

Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino



**TP**

Test Plan

**HotelSmart**

Riferimento	G9_TP_ver.1.0
Versione	1.0
Data	13/12/2021
Destinatario	Prof. Carmine Gravino
Presentato da	Team G9
Approvato da	Raffaele Sais



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

### Revision history

<b>Data</b>	<b>Versione</b>	<b>Descrizione</b>	<b>Autore</b>
3/12/2021	0.1	Prima stesura	AD
06/12/2021	0.2	Scrittura parti testuali	RS
09/12/2021	0.3	Scrittura test frame	Tutto il team
11/12/2021	0.4	Completamento test case	AD
13/12/2021	0.5	Revisione e correzione degli errori	AD
01/02/2022	0.6	Correzione finale test frame	AD
12/02/2022	1.0	Correzione e revisione degli errori	Tutto il team



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## Team members

Nome	Ruolo nel progetto	Acronimo	Info di contatto
Alessandro D'Esposito	Team member	AD	<a href="mailto:desposito2@studenti.unisa.it">desposito2@studenti.unisa.it</a>
Pierpaolo Cammardella	Team member	PC	<a href="mailto:p.cammardella@studenti.unisa.it">p.cammardella@studenti.unisa.it</a>
Raffaele Sais	Project manager	RS	<a href="mailto:r.sais@studenti.unisa.it">r.sais@studenti.unisa.it</a>
Giovanni De Pierro	Team member	GD	<a href="mailto:g.depiero@studenti.unisa.it">g.depiero@studenti.unisa.it</a>



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## **Sommario**

1. Introduzione
2. Relazioni con altri documenti
3. Panoramica del sistema
4. Feature da testare/da non testare
5. Pass/Fail criteria
6. Approccio
7. Sospensione e ripristino
8. Materiale di testing
9. Test cases
10. Testing schedule



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

# 1.Introduzione

L'obiettivo di HotelSmart è quello di migliorare e semplificare l'esperienza utente e l'utilizzo sia per i clienti che per i gestori degli Hotel.

Il presente documento di TestPlan serve a descrivere tutte le attività di testing che sono eseguite per identificare gli errori nel sistema.

L'obiettivo è quindi, una volta identificati gli errori, andarli a sistemare in modo da far funzionare il sistema in modo corretto.

Le attività di testing pianificate sono:

- Gestione camere
- Gestione prenotazioni
- Gestione Servizi
- Gestione Utenti
- Autenticazione
- Modifica Anagrafica



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 2. Relazione con altri documenti

Per l'individuazione dei test case ci siamo relazionati ai nostri documenti scritti precedentemente, ovvero:

**Relazioni con il Requirement Analysis Document RAD:**

I TestCase sono presi in base ai requisiti funzionali e non funzionali

**SDD:**

I Test case devono rispettare la divisione fatta in sottosistemi presente nel SDD

## 3. Panoramica del sistema

Il sistema proposto si basa su un'architettura 3 tier, le tecnologie utilizzate invece saranno:

- Per la gestione della logica e quindi il back-end verranno utilizzate le servlet (quindi Java)
- Per il lato front-end verrà utilizzato HTML5, CSS3 e JS
- Per la gestione del database verrà utilizzato SQL e per il collegamento i DAO



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 4. Funzionalità da testare e non testare

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

- **Gestione Camere:**
  - Creazione camera
  - Modifica camera
  
- **Gestione Prenotazioni:**
  - Ricerca camera
  - Prenotazione camera
  - Modifica stato prenotazione
  
- **Gestione Servizi:**
  - Prenotazione servizio
  
- **Anagrafica:**
  - Modifica anagrafica
  - Modifica password
  
- **Gestione Utenti:**
  - Autenticazione
  - Registrazione

Le funzionalità sono state scelte principalmente guardando la priorità dei requisiti funzionali associati, inoltre sono state escluse le funzionalità che non prevedono input manuale da parte dell'utente come, ad esempio, sola visualizzazione di dati.



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 5. Criteri di Pass/Fail

Il risultato della fase di testing può essere un **Fail** o un **Pass**, questo risultato è determinato dal confronto dall'output del test e l'oracolo (l'output che ci aspettiamo).

Dove se i due output sono uguali allora c'è un Pass, altrimenti un Fail.

In seguito, elenchiamo i vincoli per il quale il testing viene considerato come valido:

- Testing di tutti i requisiti funzionali con alta priorità
- Per ogni modifica e /o aggiunta testare di nuovo tutti i casi di test
- Avere un branch coverage superiore al 75%

## 6. Approccio

Il testing effettuato si decompone in tre attività: Testing di unità, Testing di integrazione e Testing di sistema. Verranno progettati ed eseguiti nello stesso ordine appena elencato.

Durante tutto lo sviluppo saranno eseguite periodiche attività di revisione sul codice.

Adesso in dettaglio le operazioni che andremo ad effettuare durante le varie fasi di test:



## Testing di Unità:

Verranno testati tutti i metodi delle classi del sistema, escludendo i Bean ed il Singleton per la connessione al Database, in quanto i Bean presentano solo metodi Get e/o Set.

L'approccio utilizzato sarà che ogni qualvolta viene terminata la realizzazione di una classe, si dovrà procedere con la realizzazione della relativa classe di Test, seguendo il pattern:

***NomeClasseTest.java***

La classe di Test sarà successivamente revisionata da uno sviluppatore differente, ma sarà comunque lui a realizzarla.

Le tecnologie utilizzate saranno:

- *Mockito*: Per la creazione di stub
- *jUnit*: Framework per il testing
- *JaCoCo*: Per visualizzare report di Line/File/Branch coverage
- *Maven*: Per build ed esecuzione dei Test

## Testing di Integrazione:

L'approccio utilizzato sarà di tipo **Bottom-Up**, in quanto ritenuto il migliore per un software OO, utilizzeremo **jUnit** per la definizioni dei casi di test, **Mockito** per il mocking e **JaCoCo** per l'esportazione dei report per i vari coverage elencati precedentemente.

Il testing verrà suddiviso in due fasi, durante la prima fase andremo a testare le componenti **Service**, mentre nella seconda fase andremo a testare i **Controller**.

L'idea è anche di utilizzare il servizio GitHub CI/CD, ma per mancanza di tempo verrà preso in considerazione come un obiettivo secondario e quindi non si garantisce la sua realizzazione.

## Testing di Sistema

Per il seguente tipo di testing sarà utilizzato Selenium IDE per registrare le azioni che un utente può effettuare da un browser web, in modo da implementare ed effettuare i test case di sistema.

**Functional testing:**

Il goal sarà quello di validare tutti i requisiti funzionali ed individuare i possibili faults generati dagli utenti durante l'inserimento di un form

**Performance testing:**

E' stato utilizzata la metodologia di configuration testing e security testing.

Dove abbiamo provato il sistema in vari macchine e/o SO differenti, inoltre abbiamo provato ad effettuare chiamate API e/o link dove non avevamo i permessi per provare a violare il sistema

**Pilot testing:**

Non sarà effettuato il pilot testing

**Acceptance testing:**

Il Project Manager simulerà la figura del cliente

**Installation testing:**

Non sarà effettuato l'installation testing

**Ispezione del codice:**

L'intenzione è quella di utilizzare BetterCodeHub, un tool gratuito che revisiona il progetto ed evidenzia gli aspetti positivi e negativi basandosi su 10 requisiti.

Oltre a l'utilizzo di BetterCodeHub si proverà ad utilizzare anche GitHub CI/CD, ma a causa del poco budget e della mancanza di tempo verrà preso in considerazione come un obiettivo secondario e quindi non si garantisce la sua realizzazione.



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 7. Sospensione e ripristino

### **Criteri di sospensione:**

Per criterio di sospensione indichiamo quando dobbiamo fermare l'operazione di testing.

Nel nostro caso il testing verrà sospeso solo alla sua terminazione, anche se dovesse trovare una failure precedentemente.

### **Criteri di ripristino:**

Per criterio di ripristino indichiamo quando dobbiamo riprendere l'operazione di testing.

Nel nostro caso il testing riprende una volta analizzato e corretto l'errore trovato durante la fase di testing.

## 8. Materiale di testing

Utilizzeremo un computer per le attività di test, non è necessaria una connessione ad Internet in quanto il server sarà avviato in locale.



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 9. Test cases

TC\_REQ\_01 Ricerca camere

### Vincoli

Data inizio sia successiva alla data attuale
Data fine sia successiva alla data inizio
Numero persone sia un numero maggiore di 1

Parametri	Categorie
Data inizio	Inserimento

Parametro: Data inizio	
Categoria	Scelte
Inserimento (ii)	1. DataInizio < DataAttuale [errore] 2. DataInizio >= DataAttuale [property <b>inserimentoIlok</b> ]

Parametri	Categorie
Data fine	Inserimento

Parametro: Data fine	
Categoria	Scelte
Inserimento (if)	1. DataFine < DataInizio [errore] 2. DataFine >= DataInizio [property <b>inserimentoIFok</b> ]

Parametri	Categorie
-----------	-----------

Numero adulti	Formato, Valore
---------------	-----------------

Parametro: Numero Persone	
Categoria	Scelte
Valore (va)	1. Valore < 1 [errore] 2. Valore >=1 [property <b>ValoreVAok</b> ]

Codice	Combinazione	Esito
TC_1_01	ii1	Errore
TC_1_02	li2.if1	Errore
TC_1_03	ii2.if2.va1	Errore
TC_1_04	ii2.if2.va2	Corretto



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ\_02 Prenotazione Camera

### Vincoli

Nome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Cognome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Codice fiscale rispetta il formato [A-z0-9] e sia di lunghezza pari a 16 caratteri
Data di nascita sia precedente alla data attuale

Parametri	Categorie
<b>Nome</b>	Formato,Lunghezza

Parametro: Nome	
Formato: [A-z'-( *)]	
Categoria	Scelte
Formato (fn)	1. Rispetta il formato [property <b>formatoFNok</b> ] 2. Non rispetta il formato [errore]
Lunghezza (ln)	1. Lunghezza < 2 [if <b>formatoFNok</b> ] [errore] 2. Lunghezza > 30 [if <b>formatoFNok</b> ] [errore] 3. Lunghezza >=2 && Lunghezza <= 30 [if <b>formatoFNok</b> ] [property <b>lunghezzaLNok</b> ]

Parametri	Categorie
<b>Cognome</b>	Formato, Lunghezza

Parametro: Cognome	
Formato: [A-z'-( *)]	
Categoria	Scelte
Formato (fc)	1. Rispetta il formato [property <b>formatoFCok</b> ] 2. Non rispetta il formato [errore]
Lunghezza (lc)	1. Lunghezza < 2 [if <b>formatoFCok</b> ] [errore] 2. Lunghezza > 30 [if <b>formatoFCok</b> ] [errore]

	3. Lunghezza >=2 && Lunghezza <= 30 [if <b>formatoFCok</b> ] [property <b>lunghezzaLCok</b> ]
--	-----------------------------------------------------------------------------------------------

Parametri	Categorie
<b>Codice fiscale</b>	Formato, Lunghezza

Parametro: Codice fiscale	
Formato: [A-z0-9]	
Categoria	Scelte
Formato (ff)	1. Rispetta il formato [property <b>formatoFFok</b> ] 2. Non rispetta il formato [errore]
Lunghezza (lf)	1. Lunghezza != 16 [if <b>formatoFFok</b> ] [errore] 2. Lunghezza = 16 [if <b>formatoFFok</b> ] [property <b>lunghezzaLFok</b> ]

Parametri	Categorie
<b>Data di nascita</b>	Inserimento

Parametro: Data di nascita	
Categoria	Scelte
Inserimento (id)	1. DataNascita > DataAttuale [errore] 2. DataNascita <= DataAttuale [property <b>inserimentoIDok</b> ]

Codice	Combinazione	Esito
TC_2_01	fn2	Errore
TC_2_02	fn1.ln1	Errore
TC_2_03	fn1.ln2	Errore
TC_2_04	fn1.ln3.fc2	Errore
TC_2_05	fn1.ln3.fc1.lc1	Errore
TC_2_06	fn1.ln3.fc1.lc2	Errore
TC_2_07	fn1.ln3.fc1.lc3.ff2	Errore
TC_2_08	fn1.ln3.fc1.lc3.ff1.lf1	Errore
TC_2_09	fn1.ln3.fc1.lc3.ff1.lf2.id1	Errore
TC_2_10	fn1.ln3.fc1.lc3.ff1.lf2.id2	Corretto



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ\_04 Prenotazione Servizio

### Vincoli

Data prenotazione sia una data pari o successiva alla data attuale
Numero persone prenotate per il servizio deve avere un valore maggiore di 1

Parametri	Categorie
<b>Data prenotazione</b>	inserimento

Parametro: Data prenotazione	
Categoria	Scelte
Inserimento (id)	<ol style="list-style-type: none"> <li>Successiva o pari alla data attuale [dataPrenotazione &gt;= dataAttuale][property <b>inserimentoIDok</b>]</li> <li>Precedente alla data attuale [dataPrenotazione &lt; dataAttuale][errore]</li> </ol>

Parametri	Categorie
<b>Numero persone prenotate</b>	inserimento

Parametro: Numero persone prenotate	
Categoria	Scelte
Valore(vp)	<ol style="list-style-type: none"> <li>Valore &lt; 1 [errore]</li> <li>Valore &gt;= 1 [property <b>valoreVPok</b>]</li> </ol>

Codice	Combinazione	Esito
TC_4_01	id2	Errore
TC_4_02	id1.vp1	Errore
TC_4_03	id1.vp2	Corretto





Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ11 Registrazione

### Vincoli

Nome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Cognome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Codice fiscale rispetta il formato [A-z 0-9] e sia di lunghezza pari a 16 caratteri
Data di nascita sia precedente a 18 anni dalla data attuale
Indirizzo Email sia nel formato [A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4} e non sia già presente nel database
Password sia di lunghezza minima 8 caratteri e rispetti il formato (?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[\]:;',?+\*~\$^=<>])

Parametri	Categorie
<b>Nome</b>	Formato, Lunghezza

Parametro: Nome	
Formato: [A-z'-( *)]	
Categoria	Scelte
Formato (fn)	<ol style="list-style-type: none"> <li>Rispetta il formato [property <b>formatoFNok</b>]</li> <li>Non rispetta il formato [errore]</li> </ol>
Lunghezza(ln)	<ol style="list-style-type: none"> <li>Lunghezza &lt; 2 [if <b>formatoFNok</b>] [errore]</li> <li>Lunghezza &gt; 30 [if <b>formatoFNok</b>] [errore]</li> <li>Lunghezza &gt;=2 &amp;&amp; Lunghezza &lt;= 30 [if <b>formatoFNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>

Parametri	Categorie
<b>Cognome</b>	Formato, Lunghezza

Parametro: Cognome	
Formato: [A-z'-( *)]	
Categoria	Scelte

Formato (fc)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFCok</b>]</li> <li>2. Non rispetta il formato [errore]</li> </ol>
Lunghezza(lc)	<ol style="list-style-type: none"> <li>1. Lunghezza &lt; 2 [if <b>formatoFCok</b>] [errore]</li> <li>2. Lunghezza &gt; 30 [if <b>formatoFCok</b>] [errore]</li> <li>3. Lunghezza &gt;=2 &amp;&amp; Lunghezza &lt;= 30 [if <b>formatoFCok</b>] [property <b>lunghezzaLCok</b>]</li> </ol>

Parametri	Categorie
<b>Codice fiscale</b>	Formato, Lunghezza

Parametro: Codice fiscale	
Formato: [A-z 0-9]	
Categoria	Scelte
Formato (ff)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFFok</b>]</li> <li>2. Non rispetta il formato [errore]</li> </ol>
Lunghezza (lf)	<ol style="list-style-type: none"> <li>1. Lunghezza != 16 [if <b>formatoFFok</b>] [errore]</li> <li>2. Lunghezza = 16 [if <b>formatoFFok</b>] [property <b>lunghezzaLFok</b>]</li> </ol>

Parametri	Categorie
<b>Data di nascita</b>	Inserimento

Parametro: Data di nascita	
Formato: [DD-MM-YYYY]	
Categoria	Scelte
Inserimento (id)	<ol style="list-style-type: none"> <li>1. DataNascita &gt; DataAttuale – 18 anni [if <b>formatoFDok</b>][errore]</li> <li>2. DataNascita &lt;= DataAttuale – 18 anni [if <b>formatoFDok</b>] [property <b>inserimentoIDok</b>]</li> </ol>

Parametri	Categorie
<b>Indirizzo Email</b>	Formato, Match

Parametro: Indirizzo Email	
Formato: [A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}	
Categoria	Scelte
Formato (fe)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFEok</b>]</li> </ol>

	2. Non rispetta il formato [errore]
Match (me)	1. Match con email = false [if <b>formatoFEok</b> ] [property <b>matchMEok</b> ] 2. March con email = true [if <b>formatoFEok</b> ][errore]

Parametri	Categorie
<b>Password</b>	Lunghezza, Formato

Parametro: Password	
Formato: (?=[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[{}:;',?+\*\$^=<>])	
Categoria	Scelte
Formato (fp)	1. Rispetta il formato [property <b>formatoFPok</b> ] 2. Non rispetta il formato [errore]
Lunghezza (lp)	1. LunghezzaPassword < 8 [if <b>formatoFPok</b> ] [errore] 2. LunghezzaPassword >= 8 [if <b>formatoFPok</b> ] [property <b>lunghezzaLPok</b> ]

Codice	Combinazione	Esito
TC_11_01	fn2	Errore
TC_11_02	fn1.ln1	Errore
TC_11_03	fn1.ln2	Errore
TC_11_04	fn1.ln3.fc2	Errore
TC_11_05	fn1.ln3.fc1.lc1	Errore
TC_11_06	fn1.ln3.fc1.lc2	Errore
TC_11_07	fn1.ln3.fc1.lc3.ff2	Errore
TC_11_08	fn1.ln3.fc1.lc3.ff1.lf1	Errore
TC_11_09	fn1.ln3.fc1.lc3.ff1.lf2.id1	Errore
TC_11_10	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe2	Errore
TC_11_11	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me2	Errore
TC_11_12	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me1.fp2	Errore
TC_11_13	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me1.fp1.lp1	Errore
TC_11_14	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me1.fp1.lp2	Corretto



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ12 Login

### Vincoli

Indirizzo Email sia nel formato [A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}
Password sia di lunghezza minima 8 caratteri e rispetti il formato (?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[\]{};:',?+\*~\$^=<>])

Parametri	Categorie
Indirizzo Email	Formato

Parametro: Indirizzo Email	
Formato: [A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}	
Categoria	Scelte
Formato (fe)	<ol style="list-style-type: none"><li>Rispetta il formato [property <b>formatoFEok</b>]</li><li>Non rispetta il formato [errore]</li></ol>
Match (me)	<ol style="list-style-type: none"><li>Match con email = false [if <b>formatoFEok</b>][error]</li><li>Match con email = true [if <b>formatoFEok</b>][property <b>matchMEok</b>]</li></ol>

Parametri	Categorie
Password	Lunghezza, Formato

Parametro: Password	
Formato: (?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[\]{};:',?+\*~\$^=<>])	
Categoria	Scelte
Formato (fp)	<ol style="list-style-type: none"><li>Rispetta il formato [property <b>formatoFPok</b>]</li><li>Non rispetta il formato [errore]</li></ol>
Lunghezza(lp)	<ol style="list-style-type: none"><li>LunghezzaPassword &lt; 8 [if <b>formatoFPok</b>] [errore]</li><li>LunghezzaPassword &gt;= 8 [if <b>formatoFPok</b>] [property <b>lunghezzaLPok</b>]</li></ol>
Match(mp)	<ol style="list-style-type: none"><li>Match con password = false [if <b>formatoFPok</b> &amp;&amp; <b>lunghezzaLPok</b>][error]</li></ol>

	2. Match con password = true [if <b>formatoFPok</b> ] && <b>lunghezzaLPok</b> ][property <b>matchMPok</b> ]
--	-------------------------------------------------------------------------------------------------------------

Codice	Combinazione	Esito
TC_12_01	fe2	Errore
TC_12_02	fe1.me1	Errore
TC_12_03	fn1.me2.fp2	Errore
TC_12_04	fn1.me2.fp1.lp1	Errore
TC_12_05	fn1.me2.fp1.lp2.mp1	Errore
TC_12_06	fn1.me2.fp1.lp2.mp2	Corretto



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ13 Modifica anagrafica

### Vincoli

Nome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Cognome rispetta il formato [A-z'-( *)] e sia di lunghezza compresa tra 2 e 30 caratteri
Codice fiscale rispetta il formato [A-z 0-9] e sia di lunghezza pari a 16 caratteri
Data di nascita sia precedente a 18 anni dalla data attuale
Indirizzo Email sia nel formato [A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4} e non sia già presente nel database

Parametri	Categorie
<b>Nome</b>	Formato,Lunghezza

Parametro: Nome	
Formato: [A-z'-( *)]	
Categoria	Scelte
Formato (fn)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFNok</b>]</li> <li>2. Non rispetta il formato [errore]</li> </ol>
Lunghezza(ln)	<ol style="list-style-type: none"> <li>1. Lunghezza &lt; 2 [if <b>formatoFNok</b>] [errore]</li> <li>2. Lunghezza &gt; 30 [if <b>formatoFNok</b>] [errore]</li> <li>3. Lunghezza &gt;=2 &amp;&amp; Lunghezza &lt;= 30 [if <b>formatoFNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>

Parametri	Categorie
<b>Cognome</b>	Formato, Lunghezza

Parametro: Cognome	
Formato: [A-z'-( *)]	
Categoria	Scelte

Formato (fc)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFCok</b>]</li> <li>2. Non rispetta il formato [errore]</li> </ol>
Lunghezza(lc)	<ol style="list-style-type: none"> <li>1. Lunghezza &lt; 2 [if <b>formatoFCok</b>] [errore]</li> <li>2. Lunghezza &gt; 30 [if <b>formatoFCok</b>] [errore]</li> <li>3. Lunghezza &gt;=2 &amp;&amp; Lunghezza &lt;= 30 [if <b>formatoFCok</b>] [property <b>lunghezzaLCok</b>]</li> </ol>

Parametri	Categorie
<b>Codice fiscale</b>	Formato, Lunghezza

Parametro: Codice fiscale	
Formato: [A-z 0-9]	
Categoria	Scelte
Formato (ff)	<ol style="list-style-type: none"> <li>3. Rispetta il formato [property <b>formatoFFok</b>]</li> <li>4. Non rispetta il formato [errore]</li> </ol>
Lunghezza (lf)	<ol style="list-style-type: none"> <li>3. Lunghezza != 16 [if <b>formatoFFok</b>] [errore]</li> <li>4. Lunghezza = 16 [if <b>formatoFFok</b>] [property <b>lunghezzaLFok</b>]</li> </ol>

Parametri	Categorie
<b>Data di nascita</b>	Inserimento

Parametro: Data di nascita	
Formato: [DD-MM-YYYY]	
Categoria	Scelte
Inserimento (id)	<ol style="list-style-type: none"> <li>1. DataNascita &gt; DataAttuale – 18 anni [if <b>formatoFDok</b>][errore]</li> <li>2. DataNascita &lt;= DataAttuale – 18 anni [if <b>formatoFDok</b>] [property <b>inserimentoIDok</b>]</li> </ol>

Parametri	Categorie
<b>Indirizzo Email</b>	Formato, Match

Parametro: Indirizzo Email	
Formato: [A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}	
Categoria	Scelte
Formato (fe)	<ol style="list-style-type: none"> <li>1. Rispetta il formato [property <b>formatoFEok</b>]</li> </ol>

	2. Non rispetta il formato [errore]
Match (me)	1. Match con email = false [if <b>formatoFEok</b> ] [property <b>matchMEok</b> ] 2. March con email = true [if <b>formatoFEok</b> ][errore]

Codice	Combinazione	Esito
TC_13_01	fn2	Errore
TC_13_02	fn1.ln1	Errore
TC_13_03	fn1.ln2	Errore
TC_13_04	fn1.ln3.fc2	Errore
TC_13_05	fn1.ln3.fc1.lc1	Errore
TC_13_06	fn1.ln3.fc1.lc2	Errore
TC_13_07	fn1.ln3.fc1.lc3.ff2	Errore
TC_13_08	fn1.ln3.fc1.lc3.ff1.lf1	Errore
TC_13_09	fn1.ln3.fc1.lc3.ff1.lf2.id1	Errore
TC_13_10	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe2	Errore
TC_13_11	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me2	Errore
TC_13_12	Fn1.ln3.fc1.lc3.ff1.lf2.id2.fe1.me1	Corretto





Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## TC\_REQ14 Modifica password

Vecchia Password sia di lunghezza minima 8 caratteri e rispetti il formato <code>(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&amp;()-[{};:',?+\*~\$^=&lt;&gt;])</code>
Nuova Password sia di lunghezza minima 8 caratteri e rispetti il formato <code>(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&amp;()-[{};:',?+\*~\$^=&lt;&gt;])</code>
Ripeti Password sia identica a Nuova Password

Parametri	Categorie
<b>Vecchia Password</b>	Lunghezza, Formato

Parametro: Vecchia Password	
Formato: <code>(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&amp;()-[{};:',?+\*~\$^=&lt;&gt;])</code>	
Categoria	Scelte
Formato (fv)	<ol style="list-style-type: none"> <li>Rispetta il formato [property <b>formatoFVok</b>]</li> <li>Non rispetta il formato [errore]</li> </ol>
Lunghezza (lv)	<ol style="list-style-type: none"> <li>LunghezzaPassword &lt; 8 [if <b>formatoFVok</b>] [errore]</li> <li>LunghezzaPassword &gt;= 8 [if <b>formatoFVok</b>] [property <b>lunghezzaLVok</b>]</li> </ol>

Parametri	Categorie
<b>Nuova Password</b>	Lunghezza, Formato

Parametro: Nuova Password	
Formato: <code>(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&amp;()-[{};:',?+\*~\$^=&lt;&gt;])</code>	
Categoria	Scelte
Formato (fn)	<ol style="list-style-type: none"> <li>Rispetta il formato [property <b>formatoFNok</b>]</li> <li>Non rispetta il formato [errore]</li> </ol>
Lunghezza (ln)	<ol style="list-style-type: none"> <li>LunghezzaPassword &lt; 8 [if <b>formatoFNok</b>] [errore]</li> <li>LunghezzaPassword &gt;= 8 [if <b>formatoFNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>

Parametri	Categorie
<b>Ripeti Password</b>	Match

Parametro: Ripeti Password	
Categoria	Scelte
Match (mr)	1. Match con Nuova Password = true [property <b>formatoMRok</b> ] 2. Match con Nuova Password = false[errore]

Codice	Combinazione	Esito
TC_14_01	fv2	Errore
TC_14_02	fv1.lv1	Errore
TC_14_03	Fv1.lv2.fn2	Errore
TC_14_04	Fv1.lv2.fn1.ln1	Errore
TC_14_05	Fv1.lv2.fn1.ln2.mr2	Errore
TC_14_06	Fv1.lv2.fn1.ln2.mr1	Corretto



Laurea Triennale in informatica – Università degli studi di Salerno

Corso di ingegneria del software – Prof. Carmine Gravino

## 10. Testing schedule

Le attività di pianificazione del testing avverranno subito dopo la fase di design necessaria per la pianificazione.

La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice.

L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema.

Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.