



UNIVERSITÀ DEGLI STUDI DI SALERNO

Facoltà di Scienze Matematiche Fisiche e Naturali

Dipartimento di Informatica

Ingegneria del Software

Anno Accademico 2015/16

# System Test Plan

## Componenti del gruppo

Giuseppe Di Martino	0512101162
Luca Diodato De Martino	0512102658
Francesco Napoli	0512101928
Giuseppe Giordano	0512101798

## RevisionHistory

Data	Versione	Descrizione	Autore
17/12/2015	1.0	Prima versione del System Test Plan.	Tutto il gruppo

## Sommario

<b>Componenti del gruppo</b> .....	2
<b>RevisionHistory</b> .....	2
Sommario .....	2
<b>Introduzione</b> .....	5
<b>2 Documenti correlati</b> .....	5
2.1 Relazioni con il documento di analisi dei requisiti (RAD) .....	5
2.2 Relazioni con il System Design Document (SDD) .....	5
2.3 Relazioni con l'Object Design Document (ODD) .....	6
<b>3. Overview del Sistema</b> .....	6
<b>4. Funzionalità da testare e da non testare</b> .....	6
<b>5. Criteri Pass/Failed</b> .....	7
<b>6. Approccio</b> .....	7
6.1. Testing di Unità .....	7
6.2. Testing d'integrazione .....	8
6.3. Testing di Sistema .....	8
<b>7. Sospensione e Ripresa</b> .....	8
7.1 Criteri di sospensione .....	8
7.2 Criteri di ripresa .....	8
<b>8. Materiale per il testing (Requisiti Hardware/software)</b> .....	8
<b>9. Test Cases</b> .....	9
9.1 Autenticazione .....	9

9.1.1 Login_test .....	9
9.1.2 Invalid_login_test .....	9
9.1.3 Logout_test .....	9
9.2 Gestione immobili .....	9
9.2.1 Inserimento_Immobile .....	9
9.2.2 Invalid_Inserimento_Immobile .....	10
9.2.3 Modifica_Immobile .....	10
9.2.4 Elimina_Immobile .....	11
9.2.5 Approvazione_Immobile .....	11
9.2.6 Visualizza_Immobile .....	12
9.3 Gestione Utente .....	12
9.3.1 Inserimento_Agente .....	12
9.3.2 Invalid_Inserimeto_Agente .....	13
9.3.3 Modifica_Agente .....	13
9.3.4 Elimina_Agente .....	14
9.3.5 Inserimento_Cliente .....	15
9.3.6 Invalid_Inserimeto_Cliente .....	15
9.3.7 Modifica_Cliente .....	16
9.3.8 Elimina_Cliente .....	16
9.4 Gestione Trattativa .....	17
9.4.1 Assegnazione .....	17
9.4.2 Elimina .....	17
9.4.3 Riassegna .....	18
10. Pianificazione del Testing .....	18
10.1. Determinazione dei ruoli .....	19
10.2. Determinazione dei rischi .....	19
10.3. Decomposizione gerarchica del sistema .....	19
10.4. Organizzazione delle attività di testing .....	20
10.5. Schedulazione delle attività di testing .....	20
11 Test Report .....	21
Organizzazione del Progetto .....	21



## **Introduzione**

Lo scopo di questo documento è quello di pianificare l'attività di test del sistema Immobilunisa al fine di verificare se esistono differenze tra il comportamento atteso e il comportamento osservato. In questa attività andremo a rilevare gli eventuali errori prodotti all'interno del codice, per evitare che essi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale.

Le attività di test sono state pianificate per le seguenti funzioni:

1. Autenticazione
2. Gestione Immobile(Inserimento, modifica, eliminazione, approvazione, catalogo)
3. Gestione Agente (Inserimento agente, modifica agente, elimina agente, inserisci cliente, modifica cliente, elimina cliente)
4. Gestione Trattative (validazione, elimina)

Si noti, tuttavia, che verranno testate esclusivamente le funzionalità implementate e specificate nell'ODD, oltre alla gestione dei test delle funzionalità, vengono anche pianificate le responsabilità dello staff e lo scheduling del test. La fase di testing è strettamente legata alle fasi ad essa precedenti; ogni documento, risultato delle differenti fasi di sviluppo, sarà punto di partenza indispensabile per poter effettuare un testing corretto e adeguato.

## **2 Documenti correlati**

Il test plan ha ovviamente una stretta relazione con il resto dei documenti che sono stati prodotti finora, poiché prima di passare alla fase di testing, oltre ad aver implementato il sistema nella gran parte, esso era stato pianificato nelle precedenti documentazioni. Questo quindi permette di rilevare le eventuali differenze tra ciò che si desiderava e ciò che invece il sistema fa.

### **2.1 Relazioni con il documento di analisi dei requisiti (RAD)**

La relazione tra test plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse nel RAD.

### **2.2 Relazioni con il System Design Document (SDD)**

Il test dei vari componenti deve rimanere fedele alle suddivisioni riportate nel System Design Document il più possibile.

## **2.3 Relazioni con l'Object Design Document (ODD)**

Il test d'integrazione farà quanto più riferimento possibile alle classinterfaces definite nell'ODD.

## **3. Overview del Sistema**

Come stabilito nel System Design Document la struttura del nostro sistema è divisa secondo un'architettura "Tree Tiers" cioè a tre livelli: presentationLayer, applicationLayer, storageLayer. In questo caso il livello più alto interagisce con il livello applicativo che a sua volta si occuperà di eseguire le operazioni nel database, cercando di garantire il più possibile basso accoppiamento e alta coesione tra le varie classi. Il sistema inoltre è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni. Abbiamo infatti i seguenti sottosistemi:

1. Autenticazione
2. Gestione Immobile
3. Gestione Utente
4. Gestione Trattative

Quasi ognuna delle precedenti gestioni prevede principalmente operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca e saranno proprio queste funzionalità ad essere testate nel corso della fase di testing del sistema.

## **4. Funzionalità da testare e da non testare**

Di seguito saranno elencate per ogni gestione quali sono le funzionalità che saranno testate.

1. Autenticazione
  - Login
2. Gestione Immobile
  - Inserimento
  - Modifica
  - Eliminazione
  - Validazione
  - Ricerca
3. Gestione Utente
  - Inserisci\_agente
  - Modifica\_agente
  - Elimina\_agente

Inserisci\_cliente

Modifica\_cliente

Elimina\_cliente

#### 4. Gestione Trattative

Assegna

Riassegna

Modifica

## **5. Criteri Pass/Failed**

I dati di input del test saranno suddivisi in classi di equivalenza, ovvero verranno raggruppati in insiemi dalle caratteristiche comuni, per i quali sarà sufficiente testare un solo elemento rappresentativo. Un input avrà superato un test se l'output risultante sarà quello atteso, cioè quello che è stato specificato dal membro di testing su tale test case, il membro di testing conosce quale dovrebbe essere l'output corretto.

## **6. Approccio**

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità. Seguirà il testing d'integrazione, che focalizzerà l'attenzione principalmente sul test delle interfacce delle suddette unità.

Infine verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Quest' ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

### **6.1. Testing di Unità**

Durante questa fase, verranno ricercate le condizioni di fallimento isolando i componenti ed usando test driver e test stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare. La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica BlackBox, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Al fine di minimizzare il numero dei test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case. Gli stati erronei scovati in questa, come in qualsiasi altra fase di testing, che comporteranno un fallimento del Sistema dovranno essere tempestivamente comunicati agli implementatori al fine di correggerli e ripristinare il testing al più presto.

## **6.2. Testing d'integrazione**

In questa fase si procederà all'integrazione delle componenti di una funzionalità che verranno testate nel complesso attraverso una strategia di Bottom-Up. Si passerà, poi, alla funzionalità successiva fino ad esaurire le funzionalità implementate. Quest'approccio mira principalmente a ridurre le dipendenze tra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

## **6.3. Testing di Sistema**

Lo scopo di questa fase del testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e che sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso la tecnica BlackBox.

# **7. Sospensione e Ripresa**

## **7.1 Criteri di sospensione**

La fase di Testing del sistema verrà sospesa quando si raggiungerà un compromesso tra qualità del prodotto e costi dell'attività di Testing. Il Testing verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

## **7.2 Criteri di ripresa**

In seguito ad ogni modifica o correzione delle componenti che genereranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandoci così di aver risolto effettivamente il problema.

# **8. Materiale per il testing (Requisiti Hardware/software)**

L'hardware necessario per l'attività di test è un PC, o possibilmente uno per ogni oggetto incaricato del testing, su cui sia installato il DBMS MySQL 5.0 e Apache 2 + PHP 5.0



## **9. Test Cases**

### **9.1 Autenticazione**

#### **9.1.1 Login\_test**

Condizione di entrata	L'utente deve essersi registrato in agenzia. L'utente deve avere accesso alla homepage del sito.
Flusso	1. L'utente inserisce username e password, e seleziona il pulsante "login".
Condizione di uscita	L'utente risulta loggato nel sistema.

#### **9.1.2Invalid\_login\_test**

Condizione di entrata	L'utente deve essersi registrato in agenzia. L'utente deve avere accesso alla homepage del sito.
Flusso	1. L'utente inserisce username e/o password errati, e seleziona il pulsante "login".
Condizione di uscita	Il sistema avverte l'utente dell'errore commesso.

#### **9.1.3 Logout\_test**

Condizione di entrata	L'utente deve essere loggato nel sistema.
Flusso	1. L'utente seleziona il pulsante logout.
Condizione di uscita	Il sistema mostra l'homepage principale.

### **9.2 Gestione immobili**

#### **9.2.1 Inserimento\_Immobile**

Condizione di entrata	Il cliente deve essersi loggato nel sistema.
Flusso	1. Il cliente seleziona la funzione "Vendi Immobile". 2. Il sistema risponde mostrando la schermata in cui vengono richiesti i dati dell'immobile. 3. Il cliente inserisce i dati: Descrizione, Comune, Metratura, Prezzo, Contratto, Tipo, Immagine. Preme "Invia" per confermare. 4. Il sistema riceve i dati, li inserisce nel database e notifica il corretto inserimento. 5. Il sistema inserisce l'immobile online.
Condizione di uscita	Il sistema notifica l'inserimento.

**9.2.2 Invalid\_Inserimento\_Immobile**

Condizione di entrata	Il cliente deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. Il cliente seleziona la funzione “Vendi Immobile”.</li><li>2. Il sistema risponde mostrando la schermata in cui vengono richiesti i dati dell’immobile.</li><li>3. Il cliente inserisce i dati (Descrizione, Comune, Metratura, Prezzo, Contratto, Tipo, Immagine) in modo errato o incompleto. Preme “invia” per confermare.</li><li>4. Il sistema si accorge del mancato inserimento di qualche campo o di eventuali errori.</li></ol>
Condizione di uscita	Il sistema notifica l’errore al cliente.

**9.2.3 Modifica\_Immobile**

Condizione di entrata	L’agente deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. L’agente seleziona “gestione immobili”.</li><li>2. Il sistema mostra l’elenco degli immobili presenti nel catalogo.</li><li>3. L’agente sceglie l’immobile da modificare premendo sul pulsante “visualizza”.</li><li>4. Il sistema risponde mostrando i dettagli dell’immobile scelto</li><li>5. L’agente preme sul tasto “modifica”</li><li>6. Il sistema risponde mostrando i campi dell’immobile da modificare</li><li>7. L’agente dopo aver modificato i campi preme sul pulsante “conferma”.</li></ol>
Condizione di uscita	Il sistema risponde con una finestra di conferma di avvenuta modifica e rimanda alla schermata home.

**9.2.4 Elimina\_Immobile**

Condizione di entrata	L'agente deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. L'agente seleziona "gestione immobili".</li><li>2. Il sistema mostra l'elenco degli immobili presenti nel catalogo.</li><li>3. L'agente sceglie l'immobile da eliminare premendo sul pulsante "visualizza".</li><li>4. Il sistema risponde mostrando i dettagli dell'immobile scelto</li><li>5. L'agente preme sul tasto "elimina".</li></ol>
Condizione di uscita	Il sistema risponde con una finestra di conferma e rimanda alla schermata "gestione immobili".

**9.2.5 Approvazione\_Immobile**

Condizione di entrata	L'agente deve essersi loggato nel sistema. L'amministratore deve aver assegnato l'immobile all'agente.
Flusso	<ol style="list-style-type: none"><li>1. L'agente trova la notifica "ti è stato assegnato un nuovo immobile" nella home.</li><li>2. L'agente preme "visualizza" sotto la notifica nella home.</li><li>3. Il sistema risponde mostrando i dettagli dell'immobile che gli è stato assegnato.</li><li>4. L'agente preme sul tasto "approva"</li><li>5. Il sistema risponde mostrando una notifica di avvenuta approvazione</li><li>6. L'agente chiude la notifica.</li></ol>
Condizione di uscita	Il sistema risponde rimandando alla schermata "Gestione Trattative".

### 9.2.6 Visualizza\_Immobile

Condizione di entrata	L'utente deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. L'utente seleziona gestione immobili.</li><li>2. Il sistema mostra l'elenco degli immobili presenti nel catalogo.</li><li>3. L'agente sceglie l'immobile da modificare premendo sul pulsante "visualizza".</li></ol>
Condizione di uscita	Il sistema risponde mostrando i dettagli dell'immobile scelto

## 9.3 Gestione Utente

### 9.3.1 Inserimento\_Agente

Condizione d'entrata	L'amministratore deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'amministratore preme il pulsante "RegistraAgente".</li><li>2. Il sistema visualizza una schermata dove viene richiesto di compilare i campi: nome,cognome, data di nascita, codice fiscale.</li><li>3. L'amministratore dopo aver compilato i campi preme "registra".</li><li>4. Il sistema visualizza una finestra con la conferma di avvenuta registrazione.</li><li>5. L'amministratore chiude la finestra di conferma.</li></ol>
Condizione di uscita	<ol style="list-style-type: none"><li>1. Il sistema rimanda alla schermata home.</li></ol>

### 9.3.2 Invalid\_Inserimeto\_Agente

Condizione d'entrata	L'amministratore deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'amministratore preme il pulsante "RegistraAgente".</li><li>2. Il sistema visualizza una schermata dove viene richiesto di compilare i campi: nome,cognome, data di nascita, codice fiscale in modo sbagliato.</li><li>3. L'amministratore dopo aver compilato i campi preme "registra".</li></ol>
Condizione di uscita	Il sistema visualizza i campi compilati in modo errato.

### 9.3.3 Modifica\_Agente

Condizione d'entrata	L'amministratore deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'amministratore clicca sulla funzione "gestione Agenti".</li><li>2. Il sistema risponde visualizzando tutti gli agenti registrati.</li><li>3. L'amministratore sceglie l'agente che vuole modificare e clicca sul pulsante "visualizza".</li><li>4. Il sistema visualizzerà i dettagli dell'agente.</li><li>5. L'amministratore clicca il pulsante modifica.</li><li>6. Il sistema visualizzerà i campi da modificare (codice fiscale, nome e cognome, data di nascita, email, telefono, username, password).</li><li>7. L'amministratore dopo aver modificato i campi preme il pulsante "conferma".</li><li>8. Il sistema visualizza una finestra di avvenuta conferma.</li><li>9. L'amministratore chiude la finestra.</li></ol>
Condizione d'uscita	Il sistema rimanda alla homepage.

**9.3.4 Elimina\_Agente**

Condizione d'entrata	L'amministratore deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'amministratore clicca sulla funzione "Gestione Agente".</li><li>2. Il sistema risponde visualizzando tutti gli agenti registrati.</li><li>3. L'amministratore sceglie l'agente che vuole eliminare e clicca sul pulsante "visualizza".</li><li>4. Il sistema visualizzerà i dettagli dell'agente.</li><li>5. L'amministratore clicca il pulsante elimina.</li><li>6. Il sistema visualizza una finestra di avvenuta conferma.</li><li>7. L'amministratore chiude la finestra.</li></ol>
Condizione d'uscita	Il sistema rimanda alla homepage.

**9.3.5 Inserimento\_Cliente**

Condizione d'entrata	L'agente deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'agente premere il pulsante "Registra Cliente".</li><li>2. Il sistema visualizza una schermata dove viene richiesto di compilare i campi: nome e cognome, data di nascita, codice fiscale, email, telefono, username, password.</li><li>3. L'agente dopo aver compilato i campi preme "registra".</li><li>4. Il sistema visualizza una finestra con la conferma di avvenuta registrazione.</li><li>5. L'agente chiude la finestra di conferma.</li></ol>
Condizione di uscita	Il sistema rimanda alla schermata home.

**9.3.6 Invalid\_Inserimeto\_Cliente**

Condizione d'entrata	L'agente deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'agente preme il pulsante "Registra Cliente".</li><li>2. Il sistema visualizza una schermata dove viene richiesto di compilare i campi: nome e cognome, data di nascita, codice fiscale, email, telefono, username, password in modo sbagliato.</li><li>3. L'agente dopo aver compilato i campi preme "registra".</li></ol>
Condizione di uscita	Il sistema visualizza i campi compilati in modo errato.

**9.3.7 Modifica\_Cliente**

Condizione d'entrata	L'agente deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'agente clicca sulla funzione “gestione Clienti”.</li><li>2. Il sistema risponde visualizzando tutti i clienti registrati.</li><li>3. L’agente sceglie il cliente che vuole modificare e clicca sul pulsante “visualizza”.</li><li>4. Il sistema visualizzerà i dettagli del cliente.</li><li>5. L'agente clicca il pulsante modifica.</li><li>6. Il sistema visualizzerà i campi da modificare (codice fiscale, nome e cognome, data di nascita, email, telefono, username, password).</li><li>7. L’agente dopo aver modificato i campi preme il pulsante “conferma”.</li><li>8. Il sistema visualizza una finestra di avvenuta conferma.</li><li>9. L’agente chiude la finestra.</li></ol>
Condizione d'uscita	Il sistema rimanda alla homepage.

**9.3.8 Elimina\_Cliente**

Condizione d'entrata	L'agente deve essersi loggato nel sistema.
Flusso d'eventi	<ol style="list-style-type: none"><li>1. L'agente clicca sulla funzione “gestione Clienti”.</li><li>2. Il sistema risponde visualizzando tutti i clienti registrati.</li><li>3. L’agente sceglie il cliente che vuole eliminare e clicca sul pulsante “elimina”.</li><li>4. Il sistema visualizza una finestra che conferma l’eliminazione.</li><li>5. L’agente chiude la finestra.</li></ol>
Condizione d'uscita	Il sistema rimanda alla homepage



## 9.4 Gestione Trattativa

### 9.4.1 Assegnazione

Condizione di entrata	L'amministratore deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. Il sistema notifica all'amministratore la creazione di una trattativa da parte di un utente</li><li>2. L'amministratore visualizza le richieste in sospeso premendo sul pulsante "Visualizza" o cliccando sulla sezione "Gestione Trattative"</li><li>3. Dopo aver cliccato su "Dettagli" per verificare la correttezza della trattativa, l'Amministratore clicca sul pulsante "Assegna" e gli viene mostrata la lista degli Agenti disponibili</li><li>4. L'Amministratore seleziona l'Agente a cui vuole assegnare la trattativa premendo il corrispondente tasto "Assegna"</li><li>5. Compare un pop up che conferma che l'assegnazione è avvenuta con successo</li></ol>
Condizione di uscita	L'amministratore clicca sul tasto "OK" del pop up e gli viene mostrata di nuovo la sua home page

### 9.4.2 Elimina

Condizione di entrata	L'agente deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. L'Agente dalla home page clicca sulla sezione "Gestione Trattative"</li><li>2. L'Agente decide la trattativa da eliminare premendo il corrispondente tasto "Elimina"</li><li>3. Compare un pop up che conferma l'eliminazione con successo della trattativa</li></ol>
Condizione di uscita	L'Agente preme il tasto "Ok" del pop up e gli viene mostrata di nuovo la sua home page.

### 9.4.3 Riassegna

Condizione di entrata	L'Amministratore deve essersi loggato nel sistema.
Flusso	<ol style="list-style-type: none"><li>1. L'Amministratore dalla home page clicca sulla sezione "Gestione Trattative"</li><li>2. L'amministratore visualizza le trattative e premendo sul pulsante "Riassegna" gli viene mostrata la lista degli Agenti disponibili</li><li>3. L'Amministratore seleziona il nuovo Agente a cui vuole assegnare la trattativa premendo il corrispondente tasto "Assegna"</li><li>4. Compare un pop up che conferma che l'assegnazione è avvenuta con successo</li></ol>
Condizione di uscita	L'amministratore clicca sul tasto "OK" del pop up e gli viene mostrata di nuovo la sua home page.

## **10.Pianificazione del Testing**

Il team per il testing deve essere composto da persone che hanno una completa e approfondita conoscenza del sistema e delle tecniche di testing con i documenti associati, quali Test Plan e Test Case Specification. Tali tecniche devono essere applicate nei tempi, nel budget e nei vincoli di qualità stabiliti. Solitamente i componenti del team di testing non sono coinvolti nella fase di sviluppo, in quanto non deve esistere una reticenza nel decomporre il sistema. Il team dedicato al controllo della qualità è responsabile dell'attività di testing e quindi della ricerca di fault. La documentazione dei fault trovati è inviata agli sviluppatori per consentire la correzione del sistema.

Il sistema revisionato è poi testato nuovamente non solo per verificare se gli errori trovati in precedenza sono stati eliminati ma soprattutto per verificare che non ne sono stati introdotti dei nuovi. L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di tale attività o un cattiva interpretazione di essa può portare al completo fallimento del sistema. Data l'importanza del testing ne risulta fondamentale la schedulazione.

### **10.1. Determinazione dei ruoli**

Il team dedicato all'attività di testing di sistema sarà composto da Francesco Napoli e Luca Diodato De Martino. Le attività relative al testing di unità saranno delegate agli stessi sviluppatori delle componenti, in modo tale da alleggerire il lavoro del team di testing, che potrà quindi dedicarsi ad un più attento lavoro di testing funzionale.

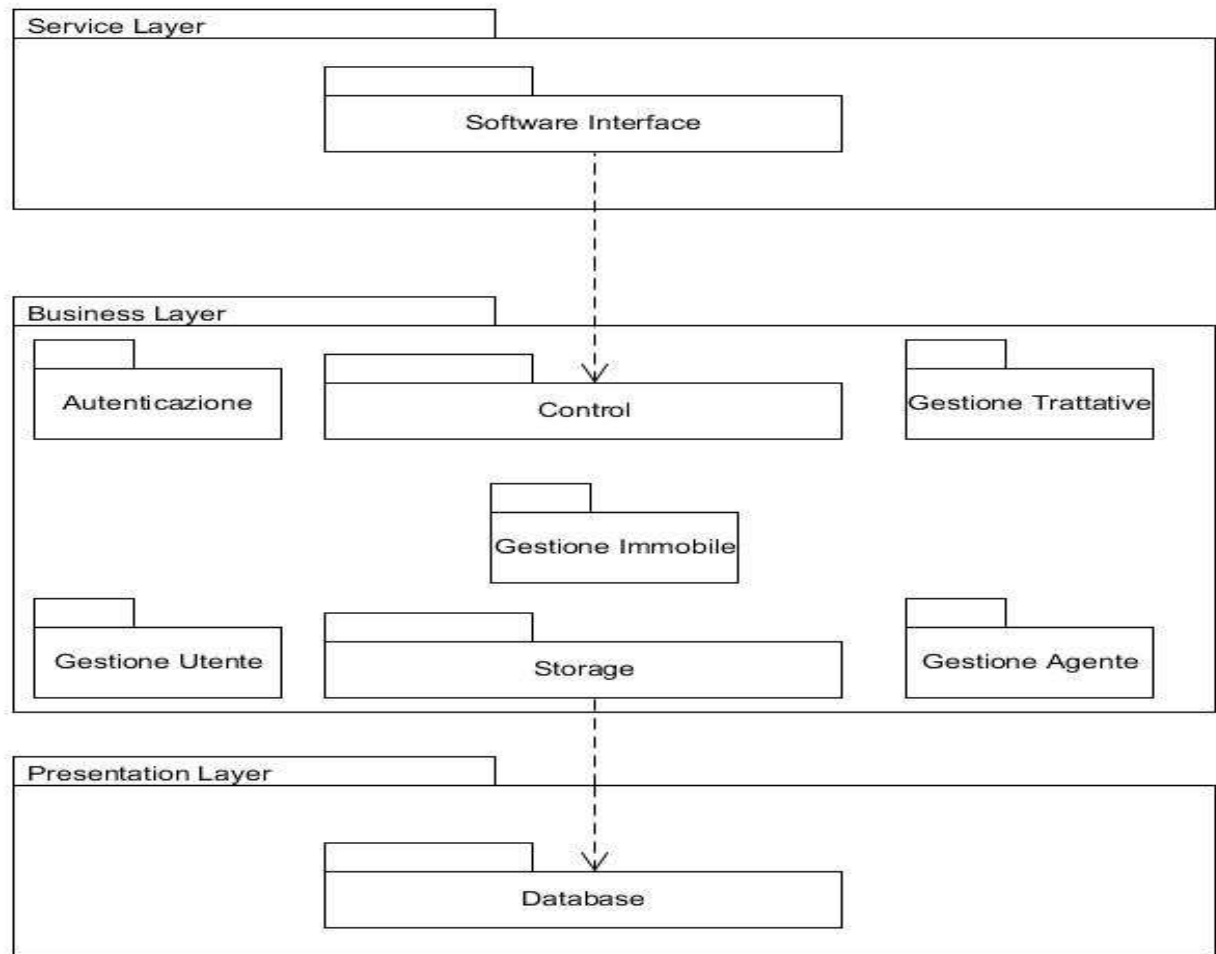
### **10.2. Determinazione dei rischi**

I rischi di un completo fallimento verranno minimizzati effettuando una pianificazione verticale delle attività di testing funzionale. Questo permetterà in caso di ritardi, dovuti ad una grande quantità di failure trovati, di rilasciare meno funzionalità del previsto, ma completamente testate. Inoltre tale pianificazione ridurrà notevolmente la produzione di driver e test stub, evitando l'introduzione di nuovi errori, dovuti all'implementazione di tali componenti.

### **10.3. Decomposizione gerarchica del sistema**

La decomposizione in sottosistemi effettuata nell'attività di design è stata mappata sulla base di tre livelli gerarchici:

- Presentation layer (interfaccia) che include la gestione dell'interfaccia utente;
- Business logic che include la gestione degli oggetti entity e control;
- Services che effettua la memorizzazione ed il recupero degli oggetti persistenti.



#### 10.4. Organizzazione delle attività di testing

Le attività di testing verranno organizzate secondo uno schema che effettuerà una divisione funzionale di tipo verticale. In questo modo al termine di ogni attività si avrà una funzionalità completamente testata nei suoi livelli gerarchici. I vantaggi principali sono che in caso di ritardi dovuti al ritrovamento di numerosi failure (fallimenti) il sistema verrà rilasciato con meno componenti, ma interamente testate e funzionanti.

#### 10.5. Schedulazione delle attività di testing

Le attività di Testing verranno espresse mediante l'uso di un PERT chart. Sarà previsto un tempo massimo di due giorni per effettuare tutti i test.

## **11 Test Report**

I test effettuati precedentemente hanno avuto esito positivo in quanto tutti gli output hanno dato il risultato previsto.

### **Organizzazione del Progetto**

La tabella seguente riporta i nomi dei componenti del team e i rispettivi ruoli.

Nome	Ruolo
Luca Diodato De Martino	Analista, Designer
Francesco Napoli	Project Manager, Analista
Giuseppe Di Martino	Programmatore
Giuseppe Giordano	Programmatore, Designer