



**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**Agriturismo “Il Miglio”**  
**TP – Test Plan**  
**Versione 2.0**



**Agriturismo Il Miglio**

Anno Accademico 2018/2019

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP - Test Plan	Data: 14/02/2019

## Top Manager:

<b>Professore</b>
De Lucia Andrea

## Partecipanti:

Nome	Matricola
Cusano Romina	0512104608
Ferrara Federica	0512104962
Garelli Luca	0512104944
Iannone Giancarlo	0512104632

<b>Scritto da:</b>	Membri del Team
--------------------	-----------------

## Revision History

Data	Versione	Descrizione	Autore
04/01/2019	1.0	Stesura del documento	Membri del Team
07/01/2019	1.1	Stesura del documento	Membri del Team
14/02/2019	2.0	Revisione	Membri del Team

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

# Indice

<b>1. INTRODUZIONE</b>	4
<b>2. DOCUMENTI CORRELATI</b>	4
2.1 Relazioni con il documento di analisi dei requisiti (RAD)	4
2.2 Relazioni con il System Design Document (SDD)	5
2.3 Relazioni con l’Object Design Document(ODD)	5
<b>3. PANORAMICA DEL SISTEMA</b>	5
<b>4. FUNZIONALITÀ DA TESTARE</b>	6
4.1 Gestione registrazione	6
4.2 Gestione autenticazione	6
4.3 Gestione utenti	6
4.4 Gestione prodotti	6
4.5 Gestione prenotazioni	6
<b>5. CRITERI PASS\FAILED</b>	7
<b>6. APPROCCIO</b>	7
6.1 Testing di unità	7
6.2 Testing di sistema	7
<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</b>	8
<b>9. TEST CASES</b>	9
9.1 Gestione Registrazione	9
9.2 Gestione Autenticazione	12
9.3 Gestione Utenti	13
9.4 Gestione Prodotti	14
9.5 Gestione Prenotazioni	17
<b>10. GLOSSARIO</b>	18

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

# 1. INTRODUZIONE

Lo scopo del Test Plan è quello di gestire lo sviluppo e le attività di test riguardanti il software AIM. Verranno pertanto individuati: vari elementi e funzionalità da testare, le strategie di testing e gli strumenti che saranno utilizzati, gli autori dei test, le risorse e le attività richieste per completare i test e i rischi associati al piano. Lo scopo del testing è quello di rilevare errori in maniera pianificata all'interno del codice realizzato al fine di evitare che tali errori si presentino durante l'utilizzo del sistema da parte dell'utente finale. I risultati uscenti dai test saranno utilizzati per capire dove intervenire per correggere gli errori o apportare modifiche con l'intenzione di migliorare il sistema.

In tale documento, verranno prese in considerazione ed analizzate le seguenti attività:

- Gestione Registrazione;
- Gestione Autenticazione;
- Gestione Utenti;
- Gestione Prodotti;
- Gestione Prenotazioni;

Per ogni gestione saranno testate solo le funzionalità contenente i requisiti con priorità **Alta**.

## 2. DOCUMENTI CORRELATI

Il test plan ha una stretta relazione con il resto dei documenti che sono stati prodotti finora poiché il sistema è stato pianificato nelle precedenti documentazioni e implementato in parte. Nella fase di testing si verificheranno le eventuali somiglianze tra il sistema desiderato e quello proposto. Di seguito verranno indicate le relazioni con i precedenti documenti.

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

La relazione tra test plan e RAD (Requirement Analysis Document) riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test verranno eseguiti su quelle funzionalità tenendo conto delle specifiche espresse nel precedente documento.

In particolare il RAD contiene lo scopo del sistema, l'ambito del sistema e gli obiettivi, mostrando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi d'uso, diagrammi e mockup del sistema.

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

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

L’SDD (System Design Document) rappresenta l’architettura del sistema MVC. In particolare, l’SDD contiene l’architettura del software corrente e proposto e i servizi dei sottosistemi.

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

In questa sezione, verrà indicato il riferimento all’ODD (Object Design Document), documento in cui vengono evidenziate le specifiche legate alle interfacce per ogni sottosistema, definite le librerie per le classi e selezionati i design pattern per risolvere problemi comuni e proteggere le classi da interventi futuri.

# **3. PANORAMICA DEL SISTEMA**

Come descritto e analizzato nell’ SDD la struttura del nostro sistema segue il pattern architetturale MVC (Model/View/Control).

L’architettura MVC si basa su tre componenti principali:

- Model (M): rappresenta lo stato centrale della nostra applicazione web (database, file, informazioni di sessione, etc.)
- View (V): sono una serie di “viste”, interfacce utente del modello.
- Controller (C): una serie di applicazioni che racchiudono la logica applicativa del nostro sito, attraverso delle modifiche al modello.

Per seguire tale architettura abbiamo deciso di dividere il nostro sistema in sottosistemi più piccoli basandoci sui diversi tipi di gestione:

- Gestione Registrazione;
- Gestione Autenticazione;
- Gestione Utenti;
- Gestione Prodotti;
- Gestione Prenotazioni.

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 4. FUNZIONALITÀ DA TESTARE

### 4.1 *Gestione registrazione*

- Registrazione

### 4.2 *Gestione autenticazione*

- Login

### 4.3 *Gestione utenti*

- Modifica Profilo

### 4.4 *Gestione prodotti*

- Creazione Prodotto
- Rimozione Prodotto
- Modifica Prodotto
- Acquisto Prodotto

### 4.5 *Gestione prenotazioni*

- Preleva Prenotazioni
- Effettua Prenotazione
- Annulla Prenotazione

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 5. CRITERI PASS\FAILED

I dati di input del test saranno raggruppati in insiemi dalle caratteristiche comuni in modo da effettuare un test su di un unico elemento rappresentativo. Il testing ha successo se l’output osservato è diverso dall’output atteso: ciò significa che la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altre componenti del sistema. La failure quindi è uno stato di condizione nel quale non si trova l’output desiderato, si può dire che è il contrario di un successo.

## 6. APPROCCIO

L’approccio alla fase di testing si divide in 3 fasi:

- La prima servirà a testare le componenti una ad una;
- La seconda prevede di testare le funzionalità delle integrazioni dei vari sottosistemi;
- La terza si occupa di testare l’intero sistema assemblato per verificare soprattutto che esso soddisfi le richieste del cliente.

### 6.1 *Testing di unità*

Per realizzare il testing di ogni singola componente verrà utilizzata la tecnica “Black-Box Testing”. Utilizzando tale tecnica si andrà ad analizzare ed esaminare le funzionalità dell’applicazione e il comportamento input/output delle singole componenti senza dare importanza alla loro struttura interna. Essendo quasi impossibile generare tutti i possibili input, verranno create classi d’equivalenza scegliendo per ognuna un test case per ridurre la ridondanza e rendere il test più efficiente. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema.

### 6.2 *Testing di sistema*

Prima di essere pronto all’uso, il sistema affronterà l’ultima fase di testing, quello di sistema, per dimostrare che siano soddisfatti tutti i requisiti richiesti. Lo scopo di questa fase è testare le funzionalità più importanti, usate maggiormente e con maggior probabilità di fallimento. Trattandosi di un sistema web-based verrà utilizzato, per effettuare il testing di sistema, il tool Selenium. Selenium si occupa di simulare l’interazione con il sistema dal punto di vista dell’utente.

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 7. SOSPENSIONE E RIPRESA

### 7.1 *Criteri di sospensione*

La fase di testing del sistema verrà sospesa quando si otterranno i risultati attesi in accordo con i tempi di sviluppo previsti, tenendo sempre conto dei costi dell’attività di testing.

Questo processo verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

### 7.2 *Criteri di ripresa*

La fase di testing potrà riprendere in seguito a modifiche o correzioni che generano errori o fallimenti, i test case verranno, quindi, sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema

## 8. MATERIALE PER IL TESTING

Gli strumenti necessari per l’attività di test sono una macchina ed una connessione ad Internet visto che, il database del sistema è stato caricato online per renderlo disponibili a tutti.



Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 9. TEST CASES

### 9.1 Gestione Registrazione

#### REGISTRAZIONE

<b>Parametro</b> → E-mail	
Formato: [A-Za-z0-9._%+-] @ [A-Za-z] [.A-Za-z]. [A-Za-z]	
<b>Categorie</b>	Scelte
<b>Formato [FE]</b>	1. E-mail==null [error] 2. Non rispetta il formato [error] 3. E-mail !=null AND Rispetta il formato [property FE_OK]
<b>Esiste [EE]</b>	1. E-mail non esistente nel Database [property EE_OK]

<b>Parametro</b> → Password	
<b>Categorie</b>	Scelte
<b>Lunghezza [LP]</b>	1. Lunghezza < 4 OR lunghezza >8 [errore]  2. Lunghezza >= 4 AND lunghezza <= 8 [property LP_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

<b>Parametro</b> → Nome Formato: $^{[A-Za-z]}+(\backslash s[A-Za-z])^{+}$ *	
<b>Categorie</b>	Scelte
<b>Lunghezza [LN]</b>	1. Lunghezza < 3 OR lunghezza >20 [errore] 2. Lunghezza >= 3 AND lunghezza <= 20 [property LN_OK]
<b>Formato [FN]</b>	1. Non rispetta il formato [if LN_OK][errore] 2. Rispetta il formato [if LN_OK][property FN_OK]

<b>Parametro</b> → Cognome Formato: $^{[A-Za-z]}+(\backslash s[A-Za-z])^{+}$ *	
<b>Categorie</b>	Scelte
<b>Lunghezza [LC]</b>	1. Lunghezza < 3 OR lunghezza >20 [errore] 2. Lunghezza >= 3 AND lunghezza <= 20 [property LC_OK]
<b>Formato [FC]</b>	1. Non rispetta il formato [if LC_OK][errore] 2. Rispetta il formato [if LC_OK][property FC_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

<b>Parametro → Indirizzo</b> Formato: [A-Za-z0-9\.\,\s]+	
<b>Categorie</b>	Scelte
<b>Lunghezza [LI]</b>	1. Lunghezza < 3 OR > 60 [errore] 2. Lunghezza >= 3 AND <= 60 [property LI_OK]
<b>Formato [FI]</b>	1. Non rispetta il formato [if LI_OK][errore] 2. Rispetta il formato [if LI_OK][property FI_OK]

<b>Parametro → Telefono</b> Formato: ^[0-9]{10}	
<b>Categorie</b>	Scelte
<b>Formato [FT]</b>	1. Non rispetta il formato [errore] 2. Rispetta il formato [property FT_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 9.2 Gestione Autenticazione

### LOGIN

<b>Parametro</b> → E-mail	
Formato: [A-Za-z0-9._%+-] @ [A-Za-z] [.A-Za-z]. [A-Za-z]	
<b>Categorie</b>	Scelte
<b>Formato [FE]</b>	1. E-mail==null [error] 2. Non rispetta il formato [error] 3. E-mail !=null AND Rispetta il formato [property FE_OK]
<b>Esiste [FE]</b>	1. E-mail non esistente nel Database [error] [if property FE_OK] 2. E-mail esistente nel Database [property EE_OK]

<b>Parametro</b> → Password	
<b>Categorie</b>	Scelte
<b>Match [MP]</b>	1. Match con password utente == false [error] 2. Match con password utente == true [property MP_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

### 9.3 Gestione Utenti

#### MODIFICA PROFILO

<b>Parametro</b> → Numero di telefono Formato: <b>^[0-9]{10}</b>	
<b>Categorie</b>	Scelte
<b>Formato [FT]</b>	1. Non rispetta il formato [errore] 2. Rispetta il formato [property_FT OK]

<b>Parametro</b> → Indirizzo Formato: <b>[A-Za-z0-9\.\,\s]+</b>	
<b>Categorie</b>	Scelte
<b>Lunghezza [LI]</b>	1. Lunghezza < 3 OR > 60 [errore] 2. Lunghezza >= 3 AND <= 60 [property LI_OK]
<b>Formato [FI]</b>	1. Non rispetta il formato [if LI_OK][errore] 2. Rispetta il formato [if LI_OK][property FI_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 9.4 Gestione Prodotti

### CREA PRODOTTO

<b>Parametro</b> → Codice Formato: <b>[0-9]*</b>	
<b>Categorie</b>	Scelte
<b>Lunghezza [LCod]</b>	1. Lunghezza < 3 OR > 12 [errore] 2. Lunghezza >= 3 AND <= 12[property LCod_OK]
<b>Formato [FCod]</b>	1. Non rispetta il formato [if LCod_OK][errore] 2. Rispetta il formato [if LI_OK][property FI_OK]

<b>Parametro</b> → Nome Formato: <b>^[A-Za-z]+(\s[A-Za-z]+)*</b>	
<b>Categorie</b>	Scelte
<b>Lunghezza [LN]</b>	1. Lunghezza < 3 OR > 60 [errore] 2. Lunghezza >= 3 AND <= 60 [property LT_OK]
<b>Formato [FN]</b>	1. Non rispetta il formato [if LT_OK][errore] 2. Rispetta il formato [if LT_OK][property FT_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

<b>Parametro</b> → Data Produzione Formato: <b>^[0-9]{2}\/[0-9]{2}\/[0-9]{4}</b>	
<b>Categorie</b>	Scelte
<b>Formato [FDP]</b>	1. Non rispetta il formato [errore] 2. Rispetta il formato [property FDN_OK]

<b>Parametro</b> → Data Scadenza Formato: <b>^[0-9]{2}\/[0-9]{2}\/[0-9]{4}</b>	
<b>Categorie</b>	Scelte
<b>Formato [FDS]</b>	1. Non rispetta il formato [errore] 2. Rispetta il formato [property FDS_OK]

<b>Parametro</b> → Quantità Formato: <b>[0-9]</b>	
<b>Categorie</b>	Scelte
<b>Lunghezza [LQ]</b>	1. Lunghezza < 0 OR > 5 [errore] 2. Lunghezza >= 0 AND <= 5 [property LQ_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

<b>Formato [FQ]</b>	1. Non rispetta il formato [if LQ_OK][errore] 2. Rispetta il formato [if LQ_OK][property FQ_OK]
---------------------	--

<b>Parametro → Prezzo</b> Formato: <b>[0-9,0-9]</b>	
<b>Categorie</b>	Scelte
<b>Lunghezza [LP]</b>	1. Lunghezza < 1 OR > 5 [errore] 2. Lunghezza >= 1 AND <= 5 [property LP_OK]
<b>Formato [FP]</b>	1. Non rispetta il formato [if LP_OK][errore] 2. Rispetta il formato [if LP_OK][property FP_OK]

<b>Parametro → Immagine</b> Formato: <b>.jpg</b>	
<b>Categorie</b>	Scelte
<b>Formato [FIM]</b>	1. Immagine != null AND Formato != .jpg [errore] 2. (Immagine == null) OR (Immagine != null AND Formato == .jpg) [property FIm OK]



Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 9.5 Gestione Prenotazioni

### EFFETTUA PRENOTAZIONE

<b>Parametro</b> → Data Prenotazione Formato: $^{[0-9]\{2\}}\backslash^{[0-9]\{2\}}\backslash^{[0-9]\{4\}}$	
<b>Categorie</b>	Scelte
<b>Formato [FPr]</b>	1. Rispetta il formato [property FPr_OK]

<b>Parametro</b> → Posti Formato: $[0-9]$	
<b>Categorie</b>	Scelte
<b>Lunghezza [LPos]</b>	1. Lunghezza $< 1$ OR $> 60$ [errore] 2. Lunghezza $\geq 1$ AND $\leq 60$ [property LPos_OK]
<b>Formato [FPos]</b>	1. Non rispetta il formato [if LPos_OK][errore] 2. Rispetta il formato [if LI_OK][property FI_OK]

Progetto: Agriturismo “Il Miglio”	Versione: 2.0
Documento: TP – Test Plan	Data: 14/02/2019

## 10. GLOSSARIO

RAD:	Requirement Analysis Document
SDD:	System Design Document
ODD:	Object Design Document
MVC:	Model View Controller
FAULT:	Fallimento del sistema
BBT:	Big Bang Testing
CPM:	Category Partition Method
FE:	Formato Email
EE:	Esiste Email
MP:	Match Password
LP:	Lunghezza Password
LN:	Lunghezza Nome
FN:	Formato Nome
LC:	Lunghezza Cognome
FC:	Formato Cognome
LI:	Lunghezza Indirizzo
FI:	Formato Indirizzo
FT:	Formato Telefono
FIM:	Formato Immagine
FT:	Numero di Telefono
LCod:	Lunghezza Codice
FCod:	Formato Codice
FDP:	Formato Data Produzione
FDS:	Formato Data Scadenza
LQ:	Lunghezza Quantità
FQ:	Formato Quantità
LPr:	Lunghezza Prezzo
FPr:	Formato Prezzo
LPos:	Lunghezza Posti
FPos:	Formato Posti