

RAD

Requirament Analysis Document

MedQueue

Afeltra Angelo

Amato Adriano

Fucile Andrea

Rapa Giovanni

Sommario

1. **Introduzione**
   1. Scopo del sistema
   2. Ambito del sistema
   3. Obiettivi e criteri di successo del sistema
   4. Definizione, acronimi e abbreviazioni
   5. Panoramica
2. **Sistema corrente**
3. **Sistema proposto**
   1. Diagramma delle attività
   2. Requisiti funzionali
      1. Requisiti per la visualizzazione code
      2. Requisiti per la gestione delle prenotazioni
      3. Requisiti per la gestione dell’utente
      4. Requisiti per la gestione dell’impiegato
   3. Requisisti non funzionali
      1. Usabilità
      2. Affidabilità
      3. Prestazioni
      4. Sostenibilità
   4. Pseudo Requisiti

3.4.1 Implementazione

* 1. Modello di sistema
     1. Scenari
     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\_GP-4: Accettazione Prenotazioni

3.5.2.5 UC\_GP-2: Eliminazione Prenotazione

* + 1. Modello ad Oggetti

3.5.3.1 Entità di dominio

3.5.3.2 OM\_VC: Visualizzazione Code

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\_GP-4: Accettazione Prenotazione

3.5.3.7 OM\_GP-5: 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

**Glossario**

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.

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’interfaccia 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, dovrà 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, il sistema lo metterà 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 il sistema 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.

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**

Il software che presentiamo parte 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 diagrams 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 Prenotazioni

Il sistema dovrà offrire la possibilità all’utente di visualizzare la coda delle prenotazioni

* RF-VC-2: Visualizzazione Presenze

Il sistema dovrà offrire la possibilità all’utente e all’impiegato di visualizzare la coda di presenza

**3.2.2 Requisiti per la gestione delle prenotazione**

* RF-GP-1: Richiesta prenotazione

Il sistema dovrà fornire all’utente la possibilità di effettuare una prenotazione

* RF-GP-2: Eliminazione prenotazione

Il sistema dovrà fornire all’utente la possibilità di eliminare una prenotazione effettuata

* RF-GP-3 Validazione prenotazione

Il sistema dovrà fornire all’utente la possibilità di validare la propria prenotazione per inserirlo nella coda di presenza

* RF-GP-4: Accettazione prenotazione

Il sistema dovrà fornire all’Impiegato la possibilità di accettare(servire) una prenotazione dalla coda di presenza

* RF-GP-5:Visualizzazione prenotazione

Il sistema dovrà fornire all’utente la possibilità di visualizzare le proprie prenotazioni

**3.2.3 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

**3.2.4 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

**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**

**3.5.1 Scenari**

|  |  |
| --- | --- |
| Nome Scenario | SC\_VC-1: Visualizzazione Coda |
| Attori Principali | Angelo: Utente Generico |
| Flusso di eventi | 1. Angelo si connette al sistema per visualizzare la coda di presenza, dopo esseri autenticato entra nella propria area personale e clicca sull’icona relativa alla coda di presenza 2. Il sistema mostra ad angelo la coda di presenza |

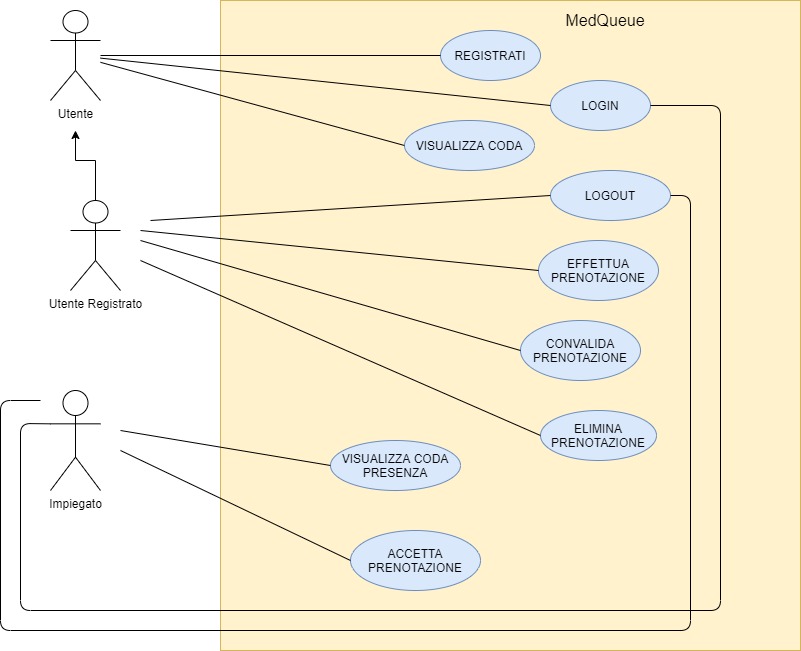
|  |  |
| --- | --- |
| Nome Scenario | SC\_GP-1: Richiesta prenotazione |
| Attori Principali | Angelo: Utente Generico |
| Flusso di eventi | 1. Angelo si connette al sistema con l’intenzione di voler prendere una prenotazione, esegue l’autenticazione e clicca sul pulsante richiedi prenotazione. 2. Il sistema chiede ad angelo di selezionare il tipo di prenotazione che si desidera 3. Angelo seleziona come tipo di operazione “Analisi Oculistica”. 4. Il sistema mostra ad Angelo tutte le strutture che eseguono l’analisi oculistica. 5. Angelo sceglie la struttura “Primo Policlinico di Napoli” 6. Il sistema mostrerà ad Angelo un calendario con giorni e orari in cui è possibile effettuare una prenotazione. 7. Angelo sceglie di effettuare una prenotazione per il giorno 12 Dicembre alle ore 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 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\_GP-4: 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 “Visita Oculistica” 6. Il sistema mostrerà a Giovanni le prenotazioni nella coda di presenza di tipo “Visita Oculistica” 7. Giovanni clicca sul pulsante “Accetta Prenotazione” 8. Il sistema mostra il codice fiscale della persona da servire e elimina la prenotazione dalla coda di presenza |

**3.5.2 Use Case**

****

**3.5.2.1 UC\_VC-1: Visualizzazione Coda Presenza**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_VC-1* | | | | *Visualizzazione Coda Presenza* | *Data* | | *19/11/20* |
| *Vers.* | *0.00.002* | |
| *Autore* | *Afeltra Angelo* | |
| **Descrizione** | | | | *Lo UC fornisce la funzionalità per visualizzare la coda dell’ufficio* | | | |
| **Attore Principale** | | | | Giovanni (utente) | | | |
| **Attori secondari** | | | | NA | | | |
| **Entry Condition** | | | | L’utente deve essere registrato ed autenticato | | | |
| **Exit condition**  On success | | | | L’utente visualizza la schermata della coda di presenza | | | |
| **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: | | Giovanni dopo essere entrato nella sua area personale decide di visualizzare la coda  Giovanni fa click sull’icona relativa alla coda di presenza. | | | | |
| 2 | Sistema: | | Il sistema consulta il database, vede le prenotazioni convalidare in attesa di essere accettate e le mostra all’utente | | | | |
|  | | | | | | | |
| **I Scenario/Flusso di eventi Alternativo:** Viene conclusa un’operazione durante la visualizzazione della coda | | | | | | | |
| **2.1** | | **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 | | | | | | | |
| **2.2** | | **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* | | *19/11/2020* |
| *Vers.* | *0.00.002* | |
| *Autore* | *Fucile Andrea* | |
| **Descrizione** | | | | | | *Lo UC fornisce le funzionalità per la prenotazione nell’ufficio ospedaliero* | | | |
| **Attore Principale** | | | | | | **Giovanni (utente)** | | | |
| **Attori secondari** | | | | | | **NA** | | | |
| **Entry Condition** | | | | | | L’utente deve essere registrato alla piattaforma.  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 | | | | | | Non viene registrata la prenotazione sul server 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 e 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 | | | | | |
| 4 | Sistema: | | | Il Sistema genera in base a 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 un calendario giorni e orari disponibili per poter prendere una prenotazione | | | | | |
| 7 | Attore: | | | L’utente sceglie data e ora | | | | | |
| 8 | Sistema: | | | Il sistema aggiunge la prenotazione alla coda delle prenotazioni ed invia una mail di conferma 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 ci sia, 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* | | *19/11/2020* |
| *Vers.* | *0.00.002* | |
| *Autore* | *Amato Adriano* | |
| **Descrizione** | | | | *Lo UC fornisce le funzionalità per la validazione della propria prenotazione* | | | |
| **Attore Principale** | | | | **Utente (Giovanni)** | | | |
| **Attori secondari** | | | |  | | | |
| **Entry Condition** | | | | L’utente ha una prenotazione effettuata sul sito per quel giorno | | | |
| **Exit condition**  On success | | | | L’utente viene inserito nella coda di presenza per essere servito | | | |
| **Exit condition**  On failure | | | | Il sistema informa l’utente che non ha una prenotazione | | | |
| **Rilevanza/User Priority** | | | | Alta | | | |
| **Frequenza stimata** | | | | 120/giorno | | | |
| **Extension point** | | | |  | | | |
| **Generalization of** | | | |  | | | |
| **Flusso di Eventi Principale/Main Scenario** | | | | | | | |
| 1 | Attore: | | L’utente arriva all’ufficio e si avvicina al totem che chiede il suo codice fiscale, l’utente lo inserisce | | | | |
| 2 | Sistema: | | Il sistema individua l’utente dal suo codice fiscale e controlla se ha una prenotazione quel giorno, in caso sia così lo aggiunge alla coda di presenza e gli comunica che verra segnalato a display il suo turno | | | | |
| 3 | Attore: | | L’attore si mette in attesa del 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 sia l’ora della prenotazione. Se essa è entro 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/vede che è errato, informa l’utente tramite video dell’errore. | | | |

**3.5.2.4 UC\_GP-4: Accettazione Prenotazioni**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_GP-4* | | | | *Accettazione prenotazioni* | *Data* | | *19/11/2020* |
| *Vers.* | *0.00.002* | |
| *Autore* | *Rapa Giovanni* | |
| **Descrizione** | | | | *Lo UC fornisce la funzionalità di accettazione delle prenotazioni* | | | |
| **Attore Principale** | | | | **Impiegato – Ernesto** | | | |
| **Attori secondari** | | | | NA | | | |
| **Entry Condition** | | | | L’impiegato si e 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 accedere alla pagina di gestione della coda | | | | |
| 2 | Sistema: | | Il sistema mostra la coda di presenza per i tipi di prenotazione scelti dall’impiegato aggiornata | | | | |
| 3 | Impiegato: | | L’impiegato visualizza la coda e clicca su servi il prossimo cliente | | | | |
| 4 | Sistema | | Il sistema preleva dalla coda la prenotazione, la elimina e stampa a video il codice fiscale 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 | | | |
|  | | | |  | | | |

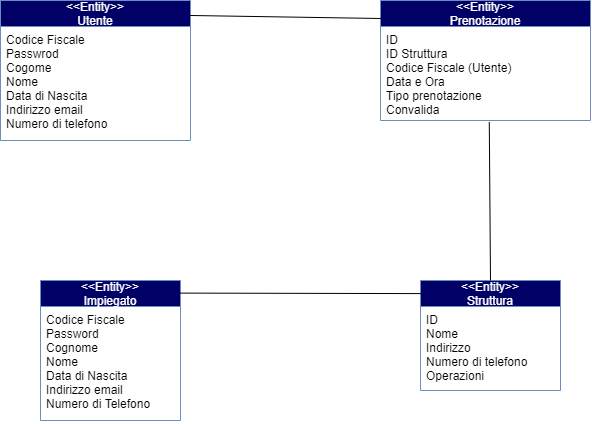
**3.5.2.5 UC\_GP-2: Eliminazione Prenotazione**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_GP-2* | | | | *Eliminazione Prenotazione* | *Data* | | *26/10/2020* |
| *Vers.* | *0.00.001* | |
| *Autore* | *Amato Adriano* | |
| **Descrizione** | | | | *Lo UC fornisce le funzionalità per annullare una prenotazione precedentemente effettuata* | | | |
| **Attore Principale** | | | | **Utente (Giovanni)** | | | |
| **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, si visualizza 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 il suo 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.

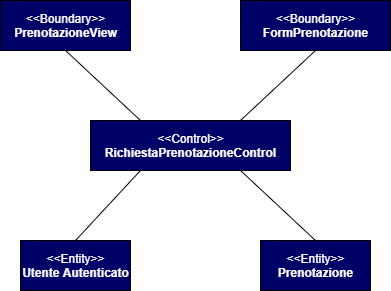
**3.5.3.1 Entità di dominio**



|  |  |  |
| --- | --- | --- |
| 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 |

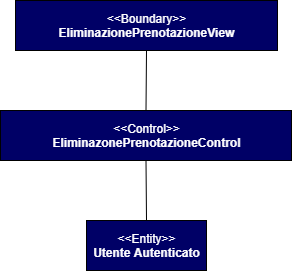
**3.5.3.2 OM\_VC: Visualizzazione Code**

**3.5.3.3 OM\_GP-1: Richiesta Prenotazione**

****

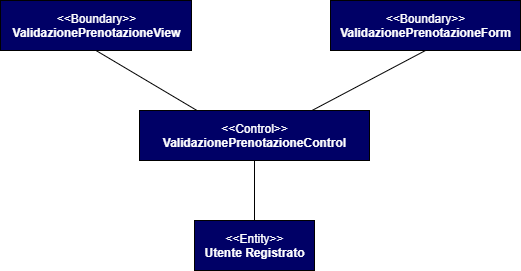
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente Autenticato | Entity | Entità che modella un utente autenticato |
| Prenotazione | Entity | Entità che modella una prenotazione dell’utente |
| RichiestaPrenotazioneControl | Control | Control che coordina le operazioni 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**

****

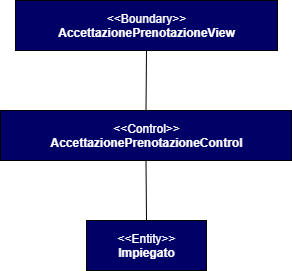
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente Autenticato | Entity | Entità che modella un utente autenticato |
| EliminazionePrenotazioneControl | Control | Control che coordina le operazioni relative alla eliminazione di una prenotazione |
| EliminazionePrenotazioneView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

**3.5.3.5 OM\_GP-3: Validazione Prenotazione**

****

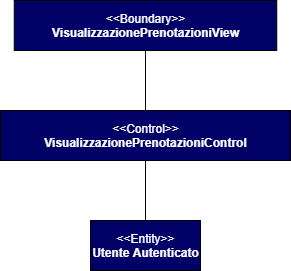
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente Registrato | Entity | Entità che modella un utente registrato |
| ValidazionePrenotazioneControl | Control | Control che coordina le operazioni relative alla validazione di una prenotazione |
| ValidazionePrenotazioneView | 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\_GP-4: Accettazione Prenotazione**

****

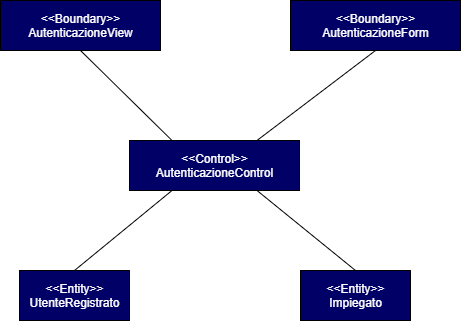
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Impiegato | Entity | Entità che modella un impiegato |
| AccettazionePrenotazioneControl | Control | Control che coordina le operazioni relative alla accettazione della prossima prenotazione in coda |
| AccettazioePrenotazioneView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

**3.5.3.7 OM\_GP-5: Visualizzazione Prenotazione**

****

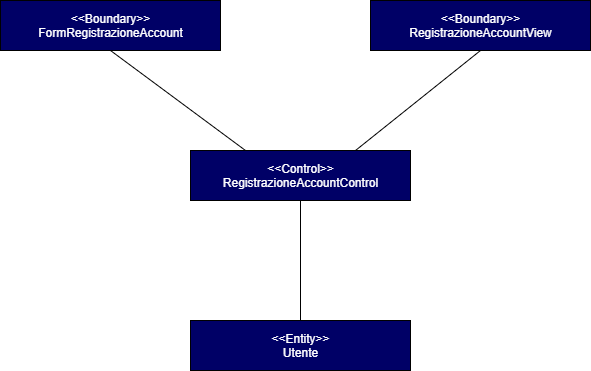
|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente Registrato | Entity | Entità che modella un utente autenticato |
| VisualizzazionePrenotazioniControl | Control | Control che coordina le operazioni relative alla visualizzazione delle prenotazioni del utente |
| VisualizzazionePrenotazioneView | 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 |
| Impiegato | Entity | Entità che modella un impiegato |
| 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 |
| 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 |