



Test Plan

AM-GP

Progetto	AM-GP
Documento	Test Plan

Versione	Descrizione	Autore
0.5	Prima bozza	Pietro Negri
1.0	Aggiunta riferimenti a Test Case Specification, Schedule	Pietro Negri

Indice generale

1. Introduzione.....	4
2. Relazioni con altri documenti.....	4
2.1 RAD:.....	4
2.2 SDD:.....	4
2.3 ODD:.....	4
3. Panoramica del Sistema.....	5
4. Funzionalità da Testare.....	5
5. Criteri Pass/Fail.....	6
5.1 Pass Criteria.....	6
5.2 Fail Criteria.....	6
6. Approccio.....	6
6.1 Testing di Unità.....	6
6.2 Test di Integrazione.....	6
6.2 Test di Sistema.....	6
7. Sospensione e ripresa.....	7
8. Materiale per il Testing.....	7
9. Test Cases.....	7
10. Testing Schedule.....	7

1. Introduzione

Lo scopo di questo documento è quello di descrivere e pianificare le attività di testing per la piattaforma privata AM-GP.

L'obiettivo è quello di analizzare a fondo il sistema proposto, cercando di massimizzarne l'affidabilità, selezionando dei casi di test che consentano di individuare eventuali comportamenti osservati che differiscono da quelli specificati nei documenti correlati.

Va specificato che non tutte le componenti specificate sono state implementate nella prima versione del sistema.

2. Relazioni con altri documenti

Il seguente Test Plan è così relazionato con i documenti consegnati precedentemente:

2.1 RAD:

Nel documento di Analisi dei Requisiti abbiamo descritto i Requisiti Funzionali del sistema proposto, oltre che quelli non funzionali. Non tutte le funzionalità descritte sono state implementate, in particolare non abbiamo implementato il ramo di funzionalità per l'utente Stratega.

La motivazione è da ricercare nel fatto che il ramo di funzionalità per l'utente Stratega, rispetto all'utente Tecnico ed è stato specificato come un requisito a media priorità, di conseguenza rappresenta un insieme di funzionalità che può essere aggiunto in seguito in una successiva versione del sistema, data la sua

2.2 SDD:

La suddivisione in sottosistemi descritta nel System Design Document sarà utilizzata per suddividere le attività di testing. I sottosistemi citati in questo documenti si rifanno direttamente a quelli specificati nel SDD.

2.3 ODD:

Le interfacce delle classi definite nell'Object Design Document saranno utilizzate per l'individuazione delle categorie su cui si basa la selezione dei casi di test. Si fa riferimento all'Object Design Document per eventuali domande riguardanti le interfacce delle classi.

3. Panoramica del Sistema

Il sistema verrà testato tenendo conto della sua decomposizione in sottosistemi.

Ricordiamo che il sistema è strutturato secondo un'architettura Three Tier, in cui sono presenti tre layer:

Presentation, Business Logic e Storage.

Ci concentreremo principalmente sul testare i sottosistemi del layer di Business, per cui i sottosistemi che verranno testati saranno:

- CircuitiManagement
- SetupManagement
- UserManagement
- MessageManagement

4. Funzionalità da Testare

In base alla parte implementata e alla suddivisione in sottosistemi effettuata, abbiamo deciso di testare le seguenti funzionalità:

- Sottosistema UserManagement:
 - Autenticazione
 - Disconnessione
 - Visualizzazione statistiche riepilogative per Utenti Piloti
- Sottosistema MessageManagement:
 - Invio Messaggio
 - Lettura Messaggio
 - Visualizzazione Messaggi Ricevuti
 - Visualizzazione Messaggi Inviati
- Sottosistema CircuitiManagement:
 - Visualizzazione Lista Circuiti
 - Visualizzazione Circuito
- Sottosistema SetupManagement:
 - Aggiunta di un Setup
 - Modifica di un Setup
 - Eliminazione di un Setup
 - Visualizzazione di un Setup
 - Visualizzazione Lista di Setup

5. Criteri Pass/Fail

5.1 Pass Criteria

Per superare la fase di testing il criterio è che siano stati eseguiti tutti i test e che almeno il 75% di essi dia esito positivo.

5.2 Fail Criteria

Se almeno il 30% dei Test Case fallisce, lo sviluppo viene interrotto e tutti i task devono essere sospesi finché non vengono individuati i problemi.

6. Approccio

Si prevede di effettuare testing a livello di unità, integrazione e sistema.

In particolare, verranno testate inizialmente le classi di cui è composto il sistema. Si procederà poi ad analizzare l'interazione tra le classi e i sottosistemi e infine verrà testato l'intero sistema.

6.1 Testing di Unità

Durante questa fase le singole componenti verranno testate in isolamento, sfruttando la specifica delle interfacce descritte nell'Object Design Document. Prevediamo di sfruttare metodologie di selezione dei casi di test Black-Box, sfruttando in particolare Category Partitioning.

6.2 Test di Integrazione

La strategia adoperata per il test di Integrazione sarà di tipo Bottom-Up. Testeremo prima le classi del livello più basso, quelle che interagiscono con il Database, per poi passare alle classi di livello più alto.

6.2 Test di Sistema

Ci focalizzeremo sulle componenti con il quale l'utente può interagire, effettuando modifiche al sistema, al fine di verificare che il sistema soddisfi i requisiti dettagliati nel documento di Analisi dei Requisiti.

7. Sospensione e ripresa

Il testing verrà terminato nel momento in cui tutte le funzionalità descritte in questo documento saranno testate, in accordo anche con il documento di Test Case Specification.

E' possibile che il testing venga sospeso per risolvere delle falle individuate. In tal caso, verrà effettuato un test di regressione per assicurare il corretto funzionamento delle funzionalità già testate, in seguito alla correzione della falla. In tal caso, verranno rieseguiti tutti i casi di test specificati nel documento di Test Case Specification.

8. Materiale per il Testing

Per effettuare il testing utilizzeremo, oltre ai documenti citati fino a questo punto, gli strumenti Selenium e la famiglia di framework Xunit.

9. Test Cases

Nel documento di Test Case Specification sono presenti sia i partizionamenti del Category Partitioning, sia la descrizione di tutti i casi di test per le funzionalità e i sottosistemi testati.

10. Testing Schedule

Task	Tempo stimato
Creazione test specification	20 Ore
Esecuzione test	35 Ore