



MedQueue

RAD Requirement Analysis Document

MedQueue

Sommario

1. Introduzione

- 1.1 Scopo del sistema
- 1.2 Ambito del sistema
- 1.3 Obiettivi e criteri di successo del sistema
- 1.4 Definizione, acronimi e abbreviazioni
- 1.5 Panoramica

2. Sistema corrente

- 2.1 Diagramma delle attività

3. Sistema proposto

- 3.1 Diagramma delle attività
 - 3.1.1 Invio, conferma, accettazione prenotazione
- 3.2 Requisiti funzionali
 - 3.2.1 Requisiti per la visualizzazione code
 - 3.2.2 Requisiti per la gestione delle prenotazioni
 - 3.2.3 Requisiti per la gestione delle prenotazioni svolte dall'impiegato
 - 3.2.4 Requisiti per la gestione dell'utente
 - 3.2.5 Requisiti per la gestione dell'impiegato
- 3.3 Requisiti non funzionali
 - 3.3.1 Usabilità
 - 3.3.2 Affidabilità
 - 3.3.3 Prestazioni
 - 3.3.4 Sostenibilità
- 3.4 Pseudo Requisiti
 - 3.4.1 Implementazione
- 3.5 Modello di sistema
 - 3.5.1 Scenari
 - 3.5.2 Use Case
 - 3.5.2.1 UC_VC-1: Visualizzazione Coda Presenza

- 3.5.2.2 [UC_GP-1: Richiesta Prenotazione](#)
- 3.5.2.3 [UC_GP-3: Validazione Prenotazione](#)
- 3.5.2.4 [UC_GPI-1: Accettazione Prenotazioni](#)
- 3.5.2.5 [UC_GP-2: Eliminazione Prenotazione](#)
- 3.5.3 [Modello ad Oggetti](#)
 - 3.5.3.1 [Entità di dominio](#)
 - 3.5.3.2 [OM_VC-1: Visualizzazione Coda Presenza](#)
 - 3.5.3.3 [OM_GP-1: Richiesta Prenotazione](#)
 - 3.5.3.4 [OM_GP-2: Eliminazione Prenotazione](#)
 - 3.5.3.5 [OM_GP-3: Validazione Prenotazione](#)
 - 3.5.3.6 [OM_GPI-1: Accettazione Prenotazione](#)
 - 3.5.3.7 [OM_GP-4: Visualizzazione Prenotazione](#)
 - 3.5.3.8 [OM_GU-1: Autenticazione](#)
 - 3.5.3.9 [OM_GU-2: Registrazione Account](#)
 - 3.5.3.10 [OM_GU-3: Logout](#)
- 3.5.4 [Modello dinamico](#)
 - 3.5.4.1 [SD_VC-1: Visualizzazione Coda Presenza](#)
 - 3.5.4.2 [SD_GP-1: Richiesta Prenotazione](#)
 - 3.5.4.3 [SD_GP-2: Eliminazione Prenotazione](#)
 - 3.5.4.4 [SD_GP-3: Validazione Prenotazione](#)
 - 3.5.4.5 [SD_GPI-1: Accettazione Prenotazione](#)
 - 3.5.4.6 [SD_GP-4: Visualizzazione Prenotazione](#)
 - 3.5.4.7 [SD_GU-1: Autenticazione](#)
 - 3.5.4.8 [SD_GU-2: Registrazione Account](#)
 - 3.5.4.9 [SD_GU-3: Logout](#)
- 3.5.5 [Statechart Diagrams](#)
 - 3.5.5.1 [SCD_GP-3: Validazione Prenotazione](#)
 - 3.5.5.2 [SCD_GP1: Richiesta Prenotazione](#)
- 3.5.6 [Interfaccia utente e mock-ups](#)

- 3.5.6.1 [UI_1: Homepage](#)
- 3.5.6.2 [UI_2: Registrazione](#)
- 3.5.6.3 [UI_3: Login](#)
- 3.5.6.4 [UI_4: Area Personale](#)
- 3.5.6.5 [UI_5: Richiesta Prenotazione 1](#)
- 3.5.6.6 [UI_6: Richiesta Prenotazione 2](#)
- 3.5.6.7 [UI_7: Visualizza Prenotazioni 1](#)
- 3.5.6.8 [UI_8: Visualizza Prenotazioni 2](#)
- 3.5.6.9 [UI_9: Visualizza coda](#)
- 3.5.6.10 [UI_10: Elimina Prenotazione 1](#)
- 3.5.6.11 [UI_11: Elimina Prenotazione 2](#)
- 3.5.6.12 [UI_12: Convalida](#)
- 3.5.6.13 [UI_13: Accettazione Prenotazione](#)

3.6 [Matrice di tracciabilità](#)

4. [Glossario](#)

Data	Versione	Cambiamenti	Autore
03/11/2020	v0.1	Stesura introduzione del RAD	Andrea Fucile
11/11/2020	v0.2	Inserimento dei primi Use Case	Andrea Fucile
28/11/2020	v1.0	Aggiunti Sequence Diagram, Class Diagram, ulteriori Use Case	[Tutti]
30/11/2020	v1.1	Revisionati Sequence Diagram e Class Diagram	Giovanni Rapa
02/12/2020	V1.1.1	Revisione finale e messa a punto della spaziatura	Adriano Amato

1. Introduzione

1.1 Scopo del sistema

Il sistema che s'intende realizzare nasce dall'idea di poter fornire un servizio veloce e sicuro per effettuare una prenotazione presso un ufficio ospedaliero, soprattutto in questo periodo in cui il COVID-19 sta cambiando le nostre abitudini e le regole sociali. Il sistema è separato in due piattaforme separate: un sito web per quanto riguarda la gestione delle prenotazioni e tutto ciò che è inerente ai pazienti; un software che permette agli impiegati la gestione delle code.

MedQueue vuole offrire aiuto a tutti coloro che sono implicati nell'effettuazione di una prenotazione presso una struttura sanitaria: per i pazienti offre una piattaforma online su cui poter prenotare un appuntamento presso un ufficio ospedaliero convenzionato nel giorno desiderato, sempre che in quel giorno ci sia disponibilità, e quindi potersi presentare il giorno stesso presso l'ufficio e quindi riducendo drasticamente le code e gli assembramenti; per quanto riguarda gli impiegati, MedQueue mette a disposizione un software di gestione delle diverse code, in modo da gestire nel modo più corretto l'affluenza agli sportelli, senza doversi preoccupare di eventuali priorità gestite in automatico dal sistema.

Inoltre MedQueue mette a disposizione anche un sistema di conferma della prenotazione direttamente presso l'ufficio; infatti il cliente prenotato dovrà, una volta arrivato presso l'ufficio selezionato, confermare la propria presenza presso un "totem" che riconoscerà il cliente attraverso il suo codice fiscale e, se la prenotazione è prevista entro 30 minuti, verrà messo direttamente in coda altrimenti visualizzerà una notifica che lo inviterà a tornare più tardi per evitare assembramenti nell'ufficio.

1.2 Ambito del Sistema

Il nostro sistema mira ad offrire un software che faciliti la fruizione del sistema ospedaliero in modo facile e innovativo.

Il sistema ovviamente non entrerà nel merito puramente medico, infatti non è ideato per prescrivere farmaci o per fare diagnosi, ma piuttosto gestisce tutta la parte che precede una visita medica, dalla prenotazione fino all'accettazione presso l'ufficio. Inoltre non si occupa neanche di prenotare visite, ma piuttosto serve per prenotare un posto presso un ufficio ospedaliero.

Il sistema non sarà collegato a nessuno dei sistemi già presenti presso le varie strutture ospedaliere per motivi legali.

1.3 Obiettivi e criteri di successo del sistema

I software che presentiamo partono dall'idea che per rendere migliore l'esperienza del cliente presso una struttura ospedaliera sia indispensabile diminuire i tempi di attesa almeno per quanto riguarda gli iter precedenti alla fruizione effettiva del servizio medico.

La nostra proposta, quindi, si basa su due principi fondamentali:

- Migliorare la gestione delle prenotazioni e renderle più veloci.
- Limitare al minimo il numero di persone presenti contemporaneamente in un ufficio per evitare assembramenti.

1.4 Definizioni, acronimi e abbreviazioni

Totem: Il totem è una struttura fisica presente in un ufficio ospedaliero composto da un dispositivo con installato il software necessario, nel nostro caso con un accesso a internet e con la pagina necessaria aperta.

Coda di prenotazione: La coda di prenotazione è la coda in cui i clienti prenotati vengono posizionati prima di effettuare la conferma della prenotazione.

Coda di presenza: Una volta che il cliente prenotato ha confermato la sua presenza nell'ufficio ospedaliero, il sistema lo posiziona automaticamente nella coda di presenza.

1.5 Panoramica

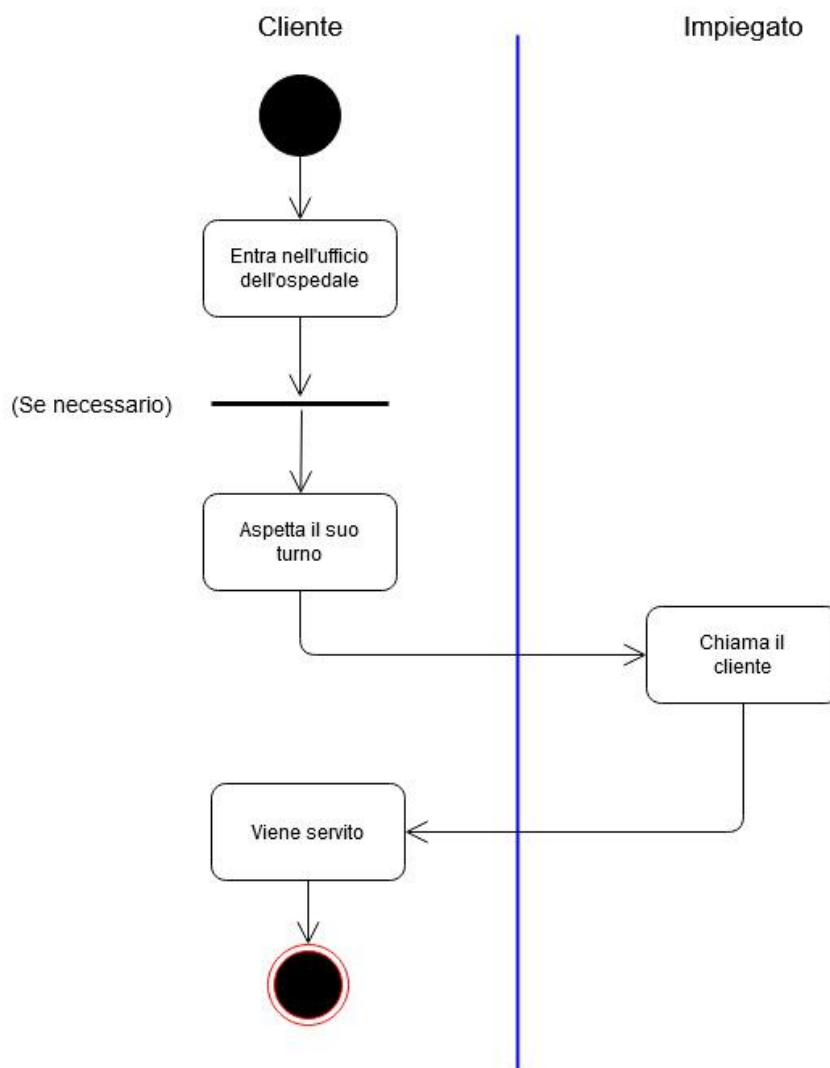
Al secondo punto verrà presentato il sistema proposto con i relativi requisiti funzionali e non funzionali. Attraverso scenari e use case verrà stabilito chi sono gli attori del sistema e come gli attori individuati interagiscono con il sistema. Verranno poi presentati mock-up dell'interfaccia utente ed i path navigazionali per l'applicazione proposta.

2. Sistema corrente

Attualmente non esiste un sistema software che si occupa della gestione prenotazioni di un ufficio ospedaliero. La procedura, infatti, prevede che la persona si presenti direttamente all'ufficio ospedaliero e attenda che venga chiamato dall'impiegato. Questa pratica risulta essere molto onerosa (in termini di tempo) ai danni delle persone, che tal volta rinunciano ad essere serviti per la molta attesa.

2.1 Diagramma delle attività

Abbiamo ritenuto opportuno sviluppare un activity diagram relativo alle operazioni svolte per la gestione dei servizi ospedalieri per meglio percepire la differenza tra quello che è il sistema che proponiamo e quello che è il sistema attuale.



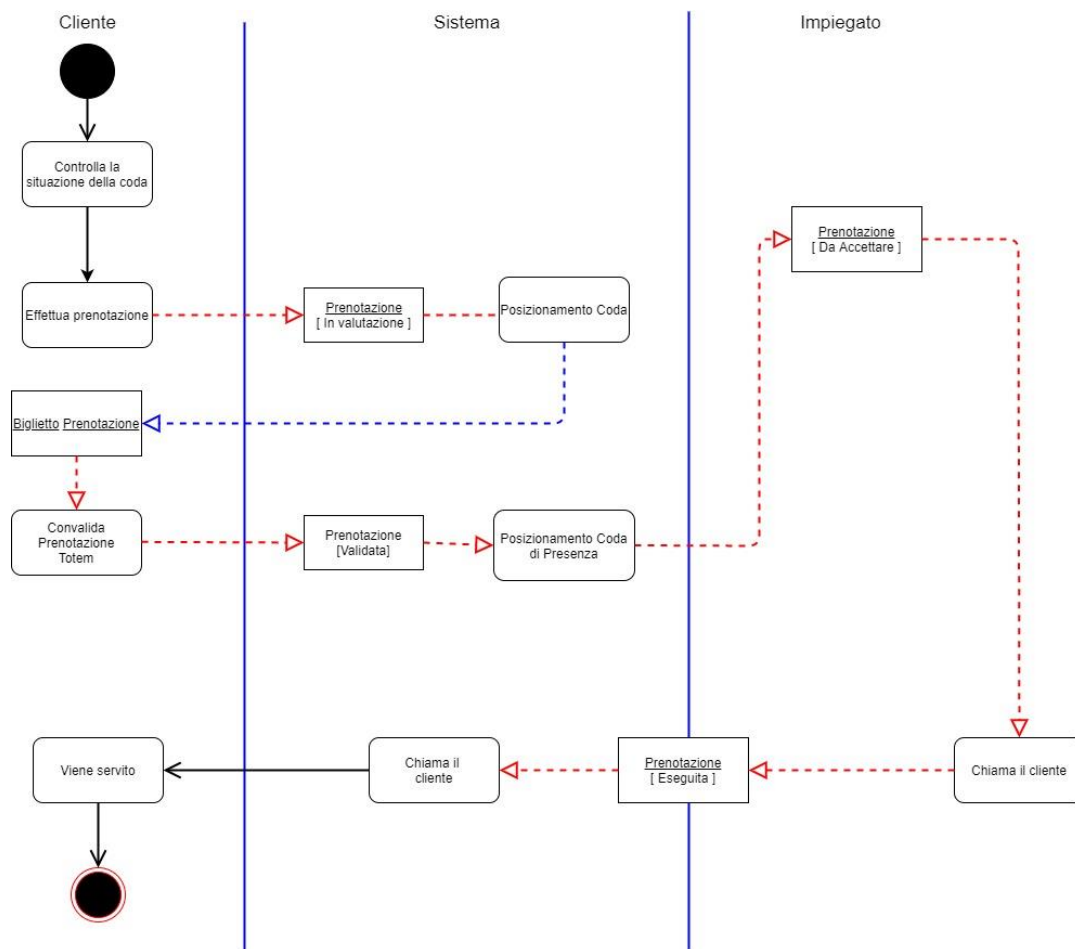
3. Sistema Proposto

3.1 Diagramma delle attività

Al fine di poter cogliere al meglio gli effetti del sistema che proponiamo sulla gestione delle prenotazioni che precedono una visita medica presentiamo l'activity diagram relativo alla funzionalità principale messa a disposizione della piattaforma.

Descriviamo come viene effettuata, gestita ed accettata una prenotazione.

3.1.1 Invio, conferma, accettazione prenotazione



3.2 Requisiti funzionali

3.2.1 Requisiti per la visualizzazione code

- RF-VC-1: Visualizzazione Coda Prenotazioni
Il sistema dovrà offrire la possibilità all'utente di visualizzare la coda delle prenotazioni

3.2.2 Requisiti per la gestione delle prenotazioni

- RF-GP-1: Richiesta prenotazione
Da utente che utilizza la piattaforma posso prenotare l'accesso presso l'ufficio convenzionato che preferisco in modo da organizzare bene tutti i miei impegni.
- RF-GP-2: Eliminazione prenotazione
Il sistema dovrà fornire all'utente la possibilità di eliminare una prenotazione effettuata
- RF-GP-3 Validazione prenotazione
Da utente che ha effettuato una prenotazione vorrei un modo sicuro di confermare la mia presenza nell'ufficio scelto così da avere la certezza della mia posizione in coda.
- RF-GP-4: Visualizzazione prenotazione
Come utente ho a disposizione un modo per tenere d'occhio tutte le mie prenotazioni in modo da non perderle mai di vista.

3.2.3 Requisiti per la gestione delle prenotazioni svolte dall'impiegato

- RF-GPI-1: Accettazione prenotazione
Come impiegato di un ufficio vorrei usufruire di un modo veloce per servire i clienti per ottimizzare il mio lavoro.

3.2.4 Requisiti per la gestione dell'utente

- RF-GU-1: Autenticazione
Il sistema dovrà fornire all'ospite la possibilità di autenticarsi
- RF-GU-2: Registrazione account
Il sistema dovrà fornire all'utente la possibilità di registrarsi
- RF-GU-3: Logout
Il Sistema dovrà fornire agli utenti registrati che si sono autenticati la possibilità di effettuare il Logout

3.2.5 Requisiti per la gestione dell'impiegato

- RF-GI-1: Autenticazione
Il sistema dovrà fornire all'impiegato la possibilità di autenticarsi
- RF-GI-3: Logout
Il Sistema dovrà fornire agli impiegati che si sono autenticati la possibilità di effettuare il Logout

3.3 Requisiti non funzionali

3.3.1 Usabilità

- RNF-U1: Il sistema deve essere facile da apprendere ed intuitivo da utilizzare senza necessariamente consultare la documentazione
- RNF-U2: Il sistema deve essere accessibile sia da dispositivi desktop che mobile

3.3.2 Affidabilità

- RNF-A1: Il sistema deve garantire la consistenza dei dati
- RNF-A2: Il sistema dovrà fornire un metodo di autenticazione sicuro in modo che i dati gestiti siano protetti da accessi non autorizzati
- RNF-A3: Il sistema deve garantire il filtraggio degli input dell'utente ed eventualmente richiedere a quest'ultimo di reinserire le informazioni richieste in caso di rilevamento di inconsistenza

3.3.3 Prestazioni

- RNF-P1: Il sistema deve elaborare le richieste o produrre output in meno di 2 secondi
- RNF-P2: Il sistema deve permettere l'interazione contemporanea di almeno 100 utenti diversi

3.3.4 Sostenibilità

- RNF-S1: Il sistema prodotto dev'essere facilmente comprensibile da sviluppatori terzi che vorranno estendere le funzionalità o risolverne i problemi
- RNF-S2: Il sistema prodotto deve offrire una buona modificabilità
- RNF-S3: Il sistema prodotto deve offrire buona estensibilità

3.4 Pseudo requisiti

3.4.1 Implementazione

- RNF-I1: Il sistema deve reperire i dati da un database relazionale
- RNF-I2: Il sistema sarà sviluppato in linguaggio Java

3.5 Modello di Sistema

3.5.1 Scenari

Nome Scenario	SC_GP-1: Richiesta prenotazione
Attori Principali	Angelo: Utente Registrato
Flusso di eventi	<ol style="list-style-type: none">1. Angelo si connette al sistema con l'intenzione di voler prendere una prenotazione, esegue l'autenticazione e clicca sul pulsante richiedi prenotazione.2. Il sistema chiede ad Angelo di selezionare l'operazione che vuole effettuare.3. Angelo seleziona "Ritira Analisi".4. Il sistema mostra ad Angelo tutti gli ospedali convenzionati.5. Angelo seleziona "Ospedale di Caserta".6. Il sistema mostra ad Angelo tutti gli ambulatori presenti nell'ospedale di Caserta e un calendario con i giorni in cui è possibile effettuare una prenotazione.7. Angelo sceglie di effettuare una prenotazione per il giorno 12 Dicembre e, dalla lista degli orari disponibili, seleziona le 11:00.8. Il sistema inserirà Angelo nella coda di prenotazione del giorno 12 Dicembre e restituirà ad Angelo una mail di conferma prenotazione

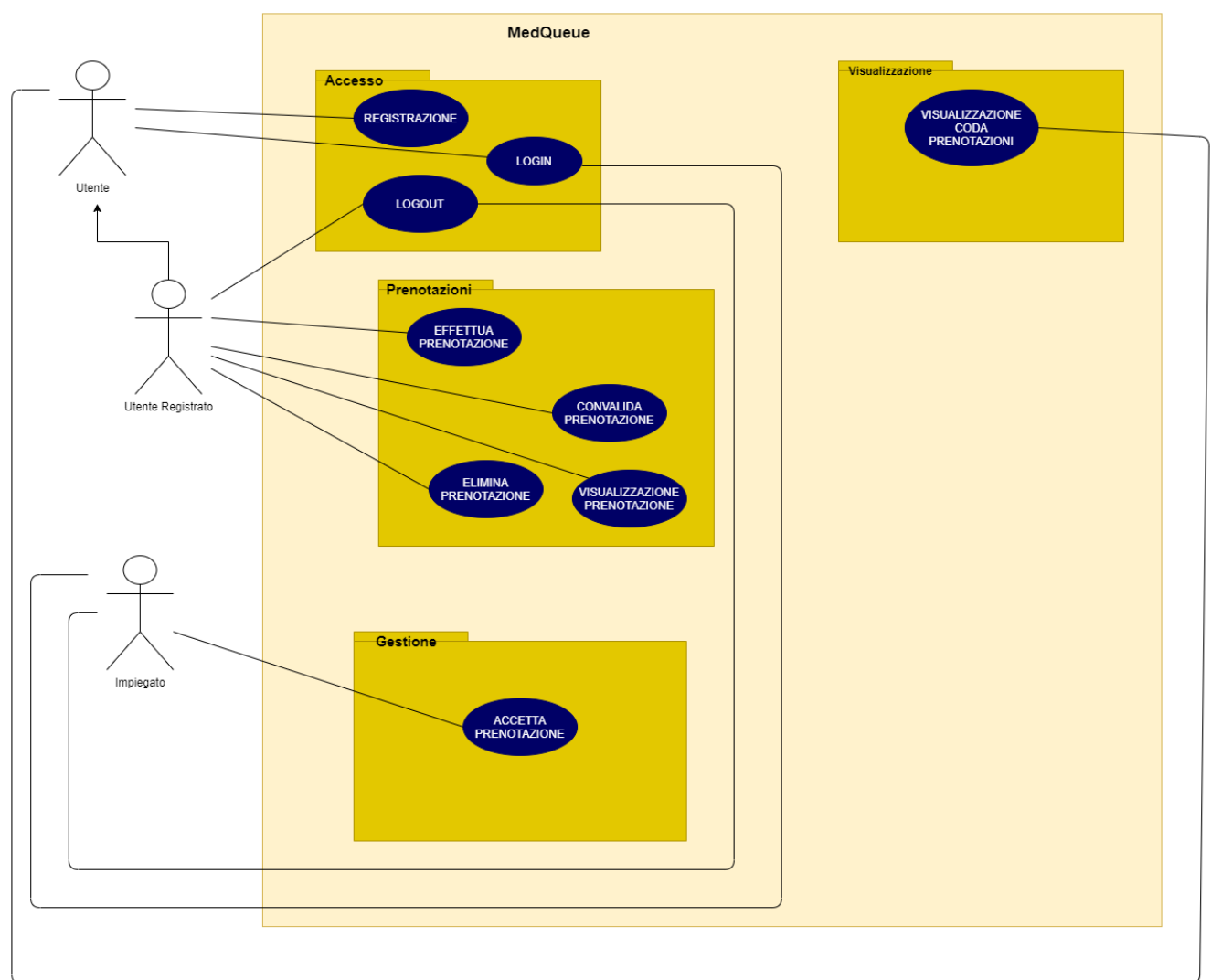
Nome Scenario	SC_GP-2: Eliminazione prenotazione
Attori Principali	Angelo: Utente Registrato
Flusso di eventi	<ol style="list-style-type: none">1. Angelo accede alla sua area personale con l'intenzione di voler eliminare una prenotazione presa, e clicca su visualizza prenotazioni2. Il sistema mostrerà ad Angelo tutte le prenotazioni ancora in attesa di convalida3. Angelo seleziona la prenotazione da voler eliminare e clicca su elimina4. Il sistema mostra ad Angelo un messaggio di conferma.5. Angelo conferma la sua intenzione di voler eliminare la prenotazione6. Il sistema elimina la prenotazione cancellandola anche dalla coda di prenotazioni e mostra ad angelo un messaggio "Prenotazione Eliminata"

Nome Scenario	SC_GP-3: Validazione prenotazione
Attori Principali	Angelo: Utente Registrato
Flusso di eventi	<ol style="list-style-type: none"> 1. Angelo il giorno 12 Dicembre si reca all'ufficio ospedaliero poiché possiede una prenotazione per quel giorno. 2. Angelo si avvicina al totem per convalidare la prenotazione usando il proprio codice fiscale 3. Il sistema conferma la prenotazione di Angelo e lo inserisce nella coda di presenza

Nome Scenario	SC_VC-1: Visualizzazione coda prenotazioni
Attori Principali	Adriano: Utente Generico
Flusso di eventi	<ol style="list-style-type: none"> 1. Adriano si connette al sistema per visualizzare la coda; entra nella propria area personale e clicca sull'icona relativa alla visualizzazione della coda. 2. Il sistema mostra ad Adriano l'elenco delle strutture disponibili. 3. Adriano seleziona l'ospedale di Napoli. 4. Il sistema Mostra ad Adriano tutte le prenotazioni presso quell'ufficio in quella data.

Nome Scenario	SC_GPI-1: Accettazione prenotazione
Attori Principali	Giovanni: Impiegato ufficio ospedaliero
Flusso di eventi	<ol style="list-style-type: none"> 1. Giovanni inizia la propria giornata lavorativa avviando l'applicazione MedQueue 2. Il sistema chiede a Giovanni di autenticarsi 3. Giovanni procede con l'autenticazione inserendo le proprie credenziali 4. Il sistema mostra a Giovanni quali tipi di prenotazioni accettare 5. Giovanni seleziona "Prenota Visita" 6. Il sistema preleva il primo della lista e lo chiama allo sportello.

3.5.2 Use Case



3.5.2.1 UC_VC-1: Visualizzazione Coda

Identificativo <i>UC_VC-1</i>		<i>Visualizzazione Coda Prenotazioni</i>	<i>Data</i>	<i>03/12/20</i>
			<i>Vers.</i>	<i>0.00.005</i>
			<i>Autore</i>	<i>Afeltra Angelo</i>
Descrizione		<i>Lo UC fornisce la funzionalità per visualizzare la coda dell'ufficio</i>		
Attore Principale		Utente		
Attori secondari		NA		
Entry Condition		NA		
Exit condition On success		L'utente visualizza la schermata della coda di prenotazioni		
Exit condition On failure		L'utente visualizza la schermata d'errore		
Rilevanza/User Priority		Elevata		
Frequenza stimata		150/giorno		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Attore:	L'utente decide di voler vedere la coda, quindi clicca sul pulsante visualizza coda di prenotazioni		
2	Sistema:	Il sistema carica una nuova pagina dove chiede all'utente di inserire la struttura per la quale visualizzare la coda.		
3	Attore:	L'Utente seleziona la struttura		
4	Sistema	Il sistema aggiorna la pagina mostrando la coda		
I Scenario/Flusso di eventi Alternativo: L'utente è Autenticato e ha una prenotazione nel giorno stesso				
2.1	Sistema:	Il sistema mostra la coda dell'ufficio presso cui l'utente ha la prenotazione		
I Scenario/Flusso di eventi Alternativo: Viene conclusa un'operazione durante la visualizzazione della coda				
2.2	Sistema:	Il sistema effettua il refresh automatico della pagina per visualizzare l'elenco aggiornato		
Il Scenario/Flusso di eventi Alternativo: Non è presente alcuna prenotazione per quella giornata				
2.3	Sistema:	Il sistema mostra una schermata senza tabella ma con una scritta "ASSENZA DI PRENOTAZIONI"		

3.5.2.2 UC_GP-1: Richiesta Prenotazione

Identificativo UC_GP-1	Richiesta Prenotazione	Data	26/11/2020
		Vers.	0.00.002
		Autore	Fucile Andrea
Descrizione	Lo UC fornisce le funzionalità per la prenotazione nell'ufficio ospedaliero		
Attore Principale	Utente		
Attori secondari	NA		
Entry Condition	L'utente deve essere registrato alla piattaforma e deve essere autenticato. L'utente non deve avere un'altra prenotazione nell'orario scelto e non deve avere altre prenotazioni nella coda selezionata.		
Exit condition On success	L'utente effettua con successo la prenotazione.		
Exit condition On failure	La prenotazione non viene registrata e all'utente viene segnalato un errore.		
Rilevanza/User Priority	Elevata		
Frequenza stimata	200 usi/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Attore:	L'utente, dopo essere entrato nella sua area personale, procede con l'effettuazione della prenotazione.	
2	Sistema:	Il Sistema mostra all'utente i tipi di prenotazioni disponibili.	
3	Attore:	L'utente sceglie il tipo di prenotazione che desidera effettuare compilando un form.	
4	Sistema:	Il Sistema mostra in base al tipo di prenotazione richiesta un elenco di strutture che eseguono quell'operazione.	
5	Attore:	L'utente seleziona la struttura preferita.	
6	Sistema:	Il sistema mostra all'utente, tramite un calendario, giorni e orari disponibili per poter prenotarsi.	
7	Attore:	L'utente sceglie data e ora e conferma la prenotazione.	
8	Sistema:	Il sistema aggiunge la prenotazione alla coda delle prenotazioni ed invia una mail di avvenuta prenotazione all'utente.	
I Scenario/Flusso di eventi di ERRORE: L'orario selezionato è stato già scelto ma la lista orari non era aggiornata			
4.1	Sistema:	Dopo che l'utente conferma la prenotazione, il Sistema controlla che non ci sia nessun'altra prenotazione con data e ora corrispondente, in caso contrario, manda un messaggio di errore all'utente segnalando che l'orario non è disponibile.	
I Scenario/Flusso di eventi alternativi: Viene effettuata la prenotazione dopo l'orario di chiusura			
2.1	Sistema:	Il Sistema mostra a Giovanni tutti i giorni disponibili per la prenotazione a partire dal giorno dopo.	

3.5.2.3 UC_GP-3: Validazione Prenotazione

Identificativo <i>UC_GP-3</i>	<i>Validazione Prenotazione</i>	<i>Data</i>	<i>01/12/2020</i>
		<i>Vers.</i>	<i>0.00.003</i>
		<i>Autore</i>	<i>Amato Adriano</i>
Descrizione	<i>Lo UC fornisce le funzionalità per la validazione della propria prenotazione</i>		
Attore Principale	Utente		
Attori secondari			
Entry Condition	L'utente ha prenotato con successo per quel determinato giorno.		
Exit condition On success	L'utente viene inserito nella coda di presenza per poi essere servito.		
Exit condition On failure	Il sistema informa l'utente che non ha una prenotazione valida.		
Rilevanza/User Priority	Alta		
Frequenza stimata	120/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Attore:	L'utente arriva all'ufficio, si avvicina al totem ed inserisce il proprio codice fiscale.	
2	Sistema:	Il sistema individua l'utente tramite il suo codice fiscale e controlla se ha una prenotazione per quel giorno, in caso sia prenotato lo aggiunge alla coda di presenza gestita da RabbitMQ e glielo comunica tramite display.	
3	Attore:	L'attore attende il suo turno.	
I Scenario/Flusso di eventi Alternativo: Misure anti-covid			
2.1	Sistema:	Il sistema, dopo aver individuato l'utente, controlla sia il giorno che l'ora della prenotazione. Se essa rientra in un intervallo di mezz'ora prima dell'ora corrente allora l'utente viene piazzato in coda, altrimenti viene mostrato un messaggio a video che invita l'utente a ritornare più tardi per rispettare le politiche di distanziamento sociale dovute al Covid.	
Il Scenario/Flusso di eventi Alternativo: Prenotazione in un altro giorno			
2.1	Sistema:	Il sistema cerca l'utente tramite il codice fiscale e vede che non ha una prenotazione attiva per oggi, mostra a video la data e l'ora dell'effettiva prenotazione.	
I Scenario/Flusso di eventi di ERRORE: Inserimento errato codice fiscale			
2.1	Sistema:	Il sistema controlla il codice fiscale e non trova nulla/verifica che è errato quindi informa l'utente tramite video dell'errore.	

3.5.2.4 UC_GPI-1: Accettazione Prenotazione

Identificativo <i>UC_GPI-1</i>	<i>Accettazione prenotazione</i>	<i>Data</i>	<i>01/12/2020</i>
		<i>Vers.</i>	<i>0.00.004</i>
		<i>Autore</i>	<i>Rapa Giovanni</i>
Descrizione	<i>Lo UC fornisce la funzionalità di accettazione delle prenotazioni</i>		
Attore Principale	Impiegato		
Attori secondari	NA		
Entry Condition	L'impiegato si è autenticato ed ha scelto il tipo o i tipi di prenotazione da gestire C'è almeno una prenotazione		
Exit condition On success	Viene chiamata e servita una prenotazione		
Exit condition On failure	Non viene servito nessun cliente		
Rilevanza/User Priority	Elevata		
Frequenza stimata	180 usi/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Impiegato:	L'impiegato accede alla pagina di visualizzazione delle accettazioni.	
2	Sistema:	Il sistema mostra le possibili code gestite da RabbitMQ tra cui può scegliere, in base al tipo di operazione che i clienti vogliono effettuare.	
3	Impiegato:	L'impiegato seleziona il tipo di coda da cui vuole prendere il prossimo cliente.	
4	Sistema	Il sistema preleva dalla coda la prenotazione, la elimina e stampa a video il nome ed il cognome della persona da servire.	
5	Impiegato	L'impiegato serve il cliente.	
Scenario/Flusso di eventi di ERRORE: il cliente cancella la prenotazione poco prima di essere servito			
2.b1	Sistema:	Mostra un messaggio d'errore per cliente assente.	
2.b2	Sistema:	Rimane in attesa di una nuova selezione.	

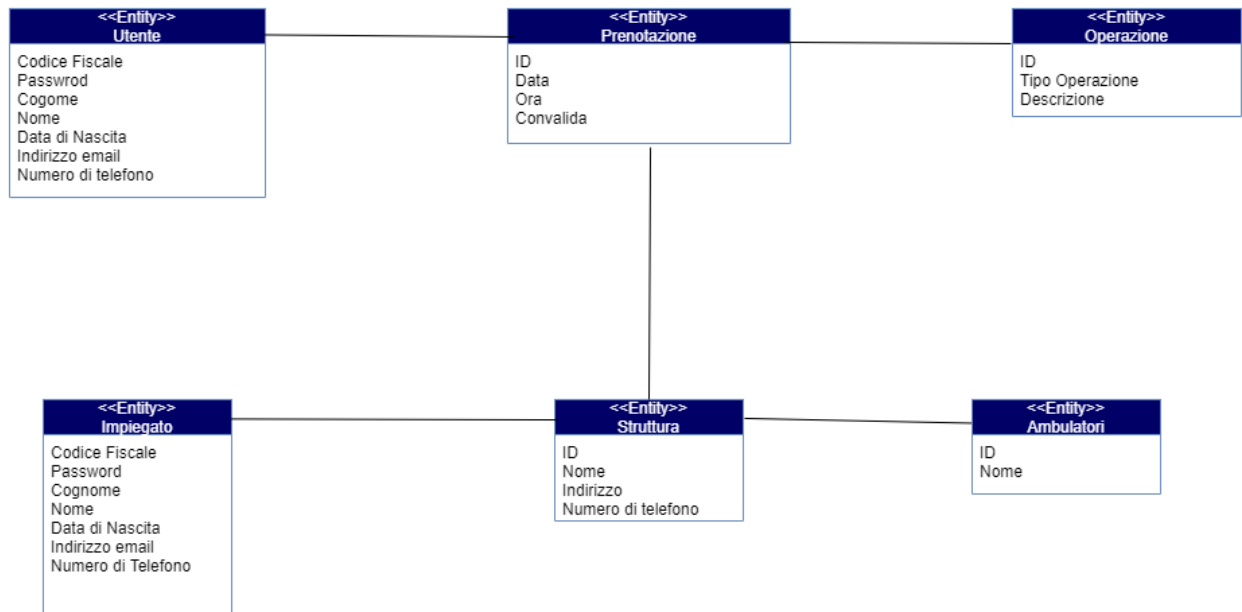
3.5.2.5 UC_GP-2: Eliminazione Prenotazione

Identificativo UC_GP-2	Eliminazione Prenotazione	Data	28/11/2020
		Vers.	0.00.002
		Autore	Amato Adriano
Descrizione	Lo UC fornisce le funzionalità per annullare una prenotazione precedentemente effettuata		
Attore Principale	Utente		
Attori secondari	NA		
Entry Condition	L'utente deve aver effettuato precedentemente una prenotazione		
Exit condition On success	La prenotazione selezionata dall'utente viene eliminata dal sistema		
Exit condition On failure	Non viene cancellata alcuna prenotazione, viene visualizzato un messaggio di errore		
Rilevanza/User Priority	Alta		
Frequenza stimata	20/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Attore:	L'utente accede alla pagina personale e visualizza le prenotazioni effettuate. Sceglie una prenotazione da cancellare e procede con la cancellazione.	
2	Sistema:	Il sistema chiede la conferma per l'eliminazione della prenotazione.	
3	Attore:	L'utente conferma l'intento di cancellare la prenotazione.	
4	Sistema:	Il sistema elimina dal database la prenotazione selezionata aggiornando le code e mostra all'utente la sua lista delle prenotazioni aggiornata.	
I Scenario/Flusso di eventi Alternativo: L'utente non conferma la cancellazione			
4.1	Sistema:	Il sistema non apporta modifiche e rimanda alla pagina personale dell'utente non modificata.	

3.5.3 Modello ad Oggetti

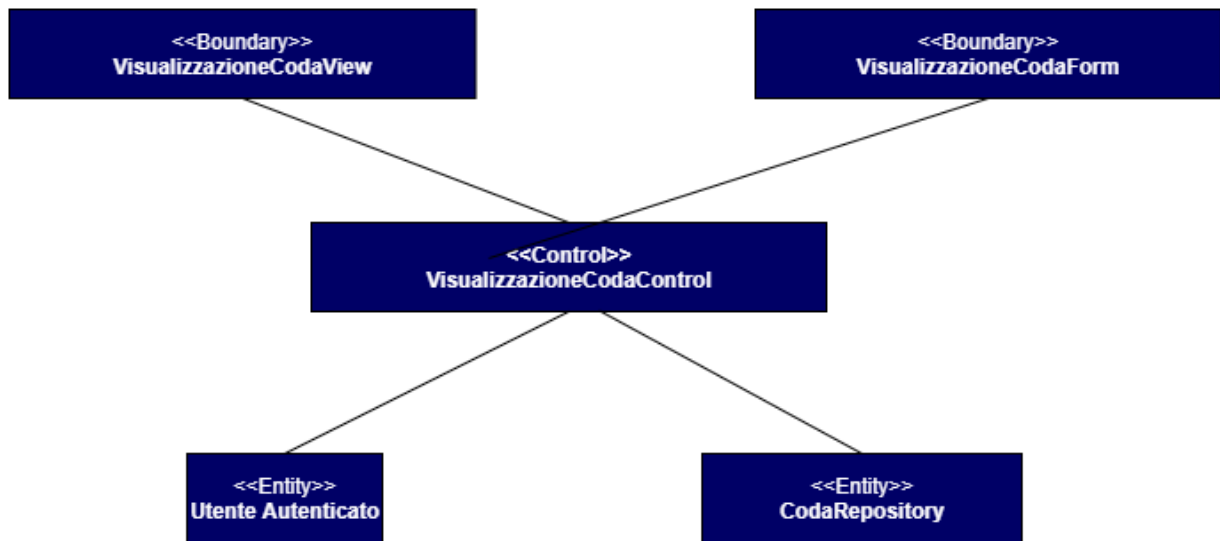
La sezione presenta un riepilogo di tutti gli oggetti individuati in fase di analisi, raggruppati nella tabella delle entità. Per ognuna delle gestioni è poi mostrato come tutti gli oggetti coinvolti si relazionano tra loro.

3.5.3.1 Entità di dominio



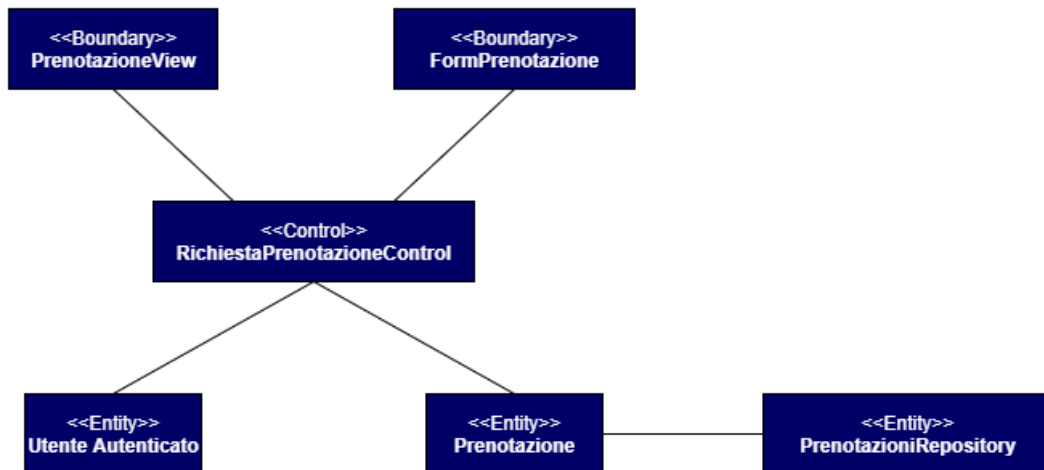
Nome oggetto	Tipologia	Descrizione
Utente	Entity	Entità che modella un utente
Prenotazione	Entity	Entità che modella una prenotazione dell'utente
Struttura	Entity	Entità che modella una struttura ospedaliera
Impiegato	Entity	Entità che modella un impiegato della struttura ospedaliera
Operazione	Entity	Entità che modella il tipo d'operazione

3.5.3.2 OM_VC-1: Visualizzazione Coda Prenotazioni



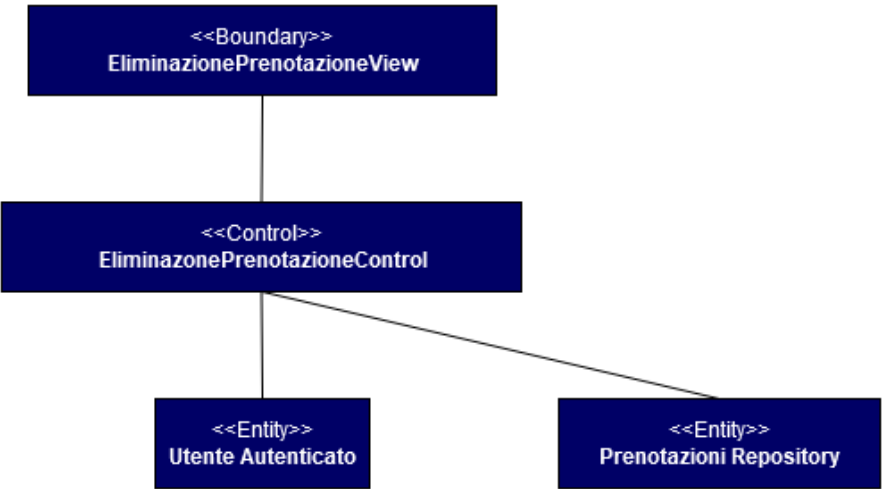
Nome Oggetto	Tipologia	Descrizione
Utente Autenticato	Entity	Entità che modella un utente autenticato
CodaRepository	Entity	Entità che modella un raccogliatore delle informazioni relative alle Code
VisualizzazioneCodaControl	Control	Control che coordina le operazioni relative alla visualizzazione della coda
VisualizzazioneCodaView	Boundary	Boundary che mostra le informazioni necessarie per ottenere una coda
VisualizzazioneCodaForm	Boundary	Boundary che permette all'utente di inserire le informazioni necessarie alla visualizzazione della coda

3.5.3.3 OM_GP-1: Richiesta Prenotazione



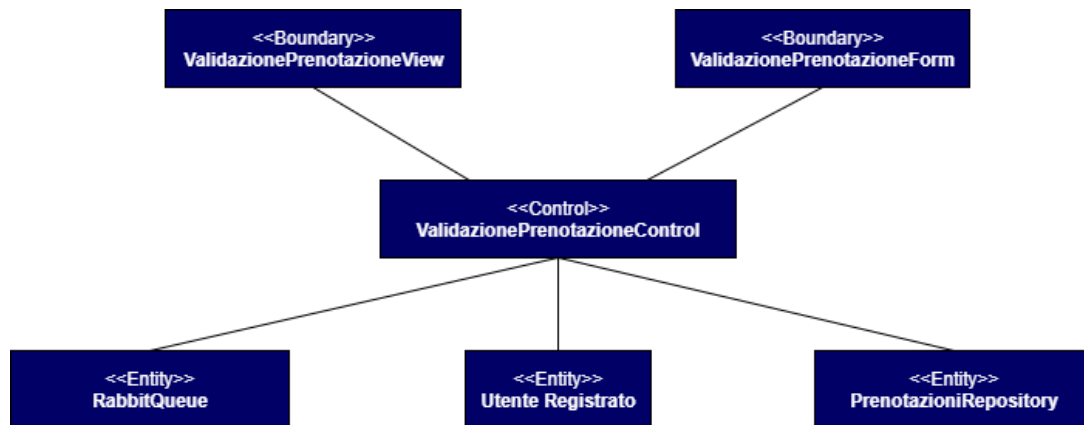
Nome oggetto	Tipologia	Descrizione
Utente Autenticato	Entity	Entità che modella un utente autenticato
Prenotazione	Entity	Entità che modella una prenotazione dell'utente
PrenotazioneRepository	Entity	Entità che modella un raccoglitore delle prenotazioni
RichiestaPrenotazioneControl	Control	Control che coordina le operazioni relative ad una richiesta di prenotazione
PrenotazioneView	Boundary	Boundary che mostra le informazioni necessarie per ottenere una prenotazione
FormPrenotazione	Boundary	Bondary che permette all'utente di inserire le informazioni necessarie per richiedere una prenotazione

3.5.3.4 OM_GP-2: Eliminazione Prenotazione



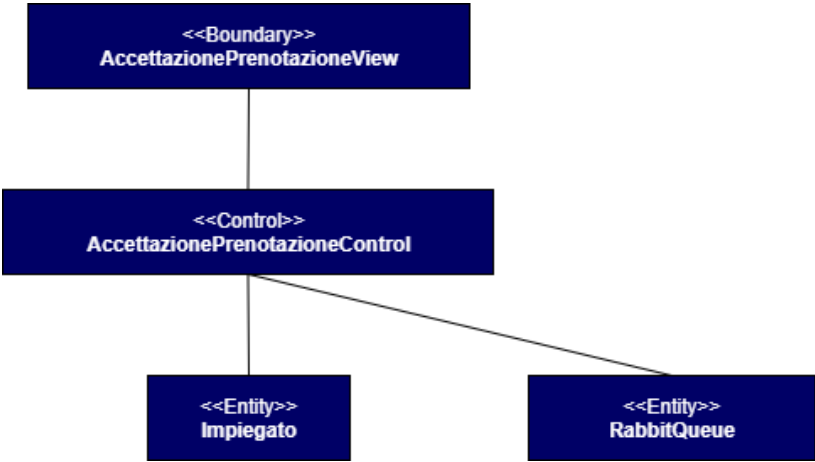
Nome oggetto	Tipologia	Descrizione
Utente Autenticato	Entity	Entità che modella un utente autenticato
Prenotazione Repository	Entity	Entità che modella un raccoglitore delle prenotazioni
EliminazionePrenotazioneControl	Control	Control che coordina le operazioni relative alla eliminazione di una prenotazione
EliminazionePrenotazioneView	Boundary	Boundary che mostra l'elenco delle prenotazioni in attesa

3.5.3.5 OM_GP-3: Validazione Prenotazione



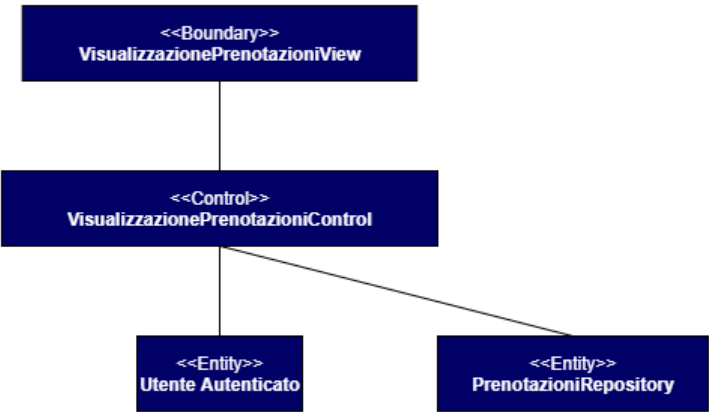
Nome oggetto	Tipologia	Descrizione
Utente Registrato	Entity	Entità che modella un utente registrato
PrenotazioniRepository	Entity	Entità che modella un raccogliatore delle prenotazioni
ValidazionePrenotazioneControl	Control	Control che coordina le operazioni relative alla validazione di una prenotazione
ValidazionePrenotazioneView	Boundary	Boundary che mostra le informazioni necessarie per la validazione di una prenotazione
ValidazionePrenotazioneForm	Boundary	Boundary che permette all'utente di inserire le informazioni necessarie per la validazione di una prenotazione

3.5.3.6 OM_GPI-1: Accettazione Prenotazione



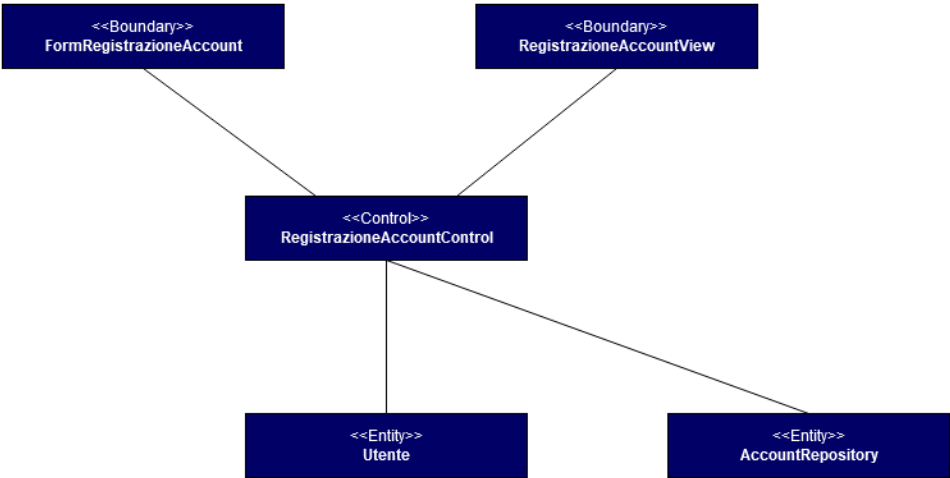
Nome oggetto	Tipologia	Descrizione
Impiegato	Entity	Entità che modella un impiegato
PrenotazioniRepository	Entity	Entità che modella un raccoglitore delle prenotazioni
AccettazionePrenotazioneControl	Control	Control che coordina le operazioni relative alla accettazione della prossima prenotazione in coda
AccettazioePrenotazioneView	Boundary	Boundary che permette la accettazione della prossima prenotazione in coda

3.5.3.7 OM_GP-4: Visualizzazione Prenotazione



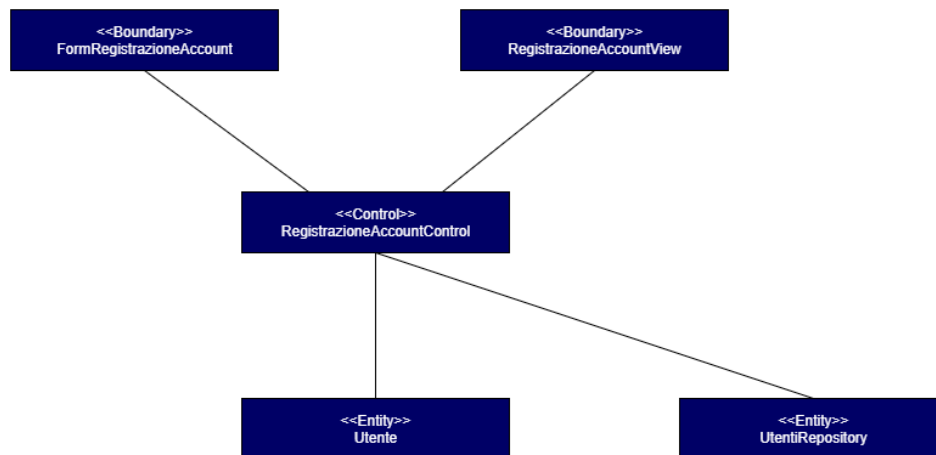
Nome oggetto	Tipologia	Descrizione
Utente Registrato	Entity	Entità che modella un utente autenticato
PrenotazioniRepository	Entity	Entità che modella un raccoglitore delle prenotazioni
VisualizzazionePrenotazioniControl	Control	Control che coordina le operazioni relative alla visualizzazione delle prenotazioni del utente
VisualizzazionePrenotazioneView	Boundary	Boundary che mostra le prenotazioni in attesa

3.5.3.8 OM_GU-1: Autenticazione



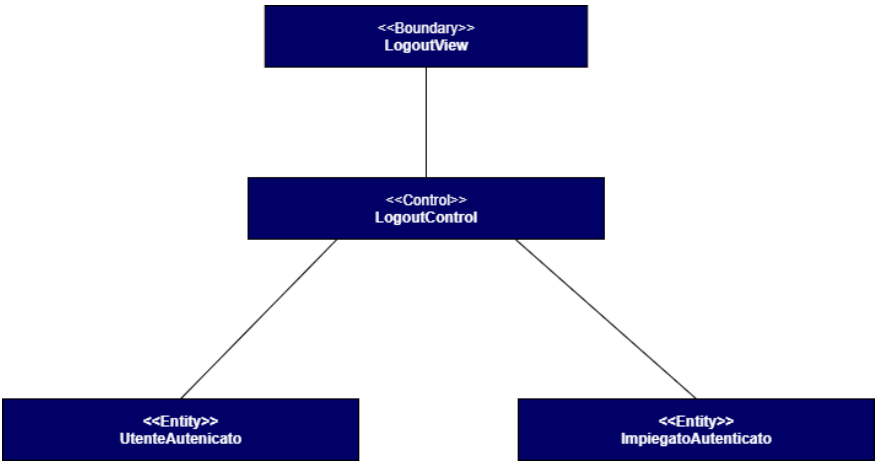
Nome oggetto	Tipologia	Descrizione
Utente Registrato	Entity	Entità che modella un utente registrato
AccountRepository	Entity	Entità che modella un raccoglitore degli account
AutenticazioneControl	Control	Control che coordina l'operazione d'autenticazione
AutenticazioneView	Boundary	Boundary che mostra le informazioni necessarie per un autenticazione
AutenticazioneForm	Boundary	Boundary che permette di inserire le informazioni necessarie per un autenticazione

3.5.3.9 OM_GU-2: Registrazione Account



Nome oggetto	Tipologia	Descrizione
Utente	Entity	Entità che modella un utente
UtenteRepository	Entity	Entità che modella un raccoglitore di utenti
RegistrazioneAccountControl	Control	Control che coordina le operazioni per la registrazione di un account
RegistrazioneAccountView	Boundary	Boundary che mostra le informazioni necessarie per la registrazione di un account
FormRegistrazioneAccount	Boundary	Boundary che permette di inserire le informazioni necessarie per la registrazione di un account

3.5.3.10 OM_GU-3: Logout

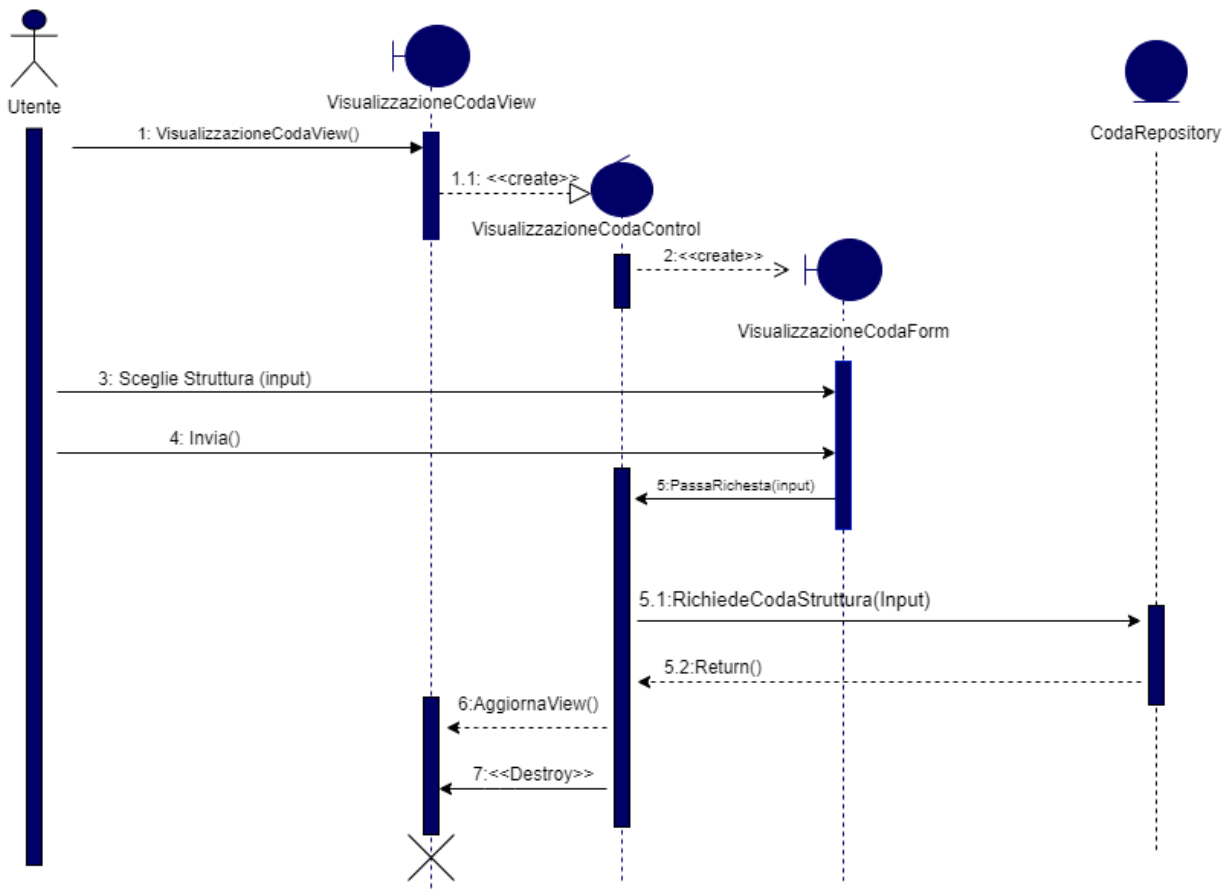


Nome oggetto	Tipologia	Descrizione
Utente Autenticato	Entity	Entità che modella un utente autenticato
Impiegato Autenticato	Entity	Entità che modella un impiegato autenticato
LogoutControl	Control	Control che coordina l'operazione di logout
LogoutView	Boundary	Boundary che permette all'utente di eseguire un logout

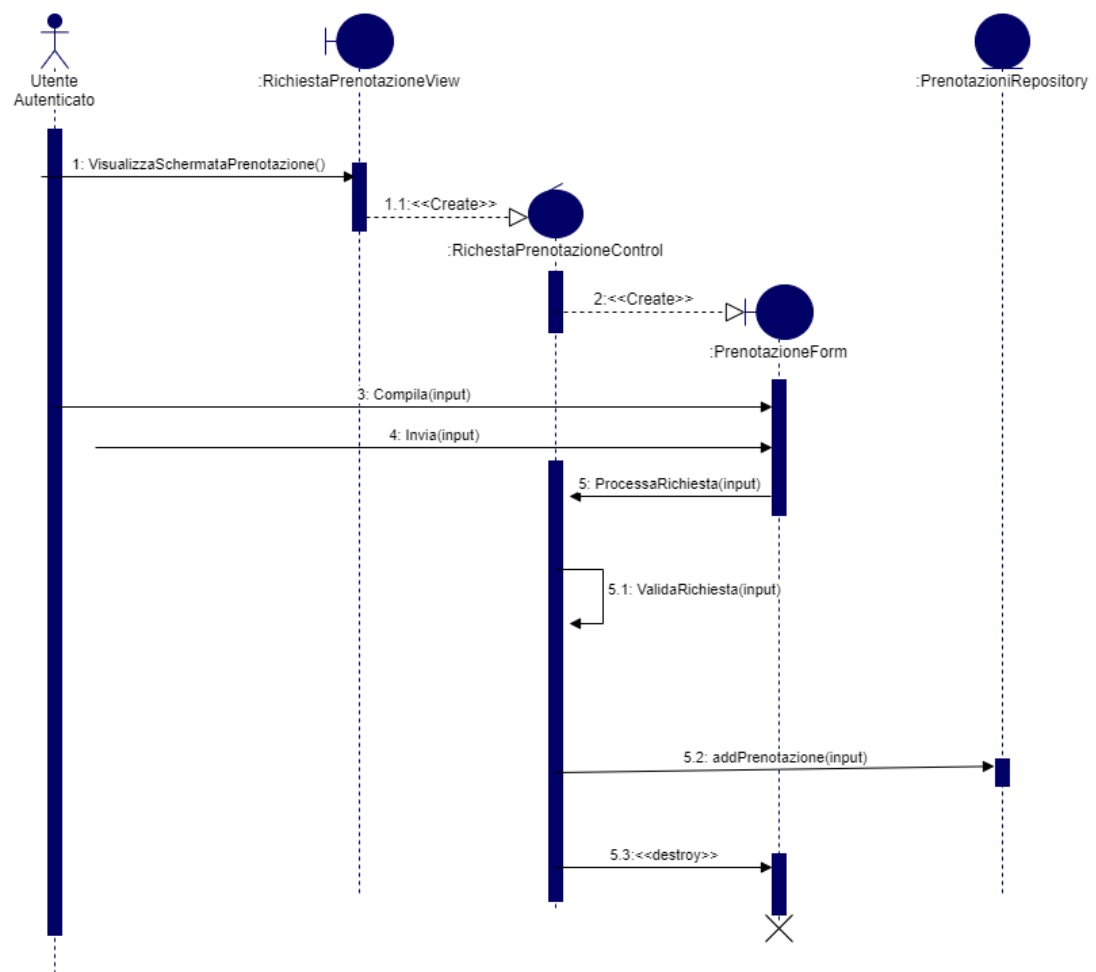
3.5.4 Modello dinamico

La descrizione del modello dinamico avviene tramite la rappresentazione delle interazioni tra l'utente ed il sistema (grazie all'utilizzo dei sequence diagrams) e la descrizione degli stati che caratterizzano il ciclo di vita degli oggetti

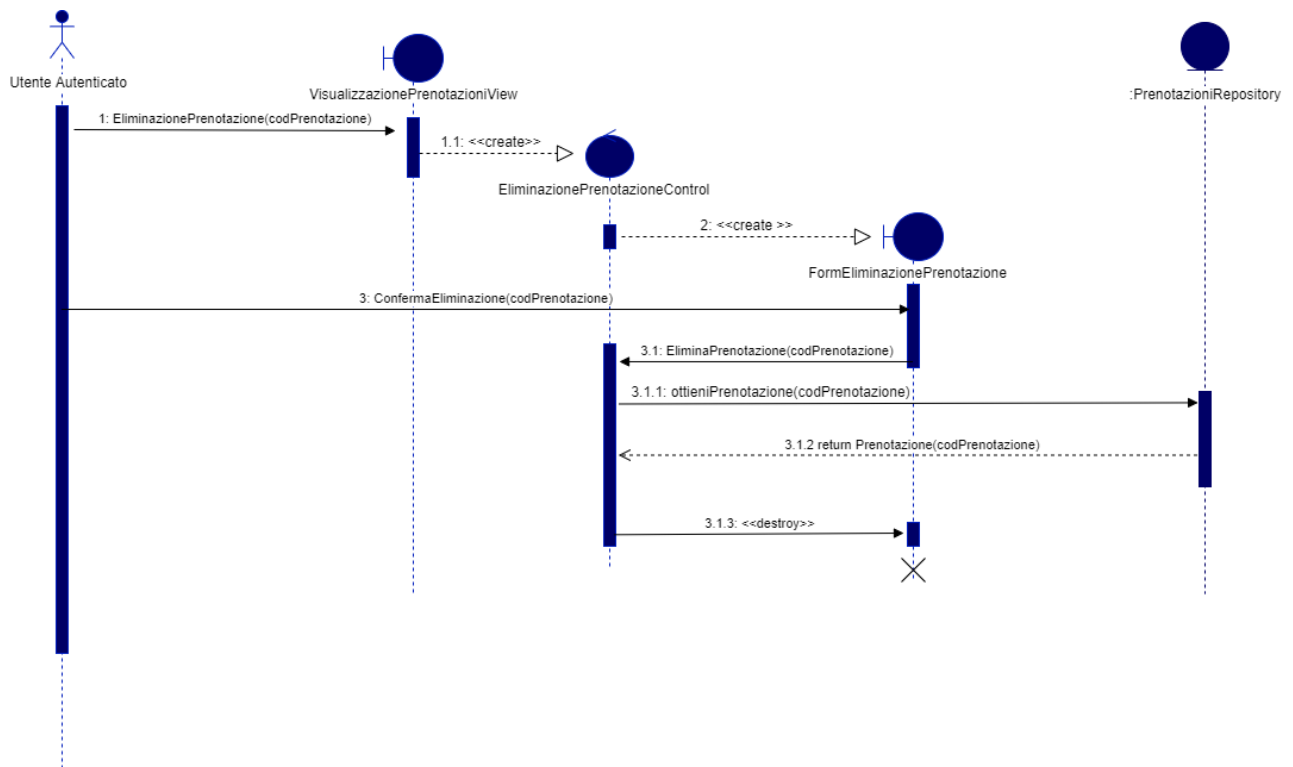
3.5.4.1 SD_VC-1: Visualizzazione Coda Prenotazione



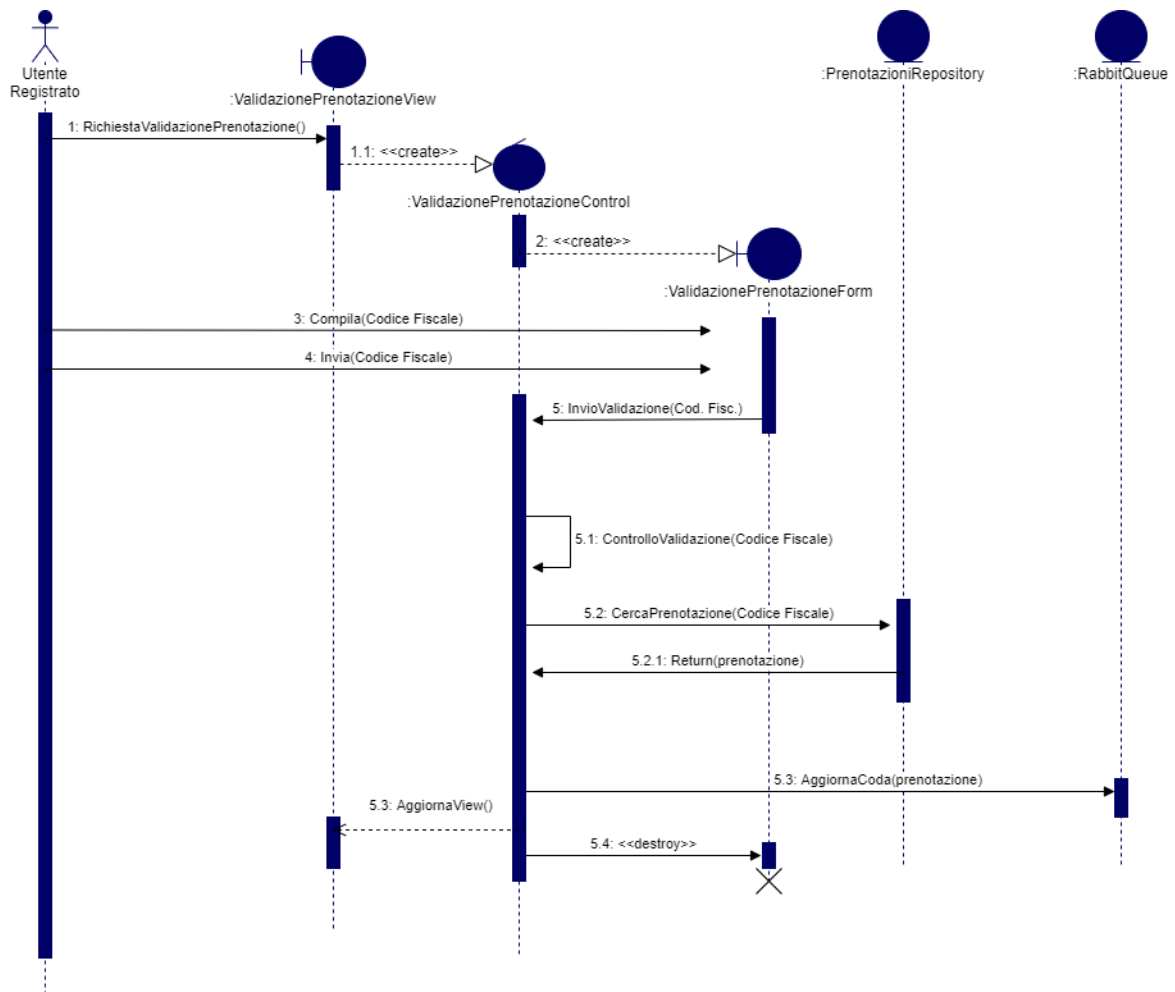
3.5.4.2 SD_GP-1: Richiesta Prenotazione



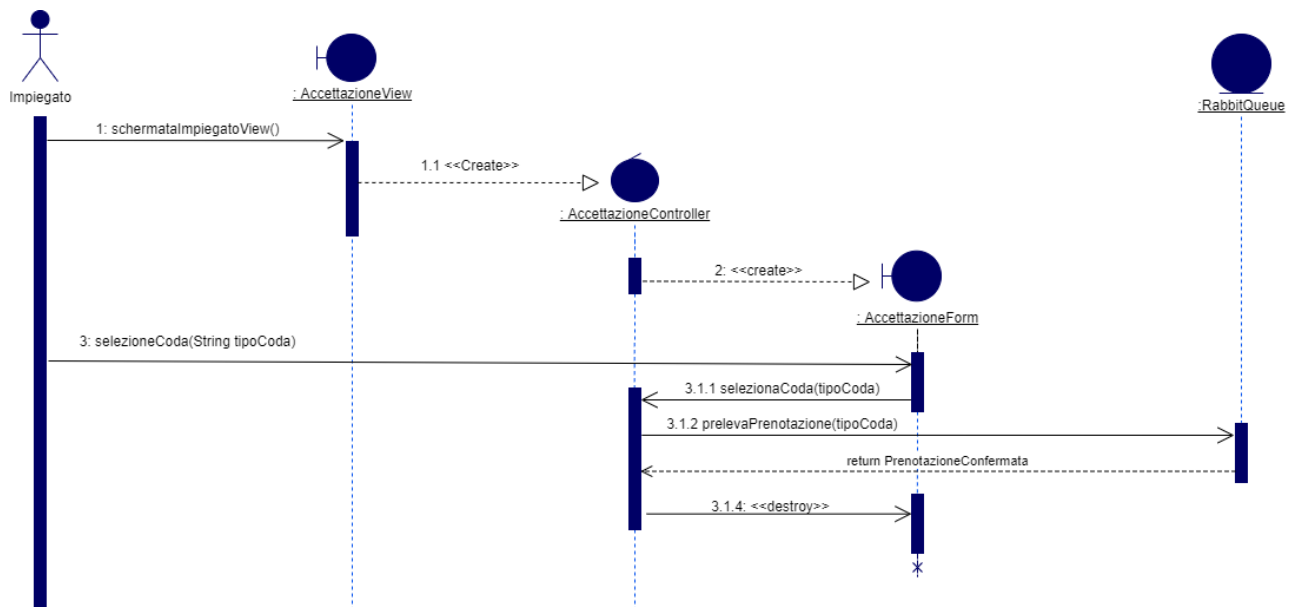
3.5.4.3 SD_GP-2: Eliminazione Prenotazione



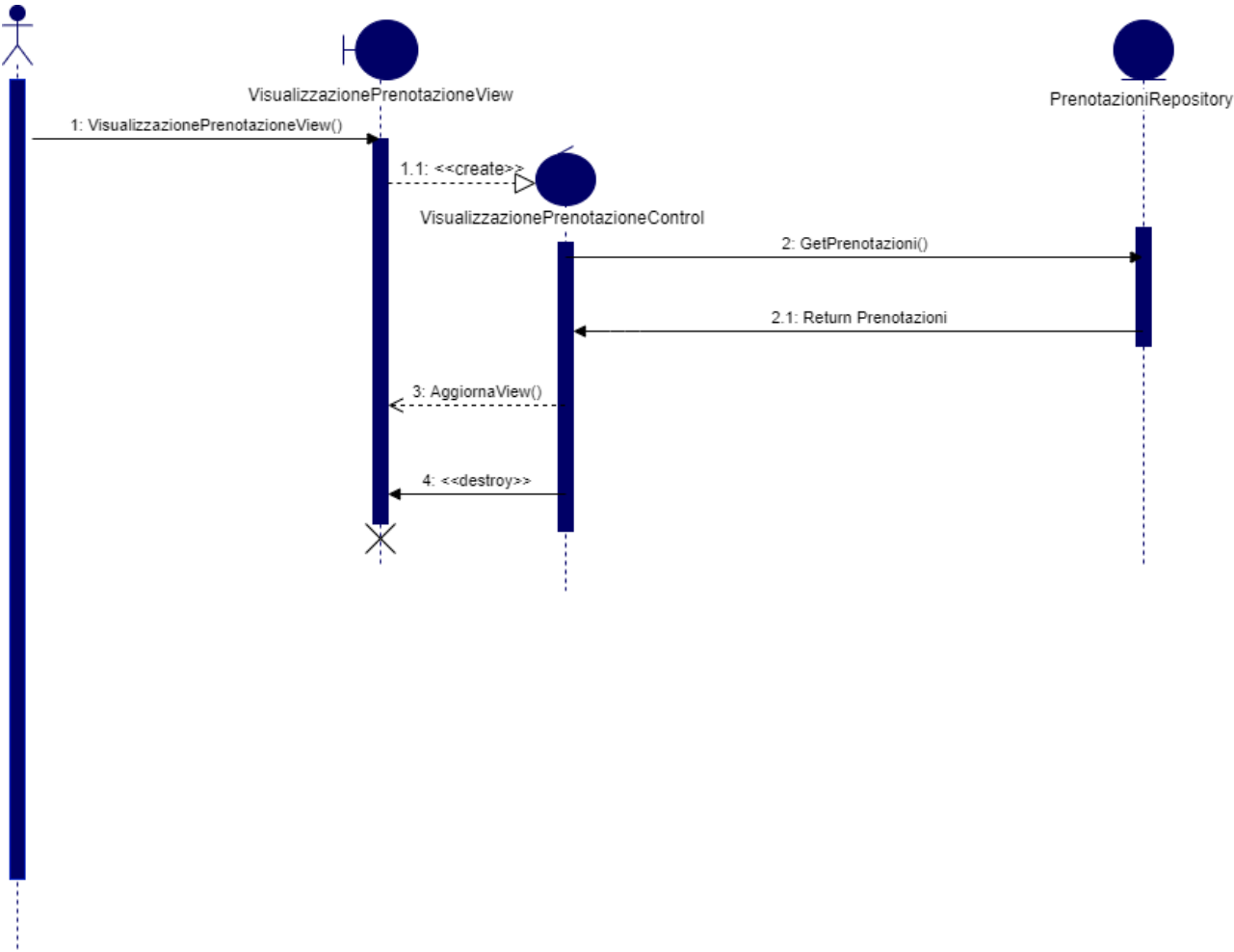
3.5.4.4 SD_GP-3: Validazione Prenotazione



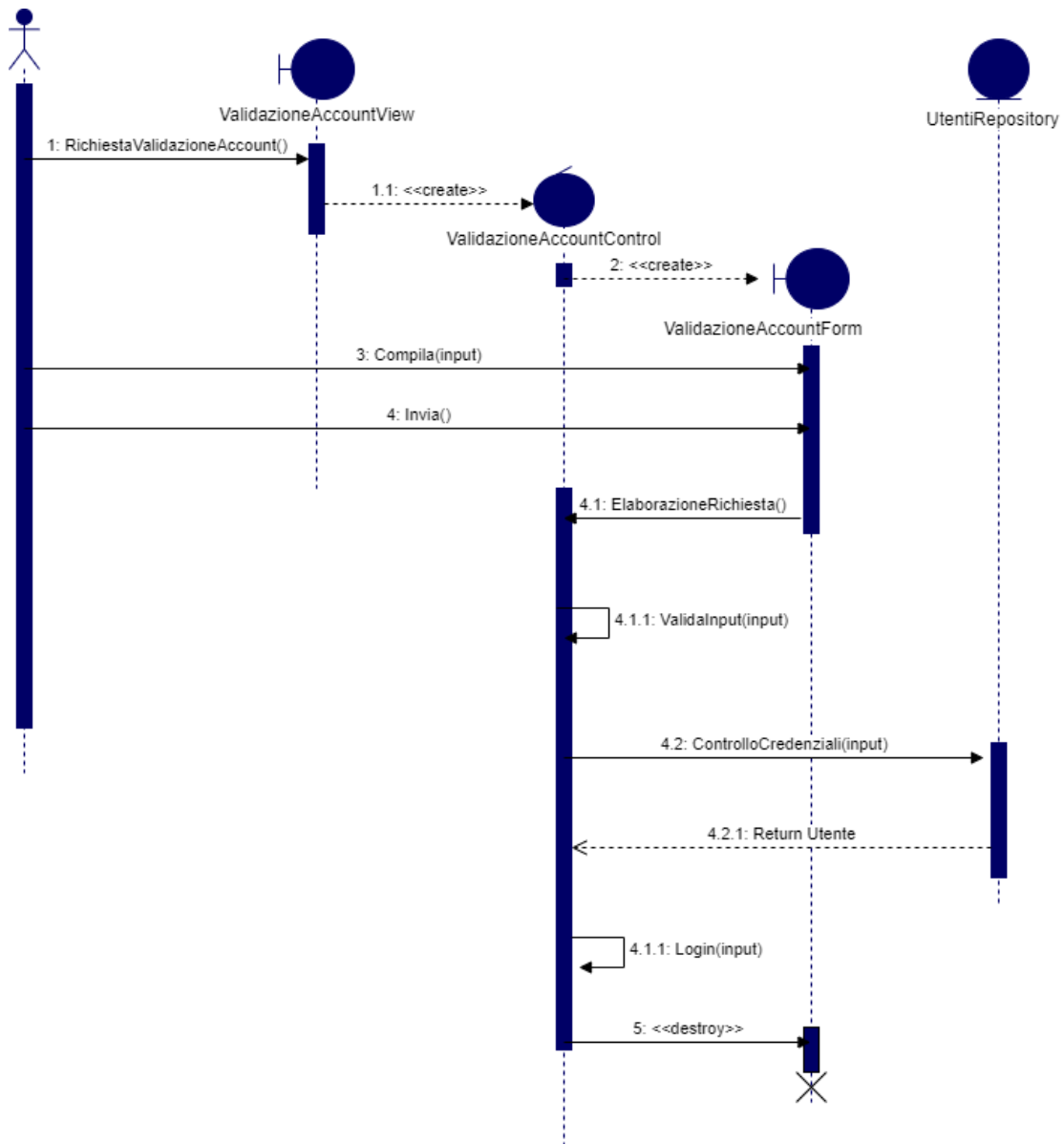
3.5.4.5 SD_GPI-1: Accettazione Prenotazione



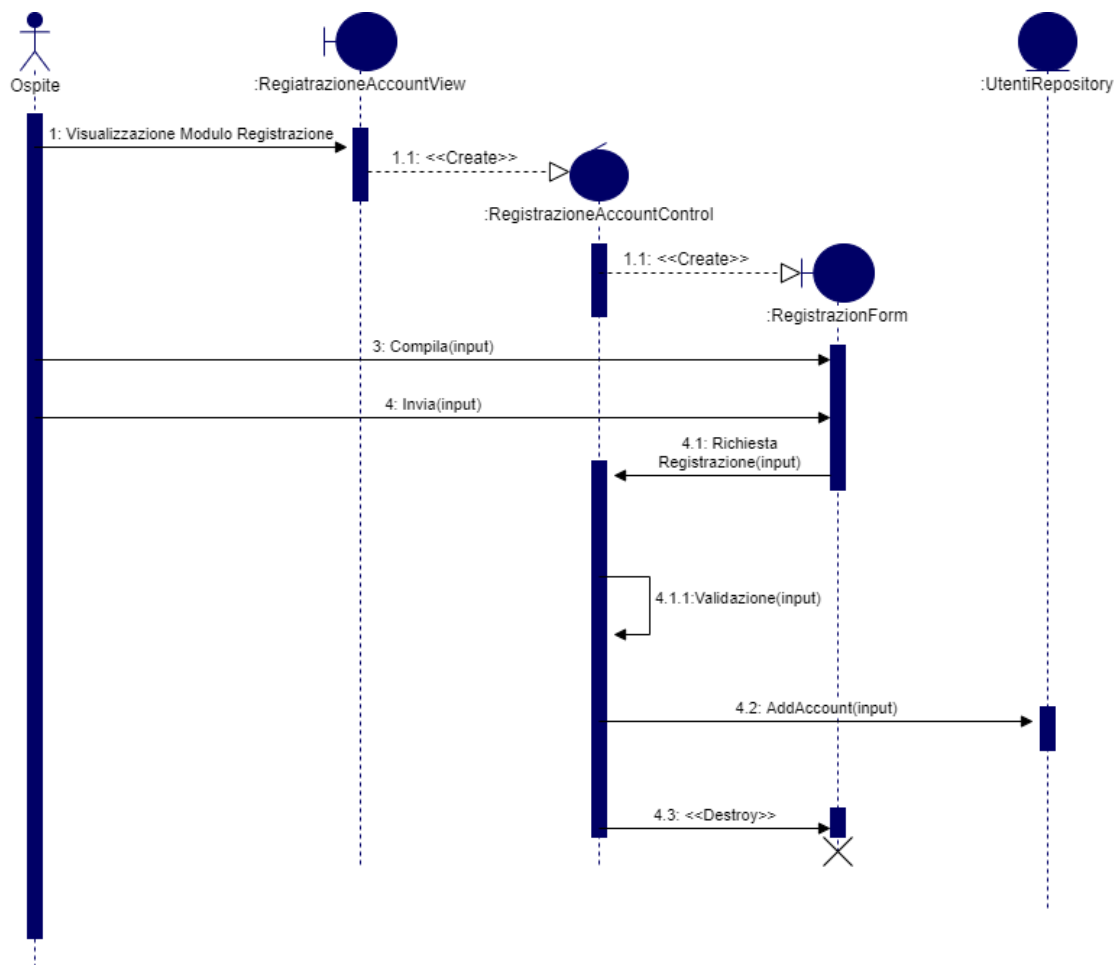
3.5.4.6 SD_GP-4 Visualizzazione Prenotazione



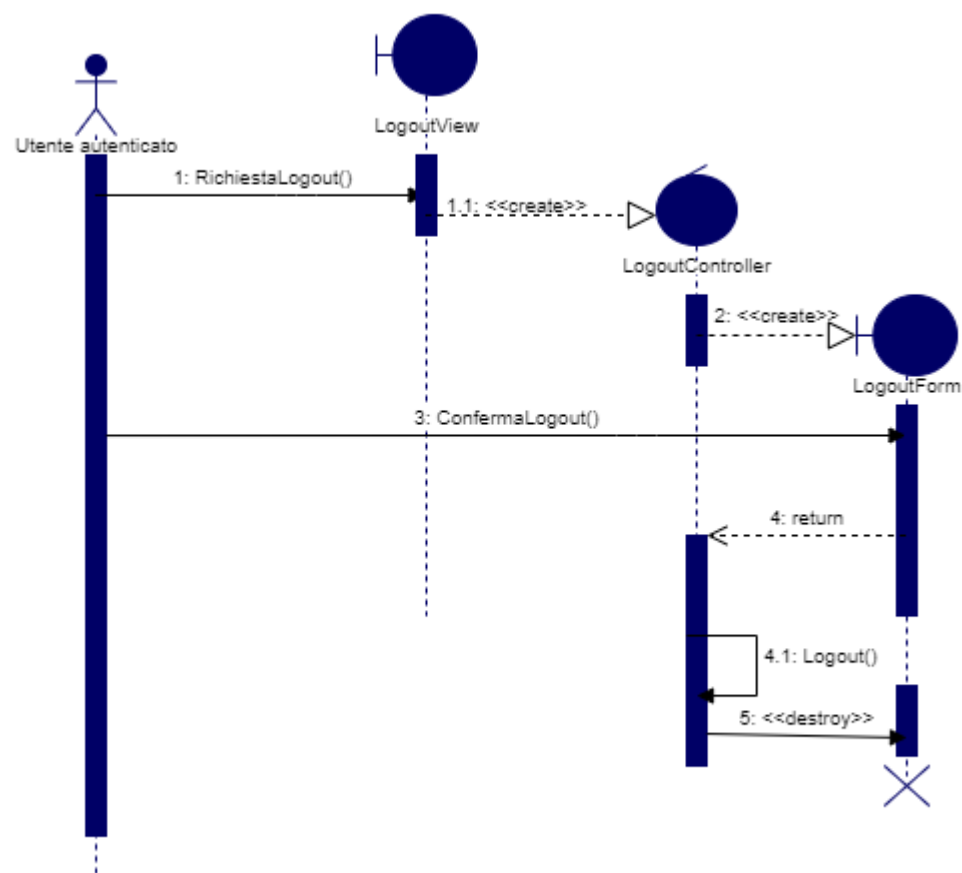
3.5.4.7 SD_GU-1: Autenticazione



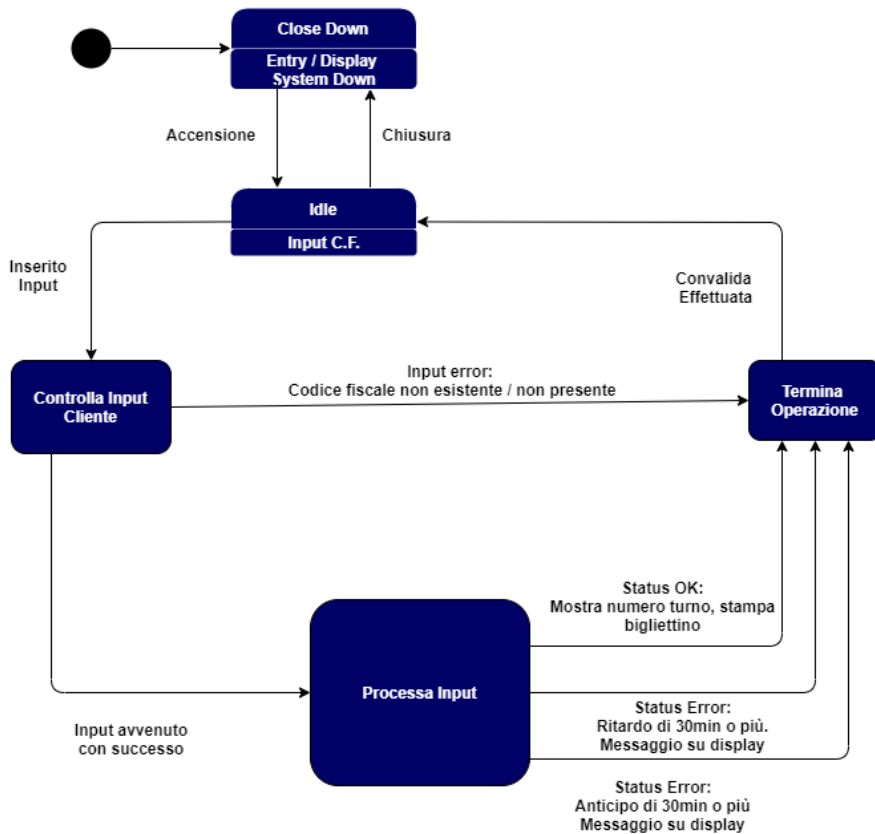
3.5.4.8 SD_GU-2: Registrazione Account



3.5.4.9 SD_GU-3: Logout

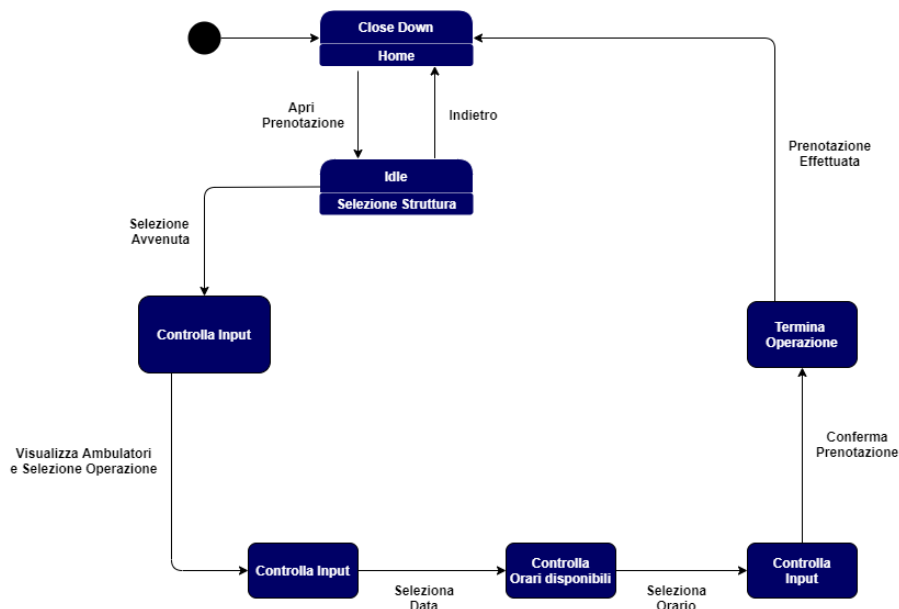


3.5.5 Statechart Diagrams



3.5.5.1 SCD_GP-3: Validazione Prenotazione

Considerato che ci sono molti output possibili abbiamo deciso, per semplificare il tutto, di aggiungere questo Statechart Diagram.

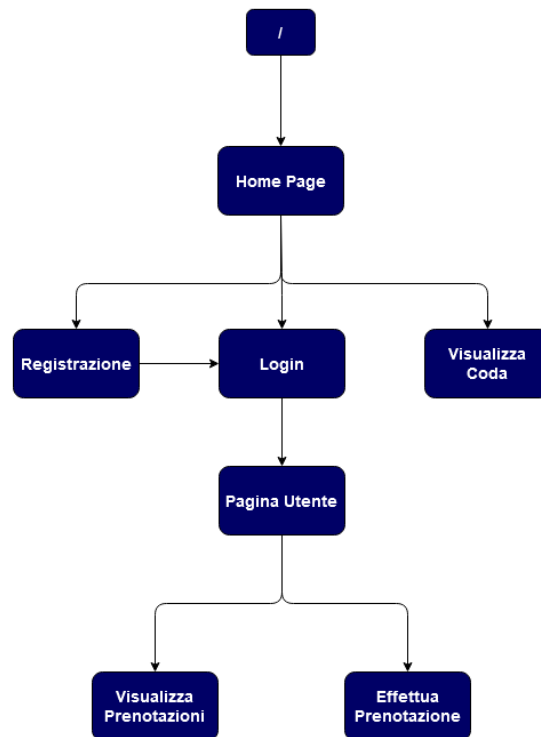


3.5.5.2 SCD_GP1: Richiesta Prenotazione

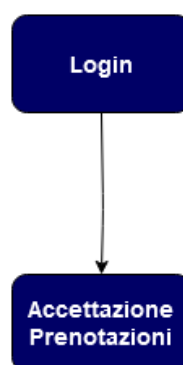
Visto che i campi della prenotazione vengono modificati in base alle scelte precedenti, per essere più chiari, abbiamo deciso di introdurre questo Statechart Diagram.

3.5.6 Interfaccia utente e Mock-ups

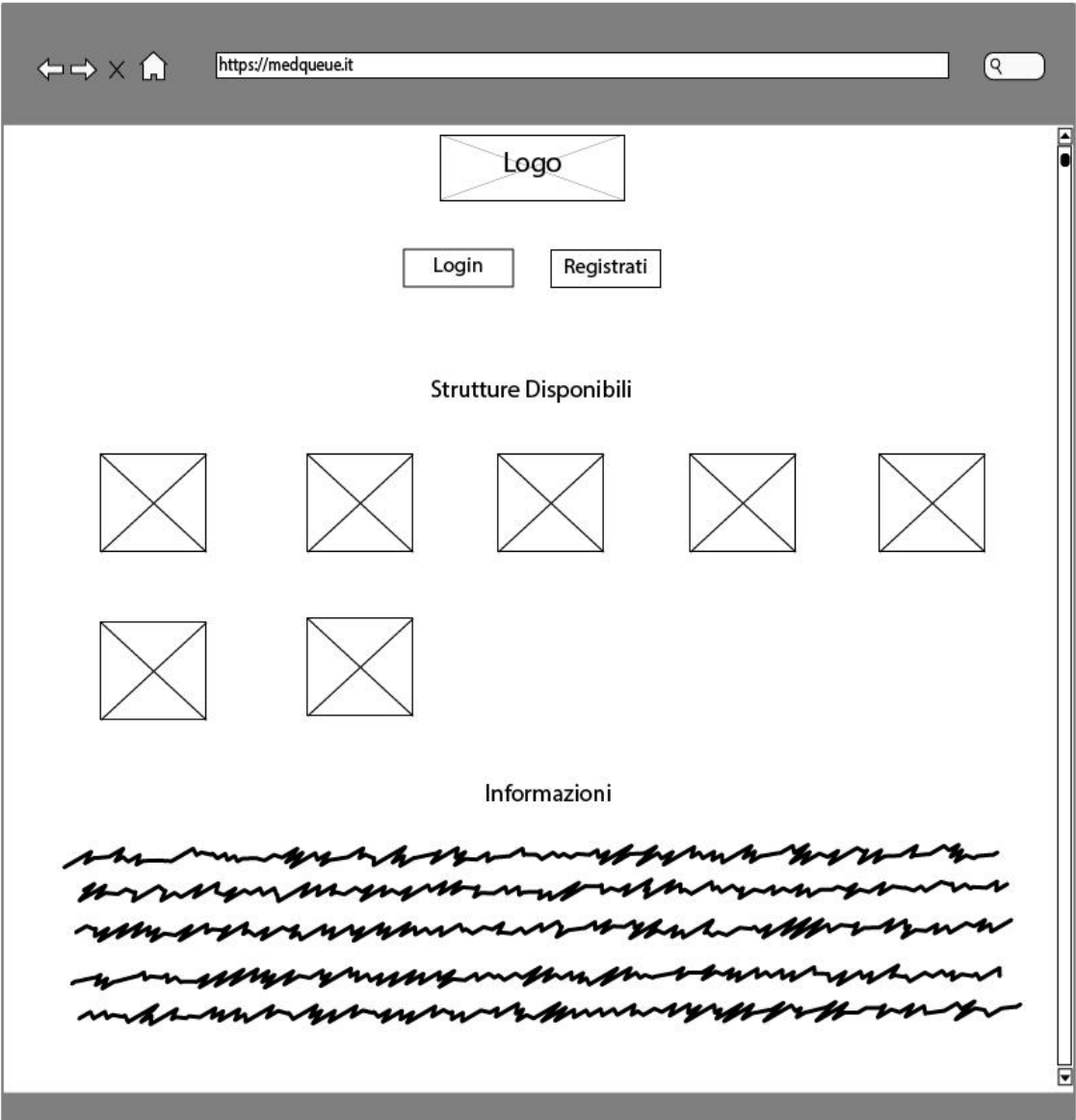
Rappresentazione di massima di come apparirà l'interfaccia utente di MedQueue da un qualsiasi web browser.



Rappresentazione di massima di come apparirà l'interfaccia dell'impiegato acceduta dall'applicazione MedQueue fornita alla struttura



3.5.6.1 UI_1: Homepage



3.5.6.2 UI_2: Registrazione

The image shows a web browser window with the address bar displaying `https://medqueue.it`. The main content area contains a registration form titled "Registrazione". The form is enclosed in a large rectangular box with a diagonal cross (X) drawn over it, indicating it is a placeholder or a design mockup. The form fields are as follows:

- Nome: _____
- Cognome: _____
- Codice Fiscale: _____
- Data di Nascita: _____
- E-Mail: _____
- Password: _____
- Numero di Telefono: _____

At the bottom of the form is a button labeled "Inva".

3.5.6.3 UI_3: Login

https://medqueue.it

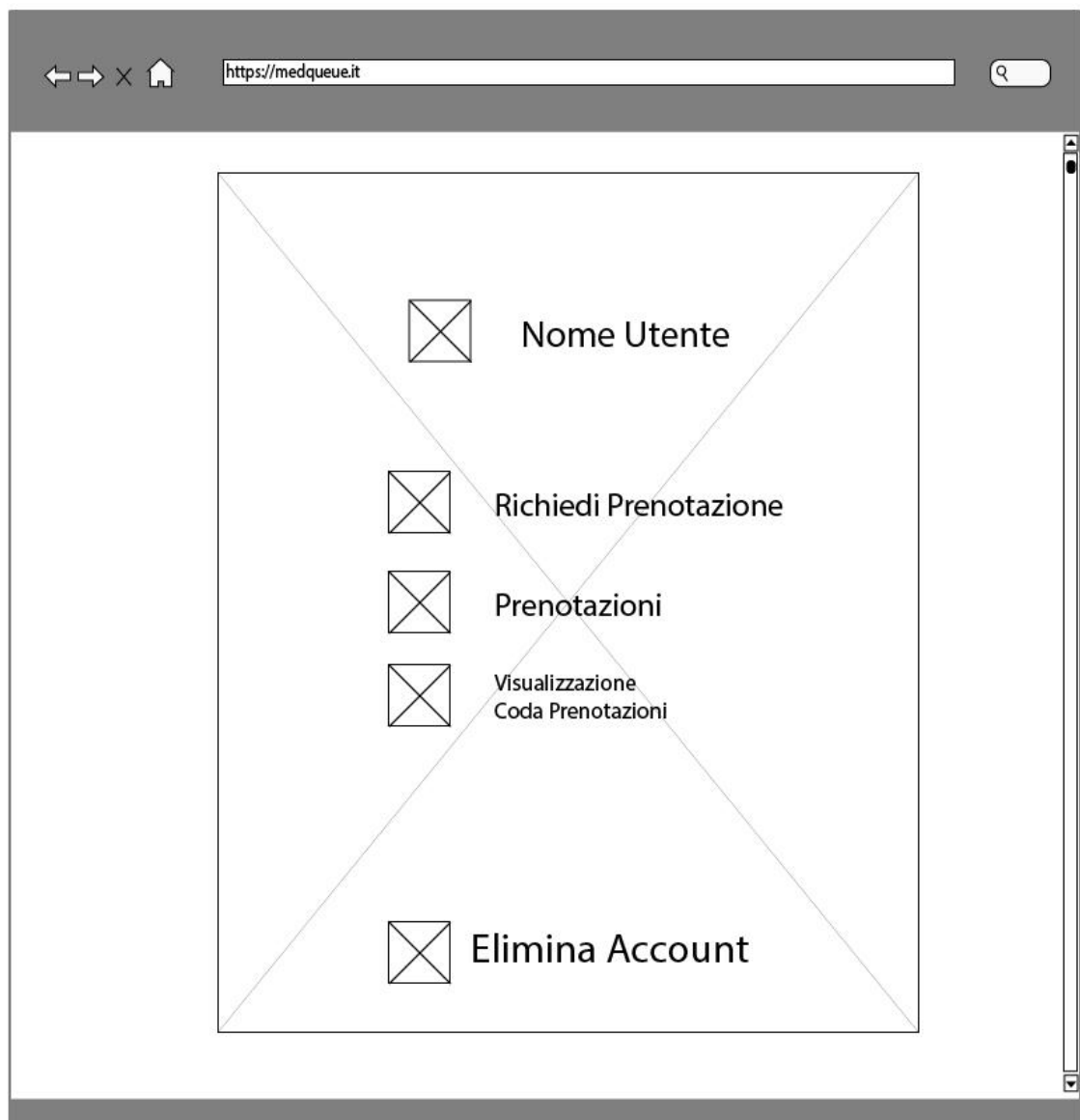
Login

Codice Fiscale: _____

Password: _____

Inva

3.5.6.4 UI_4: Area Personale



3.5.6.5 UI_5: Richiesta Prenotazione 1

Richiesta Prenotazione

Struttura: +

Operazione:

Data:

1	2	3	4	5	6	7

Ora:

Conferma

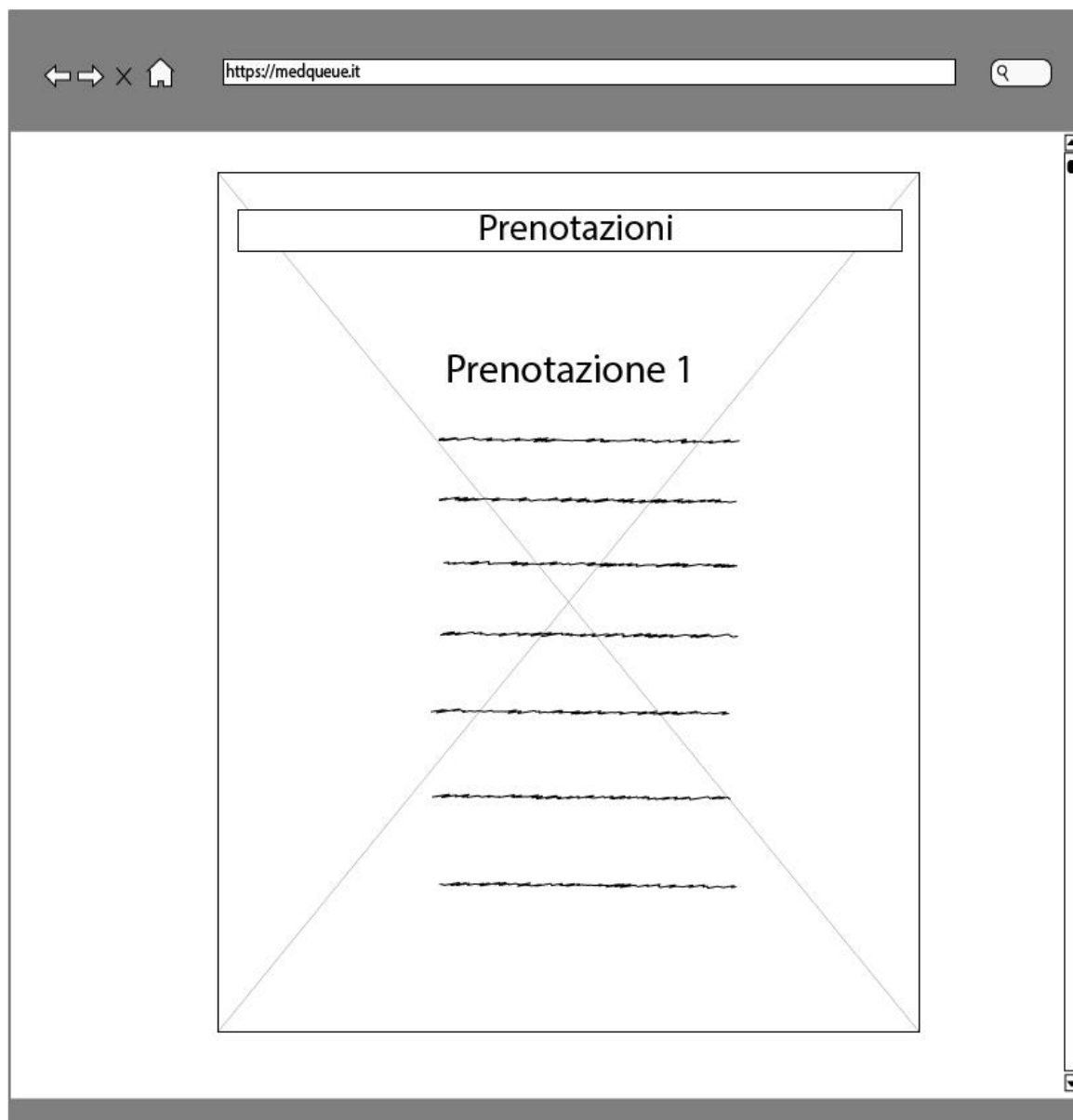
3.5.6.6 UI_6: Richiesta Prenotazione 2

The screenshot shows a web browser window with the address bar displaying 'https://medqueue.it'. The main content area contains a form titled 'Richiesta Prenotazione' (Appointment Request). The form is enclosed in a large square frame with diagonal lines. It includes the following elements:

- Struttura:** A text input field with a wavy line indicating a placeholder or a dropdown menu.
- Operazione:** A text input field.
- Data:** A date selection interface. It features a header row with numbers 1 through 7, and a grid of smaller boxes below it, some containing wavy lines.
- Ora:** A text input field.
- Conferma:** A large rectangular button at the bottom of the form.

There are also some additional visual elements: a small box with wavy lines to the right of the 'Struttura' field, and a larger box with multiple wavy lines to the right of the 'Operazione' field. The browser's navigation bar at the top includes back, forward, and home icons.

3.5.6.7 UI_7: Visualizza Prenotazioni 1

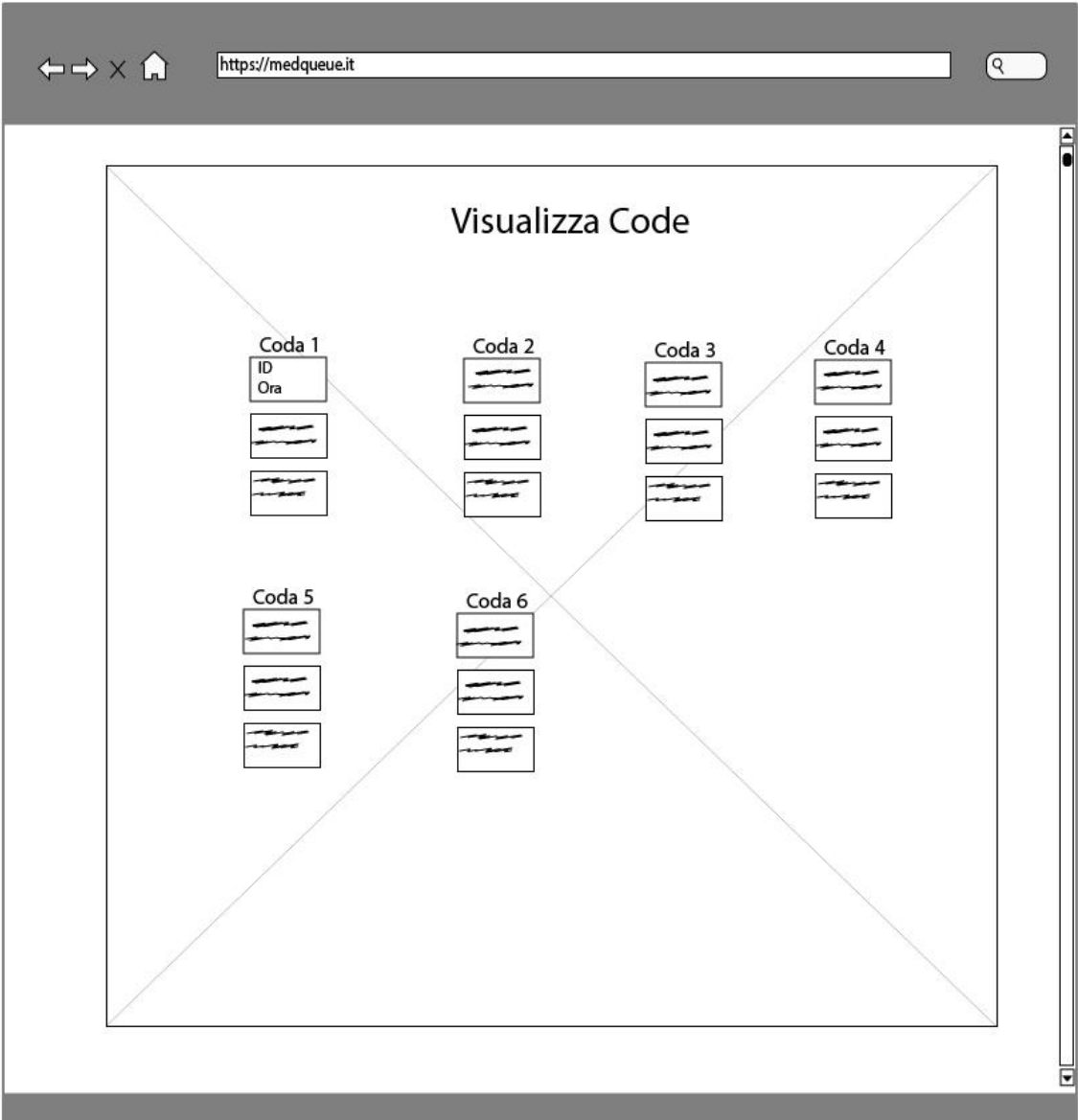


3.5.6.8 UI_8: Visualizza Prenotazioni 2

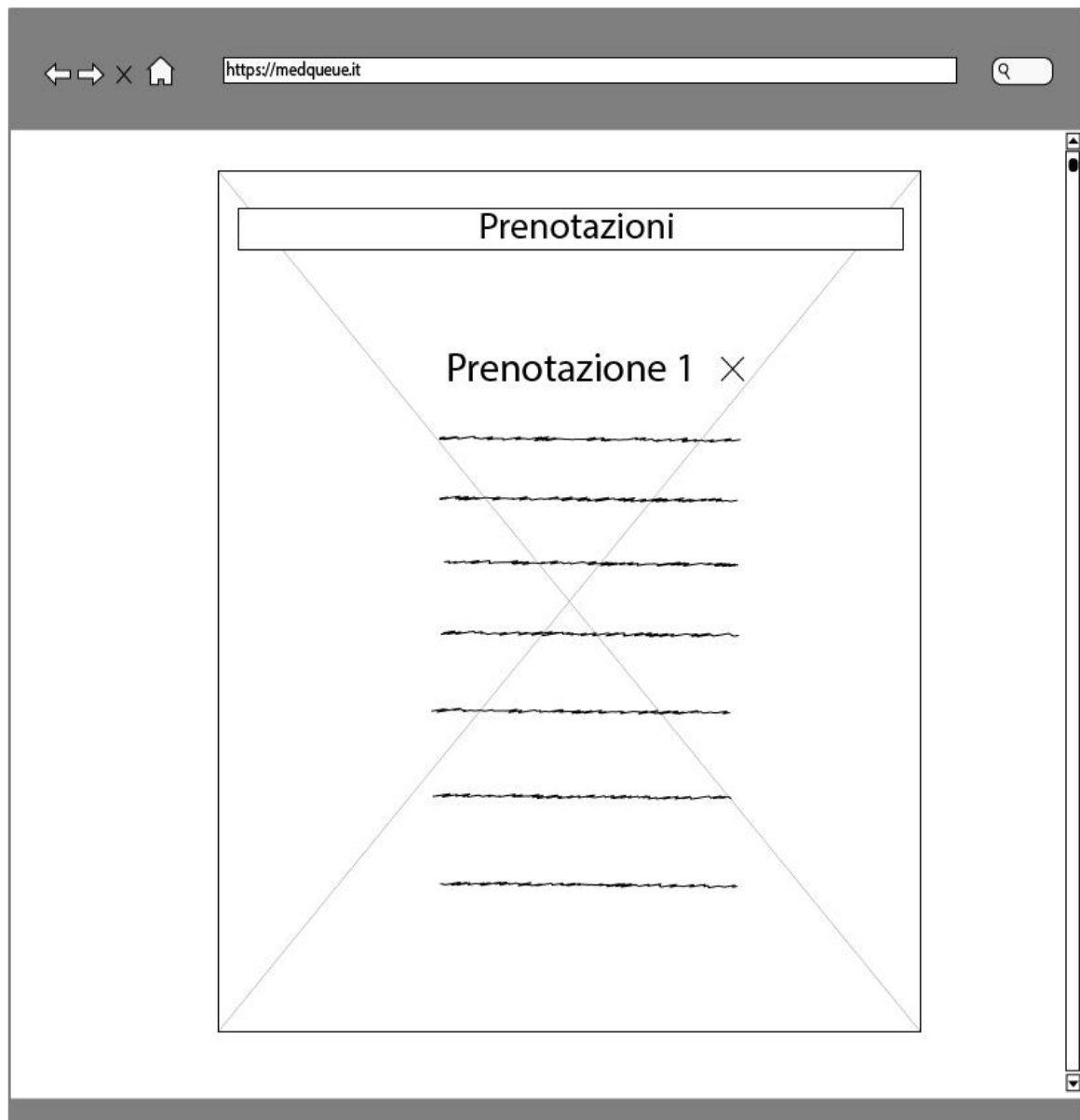
The image shows a web browser window with the address bar displaying `https://medqueue.it`. A modal window titled "La tua Prenotazione" is centered on the screen, overlaid on a background labeled "Prenotazioni". The modal contains the following elements:

- ID:** A text input field consisting of 10 empty boxes.
- Ora:** A time selection interface with two boxes for hours, a colon separator, and two boxes for minutes.
- Data:** A date selection interface with two boxes for the day, a slash separator, two boxes for the month, a slash separator, and four boxes for the year.
- Nome Struttura:** A text input field.
- Cancella:** A rectangular button.

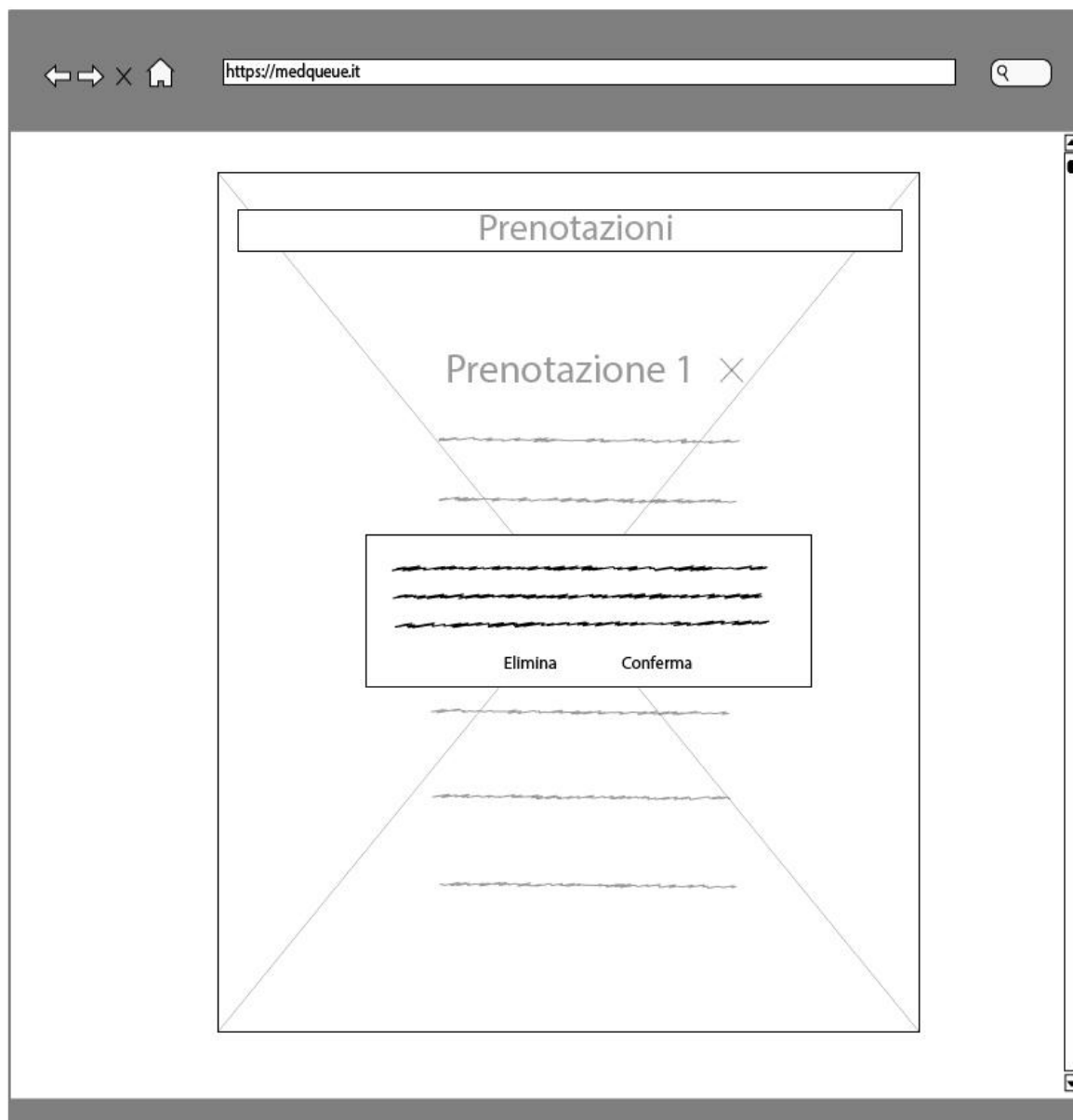
3.5.6.9 UI_9: Visualizza coda



3.5.6.10 UI_10: Elimina Prenotazione 1



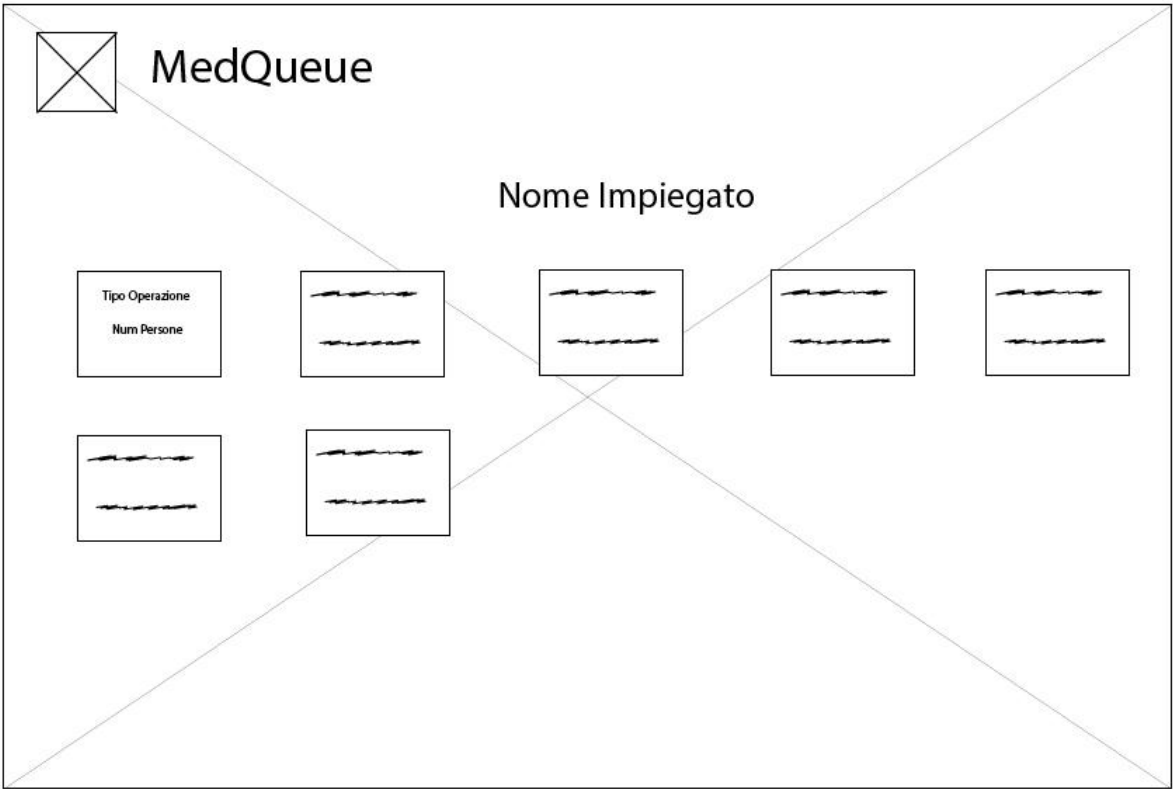
3.5.6.11 UI_11: Elimina Prenotazione 2



3.5.6.12 UI_12: Convalida

The image shows a web browser window with the address bar displaying `https://medqueue.it`. The main content area contains a form titled "Convalida". Inside the form, there is a label "Codice Fiscale:" followed by a text input field. At the bottom of the form, there is a button labeled "Inva". The form is enclosed in a large rectangular box with a diagonal crosshair, suggesting it might be a placeholder or a specific design element.

3.5.6.13 UI_13: Accettazione Prenotazione



3.6 Matrice di tracciabilità

Nome Progetto:

Descrizione del progetto:

MedQueue

Piattaforma per la gestione di prenotazioni
di un ufficio ospedaliero

ID	Utente	Requisito Funzionale	Nome	Stato	Scenario	Mock-Up
RF1	Utente Registrato	Richiesta prenotazione	Prenotazione	E	SC_GP-1	UI_5/UI_6
RF2	Utente Registrato	Eliminazione prenotazione	Eliminazione prenotazione	E	SC_GP-2	UI_10
RF3	Utente Registrato	Validazione prenotazione	Validazione	E	SC_GP-3	UI_12
RF4	Utente/Utente Registrato	Visualizzazione coda prenotazioni	Visualizzazione coda	E	SC_VC-1	UI_9
RF5	Impiegato	Accettazione prenotazione	Accettazione	E	SC_GPI-1	UI_13

ID	Use Case	Sequence Diagram	Object Diagram	Statechart	Componenti del Sistema	Moduli di Sistema	Test Case
RF1	UC_GP-1	SD_GP-1	OM_GP-1	SCD_GP1			
RF2	UC_GP-2	SD_GP-2	OM_GP-2	/			
RF3	UC_GP-3	SD_GP-3	OM_GP-3	SCD_GP-3			
RF4	UC_VC-1	SD_VC-1	OM_VC-1	/			
RF5	UC_GPI-1	SD_GPI-1	OM_GPI-1	/			

4. Glossario

Requisito funzionale: Funzionalità del sistema.

Requisito non funzionale: Tutti quei requisiti che non riguardano direttamente una funzionalità del sistema bensì riguardano caratteristiche di qualità come ad esempio Usabilità, Performance, Supportabilità ed Affidabilità.

Use case: Il caso d'uso in informatica è una tecnica usata nei processi di Ingegneria del Software per effettuare in maniera esaustiva e non ambigua la raccolta dei requisiti al fine di produrre software di qualità. In altre parole, un caso d'uso viene utilizzato per esprimere l'interazione che avviene tra utente e sistema.

Diagrammi dei casi d'uso (UCD): Modellano il comportamento esterno di un sistema in termini delle funzioni che esso mette a disposizione agli attori che interagiscono con essi (utenti, altri sistemi software...). Gli UCS sono il diagramma principale nella vista dei casi d'uso.

Sequence diagram: Diagramma previsto dall'UML utilizzato per descrivere uno use case attraverso le relazioni che intercorrono in termini di messaggi tra attori, oggetti di business, oggetti o entità del sistema.

Statechart diagram: Diagramma previsto dall'UML utilizzato per descrivere il comportamento di oggetti del sistema in termini di stato. In altre parole, modella quello che è il ciclo di vita di un oggetto.

Class diagram: Diagramma previsto dall'UML che consente di descrivere tipi di entità con le loro caratteristiche e le eventuali relazioni tra questi tipi.

Mock-ups: Prototipazione digitale che permette la visione complessiva e generale di un'interfaccia utente.

Navigation path: Grafico per la vista complessiva delle pagine presenti nel sistema e dei percorsi possibili per il loro raggiungimento.

Form: È un modulo elettronico che tramite un'interfaccia di un programma consente a un utente di un sito web di inserire ed inviare uno o più dati.

RabbitMQ: È un sistema di messaggistica utilizzato per la gestione di coda a priorità.

RabbitQueue: È una coda gestita da RabbitMQ.