

RAD

Requirament Analysis Document

MedQueue

Afeltra Angelo

Amato Adriano

Fucile Andrea

Rapa Giovanni

Sommario

1. **Introduzione**
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

**3.1 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: Il sistema dovrà offrire la possibilità all’utente di visualizzare la coda delle prenotazioni
* RF-VC-2: 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-GU-1:Autenticazione

Il sistema dovrà fornire all’impiegato la possibilità di autenticarsi

* RF-GU-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 prenotazione del 12 dicembre, e clicca su visualizza coda prenotazione 2. Il sistema mostra ad Angelo la coda per il giorno corrente e un calendario in cui selezionare i giorni per visualizzare altre code 3. Angelo seleziona nel calendario il giorno 12 dicembre 4. Il sistema mostra ad Angelo la coda di prenotazione del giorno 12 dicembre |

|  |  |
| --- | --- |
| 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 ed esegue l’autenticazione 2. Il sistema presenta ad Angelo un calendario in cui selezionare il giorno per il quale si desidera una prenotazione e una coda di prenotazione per quel giorno 3. Angelo decidere di prendere una prenotazione per il giorno 12 Dicembre 4. Il sistema inserisce Angelo nella coda di prenotazione per il giorno 12 Dicembre e gli restituisce un biglietto di prenotazione |

|  |  |
| --- | --- |
| 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 si connette al sistema per autenticarsi e iniziare la propria giornata di lavoro 2. Il sistema mostra a Giovanni la coda di presenza 3. Giovanni seleziona la prima persona nella coda di presenza per accettare la prenotazione 4. Il sistema accetta la prenotazione la elimina dalla coda di prenotazione e mostra la persona che verrà servita |

**3.5.2 Use Case**