



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

Documentazione sistema per la prenotazione dei passaporti

Ingegneria del Software

Settembre 2023

Filippo Barbieri (VR472408)

Indice

Requisiti ed interazioni utente-sistema	1
Specifiche sistema	1
Casi d'uso relativi al personale.....	1
Gestione slot (apertura)	2
Gestione slot (chiusura)	2
Gestione slot (visualizzazione)	3
Casi d'uso relativi ai cittadini	4
Prenotazione slot (rilascio).....	4
Prenotazione slot (rinnovo).....	4
Prenotazione slot (ritiro)	5
Registrazione	5
Diagrammi delle attività.....	6
Progettazione e implementazione del sistema.....	7
Processo di sviluppo.....	7
Progettazione e pattern architetturali usati	7
Implementazione e design pattern usati	9
Attività di test e validazione.....	10
Ispezione codice e documentazione	10
Alpha testing	10
Beta testing	10

Requisiti ed interazioni utente-sistema

Specifiche sistema

Il sistema informatico progettato per la gestione del servizio di prenotazione dei passaporti presso le diverse sedi preposte offre una serie di funzionalità che contribuiscono a semplificare il processo di prenotazione e migliorare l'efficienza complessiva.

Funzionalità del personale

- **Gestione delle disponibilità:** ogni dipendente gestisce gli slot automaticamente inseriti dal sistema nella propria sede di appartenenza nei giorni per cui è stata fornita la disponibilità. La gestione degli slot non gestiti consiste in aprire o chiudere gli slot in una determinata data/ora e indicare il tipo di servizio prestato.
In caso lo slot sia già stato aperto e non sia ancora stato occupato può essere chiuso, mentre se uno slot è occupato o chiuso se ne possono verificare i dati.
- **Modifica dei dati personali:** i dipendenti possono effettuare modifiche ai propri dati personali aggiornando la sede lavorativa, le disponibilità e la password di accesso.

Funzionalità dei cittadini

- **Registrazione:** i cittadini che desiderano prenotare un appuntamento devono registrarsi fornendo le proprie informazioni personali (nome, cognome, data e luogo di nascita, etc.). Con i dati forniti il sistema calcola il codice fiscale e verifica se il cittadino è presente nell'anagrafica di sistema, in caso negativo viene visualizzato un messaggio di errore con un contatto per richiedere assistenza.
- **Prenotazione appuntamento:** il cittadino registrato può accedere al sistema e prenotare appuntamento in uno slot libero a seconda del servizio e della sede richiesti.
Il tipo di servizio prenotato deve essere coerente rispetto alle precedenti richieste, cioè non può essere prenotato un ritiro se almeno un mese prima non era stato prenotato un rilascio. Conclusa la prenotazione, viene mostrata la lista di documenti che il cittadino dovrà presentare all'appuntamento.
- **Avvisi e notifiche:** il cittadino che ne abbia fatto richieste viene notificato se si liberano slot in un certo periodo di tempo per il servizio indicato.
Inoltre, in caso qualunque operazione dia esito negativo, l'utente visualizza un avviso che ne specifichi le motivazioni.
- **Modifica dei dati personali:** I cittadini registrati nel sistema hanno la possibilità di apportare modifiche ai loro dati personali non anagrafici, cioè informazioni come, la password e il numero di carta d'identità.

Il sistema supporta la concorrenza di più utenti che compiono la stessa operazione; nel caso della gestione degli slot da parte dei dipendenti e della prenotazione da parte dei cittadini, l'interfaccia utente viene aggiornata ogni 20 secondi.

In sintesi, il sistema di prenotazione proposto offre un'esperienza utente intuitiva per i cittadini e una gestione efficiente delle disponibilità per il personale, contribuendo a semplificare il processo di rilascio dei passaporti e a migliorare la qualità del servizio offerto.

Casi d'uso relativi al personale

Dopo aver completato il processo di autenticazione utilizzando le proprie credenziali di accesso (username e password), il dipendente avrà accesso a un'interfaccia dedicata che gli consentirà di gestire gli slot già predisposti dal sistema.

Gestione slot (apertura)

Tramite una tabella che visualizza gli slot assegnati al dipendente, quest'ultimo può aprire al pubblico uno slot non gestito e selezionarne il tipo.

Attori: Dipendente
Precondizioni: autenticazione, esistenza slot
Passi: <ol style="list-style-type: none">1. Il dipendente visualizza gli slot che il sistema gli ha assegnato nella propria sede lavorativa, di colore differente a seconda dello stato2. Il dipendente seleziona uno slot grigio (non gestito)3. Si apre un popup con il quale il dipendente può interagire4. Il dipendente seleziona il tipo e apre lo slot
Postcondizioni: lo slot è aperto al pubblico e colorato di verde nell'interfaccia

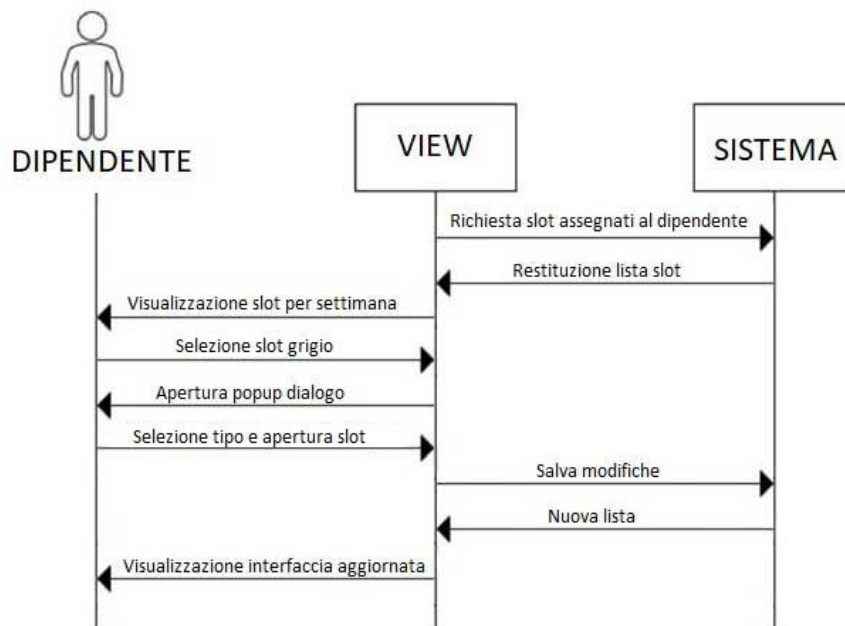


Figura 1: Sequence diagram apertura slot

Gestione slot (chiusura)

Tramite una tabella che visualizza gli slot assegnati al dipendente, quest'ultimo può chiudere uno slot non gestito o aperto (ma non occupato).

Attori: Dipendente
Precondizioni: autenticazione, esistenza slot
Passi: <ol style="list-style-type: none">1. Il dipendente visualizza gli slot che il sistema gli ha assegnato nella propria sede lavorativa, di colore differente a seconda dello stato2. Il dipendente seleziona uno slot grigio (non gestito) o verde (aperto)3. Si apre un popup con il quale il dipendente può interagire4. Il dipendente chiude lo slot che non sarà più prenotabile
Postcondizioni: lo slot non è prenotabile e viene colorato di rosso nell'interfaccia

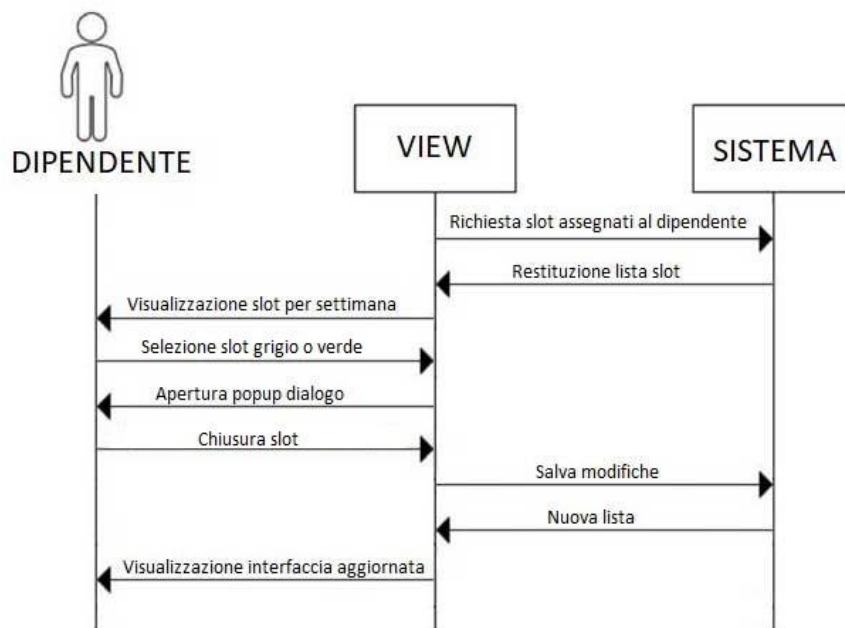


Figura 2: Sequence diagram chiusura slot

Gestione slot (visualizzazione)

Un dipendente può visualizzare le informazioni (quali tipo e cittadino) di uno slot già chiuso o occupato selezionandolo dall'interfaccia

Attori: Dipendente

Precondizioni: autenticazione, esistenza slot

Passi:

1. Il dipendente visualizza gli slot che il sistema gli ha assegnato nella propria sede lavorativa, di colore differente a seconda dello stato
2. Il dipendente seleziona uno slot rosso (chiuso) o arancione (occupato)
3. Si apre un popup che visualizza le informazioni
4. Il dipendente chiude il popup

Postcondizioni:

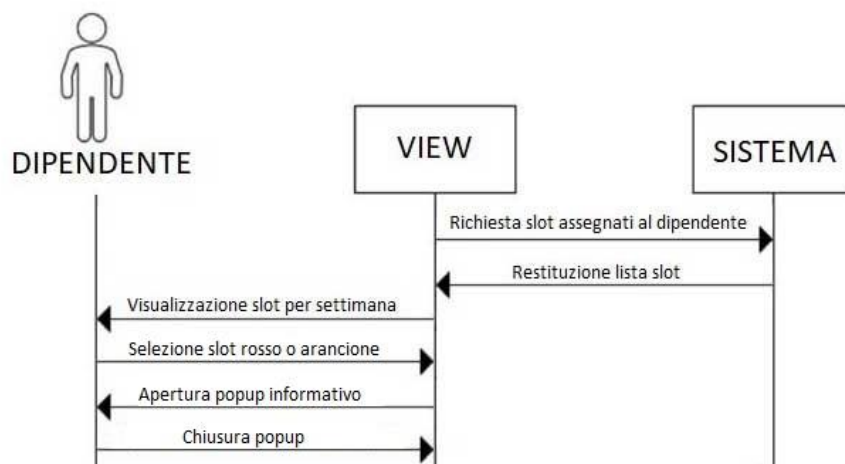


Figura 3: Sequence diagram visualizzazione slot

Casi d'uso relativi ai cittadini

Dopo essersi registrato una tantum ed essersi autenticato utilizzando le proprie credenziali di accesso (codice fiscale e password), il cittadino può prenotare uno slot per usufruire dei servizi messi a disposizione.

Prenotazione slot (rilascio)

Tramite una tabella che visualizza gli slot adibiti al rilascio nella sede selezionata, il cittadino può prendere appuntamento.

Attori: Cittadino
Precondizioni: registrazione, autenticazione, esistenza slot, non possesso del passaporto
Passi: <ol style="list-style-type: none">1. Il cittadino seleziona la sede2. Il cittadino seleziona il servizio di rilascio3. Il cittadino visualizza gli slot a disposizione4. Il cittadino seleziona uno slot verde (libero)5. Si apre un popup raffigurante i documenti necessari e il pulsante per confermare6. Il cittadino conferma l'appuntamento
Postcondizioni: lo slot viene assegnato al cittadino e colorato di arancione nell'interfaccia

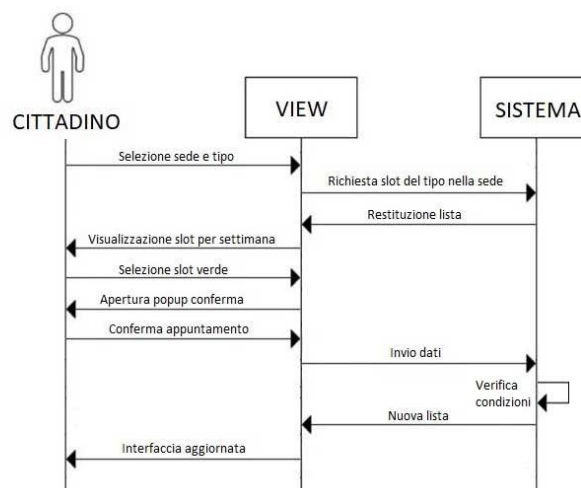


Figura 4: Sequence diagram prenotazione slot

Prenotazione slot (rinnovo)

Tramite una tabella che visualizza gli slot adibiti al rinnovo nella sede selezionata, il cittadino può prendere appuntamento.

Attori: Cittadino
Precondizioni: registrazione, autenticazione, esistenza slot, passaporto in scadenza
Passi: <ol style="list-style-type: none">1. Il cittadino seleziona la sede2. Il cittadino seleziona il servizio di rinnovo3. Il cittadino visualizza gli slot a disposizione4. Il cittadino seleziona uno slot verde (libero) nei sei mesi precedenti alla scadenza5. Si apre un popup raffigurante i documenti necessari e il pulsante per confermare6. Il cittadino conferma l'appuntamento
Postcondizioni: lo slot viene assegnato al cittadino e colorato di arancione nell'interfaccia

Il sequence diagram, condizioni escluse, corrisponde alla prenotazione di un rilascio.

Prenotazione slot (ritiro)

Tramite una tabella che visualizza gli slot adibiti al ritiro nella sede selezionata, il cittadino può prendere appuntamento per ritirare il documento precedentemente richiesto.

Attori: Cittadino
Precondizioni: registrazione, autenticazione, esistenza slot, richiesta rilascio o rinnovo
Passi: <ol style="list-style-type: none">1. Il cittadino seleziona la sede2. Il cittadino seleziona il servizio di ritiro3. Il cittadino visualizza gli slot a disposizione4. Il cittadino seleziona uno slot verde (libero) dopo i 30 giorni dalla precedente richiesta5. Si apre un popup di conferma6. Il cittadino conferma l'appuntamento
Postcondizioni: lo slot viene assegnato al cittadino e colorato di arancione nell'interfaccia

Il sequence diagram, condizioni escluse, corrisponde alla prenotazione di un rilascio.

Registrazione

Compilando un form, il cittadino può registrarsi sul portale, purché presente nell'anagrafica del sistema e non precedentemente registrato.

Attori: Cittadino
Precondizioni: presenza nell'anagrafica, prima registrazione
Passi: <ol style="list-style-type: none">1. Il cittadino visualizza il form di registrazione2. Il cittadino compila debitamente il form3. Il cittadino visualizza popup di conferma del codice fiscale calcolato dal sistema4. Il cittadino conferma la correttezza e termina la registrazione5. Il cittadino viene reindirizzato alla pagina di login
Postcondizioni: viene registrato il cittadino associandolo a una certa anagrafica

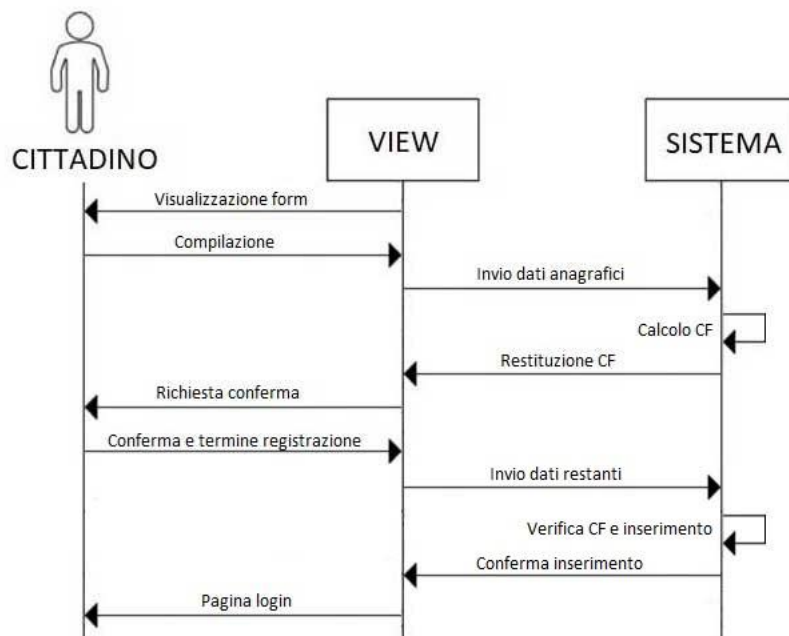


Figura 5: Sequence diagram registrazione cittadino

Diagrammi delle attività

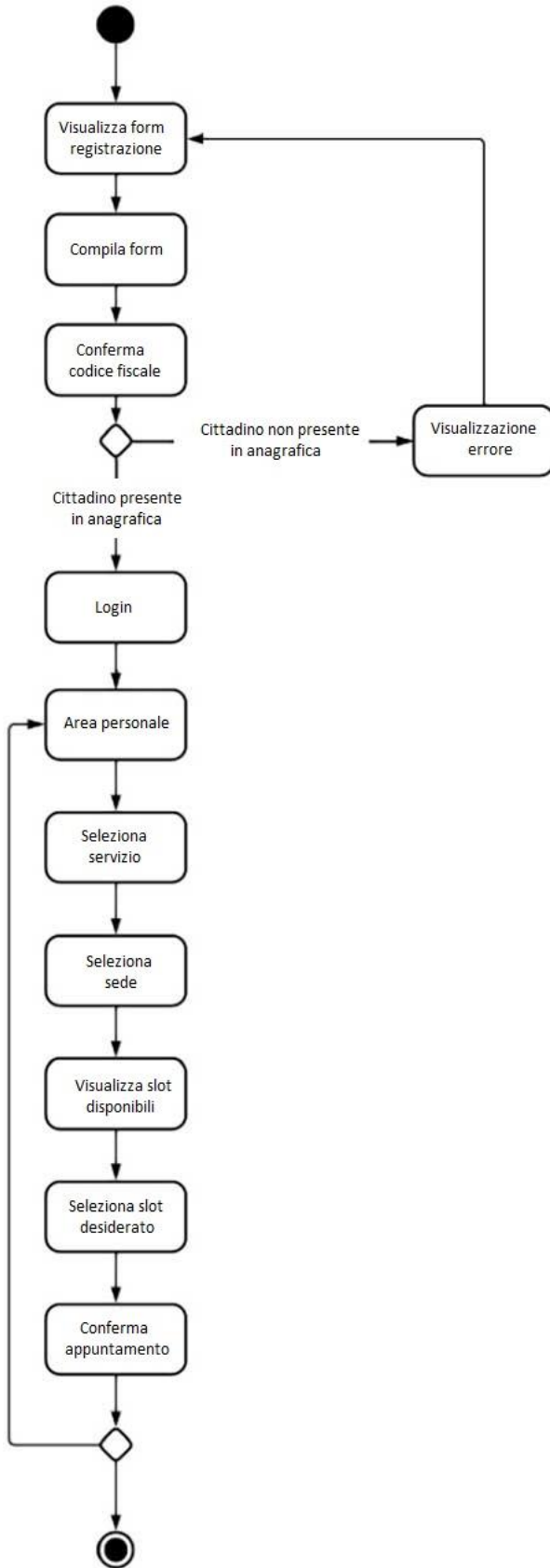


Figura 6: Activity diagram cittadino

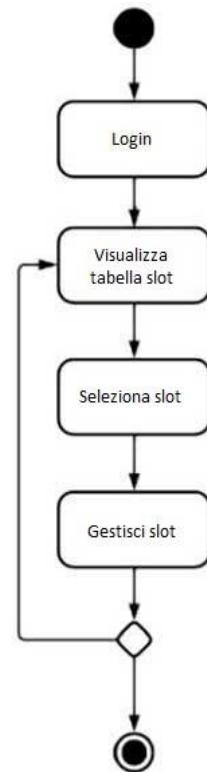


Figura 7: Activity diagram dipendente

Progettazione e implementazione del sistema

Processo di sviluppo

Il processo di sviluppo adottato per il progetto ha seguito un approccio agile. Le fasi di specifica, sviluppo e validazione sono state affrontate con l'obiettivo di alternare le attività in modo coeso rispetto all'implementazione del sistema. Questa metodologia ha consentito la creazione di molteplici versioni intermedie del sistema, ciascuna implementante un sottoinsieme delle specifiche. Il comportamento del programma è stato verificato dopo ogni modifica, anche se non sostanziale, garantendo un progresso sicuro verso le fasi successive.

La documentazione del sistema è stata redatta principalmente a progetto quasi terminato. Durante lo sviluppo del software sono stati creati diagrammi e schemi concettuali, non necessariamente rigorosi, per rappresentare il funzionamento generale del sistema e aiutare lo sviluppatore nell'implementazione.

Il flusso di sviluppo si è concentrato inizialmente sull'implementazione della business logic dell'applicazione cercando minuziosamente ogni possibile funzionamento non desiderato, dopodiché si è proseguito con l'interfaccia utente e la gestione delle interazioni con quest'ultimo, incluso lo scambio d'informazioni tra frontend e backend.

Progettazione e pattern architetturali usati

Il sistema è stato progettato utilizzando le tecniche di modellazione ad oggetti, seguendo un'architettura basata sul modello MVC (Model-View-Controller), implementato grazie al framework di Java, Spring, per il backend e alla libreria di JavaScript, React, per il frontend. Questa scelta architetturale ha permesso una netta suddivisione delle responsabilità e delle fasi di sviluppo delle componenti principali, semplificando il processo di sviluppo, verifica e revisione del codice.

- Model: responsabile della gestione dei dati e delle informazioni memorizzate. Definisce come i dati vengono strutturati e fornisce le funzionalità necessarie per accedere, modificare e gestire questi dati. (Spring service/entity/repository, MySQL)
- View: questo elemento del sistema si concentra sull'aspetto visuale e interattivo dell'interfaccia utente. (componenti React)
- Controller: i controller gestiscono la logica applicativa del sistema. gestiscono le richieste provenienti dagli utenti e di coordinare l'interazione tra il Model e la View. (Spring controller e React router)

Per mantenere il codice organizzato e facilmente leggibile, abbiamo assegnato un controller e un model specifico per ciascuna entità. Le azioni degli utenti sulla vista vengono inviate ai controller che poi aggiornano i dati e riflettono le modifiche nella vista.

L'approccio migliora sensibilmente manutenibilità e scalabilità del sistema.

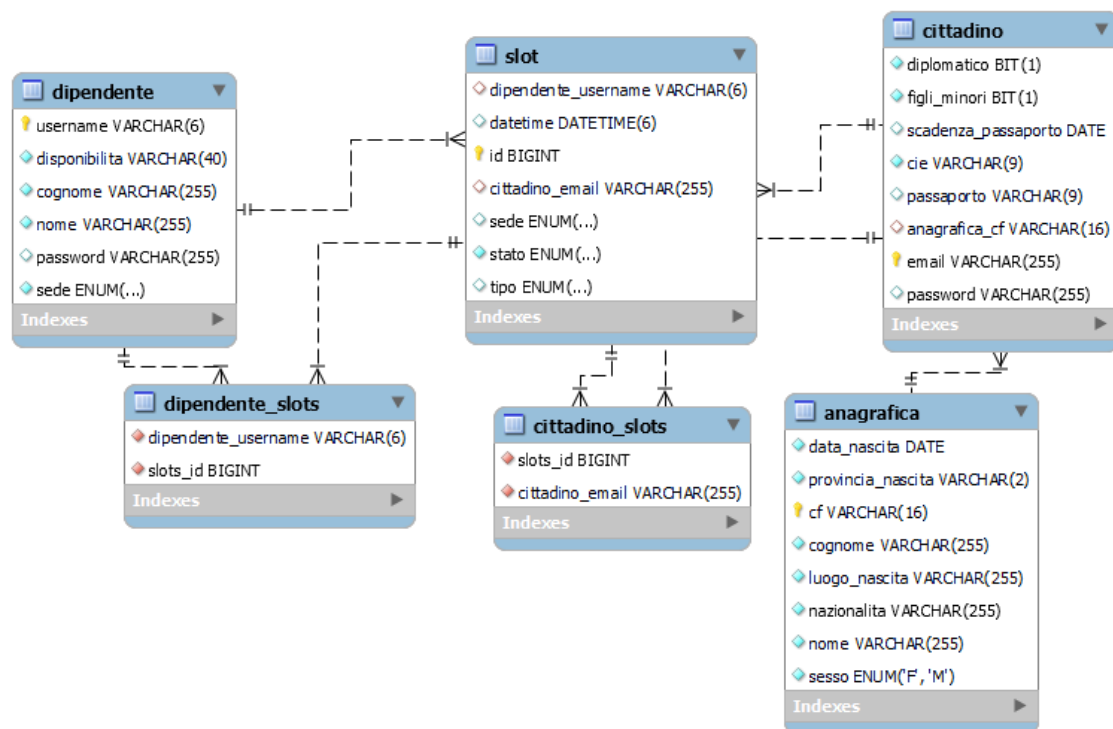


Figura 8: Diagramma E-R database

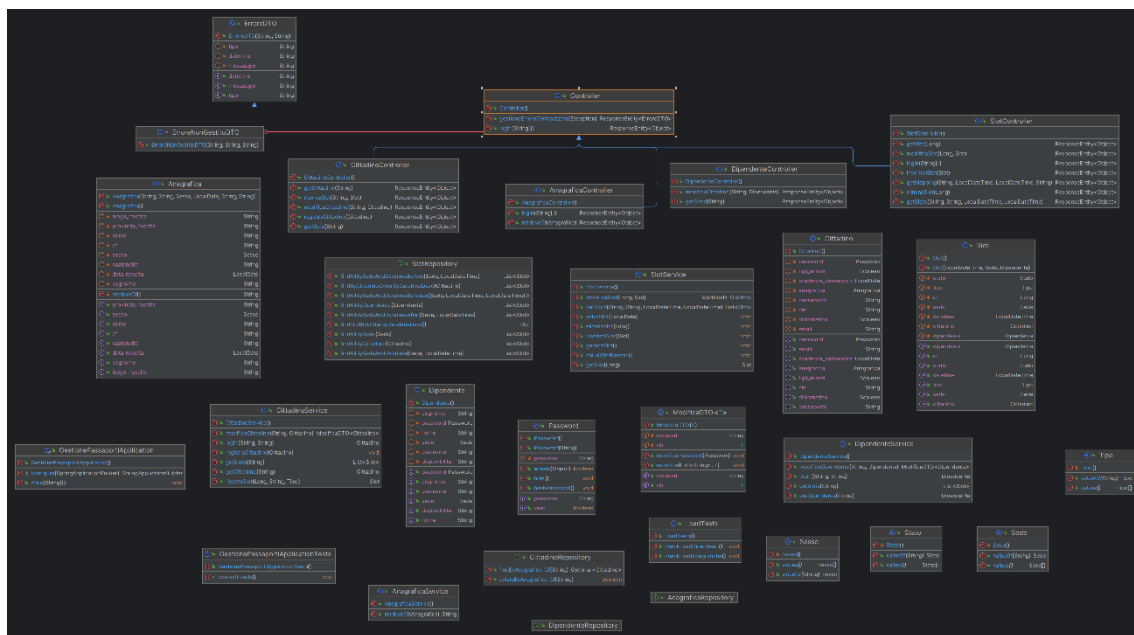


Figura 9: Diagramma delle classi (molti collegamenti non sono visibili perché "iniettati" da Spring)

Implementazione e design pattern usati

Per quanto riguarda l'interazione con l'utente, è stato implementato il pattern Observer attraverso i React State e le direttive OnClick.

Avendo bisogno, in certe circostanze, di semplificare il passaggio di dati tra le tre parti, è stato utilizzato il pattern DTO (Data Transfer Object) con l'implementazione di due classi: ErroreDTO, per restituire in caso di errore un oggetto che fornisca un'analisi completa, e ModificaDTO<T> che aiuta l'aggiornamento di un'oggetto eseguendo un controllo dei campi da rinnovare rispetto a una nuova versione.

Spring implementa il Singleton delle classi service e delle interfacce repository attraverso l'annotazione @Autowired e a livello sottostante, tramite Hibernate, mantiene una sorta di architettura a repository per la gestione dei dati con il model.

L'utilizzo di framework e librerie in generale garantisce una serie di funzionalità predisposte affinché il programmatore concentri gli sforzi sull'effettiva logica del programma.

Attività di test e validazione

Ispezione codice e documentazione

Durante questa fase, il documento contenente le specifiche è stato sottoposto a una serie di revisioni approfondite al fine di garantire la correttezza e la coerenza rispetto alle specifiche fornite. Una volta completata quest'attività di revisione, è stato effettuato un rigoroso controllo del codice sorgente per assicurarsi dell'assenza di errori e per mantenere il codice facilmente leggibile e comprensibile. Questo processo ha contribuito a garantire l'integrità e la qualità del sistema, oltre a favorire una migliore comprensione delle logiche di progettazione.

Alpha testing

In questa fase, lo sviluppatore ha condotto una serie di test per valutare il comportamento del sistema con input diversificati. L'obiettivo era verificare se il software rispondeva in modo accurato a ogni richiesta, compresi i casi limite. A seguire alcuni dei test eseguiti:

- Verifica della registrazione: compilazione del form di registrazione con valori corrispondenti all'anagrafica inserita per controllare la correttezza della registrazione e inserimento di valori fittizi per assicurarsi che si generi un errore.
- Accesso utente: tentativi di accesso con username/cf e password vari per controllare che le verifiche dell'esistenza dell'utente e della cifratura della password siano corrette.
- Utente già in sessione: nel caso in cui la sessione del browser contenga già un utente dello stesso tipo per cui si chiede il login, si viene reindirizzati automaticamente alla pagina successiva.
- Concorrenza: controllo aggiornamento delle tabelle di disponibilità quando più utenti eseguono operazioni nello stesso momento.
- Gestione slot: aggiornamento del database e dell'interfaccia quando un dipendente applica modifiche allo slot
- Prenotazione slot: verifica che il sistema impedisca la prenotazione nel caso in cui non siano rispettate le condizioni dettate dal tipo.
- Postman: utilizzando il programma Postman, sono state simulate le diverse HTTP request che il backend deve essere in grado di gestire.
- MySQL Workbench: strumento che facilita il controllo delle query eseguite dal programma visualizzando e operando direttamente sul database.

In sintesi, questa fase è stata fondamentale per garantire che il software sia robusto e affidabile.

Beta testing

Come fase finale di testing, il software è stato sottoposto a una prova condotta da utenti con limitate competenze informatiche e senza alcuna conoscenza del processo di sviluppo del software stesso. Durante questa fase, agli utenti è stato brevemente illustrato lo scopo generale del software, ma non sono stati forniti dettagli o istruzioni specifiche sul suo utilizzo. Gli utenti sono stati lasciati liberi di esplorare il software senza essere guidati in alcun modo, e non sono stati informati riguardo ai casi limite o alle situazioni di errore del sistema.

Questo tipo di test ha consentito di identificare errori e problemi che potrebbero non essere stati evidenti per gli sviluppatori, in quanto si basavano sull'esperienza degli utenti reali. Inoltre, ha aiutato a individuare nuove potenziali funzionalità o miglioramenti che il sistema avrebbe potuto implementare per migliorare l'usabilità e la soddisfazione degli utenti. In sostanza, questo test ha contribuito a migliorare la qualità complessiva del software e a renderlo più adatto alle esigenze degli utenti finali.