

RAD

Requirement Analysis Document

MedQueue

Afeltra Angelo

Amato Adriano

Fucile Andrea

Rapa Giovanni

Sommario

1. [**Introduzione**](#Introduzione)
   1. [Scopo del sistema](#Scopodelsistema)
   2. [Ambito del sistema](#ambitodelsistema)
   3. [Obiettivi e criteri di successo del sistema](#obiettiviecriteri)

* 1. [Definizione, acronimi e abbreviazioni](#definizioniacronimi)
  2. [Panoramica](#panoramica)

1. [**Sistema corrente**](#sistemacorrente)

2.1 [Diagramma delle attività](#diagrammaattività)

1. [**Sistema proposto**](#sistemaproposto)
   1. [Diagramma delle attività](#diagrammaatt2)

3.1.1 [Invio, conferma, accettazione prenotazione](#invioconfermaaccprenot)

* 1. [Requisiti funzionali](#requisitifunzionali)
     1. [Requisiti per la visualizzazione code](#requisitiperlavisualizzazionecode)
     2. [Requisiti per la gestione delle prenotazioni](#requisitiperlagestionedelleprenotazioni)
     3. [Requisiti per la gestione delle prenotazioni svolte dall’impiegato](#requisitiperlagestionedellepreimp)
     4. [Requisiti per la gestione dell’utente](#reqperlagestioneutente)
     5. [Requisiti per la gestione dell’impiegato](#reqperlagestioneimpiegato)
  2. [Requisisti non funzionali](#reqnonfunzionali)
     1. [Usabilità](#usabilità)
     2. [Affidabilità](#affidabilità)
     3. [Prestazioni](#prestazioni)
     4. [Sostenibilità](#sostenibilità)
  3. [Pseudo Requisiti](#pseudorequisiti)

3.4.1 [Implementazione](#implementazione)

* 1. [Modello di sistema](#modellodisistema)
     1. [Scenari](#scenari)
     2. [Use Case](#usecase)

3.5.2.1 [UC\_VC-1: Visualizzazione Coda Presenza](#ucvc1)

3.5.2.2 [UC\_GP-1: Richiesta Prenotazione](#ucgp1)

3.5.2.3 [UC\_GP-3: Validazione Prenotazione](#ucgp3)

3.5.2.4 [UC\_GPI-1: Accettazione Prenotazioni](#ucgpi1)

3.5.2.5 [UC\_GP-2: Eliminazione Prenotazione](#ucgp2)

* + 1. [Modello ad Oggetti](#modelloadoggetti)

3.5.3.1 [Entità di dominio](#entitàdidominio2)

3.5.3.2 [OM\_VC-1: Visualizzazione Coda Presenza](#omvc1)

3.5.3.3 [OM\_GP-1: Richiesta Prenotazione](#omgp1)

3.5.3.4 [OM\_GP-2: Eliminazione Prenotazione](#omgp2)

3.5.3.5 [OM\_GP-3: Validazione Prenotazione](#omgp3)

3.5.3.6 [OM\_GPI-1: Accettazione Prenotazione](#omgpi1)

3.5.3.7 [OM\_GP-4: Visualizzazione Prenotazione](#omgp4)

3.5.3.8 [OM\_GU-1: Autenticazione](#omgu1)

3.5.3.9 [OM\_GU-2: Registrazione Account](#omgu2)

3.5.3.10 [OM\_GU-3: Logout](#omgu3)

3.5.4 [Modello dinamico](#modellodinamico)

3.5.4.1 [SD\_VC-1: Visualizzazione Coda Presenza](#sdvc1)

3.5.4.2 [SD\_GP-1: Richiesta Prenotazione](#sdgp1)

3.5.4.3 [SD\_GP-2: Eliminazione Prenotazione](#sdgp2)

3.5.4.4 [SD\_GP-3: Validazione Prenotazione](#sdgp3)

3.5.4.5 [SD\_GPI-1: Accettazione Prenotazione](#sdgpi1)

3.5.4.6 [SD\_GP-4: Visualizzazione Prenotazione](#sdgp4)

3.5.4.7 [SD\_GU-1: Autenticazione](#sdgu1)

3.5.4.8 [SD\_GU-2: Registrazione Account](#sdgu2)

3.5.4.9 [SD\_GU-3: Logout](#sdgu3)

3.5.5 [Statechart Diagrams](#statecharts)

3.5.5.1 [SCD\_GP-3: Validazione Prenotazione](#scdgp5)

3.5.5.2 [SCD\_GP1: Richiesta Prenotazione](#scdgp1)

3.5.6 [Interfaccia utente e mock-ups](#utentemockups)

3.5.6.1 [UI\_1: Homepage](#ui1)

3.5.6.2 [UI\_2: Registrazione](#ui2)

3.5.6.3 [UI\_3: Login](#ui3)

3.5.6.4 [UI\_4: Area Personale](#ui4)

3.5.6.5 [UI\_5: Richiesta Prenotazione 1](#ui5)

3.5.6.6 [UI\_6: Richiesta Prenotazione 2](#ui6)

3.5.6.7 [UI\_7: Visualizza Prenotazioni 1](#ui7)

3.5.6.8 [UI\_8: Visualizza Prenotazioni 2](#ui8)

3.5.6.9 [UI\_9: Visualizza coda](#ui9)

3.5.6.10 [UI\_10: Elimina Prenotazione 1](#ui10)

3.5.6.11 [UI\_11: Elimina Prenotazione 2](#ui11)

3.5.6.12 [UI\_12: Convalida](#ui12)

3.5.6.13 [UI\_13: Accettazione Prenotazione](#ui13)

**4.** [**Glossario**](#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. **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. **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. **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. **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. **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.

1. **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.

* 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.



1. **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** **Requisisti 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 mendo 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**

|  |  |
| --- | --- |
| Nome Scenario | SC\_GP-1: Richiesta prenotazione |
| Attori Principali | Angelo: Utente Generico |
| Flusso di eventi | 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 |

**3.5.1** **Scenari**

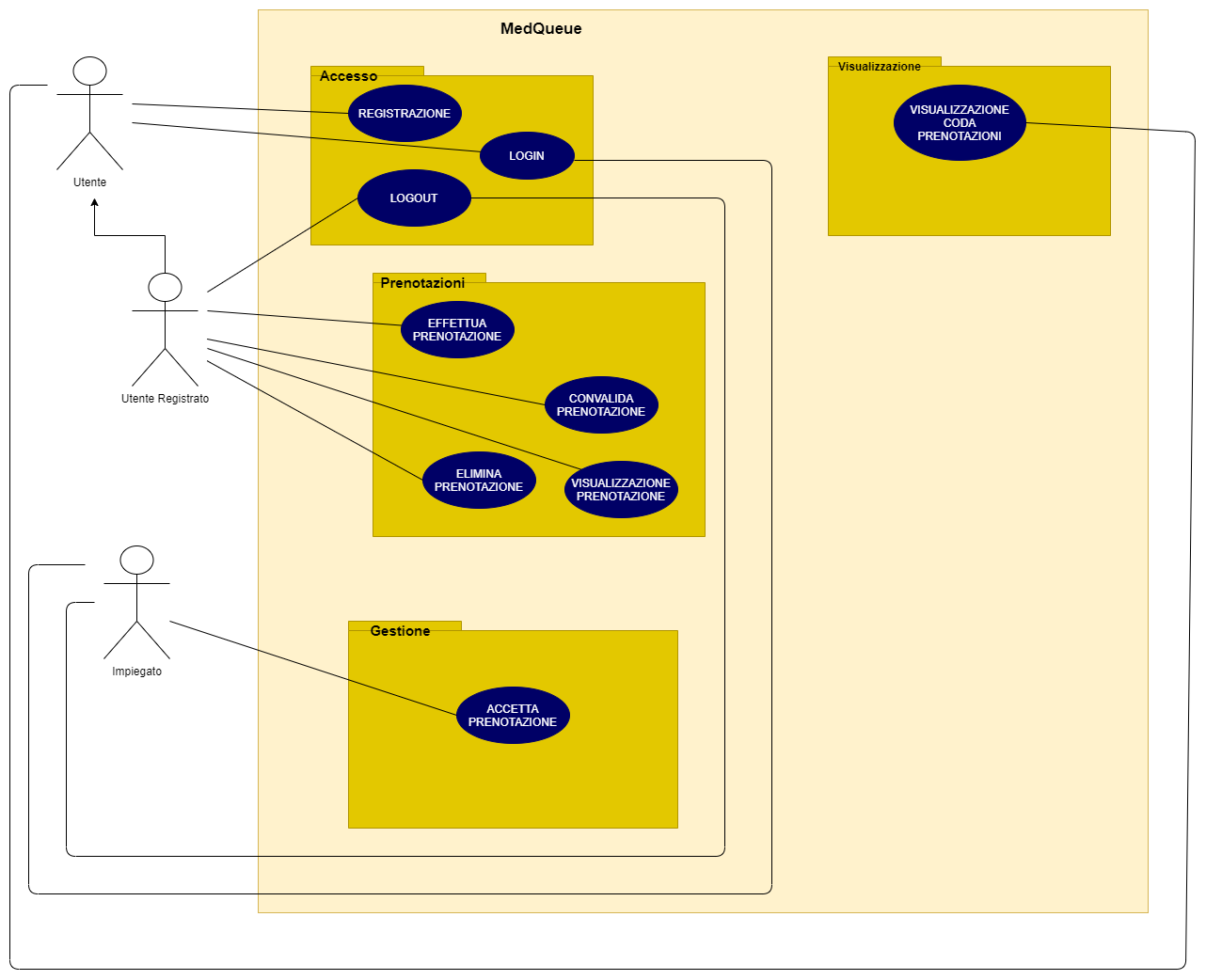
|  |  |
| --- | --- |
| Nome Scenario | SC\_GP-2: Eliminazione prenotazione |
| Attori Principali | Angelo: Utente Generico |
| Flusso di eventi | 1. Angelo accede alla sua area personale con l’intenzione di voler eliminare una prenotazione presa, e clicca su visualizza prenotazioni 2. Il sistema mostrerà ad Angelo tutte le prenotazioni ancora in attesa di convalida 3. Angelo seleziona la prenotazione da voler eliminare e clicca su elimina 4. Il sistema mostra ad Angelo un messaggio di conferma. 5. Angelo conferma la sua intenzione di voler eliminare la prenotazione 6. 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 Generico |
| Flusso di eventi | 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 | Angelo: Utente Generico |
| Flusso di eventi | 1. Angelo 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 Angelo l’elenco delle strutture disponibili. 3. Angelo seleziona l’ospedale di Napoli. 4. Il sistema Mostra ad Angelo 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 | 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**

****

* + - 1. **UC\_VC-1: Visualizzazione Coda**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_VC-1* | | | | *Visualizzazione Coda Prenotazioni* | *Data* | | *03/12/20* |
| *Vers.* | *0.00.005* | |
| *Autore* | *Afeltra Angelo* | |
| **Descrizione** | | | | *Lo UC fornisce la funzionalità per visualizzare la coda dell’ufficio* | | | |
| **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 deicide 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 | | | |
| **II 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**  *UC\_GP-3* | | | | *Validazione Prenotazione* | *Data* | | *01/12/2020* |
| *Vers.* | *0.00.003* | |
| *Autore* | *Amato Adriano* | |
| **Descrizione** | | | | *Lo UC fornisce le funzionalità per la validazione della propria prenotazione* | | | |
| **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. | | | |
| **II 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**

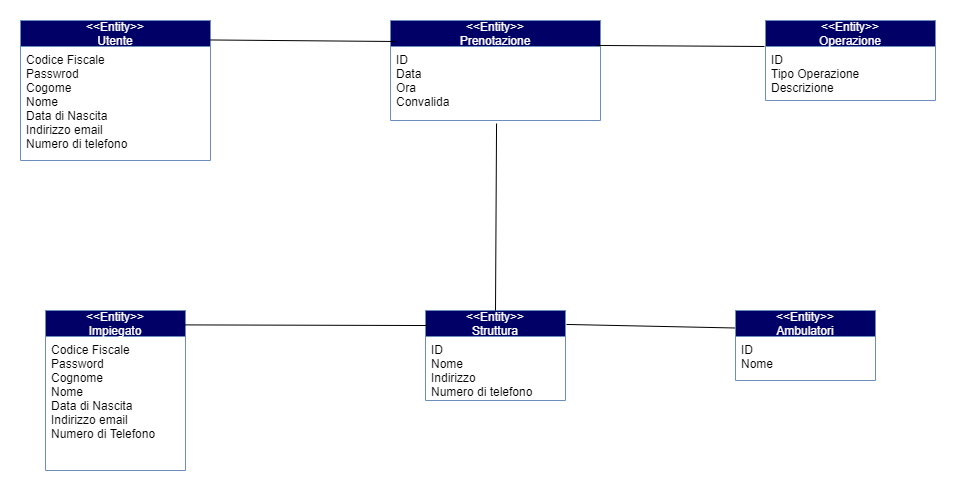
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_GPI-1* | | | | *Accettazione prenotazione* | *Data* | | *01/12/2020* |
| *Vers.* | *0.00.004* | |
| *Autore* | *Rapa Giovanni* | |
| **Descrizione** | | | | *Lo UC fornisce la funzionalità di accettazione delle prenotazioni* | | | |
| **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 raccoglitore 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 raccoglitore 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 |
| PrenotazoniRepository | 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**

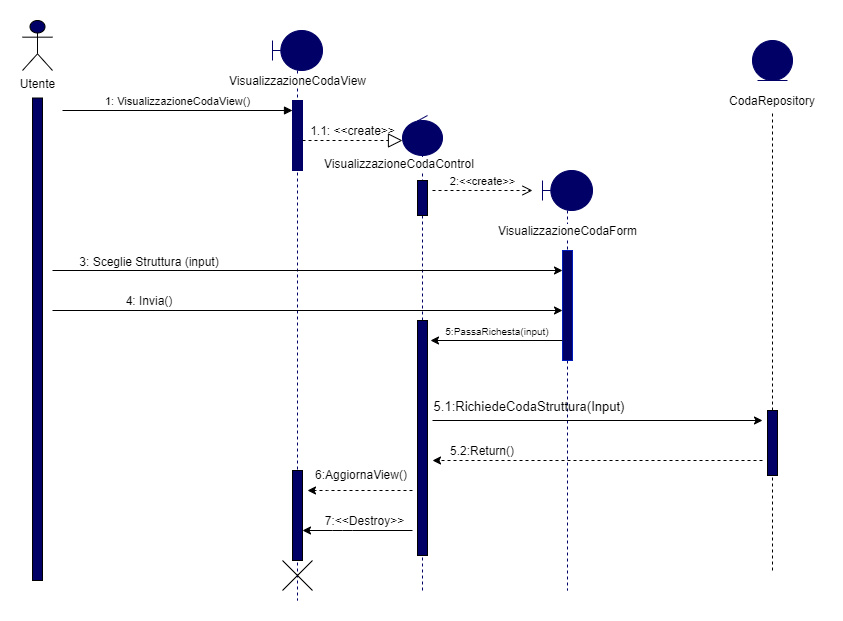
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente | Entity | Entità che modella un utente |
| UtenteRepository | Entity | Entità che modella un raccoglitore di utenti |
| RegistrazioneAccountControl | Control | Control che coordina le operazioni per la registrazione di 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**

****

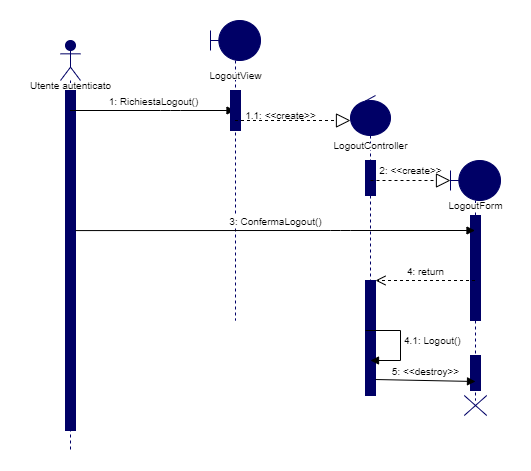
**3.5.4.6** **SD\_GP-4 Visualizzazione Prenotazione**

****

** 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.

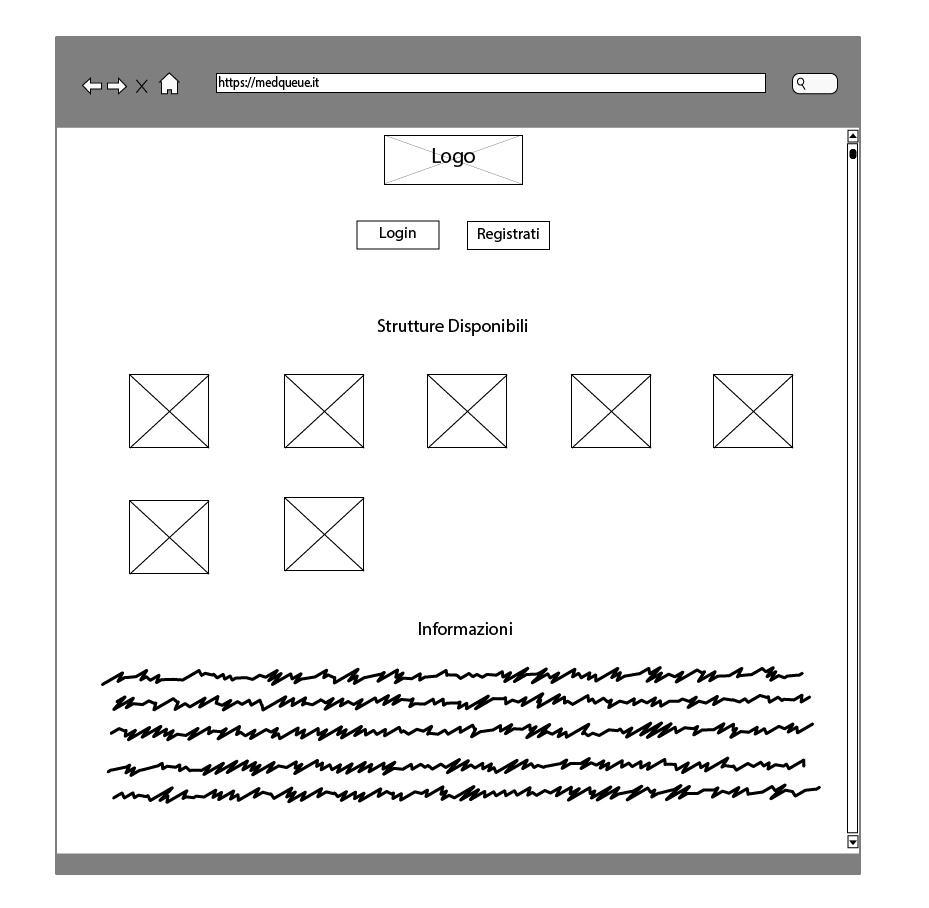
**3.5.6** **Interfaccia utente e Mock-ups**

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



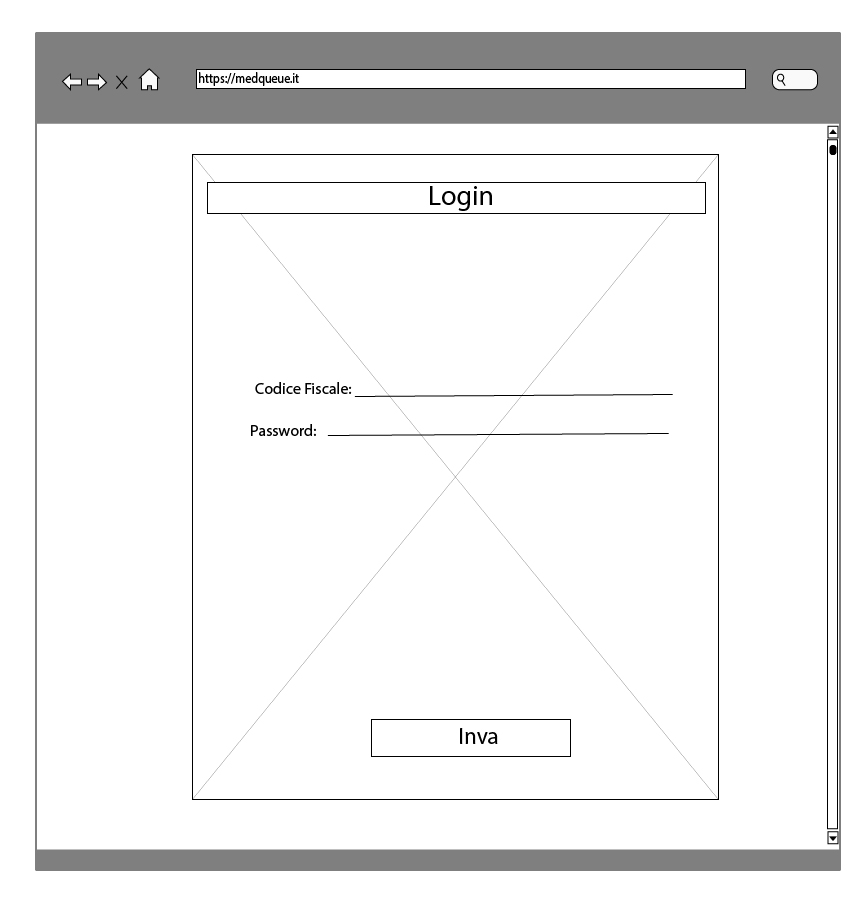
**3.5.6.1** **UI\_1: Homepage**



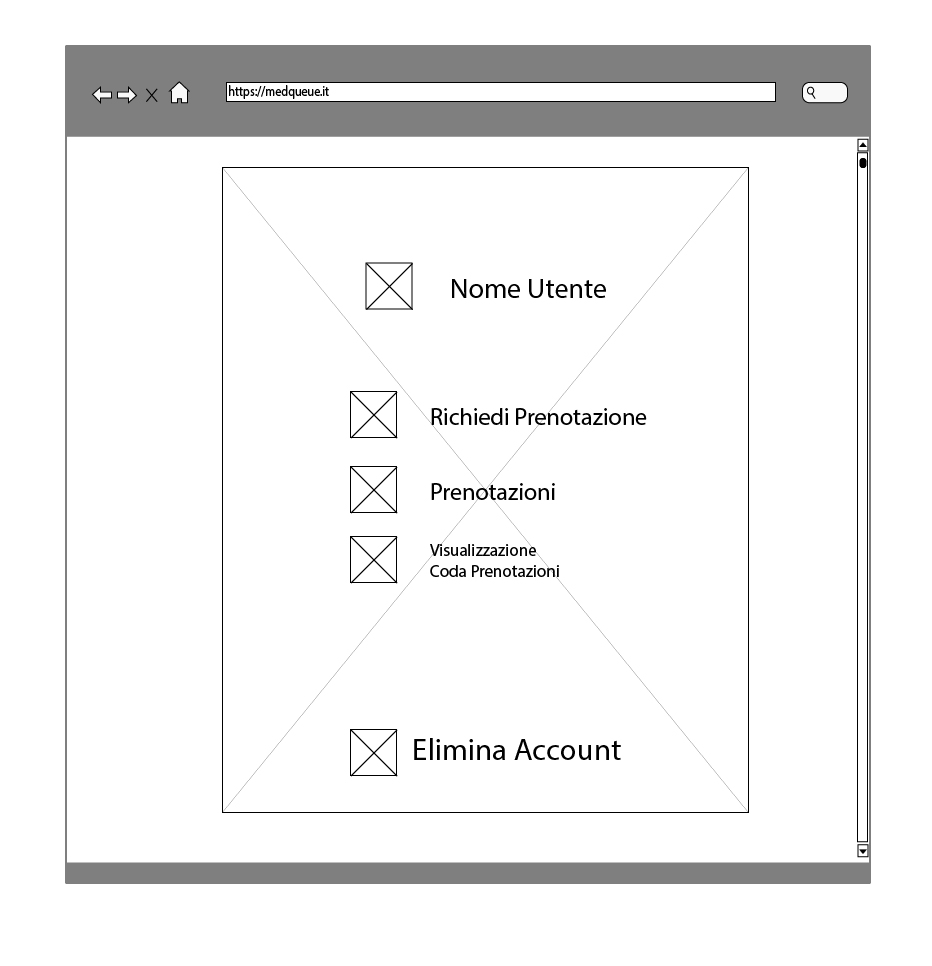
**3.5.6.2** **UI\_2: Registrazione**



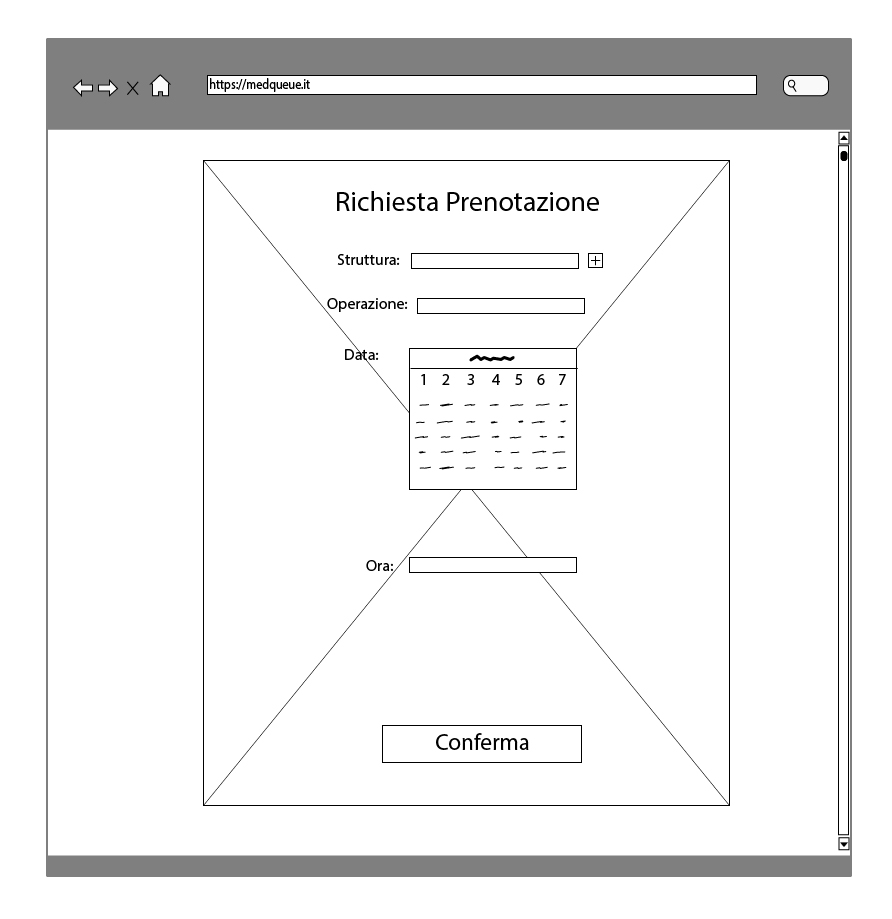
**3.5.6.3** **UI\_3: Login**



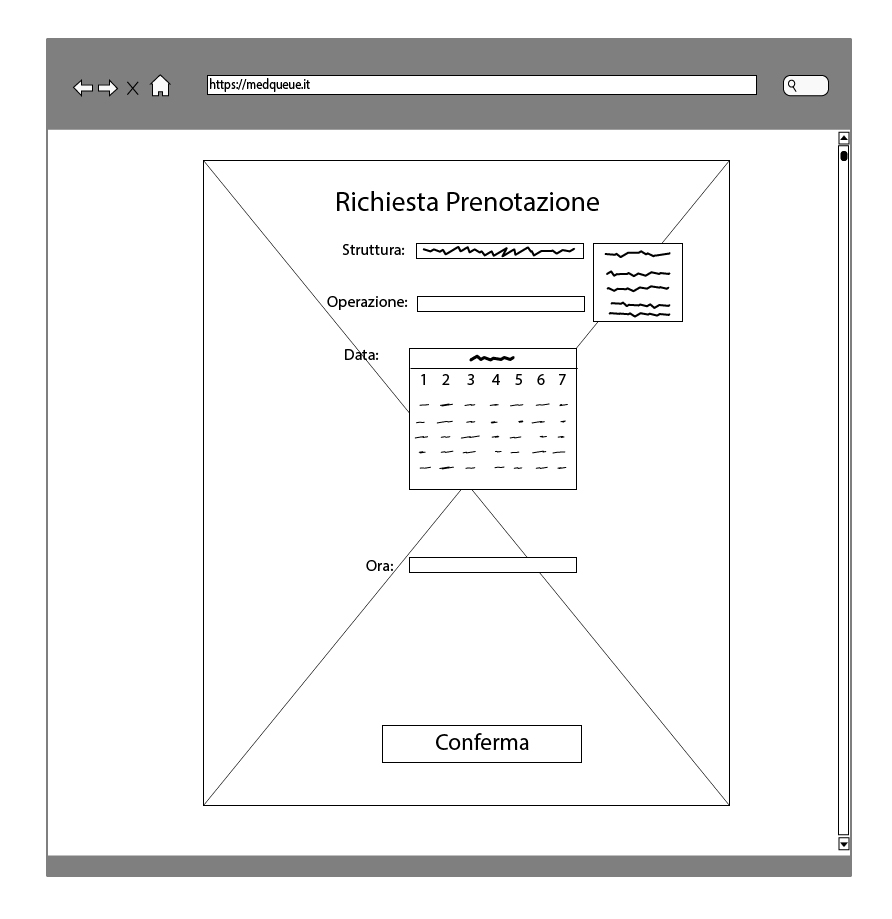
**3.5.6.4** **UI\_4: Area Personale**

****

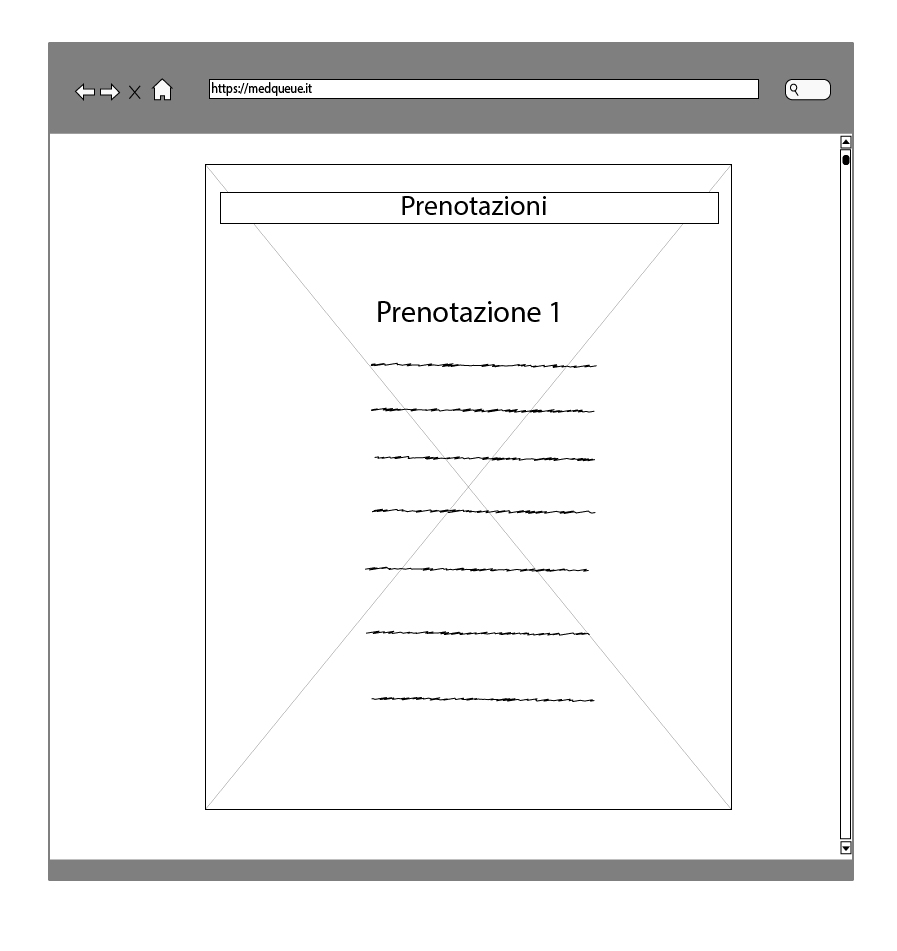
**3.5.6.5** **UI\_5: Richiesta Prenotazione 1**



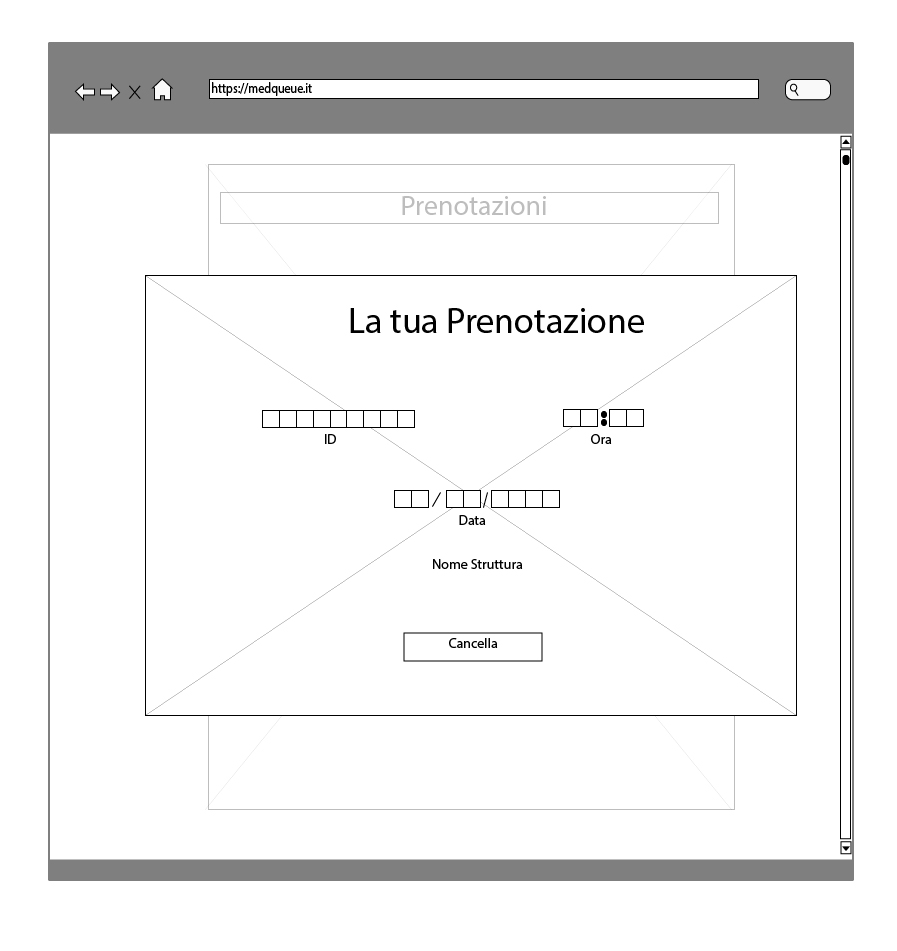
**3.5.6.6** **UI\_6: Richiesta Prenotazione 2**



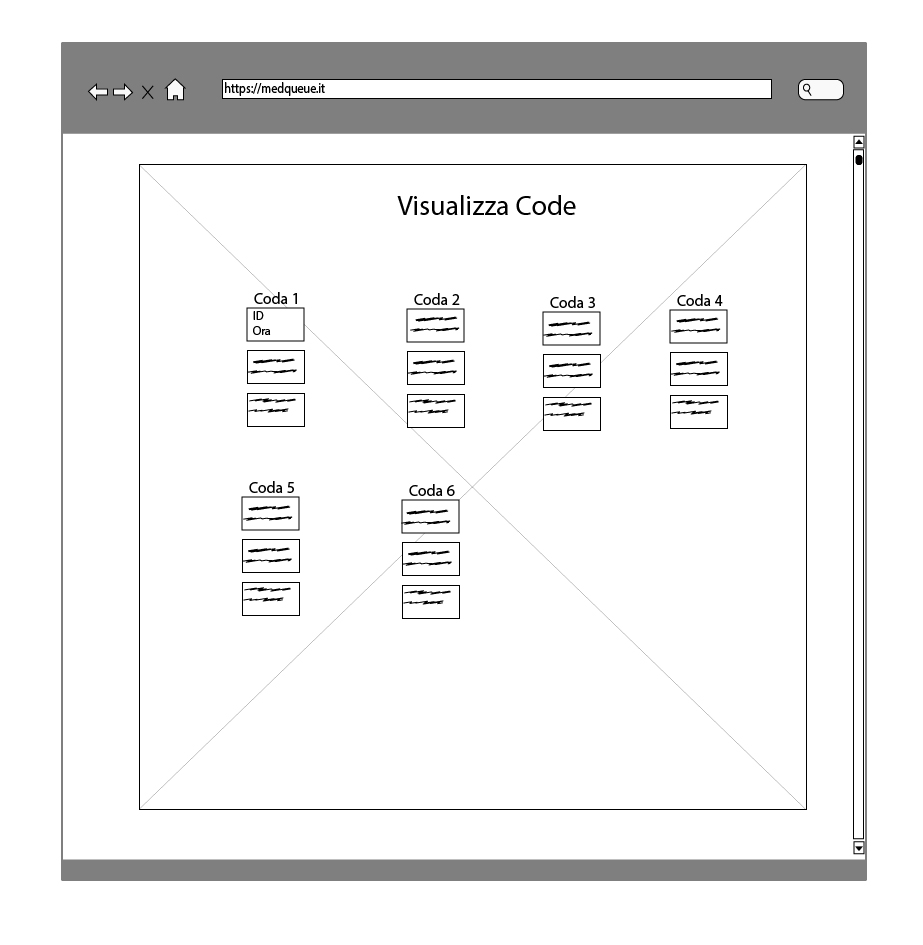
**3.5.6.7** **UI\_7: Visualizza Prenotazioni 1**



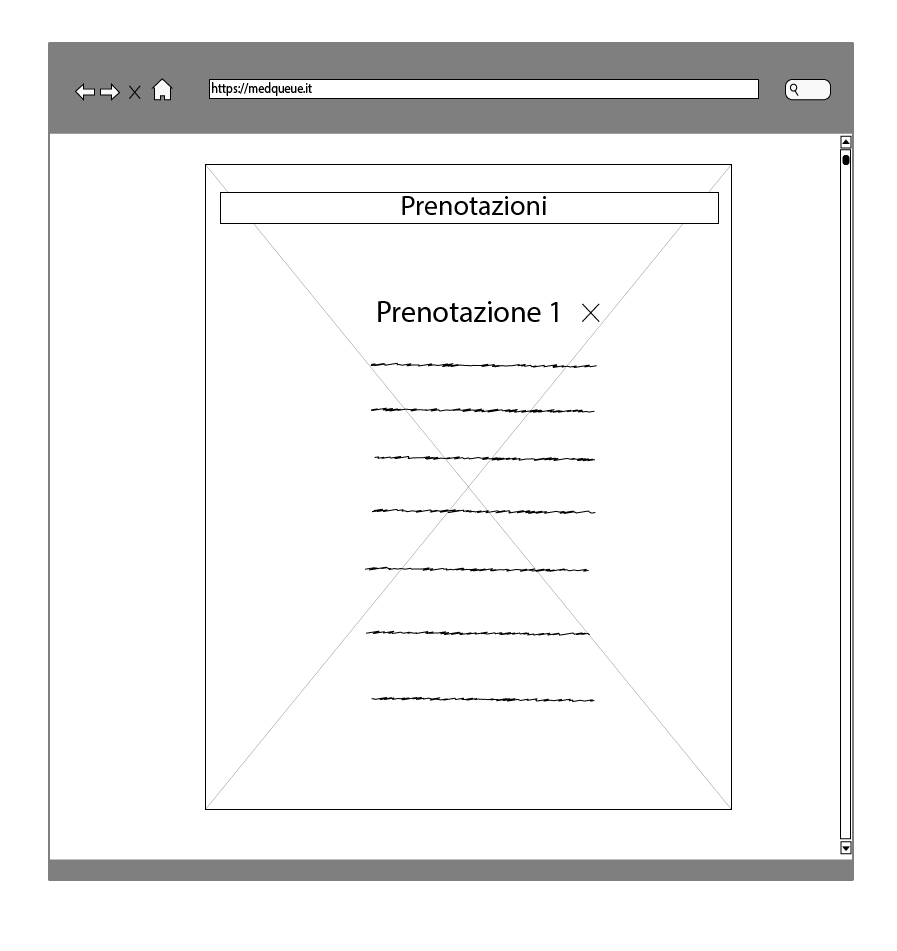
**3.5.6.8** **UI\_8: Visualizza Prenotazioni 2**



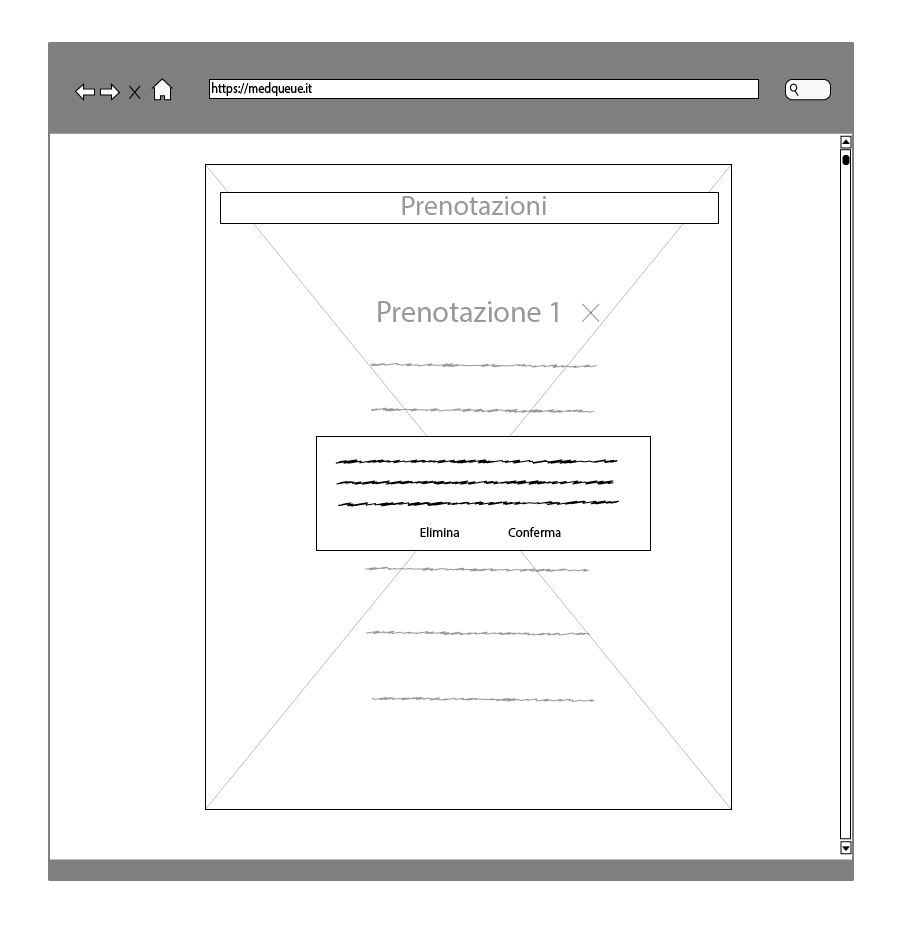
**3.5.6.9** **UI\_9: Visualizza coda**



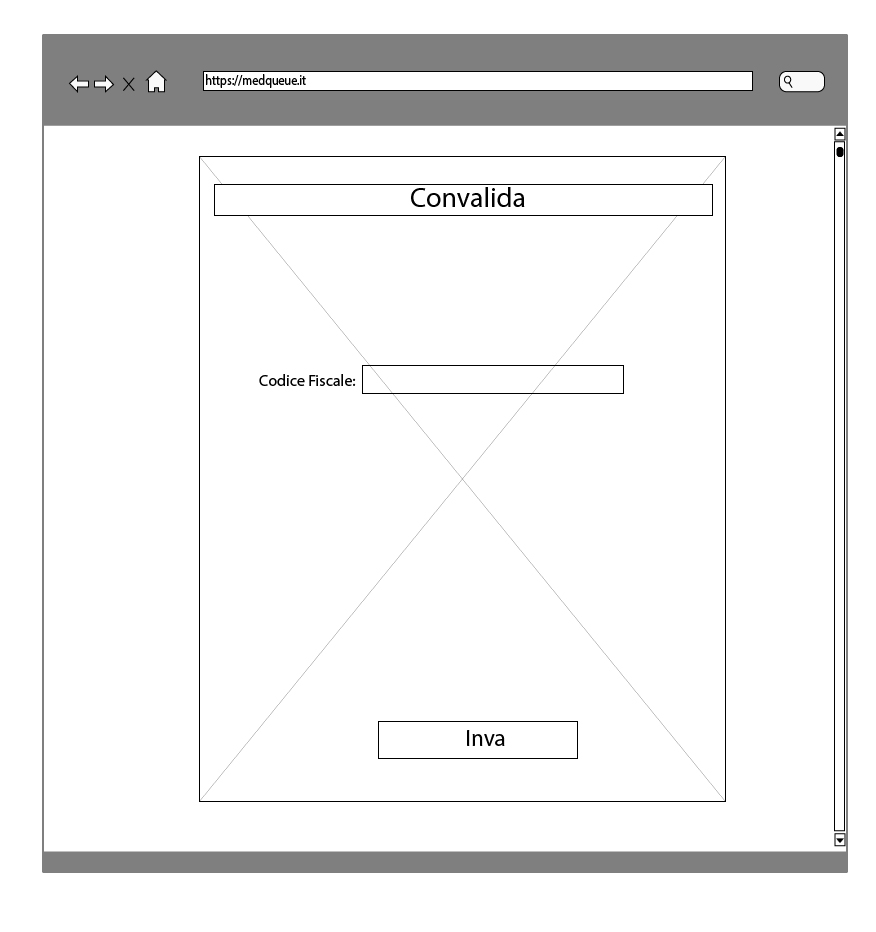
**3.5.6.10** **UI\_10: Elimina Prenotazione 1**



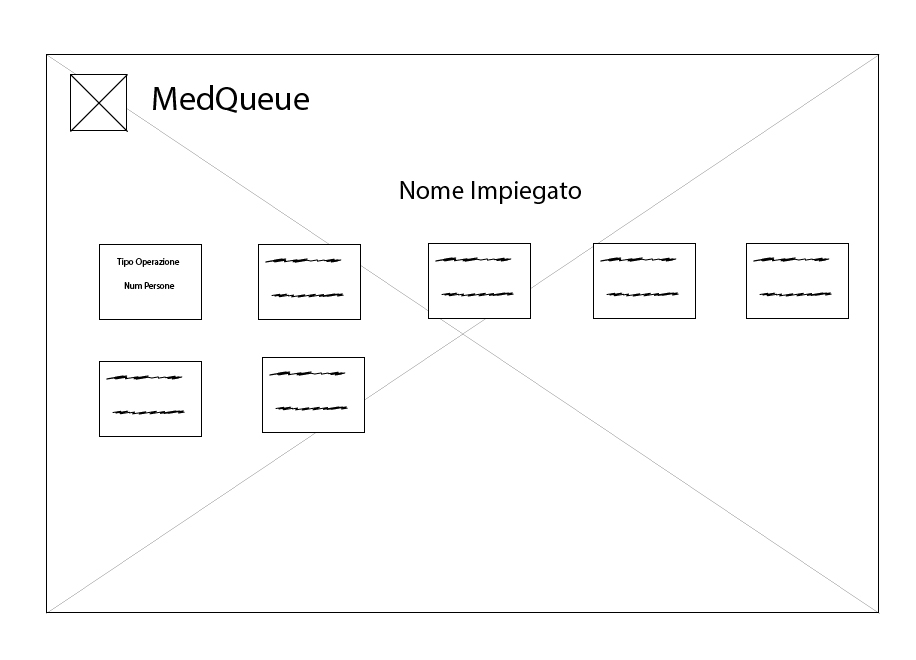
**3.5.6.11** **UI\_11: Elimina Prenotazione 2**



**3.5.6.12** **UI\_12: Convalida**



**3.5.6.13** **UI\_13: Accettazione Prenotazione**



**3.6 Matrice di tracciabilità**

****

****

****

**4.** **Glossario**

**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, Sopportabilità 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.