



TP - Test Plan

| | |
|----------------------|--|
| Riferimento | <u>2024_C10_TP_beehAlve_V2.0</u> <u>2024_C10_RAD_beehAlve_V2.0</u> <u>2024_C10_SDD_beehAlve_V2.0</u> |
| Versione | 2.0 |
| Data | 22/01/2024 |
| Destinatario | Prof.ssa Filomena Ferrucci |
| Presentato da | N. Gallotta, F. Festa, S. Valente, A. De Pasquale, L. Milione, C. Boninfante |
| Approvato da | Nicolò Delogu, Dario Mazza |



| Composizione Gruppo | |
|---------------------|-------------|
| Francesco Festa | 05121-13547 |
| Nicolò Gallotta | 05121-14639 |
| Andrea De Pasquale | 05121-14909 |
| Sara Valente | 05121-14627 |
| Lorenzo Milione | 05121-14107 |
| Carmine Boninfante | 05121-13309 |



Cronologia revisioni

| Data | Versione | Descrizione | Autori |
|------------|----------|----------------------------------|------------------------------|
| 04/12/2023 | 0.1 | Creazione Documento | Tutto il Team |
| 05/12/2023 | 0.2 | Scrittura primi 8 capitoli | Nicolò Delogu Dario Mazza |
| 05/12/2023 | 0.3 | Inserimento Test cases | Tutto il Team |
| 05/12/2023 | 0.4 | Integrazione e Revisione | Tutto il Team |
| 12/12/2023 | 1.0 | Revisione Finale | Nicolò Delogu Dario Mazza |
| 21/01/2024 | 1.1 | Revisione Test Cases e strumenti | Tutto il Team |
| 22/01/2024 | 2.0 | Revisione e sistemazione finale | Tutto il Team |



Sommario

| | |
|---|-----------|
| Cronologia revisioni..... | 3 |
| Sommario..... | 4 |
| 1. Introduzione..... | 5 |
| 2. Documenti Correlati..... | 6 |
| 3. Panoramica del sistema..... | 7 |
| 4. Funzionalità da testare e da non testare..... | 8 |
| 5. Criteri pass/failed..... | 9 |
| 6. Approccio..... | 10 |
| 6.1 Testing di unità..... | 10 |
| 6.2 Testing di integrazione..... | 10 |
| 6.3 Testing di sistema..... | 10 |
| 6.3.1 Functional Testing..... | 11 |
| 6.3.2 Acceptance Testing..... | 11 |
| 6.3.3 Performance Testing, Pilot Testing, Installation Testing..... | 11 |
| 7. Sospensione e ripresa..... | 12 |
| 7.1 Criteri di sospensione..... | 12 |
| 7.2 Criteri di ripresa..... | 12 |
| 8. Materiale per il testing..... | 13 |
| 9. Test case..... | 14 |
| 9.1 Gestione Profilo..... | 14 |
| 9.1.1 Registrazione..... | 14 |
| 9.1.2 Login..... | 19 |
| 9.1.3 Modifica Dati Personali..... | 20 |
| 9.1.4 Modifica Password..... | 23 |
| 9.2 Gestione Interventi..... | 25 |
| 9.2.1 Pianificazione Intervento..... | 25 |
| 9.2.2 Modifica Intervento..... | 27 |
| 9.3 Gestione Dashboard..... | 29 |
| 9.3.1 Creazione Arnia..... | 29 |
| 9.3.2 Modifica Arnia..... | 31 |
| 10. Testing schedule..... | 33 |
| 11. Glossario..... | 34 |



1. Introduzione

BeehAlve è un'applicazione web che ha come obiettivo principale quello di supportare il lavoro degli apicoltori, offrendo loro funzionalità di gestione delle arnie e di previsione di anomalie attraverso dei sensori di IoT che monitorano costantemente lo stato di salute delle arnie.

Questo documento ha l'obiettivo di descrivere e analizzare le attività di testing per il sistema BeehAlve al fine di individuare errori così da garantire un'elevata affidabilità e ridurre la probabilità che possano verificarsi malfunzionamenti durante l'utilizzo.

Sono state pianificate attività di testing per i seguenti sottosistemi:

- *Gestione Profilo;*
- *Gestione Interventi;*
- *Gestione Dashboard.*



2. Documenti Correlati

I test case sono basati sulle funzionalità del sistema individuate nei documenti precedentemente prodotti.

- **Relazioni con il Requirements Analysis Document (RAD):** I Test Case sono stati pianificati in relazione ai requisiti funzionali e non presenti nel RAD
- **Relazioni con il System Design Document (SDD):** I Test Case sono stati pianificati rispettando la suddivisione in sottosistemi presentata nel SDD
- **Relazioni con l'Object Design Document (ODD):** Per ciò che riguarda il test di unità e di integrazione, legati maggiormente all'ODD e alla divisione in package del sistema, essi saranno scritti e documentati all'interno del codice dell'applicativo. Per tale motivo, nel presente documento, non vi saranno riferimenti al loro design.



3. Panoramica del sistema

Il sistema utilizza un'architettura Three-Tier. Per il backend verrà utilizzato il framework Java Spring, mentre per il frontend verranno utilizzati HTML5, CSS3, Bootstrap, Javascript e Ajax. In particolare il modello di Machine Learning verrà integrato rendendolo un web service con Flask. Per il database verrà invece utilizzato MySQL connesso attraverso l'utilizzo di Spring JPA.



4. Funzionalità da testare e da non testare

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

- Gestione Profilo
 - Registrazione
 - Login
 - Modifica Dati Personali
 - Modifica Password
- Gestione Interventi
 - Pianificazione Intervento
 - Modifica Intervento
- Gestione Dashboard
 - Creazione Arnia
 - Modifica Arnia

Le funzionalità di cui non si andrà ad effettuare il testing riguardano requisiti funzionali di bassa priorità o funzionalità che non prevedono input da parte dell'apicoltore, come attività che riguardano la visualizzazione di informazioni.



5. Criteri pass/failed

Le attività di testing sono mirate ad identificare la presenza di errori all'interno del sistema, per effettuare poi un successivo intervento di eliminazione. L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti.

Un test fallisce (fail) se, dato un input al sistema, l'output ottenuto è diverso dall'output atteso dall'oracolo.

Un test ha successo (pass) se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.

Tutto il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

- Testare tutti i requisiti funzionali ad alta priorità;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;



6. Approccio

Il testing verrà suddiviso in tre fasi:

- Testing di unità: ha l'obiettivo di verificare il comportamento delle singole unità del sistema, dove per unità intendiamo una singola classe o metodo.
- Testing di integrazione: viene eseguito per la verifica del comportamento delle componenti quando vengono integrate
- Testing di sistema: sarà eseguito per la verifica del comportamento del sistema nella sua interezza

6.1 Testing di unità

Per il testing di unità la strategia comprende la verifica dei principali metodi delle classi del sistema, escluse le Entity, in quanto presentano soltanto getters e setters. I casi di test saranno definiti tramite tecniche black-box.

Per l'implementazione dei casi test, verrà utilizzato il framework JUnit. Per ogni classe da testare verrà prodotta una corrispondente classe di testing, che dovranno essere scritte prima della produzione della classe stessa per eliminare eventuali Bias dovuti alla conoscenza del codice.

Verrà inoltre utilizzato Mockito per la costruzione degli stub e l'isolamento della componente testata dalle sue dipendenze.

6.2 Testing di integrazione

Per il testing di integrazione verrà utilizzato un approccio bottom-up. I tool che verranno utilizzati per questa fase sono gli stessi utilizzati nel testing di unità: la definizione dei test case avverrà attraverso JUnit e verrà utilizzato Mockito per il Mocking. Per realizzare la Continuous Integration verrà invece utilizzato GitHub Actions. Per ogni componente da testare verranno testate prima le classi Service e DAO e successivamente le classi Controller.

6.3 Testing di sistema

Una volta completati con successo i test di unità e integrazione, si procederà con il testing di sistema, il cui obiettivo è garantire che il prodotto finale soddisfi gli standard qualitativi stabiliti. Questa fase comprenderà la verifica delle specifiche



funzionali e non funzionali dell'intero sistema, concentrandosi sui comportamenti, la progettazione e le aspettative del cliente.

Per il testing di sistema, è stato scelto di utilizzare il tool Selenium IDE, che permette di registrare le azioni che un apicoltore può intraprendere sul browser, così da poter implementare ed eseguire i test case di sistema. Il server invece, verrà deployato in localhost.

6.3.1 Functional Testing

Nel Functional Testing verranno validati i requisiti funzionali attraverso un approccio Black-Box, quindi prendendo in esame solo l'input fornito e l'analisi dell'output. Questo test verrà condotto per individuare possibili faults generati dagli input degli apicoltori. Per rendere il test più efficiente verrà adottata la tecnica del Category Partition.

6.3.2 Acceptance Testing

L'Acceptance Testing sarà effettuato sul functional testing simulando l'utilizzo della piattaforma. I Project Manager simuleranno la figura del cliente.

6.3.3 Performance Testing, Pilot Testing, Installation Testing

Non si assicura l'esecuzione a causa del basso budget a disposizione.



7. Sospensione e ripresa

In questa sezione verranno specificati i criteri di sospensione dei test e le attività di test che dovranno essere ripetute quando si riprende il test

7.1 Criteri di sospensione

Il testing non verrà mai sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere sospeso solo nel caso in cui, al momento dell'esecuzione, venga restituito un errore nella definizione di uno dei test stessi.

7.2 Criteri di ripresa

La fase di testing verrà ripresa quando tutti gli errori individuati precedentemente verranno risolti.



8. Materiale per il testing

L'hardware richiesto per l'esecuzione dell'attività di test consiste in un computer di base, non necessariamente connesso a Internet, poiché il sistema non è ancora stato rilasciato. I sensori di IoT non verranno testati perché simulati in fase di sviluppo.

Durante il processo di test, verranno impiegati diversi strumenti, tra cui:

- Un server operante sul sistema;
- Client-web per inviare richieste al server;
- Un Database Management System (DBMS) responsabile della gestione dell'utilizzo del database.



9. Test case

9.1 Gestione Profilo

9.1.1 Registrazione

| Parametro: Email | |
|---|--|
| Formato: $^{\wedge}[a-zA-Z0-9_ \% + -] + @ [a-zA-Z0-9 _ + \.] [a-zA-Z] \{ 2, \} \$$ | |
| Nome Categoria | Scelte per la categoria |
| Formato[FE] | <ol style="list-style-type: none">1. Non rispetta il formato [Error]2. Rispetta il formato [Property formatoFEok] |
| Esiste[EE] | <ol style="list-style-type: none">1. L'email esiste già all'interno del database [if formatoFEok] [Error]2. L'email non esiste nel database [if formatoFEok] [Property esisteEEok] |

| Parametro: Passwordhash | |
|---|---|
| Formato: $^{\wedge} (? = . * [a - z]) (? = . * [A - Z]) (? = . * \backslash d) (? = . * [@ . \$! \% * ? \& _ -]) [A - Z a - z \backslash d @ . \$! \% * ? \& _ -] + \$$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LP] | <ol style="list-style-type: none">1. Lunghezza < 8 [Error]2. Lunghezza >= 8 [Property lunghezzaLPok] |
| Formato[FP] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLPok] [Error]2. Rispetta il formato [if lunghezzaLPok] [Property formatoFPok] |



| Parametro: Confirm_password | |
|--|--|
| Formato: $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@.$!%*?&_-])[A-Za-z\d@.$!%*?&_-]+$$ | |
| Nome Categoria | Scelte per la categoria |
| Corrisponde[CP] | <ol style="list-style-type: none">1. La conferma della password non corrisponde con la password [Error]2. La conferma della password corrisponde alla password [Property corrispondenzaCPok] |

| Parametro: First_name | |
|---|--|
| Formato: $^[A-Z][a-z'-]*?(?:[A-Z][a-z'-]+)*$$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LFN] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 15 [Error]3. Lunghezza >= 2 AND Lunghezza <= 15 [Property lunghezzaLFNok] |
| Formato[FFN] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLFNok] [Error]2. Rispetta il formato [if lunghezzaLFNok] [Property formatoFFNok] |



| Parametro: Last_name | |
|--|--|
| Formato: $^{[A-Z][a-z'-]*}(: [A-Z][a-z'-]+)^*\$$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LLN] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 15 [Error]3. Lunghezza >= 2 AND Lunghezza <= 15 [Property lunghezzaLLNok] |
| Formato[FLN] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLLNok] [Error]2. Rispetta il formato [if lunghezzaLLNok] [Property formatoFLNok] |

| Parametro: Company_name | |
|--|---|
| Formato: $^{[a-zA-Z0-9\s.'",&()-]*}\$$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LCN] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 100 [Error]3. Lunghezza >= 2 AND Lunghezza <= 100 [Property lunghezzaLCNok] |
| Formato [FCN] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLCNok] [Error]2. Rispetta il formato [if lunghezzaLCCok] [Property formatoFCNok] |



| Parametro: Company_PIVA | |
|-------------------------|--|
| Formato: ^[\d-]+\$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LCP] | 1. Lunghezza < 9 [Error] 2. Lunghezza > 20 [Error] 3. Lunghezza >= 9 AND Lunghezza <= 20 [Property lunghezzaLCPok] |
| Formato [FCP] | 1. Non rispetta il formato [if lunghezzaLCPok] [Error] 2. Rispetta il formato [if lunghezzaLCPok] [Property formatoFCPok] |

| Test Case ID | Test Frame | Esito |
|--------------|-------------------------------|---|
| TC_GP_1.1 | FE1 | Errato: formato email non valido. |
| TC_GP_1.2 | FE2, EE1 | Errato: email già esistente. |
| TC_GP_1.3 | FE2. EE2, LP1 | Errato: Lunghezza password insufficiente. |
| TC_GP_1.4 | FE2. EE2, LP2, FP1 | Errato: formato password non valido. |
| TC_GP_1.5 | FE2, EE2, LP2, FP2, CP1 | Errato: la conferma della password non corrisponde. |
| TC_GP_1.6 | FE2, EE2, LP2, FP2, CP2, LFN1 | Errato: lunghezza first_name insufficiente. |
| TC_GP_1.7 | FE2, EE2, LP2, FP2, CP2, LFN2 | Errato: lunghezza first_name troppo grande. |



| | | |
|------------|--|---|
| TC_GP_1.8 | FE2, EE2, LP2, FP2, CP2, LN3, FFN1 | Errato: formato first_name non valido. |
| TC_GP_1.9 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN1 | Errato: lunghezza last_name insufficiente. |
| TC_GP_1.10 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN2 | Errato: lunghezza last_name troppo grande. |
| TC_GP_1.11 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN1 | Errato: formato last_name non valido. |
| TC_GP_1.12 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN1 | Errato: lunghezza company_name insufficiente. |
| TC_GP_1.13 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN2 | Errato: lunghezza company_name troppo grande. |
| TC_GP_1.14 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN3, FCN1 | Errato: formato company_name non valido. |
| TC_GP_1.15 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN3, FCN2, LCP1 | Errato: lunghezza PIVA insufficiente. |
| TC_GP_1.16 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN3, FCN2, LCP2 | Errato: lunghezza PIVA troppo grande. |
| TC_GP_1.17 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, LN, LCN3, FCN2, LCP3, FCP1 | Errato: formato PIVA non valido. |
| TC_GP_1.18 | FE2, EE2, LP2, FP2, CP2, LN3, FFN2, LLN3, FLN2, LCN3, FCN2, LCP3, FCP2 | Registrazione effettuata con successo. |



9.1.2 Login

| Parametro: Email | |
|---|---|
| Formato: <code>^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Esiste[EE] | <ol style="list-style-type: none"> 1. L'email esiste già all'interno del database [if formatoEEok] [Error] 2. L'email non esiste nel database [if formatoEEok] [Property esisteEEok] |

| Parametro: PasswordHash | |
|--|--|
| Formato: <code>^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@.\$!%*?&_-])[A-Za-z\d@.\$!%*?&_-]+\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Corrisponde[CPH] | <ol style="list-style-type: none"> 1. La password non corrisponde alla email inserita [if esisteEEok] [Error] 2. La password corrisponde all'e-mail inserita [if esisteEEok] [Property corrispondenzaCPHok] |

| Test Case ID | Test Frame | Esito |
|--------------|------------------|---|
| TC_GP_2.1 | EE1 | Errato: l'email non esiste nel database. |
| TC_GP_2.2 | EE2, CPH1 | Errato: la password non corrisponde all'email inserita. |
| TC_GP_2.3 | EE2, CPH2 | Login effettuato con successo. |



9.1.3 Modifica Dati Personali

| Parametro: Company_name | |
|------------------------------------|--|
| Formato: ^[a-zA-Z0-9\s.'",&()-]+\$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LCN] | 1. Lunghezza < 2 [Error] 2. Lunghezza > 100 [Error] 3. Lunghezza >= 2 AND Lunghezza <= 100 [Property lunghezzaLCNok] |
| Formato [FCN] | 3. Non rispetta il formato [if lunghezzaLCNok] [Error] 4. Rispetta il formato [if lunghezzaLCNok] [Property formatoFCNok] |

| Parametro: First_name | |
|--|--|
| Formato: ^[A-Z][a-z'-]*(:[A-Z][a-z'-]+)*\$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LFN] | 1. Lunghezza < 2 [Error] 2. Lunghezza > 15 [Error] 3. Lunghezza >= 2 and Lunghezza <= 15 [Property lunghezzaLFNok] |
| Formato[FFN] | 1. Non rispetta il formato [if lunghezzaLFNok] [Error] 2. Rispetta il formato [if lunghezzaLFNok] [Property formatoFFNok] |



| Parametro: Last_name | |
|--|--|
| Formato: ^[A-Z][a-z'-]*(?: [A-Z][a-z'-]+)*\$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LLS] | 1. Lunghezza < 2 [Error] 2. Lunghezza > 15 [Error] 3. Lunghezza >= 2 and Lunghezza <= 15 [Property lunghezzaLLSok] |
| Formato[FLS] | 1. Non rispetta il formato [if lunghezzaLLSok] [Error] 2. Rispetta il formato [if lunghezzaLLSok] [Property formatoFLSok] |

| Test Case ID | Test Frame | Esito |
|--------------|------------------------|---|
| TC_GP_3.1 | LCN1 | Errato: lunghezza company_name insufficiente. |
| TC_GP_3.2 | LCN2 | Errato: lunghezza company_name troppo grande. |
| TC_GP_3.3 | LCN3, FCN1 | Errato: formato company_name non valido. |
| TC_GP_3.4 | LCN3, FCN2, LFN1 | Errato: lunghezza first_name insufficiente. |
| TC_GP_3.5 | LCN3, FCN2, LFN2 | Errato: lunghezza first_name troppo grande. |
| TC_GP_3.6 | LCN3, FCN2, LFN3, FFN1 | Errato: formato first_name non valido. |



| | | |
|------------|--|--|
| TC_GP_3.7 | LCN3, FCN2, LFN3, FFN2, LLN1 | Errato: lunghezza last_name insufficiente. |
| TC_GP_3.8 | LCN3, FCN2, LFN3, FFN2, LLN2 | Errato: lunghezza last_name troppo grande. |
| TC_GP_3.9 | LCN3, FCN2, LFN3, FFN2, LLN3, FLN1 | Errato: formato surname non valido. |
| TC_GP_3.10 | LCN3, FCN2, LFN3, FFN3, LLN3, FLN2 | Dati modificati correttamente. |



9.1.4 Modifica Password

| Parametro: Current_password | |
|--|---|
| Formato: $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@.$!%*?&_-])[A-Za-z\d@.$!%*?&_-]+\$$ | |
| Nome Categoria | Scelte per la categoria |
| Corrisponde[CCP] | <ol style="list-style-type: none">1. Non corrisponde alla email associata al profilo [Error]2. Corrisponde alla email associata al profilo [Property corrispondenzaCCPok] |

| Parametro: New_password | |
|--|--|
| Formato: $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@.$!%*?&_-])[A-Za-z\d@.$!%*?&_-]+\$$ | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LNP] | <ol style="list-style-type: none">1. Lunghezza < 12 [Error]2. Lunghezza >= 12 [Property lunghezzaLNPok] |
| Formato[FNP] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLNPok] [Error]2. Rispetta il formato [if lunghezzaLNPok] [Property formatoFNPok] |

| Parametro: Confirm_new_password | |
|--|---|
| Formato: $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@.$!%*?&_-])[A-Za-z\d@.$!%*?&_-]+\$$ | |
| Nome Categoria | Scelte per la categoria |
| Corrisponde[CNP] | <ol style="list-style-type: none">1. La conferma della password non corrisponde con la password [Error]2. La conferma della password corrisponde alla password [Property corrispondenzaCNPok] |



| Test Case ID | Test Frame | Esito |
|--------------|---------------------------|--|
| TC_GP_4.1 | CCP1 | Errato: password errata. |
| TC_GP_4.2 | CCP2, LNP1 | Errato: lunghezza new_password insufficiente. |
| TC_GP_4.3 | CCP2, LNP2, FNP1 | Errato: formato new_password non valido. |
| TC_GP_4.4 | CCP2, LNP2, FNP2, CNP1 | Errore: la conferma password non corrisponde con la password |
| TC_GP_4.5 | CCP2, LNP2, FNP2, CNP2 | Password modificata con successo. |



9.2 Gestione Interventi

9.2.1 Pianificazione Intervento

| Parametro: Operation_name | |
|--|--|
| Formato: <code>^[a-zA-Z0-9\s\-_()'""]+\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LON] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 70 [Error]3. Lunghezza >= 2 AND Lunghezza <= 70 [Property lunghezzaLONok] |
| Formato[FON] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLONok] [Error]2. Rispetta il formato [if lunghezzaLONok] [Property lunghezzaFONok] |

| Parametro: Notes | |
|---|--|
| Formato: <code>^[a-zA-Z0-9\s\-_()'"",?!.:]*\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LN] | <ol style="list-style-type: none">1. Lunghezza > 300 [Error]2. Lunghezza <= 300 [Property lunghezzaLNok] |

| Parametro: Operation_date | |
|---|--|
| Formato: <code>[yyyy/MM/dd HH:mm:ss]</code> | |
| Nome Categoria | Scelte per la categoria |
| Correttezza[COD] | <ol style="list-style-type: none">1. Date < Current date [Error]2. Date > Current date [Property correttezzaCODok] |



| Parametro: Operation_type | |
|---|--|
| Formato: [Medical Inspection Medical Treatment Check population Extraction Vet Visit Feeding Transfer Maintenance] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FOT] | 1. Non rispetta il formato [Error] 2. Rispetta il formato [Property formatoFOTok] |

| Test Case ID | Test Frame | Esito |
|--------------|-----------------------------|---|
| TC_GI_1.1 | LON1 | Errato: lunghezza operation_name insufficiente. |
| TC_GI_1.2 | LON2 | Errato: lunghezza operation_name troppo grande. |
| TC_GI_1.3 | LON3, FON1 | Errato: formato operation_name non valido. |
| TC_GI_1.4 | LON3, FON2, LN1 | Errato: lunghezza notes troppo grande. |
| TC_GI_1.5 | LON3, FON2, LN2, COD1 | Errato: operation_date è precedente alla data corrente. |
| TC_GI_1.6 | LON3, FON2, LN2, COD2, FOT1 | Errato: formato operation_type non valido. |
| TC_GI_1.7 | LON3, FON2, LN2, COD2, FOT2 | Pianificazione intervento avvenuta correttamente. |



9.2.2 Modifica Intervento

| Parametro: Operation_name | |
|--|--|
| Formato: <code>^[a-zA-Z0-9\s\-_()'"]+\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LON] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 70 [Error]3. Lunghezza >= 2 AND Lunghezza <= 70 [Property lunghezzaLONok] |
| Formato[FON] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLONok] [Error]2. Rispetta il formato [if lunghezzaLONok] [Property formatoFONok] |

| Parametro: Notes | |
|--|--|
| Formato: <code>^[a-zA-Z0-9\s\-_()'".,?!:]*\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LN] | <ol style="list-style-type: none">1. Lunghezza > 300 [Error]2. Lunghezza <= 300 [Property lunghezzaLNok] |

| Parametro: Operation_date | |
|---|--|
| Formato: <code>[yyyy/MM/dd HH:mm:ss]</code> | |
| Nome Categoria | Scelte per la categoria |
| Correttezza[COD] | <ol style="list-style-type: none">1. Date < Current date [Error]2. Date > Current date [Property correttezzaCODok] |



| Parametro: Operation_type | |
|--|--|
| Formato: [Medical Inspection Medical Treatment Check population Extraction Vet Visit Feeding Transfer Maintenance] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FOT] | 1. Non rispetta il formato [Error] 2. Rispetta il formato [Property formatoFOTok] |

| Test Case ID | Test Frame | Esito |
|--------------|-----------------------------|--|
| TC_GI_2.1 | LON1 | Errato: lunghezza operation_name insufficiente. |
| TC_GI_2.2 | LON2 | Errato: lunghezza operation_name troppo grande. |
| TC_GI_2.3 | LON3, FON1 | Errato: formato operation_name non valido. |
| TC_GI_2.4 | LON3, FON2, LN1 | Errato: lunghezza notes troppo grande. |
| TC_GI_2.5 | LON3, FON2, LN2, COD1 | Errato: operation_data inserita minore della data corrente |
| TC_GI_2.6 | LON3, FON2, LN2, COD2, FOT1 | Errato: formato operation_type non valido. |
| TC_GI_2.7 | LON3, FON2, LN2, COD2, FOT2 | Pianificazione intervento modificata correttamente. |



9.3 Gestione Dashboard

9.3.1 Creazione Arnia

| Parametro: Nickname | |
|--|--|
| Formato: <code>^[a-zA-Z0-9\s\-_()''"]+\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LNI] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 50 [Error]3. Lunghezza >= 2 AND Lunghezza <= 50 [Property lunghezzaLNlok] |
| Formato[FNI] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLNlok] [Error]2. Rispetta il formato [if lunghezzaLNlok] [Property formatoFNlok] |

| Parametro: Hive_type | |
|---|--|
| Formato: [Langstroth Warre Top-Bar Horizontal Vertical Bee Skep WBC Dadant] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FHT] | <ol style="list-style-type: none">1. Non rispetta il formato [Error]2. Rispetta il formato [Property formatoFHTok] |

| Parametro: Bee_species | |
|--|--|
| Formato: [Apis mellifera Apis cerana Apis dorsata Apis florea] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FBS] | <ol style="list-style-type: none">1. Non rispetta il formato [Error]2. Rispetta il formato [Property formatoFBSok] |



| Test Case ID | Test Frame | Esito |
|--------------|---------------------------|---|
| TC_GD_1.1 | LNI1 | Errato: lunghezza nickname insufficiente. |
| TC_GD_1.2 | LNI2 | Errato: lunghezza nickname troppo grande. |
| TC_GD_1.3 | LNI3, FNI1 | Errato: formato nickname non valido. |
| TC_GD_1.4 | LNI3, FNI2, FHT1 | Errato: formato hive_type non valido. |
| TC_GD_1.5 | LNI3, FNI2, FHT2, FBS1 | Errato: formato Bee_species non valido. |
| TC_GD_1.6 | LNI3, FNI2, FHT2, FBS2 | Creazione arnia avvenuta correttamente. |



9.3.2 Modifica Arnia

| Parametro: Nickname | |
|---|--|
| Formato: <code>^[a-zA-Z0-9\s\-'"]+\$</code> | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza[LNI] | <ol style="list-style-type: none">1. Lunghezza < 2 [Error]2. Lunghezza > 50 [Error]3. Lunghezza >= 2 AND Lunghezza <= 50 [Property lunghezzaLNlok] |
| Formato[FNI] | <ol style="list-style-type: none">1. Non rispetta il formato [if lunghezzaLNlok] [Error]2. Rispetta il formato [if lunghezzaLNlok] [Property formatoFNlok] |

| Parametro: Hive_type | |
|---|--|
| Formato: [Langstroth Warre Top-Bar Horizontal Vertical Bee Skep WBC Dadant] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FHT] | <ol style="list-style-type: none">1. Non rispetta il formato [Error]2. Rispetta il formato [Property formatoFHTok] |

| Parametro: Bee_species | |
|--|--|
| Formato: [Apis mellifera Apis cerana Apis dorsata Apis florea] | |
| Nome Categoria | Scelte per la categoria |
| Formato[FBS] | <ol style="list-style-type: none">3. Non rispetta il formato [Error]4. Rispetta il formato [Property formatoFBSok] |



| Test Case ID | Test Frame | Esito |
|--------------|---------------------------|---|
| TC_GD_2.1 | LNI1 | Errato: lunghezza nickname insufficiente. |
| TC_GD_2.2 | LNI2 | Errato: lunghezza nickname troppo grande. |
| TC_GD_2.3 | LNI3, FNI1 | Errato: formato nickname non valido. |
| TC_GD_2.4 | LNI3, FNI2, FHT1 | Errato: formato hive:type non valido. |
| TC_GD_2.5 | LNI3, FNI2, FHT2, FBS1 | Errato: formato Bee_species non valido. |
| TC_GD_2.6 | L31, FNI2, FHT2, FBS2 | Modifica arnia avvenuta con successo. |



10. Testing schedule

Le attività di testing avverranno come definito precedentemente, ovvero immediatamente dopo la fase di design necessaria per la pianificazione. La scrittura dei casi di test avverrà contemporaneamente con lo sviluppo del codice. L'esecuzione dei test avverrà sia in contemporanea che successivamente all'implementazione del sistema. Al termine dello sviluppo, tutti i test saranno eseguiti nuovamente per garantire il corretto funzionamento. Per altre informazioni si rimanda ai documenti di management sulle schedule.



11. Glossario

FE: Formato Email
EE: Esiste Email
LP: Lunghezza PasswordHash
FP: Formato PasswordHash
CP: Corrispondenza passwords
LFN: Lunghezza First_name
FFN: Formato First_name
LLN: Lunghezza Last_name
FLN: Formato Last_name
LCN: Lunghezza Company_name
FCN: Formato Company_name
LCP: Lunghezza Company_PIVA
FCP: Formato Company_PIVA
CPH: Corrisipondenza PasswordHash
CCP: Corrispondenza Current_password
LNP: Lunghezza New_password
FNP: Formato New_password
CNP: Corrispondenza New_passwords
LON: Lunghezza Operation_name
FON: Formato Operation_name
LN: Lunghezza notes
COD: Correttezza Operation_name
FOT: Formato Operation_type
LNI: Lunghezza nickname arnia
FNI: Formato nickname arnia
FHT: Formato Hive_type
FBS: Formato Bee_species