

Ingegneria, Gestione ed Evoluzione del Software

Test Plan Unità Social Notes



Luigi Allocca	0522501559	l.allocca8@studenti.unisa.it
Rocco Iuliano	0522501533	r.iuliano13@studenti.unisa.it
Simone Della Porta	0522501534	s.dellaporta6@studenti.unisa.it

Approccio

Come riportato nel Master Test Plan, andremo ad applicare il Category Partition per la definizione dei test case; inoltre applicheremo l'approccio di Test Driven Development che consente di sviluppare prima i casi di test e successivamente sviluppare o modificare la componente in modo da evitare di introdurre potenziali failure nel sistema.

1 Relazione con altri documenti

Il test plan di unità è strettamente relato ai seguenti documenti:

- TP_unità precedente;
- Master Test Plan.

2 Modifiche test di unità

N.B: Numerazione e nomenclatura dei test fanno riferimento al Test Plan originale di SocialNotes.

2.1 Test cases CR1 modificati

Rimozione testCheckLogin(String name,String password).

Test cases UserModelDSTest

Table 1: Test case doSave()

Metodo	doSave(UserBean item)
Parametri	item
Oggetti d'ambiente	Database
Categorie	Scelte
Item	IT1: il nuovo utente che stiamo cercando di inserire non è già presente nel DB IT2: il nuovo utente che stiamo cercando di inserire è già presente nel DB [single] IT3: item=null [single]
Test frame	
IT1	ORACOLO: viene inserita una nuova istanza nella tabella <i>Utente</i> con le informazioni contenute in item. Inoltre il campo <i>verificato</i> sarà settato a false e il campo <i>bloccato</i> sarà settato a null
IT2	ORACOLO: viene generato un errore perché si cerca di salvare un utente già presente nel DB

IT3	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché non abbiamo fornito il parametro item
------------	--

Table 2: Test case doSave()

Metodo	testDoRetrieveByUsername(UserBean item)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
Item	IT1: email presente nel DB IT2: l'utente con username non è presente nel DB IT3: username = "" IT4: username = null
Test frame	
IT1	ORACOLO: viene restituita un'istanza di UserBean che contiene i dati dell'utente che ha l'username indicato dal parametro in input
IT2	ORACOLO: viene restituito null perché all'interno del DB non c'è nessun utente che ha l'username indicato dal parametro in input
IT3	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché non abbiamo fornito il parametro username
IT4	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché non abbiamo fornito il parametro username

Table 3: Test case doRetrieveByEmail()

Metodo	doRetrieveByEmail(String email)
Parametri	email
Oggetti d'ambiente	Database
Categorie	Scelte
Email	EM1: email presente nel DB EM2: email non presente nel DB EM3: email="" [single] EM4: email=null [single]
Test frame	
EM1	ORACOLO: viene restituita un'istanza di UserBean che contiene i dati dell'utente che ha la mail indicata dal parametro in input
EM2	ORACOLO: viene restituito null perché all'interno del DB non c'è nessun utente che ha la mail indicata dal parametro in input

EM3	ORACOLO: : viene generato un errore in quanto il metodo non può essere eseguito perché non abbiamo fornito il parametro email
EM4	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché non abbiamo fornito il parametro email

2.2 Test cases CR2 modificati

Rimozione testCheckLogin(String name,String password).

Modifica del test case per il metodo *doSave()* di UserModelIDSTest come riportato nella Tabella 1.

Modifica dei test case che riguardano i metodi *doRetrieve* come riportato nelle tabelle 2 e 3.

2.3 Test cases CR3 modificati

Test cases MaterialModelDS

Table 4: Test doSave

Metodo	doSave(MaterialBean item)
Parametri	MaterialBean item
Oggetti d'ambiente	Database
Categorie	Scelte
item	INPD: il nuovo materiale che stiamo inserendo nel database non è un materiale già esistente[property item_ok] IN: Item=null[Single]
Test frame	
INPD	ORACOLO: viene inserita una nuova tupla nella tabella Materiale del database che contiene i dati contenuti nel parametro Item con annesso riferimento alla cartella del server nella quale è contenuto il file pdf relativo a quegli appunti
IN	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché il parametro ha un riferimento a null e quindi non viene salvato nulla nel database

Table 5: Test doRetrieveByKey

Metodo	doRetrieveByKey(String code)
Parametri	code
Oggetti d'ambiente	Database

Categorie	Scelte
code	CPD: code presente nel database [property code_ok] CNPD: code non presente nel database [property code_no] CV: code="" [Single] CN: code=null [Single]
Test frame	
CPD	ORACOLO: viene restituita un'istanza di MaterialBean che contiene i dati del materiale che ha come CodiceMateriale <i>code</i>

Table 6: Test notValidated

Metodo	notValidated()
Parametri	
Oggetti d'ambiente	Database
Categorie	Scelte
Tabella materiale	TB1: la tabella ha almeno un'istanza TB2: la tabella ha 0 istanze
Test frame	
TB1	ORACOLO: viene restituita una Collection(MaterialBean)che contiene le informazioni di tutti i materiali che hanno il campo <i>Hidden</i> pari a 1

Table 7: Test doRetrieveByString

Metodo	doRetrieveByString(String str)
Parametri	str
Oggetti d'ambiente	Database
Categorie	Scelte
str	SPD: str presente nel database [property str_ok] SNPD: str non presente nel database [property str_no] SV: str="" [single] SN: str=null [single]
Test frame	
SPD	ORACOLO: viene restituita una Collection(MaterialBean)che contiene le informazioni di tutti i materiali che contengono nel campo <i>Descrizione</i> il parametro <i>str</i> oppure che contengono nel campo <i>Corso</i> il parametro <i>str</i> presenti nel database

Table 8: Test doRetrieveByUsername

Metodo	doRetrieveByUsername(String username)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
username	UPD: username presente nel database [property username_ok] UNPD: username non presente nel database [property username_no] UV: username="" [single] UN: username=null [single]
Test frame	
UPD	ORACOLO: viene restituita una Collection(MaterialBean) che contiene le informazioni di tutti i materiali caricati dall'utente che ha lo username specificato come parametro

Table 9: Test doRetrieveByParameters

Metodo	doRetrieveByParameters(String str, String ratingOrder, int rating)
Parametri	str, ratingOrder, rating
Oggetti d'ambiente	Database
Categorie	Scelte
str	ST1: str presente nel database ST2: str non presente nel database [single] ST3: str="" [single] ST4: str=null [single]
ratingOrder	RTO1: ratingOrder="ASC" RTO2: ratingOrder="DESC" RTO3: ratingOrder="novalue" RTO4: ratingOrder="" [single] RTO5: ratingOrder=null [single]
ratingOrder	RT1: $0 < \text{rating} \leq 5$ RT2: rating=0
Test frame	

ST1-RTO1-RT1	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>; • Che hanno come rating il valore del parametro <i>rating</i>; • Sono ordinati in ordine crescente rispetto al valore del <i>rating</i>.
ST1-RTO1-RT2	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>; • Sono ordinati in ordine crescente rispetto al valore del <i>rating</i>.
ST1-RTO2-RT1	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>; • Che hanno come rating il valore del parametro <i>rating</i>; • Sono ordinati in ordine decrescente rispetto al valore del <i>rating</i>.
ST1-RTO2-RT2	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>; • Sono ordinati in ordine decrescente rispetto al valore del <i>rating</i>.
ST1-RTO3-RT1	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>; • Che hanno come rating il valore del parametro <i>rating</i>.

ST1-RTO3-RT2	<p>ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni dei materiali che:</p> <ul style="list-style-type: none"> • Contengono nel campo <i>Descrizione</i> o <i>Corso</i> il valore del parametro <i>str</i>.
---------------------	---

Table 10: Test doRetrieveByOrderDate

Metodo	doRetrieveByOrderDate()
Parametri	
Oggetti d'ambiente	Database
Categorie	Scelte
Tabella materiale	<p>TB1: la tabella ha almeno un'istanza</p> <p>TB2: la tabella ha 0 istanze</p>
Test frame	
TB1	ORACOLO: viene restituita una <code>Collection<MaterialBean></code> che contiene le informazioni di tutti i materiali ordinati in modo decrescente rispetto alla data di caricamento

2.4 Test cases CR4 modificati

La change request 4 non modifica nessun test unitario.

2.5 Test cases CR5 modificati

Test cases CourseModelDSTest

Table 11: Test case doSave

Metodo	doSave(CourseBean item)
Parametri	item
Oggetti d'ambiente	Database
Categorie	Scelte
Item	<p>INPD: il corso che stiamo inserendo nel database non è un corso già esistente [property item_ok]</p> <p>IN: Item=null [Single]</p>
Test frame	

INPD	ORACOLO : viene inserita una nuova tupla nella tabella <i>Corso</i> del database che contiene i dati contenuti nel parametro <i>item</i> e sarà associata tramite chiave esterna al relativo dipartimento
IN	ORACOLO : viene generato un errore in quanto il metodo non può essere eseguito perché il parametro ha un riferimento a null e quindi non viene salvato nulla nel database

3 Nuovi unit test case

3.1 Test cases CR1 nuovi

Test cases UserModelDSTest

Table 12: Test case setBloccato()

Metodo	doUpdateBloccato(String usernameOrEmail, Date bloccato)
Parametri	usernameOrEmail, date
Oggetti d'ambiente	Database
Categorie	Scelte
Utente	M1: usernameOrEmail presente nel database M2: usernameOrEmail non presente nel database M3: usernameOrEmail=null [single] M4: usernameOrEmail="" [single]
Data	DV: data valida DNV: data null [single]
Test frame	
M1-DV	ORACOLO: viene settato il campo bloccato dell'utente con il valore contenuto nel parametro <i>date</i>
M1-DNV	ORACOLO: viene lanciata un'eccezione di <i>IllegalArgumentException</i> .
M2	ORACOLO: viene lanciata una <i>SQLException</i> in quanto non esiste un utente con quella mail
M3	ORACOLO: viene lanciata una <i>IllegalArgumentException</i> in quanto la mail non è valida
M4	ORACOLO: viene lanciata una <i>IllegalArgumentException</i> in quanto la mail non è valida

3.2 Test cases CR2 nuovi

Test cases UserModelDSTest tabelle doRetrieveAll

Table 13: Test case setVerificato()

Metodo	doUpdateVerificato(String mail, boolean value)
Parametri	mail
Oggetti d'ambiente	Database
Categorie	Scelte
Utente	M1: mail presente nel database M2: mail non presente nel database M3: mail=null [single] M4: mail="" [single]
Test frame	
M1	ORACOLO: viene settato il campo verificato dell'utente con il valore contenuto nel parametro <i>value</i>
M2	ORACOLO: viene lanciata una <i>SQLException</i> in quanto non esiste un utente con quella mail
M3	ORACOLO: viene lanciata una <i>IllegalArgumentException</i> in quanto la mail non è valida
M4	ORACOLO: viene lanciata una <i>IllegalArgumentException</i> in quanto la mail non è valida

3.3 Test cases CR3 nuovi

La change request 3 non crea nessun nuovo test unitario.

3.4 Test cases CR4 nuovi

La change request 4 non crea nessun nuovo test unitario.

3.5 Test cases CR5 nuovi

Test cases CourseModelDSTest

Table 14: Test case doRetriveAll()

Metodo	doRetriveAll()
Parametri	-
Oggetti d'ambiente	Database
Categorie	Scelte
Tabella Corso	TCP: almeno una tupla presente nella tabella Corso TCNP: nessuna tupla presente nella tabella Corso
Test frame	

TCP	ORACOLO: viene restituita una lista di CourseBean che contiene tutti i corsi presenti nel database
TCNP	ORACOLO: viene restituita lista di CourseBean vuota perché non sono presenti corsi nel database

Test cases InteressiModelIDSTest

Table 15: Test case doRetriveByUsername()

Metodo	doRetriveByUsername(String username)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
Username	UP: username presente nel database UNP: username non presente nel database UN: username=null UE: username=""
Test frame	
UP	ORACOLO: viene restituita una lista di InteressiBean che contiene tutti gli interessi dell'utente che ha come username il valore del parametro in input
UNP	ORACOLO: viene restituita lista di CourseBean vuota in quanto lo username non è presente nel database
UN	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido
UE	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido

Table 16: Test case doRetriveByCorso()

Metodo	doRetriveByCorso(int codiceCorso)
Parametri	codiceCorso
Oggetti d'ambiente	Database
Categorie	Scelte
Corso	CP: codiceCorso presente nel database CNP: codiceCorso non presente nel database
Test frame	
CP	ORACOLO: viene restituita una lista di InteressiBean che contiene tutti gli utenti interessati a quel corso

CNP	ORACOLO: viene restituita lista di CourseBean vuota in quanto non esiste un corso con quel codice
------------	---

Table 17: Test case doSave()

Metodo	doSave(InteressiBean item)
Parametri	item
Oggetti d'ambiente	Database
Categorie	Scelte
Item	INP: item non presente nel database IP: item presente nel database IN: item non inserito
Test frame	
INP	ORACOLO: viene inserita una nuova tupla nella tabella <i>Interessi</i> del database che contiene i dati contenuti nel parametro item
IP	ORACOLO: viene generato un errore in quanto la tupla esiste già nel database
IN	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché il parametro ha un riferimento a null e quindi non viene salvato nulla nel database

Table 18: Test case doDelete()

Metodo	doDelete(String username, int codiceCorso)
Parametri	username, codiceCorso
Oggetti d'ambiente	Database
Categorie	Scelte
Utente	UP: username presente nel database UNP: username non presente nel database UN: username=null [single] UE: username="" [single]
Corso	CP: corso presente nel database CNP: corso non presente nel database
Test frame	
UP-CP	ORACOLO: viene cancellata la tupla nella tabella <i>Interessi</i> che presenta i dati contenuti nei parametri
UP-CNP	ORACOLO: viene generato un errore in quanto la tupla non esiste nel database

UNP-CP	ORACOLO: viene generato un errore in quanto la tupla non esiste nel database
UN-CP	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché il parametro <i>username</i> ha un riferimento a null
UE-CP	ORACOLO: viene generato un errore in quanto il metodo non può essere eseguito perché il parametro <i>username</i> è vuoto
UNP-CNP	ORACOLO: viene generato un errore in quanto la tupla non esiste nel database

Table 19: Test case doRetriveNewInteressi()

Metodo	doRetriveNewInteressi(String username)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
Username	UP: username presente nel database UN: username=null UE: username=""
Test frame	
UP	ORACOLO: viene restituita una lista di CourseBean che contiene tutti i corsi per il quale l'utente non ha espresso interessi
UN	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido
UE	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido

Table 20: Test case getInteressi()

Metodo	getInteressi(String username)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
Username	UP: username presente nel database UNP: username senza interessi UN: username=null UE: username=""
Test frame	

UP	ORACOLO: viene restituita una lista di CourseBean che contiene tutti i corsi per il quale l'utente ha espresso interessi
UNP	ORACOLO: viene restituita lista di CourseBean vuota in quanto lo username non è presente nel database
UN	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido
UE	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido

Table 21: Test case getInteressi()

Metodo	doRetrieveMaterialByInteressi(String username)
Parametri	username
Oggetti d'ambiente	Database
Categorie	Scelte
Username	UP: username presente nel database UNP: username senza interessi UN: username=null UE: username=""
Test frame	
UP	ORACOLO: viene restituita una lista di MaterialBean che contiene al più tre materiali per gli ultimi tre interessi inseriti
UNP	ORACOLO: viene restituita lista di CourseBean vuota in quanto lo username non è presente nel database
UN	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido
UE	ORACOLO: viene lanciata l'eccezione <i>IllegalArgumentException</i> in quanto il parametro non è valido