

# Versione 1.1

## HotelMgM

Daniele Calabrò  
Francesco Orciuoli  
Raffaele Costantino



**[TEST PLANNING]**

# Indice

1.INTRODUZIONE.....	3
2. Documenti correlati.....	5
2.1 Relazione con ODD.....	5
2.2 Relazione con SDD .....	6
2.3 Relazione con RAD.....	6
3.Overview del sistema .....	7
4.Funzionalità testate/Funzionalità non testate.....	10
5.Criteri di Pass/Failed .....	12
6. Approccio.....	13
6.1 Testing di Unità (Unit Testing).....	13
6.2 Testing di Integrazione (Integration Testing) .....	14
6.3 Testing di Sistema (System Testing) .....	14
7. Sospensione e Ricerca .....	15
7.1 Criteri di sospensione .....	15
7.2 Criteri di ripristino.....	15
8. Requisiti Hardware/Software .....	16
Viste con i dati contenuti all'interno delle tabelle utilizzate per la parte di software implementata:	18
ACCOUNT.....	18
PERSONA.....	19
ASSOCIA.....	20
CLIENTE .....	21
PRENOTA.....	22
CAMERA.....	23
POSSIEDE .....	24
CALENDARIO .....	24
HA.....	25
EXTRA.....	26
9. Casi di test (Test Cases).....	27
9.1 Login.....	27
9.2 Aggiungi Utente .....	31
9.3 Aggiungi Camera.....	36
9.4 Ricerca Camera .....	40
9.5 Ricerca Utenti .....	43
9.6 Aggiungi cliente .....	46
9.7 Ricerca Disponibilità .....	51
9.8 Associa Extra.....	55
9.9 Rilascia Camera .....	59
9.10 Prenota Camera .....	61
3.Pianificazione.....	67

# 1.INTRODUZIONE

Il ciclo di vita di un progetto software si chiude, almeno nella prima fase, con la consegna del prodotto applicativo sviluppato e la messa in funzione del sistema presso il cliente. Questa fase che a prima vista sembra non avere attività di sviluppo, se non nelle varie fasi di installazione del sistema, in realtà è il preludio a quella che è la fase più difficile di un rapporto contrattuale cliente-fornitore: la “chiusura” del contratto o la regolazione del compenso stabilito a fronte di una fornitura.

Il testing è una tecnica di “fault detection” il cui scopo è appunto quello di rilevare errori in maniera pianificata all'interno del codice prodotto. L'obiettivo è, quindi, evitare che questi si presentino in fase di utilizzo da parte dell'utente finale.

L'attività di testing risulta essere una componente fondamentale nell'ambito dello sviluppo di un prodotto software efficiente e funzionale, in particolar modo perchè permette di rilevare possibili errori e/o malfunzionamenti all'interno del sistema, che potrebbero verificarsi in seguito alla scrittura del codice sorgente ed alle varie trasformazioni effettuate durante l'attività di mapping.

L'obiettivo fondamentale è quindi quello di individuare in modo pianificato, tramite un lavoro attento ed ordinato, le possibili falle presenti nel software, in modo da evitare che queste possano ripresentarsi una volta che il prodotto verrà utilizzato dall'utente finale. La peculiarità risulta quindi quella di dover garantire la correttezza del codice sorgente implementato, modificandolo ogniqualvolta si riscontri una possibile inesattezza, prestando comunque attenzione a revisionarlo in modo corretto, senza provocare ulteriori errori.

Il presente documento ha lo scopo principale di pianificare e descrivere in modo dettagliato l'attività di testing che il team di sviluppo si prepara ad effettuare ; in particolar modo verranno analizzate le seguenti funzionalità:

- Gestione Camere
- Gestione Account
- Gestione Camere Operatore
- Gestione Clienti
- Gestione Extra

Il testing prende in considerazione la maggior parte delle funzionalità implementate e descritte nell'ODD, tralasciandone alcune che non comportano la gestione di casi di errore, in seguito specificheremo il motivo per cui abbiamo deciso di tralasciare il testing di queste ultime funzionalità.

## 2. Documenti correlati

Il Test Planning ha una relazione importante con gli altri documenti che corredano l'intera documentazione del nostro sistema software, soprattutto perché l'attività di testing viene effettuata solo dopo che il software è stato completamente implementato e solo dopo aver completato la revisione di tutte le altre attività svolte.

In questa fase, quindi, risultano estremamente importanti i documenti precedentemente sviluppati poiché ognuno di essi contiene informazioni necessarie fortemente correlate al processo di testing. Vi è quindi una relazione tra il testing ed i documenti:

- ODD (Object Design Document)
- SDD (System Design Document)
- RAD (Requirements Analysis Document)

### ***2.1 Relazione con ODD***

Il testing fa riferimento ai pacchetti e alle classi definite nel documento di Object Design. Le classi specificate nel documento di Object Design sono state quelle effettivamente implementate e quindi testate una volta terminata la scrittura del codice sorgente. Durante la fase di testing delle classi specificate nell'ODD si è prestato attenzione al comportamento atteso delle azioni specificate e si è verificato che non ci fossero risultati e comportamenti contrastanti a ciò che effettivamente ci si aspettava. Gli errori che sono venuti fuori sono stati opportunamente corretti.

## ***2.2 Relazione con SDD***

La pianificazione dei test delle componenti rispecchia la decomposizione in sottosistemi e in layer (strati di un sottosistema) specificata nel documento di System Design.

## ***2.3 Relazione con RAD***

La relazione tra il testing e questo documento riguarda soprattutto i requisiti funzionali e non funzionali del sistema che sono stati elencati nel RAD. Inoltre un riferimento importante per il testing sono stati gli USE CASE contenuti nel RAD poiché essi esplicitano in maniera dettagliata le azioni e i risultati attesi dal sistema.

# 3. Overview del sistema

L'architettura del sistema HotelMgM è basata su una architettura di tipo **Client/Server**. La scelta di questo tipo di architettura è indispensabile per l'uso della rete da parte del sistema. Il sistema infatti è progettato per utilizzare sia la rete Internet che la rete interna locale.

L'intera architettura Client/Server del sistema software HotelMgM è di tipo “**Three-Tier**”, prevediamo quindi una scomposizione del sistema in sottosistemi. Ogni sottosistema è suddiviso in tre strati.

I sottosistemi del sistema HotelMgM sono sviluppati su tre livelli logico-funzionali:

- **PresentationLayer**: si occupa di presentare i risultati e raccogliere input degli utenti, questo include tutta la gestione delle interfacce utente.
- **ApplicationLayer**: si occupa di fornire al software tutte le specifiche funzionalità per cui esso è stato progettato. Include la gestione degli oggetti e delle classi con relativi metodi e controlli.
- **StorageLayer**: si occupa della gestione del database del sistema e quindi include la memorizzazione e il recupero di oggetti persistenti.

In definitiva il nucleo del nostro sistema è il DataBase il quale avrà al suo interno i dati riguardanti gli account, le camere, i clienti, le persone associate, gli extra, le prenotazioni e il calendario di occupazione. Il DataBase presenterà all'utente che sta utilizzando il software alcune viste tramite il PresentationLayer, lasciando all'ApplicationLayer il compito di gestire gli input inseriti dagli utenti e di effettuare operazioni sugli oggetti del sistema.

Come anticipato nel documento di System Design (SDD), l'architettura del nostro sistema è suddivisa in 4 sottosistemi, ogni sottosistema si occupa di svolgere determinate attività:

**1)SottosistemaAmministratore:**

1. Gestione Camere
2. Gestione Account
3. Funzioni Operatore 1
4. Funzioni Operatore 2

**2)SottosistemaOperatore1:**

1. Gestione Camere Operatore
2. Gestione Clienti

**3)SottosistemaOperatore2:**

1. Gestione Extra

**4)SottosistemaUtenteFinale:**

1. Gestione Web

Come precedentemente indicato nel documento di Object Design (ODD), non tutte le funzionalità sono state implementate. Per correttezza riportiamo l'elenco delle funzionalità implementate.

**1. SottosistemaAmministratore:**

1.1.PacchettoGUI

1.2.PacchettoGestioneAccount

- Aggiungere un operatore
- Ricerca operatore
- Visualizzare Elenco Operatori



### 1.3. Pacchetto Gestione Camere Amministratore:

- Aggiungere una camera
- Ricerca camera
- Visualizzare Elenco Camera

## 2. Sottosistema Operatore1:

### 2.1. Pacchetto GUI

### 2.2. Pacchetto Gestione Camere Operatore:

- Ricerca disponibilità camera
- Prenotare una camera
- Occupa una camera
- Rilasciare una camera
- Visualizzare Elenco Camere

### 2.3. Pacchetto Gestione Clienti:

- Inserire dati cliente
- Ricerca cliente
- Visualizzare Elenco clienti
- Aggiungere persone associate
- Visualizzare persone associate al cliente

## 3. Sottosistema Operatore2:

### 3.1. Pacchetto GUI

### 3.2. Pacchetto Gestione Extra:

- Aggiornare menù bar
- Visualizzare menù bar
- Associare extra al cliente

Le funzionalità relative al sottosistema Sottosistema Utente Finale non sono state implementate mentre gran parte delle funzionalità dei restanti sottosistemi sono state implementate.

# 4. Funzionalità testate / Funzionalità non testate

Tra tutte le funzionalità implementate il team ha deciso di **testare** le seguenti funzionalità:

## TESTATE

### 1) SOTTOSISTEMA AMMINISTRATORE:

- Aggiungi Camera
- Ricerca Camera
- Aggiungi Utente
- Ricerca Utente

### 2) SOTTOSISTEMA OPERATORE LIVELLO 1:

- Aggiungi Cliente
- Ricerca Disponibilità Camera
- Prenota Camera
- Rilascia Camera

### 3)SOTTOSISTEMA OPERATORE LIVELLO 2:

- Associa Extra

Invece le funzionalità non testate sono:

## **NON TESTATE**

### 1)SOTTOSISTEMA AMMINISTRATORE:

- Elenco Camere
- Elenco Utenti

### 2)SOTTOSISTEMA OPERATORE LIVELLO 1:

- Elenco Clienti
- Elenco Persone Associate
- Elenco Camere
- Ricerca Cliente
- Aggiungi Persone Associate
- Occupa Camera

### 3)SOTTOSISTEMA OPERATORE LIVELLO 2:

- Elenco Extra
- Aggiungi Extra

La scelta di non testare le funzioni relative agli “elenchi” è stata fatta poiché queste ultime non presentano casi di errori particolari in quanto non richiedono input da parte dell'utente, al massimo presentano errore solo in caso di connessione fallita con il DataBase o in caso di elenco vuoto, e quindi basta cliccare il relativo pulsante per visualizzare l'elenco. Altre funzioni non sono state testate poiché sono molto simili ad altre già testate precedentemente e presentano le stesse condizioni di input/output.

# 5. Criteri di Pass/Failed

La fase di testing ruota intorno agli input che vengono forniti e gli output che vengono restituiti, che permettono di verificare se una determinata componente supera o meno il test effettuato.

I dati di input del test sono stati suddivisi in classi di equivalenza, ed un input appartenente ad una data classe supera il test se l'output ottenuto corrisponde ai risultati attesi.

Noi sviluppatori siamo a conoscenza del comportamento che dovrebbe avere una determinata funzionalità, quindi abbiamo verificato che ogni componente sotto testing risponda o meno ai requisiti richiesti.

Ogni funzionalità che è stata sottoposta al processo è stata attentamente analizzata e, forniti gli input, sono state specificate tutte le possibili condizioni che potrebbero portare ad un errore nel sistema; l'intero team, alla presenza di un fallimento nella procedura di testing, ha provveduto a prendere atto della situazione e a modificare la parte del codice che provoca il problema.

# 6. Approccio

L'attività di testing è stata pianificata seguendo un approccio ben preciso, scorrendo attraverso una serie di fasi si verificano le specifiche funzionalità del sistema.

In una prima fase si esegue il testing delle singole componenti che formano un sottosistema. In un secondo momento viene eseguito il testing d'integrazione che viene effettuato su gruppi di sottosistemi con l'obiettivo di testare le interfacce tra i sottosistemi.

Nella terza ed ultima fase viene effettuato il testing di sistema ovvero sarà testato l'intero sistema con l'obiettivo di determinare se il sistema assemblato rispecchia i requisiti.

Quindi il processo di testing può essere articolato e classificato in tre fasi principali.

## 6.1 Testing di Unità (Unit Testing)

Durante questa fase l'attenzione viene focalizzata sulle singole componenti, in particolare vengono ricercate condizioni di fallimento isolando le componenti tramite utilizzo di Test Stub (implementazione completa o parziale di una componente dalla quale dipende un'altra componente sotto testing) e Test Driver (implementazione completa o parziale di una componente che fa uso di un'altra componente sotto testing).

Tali test vengono applicati dopo aver isolato ogni singola funzionalità da testare in modo da verificare una completa validità delle componenti e tener traccia dei possibili errori che vengono riscontrati.

Come strategia utilizzata per l'Unit Testing è stata usata la tecnica BlackBox , che si focalizza sul comportamento Input/Output ignorando la struttura interna della componente.

Al fine di minimizzare il numero dei test cases gli input sono stati partizionati in classi di equivalenza. Se si verifica un errore con dei risultati inattesi si interviene in maniera tempestiva sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

## ***6.2 Testing di Integrazione (Integration Testing)***

Durante questa fase si procede ad inglobare ed integrare tutte le componenti che precedentemente erano state isolate e testate singolarmente, in modo da verificare il corretto funzionamento del sistema anche dal punto di vista globale e non solo dal punto di vista delle singole componenti.

La tecnica di approccio a questo tipo di operazione prevede l'uso combinato di strategie Bottom-Up e Top-Down, in particolare una strategia Sandwich Testing. L'uso di questa tecnica è fatto fino a testare tutte le funzionalità implementate.

Questo approccio mira principalmente a ridurre l'accoppiamento tra sottosistemi differenti ed a focalizzare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

## ***6.3 Testing di Sistema (System Testing)***

Lo scopo principale di questa fase di testing è quello di dimostrare che il sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non funzionali descritti nel documento RAD.

Tramite un approccio BalckBox sarà verificata la qualità e l'affidabilità dell'intero prodotto software, in modo che possa essere utilizzato dall'utente finale senza margini di errori.

# 7. Sospensione e Ricerca

La fase di testing occupa una parte importante nell'intera attività inerente il progetto, essa è anche molto delicata e può causare imprevisti e slittamenti dei tempi a causa di errori e malfunzionamenti del sistema.

Per questo motivo il nostro team ha deciso di varare delle condizioni di eventuale sospensione e ricerca del testing.

## ***7.1 Criteri di sospensione***

Il testing sarà sospeso quando saranno state testate tutte le classi scelte per ogni possibile input e il risultato sarà quello atteso.

Nel momento in cui il testing rivelerà un errore si dovrà passare alla fase di correzione che dovrà essere seguita da una ripetizione dell'intero processo di testing per rilevare la presenza di eventuali errori introdotti dalle correzioni stesse.

## ***7.2 Criteri di ripristino***

La ripresa del test avviene soltanto quando tutti i problemi relativi alla sospensione dello stesso sono stati risolti. L'attività di testing riprenderà a partire dal test case che ha causato la sospensione.

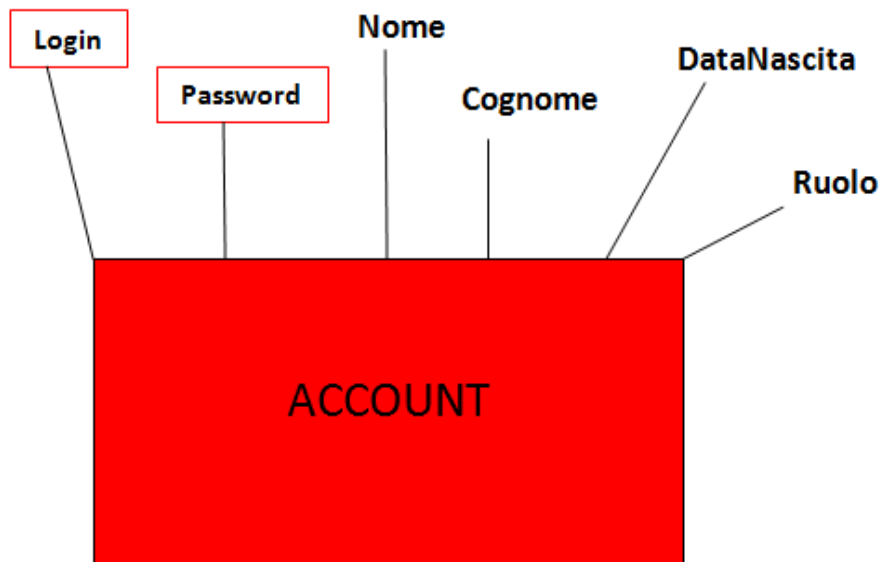
# 8. Requisiti Hardware/Software

L'hardware necessario per l'attività di test è un PC di medio/bassa potenza ed eventualmente un altro PC che costituisca il server per poter testare il funzionamento della tecnologia RMI.

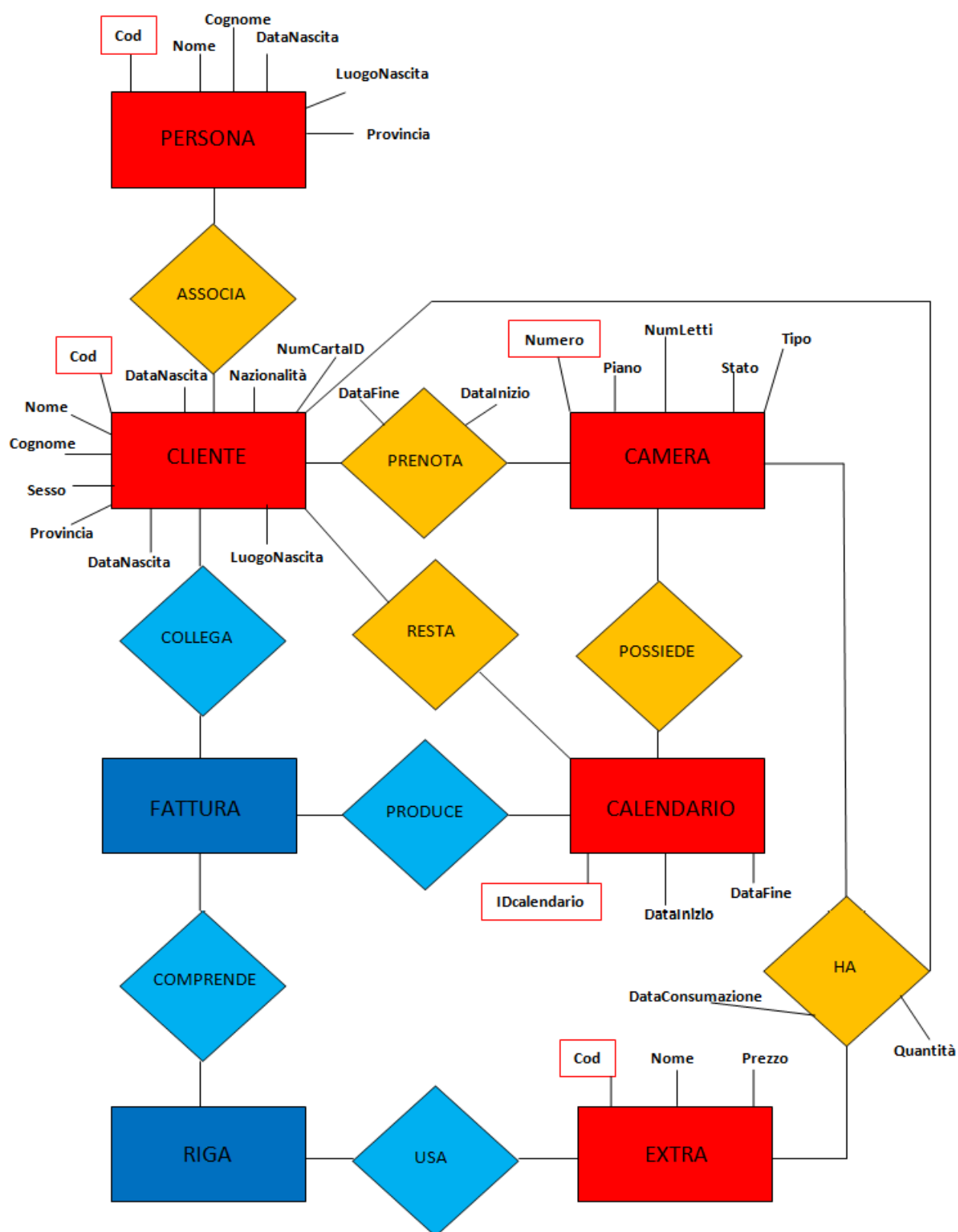
Sui PC devono essere installati un DBMS quale MySQL, e una JVM 1.5 o superiore per il supporto del linguaggio Java.

Inoltre come materiale utilizzato per il testing consideriamo di grande importanza la struttura del nostro database contenente già molti dati di prova caricati.

A tal proposito mostriamo la struttura del nostro database e le viste di tabelle che ci sono servite per effettuare il testing sulla parte implementata.







In blu/celeste tabelle e associazioni che non rientrano nella parte implementata  
In rosso/arancio tabelle e associazioni che contengono dati utili per la parte implementata

Viste con i dati contenuti all'interno delle tabelle utilizzate per la parte di software implementata:

### ACCOUNT

Login	Password	Nome	Cognome	DataNascita	Ruolo
admin89	salerno89	Daniele	Calabrò	05/05/1989	Amministratore
casa89	abitazione89	Raffaele	Costantino	11/05/1989	Operatore livello2
dottore85	unisa	Gabriele	Bavota	19/11/1985	Amministratore
fraorc	studente89	Francesco	Orciuoli	09/08/1989	Operatore livello1
manager	ingegneria	Andrea	De Lucia	30/06/1968	Amministratore
simy	bella	Simona	Marino	19/11/1993	Operatore livello2

## PERSONA

Cod	Nome	Cognome	LuogoNascita	Provincia	DataNascita
1	Rosa	Buono	Pontecagnano	Salerno	23/06/1989
2	Marco	Sessa	Scafati	Salerno	10/11/1993
3	Ugo	Sessa	Pompei	Napoli	31/07/1987
4	Carmela	Sessa	Scafati	Salerno	28/02/1990
5	Giuseppina	Russo	Nocera Inferiore	Salerno	09/09/1960
6	Marta	Pierri	Bologna	Bologna	07/08/1996
7	Francesco	Pierri	Rimini	Rimini	27/12/1993
8	Anna	Di Lorenzo	Bologna	Bologna	22/11/1962
9	Katrin	Muller	Berlino	Berlino	06/05/1978
10	Benedetta	Cantarella	Salerno	Salerno	03/09/1994
11	Simone	Cantarella	Baronissi	Salerno	15/12/1992
12	Martina	Cantarella	San Severino	Salerno	14/01/1986
13	Francesca	Cantarella	Battipaglia	Salerno	19/07/1989
14	Emiliana	Marino	Salerno	Salerno	24/08/1961
15	Anna	Persano	Eboli	Salerno	25/12/1933
16	Domenico	Marino	Agropoli	Salerno	21/04/1929
17	Sara	Costantino	San Severino	Salerno	18/03/1989
18	Nicola	Rossi	Roma	Roma	17/07/1987
19	Elena	Rossi	Roma	Roma	13/01/1990
20	Giovanna	Cerrato	Gaeta	Latina	01/05/1960
21	Gilda	Trezza	Salerno	Salerno	21/11/1990
22	Roberta	Capri	Sala Consilina	Salerno	07/07/1973
23	Nicola	Neri	Avellino	Avellino	03/04/2003

## ASSOCIA

CodCliente	CodPersona
1	1
2	2
2	3
2	4
2	5
4	6
4	7
4	8
5	9
6	10
6	11
6	12
6	13
6	14
6	15
6	16
7	17
8	18
8	19
8	20
9	21
10	22
10	23

## CLIENTE

Cod	Nome	Cognome	LuogoNascita	Provincia	DataNascita	NumCartaID	CodFiscale	Nazionalita	Sesso
1	Rizziero	Gaeta	Salerno	Salerno	19/05/1979	AG0897654345	GTARZR89E19H703 A	Italiana	M
2	Roberto	Sessa	Nocera Inferiore	Salerno	23/10/1958	AC5749465909	SSSRRT86R23F912E	Italiana	M
3	Marta	Citro	Napoli	Napoli	05/06/1989	NA6485638294	MRTCTR89H45F839K	Italiana	F
4	Daniele	Pierri	Bologna	Bologna	05/07/1960	BO34657563849	DNLPRR86E05A944P	Italiana	M
5	Elena	Landi	Monaco	Monaco	17/09/1978	MO57376357394	LNELND88P57Z123Q	Tedesca	F
6	Andrea	Cantarella	Avellino	Avellino	29/08/1959	AV46484657548	NDRCTR89M29A509L	Italiana	M
7	Andrea	Merola	Curteri	Salerno	04/04/1984	SA57474384735483	MRLNDR99D04F138T	Italiana	M
8	Mario	Rossi	Roma	Roma	26/03/1956	AL956768	MRARSSE58TYW471G	Italiana	M
9	Francesco	Verdi	Montoro Inferiore	Avellino	02/11/1977	AY575768	FRNVRD5E58TY471P	Italiana	M
10	Carlo	Neri	Venosa	Potenza	08/12/1968	ER575768	CRLNRE5E88TS471Z	Italiana	M
11	Michele	Bianchi	Mestre	Venezia	17/01/1981	OT575734	MCHBNC7E3PCR471F	Italiana	M

## **PRENOTA**

Cod	Numero	DataInizio	DataFine
3	101	11/04/2011	28/04/2011
5	105	11/04/2011	27/04/2011
7	110	11/04/2011	24/04/2011
10	204	11/04/2011	30/04/2011
2	301	25/07/2011	10/08/2011
4	208	18/07/2011	30/07/2011
5	106	25/07/2011	07/08/2011
5	106	01/10/2011	10/10/2011
6	208	01/08/2011	17/08/2011
6	201	01/08/2011	17/08/2011
6	202	01/08/2011	17/08/2011
7	200	06/06/2011	20/06/2011
7	200	01/09/2011	12/09/2011
8	208	26/06/2011	15/07/2011
9	200	11/07/2011	28/07/2011
10	205	06/08/2011	16/08/2011
11	102	02/06/2011	07/06/2011
1	200	01/08/2011	19/08/2011

## CAMERA

Numero	Piano	NumLetti	Tipo	Stato
100	1	1	Singola	1
101	1	1	Singola	0
102	1	1	Singola	1
103	1	1	Singola	1
104	1	1	Singola	1
105	1	2	Doppia	0
106	1	2	Doppia	1
107	1	2	Doppia	1
108	1	2	Doppia	1
109	1	2	Doppia	1
110	1	2	Matrimoniale	0
200	2	2	Matrimoniale	1
201	2	2	Matrimoniale	1
202	2	2	Matrimoniale	1
203	2	2	Matrimoniale	1
204	2	3	Tripla	0
205	2	3	Tripla	1
206	2	3	Tripla	1
207	2	3	Tripla	1
208	2	4	Quadrupla	1
209	2	4	Quadrupla	1
210	2	4	Quadrupla	1
300	3	4	Quadrupla	1
301	3	5	Multipia	1
302	3	5	Multipia	1
303	3	5	Multipia	1
304	3	5	Multipia	1
305	3	5	Multipia	1
306	3	2	Suite	1
307	3	2	Suite	1

## **POSSIEDE**

IDcalendario	Numero
20	101
21	105
22	110
23	204

## **CALENDARIO**

IDcalendario	DataInizio	DataFine	CodCliente
20	11/04/2011	28/04/2011	3
21	11/04/2011	27/04/2011	5
22	11/04/2011	24/04/2011	7
23	11/04/2011	30/04/2011	10



## HA

Numero	CodCliente	CodExtra	DataConsumazione	Quantita
101	3	21	11/04/2011	1
101	3	3	11/04/2011	1
105	5	2	11/04/2011	2
105	5	11	11/04/2011	2
105	5	3	11/04/2011	2
110	7	1	11/04/2011	2
110	7	6	11/04/2011	2
110	7	3	11/04/2011	2
204	10	2	11/04/2011	3
204	10	19	11/04/2011	1
204	10	22	11/04/2011	3

## **EXTRA**

Cod	Nome	Prezzo
1	Caffè	0.7
2	Cappuccino	1
3	Pizza Margherita	3
4	Croccantelle	0.5
6	Cornetto	0.8
7	Patatine	1
8	Coca Cola	1
9	Cioccolata	1.2
10	Spremuta	2
11	Hot Dog	3
12	Primo Piatto	3
13	Secondo Piatto	3
14	Contorno	1
15	Vino	2
16	Frutta	0.5
17	Caffè corretto	1.5
18	Acqua	0.4
19	Gelato	1
20	Torte	10
21	Dolce	2
22	Succo di Frutta	1
23	Yogurt	1
25	Sigari	4
26	Sigarette HotelMgM	4
27	Ricarica Vodafone	10
28	Ricarica Wind	10
29	Ricarica Tim	10
30	Ricarica 3	10
31	Birra	2.5
32	Spumante	50

# 9. Casi di test (Test Cases)

## 9.1 Login

Il test riguarda in particolare le funzionalità di login di accesso al sistema

### Classi di equivalenza

Gli input del test sono il Nome Utente e la Password

Input	Nome Utente,Password	
Classi Valide	CL_001	Stringa alfanumerica compresa tra 4 e 20 caratteri
Classi non valide	CL_002	Stringa di lunghezza maggiore di 20 caratteri
	CL_003	Stringa di lunghezza minore di 4 caratteri
	CL_004	Nome Utente o Password non presente nel DataBase
	CL_005	Stringa vuota

### Identificazione test case

TestCase: TestCaseLogin 01		
Input	Classe	Membro
Nome Utente	CL_003	fra
Password	CL_003	orc

TestCase: TestCaseLogin 02		
Input	Classe	Membro
Nome Utente	CL_003	fra
Password	CL_005	

TestCase: TestCaseLogin 03		
Input	Classe	Membro
Nome Utente	CL_005	
Password	CL_005	

TestCase: TestCaseLogin 04		
Input	Classe	Membro
Nome Utente	CL_002	daniele_calabro05-05-1989

Password	CL_002	studente_universitario1989
----------	--------	----------------------------

TestCase: TestCaseLogin 05		
Input	Classe	Membro
Nome Utente	CL_002	daniele_calabro05-05-1989
Password	CL_003	orc

TestCase: TestCaseLogin 06		
Input	Classe	Membro
Nome Utente	CL_005	
Password	CL_002	studente_universitario1989

TestCase: TestCaseLogin 07		
Input	Classe	Membro
Nome Utente	CL_001	admin89
Password	CL_003	sal

TestCase: TestCaseLogin 08		
Input	Classe	Membro
Nome Utente	CL_005	

Password	CL_001	salerno89
----------	--------	-----------

TestCase: TestCaseLogin 09		
Input	Classe	Membro
Nome Utente	CL_001	admin89
Password	CL_001	salerno89

## 9.2 Aggiungi Utente

Il test riguarda le funzionalità di aggiunta di un utente del sistema

### Classi di equivalenza

Gli input del test sono: Nome Utente, Password, Nome, Cognome, Data di Nascita, Ruolo

Input	Nome Utente, Password, Nome, Cognome, Data di Nascita, Ruolo	
Classi Valide	CL_001	Stringa alfanumerica compresa tra 4 e 20 caratteri
	CL_006	Data corretta
	CL_011	Stringa alfanumerica minore uguale di 20 caratteri
Classi non valide	CL_002	Stringa di lunghezza maggiore di 20 caratteri
	CL_003	Stringa di lunghezza minore di 4 caratteri
	CL_007	Data non corretta
	CL_005	Stringa vuota

### Identificazione test case

TestCase: TestCaseAggiungiUtente 01		
Input	Classe	Membro
Nome Utente	CL_003	Fra
Password	CL_003	Orc

Nome	CL_011	Francesco
Cognome	CL_011	Orciuoli
Data di Nascita	CL_006	09/08/1989
Ruolo		Amministratore

TestCase: TestCaseAggiungiUtente 02		
Input	Classe	Membro
Nome Utente	CL_003	Fra
Password	CL_001	Orciuoli
Nome	CL_011	Francesco
Cognome	CL_011	Orciuoli
Data di Nascita	CL_006	09/08/1989
Ruolo		Amministratore

TestCase: TestCaseAggiungiUtente 03		
Input	Classe	Membro
Nome Utente	CL_002	daniele_calabro05-05-1989
Password	CL_002	studente_universitario1989
Nome	CL_011	Daniele
Cognome	CL_011	Calabrò
Data di Nascita	CL_006	09/08/1989



Ruolo		Amministratore
-------	--	----------------

TestCase: TestCaseAggiungiUtente 04		
Input	Classe	Membro
Nome Utente	CL_002	daniele_calabro05-05-1989
Password	CL_001	salerno89
Nome	CL_011	Daniele
Cognome	CL_011	Calabrò
Data di Nascita	CL_006	09/08/1989
Ruolo		Operatore livello 2

TestCase: TestCaseAggiungiUtente 05		
Input	Classe	Membro
Nome Utente	CL_001	Francesco
Password	CL_001	Orciuoli
Nome	CL_011	Francesco
Cognome	CL_011	Orciuoli
Data di Nascita	CL_007	09/18/89
Ruolo		Amministratore

TestCase: TestCaseAggiungiUtente 06		
Input	Classe	Membro
Nome Utente	CL_005	
Password	CL_005	
Nome	CL_011	Raffaele
Cognome	CL_011	Costantino
Data di Nascita	CL_005	
Ruolo		Operatore livello 1

TestCase: TestCaseAggiungiUtente 07		
Input	Classe	Membro
Nome Utente	CL_001	casa89
Password	CL_001	abitazione89
Nome	CL_005	
Cognome	CL_005	
Data di Nascita	CL_006	09/08/1989
Ruolo		Amministratore

TestCase: TestCaseAggiungiUtente 08		
Input	Classe	Membro
Nome Utente	CL_001	casa89

Password	CL_001	abitazione89
Nome	CL_011	ciao
Cognome	CL_002	daniele_Gilda_francesco
Data di Nascita	CL_006	09/08/1989
Ruolo		Amministratore

TestCase: TestCaseAggiungiUtente 09		
Input	Classe	Membro
Nome Utente	CL_001	admin89
Password	CL_001	salerno89
Nome	CL_011	Daniele
Cognome	CL_011	Calabrò
Data di Nascita	CL_006	05/05/1989
Ruolo		Amministratore

## 9.3 Aggiungi Camera

Il test riguarda le funzionalità di aggiunta di una camera del sistema

### Classi di equivalenza

Gli input del test sono: Numero Camera, Piano, Numero Letti, Tipo

Input	Numero Camera, Piano, Numero Letti, Tipo	
Classi Valide	CL_008	Numero positivo maggiore di 0
	CL_026	Camera non esistente
Classi non valide	CL_009	Numero negativo
	CL_027	Camera già esistente
	CL_005	Campo vuoto

### Identificazione test case

TestCase: TestCaseAggiungiCamera 01		
Input	Classe	Membro
Numero Camera	CL_009	-103
Piano	CL_008	3
Numero Letti	CL_008	1
Tipo		Singola

TestCase: TestCaseAggiungiCamera 02		
Input	Classe	Membro
Numero Camera	CL_026	1018
Piano	CL_009	-1
Numero Letti	CL_008	2
Tipo		Matrimoniale

TestCase: TestCaseAggiungiCamera 03		
Input	Classe	Membro
Numero Camera	CL_027	103
Piano	CL_008	3
Numero Letti	CL_009	-1
Tipo		Singola

TestCase: TestCaseAggiungiCamera 04		
Input	Classe	Membro
Numero Camera	CL_009	-103
Piano	CL_009	-3
Numero Letti	CL_009	-1

Tipo		Singola
------	--	---------

TestCase: TestCaseAggiungiCamera 05		
Input	Classe	Membro
Numero Camera	CL_005	
Piano	CL_005	
Numero Letti	CL_005	
Tipo		Singola

TestCase: TestCaseAggiungiCamera 06		
Input	Classe	Membro
Numero Camera	CL_005	
Piano	CL_008	3
Numero Letti	CL_008	1
Tipo		Singola

TestCase: TestCaseAggiungiCamera 07		
Input	Classe	Membro
Numero Camera	CL_008	105
Piano	CL_008	1

Numero Letti	CL_008	4
Tipo		Quadrupla

## 9.4 Ricerca Camera

Il test riguarda le funzionalità di ricerca di una camera del sistema

### Classi di equivalenza

Gli input del test sono: Numero Camera, Piano, Numero Letti, Tipo

Input	Numero Camera, Piano, Numero Letti, Tipo	
Classi Valide	CL_008	Numero positivo maggiore di 0
Classi non valide	CL_009	Numero negativo
	CL_005	Campo vuoto

### Identificazione dei test case

TestCase: TestCaseRicercaCamera 01		
Input	Classe	Membro
Numero Camera	CL_008	101
Piano	CL_005	
Numero Letti	CL_005	
Tipo	CL_005	--



TestCase: TestCaseRicercaCamera 02		
Input	Classe	Membro
Numero Camera	CL_009	-101
Piano	CL_005	
Numero Letti	CL_009	-4
Tipo	CL_005	--

TestCase: TestCaseRicercaCamera 03		
Input	Classe	Membro
Numero Camera	CL_009	-101
Piano	CL_009	-3
Numero Letti	CL_009	-2
Tipo	CL_005	--

TestCase: TestCaseRicercaCamera 04		
Input	Classe	Membro
Numero Camera	CL_008	201
Piano	CL_005	
Numero Letti	CL_008	4
Tipo	CL_005	--

TestCase: TestCaseRicercaCamera 05		
Input	Classe	Membro
Numero Camera	CL_008	101
Piano	CL_008	1
Numero Letti	CL_008	2
Tipo		Matrimoniale

## 9.5 Ricerca Utenti

Il test riguarda le funzionalità di ricerca di un utente del sistema

### Classi di equivalenza

Gli input del test sono: Nome Utente, Nome, Cognome

Input	Nome Utente, Nome, Cognome	
Classi Valide	CL_001	Stringa alfanumerica compresa tra 4 e 20 carattere
	CL_010	Coppia Nome e Cognome non vuota
	CL_011	Stringa alfanumerica minore uguale di 20 caratteri
Classi non valide	CL_002	Stringa di lunghezza maggiore di 20 caratteri
	CL_003	Stringa di lunghezza minore di 4 caratteri
	CL_005	Stringa vuota

### Identificazione test case

TestCase: TestCaseRicercaUtente 01		
Input	Classe	Membro
Nome Utente	CL_005	
Nome	CL_005	
Cognome	CL_005	

TestCase: TestCaseRicercaUtente 02		
Input	Classe	Membro
Nome Utente	CL_003	Dan
Nome	CL_005	
Cognome	CL_005	

TestCase: TestCaseRicercaUtente 03		
Input	Classe	Membro
Nome Utente	CL_002	daniele_calabro05-051-989
Nome	CL_005	
Cognome	CL_005	

TestCase: TestCaseRicercaUtente 04		
Input	Classe	Membro
Nome Utente	CL_005	
Nome	CL_011	Francesco
Cognome	CL_005	

TestCase: TestCaseRicercaUtente 05		
Input	Classe	Membro
Nome Utente	CL_005	
Nome	CL_005	
Cognome	CL_011	Bavota

TestCase: TestCaseRicercaUtente 06		
Input	Classe	Membro
Nome Utente	CL_001	admin89
Nome	CL_005	
Cognome	CL_011	Calabrò

TestCase: TestCaseRicercaUtente 07		
Input	Classe	Membro
Nome Utente	CL_005	
Nome	CL_010	Daniele
Cognome	CL_010	Calabrò

## 9.6 Aggiungi cliente

Il test riguarda le funzionalità di aggiunta di un cliente

### Classi di equivalenza

Gli input del test sono: Nome, Cognome, Data di nascita, Luogo di nascita, Provincia, Nazionalità, Sesso, Numero Carta ID, Codice fiscale

Input	Nome, Cognome, Data di nascita, Luogo di nascita, Provincia, Nazionalità, Sesso, Numero Carta ID, Codice fiscale	
Classi Valide	CL_011	Stringa alfanumerica minore uguale di 20 caratteri
	CL_012	Stringa alfanumerica uguale 16
	CL_006	Data corretta
Classi non valide	CL_002	Stringa di lunghezza maggiore di 20 caratteri
	CL_013	Stringa di lunghezza diversa da 16
	CL_007	Data non corretta
	CL_005	Stringa vuota

### Identificazione case test

TestCase: TestCaseAggiungiCliente 01		
Input	Classe	Membro

Nome	CL_002	Antonio_Francesco_Nicola
Cognome	CL_011	Costantino
Data di nascita	CL_006	09/08/1989
Luogo di nascita	CL_011	Salerno
Provincia	CL_011	Salerno
Nazionalità	CL_011	Italiana
Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_012	AHIKD30I8490DADS

TestCase: TestCaseAggiungiCliente 02		
Input	Classe	Membro
Nome	CL_011	Raffaele
Cognome	CL_002	Costantino_Orciuoli_Calabrò
Data di nascita	CL_007	09/13/89
Luogo di nascita	CL_011	Battipaglia
Provincia	CL_011	Salerno
Nazionalità	CL_011	Italiana
Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_012	AHIKD30I8490DADS

TestCase: TestCaseAggiungiCliente 03		
Input	Classe	Membro
Nome	CL_011	Antonio
Cognome	CL_011	Costantino
Data di nascita	CL_006	09/08/1989
Luogo di nascita	CL_011	Battipaglia
Provincia	CL_011	Salerno
Nazionalità	CL_002	Italiano_Polacco_Coreano
Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_012	AHIKD30I8490DADS

TestCase: TestCaseAggiungiCliente 04		
Input	Classe	Membro
Nome	CL_011	Antonio
Cognome	CL_002	Costantino_Trezza_Calabrò
Data di nascita	CL_007	09/08/89
Luogo di nascita	CL_011	Battipaglia
Provincia	CL_011	Salerno
Nazionalità	CL_011	Italiana



Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_012	AHIKD30I8490DADS

TestCase: TestCaseAggiungiCliente 05		
Input	Classe	Membro
Nome	CL_005	
Cognome	CL_011	Costantino
Data di nascita	CL_006	09/08/1989
Luogo di nascita	CL_011	Battipaglia
Provincia	CL_005	
Nazionalità	CL_011	Italiana
Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_013	AHIKD30I8490DADS23

TestCase: TestCaseAggiungiCliente 06		
Input	Classe	Membro
Nome	CL_002	Antonio_Francesco_Nicola
Cognome	CL_011	Costantino
Data di nascita	CL_006	09/08/1989

Luogo di nascita	CL_011	Battipaglia
Provincia	CL_011	Salerno
Nazionalità	CL_005	
Sesso		M
Numero Carta ID	CL_011	AK959413
Codice Fiscale	CL_013	AHIKD30I8490DADS32

TestCase: TestCaseAggiungiCliente 07		
Input	Classe	Membro
Nome	CL_002	Antonio_Francesco_Nicola
Cognome	CL_011	Costantino
Data di nascita	CL_006	09/08/1989
Luogo di nascita	CL_011	Battipaglia
Provincia	CL_011	Salerno
Nazionalità	CL_011	Italiano
Sesso		M
Numero Carta ID	CL_002	AK959413ANCNOSOW
Codice Fiscale	CL_013	AHIKD30I8490DADS

## 9.7 Ricerca Disponibilità

Il test riguarda le funzionalità di ricerca disponibilità di una camera in un determinato periodo

### Classi di equivalenza

Gli input del test sono: Numero camera, Piano, Numero Letti, Tipo, Data di inizio, Data di fine

Input	Numero camera, Piano, Numero Letti, Tipo, Data di inizio, Data di fine	
Classi Valide	CL_008	Numero positivo maggiore di 0
	CL_006	Data corretta
	CL_014	Data inizio <= Data fine
Classi non valide	CL_009	Numero negativo
	CL_007	Data formato non corretto
	CL_016	Data inizio > Data fine

### Identificazione test

TestCase: TestCaseRicercaDisponibilitàCamera 01		
Input	Classe	Membro
Numero camera	CL_009	-103
Piano	CL_009	-3
Numero Letti	CL_009	-1

Tipo		Singola
Data di inizio	CL_006	23/06/2011
Data di fine	CL_006	29/06/2011

TestCase: TestCaseRicercaDisponibilitàCamera 02		
Input	Classe	Membro
Numero camera	CL_009	-203
Piano	CL_008	2
Numero Letti	CL_009	-3
Tipo		Tripla
Data di inizio	CL_007	23/19/2011
Data di fine	CL_006	29/06/2011

TestCase: TestCaseRicercaDisponibilitàCamera 03		
Input	Classe	Membro
Numero camera	CL_008	103
Piano	CL_008	3
Numero Letti	CL_009	-2
Tipo		Matrimoniale
Data di inizio	CL_006	23/06/2011
Data di fine	CL_007	32/06/2011

TestCase: TestCaseRicercaDisponibilitàCamera 04		
Input	Classe	Membro
Numero camera	CL_008	103
Piano	CL_008	3
Numero Letti	CL_008	1
Tipo		Singola
Data di inizio	CL_006	23/06/2011
Data di fine	CL_006	29/06/2011

TestCase: TestCaseRicercaDisponibilitàCamera 05		
Input	Classe	Membro
Numero camera	CL_009	-103
Piano	CL_008	3
Numero Letti	CL_009	-1
Tipo		Singola
Data di inizio	CL_006	23/06/2011
Data di fine	CL_006	29/06/2011

TestCase: TestCaseRicercaDisponibilitàCamera 06		
Input	Classe	Membro
Numero camera	CL_009	-103
Piano	CL_008	3
Numero Letti	CL_009	-1
Tipo		Singola
Data di inizio	CL_016	23/07/2011
Data di fine	CL_006	01/07/2011

TestCase: TestCaseRicercaDisponibilitàCamera 07		
Input	Classe	Membro
Numero camera	CL_008	103
Piano	CL_008	3
Numero Letti	CL_008	1
Tipo		Singola
Data di inizio	CL_016	12/06/2011
Data di fine	CL_006	01/06/2011

## 9.8 Associa Extra

Il test riguarda le funzionalità di associazione di un extra ad un cliente che consuma

### Classi di equivalenza

Gli input del test sono: Numero camera, quantità, codice extra, Data di consumazione

Input	Numero camera, quantità, codice extra, Data di consumazione	
Classi Valide	CL_008	Numero positivo maggiore di 0
	CL_006	Data corretta
	CL_022	Camera presente nel database
	CL_024	Codice extra corretto
Classi non valide	CL_009	Numero negativo
	CL_007	Data formato non corretto
	CL_023	Camera non presente nel database
	CL_025	Codice extra non corretto
	CL_005	Campo vuoto

### Identificazione case test

TestCase: TestCaseAssociaExtra 01		
Input	Classe	Membro
Numero camera	CL_009	-103
Quantità	CL_009	-3
Codice extra	CL_009	-1
Data consumazione	CL_007	23/06/2011

TestCase: TestCaseAssociaExtra 02		
Input	Classe	Membro
Numero camera	CL_009	-103
Quantità	CL_009	-3
Codice extra	CL_009	-1
Data consumazione	CL_007	23/16/2011

TestCase: TestCaseAssociaExtra 03		
Input	Classe	Membro
Numero camera	CL_008	203
Quantità	CL_009	-3



Codice extra	CL_008	1
Data consumazione	CL_006	

TestCase: TestCaseAssociaExtra 04		
Input	Classe	Membro
Numero camera	CL_008	203
Quantità	CL_009	-3
Codice extra	CL_024	1
Data consumazione	CL_006	14/03/2011

TestCase: TestCaseAssociaExtra 05		
Input	Classe	Membro
Numero camera	CL_008	203
Quantità	CL_009	-3
Codice extra	CL_025	300
Data consumazione	CL_006	14/12/2011

TestCase: TestCaseAssociaExtra 06		
Input	Classe	Membro
Numero camera	CL_022	203

Quantità	CL_008	3
Codice extra	CL_005	
Data consumazione	CL_006	14/12/2011

TestCase: TestCaseAssociaExtra 07		
Input	Classe	Membro
Numero camera	CL_023	2003
Quantità	CL_009	
Codice extra	CL_009	-11
Data consumazione	CL_006	14/12/2011

## 9.9 Rilascia Camera

Il test riguarda le funzionalità di rilascio di una camera

### Classi di equivalenza

Gli input del test sono: Numero Camera

Input	Numero Camera	
Classi Valide	CL_008	Numero positivo maggiore di 0
	CL_018	Camera presente nel database e occupata
Classi non valide	CL_009	Numero negativo
	CL_005	Campo vuoto
	CL_019	Camera non presente nel database o libera

### Identificazione case test

TestCase: TestCaseRilasciaCamera 01		
Input	Classe	Membro
Numero camera	CL_009	-103

### TestCase: TestCaseRicercaCamera 02

Input	Classe	Membro
Numero camera	CL_008	103

TestCase: TestCaseRicercaCamera 03		
Input	Classe	Membro
Numero camera	CL_019	205

TestCase: TestCaseRicercaCamera 04		
Input	Classe	Membro
Numero camera	CL_018	301

TestCase: TestCaseRicercaCamera 05		
Input	Classe	Membro
Numero camera	CL_005	

## 9.10 Prenota Camera

Il test riguarda le funzionalità di rilascio di una camera

### Classi di equivalenza

Gli input del test sono: Data inizio prenotazione, Data fine prenotazione, Numero camera, Nome, Cognome, Data di nascita

Input	Data inizio prenotazione, Data fine prenotazione, Numero camera, Nome. Cognome, Data di nascita	
Classi Valide	CL_008	Numero positivo maggiore di 0
	CL_006	Data corretta
	CL_011	Stringa minore uguale di 20 caratteri
	CL_022	Camera presente nel database e libera
	CL_014	Data inizio <= Data fine
	CL_020	Cliente registrato
Classi non valide	CL_009	Numero negativo
	CL_002	Stringa maggiore di 20 caratteri
	CL_007	Data formato non corretto
	CL_016	Data inizio > Data fine
	CL_023	Camera non presente nel database o occupata
	CL_021	Cliente non registrato
	CL_005	Campo vuoto

### Identificazione case test

TestCase: TestCasePrenotaCamera 01		
Input	Classe	Membro
Data inizio prenotazione	CL_006	12/03/2011
Data fine prenotazione	CL_007	15/33/11
Numero camera	CL_009	-103
Nome	CL_002	Francesco_Daniele_Raffaele
Cognome	CL_011	Calabrò
Data di nascita	CL_006	09/08/1989

TestCase: TestCasePrenotaCamera 02		
Input	Classe	Membro
Data inizio prenotazione	CL_007	12/33/11
Data fine prenotazione	CL_006	15/03/2011
Numero camera	CL_009	-103
Nome	CL_011	Francesco
Cognome	CL_002	Calabrò_Orciuoli_Costantino
Data di nascita	CL_007	09/18/89

**TestCase: TestCasePrenotaCamera 03**

Input	Classe	Membro
Data inizio prenotazione	CL_006	12/03/2011
Data fine prenotazione	CL_006	15/03/2011
Numero camera	CL_023	1020
Nome	CL_002	Francesco_Daniele_Raffaele
Cognome	CL_002	Calabrò_Orciuoli_Costantino
Data di nascita	CL_006	09/08/1989

**TestCase: TestCasePrenotaCamera 04**

Input	Classe	Membro
Data inizio prenotazione	CL_016	15/03/2011
Data fine prenotazione	CL_006	12/03/2011
Numero camera	CL_008	103
Nome	CL_002	Francesco_Daniele_Raffaele
Cognome	CL_005	
Data di nascita	CL_006	09/08/1989

**TestCase: TestCasePrenotaCamera 05**

Input	Classe	Membro
Data inizio prenotazione	CL_006	12/03/2011
Data fine prenotazione	CL_007	15/33/11
Numero camera	CL_009	-103
Nome	CL_005	
Cognome	CL_011	Calabrò
Data di nascita	CL_007	39/08/1989

TestCase: TestCasePrenotaCamera 06		
Input	Classe	Membro
Data inizio prenotazione	CL_006	12/03/2011
Data fine prenotazione	CL_007	15/33/11
Numero camera	CL_009	-103
Nome	CL_002	Francesco_Daniele_Raffaele
Cognome	CL_011	Calabrò
Data di nascita	CL_006	09/08/1989

TestCase: TestCasePrenotaCamera 07		
Input	Classe	Membro
Data inizio prenotazione	CL_006	12/03/2011



Data fine prenotazione	CL_005	
Numero camera	CL_008	203
Nome	CL_011	Francesco
Cognome	CL_011	Calabrò
Data di nascita	CL_006	09/08/1989

TestCase: TestCasePrenotaCamera 08		
Input	Classe	Membro
Data inizio prenotazione	CL_014	12/03/2011
Data fine prenotazione	CL_006	15/03/2011
Numero camera	CL_008	103
Nome	CL_011	Francesco
Cognome	CL_011	Orciuoli
Data di nascita	CL_006	09/08/1989

TestCase: TestCasePrenotaCamera 09		
Input	Classe	Membro
Data inizio prenotazione	CL_007	12/33/11
Data fine prenotazione	CL_006	15/03/2011
Numero camera	CL_022	102

Nome	CL_020	Francesco
Cognome	CL_002	Calabrò_Orciuoli_Costantino
Data di nascita	CL_007	09/18/89

TestCase: TestCasePrenotaCamera 10		
Input	Classe	Membro
Data inizio prenotazione	CL_007	12/33/11
Data fine prenotazione	CL_006	15/03/2011
Numero camera	CL_023	1002
Nome	CL_021	Francesco
Cognome	CL_021	Calabrò
Data di nascita	CL_007	09/18/89

# 3. Pianificazione

Il testing verrà effettuato dai componenti del gruppo che sono anche gli sviluppatori del sistema. Il modo migliore di effettuare il testing è quello di farlo provare a persone che non conoscono il funzionamento e che quindi sono estranei alla fase di implementazione. A causa di mancanza di personale e di tempo siamo costretti ad effettuare noi il testing del nostro sistema.

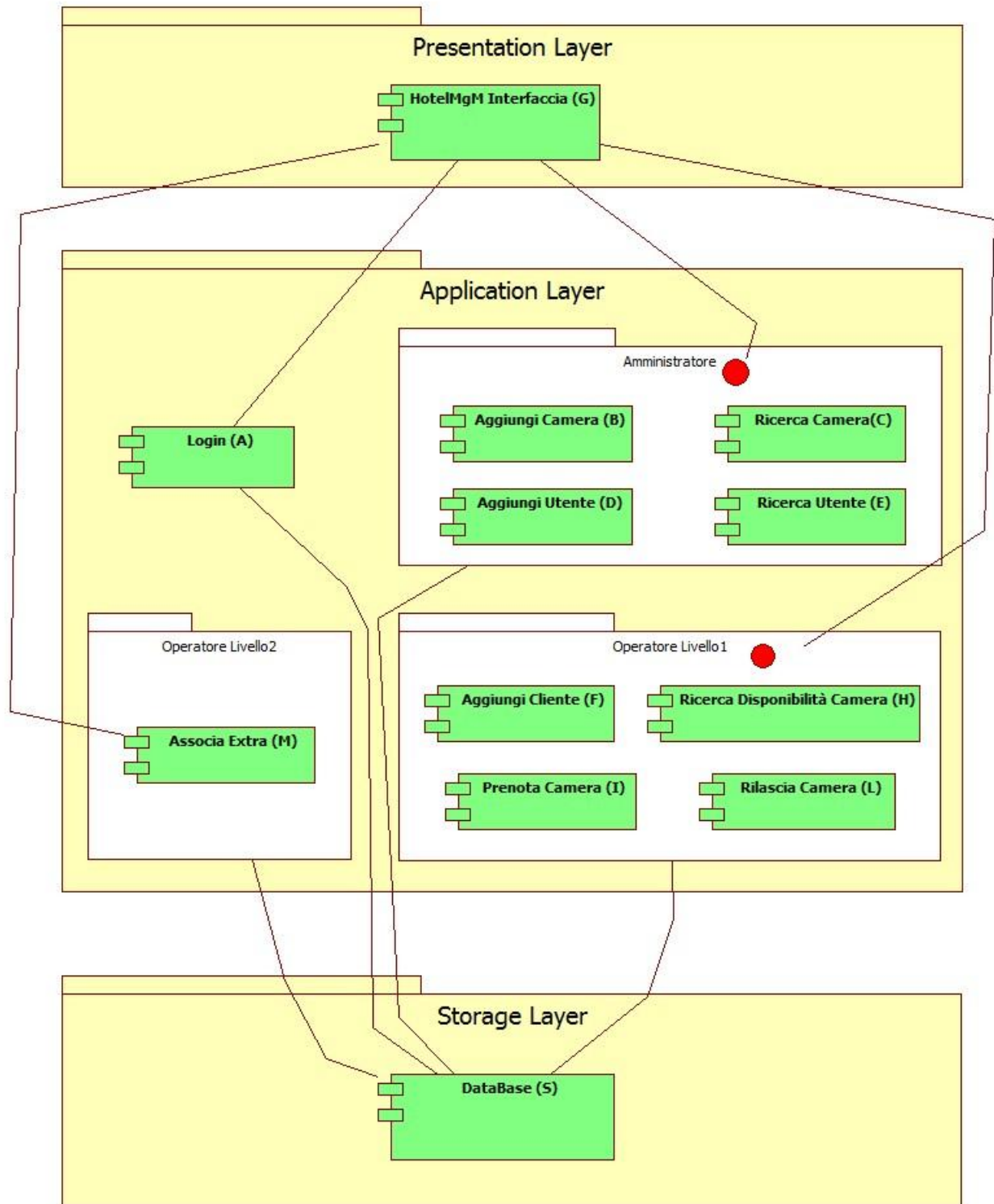
Ricordiamo in particolare un dettaglio, che l'implementazione è stata realizzata suddividendola in sottosistemi ed ogni componente del team ha sviluppato una parte di sistema, quindi poiché ciascun componente non conosce o meglio non è a conoscenza delle eccezioni e degli errori considerati dal collega del team, ogni componente del team eseguirà il testing di una parte del sistema che è stata implementata da un altro sviluppatore.

Prevediamo 4 giorni di tempo per completare il testing di tutte le componenti implementate.

Per quanto riguarda la qualità, si cercherà di scovare il più alto numero possibile di errori. Gli eventuali fault di una componente verranno notificati e saranno prontamente corretti.

Dopo la correzione e la revisione, il sistema verrà nuovamente testato per verificare se le modifiche apportate per risolvere gli errori individuati non ne abbiamo introdotti di nuovi.

Allo scopo di schedulare nel migliore dei modi la fase di testing, si è provveduto a rimodellare i sottosistemi che costituiscono il prodotto software completo, al fine di evidenziare le componenti che saranno sottoposte al processo e descrivere lo svolgimento dell'intera attività.



Di seguito viene invece mostrata una rappresentazione grafica e dettagliata dell'approccio utilizzato per organizzare e schedare il lavoro, seguita dal rispettivo

diagramma di Pert, che ha lo scopo di evidenziare l'intero iter di sviluppo e completamento della fase di testing.

