Università degli Studi di Salerno

Corso di Ingegneria del Software

UniClass
Test Plan and Case Specification
Versione 1.0

Data: 16/12/2024

Progetto: UniClass	Versione: 1.0
Documento: Test Plan and Case Specification	Data: 16/12/2024

Coordinatore del progetto:

Nome	**Matricola**
Giuseppe Sabetta	0512117895

Partecipanti:

Nome	**Matricola**
Giuseppe Sabetta	0512117895
Sara Gallo	0512117262
Saverio D'Avanzo	0512118330
Gerardo Antonio Cetrulo	0512117856

Scritto da:	Giuseppe Sabetta (GS), Sara Gallo (SG), Saverio D'Avanzo (SD), Gerardo Antonio Cetrulo (AC)

Revision History

Data	**Versione**	**Descrizione**	**Autore**
16/12/2024	1.0	Test Plan and Case Specification	GS, SG, SD, AC

Sommario

1. Introduzione

Il testing rappresenta una fase cruciale nello sviluppo del progetto UniClass, poiché UniClass si propone di fornire un'esperienza utente ottimale e intuitiva, è essenziale garantire che tutte le

funzionalità principali siano accuratamente testate per assicurare stabilità, affidabilità e usabilità del sistema.
L'obiettivo principale del testing è individuare eventuali malfunzionamenti, garantendo che ogni componente dell'applicazione
rispetti i requisiti di progetto e offra un'esperienza coerente, fluida e priva di bug.
Inoltre, il testing sarà orientato a verificare che le scelte progettuali in termini di trade-off
(flessibilità, manutentibilità, scalabilità e ottimizzazione della memoria) non compromettano l'usabilità e la performance complessiva.

2. Relazione con altri documenti

Il Relazione con altri documenti
Il documento attuale fa riferimento:

- ***Requirements Analysis Document (RAD)***
- ***System Design Document (SDD)***
- ***Object Design Document (ODD)***

3. Feature da Testare

Per la sistema UniClass saranno messo in atto un testing dei requisiti funzionali e non funzionali, dando priorità alle funzionalità in cui è richiesto un input libero. Le funzionalità che saranno particolarmente sotto test sono le seguenti:

Autenticazione

- *Autenticazione*

Gestione Account

- *Creazione dell'account UniClass*
- *Rimozione dell'account UniClass*

Agenda

- *Aggiunta di una lezione alla proprio agenda*
- *Rimozione di una lezione dalla propria agenda*

Messaggistica

- *Invio dei messaggi lato studente*
- *Invio dei messaggi lato docente*
- *Invio messaggio broadcast da parte del Docente*
- *Invio messaggio broadcast da parte del Coordinatore*

Appelli

- *prenotazione appello*

Lezioni

- *Creazione Lezione*
- *Rimozione Lezione*

4. Criteri di successo/insuccesso

I criteri per determinare il successo o l'insuccesso del piano di test si basano sulla conformità ai requisiti sia funzionali che non funzionali del sistema. Un test si considera passato se i risultati ottenuti corrispondono esattamente a quelli previsti nei casi di test e se non vengono rilevati eventuali difetti che possano compromettere le altre funzionalità. In caso di successo, tutte le funzionalità verificate devono rispettare i criteri stabiliti, i test di performance devono soddisfare i livelli accettati, e non devono emergere errori bloccanti o di alta priorità durante l'esecuzione. In caso di insuccesso, si valutano come critici i risultati che non coincidono con quelli previsti, la presenza di bug bloccanti o critici, e il mancato rispetto dei requisiti non funzionali, come tempi di risposta oltre i limiti o errori in condizioni di stress.

5. Approccio

L'approccio al testing adottato per questo progetto combina due metodologie chiave: il testing di unità e il testing di integrazione. Queste tecniche sono fondamentali per assicurare che ogni componente del sistema funzioni correttamente sia in isolamento che in interazione con altri componenti.

5.1. Approccio Combinato (Sandwich)

Adotteremo una strategia ibrida che combina i metodi di testing Bottom-Up e Top-Down, attraverso l'uso di driver e stub per coprire tutte le necessità di test del sistema:

5.1.1. Bottom-Up

Fase 1: Testing delle classi Java individuali a livello unitario. Questo processo coinvolge:
Verifica che ogni classe Java funzioni correttamente in isolamento.
Utilizzo di test unitari per garantire che ogni metodo all'interno delle classi Java risponda correttamente agli input previsti. Questo tipo di test si concentra su funzionalità specifiche come la correttezza della logica di business, la gestione degli errori, e la robustezza dei metodi.

5.1.2. Top-Down

Fase 2: Dopo aver verificato le classi Java di base, si procede con un approccio Top-Down:
Testing delle JSP: Si testeranno le JavaServer Pages per assicurare che l'interfaccia utente funzioni come previsto, includendo la verifica dell'aspetto visuale, della navigazione e della responsività.
Integrazione tra JSP e Servlet: Si testerà l'interazione tra le JSP e le Servlet per garantire che le richieste dell'interfaccia utente invochino correttamente la logica applicativa delle Servlet.
Integrazione con Classi Java: Questo include il testing della comunicazione tra Servlet e le classi Java che sono state già testate nella fase Bottom-Up, verificando che il flusso di dati e le chiamate ai metodi siano integrati correttamente.

5.2. Sequenza di Testing

Unit Testing: Prima di tutto, si eseguono i test unitari per le singole classi Java. Questo assicura che ogni componente fondamentale del sistema funzioni come previsto in isolamento.
Integration Testing: Successivamente, si procede con l'integrazione dei test che esaminano come le classi Java interagiscono tra di loro. Qui si verifica che le interfacce tra i componenti siano correttamente implementate.
System Testing: Infine, si effettua il testing di sistema che coinvolge l'interfaccia JSP con le Servlet e le classi Java sottostanti, per garantire che l'intero sistema funzioni in modo integrato e soddisfi i requisiti complessivi.

Questa sequenza di testing garantisce una copertura completa, dall'unità più piccola fino al sistema intero, assicurando una qualità elevata del prodotto finale.

6. Sospensione e ripristino

Poiché il test verrà realizzato su singole funzionalità, potrà essere portato a termine senza alcuna interruzione. Questo approccio consente di garantire una valutazione continua e fluida delle varie caratteristiche.

7. Materiale di Testing

Per coprire le funzionalità di testing descritte fin ora, Si utilizzerà un computer condiviso, un dispositivo multifunzionale che soddisfa varie necessità, permettendo a tutti di accedere alle sue capacità tecnologiche per le proprie attività.

8. Test Case

8.1. TC_01

Questa specifica dei casi di test riguarda i casi d'uso relativi all'autenticazione dell'utente

- Parametri:
  - Email (il suffisso può essere solo "studenti.unisa.it" oppure "unisa.it")
  - Password (deve contenere caratteri uppercase, lowercase, numeri e caratteri speciali come i seguenti: @, #, $, €, &, %)
- Oggetti dell'ambiente: Database

Parametro: Email	
Categorie	*Valore*
Formato	1. *Corretto [property EFCorretto]* 2. *Non Corretto [property EFNCorretto]*

Database	1. *Presente [property EFDB][if EFCorretto]* 2. *Non presente*

Parametro: Password	
Categorie	*Valore*
Formato	1. *Corretto [property PFCorretto]* 2. *Non Corretto [property PFNCorretto]*
Associazione	1. *Associata ad un'email nel database [if EFCorretto AND EFDB AND PFCorretto]* 2. *Non associato ad un'email nel database*

Dettagli della notazione
EF1: Email formato corretto
EF2: Email formato incorretto
ED1: Email presente nel database
ED2: Email non presente nel database
PF1: Password formato corretto
PF2: Password formato incorretto
PA1: Password associata nel database
PA2: Password non associata nel database

Test frame relativi a TC1

EF1, ED1, PF1, PA1
Oracolo: Email formato corretto, presente nel database, password formato corretto, associata nel database.

EF1, ED1, PF2, PA2
Oracolo: Email formato corretto, presente nel database, password formato incorretto, non associata nel database.

EF1, ED1, PF1, PA2
Oracolo: Email formato corretto, presente nel database, password formato corretto, non associata nel database.

EF1, ED2, PF1, PA2
Oracolo: Email formato corretto, non presente nel database, password formato corretto, non associata nel database.

EF1, ED2, PF2, PA2
Oracolo: Email formato corretto, non presente nel database, password formato incorretto, non associata nel database.

EF2, ED2, PF1, PA2
Oracolo: Email formato incorretto, non presente nel database, password formato corretto, non associata nel database.

EF2, ED2, PF2, PA2
Oracolo: Email formato incorretto, non presente nel database, password formato incorretto, non associata nel database.

EF1, ED1, PF1, PA1

Oracolo: Email formato corretto, presente nel database, password formato corretto, associata nel database.
EF1, ED1, PF2, PA2

Oracolo: Email formato corretto, presente nel database, password formato incorretto, non associata nel database.
EF1, ED1, PF1, PA2

Oracolo: Email formato corretto, presente nel database, password formato corretto, non associata nel database.
EF1, ED1, PF2, PA1

Oracolo: Email formato corretto, presente nel database, password formato incorretto, associata nel database.
EF1, ED2, PF1, PA2

Oracolo: Email formato corretto, non presente nel database, password formato corretto, non associata nel database.
EF1, ED2, PF2, PA2

Oracolo: Email formato corretto, non presente nel database, password formato incorretto, non associata nel database.

9. Test Case Specification

9.1. Test Case n°1

Specifica dei casi di test per i casi d'uso relativi all'autenticazione.

TC1_01
Pre-condizione: L'utente non è autenticato.
(Nel database è presente email: mario.rossi2@studenti.unisa.it password: Prova1234).

Categoria	***Scelta***
Email	mario.rossi2@studenti*.unisa.it*
Password	*Prova1234*

ORACOLO: Utente autenticato e reindirizzato alla homepage della piattaforma.

TC1_02
Pre-condizione: L'utente non è autenticato.

Categoria	***Scelta***
Email	mario.rossi2@studenti
Password	*Prova1234*

ORACOLO: Il sistema visualizza un messaggio di errore: “Attenzione: formato email errato.”

TC1_03
Pre-condizione: L'utente non è autenticato.
(Nel database non è presente email: maria.bianchi@studenti.unisa.it password: Prova1234).

Categoria	***Scelta***
Email	maria.bianchi@studenti.unisa.it
Password	*Prova1234*

ORACOLO: Il sistema visualizza un messaggio di errore: “Attenzione: mail e/o password errate”

TC1_04
Pre-condizione: L'utente non è autenticato.
(Nel database è presente email: mario.rossi2@studenti.unisa.it password: Prova1234).

Categoria	***Scelta***
Email	mario.rossi2@studenti.unisa.it
Password	*Prova1233*

ORACOLO: Il sistema visualizza un messaggio di errore: “Attenzione: mail e/o password errate”