

Easy Fut5al



Test Plan Document

Sommario

1. Introduzione.....	2
2. Riferimenti	Errore. Il segnalibro non è definito.
3. Obiettivi del testing	Errore. Il segnalibro non è definito.
4. Funzionalità da testare	Errore. Il segnalibro non è definito.
1. Introduzione.....	2
2. Riferimenti	2
3. Obiettivo del testing	2
4. Funzionalità da testare	2
5. Funzionalità da non testare	3
6. Approccio	3
7. Criteri di sospensione e di uscita	3
8. Test cases	4

1. Introduzione

Il Test plan si focalizza sugli aspetti manageriali del testing, in particolare sullo scopo, l'approccio e lo scheduling di tutte le attività del testing del progetto Easy-Fut5al.

2. Riferimenti

- Bernd Bruegge & Allen H. Dutoit, *Object-Oriented Software*;
- RAD-EasyFut5al;

3. Obiettivo del testing

L'obiettivo del testing che vogliamo effettuare è quello di verificare le funzionalità individuate per il progetto EasyFut5al, in particolare si vogliono individuare le differenze tra il comportamento atteso specificato nel modello di sistema e il comportamento osservato con l'immissione di alcuni input selezionati.

Nello specifico, ci è stato richiesto di sviluppare la documentazione di pianificazione ed esecuzione del testing per almeno uno dei sottosistemi. Per motivi di tempistica ne testeremo soltanto uno: **Storage**, il quale ci risulta il più agevole per effettuare testing black box non avendo a che fare con UI a eventi o Servlets.

4. Funzionalità da testare

Sottosistema di appartenenza	Nome funzionalità	Descrizione
Storage	Salva	Salva un oggetto Bean nel database.
Storage	GetLista	Torna una lista di Bean del tipo specificato.
Storage	GetOggetto	Torna un oggetto identificato da un ID.
Storage	Aggiorna	Sostituisce un record del db .
Storage	Elimina	Elimina un record nel db che corrisponde ai valori del bean dato in input.

5. Funzionalità da non testare

Tutte le funzioni relative ai package:

- Middleware**
- Prenotazioni Management**
- Conferma Campetto**
- Login/Logout**
- Registrazione Atleta/ Registrazione Gestore**
- Activity Atleta (comprende Sezione Home, FragmentNuovaPartita, Inviti)**
- Activity Gestore**
- Activity Autenticazione**
- Profilo**
- Calendario**
- ConnectionTask**

6. Approccio

Nella sessione di testing del sistema verrà utilizzato un approccio di tipo "BLACK BOX", che prevede che i test vengano effettuati in maniera da non scendere nei dettagli del codice, ma basandoci sulle specifiche delle funzionalità da testare.

7. Criteri di sospensione e di uscita

Criteri di sospensione

Abbiamo stabilito che se i membri del team riportano che più del 20% dei test case sono falliti, il testing dovrà essere sospeso fino a quando non verranno individuati e risolti i bug che hanno causato il fallimento.

Criteri di uscita

Specifica dei criteri che denotano il successo della fase di testing:

- La percentuale di successo è dell'80% ed è la percentuale minima per il completamento del testing.

8. Test cases

Per sviluppare i test case sarà utilizzato il Metodo **WECT**:

Inizieremo a partizionare il dominio di input in diverse classi di equivalenza.

Nel caso di interi (es: ID), si crea una classe di equivalenza VALIDA per i valori nell'intervallo accettato e classi NON VALIDE per valori minori o maggiori che sfiorano l'intervallo.

I1: $1 \leq ID \leq \text{NumeroDiRecord}$

I2: $ID < 1$

I3: $ID > \text{NumeroDiRecord}$

Se trattasi di input di un elemento di un insieme discreto, crea una classe per ogni valore VALIDO e una classe che rappresenta i valori non ammissibili

S1: "storage.Atleta"

S2: "storage.Campetto"

S3: "storage.Gestore"

S4: "storage.Gioca"

S5: "storage.Partita"

S6: "Albicocca" //Non valido

Dopodiché si sceglie un valore per una variabile da ogni classe.

Test Case	I	S
WE1	I1	S1
WE2	I2	S2
WE3	I3	S3
WE4	I1	S4
WE5	I2	S5
WE6	I3	S6

Con queste premesse, sviluppiamo i test per le funzionalità indicate in precedenza:

Salva

Parametro: Bean da salvare

Codice TestCase	Combinazione	Esito
Salva_Atleta	S1	1
Salva_Gestore	S2	1
Salva_Campetto	S3	1
Salva_Gioca	S4	1
Salva_Partita	S5	1
Salva_ERR	S6	0

GetLista

Parametro: Nome lista

Codice TestCase	Combinazione	Esito
GetLista_Atleta	S1	Torna Lista
GetLista_Gestore	S2	Torna Lista
GetLista_Campetto	S3	Torna Lista
GetLista_Gioca	S4	Torna Lista
GetLista_Partita	S5	Torna Lista
GetLista_ERR	S6	Torna vuota

GetOggetto

Parametri: Stringa tipo oggetto, ID oggetto

Codice TestCase	Combinazione	Esito
Get_Oggetto_AtletaValido	I1.S1	BeanAtleta
Get_Oggetto_GestoreNonValido	I2.S2	BeanGestore
Get_OggettoCampettoNonValido	I3.S3	BeanCampetto
Get_Oggetto_GiocaValido	I1.S4	BeanGioca
Get_OggettoPartitaNonValido	I2.S5	BeanPartita
Get_OggettoNullNonValido	I3.S6	Null

Aggiorna

Parametri: Bean da rimpiazzare

Codice TestCase	Combinazione	Esito
Aggiorna_Atleta	S1	1
Aggiorna_Gestore	S2	1
Aggiorna_Campetto	S3	1
Aggiorna_Gioca	S4	1
Aggiorna_Partita	S5	1
Aggiorna_ERR	S6	0

Elimina

Parametri: Bean da eliminare

//E' stata implementata solo la funzione Elimina per il Bean Gioca

Codice TestCase	Combinazione	Esito
Elimina_Gioca	S1	1

