UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Ingegneria Software

FATTORIA ROSABELLA
TEST PLAN
VERSIONE 1.0

FATTORIA ROSABELLA 1992

15/01/2021

Progetto: FATTORIA ROSABELLA	Versione: 1.0
Documento: Test Plan	Data: 15/01/2021

Coordinatore del progetto:

Nome	**Matricola**
Mignoli Pasquale Antonio	0512105826

Partecipanti:

Nome	**Matricola**
Mignoli Pasquale Antonio	0512105826
Pastore Carmine	0512106058
Laurino Nicola	0512105970

Scritto da:	Mignoli, Laurino, Pastore

Revision History

Data	**Versione**	**Descrizione**	**Autore**
15/01/2021	1.0	Stesura primo documento	Nicola Laurino

Sommario

1.0 Introduzione

Il bioparco Fattoria Rosabella offre una serie di attività all'aria parte e immersi nel verde della natura. La necessità di stare al passo con lo sviluppo tecnologico e digitale porta sempre più azienda a rivolgersi al mondo dell'informatica per raggiungere il maggior numero di clienti possibili e per sfruttare tutto il potenziale offerto dai servizi informatici. Nasce così il bisogno, da parte del bioparco, di avere a disposizione un portale web che permette ai propri clienti di gestire le proprie prenotazioni comodamente da casa.

Il sito sviluppato deve essere allora rivisto al fine di rilevare errori commessi in fase di sviluppo e correggerli. Questa fase è necessaria e richiesta per evitare che errori logici si presentino durante la navigazione degli utenti finali.

Il sito è molto complesso, perciò, richiede una attenta rivisitazione con il fine di individuare e correggere il maggior numero di errori. Tale fase è importante per offrire un servizio affidabile e sicuro agli utenti finali che interagiscono con le funzionalità del sistema.

Con lo scopo di raggiungere gli obiettivi precedente citati, è stato stabilito un piano di testing specifico per determinati componenti del sistema: assicurare una corretta registrazione dell'utente, garantire un'efficace e sicura gestione della prenotazione, inserire agevolmente nuove attività e fornire cautele per l'aggiunta di nuove carte di credito per l'utente.

Una corretta interpretazione degli output prodotti dai test permetterà una corretta individuazione e correzione di eventuali errori commessi in fase di sviluppo.

Lo scopo di questa fase è, dunque, stabilire la correttezza dei moduli sviluppati: bisogna stabilire una corrispondenza tra valore atteso e comportamento osservato dal sito di Fattoria Rosabella.

2.0 Documenti correlati

2.1 Relazione con Requirements Analysis Document

Nel documento di analisi dei requisiti vengono descritte, in modo molto dettagliato, le relazioni che si creano tra il sistema e gli utenti finali. Gli scenari e i casi d'uso forniscono un ottimo mezzo per controllare accuratamente che le funzionalità implementate dal sistema corrispondano con quelle descritte nel documento sopra citato.

2.2 Relazione con System Design Document

Nella documentazione di system design vengono descritti accuratamente i tre livelli logici in cui il sistema è stato diviso: view, controller e model. In particolare, per ciascun livello è stata fornita una descrizione dettagliata e accurata dei componenti che lo compongono, del loro ruolo e dei loro attributi.

2.3 Relazione con Object Design Document

Il documento di object design specifica le classi che compongono il sistema. Dunque, faremo riferimento a questo documento per recuperare le porzioni di codice interessata dai test cases.

3.0 Panoramica del sistema

Il sito sviluppato per Fattoria Rosabella è una piattaforma web che permette ai clienti del bioparco di prenotare le proprie attività preferite comodamente da casa e ad un gestore, operatore del bioparco, di poter gestire le attività e le prenotazioni direttamente dal web.

Lo sviluppo della piattaforma si è concentrata sul ruolo di tre attori principali:

- Un utente generico, visitatore del sito, non ancora registrato;
- Un utente registrato, il quale può accedere a quasi tutte le funzionalità del sistema e possiede un'area riservata dove gestire, in piena autonomia, il proprio profilo e le proprie informazioni;
- Un utente gestore, operai del bioparco, incaricato di gestire le attività e il calendario.

Durate lo sviluppo del sito web si sono distinti diversi sottosistemi, ognuno dei quali raggruppa una serie di funzionalità per un certo attore.

- **Prenotazione**.
  Un utente registrato accede al sito e, attraverso il menu di navigazione, accede alla sezione prenotazione. Qui seleziona una data e una categoria riuscendo così a filtrare le attività disponibili secondo le sue preferenze. A questo punto riesce a selezionare una o più attività e ad accedere alla sezione riepilogo dove può completare l'acquisto.
- **Registrazione**.
  Un utente generico accede al sito e, accedendo alla pagina di login, riesce a registrarsi al sistema. Tale registrazione prevede diverse fasi.
  - L'utente fornisce le sue informazioni, compreso un indirizzo e-mail valido, e preme su registrati.
  - Il sistema manda una e-mail di conferma all'indirizzo di posta elettronica fornito per verificare che quest'ultimo sia valido.
  - L'utente accede alla propria casella di posta elettronica e, attraverso il link fornito nell'e-mail, riesce a confermare il proprio indirizzo e-mail.
  - Il sistema conferma la registrazione dell'utente e abilita il profilo ad accedere alle funzionalità offerte dal sito.
- **Aggiunta di una nuova attività**.
  L'utente gestore, attraverso una propria area riservata, riesce ad inserire una nuova attività e a settare un calendario.
- **Gestione carte di credito**.
  All'utente registrato viene fornita la possibilità di gestire, in tutta autonomia, i propri metodi di pagamento, in particolare le carte di credito.

4.0 Funzionalità da testare

La fase di test case andrà a toccare diverse parti sviluppate per il sottosistema, escludendone altre. Questa scelta si rende necessaria per motivi di budget. Nonostante ciò, il testing andrà ad interessare le parti maggiormente delicate del sistema, ovvero quei sottosistemi che vanno a lavorare con dati sensibili, come carte di credito e informazioni personali, o che vanno a modificare informazioni importanti dell'azienda stessa, quali le attività offerte.

Come conseguenza delle scelte suddette, le funzionalità che andremo a testare solo le seguenti.

1. **Registrazione di un utente**.
   In questa fase, un utente che intende registrarsi andrà a fornire informazioni personali, quali nominativo, data di nascita e un indirizzo e-mail valido. Inoltre, al fine di evitare la registrazione di indirizzi di posta elettronica fasulli che possono compromettere la stabilità del sistema, viene introdotto un meccanismo di conferma dell'indirizzo e-mail fornito.
2. **Aggiunta di una nuova attività e popolamento del calendario**.
   Il bioparco Fattoria Rosabella offre come servizi al pubblico una serie di attività. Quest'ultima può essere disponibile per un certo periodo tempo o può variare nel tempo. Allora, è stato introdotto un sottosistema per la gestione delle attività il quale permette, in particolare, l'aggiunta di una nuova attività e la definizione del calendario in modo automatizzato.
3. **Prenotazione di una attività**.
   Una delle funzionalità di maggior spicco del sito. Si offre la possibilità di prenotare una o più attività ad un utente registrato direttamente dal sito. Offrire la migliore funzionalità possibile è uno dei nostri obiettivi più grandi.
4. **Aggiunta di una nuova carta di credito**.
   In questa fase vengono gestite informazioni di pagamento dell'utente, in particolare le carte di credito. Dunque, è necessario che questa funzionalità sia ben funzionante in quanto lavora con dati molto sensibili.

5.0 Criteri di successo/fallimento

Affinché il test funzioni correttamente, sono stati decisi una serie di input da fornire ai sottosistemi precedentemente individuati. Abbiamo poi individuato due possibili esiti dei test:

1. **Successo**.
   Il test ha successo nel caso in cui, fornendo gli input scelti, il sottosistema fornisce un comportamento diverso da quello atteso e indicato in un oracolo. Successivamente vengono individuati gli errori e corretti. Questa fase si ripete finché l'esito del test è diverso da successo.
2. **Fallimento**.
   Il test fallisce se il comportamento ottenuto dal sistema è proprio quello descritto nell'oracolo**.**

6.0 Approccio

Al fine di effettuare un testing quanto più completo e accurato, suddivideremo il processo in tre fasi.

1. Una prima fase si occuperà di andare a scovare errori nelle singole componenti coinvolte.
2. Una seconda fase andrà a testare le funzionalità individuate nell'integrazione dei vari sottosistemi.
3. Una terza, ed ultima fase, andrà a verificare che l'intero sistema funzione correttamente andando a soddisfare le richieste del cliente.

Abbiamo allora individuate una serie di approcci per effettuare testing.

6.1 Testing di unità

In questa fase ogni singola componente viene testata individualmente al fine di catturare tutti i possibili errori commessi durante la stesura del codice. Eseguiremo allora testing in modalità *black-box*: ci concentreremo unicamente sull'input e sull'output del componente, senza andare ad indagare nel codice.

6.2 Testing di integrazione

Testate le singole componenti e corretti eventuali errori riscontrati, possiamo integrarle in sottosistemi più grandi e andare a testare le funzionalità di questi ultimi. Adotteremo allora un approccio bottom-up: andremo a testare prima le componenti nei livelli più bassi e man mano quelle ai livelli superiori. Questo metodo ci permette di testare l'affidabilità dei livelli sviluppati.

6.3 Testing di sistema

Completati i due test precedenti, possiamo procedere ad effettuare un testing per l'intero sistema. Ciò ci permette di verificare l'integrità e l'affidabilità generale dell'intero sistema sviluppato, che tutti i sottosistemi riesco a comunicare correttamente tra loro senza generare conflitti.

Questa fase è conclusiva: la riuscita di questo testing ci permette di terminare la fase di testing del sistema ed il primo ciclo di sviluppo. Il framework *Selenium* ci permette di fornire diversi input e verificare il comportamento del sistema.

6.4 Testing di usabilità

Terminato il primo ciclo di testing, il sistema entra in modalità di beta-test. Utenti reali, che non conoscono come il sito è stato sviluppato, interagiscono con il sistema. Questa fase si rende necessaria per raccogliere feedback sull'affidabilità e l'usabilità del sistema.

Come più volte detto nei precedenti documenti, fornire un'interfaccia *user-friendly* è uno degli obiettivi del sistema. Il modo migliore per verificare che tale obiettivo sia stato raggiunto e lasciare che un utente reali accedi al sito ed effettui delle operazioni. Alla fine di tale interazione, raccogliamo i pareri degli utenti e andiamo a stilare una lista delle problematiche riscontrate.

7.0 Sospensione e ripresa

La fase di testing è molto lunga e complessa. Si rende allora necessario individuare dei criteri per stabilire come e quando un test può essere sospeso o può considerarsi completato.

7.1 Criteri di sospensione

Il test di un sottosistema viene sospeso nel momento in cui si verificano errori almeno nel 20% dei casi. In queste condizioni il team deve intervenire affinché vengano individuati e corretti tutti gli errori riscontrati prima di procedere con il testing.

7.2 Criteri di ripresa

Rilasciato il sistema, possono esserci modifiche future. Prevediamo allora la necessità di testare di nuovo i sottosistemi quando nuove funzionalità vengono introdotto. Ciò si rende necessario affinché il sistema continui a funzionare correttamente: le nuove funzionalità possono andare a toccare sottosistemi già presenti, occorre allora testare anche queste ultime.

7.3 Criteri di terminazione

Il test si considera *terminato* nel momento in cui tutti i test effettuati sul sistema danno esito negativo.

8.0 Materiale per il testing

I test verranno effettuati sul sito in esecuzione. L'esecuzione deve avvenire su un server correttamente configurato, il quale permette l'esecuzione di comandi Java e SQL. In particolare, tale server deve permettere di eseguire parallelamente il server di MySQL.

Il testing di unità viene effettuato con il framework JUnit in quanto il back end del sistema è stato sviluppato con linguaggio di programmazione Java. Andiamo a testare le classi interessati e individuare errori commessi in fase di sviluppo.

Il testing di sistema viene effettuato, invece, con Selenium, un framework per testare applicazione web. Esso fornisce uno strumento di riproduzione per creare test funzionali, i quali permettono di automatizzare il processo di inserimento di dati nei form al fine di verificare il comportamento del sistema.

9.0 Test cases

Andiamo a testare, per ciascun sottosistema, le funzionalità indicate precedentemente. Per ciascuna funzionalità viene riportato, in una tabella, il parametro e l'insieme delle possibili scelte.

9.1 Utente

Per il sottosistema individuato, andiamo a testera la funzionalità che permette l'iscrizione alla piattaforma. In particolare, viene mostrato un form di registrazione per permette l'inserimento dei dati personali necessari per l'iscrizione al sito. I dati richiesti sono i seguenti.

- **Nome**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Cognome**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Data di nascita**.
  Contiene una data, espressa secondo il formato:
  - Giorno, intero compreso tra 1 e 31;
  - Mese, intero compreso tra 1 e 12;
  - Anno di nascita, intero minore l'anno attuale.
- **Indirizzo**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Città**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **E-mail**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255 dove è obbligatoria la presenza del carattere *@*.
- **Password**.
  Contiene una stringa di caratteri con lunghezza compresa tra 6 e 255.
- **Ripeti password**.
  Contiene una stringa di caratteri con lunghezza compresa tra 6 e 255 la quale deve combaciare con la stringa contenuta nel campo *password*.

Parametro	**Nome**
NL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Cognome**
CL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Data di nascita**
DDN - Valore	1. Data non valida La data inserita non rispetta il formato *[ERROR]* 2. Data valida La data inserita rispetta il formato *[PROPRIETÀ RISPETTATA]*

Parametro	**Indirizzo**
IL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Città**
CL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**E-mail**
EL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*
Formato	/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/
EF – Formato	1. Formato non rispettato *[ERROR]* 2. Formato rispettato *[PROPRIETÀ RISPETTATA]*

Parametro	**Password**
PL - Lunghezza	1. < 6 OR > 255 *[ERROR]* 2. ≥ 6 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*
Formato	/^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{7,15}$/
PF – Formato	1. Formato non rispettato *[ERROR]* 2. Formato rispettato *[PROPRIETÀ RISPETTATA]*

Parametro	**Ripeti password**
RPL - Lunghezza	1. < 6 OR > 255 *[ERROR]* 2. ≥ 6 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*
Formato	/^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{7,15}$/
RPF - Formato	1. Formato non rispettato *[ERROR]* 2. Formato rispettato *[PROPRIETÀ RISPETTATA]*
RPF – Corrispondenza	1. La seconda password non corrisponde con la password inserita nel campo *password [ERROR]* 2. La seconda password corrisponde con la password inserita nel campo *password [PROPRIETÀ RISPETTATA]*

9.2 Attività

L'inserimento di una nuova attività e di un calendario è una funzionalità offerta ad un utente gestore. Il sistema fornisce un form attraverso il quale è possibile specificare tutte le informazioni relative alla nuova attività di aggiungere.

I campi da compilare sono i seguenti.

- **Nome**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Categoria**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Numero massimo di partecipanti**.
  Contiene un intero maggiore di 0.
- **Prezzo**.
  Contiene un intero maggiore di 0.
- **Descrizione.**
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Orario**.
  Contiene un insieme di orario, rappresentati attraverso stringhe con lunghezza compresa tra 1 e 255.
- **Giorni**.
  Contiene un insieme di giorni, rappresentati attraverso stringhe con lunghezza compresa tra 1 e 255.
- **Mesi**.
  Contiene un insieme di mesi, rappresentati attraverso stringhe con lunghezza compresa tra 1 e 255.

Parametro	**Nome**
NL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Categoria**
CatL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Numero massimo di partecipanti**
NMaxF - Formato	1. < 1 *[ERROR]* 2. ≥ 1 *[PROPRIETÀ RISPETTATA]*

Parametro	**Prezzo**
PF - Formato	1. < 1 *[ERROR]* 2. ≥ 1 *[PROPRIETÀ RISPETTATA]*

Parametro	**Descrizione**
DL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Orario**
OF - Formato	1. L'insieme è vuoto *[ERROR]* 2. L'insieme ha almeno un elemento *[PROPRIETÀ RISPETTATA]*

Parametro	**Giorni**
GF - Formato	1. L'insieme è vuoto *[ERROR]* 2. L'insieme ha almeno un elemento *[PROPRIETÀ RISPETTATA]*

Parametro	**Mesi**
MF - Formato	1. L'insieme è vuoto *[ERROR]* 2. L'insieme ha almeno un elemento *[PROPRIETÀ RISPETTATA]*

9.3 Prenotazione

Per effettuare una prenotazione, un utente registrato accede alla sezione *prenotazione* e sceglie una data e una categoria. Dopodiché, vengono visualizzate le attività disponibili per quella data e l'utente seleziona poi un orario e numero di partecipanti.

Di seguito i dati da inserire all'interno dei form.

- **Data**.
  Contiene una data, espressa secondo il formato:
  - Giorno, intero compreso tra 1 e 31 e maggiore o uguale al giorno attuale;
  - Mese, intero compreso tra 1 e 12 e maggiore o uguale al mese attuale;
  - Anno di nascita, intero maggiore o uguale all'anno attuale.
- **Categoria**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Orario**.
  Contiene un orario rappresentato attraverso una stringa con lunghezza compresa tra 1 e 255.
- **Partecipanti**.
  Contiene un intero maggiore di 0.

Parametro	**Data**
D - Valore	1. Data non valida La data inserita non rispetta il formato *[ERROR]* 2. Data valida La data inserita rispetta il formato *[PROPRIETÀ RISPETTATA]*

Parametro	**Categoria**
CatL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Orario**
OL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Partecipanti**
PF - Formato	1. < 1 *[ERROR]* 2. ≥ 1 *[PROPRIETÀ RISPETTATA]*

9.4 Carta di credito

Il sottosistema che gestisce l'inserimento di una carta di credito prevede un form dove l'utente può inserire i dati della propria carta di credito. Il form citato prevede i seguenti campi.

- **Numero**.
  Contiene una stringa di numeri con lunghezza uguale a 16.
- **Nome intestatario**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Cognome intestatario**.
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **CVV**.
  Contiene una stringa di numeri con lunghezza uguale a 3.
- **Mese di scadenza.**
  Contiene una stringa di caratteri con lunghezza compresa tra 1 e 255.
- **Anno di scadenza.**
  Contiene una stringa di caratteri con lunghezza uguale a 4 e deve essere maggiore a uguale alla stringa contenente l'anno attuale.

Parametro	**Numero**
NL - Lunghezza	1. Lunghezza diversa da 16 *[ERROR]* 2. Lunghezza uguale a 16 *[PROPRIETÀ RISPETTATA]*

Parametro	**Nome**
NL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Cognome**
CL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**CVV**
CvvL - Lunghezza	1. Lunghezza diversa da 3 *[ERROR]* 2. Lunghezza uguale a 3 *[PROPRIETÀ RISPETTATA]*

Parametro	**Mese di scadenza**
MdsL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*

Parametro	**Anno di scadenza**
AdsL - Lunghezza	1. < 1 OR > 255 *[ERROR]* 2. ≥ 1 OR ≤ 255 *[PROPRIETÀ RISPETTATA]*
Ads – Valore	1. Minore dell'anno attuale *[ERROR]* 2. Maggiore o uguale dell'anno attuale *[PROPRIETÀ RISPETTATA]*

10. Riferimenti ad altri documenti di test

//DOCUMENTI