



Laurea Triennale in Informatica-Università di Salerno Corso di  
Ingegneria del Software- Prof C. Gravino

# Test Plan

## MediCare

Riferimento	
Versione	1.1
Data	02/02/2024
Destinatario	C. Gravino
Presentato da	Andrea Gisolfi, Giacomo Favale, Giovanni Nigro, Antonio Merola
Approvato da	



Laurea Triennale in Informatica-Università di Salerno Corso di  
Ingegneria del Software- Prof C. Gravino

## Team Composition

Ruolo	Nome	Acronimo	Contatti
Team Member	Andrea Gisolfi	AG	a.gisolfi4@studenti.unisa.it
Team Member	Giacomo Favale	GF	g.favale1@studenti.unisa.it
Team Member	Giovanni Nigro	GN	g.nigro32@studenti.unisa.it
Team Member	Antonio Merola	AM	a.merola29@studenti.unisa.it



## Sommario

1. Introduzione.....	5
2. Relazione con altri documenti.....	5
3. Panoramica del sistema .....	6
4. Funzionalità da testare/da non testare.....	6
5. Pass/fail criteria .....	6
6. Approccio.....	7
7. Sospensione e ripristino .....	7
8. Materiale di testing.....	7
9. Test cases.....	7
9.1 Gestione Ricerca Ospedali.....	8
9.1.1 Ricerca tramite località.....	8
9.2 Gestione Chatbot .....	8
9.2.1 Diagnosi preliminare .....	8
9.3 Gestione Prenotazione .....	9
9.3.1 Effettuare prenotazione .....	9
9.4 Gestione Utente.....	10
9.4.1 Registrazione al sistema.....	10



## Revision History

Data	Versione	Descrizione	Autori
04/12/2023	0.1	Prima stesura	AG, GF, GN, AM
04/12/2023	0.2	Definiti i primi test case	AG, GF, GN, AM
11/12/2023	0.3	Redatti capitoli 3-4-5	GF, GN
11/12/2023	0.3	Redatti capitolo 6-8-9	AG, AM
11/12/2023	0.3	Stabilito l'approccio (Cap. 6)	AG, GF, GN, AM
17/12/2023	1.0	Revisione	AG, GF, GN, AM
28/01/2024	1.1	Revisione Finale	AG, GF, GN, AM



## 1. Introduzione

MediCare è stato concepito e progettato come una piattaforma versatile compatibile con Windows, offrendo un sistema completo per prenotazioni, localizzazione e diagnosi. L'obiettivo primario di MediCare è di fornire un servizio all'altezza delle aspettative nel medio-lungo termine, coinvolgendo utenti di tutte le fasce d'età e incentivando un'interazione più diffusa con il settore medico.

I criteri di successo stabiliti per MediCare includono il rispetto delle scadenze concordate con il committente, l'implementazione completa dei requisiti ad alta priorità nel sistema e la creazione di un'interfaccia accattivante e intuitiva per persone di tutte le età.

Il benessere e la salute sono sempre stati tra le massime priorità della popolazione, e negli ultimi anni, con l'emergenza COVID-19, è emerso chiaramente come una gestione inefficace degli ambienti pubblici e della sanità generale possa portare al caos totale. Questa è una delle missioni fondamentali di MediCare: fornire un sistema efficiente e intuitivo che si rivolga sia ai giovani che agli anziani. Inoltre, l'accessibilità alla sanità è un punto chiave di MediCare, che cerca di avvicinare le persone al mondo della salute attraverso un'interfaccia semplice ed intuitiva pensata per tutte le età.

## 2. Relazione con altri documenti

Vengono riportati in questa sezione i riferimenti ad altri documenti; i test case sono basati sulle funzionalità presenti nel documento di raccolta ed analisi dei requisiti.

### **Relazioni con il “Requirements Analysis Document” (RAD)**

I test case che sono stati individuati e pianificati in questo documento sono stati realizzati facendo riferimento ai requisiti funzionali e non, descritti nel documento di “Requirements Analysis Document”.

### **Relazioni con il “System Design Document” (SDD)**

I test case pianificati in questo documento sono stati suddivisi rispettando la suddivisione presente nel documento di “System Design Document”.



### 3. Panoramica del sistema

MediCare è basato secondo una architettura three tier e verrà sviluppato come software applicativo; la gestione della persistenza dei dati avverrà mediante MySQL (Oracle), e verranno usati Java ed il framework JavaFX rispettivamente per gestione della logica applicativa e generazione delle varie View.

### 4. Funzionalità da testare/da non testare

In questa sezione vengono riportate le features di cui si effettuerà il testing per le gestioni individuate, più precisamente:

- **Gestione ricerca ospedali**
  - Ricerca tramite località
- **Gestione chatbot**
  - Diagnosi preliminare
- **Gestione prenotazione**
  - Effettuare una prenotazione
- **Gestione Utente**
  - Registrazione al sistema

Le funzionalità che non richiedono l'input da parte dell'utente oppure che hanno il solo scopo di visualizzazione non verranno testati.

### 5. Pass/fail criteria

Le attività di test sono mirate all'identificazione di eventuali errori nel sistema, con l'obiettivo di intraprendere azioni successive per la loro correzione. I risultati dei casi di test sono determinati attraverso un oracolo, definito come il risultato previsto dell'esecuzione in base ai requisiti specificati.

Un test è considerato valido se soddisfa tutti i criteri definiti nelle specifiche e produce un output conforme a quanto atteso dall'oracolo.

In contrasto, un test è considerato fallito se non soddisfa uno o più criteri o se genera un output diverso da quanto previsto dall'oracolo. La corretta valutazione dei criteri "pass" e "fail" riveste un'importanza cruciale per garantire il corretto funzionamento del sistema e il rispetto dei requisiti stabiliti.



## 6. Approccio

Le fasi determinanti per la verifica del sistema sono tre:

- Test di unità, tramite category partition;
- Test di sistema, tramite category partition.

Precedentemente alla fase di implementazione del sistema, verranno creati prima i casi di test per il sistema stesso, che saranno poi successivamente ottimizzati durante la rispettiva esecuzione.

## 7. Sospensione e ripristino

La fase di testing non sarà interrotta fino a quando non sarà completata, anche nel caso in cui venga rilevato un fallimento; ciò nonostante, potrebbe essere temporaneamente sospeso nell'evenienza in cui si verifichi un errore nella definizione degli stessi test durante la sua esecuzione. In questo specifico caso, sarà necessario correggere la problematica prima di poter riprendere il testing.

Bisogna ricordare che completare il testing è importante affinché venga garantito che il sistema sia funzionante e che soddisfi i requisiti stabiliti. Nel caso in cui vengano effettuate delle modifiche al sistema, la fase di testing verrà ripresa.

È molto importante effettuare di nuovo i test per garantire ed assicurarsi che il sistema mantenga le sue funzionalità e la sua integrità anche dopo eventuali cambiamenti.

## 8. Materiale di testing

Per svolgere l'attività di testing è richiesto un dispositivo con sistema operativo Windows connesso a Internet, che fornisce accesso alla propria posizione.

## 9. Test cases

In questa sezione vengono riportati i casi di test per le funzionalità da testare.



## 9.1 Gestione Ricerca Ospedali

### 9.1.1 Ricerca tramite località

Parametro: Località	
Nome categoria	Scelta per la categoria
CampoVuotoLocalità [CVL]	<ol style="list-style-type: none"><li>1. Il campo località è un campo vuoto: [CATEGORY_CVL_ERROR]</li><li>2. Il campo località presenta il nome di una località: [CATEGORY_CVL_ERROR]</li></ol>
NomeInesistenteLocalità [NIL]	<ol style="list-style-type: none"><li>1. Il nome inserito non rispecchia un nome reale di località: [CATEGORY_NIL_ERROR]</li><li>2. Il nome inserito è un nome reale di una località esistente: [CATEGORY_NIL_ERROR]</li></ol>

Test case ID	Test Frame	Esito
TC_1.1_1	CVL1	Errore: non è stata inserita una località.
TC_1.1_2	CVL2,NIL1	Errore: la località inserita non esiste.
TC_1.1_3	CVL2,NL2	Successo.

## 9.2 Gestione Chatbot

### 9.2.1 Diagnosi preliminare

Parametro: Sintomo	
Nome categoria	Scelta per la categoria
CheckSintomo [CS]	<ol style="list-style-type: none"><li>1. Non viene inserito il sintomo: [CATEGORY_CS_ERROR]</li><li>2. Viene inserito un sintomo: [CATEGORY_CS_SUCCESS]</li></ol>
Validità [V]	<ol style="list-style-type: none"><li>1. Il sintomo inserito non è valido: [CATEGORY_V_ERROR]</li><li>2. Il sintomo inserito risulta essere valido: [CATEGORY_V_SUCCESS]</li></ol>





Test case ID	Test Frame	Esito
TC_1.2_1	CS1	Errore: non è stato inserito il sintomo.
TC_1.2_2	CS2,V1	Errore: è stato inserito un sintomo non valido.
TC_1.2_3	CS2,V2	Successo.

## 9.3 Gestione Prenotazione

### 9.3.1 Effettuare prenotazione

Parametro: Nome	
Nome categoria	Scelta per la categoria
CheckNomeP [CNP]	<ol style="list-style-type: none"> <li>1. Il nome è un campo vuoto: [CATEGORY_CNP_ERROR]</li> <li>2. Il nome contiene lettere: [CATEGORY_CNP_SUCCESS]</li> </ol>
Parametro: Cognome	
Nome categoria	Scelta per la categoria
CheckCognomeP [CCP]	<ol style="list-style-type: none"> <li>1. Il cognome è un campo vuoto: [CATEGORY_CCP_ERROR]</li> <li>2. Il cognome contiene lettere: [CATEGORY_CCP_SUCCESS]</li> </ol>
Parametro: Codice fiscale	
Nome categoria	Scelta per la categoria
Formato [F]	<ol style="list-style-type: none"> <li>1. Il codice fiscale non rispetta il formato /^[A-Z]{6}\d{2}[A-Z]\d{2}[A-Z]\d{3}[A-Z]\$/i: [CATEGORY_F_ERROR]</li> <li>2. Il codice fiscale rispetta il formato: [CATEGORY_F_SUCCESS]</li> </ol>
Parametro: Data	
Nome categoria	Scelta per la categoria
CheckData [CD]	<ol style="list-style-type: none"> <li>1. Il campo data è vuoto: [CATEGORY_CD_ERROR]</li> <li>2. Il campo data è compilato: [CATEGORY_CD_SUCCESS]</li> </ol>



Parametro: Ora	
Nome categoria	Scelta per la categoria
CheckOra [CO]	1. Il campo ora è vuoto: [CATEGORY_CO_ERROR] 2. Il campo ora è compilato: [CATEGORY_CO_SUCCESS]

Test case ID	Test Frame	Esito
TC_1.3_1	CNP1	Errore: il campo nome risulta vuoto.
TC_1.3_2	CNP2, CCP1	Errore: il campo cognome risulta vuoto.
TC_1.3_3	CNP2, CCP2, F1	Errore: il codice fiscale non rispetta il formato richiesto.
TC_1.3_4	CNP2, CCP2, F2, CD1	Errore: il campo data risulta vuoto.
TC_1.3_5	CNP2, CCP2, F2, CD2, CO1	Errore: il campo ora risulta vuoto.
TC_1.3_6	CNP2, CCP2, F2, CD2, CO2	Successo.

## 9.4 Gestione Utente

### 9.4.1 Registrazione al sistema

Parametro: Nome	
Nome categoria	Scelta per la categoria
CheckNome [CNR]	1. Il campo nome è vuoto: [CATEGORY_CNR_ERROR] 2. Il campo nome è compilato: [CATEGORY_CNR_SUCCESS]
FormatoNome [FN]	1. Il campo nome contiene caratteri speciali e/o numeri: [CATEGORY_FN_ERROR] 2. Il campo nome contiene solo lettere: [CATEGORY_FN_SUCCESS]
Parametro: Cognome	
Nome categoria	Scelta per la categoria
CheckCognomeR [CCR]	1. Il campo cognome è vuoto:



	[CATEGORY_CCR_ERROR] 2. Il campo cognome è compilato: [CATEGORY_CCR_SUCCESS]
FormatoCognome [FC]	1. Il campo cognome contiene caratteri speciali: [CATEGORY_FC_ERRORR] 2. Il campo cognome contiene solo lettere: [CATEGORY_FC_SUCCESS]
<b>Parametro: Data di Nascita</b>	
<b>Nome categoria</b>	<b>Scelta per la categoria</b>
CheckAnno [CA]	1. Anno corrente – anno di nascita < 18: [CATEGORY_CA_ERROR] 2. Anno corrente – anno di nascita >= 18: [CATEGORY_CA_SUCCESS]
<b>Parametro: Email</b>	
<b>Nome categoria</b>	<b>Scelta per la categoria</b>
FormatoEmail [FE]	1. L'email non contiene @ e/o . dopo di essa: [CATEGORY_FE_ERROR] 2. L'email rispetta il formato richiesto: [CATEGORY_FE_SUCCESS]
<b>Parametro: Password</b>	
<b>Nome categoria</b>	<b>Scelta per la categoria</b>
LunghezzaPassword [LP]	1. Lunghezza < 8 : [CATEGORY_LP_ERROR] 2. Lunghezza >= 8 : [CATEGORY_LP_SUCCESS]
<b>Parametro: Numero di telefono</b>	
<b>Nome categoria</b>	<b>Scelta per la categoria</b>
FormatoNumeroDiTelefono [FNT]	1. Il campo numero contiene lettere e/o caratteri speciali: [CATEGORY_FNT_ERROR] 2. Il campo numero contiene solo numeri: [CATEGORY_FNT_SUCCESS]



Test case ID	Test Frame	Esito
TC_1.4_1	CNR1	Errore: il campo nome è vuoto.
TC_1.4_2	CNR2, FN1	Errore: il nome contiene caratteri speciali.
TC_1.4_3	CNR2, FN2, CCR1	Errore: il campo cognome è vuoto.
TC_1.4_4	CNR2, FN2, CCR2, FC1	Errore: il cognome contiene caratteri speciali.
TC_1.4_5	CNR2, FN2, CCR2, FC2, CA1	Errore: età inferiore ai 18 anni.
TC_1.4_6	CNR2, FN2, CCR2, FC2, CA2, FE1	Errore: l'email non rispetta il formato richiesto.
TC_1.4_7	CNR2, FN2, CCR2, FC2, CA2, FE2, LP1	Errore: la password non rispetta la lunghezza richiesta.
TC_1.4_8	CNR2, FN2, CCR2, FC2, CA2, FE2, LP2, FNT1	Errore: il numero contiene caratteri non ammessi.
TC_1.4_9	CNR2, FN2, CCR2, FC2, CA2, FE2, LP2, FNT2	Successo.