Università degli studi di Salerno  
Corso di Ingegneria del Software

**RooManageR**

Presentazione



11/04/2016

Coordinatore del progetto:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Andrea De Lucia |  |

Partecipanti:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Gargiulo Emanuele | 0512102244 |
| Izzo Giandomenico | 0512102292 |

|  |
| --- |
| Scritto da: Gargiulo Emanuele, Izzo Giandomenico |

Sommario

[Cos’è RooManager? 4](#_Toc448151964)

[Perché RooManager? 4](#_Toc448151965)

[Panoramica sulla documentazione 5](#_Toc448151966)

[**Documento di Analisi dei Requisiti** 5](#_Toc448151967)

[**Requisiti Funzionali:** 5](#_Toc448151968)

[**Requisiti Non Funzionali:** 6](#_Toc448151969)

[**Modello a Oggetti** 6](#_Toc448151970)

[**Modello Dinamico** 6](#_Toc448151971)

[**Descrizione dell’interfaccia grafica** 6](#_Toc448151972)

[**System Design Document** 7](#_Toc448151973)

[**Panoramica del Sistema** 7](#_Toc448151974)

[**Vincoli di Design** 7](#_Toc448151975)

[**Architettura del sistema e del database** 7](#_Toc448151976)

[**Sicurezza e controllo degli accessi** 8](#_Toc448151977)

[**Flusso di Controllo Globale** 8](#_Toc448151978)

[**Configuration** 8](#_Toc448151979)

[**Start up and Shutdown** 8](#_Toc448151980)

[**Exception Use Cases** 9](#_Toc448151981)

# Cos’è RooManager?

RooManager è un software che si propone di essere semplice e veloce e di permettere la gestione di tutte le funzioni indispensabili alla gestione di una o più strutture.  
Saranno presenti due diverse modalità, una per la tipologia d’utente proprietario e una per la tipologia d’utente portinaio.

L’utente portinaio avrà a disposizione un sottoinsieme delle funzioni del proprietario.

L’utente verrà portato alla schermata contenente le funzioni relative alla sua mansione in modo automatico in seguito al login.

In particolare sarà possibile:

* Aggiungere e rimuovere le strutture (Modalità proprietario)
* Aggiungere e rimuovere le stanze (Modalità proprietario)
* Assegnare e rimuovere un portinaio a una struttura (Modalità proprietario)
* Iniziare e terminare una visita o una permanenza in una determinata stanza
* Visualizzare l’elenco di tutte le operazioni svolte
* Visualizzare il registro clienti, contenente tutte le visite e le permanenze che sono state avviate e concluse dall’inizio dell’utilizzo di RooManager, dettagliate di data e ora di inizio e conclusione.
* Inserire, leggere e modificare le anagrafiche relative alle visite, e alle permanenze
* Inserire, leggere e modificare le anagrafiche relative ai portinai delle varie strutture (modalità proprietario)

Il software client sarà scritto in java, mentre i dati permanenti saranno archiviati in un database remoto.

# Perché RooManager?

Abbiamo scelto di creare RooManager dopo aver notato che nonostante la grande diffusione dell’informatica negli ultimi anni, molti gestori di strutture utilizzano ancora registri cartacei per segnalare ingressi ed uscite relativi a visite e permanenze, con relativo accumulo di materiale fisico nel corso del tempo e danno a carico dell’ambiente.

Molto spesso, tuttavia, in queste strutture sono presenti anche dei computer, per cui una soluzione come RooManager che non presenta requisiti tecnici particolarmente gravosi (basta un computer in grado di eseguire Java ed una connessione ad internet) è eseguibile dalla grande maggioranza delle persone, richiedendo costi di aggiornamento molto bassi o addirittura nulli.

# Panoramica sulla documentazione

## **Documento di Analisi dei Requisiti**

La fase di Analisi dei requisiti comprende i compiti che servono a determinare le condizioni da rispettare nella realizzazione di un progetto, tenendo conto dei requisiti possibilmente in conflitto da parte dei vari stakeholders e analizzando, documentando, convalidando e gestendo i requisiti software e di sistema.

La fase di analisi dei requisiti è fondamentale per determinare il successo o il fallimento di un progetto.   
I requisiti dovrebbero essere documentati, tracciabili e sufficientemente dettagliati per poter essere implementati in una fase successiva.

La fase di analisi dei requisiti include tre tipi di attività:

* **Acquisizione della conoscenza**: Si discute con i clienti e gli interessati dal progetto
* **Analisi dei requisiti**: si determina se i requisiti sono chiari, completi, consistenti e non ambigui, e si risolvono tutti i conflitti individuati.
* **Registrazione dei requisiti**: In questa fase i requisiti possono essere documentati in varie forme, che solitamente includono un sommario e spesso documenti in linguaggio naturale come casi d’uso e scenari.

Per RooManager, dopo aver introdotto il sistema e averne individuato requisiti funzionali e non funzionali, sono stati inseriti i modelli del sistema.   
Sono stati descritti due scenari che descrivono la un flusso di eventi tipico per le due tipologie di utente e sono stati descritti tutti i possibili casi d’uso del sistema per le due tipologie d’utente.

**Requisiti Funzionali:**

Il **Proprietario** deve poter:

* Creare e gestire le proprie strutture
* Creare e gestire le stanze locate all’interno delle proprie strutture
* Creare e gestire le anagrafiche
* Creare e gestire le visite all’interno delle proprie strutture
* Gestire gli inquilini all’interno delle proprie strutture
* Nominare un’anagrafica come portinaio per una o più delle sue strutture
* Visualizzare un registro operazioni
* Visualizzare il registro movimenti per ognuna delle sue strutture

Il **Portinaio** deve poter:

* Creare e gestire le visite all’interno della propria struttura
* Gestire gli inquilini all’interno della propria struttura
* Creare e gestire le anagrafiche
* Visualizzare un registro operazioni
* Visualizzare il registro movimento per la struttura presso cui è in servizio

### Requisiti Non Funzionali:

* Il tempo di risposta del server deve essere minore di 2 secondi
* Deve avere tempo di avvio inferiore a 5 secondi
* Deve poter garantire un “up-time” del 98% annuo
* L’applicazione non deve avere “freeze” dell’interfaccia
* L’applicazione non deve avere alcun “**ANR**” (**A**pplication **N**ot **R**esponding)
* Dev’essere multipiattaforma

### Modello a Oggetti

Per quanto concerne il modello ad oggetti:

* È stato inserito il class diagram che rappresenta gli entity object del sistema

### Modello Dinamico

Relativamente al modello dinamico:

* Sono stati inseriti i sequence diagram per ogni caso d’uso e per ogni eccezione connessa ai particolari casi d’uso
* Sono stati inseriti gli state chart diagram per le entità visita e permanenza, le uniche che possiedono un vero stato (Iniziata – in corso – terminata) all’interno del sistema.

### Descrizione dell’interfaccia grafica

Per la descrizione dell’interfaccia grafica:

* Sono stati inseriti screenshot dell’intero sistema ed è stato descritto il navigational path che le due tipologie di utenti potranno seguire per svolgere le varie operazioni.

## **System Design Document**

Il processo di System Design serve a definire l’architettura, i componenti, i moduli e i dati necessari a soddisfare i requisiti di un sistema.

La fase di System Design può essere vista come l’applicazione della teoria dei sistemi allo sviluppo di un prodotto.

C’è sovrapposizione con le discipline di analisi dei sistemi, architettura dei sistemi ed ingegneria dei sistemi.

Questo documento descrive il sistema a livello architetturale, includendo i sottosistemi e i loro servizi, il mapping hardware, la gestione dei dati, la matrice di controllo degli accessi, la struttura di controllo globale del software e le boundary condition.

Il documento di System Design dovrebbe definire come implementare tutti i requisiti del RAD, e dovrebbe fornire una guida di base per poi scendere ulteriormente in dettaglio nel processo che condurrà verso una soluzione realizzabile.

I destinatari del documento di System Design sono tutti i software architect e i liaison che fanno parte del team di sviluppo di ciascun sottosistema.

### Panoramica del Sistema

Nello scrivere il documento di System Design di RooManager, abbiamo introdotto il nostro sistema spiegando a chi è rivolto e quali sono le possibilità offerte alle due tipologie di utenti che potranno effettuare il login.

### **Vincoli di Design**

Abbiamo poi analizzato i vincoli di design, relativi nel caso specifico alle macchine su cui dovrà girare il sistema software e sui requisiti di connessione richiesti, e abbiamo individuato l’architettura del sistema, che sarà formato da:

* Un’applicazione Java che girerà sulla macchina dell’utente
* Un’applicazione server in php che interagirà con il database
* Un database relazionale remoto su cui saranno archiviati i dati

### Architettura del sistema e del database

Sono stati descritti i sottosistemi in cui è scomposta l’applicazione Java, e l’architettura del database con le relative tabelle e campi.

### Sicurezza e controllo degli accessi

Nella sezione del documento dedicata alla sicurezza, è stato descritto come in fase di login l’utente dovrà inserire il proprio codice fiscale e password, e in base alla mansione assegnatagli internamente al database sarà indirizzato in modo automatico alla modalità del programma che contiene le funzionalità per lui pensate.

In oltre, è stata rappresentata la matrice di controllo degli accessi specificando per ogni coppia Tipologia di utente – Azione la possibilità di svolgimento.

### Flusso di Controllo Globale

Nella sezione dedicata al flusso di controllo globale è stato specificato che in RooManager non esiste una sequenza di operazioni prestabilita, ma il flusso è guidato dagli eventi (event-driven), ma viene avvito un thread quando è necessario interagire con il server per inviare e ricevere risposte dal database.

Questa scelta è stata presa per evitare “freeze” dell’interfaccia grafica in attesa di una risposta.

### Configuration

Nella sezione Configuration è stato descritta la fase di creazione e distruzione per gli oggetti persistenti.

### Start up and Shutdown

Nella sezione Start up and Shutdown viene descritto quando, come e da chi dovrebbero essere avviate le diverse componenti del sistema.

### Exception Use Cases

Nella sezione Exception Use Cases è stato descritto il principale errore che può affliggere RooManager, cioè quello relativo alla comunicazione con il server, e la reazione che dovrebbe avere il sistema software nel verificarsi di tale situazione.

## **Object Design Document**