

T.O.A.S.T

Un bot Discord per l'ispezione manuale
dei Community Smells

Pre-Maintenance Test Case Specification

Corso: Ingegneria Gestione ed Evoluzione del Software

Prof. Andrea De Lucia

Dott. Antonio Della Porta

Università degli Studi di Salerno

Anno accademico: 2023-2024

Repository GitHub:

<https://github.com/Ferdi00/toast-tool>

Team Members

Ferdinando Boccia

f.boccia28@studenti.unisa.it

Domenico D'Antuono

d.dantuono7@studenti.unisa.it

Contents

1	Introduzione	2
2	Testing funzionale	2
2.1	RF_1 - Aggiunta di un collaboratore	2
2.1.1	Tabella: Test Frame RF_1	4
2.2	RF_2	4
2.2.1	Tabella: Test Frame RF_2	5
2.3	Gestione della persistenza	5
2.3.1	Tabella: Test Frame GP	7
3	Test Case Specification	7
3.1	Table 1: Test Case TC-RF1.1	7
3.2	Table 2: Test Case TC-RF1.2	7
3.3	Table 3: Test Case TC-RF2.1	8
3.4	Table 4: Test Case TC-RF2.2	9
3.5	Table 5: Test Case TC-GP.1	10
3.6	Table 6: Test Case TC-GP.2	11

1 Introduzione

Il presente documento esamina le attività di testing disponibili per lo strumento Toast, propedeutiche al processo di manutenzione ed evoluzione. Si evidenzia che non sono stati forniti test case relativi al progetto originale.

A seguito del reverse engineering effettuato sullo strumento, sono stati individuati i seguenti requisiti funzionali:

1. **RF_1 Aggiunta di un collaboratore:** Il sistema deve consentire al manager l'inserimento di uno o più collaboratori nel team da lui gestito.
2. **RF_2 Analisi del collaboratore** Il sistema deve fornire la possibilità di eseguire l'analisi dei collaboratori tramite un questionario, finalizzato all'individuazione di eventuali community smell.
3. **RF_3 Gestione della persistenza:** Il sistema deve garantire la memorizzazione persistente dei dati relativi agli utenti (manager e collaboratori).

Poiché le Change Requests influenzeranno tutti i requisiti, si procederà con il testing funzionale relativo a ciascuno di essi.

2 Testing funzionale

2.1 RF_1 - Aggiunta di un collaboratore

Descrizione Il bot Discord deve ricevere come input il nome, il cognome e l'ID del collaboratore da aggiungere, e deve restituire un messaggio di conferma dell'avvenuta aggiunta del collaboratore.

Parametri di input: nome, cognome, ID.

Nome Categoria	Valori Categoria
Dati Collaboratore	<ul style="list-style-type: none"> • DC1: Dati del collaboratore validi. • DC2: Campi mancanti (es. manca il nome o l'ID collaboratore).

Attributo	Vincolo
Dati Collaboratore	<ul style="list-style-type: none"> • DC1: Dati validi [proprietà dati_corretti]. • DC2: Campi mancanti [errore].

2.1.1 Tabella: Test Frame RF_1

ID Caso di Test	Test Frame	Risultato Atteso
TC-RF1.1	DC1	Successo: Collaboratore aggiunto correttamente.
TC-RF1.2	DC2	Errore: Dati del collaboratore incompleti.

2.2 RF_2

Descrizione

Il bot discord deve prendere in input le risposte alle domande del questionario e dare in output i risultati dell'analisi tramite messaggio.

Parametri: Risposte dell' utente

Nome Categoria	Valori Categoria
Risposte dell' utente	<ul style="list-style-type: none">• RU1: Tutte le risposte richieste sono state inserite.• RU2: Risposte mancanti.

Attributo	Vincolo
Risposte dell' utente	<ul style="list-style-type: none"> • RU1: JSON valido e completo delle risposte richieste [proprietà risposte_valide]. • RU2: Risposte mancanti [il bot resta in attesa].

2.2.1 Tabella: Test Frame RF_2

ID Test	Caso di	Test Frame	Risultato Atteso
TC_RF2.1		RU1	Successo : Restituisce i risultati dell'analisi in un messaggio.
TC_RF2.2		RU2	Errore : Il bot resta in attesa.

2.3 Gestione della persistenza

Descrizione

Il bot discord deve prendere in input il nome, il cognome e l'id del collaboratore e inserire correttamente i dati nel file di persistenza.

Parametri: nome, cognome, id.

Nome Categoria	Valori Categoria
Dati Collaboratore	<ul style="list-style-type: none"> • DC1: Tutti i dati sono stati inesriti. • DC2: Campi mancanti (es. manca il nome o l'ID collaboratore).

Attributo	Vincolo
Dati Collaboratore	<ul style="list-style-type: none"> • DC1: Tutti i dati sono stati inesriti [proprietà dati_corretti]. • DC2: Campi mancanti [errore].

2.3.1 Tabella: Test Frame GP

ID Caso di Test	Test Frame	Risultato Atteso
TC-GP.1	DC1	Successo: Collaboratore aggiunto correttamente nel file di persistenza.
TC-GP.2	DC2	Errore: Dati del collaboratore incompleti.

3 Test Case Specification

3.1 Table 1: Test Case TC-RF1.1

Test Case ID	TC-RF1.1
Test Frame	DC1
Pre-condizioni	<ul style="list-style-type: none">• Il sistema è attivo e in esecuzione.• L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.
Flusso degli eventi	<ul style="list-style-type: none">• L'utente avvia il bot tramite il comando start.• Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni.• L'utente seleziona "add a new collaborator".• Un form per l'aggiunta dei collaboratori viene mostrato all'utente• L'utente inserisci questi valori nel form:<ul style="list-style-type: none">– { "collaboratorId": "23", "name": "Simone", "surname": "Scala" }
Risultato atteso	Il bot risponde con il messaggio "Data saved".

3.2 Table 2: Test Case TC-RF1.2

Test Case ID	TC-RF1.2
Test Frame	DC2
Pre-condizioni	<ul style="list-style-type: none"> • Il sistema è attivo e in esecuzione. • L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.
Flusso degli eventi	<ul style="list-style-type: none"> • L'utente avvia il bot tramite il comando start. • Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni. • L'utente seleziona "add a new collaborator". • Un form per l'aggiunta dei collaboratori viene mostrato all'utente • L'utente inserisci questi valori nel form: <ul style="list-style-type: none"> – { "collaboratorId": "12", "name": "", "sur-name": "Rossi" }
Risultato atteso	Una notifica avvisa di inserire il nome prima di proseguire.

3.3 Table 3: Test Case TC-RF2.1

Test Case ID	TC-RF2.1
Test Frame	RU1
Pre-condizioni	<ul style="list-style-type: none"> • Il sistema è attivo e in esecuzione. • L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.

Flusso degli eventi	<ul style="list-style-type: none"> • L'utente avvia il bot tramite il comando start. • Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni. • L'utente seleziona "Start the analysis of a collaborator". • Il bot chiede di scegliere il collaboratore da analizzare. • L'utente seleziona il collaboratore con id 1234. • Il bot sottopone le domande del questionario all'utente. • L'utente risponde a tutte le domande del bot selezionando ogni volta il bottone relativo alla risposta desiderata e fornisce questi valori: <ul style="list-style-type: none"> – { "ans1": "dis", "ans2": "neutral", "ans3": "agree", "ans4": "dis", "ans5": "agree", "ans6": "dis" }
Risultato atteso	Il bot risponde con un messaggio che contiene questi risultati: { "LW": "0", "PD": "2.3508905", "BC": "1.7881185" }

3.4 Table 4: Test Case TC-RF2.2

Test Case ID	TC-RF2.2
Test Frame	RU2
Pre-condizioni	<ul style="list-style-type: none"> • Il sistema è attivo e in esecuzione. • L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.

Flusso degli eventi	<ul style="list-style-type: none"> • L'utente avvia il bot tramite il comando start. • Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni. • L'utente seleziona "Start the analysis of a collaborator". • Il bot chiede di scegliere il collaboratore da analizzare. • L'utente seleziona il collaboratore con id 1234. • Il bot sottopone le domande del questionario all'utente. • L'utente risponde alle domande del bot selezionando ogni volta il bottone relativo alla risposta desiderata e fornisce questi valori: <ul style="list-style-type: none"> – { "ans1": "dis", "ans2": "neutral", "ans3": "agree", "ans4": "dis", "ans5": "agree", "ans6": "" }
Risultato atteso	Il bot rimane in attesa dell'ultima risposta

3.5 Table 5: Test Case TC-GP.1

Test Case ID	TC-RF1.1
Test Frame	DC1
Pre-condizioni	<ul style="list-style-type: none"> • Il sistema è attivo e in esecuzione. • L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.

Flusso degli eventi	<ul style="list-style-type: none"> • L'utente avvia il bot tramite il comando start. • Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni. • L'utente seleziona "add a new collaborator". • Un form per l'aggiunta dei collaboratori viene mostrato all'utente • L'utente inserisci questi valori nel form: <ul style="list-style-type: none"> – { "collaboratorId": "23", "name": "Davide", "surname": "Coccorullo" }
Risultato atteso	Il dati del nuovo collaboratore vengono aggiunti al file di persistenza.

3.6 Table 6: Test Case TC-GP.2

Test Case ID	TC-GP.2
Test Frame	DC2
Pre-condizioni	<ul style="list-style-type: none"> • Il sistema è attivo e in esecuzione. • L'utente ha fatto il login a discord e si trova in un gruppo dove il bot è stato aggiunto.

Flusso degli eventi	<ul style="list-style-type: none"> • L'utente avvia il bot tramite il comando start. • Il bot risponde con un messaggio di benvenuto con un campo per selezionare le possibili azioni. • L'utente seleziona "add a new collaborator". • Un form per l'aggiunta dei collaboratori viene mostrato all'utente • L'utente inserisci questi valori nel form: <ul style="list-style-type: none"> – { "collaboratorId": "12", "name": "", "sur-name": "Rossi" }
Risultato atteso	Una notifica avvisa di inserire il nome prima di proseguire.