

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TP  Test Plan  CoralloSmart   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.1 | | Data | 15/11/2022 | | Destinatario | ARPA | | Presentato da | Gerardo Iuliano e Antonio Trovato | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 15/11/2022 | 0.1 | Prima stesura | Gerardo Iuliano, Antonio Trovato |
| 20/11/2022 | 0.2 | Stesura completa | Gerardo Iuliano |
| 13/01/2023 | 0.3 | Aggiunto TC\_GM\_1.5 | Team |

Sommario

[RevisionHistory 2](#_Toc120955287)

[Test Plan CoralloSmart 4](#_Toc120955288)

[1. Introduzione 4](#_Toc120955289)

[2. Relazione con altri documenti 4](#_Toc120955290)

[3. Panoramica del sistema 4](#_Toc120955291)

[4. Feature da testare/da non testare 4](#_Toc120955292)

[5. Pass/Fail criteria 5](#_Toc120955293)

[6. Approccio 5](#_Toc120955294)

[7. Sospensione e rispristino 5](#_Toc120955295)

[7.1. Criteri di sospensione 5](#_Toc120955296)

[7.2. Criteri di ripristino 5](#_Toc120955297)

[8. Materiale di testing 5](#_Toc120955298)

[9. Test cases 6](#_Toc120955299)

[10. Testing schedule 6](#_Toc120955300)

Test Plan  
CoralloSmart

1. Introduzione

CoralloSmart nasce con l’obiettivo di supportare l’ARPA attraverso dei contributi economici per la salvaguardia della barriera. Inoltre, in base all’importo versato il sostenitore potrà essere premiato con dei voucher, allo scopo di incentivare ulteriori contributi, con l’obiettivo ultimo di disporre ulteriori dispositivi IoT per il monitoraggio della barriera corallina e delle acque. L’ente ha necessità di fornire uno strumento semplice da usare sia per i sostenitori che per i propri dipendenti che dovranno monitorare i sistemi IoT ed i contributi effettuati.

Tale documento ha l’obiettivo di descrivere l’attività di testing relativa al sistema CoralloSmart con il fine ultimo di garantire che ogni aspetto funzioni in modo corretto.

Di seguito viene riportata la strategia e l’approccio utilizzato per la fase di testing.

1. Relazione con altri documenti

Faremo riferimento ad altri documenti come:

* **RAD**: i test case sono realizzati sulla base dei requisiti funzionali e non funzionali.
* **SDD**: i test case rispetteranno la suddivisione in sottosistemi.

1. Panoramica del sistema

Il sistema proposto si basa su un’architettura three-tier.

Sarà sviluppato utilizzando Java per il back-end, HTML, CSS e Javascript per il front-end.

Per la gestione dei dati in modo persistente MySQL.

1. Feature da testare/da non testare

**Le feature da testare sono:**

Gestione Utente:

* Modifica profilo

Gestione Contributi

* Versamento contributo

Gestione Voucher

* Modifica voucher
* Aggiungi voucher

Gestione Monitoraggio

* Visualizza stato salute delle acque

**Le feature da non testare sono**:

Gestione Utente:

* Logout
* Creazione profilo responsabile ARPA
* Registrazione

Gestione Contributo

* Visualizzazione storico contributi sostenitore
* Visualizzazione storico contributi Responsabile ARPA
* Visualizzazione incassi contributi Responsabile ARPA
* Visualizzazione impatto di tutti i contributi

Gestione Voucher

* Visualizzazione voucher ottenuti
* Visualizzazione voucher ottenibili
* Ottieni voucher in seguito a versamento contributo
* Invia mail di riepilogo e voucher ottenuto
* Elimina voucher

Gestione Monitoraggio

* Visualizzazione stato dispostivi IoT
* Inserimento Rilevamento

1. Pass/Fail criteria

Il testing è finalizzato a identificare la presenza di fault nel sistema in modo da poter effettuare un intervento e rimuoverlo. Un test è composto da precondizione, input, output, oracolo e risultato.

Un test si considera:

* “*pass*” se l’output è diverso dall’oracolo.
* “*fail*” se l’output è uguale all’oracolo.

Il testing in generale, invece, si considera valido se:

* Ogni team member dovrà effettuare il testing di unità, tramite category partition, di esattamente un metodo di una classe sviluppata.
* Ogni team member dovrà effettuare il testing di sistema, tramite category partition, di esattamente una funzionalità del sistema sviluppato.

1. Approccio

Il testing è composto da tre parti: testing di unità, testing di integrazione e testing di sistema.

La progettazione dei test case di sistema ed integrazione sarà fatta prima della fase di implementazione del sistema. Saranno raffinati durante la loro esecuzione se necessario. I test case di unità saranno progettati durante l’implementazione.

L’esecuzione dei test case avverrà in parallelo alla stesura del codice, in modo da avere subito un riscontro. Per ogni nuova funzionalità implementata saranno eseguiti i test case di unità ad essa legata e i test di integrazione che la coinvolgono.

Ad ogni modifica di una componente implementata verranno rilanciati i test di unità e di integrazione delle componenti direttamente impattate dalla modifica. Inoltre, verrà effettuato il testing di regressione, rilanciando i test case delle componenti apparentemente non impattate o impattate in modo indiretto.

I test frame saranno definiti tramite il category partition, al fine di minimizzare il numero di test case rimuovendo eventuali combinazioni inutili; gli input saranno partizionati in classi di equivalenza.

**Testing di sistema:** sarà effettuato tramite il tool *Selenium IDE,* che permette di registrare delle interazioni che l’utente esegue su un browser, così da implementare ed eseguire i casi di test di sistema.

**Testing di integrazione:** verrà effettuato tramite approccio bottom-up, in quanto più adatto ad un software Object Oriented. I test saranno definiti tramite il framework *JUnit*, mentre verrà utilizzato *Mokito* per la creazione dei driver.

**Testing di unità:** andràa testare ogni metodo di ciascuna classe del sistema, fatta eccezione per le interfacce e le classi di domain, poiché quest’ultime presentano solo i metodi *getter* e *setter*autogenerati dal tool *Lombok.* Verrà definita una classe di test per ciascuna classe sorgente e i test saranno definiti tramite il framework *JUnit.*

Il test sarà gestito in modo automatico tramite *Maven* e verrà utilizzato *Travis CI* per la Continuous Integration.

1. Sospensione e rispristino

In questa sezione verranno specificati i criteri di sospensione del test.

* 1. Criteri di sospensione

La fase di testing non sarà sospesa fino alla sua terminazione anche in caso di una failure. Si potrà verificare una sospensione nel momento in cui l’esecuzione ritorna un errore nella definizione di uno dei test.

* 1. Criteri di ripristino

Il testing verrà ripreso solo dopo aver risolto i fault individuati nell’esecuzione precedente.

1. Materiale di testing

Per l’attività di testing, come hardware è sufficiente un computer.

1. Test cases
   1. System test cases

|  |  |
| --- | --- |
| **Aggiungi voucher** | |
| **Parametro** | Descrizione |
| Formato descrizione:  ^[a-zA-Z]+(.|\s)\*$ | |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [DL] | 1. Lunghezza > 50 or Lunghezza < 10 = false[error] 2. Lunghezza <= 50 and Lunghezza >=10 = true [property DL\_OK] |
| Formato [DF] | 1. Formato adatto = false [error] 2. Formato adatto = true [property DF\_OK] |
| **Parametro** | Importo |
| Formato importo:  ^[0-9]+$ | |
| **Categoria** | **Scelte per la categoria** |
| Formato [IF] | 1. Formato adatto = false [error] 2. Formato adatto = true [property IF\_OK] |
| Range [IR] | 1. Range > 150 or Range < 5 = false[error] 2. Range <= 150 and Range >=5 = true [property IR\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_GV\_A\_1.0 | DL1 | Errato: lunghezza descrizione non valida |
| TC\_GV\_A \_1.1 | DL2, DF1 | Errato: formato descrizione non valido |
| TC\_GV\_A \_1.2 | DL2, DF2, IF1 | Errato: formato importo non valido |
| TC\_GV\_A \_1.3 | DL2, DF2, IF2, IR1 | Errato: importo fuori range |
| TC\_GV\_A \_1.4 | DL2, DF2, IF2, IR2 | Corretto: voucher inserito correttamente |

|  |  |
| --- | --- |
| **Modifica voucher** | |
| **Parametro** | Descrizione |
| Formato descrizione:  ^[a-zA-Z]+(.|\s)\*$ | |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [DL] | 1. Lunghezza > 50 or Lunghezza < 10 = false[error] 2. Lunghezza <= 50 and Lunghezza >=10 = true [property DL\_OK] |
| Formato [DF] | 1. Formato adatto = false [error] 2. Formato adatto = true [property DF\_OK] |
| **Parametro** | Importo |
| Formato importo:  ^[0-9]+$ | |
| **Categoria** | **Scelte per la categoria** |
| Formato [IF] | 1. Formato adatto = false [error] 2. Formato adatto = true [property IF\_OK] |
| Range [IR] | 1. Range > 150 or Range < 5 = false[error] 2. Range <= 150 and Range >=5 = true [property IR\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_GV\_M \_1.0 | DL1 | Errato: lunghezza descrizione non valida |
| TC\_GV\_M \_1.1 | DL2, DF1 | Errato: formato descrizione non valido |
| TC\_GV\_M \_1.2 | DL2, DF2 | Corretto: modifica descrizione valida |
| TC\_GV\_M \_1.3 | DL2, DF2, IF1 | Errato: formato importo non valido |
| TC\_GV\_M \_1.4 | DL2, DF2, IF2, IR1 | Errato: importo fuori range |
| TC\_GV\_M \_1.5 | DL2, DF2, IF2, IR2 | Corretto: modifica valida |

|  |  |
| --- | --- |
| **Visualizza stato salute delle acque** | |
| **Parametro** | Data inizio |
| Formato data:  gg/mm/aaaa | |
| **Categoria** | **Scelte per la categoria** |
| Selezionato [IS] | 1. Data non selezionata [property IS\_OK] 2. Data selezionata [property IS\_OK] |
| Data [ID] | 1. Data inizio > Data fine [if IS\_OK] [error] 2. Data inizio <= Data fine [if IS\_OK] [property ID\_OK] |
| **Parametro** | Data fine |
| Formato data:  gg/mm/aaaa | |
| **Categoria** | **Scelte per la categoria** |
| Selezionato [FS] | 1. Data non selezionata [property FS\_OK] 2. Data selezionata [property FS\_OK] |
| Data [FD] | 1. Data fine < Data inizio [if FS\_OK] [error] 2. Data fine >= Data inizio [if FS\_OK] [property DD\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_GM\_1.0 | IS1 | Corretto: visualizza rilevamenti senza filtri |
| TC\_GM\_1.1 | IS2, ID1 | Errato: data non valida |
| TC\_GM\_1.2 | IS2, ID2, FS1 | Corretto: visualizza rilevamenti a partire da |
| TC\_GM\_1.3 | IS2, ID2, FS2, FD1 | Errato: data non valida |
| TC\_GM\_1.4 | IS2, ID2, FS2, FD2 | Corretto: visualizza rilevamenti nel range di date |
| TC\_GM\_1.5 | IS1, FS2, FD2 | Corretto: visualizza rilevamenti fino a |

|  |  |
| --- | --- |
| **Modifica profilo** | |
| **Parametro** | Username |
| Formato username:  “^[A-Za-z][A-Za-z0-9\_]{7,29}$” | |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [UL] | 1. Lunghezza > 29 or Lunghezza < 7 = false[error] 2. Lunghezza <= 29 and Lunghezza >=7 = true [property UL\_OK] |
| **Parametro** | Password |
| Formato password:  “^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[\*.!@$%^&(){}[]:;<>,.?/~\_+-=|\]).{8,32}$” | |
| **Categoria** | **Scelte per la categoria** |
| Formato [PF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [property PF\_OK] |
| **Parametro** | Conferma password |
| **Categoria** | **Scelte per categoria** |
| Match[PCM] | 1. Uguale al parametro Password = false [error] 2. Uguale al parametro Password = true [PCM\_OK] |
| **Parametro** | Email |
| Formato e-mail:  “^[A-z0-9.\_%+-]+@[A-z0-9.-]+\.[A-z]{2,10}$” | |
| **Categoria** | **Scelte per la categoria** |
| Formato [EF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [property EF\_OK] |
| **Parametro** | Nome |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [NL] | 1. Lunghezza > 30 or Lunghezza = 0 = false [error] 2. Lunghezza <= 30 and Lunghezza > 0 = true [property NL\_OK] |
| **Parametro** | Cognome |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [SL] | 1. Lunghezza > 30 or Lunghezza = 0 = false [error] 2. Lunghezza <= 30 and Lunghezza > 0 = true [property NL\_OK] |
| **Parametro** | Codice Fiscale |
| Formato codice fiscale:  “^[a-zA-Z]{6}[0-9]{2}[a-tA-T]{1}[0-9]{2}([a-zA-Z]{1}[0-9]{3})[a-zA-Z]{1}$” | |
| **Categoria** | **Scelte per la categoria** |
| Formato [FCF] | 1. Formato rispettato = false [error] 2. Formato rispettato = true [property FCF\_OK] |
| **Parametro** | Telefono |
| Formato telefono:  “^((00|\+)39[\. ]??)??3\d{2}[\. ]??\d{6,7}$” | |
| **Categoria** | **Scelte per la categoria** |
| Formato [TF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [property TF\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_GU\_2.0 | UL1 | Errato: lunghezza username non valida |
| TC\_GU\_2.1 | UL2, PF1 | Errato: username valido ma formato password non valido |
| TC\_GU\_2.2 | UL2, PF2, PCM1 | Errato: username valido, password valida ma conferma password non uguale a password |
| TC\_GU\_2.3 | UL2, PF2, PCM2, EF1 | Errato: username valido, password valida, conferma password uguale a password ma formato e-mail non valido |
| TC\_GU\_2.4 | UL2, PF2, PCM2, EF2, NL1 | Errato: username valido, password valida, conferma password uguale a password, formato e-mail valido, nome non valido |
| TC\_GU\_2.5 | UL2, PF2, PCM2, EF2, NL2, SL1 | Errato: username valido, password valida, conferma password uguale a password, formato e-mail valido, nome valido, cognome non valido |
| TC\_GU\_2.6 | UL2, PF2, PCM2, EF2, NL2, SL2, FCF1 | Errato: username valido, password valida, conferma password uguale a password, formato e-mail valido, nome valido, cognome valido, codice fiscale non valido |
| TC\_GU\_2.7 | UL2, PF2, PCM2, EF2, NL2, SL2, FCF2, TF1 | Errato: username valido, password valida, conferma password uguale a password, formato e-mail valido, nome valido, cognome valido, codice fiscale valido, formato numero di telefono non valido |
| TC\_GU\_2.8 | UL2, PF2, PCM2, EF2, NL2, SL2, FCF2, TF2 | Corretto: modifica effettuata |

|  |  |
| --- | --- |
| **Versamento contributo** | |
| **Parametro** | Intestatario |
| Formato intestatario:  “^[A-Za-z][a-zA-Z]\*$” | |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [AL] | 1. Lunghezza > 60 or Lunghezza = 0 = false[error] 2. Lunghezza <= 60 and Lunghezza > 0 = true [property AL\_OK] |
| **Parametro** | Numero della carta |
| **Categoria** | **Scelte per la categoria** |
| Lunghezza [CNL] | 1. Lunghezza > 30 or Lunghezza = 0 = false[error] 2. Lunghezza <= 30 and Lunghezza > 0 = true [property CNL\_OK] |
| **Parametro** | Scadenza |
| Formato scadenza:  "^(0[1-9]|1[0-2])\/?([0-9]{2})$" | |
| **Categoria** | **Scelte per la categoria** |

|  |  |
| --- | --- |
| Formato [DF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [property DF\_OK] |
| **Parametro** | CVV |
| Formato cvv:  “^[0-9]{3, 4}$” | |
| **Categoria** | **Scelte per la categoria** |
| Formato [CVVF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [property CVVF\_OK] |
| **Parametro** | Importo |
| Formato importo:  “^[0-9]+$” | |
| **Categoria** | **Scelta per la categoria** |
| Formato [IF] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [proporti IF\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_GC\_1.0 | AL1 | Errato: lunghezza intestatario non valida |
| TC\_GC\_1.1 | AL2, CNL1 | Errato: lunghezza intestatario valida ma lunghezza numero di carta non valida |
| TC\_GC\_1.2 | AL2, CNL2, DF1 | Errato: lunghezza intestatario valida, lunghezza numero di carta valida, formato scadenza non valido |
| TC\_GC\_1.3 | AL2, CNL2, DF2, CVVF1 | Errato: lunghezza intestatario valida, lunghezza numero di carta valida, formato scadenza valido, formato cvv non valido |
| TC\_GC\_1.4 | AL2, CNL2, DF2, CVVF2, IF1 | Errato: lunghezza intestatario valida, lunghezza numero di carta valida, formato scadenza valido, formato cvv non valido, formato importo nom valido |
| TC\_GC\_1.5 | AL2, CNL2, DF2, CVV2, IF2 | Corretto: contributo versato |

* 1. Unit test cases

|  |  |
| --- | --- |
| **AddRoute** | |
| **Parametro** | Metodo |
| **Categoria** | **Scelte per la categoria** |
| Tipo[TP] | 1. Metodo = null [error] 2. Metodo = GET or Metodo = POST [property TP\_OK] |
| **Parametro** | Url |
| **Categoria** | **Scelte per la categoria** |
| Match[MT] | 1. Url = null or Url = not /something [error] 2. Url = /something [MT\_OK] |
| **Parametro** | Action |
| **Categoria** | **Scelte per la categoria** |
| isPresent[IP] | 1. Action = null [error] 2. Action = not null [IP\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_U\_AR\_1.0 | TP1 | Errato: il metodo della request deve essere GET o POST |
| TC \_U\_AR\_2.0 | TP2, MT1 | Errato: il metodo della request è valido, l’url non è valido |
| TC \_U\_AR\_3.0 | TP2, MT2, IP1 | Errato: il metodo della request è valido, l’url corrisponde è del formato corretto, non è stata inserita una Action |
| TC \_U\_AR\_4.0 | TP2, MT2, IP2 | Corretto: percorso aggiunto al router |

|  |  |
| --- | --- |
| **HandleRoute** | |
| **Parametro** | Metodo |
| **Categoria** | **Scelte per la categoria** |
| Tipo[TP] | 1. Metodo = null [error] 2. Metodo = GET or Metodo = POST[property TP\_OK] |
| **Parametro** | Url |
| **Categoria** | **Scelte per la categoria** |
| Match[MT] | 1. Url = null or Url= not /something [error] 2. Url= /something [MT\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_U\_HR\_1.0 | TP1 | Errato: il metodo della request deve essere GET o POST |
| TC \_U\_HR\_1.1 | TP2, MT1 | Errato: il metodo della request è valido, l’url non è valido |
| TC \_U\_HR\_1.2 | TP2, MT2 | Corretto: restituisce Action Strategy |

|  |  |
| --- | --- |
| **Get** | |
| **Parametro** | Url |
| **Categoria** | **Scelte per la categoria** |
| Match[MT] | 1. Url = null or Url = not /something [error] 2. Url = /something [MT\_OK] |
| **Parametro** | Action |
| **Categoria** | **Scelte per la categoria** |
| isPresent[IP] | 1. Action = null [error] 2. Action = not null [IP\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_U\_G\_1.0 | MT1 | Errato: l’url non è valido |
| TC \_U\_G\_1.1 | MT2, IP1 | Errato: l’url è valido ma la Action non è valida |
| TC \_U\_G \_1.2 | MT2, IP2 | Corretto: metodo get andato a buon fine. |

|  |  |
| --- | --- |
| **View** | |
| **Parametro** | viewName |
| **Categoria** | **Scelte per la categoria** |
| isPresent[IP] | 1. viewName = null [error] 2. viewName = not null [IP\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_U\_V\_1.0 | IP1 | Errato: la viewName non è valida. |
| TC \_U\_V\_1.1 | IP2 | Corretto: il metodo View fa visualizzare la viewName.jsp |

|  |  |
| --- | --- |
| **Redirect** | |
| **Parametro** | Url |
| **Categoria** | **Scelte per la categoria** |
| isPresent[IP] | 1. Url = null [error] 2. Url = not null [IP\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_U\_R\_1.0 | IP1 | Errato: l’url non è valido. |
| TC \_U\_R\_1.1 | IP2 | Corretto: il metodo Redirect manda il controllo all’url specificato |

* 1. Integration test cases
     1. Router-Get

|  |
| --- |
| **Get** |

|  |
| --- |
| Il test di integrazione di Get generebbe gli stessi Test case del test di unità di Get per tanto abbiamo ritenuto opportuno non ripetere la stesura delle tabelle, definendo esclusivamente gli id dei test case.  La differenza tra il test di unità ed il test di integrazione del metodo get sta nell’appropriato utilizzo di Mokito |

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Frame** | **Esito** |
| TC\_I\_G\_1.0 | MT1 | Errato: l’url non è valido |
| TC \_I\_G\_1.1 | MT2, IP1 | Errato: l’url è valido ma la Action non è valida |
| TC \_I\_G \_1.2 | MT2, IP2 | Corretto: metodo get andato a buon fine. |

1. Testing schedule

Le attività di pianificazione dei test e di generazione dei test case saranno effettuate dopo la fase di System Design e prima della fase di Object Design.

L’implementazione e la conseguente esecuzione dei casi di test avverranno successivamente alla fase di implementazione.