password, associati in maniera univoca a ciascun tipo di attore.

Questo approccio consente di distinguere chiaramente tra i tre ruoli presenti nel sistema (Paziente, Personale di Laboratorio e Amministratore) fornendo a ciascuno un menu dedicato con funzionalità specifiche.

Durante i test manuali sono stati simulati i tre accessi distinti: Amministratore accede con le credenziali cf 'a' e password 'a', Personale di Laboratorio accede con le credenziali cf 'b' e password 'b' e Paziente accede con le credenziali cf 'c' e password 'c'.

Una volta autenticato, l'utente viene indirizzato ad un menu coerente con il proprio ruolo. Questo menu consente di accedere esclusivamente alle operazioni di competenza, evitando confusioni o errori. Il paziente, ad esempio, può registrare la sede, prenotare esami, visualizzare i referti e lasciare una recensione. Il personale ha accesso alle prenotazioni, può modificare lo stato e gestire i referti, mentre l'amministratore può supervisionare il sistema, gestire sedi, personale, esami e generare report statistici. Il refactoring del sistema di autenticazione ha permesso quindi una navigazione più fluida e sicura, rendendo il flusso dell'interazione utente ben definito e in linea con il modello dei casi d'uso del progetto.

Test di Accettazione

Al termine dello sviluppo, è stato eseguito un test di accettazione manuale su tutto il sistema. L'obiettivo era simulare l'esperienza reale degli utenti nei loro rispettivi ruoli e verificare che tutte le funzionalità rispondessero correttamente ai requisiti individuati durante l'analisi.

Durante la fase di testing sono stati esplorati e verificati i vari casi d'uso identificativi.

Questa fase ha avuto lo scopo di validare il comportamento complessivo del sistema, assicurandosi che ogni attore potesse operare senza incongruenze e che i dati visualizzati fossero coerenti con quelli memorizzati internamente.

La fase di testing ha anche messo in luce la reattività e la stabilità dell'interfaccia, confermando che il sistema è pronto per un utilizzo reale. La combinazione tra test automatici e test manuali ha rafforzato la qualità generale dell'applicazione.