



Laurea Magistrale in informatica-Università di Salerno
Corso di Ingegneria del Software - Prof.ssa F. Ferrucci, Prof. F. Palomba



Test Plan

Environmental Intelligence for Agriculture

Riferimento	
Versione	1.0
Data	10/12/2022
Destinatario	Prof.ssa Filomena Ferrucci Prof. Fabio Palomba
Presentato da	C04
Approvato da	Carmine Laudato, Pierluigi Lambiase



Laurea Magistrale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

Revision History

Data	Versione	Descrizione	Autori
07/12/2022	0.1	Aggiunta Introduzione, Panoramica del sistema	Carmine Laudato Pierluigi Lambiase
08/12/2022	0.3	Aggiunta Funzionalità da testare e Funzionalità da non testare	Carmine Laudato Pierluigi Lambiase
09/12/2022	0.5	Aggiunta Approccio, Criteri Pass/Fail, Criteri di Sospensione e Ripresa, Materiale per il testing	Carmine Laudato Pierluigi Lambiase
10/12/2022	0.9	Aggiunta Test Cases, Testing Schedule	Maria Lombardi Benedetto Scala Francesco Maria Puca Gerardo Frino Francesco Fattorusso
10/12/2022	1.0	Revisione Test Plan	Maria Lombardi Gerardo Frino Francesco Maria Puca
30/01/2022	1.1	Eliminato test case sugli alert, aggiunto test case sulla decision intelligence, aggiunto test case sulla visualizzazione meteo	Benedetto Scala
07/02/2023	1.2	Revisione e correzione del documento	Francesco Maria Puca Maria Lombardi



Laurea Magistrale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

Sommario

Revision History	2
1. Introduzione	4
1.1 Definizioni, Acronimi e Abbreviazioni	4
1.2 Riferimenti ad altri documenti	5
2. Panoramica del sistema.....	5
3. Funzionalità da testare	6
4. Funzionalità da non testare	6
5. Approccio	7
6. Criteria Pass/Fail	8
7. Criteri di Sospensione e Ripresa	8
8. Materiale per il testing	9
9. Test cases.....	9
9.1 Gestione Utente	9
9.1.1 Registrazione Utente tramite codice di accesso	9
9.2 Gestione Ambiente Agricolo.....	11
9.2.1 Inserimento di un terreno	11
9.3 Gestione Meteo	13
9.3.1 Visualizzazione Meteo.....	13
9.4 Gestione Azienda Agricola	14
9.4.1 Aggiunta di Utente e Assegnazione Dei Ruoli	14
9.5 Gestione degli insight dell'AI	14
9.5.1 Decision Intelligence	14
10. Testing schedule	15



1. Introduzione

In questo documento saranno scelte le linee guida e le attività di testing attuate per la piattaforma EnIA. Inoltre, saranno identificate le principali funzionalità che dovranno essere testate e quelle che non verranno testate. Lo scopo principale del testing è quello di individuare e prevedere quanti più fault possibili esistono in una funzionalità e di conseguenza nel sistema. Infatti, potremmo dire che le attività di testing hanno avuto successo se identificheremo il maggior numero di fault e failure presenti nel sistema.

1.1 Definizioni, Acronimi e Abbreviazioni

Definizioni:

- Branch Coverage: copertura da parte dei test di tutti i rami o stati condizionali del sistema;
- Failure: mancata prestazione di un servizio atteso;
- Fault: causa di un failure, insieme di istruzioni che nel momento in cui vengono eseguite generano un fallimento;
- Three Tier: modello architetturale che viene utilizzato per dividere il prodotto software in tre blocchi ognuno con una funzionalità ben distinta.

Acronimi e Abbreviazioni:

- RAD: Abbreviazione utilizzata per indicare il Requirement Analysis Document;
- SDD: Abbreviazione utilizzata per indicare il System Design Document;
- ODD: Abbreviazione utilizzata per indicare il Object Design Document;
- SOW: Abbreviazione utilizzata per indicare lo Statement Of Work;
- BC: Abbreviazione utilizzata per indicare il Business Case;
- TP: Abbreviazione utilizzata per indicare il Test Plan;
- STC: Abbreviazione utilizzata per indicare il System Test Case;
- DB: Abbreviazione utilizzata per indicare il Database;
- TC: Abbreviazione utilizzata per indicare il Test Case



1.2 Riferimenti ad altri documenti

Per la corretta individuazione dei casi di test si fa riferimento ai seguenti documenti:

- Relazione con il Requirements Analysis Document (RAD): Tale relazione si fonda sul fatto che i test case verranno selezionati sulla base dei requisiti funzionali e non funzionali espressi nel documento sopracitato.
- Relazione con il System Design Document (SDD): Tale relazione comprende la suddivisione in sottosistemi effettuata in fase di system design e che dovrà essere preservata durante la fase di pianificazione del testing.
- Relazione con l'Object Design Document (ODD): In questo documento (ancora non sviluppato al momento del rilascio dell'C04_TP_Vers.1.0) sono contenuti i package e le classi del sistema.
- Relazione con lo Statement of Work (SOW): Tale relazione comprende la suddivisione in sottosistemi effettuata in fase di system design e che dovrà essere preservata durante la fase di pianificazione del testing.

2. Panoramica del sistema

Lo scopo del progetto EnIA è quello di offrire ai propri clienti una piattaforma web per il supporto nelle decisioni relative ad attività agroindustriali. Gli obiettivi principali di EnIA sono:

- ridurre al minimo i consumi delle risorse idriche, diminuendo al contempo l'impatto ambientale;
- fornire un tracciamento del livello di inquinamento ed esposizione ambientale della coltivazione;
- localizzazione e gestione dei vari terreni.

Tali obiettivi saranno raggiunti tramite le seguenti implementazioni di funzionalità:

- la localizzazione e gestione dei vari terreni;
- le previsioni microclimatiche e delle precipitazioni per i luoghi di interesse provenienti da open source intelligence;
- MODULO FIA: sviluppo di un modulo di intelligenza artificiale per supportare l'agricoltore nella fase decisionale tramite gli insight della piattaforma;
- informazioni inerenti ai livelli di inquinamento dei luoghi di interesse;
- una fase di analisi delle previsioni meteo con lo scopo di gestire gli impianti di irrigazione;
- una fase di analisi dell'esposizione ambientale delle varie coltivazioni.



L'architettura utilizzata è di tipo three tier. In questo tipo di architettura i sottosistemi che compongono il software sono suddivisi in tre livelli: data tier, application tier e presentation tier.

3. Funzionalità da testare

Nell'ambito del testing di sistema si andranno a verificare le principali funzionalità per ciascuna gestione funzionale individuata.

Le funzionalità selezionate da testare sono:

- Gestione Ambiente Agricolo
 - Aggiunta Ambiente Agricolo
 - Modifica Ambiente Agricolo
 - Eliminazione Ambiente Agricolo
 - Visualizzazione Ambiente Agricolo
 - Visualizzazione Meteo
- Gestione Utente
 - Registrazione Farmer
 - Gestione account utente
 - Visualizzazione area utente
 - Modifica dati utenti
- Gestione Azienda Agricola
 - Visualizzazione degli utenti
 - Aggiunta di utente e assegnazione dei ruoli
 - Rimozione utente
- Gestione degli insight
 - Decision Intelligence

4. Funzionalità da non testare

Le funzionalità escluse dal testing riguardano i requisiti funzionali di bassa e media priorità che verranno implementati nelle successive release.

Le funzionalità non selezionate per il testing sono:

- Gestione Ambiente Agricolo
 - Assegnazione priorità ambiente agricolo
- Gestione Inquinamento



Laurea Magistrale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

- Download Report Agenti Inquinanti
- Visualizzazione Agenti Inquinanti
- Gestione degli insight
 - Sistema di alert
 - Storico eventi
 - Rilevamento eventi estremi

5. Approccio

La fase di test dell'intero sistema si divide in tre fasi:

- Unit testing;
- Integration testing;
- System testing.

I test rispettivi saranno eseguiti in quest'ordine, e progettati in ordine inverso.

La progettazione dei casi di system test sarà effettuata prima della fase di implementazione del sistema. Tali casi saranno poi migliorati e perfezionati durante la loro fase di esecuzione. Il codice sarà sottoposto a periodiche attività di revisione durante tutto lo sviluppo.

Il testing sarà effettuato sfruttando un approccio di tipo Bottom-up, partendo dal testing e dall'integrazione delle singole componenti del livello più basso, per poi passare all'application tier, fino a raggiungere le interfacce degli utenti.

Le tecnologie utilizzate durante la fase di testing sono:

- **PyBuilder:** utilizzato per effettuare build del progetto, che hanno come preconditione l'esecuzione dei test e l'analisi del progetto.
- **Unittest:** Libreria utilizzata da PyBuilder per il testing di unità e di sistema;
- **Coverage.Py:** Libreria utilizzata da PyBuilder per effettuare l'analisi del code coverage e valutare l'efficacia dei test;
- **Selenium:** utilizzata per testare l'interazione dell'utente con la piattaforma.

Unit testing

Durante questa fase verranno testate le singole unità software attraverso un controllo delle varie classi e metodi al fine di ricercare eventuali condizioni di fallimento.

Lo Unit testing sarà effettuato dal team di sviluppo tramite l'utilizzo di classi di test, sfruttando il framework Unittest (sarà creata una relativa classe Unittest per ogni test case).



Integration testing

L'Integration testing sarà effettuato per assicurare il corretto funzionamento di componenti, i quali funzionano bene individualmente, nel momento in cui vengono associati ad altri componenti.

Lo scopo è quello di individuare eventuali problemi inerenti alla loro interfaccia.

Verrà applicato un approccio di tipo bottom-up, e saranno creati dei driver utili a simulare le componenti più ad alto livello non ancora testate.

System testing

Il System testing serve a verificare che l'intero sistema sia coerente con quanto richiesto nei requisiti funzionali e non funzionali e quindi non richiede il codice sorgente del sistema.

Per questa fase verrà adottato un approccio di tipo black-box per la definizione dei suites test, più precisamente verrà utilizzata la metodologia Category Partition, la quale permette di ottenere un buon livello di dettaglio nel tempo a nostra disposizione.

Il system testing verrà effettuato nelle fasi finali dell'implementazione tramite l'ausilio di Selenium, al fine di testare in modo completo l'interfaccia utente.

6. Criteria Pass/Fail

Un test case avrà successo (pass) se la sua esecuzione porterà ad un risultato uguale a quello atteso.

Viceversa, sarà considerato fallito (fail) se il risultato ottenuto sarà differente da quello atteso.

Tutto il testing sarà considerato valido se verranno rispettate tutte le seguenti condizioni:

- Testare tutti i requisiti da testare indicati precedentemente;
- Effettuare test di regressione ogni qualvolta verranno effettuate le modifiche alle componenti del sistema;

7. Criteri di Sospensione e Ripresa

Si andranno a definire i criteri di sospensione del testing. Inoltre, si specificano i criteri per riprendere le attività di testing in caso di sospensione.

Criteri di sospensione



La fase di testing verrà interrotta quando verranno raggiunti i risultati attesi in accordo con i tempi e i costi allocati alle attività di testing.

Criteri di ripresa

Le attività di testing riprenderanno in seguito a delle modifiche che potrebbero introdurre nuovi errori all'interno del sistema.

8. Materiale per il testing

L'hardware necessario per l'attività di testing sarà un pc, non necessariamente avente connessione ad Internet, in quanto il sistema verrà testato in locale.

9. Test cases

La definizione dei test case sarà effettuata attraverso l'utilizzo del category partition. Per definire l'output atteso si userà un oracolo.

9.1 Gestione Utente

9.1.1 Registrazione Utente tramite codice di accesso

Parametro: E-Mail	
FORMATO: /[A-Z0-9._%+-]+@[A-Z0-9-]+.[A-Z]{2,4}/	
Nome Categoria	Scelte per la categoria
Formato [FE]	1. Formato non valido [error] 2. Formato valido [PROPERTY FE_OK]
Parametro: Password	
Nome Categoria	Scelte per la categoria
FORMATO: /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[\$@!%*?&_])[A-Za-z\d@\$!%*?&_]{8,20}\$/	
Formato [FP]	1. Formato non valido [error] 2. Formato valido [PROPERTY FP_OK]
Parametro: Nome	
FORMATO: /^[a-z]{2,30}\$/	
Nome Categoria	Scelte per la categoria
Formato [FNO]	1. Formato non valido [error] 2. Formato valido [PROPERTY FNO_OK]
Parametro: Cognome	
FORMATO: /^[a-z]{2,30}\$/	



Nome Categoria	Scelte per la categoria
Formato [FCN]	1. Formato non valido [error] 2. Formato valido [PROPERTY FCN_OK]
Parametro: Codice Di Accesso	
Nome Categoria	Scelte per la categoria
MatchDB [DBCA]	1. Codice non presente nel sistema [error] 2. Codice presente nel sistema [PROPERTY DBCA_OK]
Non Usato [NUCA]	1. Codice usato [error] 2. Codice non usato [PROPERTY NUCA_OK]
Parametro: Indirizzo	
FORMATO: /^[A-Za-zÀ-Û0-9,‘]+\$/	
Nome Categoria	Scelte per la categoria
Formato [FI]	1. Formato non valido [error] 2. Formato valido [PROPERTY FI_OK]
Esistenza [EI]	1. Indirizzo non esistente [error] 2. Indirizzo esistente [PROPERTY EI_OK]
Parametro: Data di Nascita	

Commentato [ML1]: Abbiamo eliminato la lunghezza perchè il codice viene generato automaticamente

Test Case ID	Test Frame	Esito
TC_1.1_1	FE1	Errore: Formato e-mail non corretto
TC_1.1_2	FE2, FP1	Errore: Formato password non valido
TC_1.1_3	FE2, FP2, FNO1	Errore: Nome non valido
TC_1.1_4	FE2, FP2, FNO2, FCN1	Errore: Cognome non valido
TC_1.1_5	FE2, FP2, FNO2, FCN2, DBCA1	Errore: Il codice di accesso non è presente nel sistema
TC_1.1_6	FE2, FP2, FNO2, FCN2, DBCA2, NUCA1	Errore: Il codice di accesso è già stato usato da un altro utente
TC_1.1_7	FE2, FP2, FNO2, FCN2, DBCA2, NUCA2, FI1	Errore: Il formato dell'indirizzo non è valido
TC_1.1_8	FE2, FP2, FNO2, FCN2, DBCA2, NUCA2, FI2, EI1	Errore: L'indirizzo non esiste
TC_1.1_9	FE2, FP2, FNO2, FCN2, DBCA2, NUCA2, FI2, EI2	Esito Valido



9.2 Gestione Ambiente Agricolo

9.2.1 Inserimento di un terreno

Parametro: Nome	
Formato: /[\\w\\s]+\$/	
Nome Categoria	Scelte per la categoria
Formato [FN]	1. Formato non valido [errore] 2. Formato valido [PROPERTY_FN_OK]
Esistenza [EN]	1. Nome non inserito [errore] 2. Nome inserito [PROPERTY_EN_OK]
Parametro: Coltura	
Formato: ["Orzo", "Fagiolo", "Cavolo", "Carota", "Cotone", "Cetriolo", "Melanzana", "Grano", "Lenticchia", "Lattuga"]	
Nome Categoria	Scelte per la categoria
Esistenza [EC]	1. Coltura non Inserita [errore] 2. Coltura inserita [PROPERTY_EC_OK]
Validità [VC]	1. Coltura non appartiene a ["Orzo", "Fagiolo", "Cavolo", "Carota", "Cotone", "Cetriolo", "Melanzana", "Grano", "Lenticchia", "Lattuga"] [errore] 2. Coltura appartiene a ["Orzo", "Fagiolo", "Cavolo", "Carota", "Cotone", "Cetriolo", "Melanzana", "Grano", "Lenticchia", "Lattuga"] [PROPERTY_V_OK]
Parametro: Posizione sulla mappa	
Formato: GEOJSON • Polygon	



Nome Categoria	Scelte per la categoria
Esistenza [EP]	1. Geojson non inserito [errore] 2. Geojson inserito [PROPERTY_EP_OK]
Formato [FP]	1. Formato non valido [errore] 2. Formato valido [PROPERTY_FP_OK]
Parametro: Priorita	
Formato: Integer	
Parametro: Preferito	
Formato: Boolean	
Nome Categoria	Scelte per la categoria
Formato [FC]	1. Formato non valido [errore] 2. Formato valido [PROPERTY_FC_OK]
Parametro: Proprietario	
Parametro: Stadio di Crescita	
Formato: ["Iniziale", "Sviluppo", "Metà Stagione", "Fine Stagione"]	
Nome Categoria	Scelte per la categoria
Validità [VS]	1. StadioCrescita non appartiene a ["Iniziale", "Sviluppo", "Metà Stagione", "Fine Stagione"] [errore] 2. StadioCrescita appartiene a ["Iniziale", "Sviluppo", "Metà Stagione", "Fine Stagione"] [PROPERTY_V_OK]

Test Case ID	Test frame	Esito
TC_2.1_1	FN1	Errato: Formato nome non valido
TC_2.1_2	FN2, EN1	Errato: Nome non inserito



TC_2.1_3	FN2, EN2, EC1	Errato: Coltura non inserita
TC_2.1_4	FN2, EN2, EC2, VC1	Errato: Formato coltura non valido
TC_2.1_5	FN2, EN2, EC2, VC2, EP1	Errato: Geojson non inserito
TC_2.1_6	FN2, EN2, EC2, VC2, EP2, FP1	Errato: Formato Geojson non valido
TC_2.1_7	FN2, EN2, EC2, VC2, EP2, FP2, FC1	Errato: formato non valido di preferito
TC_2.1_8	FN2, EN2, EC2, VC2, EP2, FP2, FC2, VS1	Errato: formato stadio crescita non valido
TC_2.1_9	FN2, EN2, EC2, VC2, EP2, FP2, FC2, VS2	Corretto

9.3 Gestione Meteo

9.3.1 Visualizzazione Meteo

Parametro: Latitudine	
Nome Categoria	Scelte per la categoria
Range [RL]	1. Latitudine > 90 or Latitudine < -90 [error] 2. Latitudine > -90 and Latitudine < 90 [PROPERTY-R-OK]
Parametro: Longitudine	
Nome Categoria	Scelte per la categoria
Range [RLA]	1. Longitudine > 180 or Longitudine < -180 [error] 2. Longitudine <= 180 and Longitudine >=-180 [PROPERTY-R-OK]

Test Case ID	Test Frame	Esito
TC_3.1_1	RL1	Errore: Il valore inserito della latitudine non ricade nel suo range concesso



Laurea Magistrale in informatica-Università di Salerno
Corso di Ingegneria del Software - Prof.ssa F. Ferrucci, Prof. F. Palomba

TC_3.1_2	RL2, RLA1	Errore: Il valore inserito della longitudine non ricade nel suo range concesso
TC_3.1_3	RL2, RLA2	Corretto

9.4 Gestione Azienda Agricola

9.4.1 Aggiunta di Utente e Assegnazione Dei Ruoli

Parametro: Ruolo	
FORMATO: ['Irrigation Manager', 'Pollution Analyst']	
Nome Categoria	Scelte per la categoria
Correttezza [CRR]	1. Ruolo diverso da "Irrigation Manager" oppure "Pollution Analyst" [errore] 2. Ruolo uguale ad "Irrigation Manager" oppure "Pollution Analyst" [PROPERTY CRR_OK]

Test Case ID	Test Frame	Esito
TC_4.1_1	CRR1	Errore: Ruolo non valido
TC_4.1_2	CRR2	Esito Positivo

9.5 Gestione degli insight dell'AI

9.5.1 Decision Intelligence

Parametro: Stadio Di Crescita	
Nome Categoria	Scelte per la categoria
Validità (VS)	3. StadioCrescita non appartiene a["Iniziale, "Sviluppo"Metà Stagione", "Fine Stagione"] [errore] 4. StadioCrescita appartiene a ["Iniziale, "Sviluppo"Metà, Stagione", "Fine Stagione"] [PROPERTY_VS_OK]
Parametro: Latitudine	
Nome Categoria	Scelte per la categoria
Range (RLA)	1. Latitudine > 90 oppure Latitudine <-90 [errore] 2. Latitudine >-90 e Latitudine <90 [PROPERTY-RLA-OK]



Parametro: Longitudine	
Nome Categoria	Scelte per la categoria
Range (RLO)	1. Longitudine > 180 oppure Longitudine <-180 [errore] 2. Longitudine <= 180 e Longitudine >=-180 [PROPERTY-RLO-OK]
Parametro: Coltura	
Nome Categoria	Scelte per la categoria
Validità (VC)	3. Coltura non appartiene a ["Orzo", "Fagiolo","Cavolo","Carota", "Cotone", "Cetriolo", "Melanzana", "Grano", "Lenticchia", "Lattuga"] [errore] 4. Coltura appartiene a ["Orzo", "Fagiolo","Cavolo","Carota", "Cotone", "Cetriolo", "Melanzana", "Grano", "Lenticchia", "Lattuga"] [PROPERTY_VC_OK]

Test Case ID	Test Frame	Esito
TC_5.1_1	VS_1	Errore: Stadio di crescita non è un valore valido.
TC_5.1_2	VS_2, RLA_1,	Errore: Il valore inserito della latitudine non ricade nel suo range concesso
TC_5.1_3	VS_2, RLA_2, RLO_1	Errore: Il valore inserito della longitudine non ricade nel suo range concesso
TC_5.1_4	VS_2, RLA_2, RLO_2, VC_1	Errore: La coltura inserita non è valida
TC_5.1_5	VS_2, RLA_2, RLO_2, VC_2	Corretto

10. Testing schedule

La pianificazione delle attività di testing sarà avviata successivamente alla fine della fase di progettazione dei test rispettivi.

Si prevede di eseguire i test durante la fase di implementazione del sistema. Inoltre, alcuni test saranno ripetuti qualora vi fosse possibilità di regressione durante lo sviluppo del codice.

I Test Case prodotti verranno raffinati e rivisti durante la fase di scrittura del codice.



Laurea Magistrale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba