



UNIVERSITÀ DEGLI STUDI
DI SALERNO

UNISA

TEST PLAN HAMPLANET.BLOG

Partecipanti:

Danese Vincenzo 0512107959

Nappi Nicola 0512109534

Russo Giovanni 0512118684

Vitiello Catello 0512110338

Presentato a:

Prof. Andrea De Lucia

Dipartimento di informatica

versione 1.240626 | 27 giugno 2024

Indice

1 INTRODUCTION	2
2 RELATIONSHIP TO OTHER DOCUMENTS	2
3 SYSTEM OVERVIEW	2
4 FEATURES TO BE TESTED / NOT TO BE TESTED	2
5 PASS / FAIL CRITERIA	2
6 APPROACH	3
6.1 UNIT TESTING	3
6.2 INTEGRATION TESTING	3
6.3 SYSTEM TESTING	3
7 SUSPENSION & RESUMPTION	3
8 TESTING MATERIALS	3
9 TEST CASES	4
10 TESTING SCHEDULE	19

1 INTRODUCTION

Nel documento pianifichiamo il testing della piattaforma HamPlanet.blog specificando le funzionalità da testare e i relativi test cases. Descriviamo inoltre come viene eseguito il testing del sistema.

2 RELATIONSHIP TO OTHER DOCUMENTS

Nel documento facciamo riferimento al Requirements Analysis Document (RAD) dove sono riportati gli Use Cases delle funzionalità da testare.

3 SYSTEM OVERVIEW

HamPlanet.blog è un blog di carattere culinario. Nel System Design Document (SDD) abbiamo individuato cinque sottosistemi:

Gestione Utente, Gestione Post, Gestione Navigazione, Gestione Commento.

S'intende testare l'applicazione nella sua interezza, coinvolgendo nel testing tutti i sottosistemi elencati.

4 FEATURES TO BE TESTED / NOT TO BE TESTED

- Gestione Utente
 - Registrazione
 - Login
 - Logout
 - Modifica profilo
 - Visualizzazione profilo
 - Eliminazione profilo
- Gestione Post
 - Creazione Post
 - Eliminazione Post
 - Like Post
 - Unlike Post
- Gestione Commento
 - Creazione Commento
 - Eliminazione Commento
- Gestione Navigazione
 - Visualizza Pagina Contenuto (Post, Home, Utente, ContentWriter)
 - Navigazione Pagine

5 PASS / FAIL CRITERIA

I test cases vengono definiti in modo da individuare il maggior numero di fault presenti all'interno del nostro sistema. Un test case viene quindi considerato di successo se in grado di rilevare dei fault, se questi sono presenti. I fault rilevati saranno documentati e corretti.

6 APPROACH

La strategia di testing che adottiamo è suddivisa in tre fasi principali:

6.1 UNIT TESTING

Inizialmente, ci focalizziamo sulle singole componenti del sistema andandole a testare in isolamento. In questa fase del testing utilizziamo la tecnica blackbox, ovvero ricaviamo i test cases necessari partendo dalle specifiche contenute nel RAD, in particolare ci soffermiamo sugli Use Cases. L'esecuzione dei test viene effettuata adoperando il framework JUnit.

6.2 INTEGRATION TESTING

Con l'integration testing testiamo le componenti aggregate in gruppi, aumentandone progressivamente la dimensione. Adottiamo la strategia bottom-up provvedendo a testare le servlet integrando i DAO già testati nello Unit Testing, in seguito verificiamo che il layer di presentazione visualizzi correttamente le informazioni ottenute dal layer della logica di business.

6.3 SYSTEM TESTING

Nella fase finale testiamo il comportamento dell'intero sistema verificando che quest'ultimo rispetti i requisiti funzionali, non funzionali e design goals individuati in precedenza nel RAD e nell'SDD. I test cases vengono selezionati attraverso la tecnica del category-partition, la quale ci permette di testare un gran numero di funzionalità con un numero ridotto di test cases. I test vengono eseguiti mediante Selenium.

7 SUSPENSION & RESUMPTION

I test previsti non sono così complessi o estesi a tal punto da necessitare delle strategie di sospensione.

8 TESTING MATERIALS

L'esecuzione dei test richiede un computer con il nostro sistema installato e sul quale poter usufruire di Selenium

9 TEST CASES

T1: Registrazione

Parametri	Email, password, username
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none"> • C1: email valida (rispetta il formato della specifica) • C2: email presente nel database • C3: password valida (rispetta il formato della specifica) • C4: username valida (rispetta il formato della specifica)
SCELTE E VINCOLI RELATIVI	<p>Scelte per C1:</p> <ul style="list-style-type: none"> • EV1: email valida [property email_valida] • EV2: email NON valida <p>Scelte per C2:</p> <ul style="list-style-type: none"> • EP1: email presente nel database [if email_valida] • EP2: email NON presente nel database [property email_non_presente] <p>Scelte per C3:</p> <ul style="list-style-type: none"> • PV1: password valida [property passowrd_valida] • PV2: password non valida [if email_valida and email_non_presente] <p>Scelte per C4:</p> <ul style="list-style-type: none"> • UV1: username valida [property username_valida] • UV2: username non valida [if email_valida and email_non_presente and password_valida]

TEST FRAME	<ul style="list-style-type: none"> • TF1: EV1, EP1, PV1, UV1 ORACOLO: Messaggio di errore accanto al campo email: “email già utilizzata” • TF2: EV1, EP2, PV1, UV1 ORACOLO: Registrazione effettuata con successo • TF3: EV1, EP2, PV2, UV1 ORACOLO: Messaggio di errore accanto al campo password: “password non rispetta il formato richiesto” • TF4: EV1, EP2, PV1, UV2 ORACOLO: Messaggio di errore accanto al campo username: “username non rispetta il formato richiesto” • TF5: EV2, EP2, PV1, UV1 ORACOLO: Messaggio di errore accanto al campo email: “email non rispetta il formato richiesto”
TEST CASE	<ul style="list-style-type: none"> • TC1: : email = “n@n.it”, password = “nappi”, username = “Nicola_Nappi” – In DB presente la coppia (email = “n@n.it”, password = “nappi”) – ORACOLO: La pagina di registrazione mostra il messaggio: “email già utilizzata” • TC2: : email = “lillo@gmail.com”, password = “posaman44”, username = “Posaman” – In DB NON presente la coppia (email = “lillo@gmail.com”, password = “posaman”) – ORACOLO: Utente registrato con successo e ridiretto alla sua homepage • TC3: email = “lillo@gmail.com”, password = “1234”, username = “Posaman” – In DB NON presente la coppia (email = “lillo@gmail.com”, password = “1234”) – ORACOLO: La pagina di registrazione mostra il messaggio: “password non rispetta il formato richiesto” • TC4: email = “lillo@gmail.com”, password = “posaman44”, username = “30-1” – In DB NON presente la coppia (email = “lillo@gmail.com”, password = “posaman44”) – ORACOLO: La pagina di registrazione mostra il messaggio: “username non rispetta il formato richiesto” • TC5: : email = “artman”, password = “gundalf31”, username = “Anello” – ORACOLO: La pagina di registrazione mostra il messaggio: “email non rispetta il formato richiesto”

T2: Login

Parametri	Email, password
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none"> • C1: email presente nel DB • C2: password corretta in relazione a quella presente nel DB in relazione all' email
SCELTE E VINCOLI RELATIVI	<p>Scelte per C1:</p> <ul style="list-style-type: none"> • EV1: email presente nel DB[property email_presente] • EV2: email NON presente nel DB <p>Scelte per C2:</p> <ul style="list-style-type: none"> • PP1: password associata all'email nel DB [if email_presente] • PP2: password non corretta rispetto a quella associata all'email nel DB
TEST FRAME	<ul style="list-style-type: none"> • TF1: EV1, PP1 ORACOLO: Utente autenticato e ridiretto alla sua homepage • TF2: EV1, PP2 ORACOLO: Messaggio di errore: "Credenziali errate" • TF3: EV2, PP2 ORACOLO: Messaggio di errore: "Credenziali errate"
TEST CASE	<ul style="list-style-type: none"> • TC1: : email = "n@n.it", password = "nappi" – In DB presente la coppia (email = "n@n.it",password = "nappi") – ORACOLO: Utente autenticato con successo e ridiretto alla sua homepage • TC2: : email = "n@n.it", password = "341455" – In DB presente la coppia (email = "n@n.it", password = "nappi") – ORACOLO: La pagina di autenticazione mostra il messaggio: "credenziali errate" • TC3: email = "gino@gmail.com", password = "gino14" – In DB NON presente la coppia (email = "gino@gmail.com", password = "gino14") –ORACOLO: La pagina di autenticazione mostra il messaggio: "credenziali errate"

T3: logout

Parametri	-
OGGETTI DELL'AMBIENTE	-
CATEGORIE	<ul style="list-style-type: none">● C1: Utente disconnesso
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">● UD1: utente disconnesso con successo
TEST FRAME	<ul style="list-style-type: none">● TF1: UD1 ORACOLO: L'utente viene ridiretto alla pagina di login
TEST CASE	<ul style="list-style-type: none">● TC1: : _ - _ - ORACOLO: In seguito al click sul pulsante di logout l'utente viene ridiretto alla pagina di login

T4: Modifica Profilo

Parametri	password, username
OGGETTI DELL'AMBIENTE	-
CATEGORIE	<ul style="list-style-type: none"> • C1: password valida (rispetta il formato della specifica) • C2: username valida (rispetta il formato della specifica)
SCELTE E VINCOLI RELATIVI	<p>Scelte per C1:</p> <ul style="list-style-type: none"> • PV1: password valida [property password_valida] • PV2: password NON valida <p>Scelte per C2:</p> <ul style="list-style-type: none"> • UV1: username valida [property username_valida] • UV2: username NON valida [id password_valida]
TEST FRAME	<ul style="list-style-type: none"> • TF1: PV1, UV1 ORACOLO: Modifiche effettuate con successo • TF2: PV1, UV2 ORACOLO: Messaggio di errore accanto al campo username: "username non rispetta il formato richiesto" • TF3: PV2, UV1 ORACOLO: Messaggio di errore accanto al campo password: "password non rispetta il formato richiesto"
TEST CASE	<ul style="list-style-type: none"> • TC1: password = "leclerc16", username = "goCharles" – ORACOLO: Modifiche effettuate con successo • TC2: password = "RedOne5", username = "521" – ORACOLO: Messaggio di errore accanto al campo username: "username non rispetta il formato richiesto" • TC3: password = "5", username = "Monaco2024" – ORACOLO: Messaggio di errore accanto al campo password: "password non rispetta il formato richiesto"

T5: Visualizza Pagina Profilo

Parametri	id
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">• C1: profilo associato all'id presente nel database
SCelte E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">• PP1: id presente nel database• PP2: id NON presente nel database
TEST FRAME	<ul style="list-style-type: none">• TF1: PV1, UV1 ORACOLO: Visualizzazione pagina con le informazioni relative al profilo• TF2: PV1, UV2 ORACOLO: Messaggio di errore: "Profilo non trovato"
TEST CASE	<ul style="list-style-type: none">• TC1: id = 5 – Su DB presente contenuto associato – ORACOLO: Visualizzazione informazioni profilo con id = 5• TC2: id = 16 – Su DB NON presente contenuto associato – ORACOLO: Messaggio di errore: "Profilo non trovato"

T6: Eliminazione Profilo

Parametri	id
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">● C1: profilo con id fornito viene eliminato da DB
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">● PE1: profilo eliminato con successo
TEST FRAME	<ul style="list-style-type: none">● TF1: PE1 ORACOLO: le informazioni relative al profilo vengono eliminate da DB e l'utente viene ridiretto alla pagina iniziale
TEST CASE	<ul style="list-style-type: none">● TC1: id = 4 – _ – ORACOLO: Dati relativi al profilo con id 4 eliminati da DB, l'utente si trova sulla pagina iniziale

T7: Mettere like ad un post

Parametri	id_post
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">• C1: Mettere like ad un post
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">• ML1: Mettere like ad un post
TEST FRAME	<ul style="list-style-type: none">• TF1: ML1 ORACOLO: L'utente ha messo like al post associato
TEST CASE	<ul style="list-style-type: none">• TC1: id_post = 5 – Su DB presente post con id=5 a cui l'utente non ha messo like – ORACOLO: Viene creata l'associazione tra l'utente e il post, che ora sarà visibile sul post. L'utente viene notificato con il messaggio “hai messo like al post”

T8: Togliere like ad un post

Parametri	id_post
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">• C1: Togliere like ad un post
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">• TL1: Togliere like ad un post
TEST FRAME	<ul style="list-style-type: none">• TF1: TL1 ORACOLO: L'utente ha tolto like al post associato e non sarà più visibile sul post
TEST CASE	<ul style="list-style-type: none">• TC1: id_post = 1 – Su DB presente post con id=1 a cui l'utente ha messo like – ORACOLO: Viene eliminata l'associazione tra l'utente e il post, che non sarà visibile sul post. L'utente viene notificato con il messaggio “hai tolto il like al post”

T9: Creazione post

Parametri	Titolo, Contenuto, Immagine
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none"> • C1: Titolo unico Post • C2: Immagine rispetta il formato richiesto • C3: Testo del post
SCelte E VINCOLI RELATIVI	<p>Scelte per C1:</p> <ul style="list-style-type: none"> • TU1: Titolo unico del post [property titolo_unico] • TU2: Titolo del post già utilizzato <p>Scelte per C2:</p> <ul style="list-style-type: none"> • IR1: Immagine rispetta il formato [property formato_valido] • IR2: Immagine NON rispetta il formato [if titolo_unico] <p>Scelte per C3:</p> <ul style="list-style-type: none"> • TP1: Testo contiene almeno una parola • TP2: Testo non contiene nulla [if titolo_unico and formato_valido]
TEST FRAME	<ul style="list-style-type: none"> • TF1: TU1, IR1, TP1 ORACOLO: Creazione post effettuata con successo, il content_writer viene reindirizzato alla home • TF2: TU2, IR1, TP1 ORACOLO: Messaggio di errore: "Titolo già utilizzato" • TF2: TU1, IR2, TP1 ORACOLO: Messaggio di errore: "Formato immagine non valido" • TF2: TU1, IR1, TP2 ORACOLO: Messaggio di errore: "Il testo non può essere vuoto"

TEST CASE	<ul style="list-style-type: none"> • TC1: titolo=“Carne rossa o carne bianca?”, immagine=“carne.jpg” valida, testo=“viva la carne bianca” – Su DB NON è presente un post con titolo “Carne rossa o carne bianca?” – ORACOLO: “Creazione post eseguita con successo” • TC2: titolo=“Carne rossa o carne bianca?”, immagine=“carne.jpg” valida, testo=“viva la carne bianca” – Su DB è presente un post con titolo “Carne rossa o carne bianca?” – ORACOLO: “Titolo già utilizzato” • TC3: titolo=“pane”, immagine=“acd.mp3” non valida, testo=“pane per sempre” – Su DB NON è presente un post con titolo “Carne rossa o carne bianca?” – ORACOLO: “Formato immagine non valido” • TC4: titolo=“pane”, immagine=“acd.jpg” valida, testo=“” – Su DB NON è presente un post con titolo “Carne rossa o carne bianca?” – ORACOLO: “Il testo non può essere vuoto”
-----------	--

T10: Eliminazione post

Parametri	id
OGGETTI DELL'AMBIENTE	Database, FileSystem
CATEGORIE	<ul style="list-style-type: none">• C1: Post con id fornito viene eliminato da DB e FS
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">• PE1: Post eliminato con successo
TEST FRAME	<ul style="list-style-type: none">• TF1: PE1 ORACOLO: le informazioni relative al post vengono eliminate da DB e FS e l'utente viene ridiretto alla pagina iniziale
TEST CASE	<ul style="list-style-type: none">• TC1: : id = 21 – _ – ORACOLO: Dati relativi al post con id 21 eliminati da DB e FS, l'utente si trova sulla pagina iniziale

T11: Eliminazione commento

Parametri	id
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">• C1: Commento con id fornito viene eliminato dal DB
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">• CE1: Commento eliminato con successo
TEST FRAME	<ul style="list-style-type: none">• TF1: CE1 ORACOLO: le informazioni relative al commento vengono eliminate dal DB la pagina del post viene ricaricata
TEST CASE	<ul style="list-style-type: none">• TC1: : id = 44 – _ – ORACOLO: Dati relativi al commento con id 44 viene eliminato dal DB l'utente si trova sulla pagina del post ricaricata

T12: Eliminazione commento

Parametri	commento
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">● C1: testo commento con formato valido
SCelte E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">● TC1: Testo commento rispetta il formato [property formato_valido]● TC2: Testo NON valido
TEST FRAME	<ul style="list-style-type: none">● TF1: TC1 ORACOLO: Creazione del commento correttamente avvenuta● TF1: TC2 ORACOLO: Messaggio di errore: "Commento non rispetta il formato"
TEST CASE	<ul style="list-style-type: none">● TC1: : commento="ciao" valido Oracolo: creazione del commento avvenuta e caricato sul DB● TC2: : commento="" non valido Oracolo: Messaggio di errore "FOrmato commento non valido"

T13: Visualizzazione Post

Parametri	id
OGGETTI DELL'AMBIENTE	Database
CATEGORIE	<ul style="list-style-type: none">● C1: Post associato all'id presente nel Database
SCELTE E VINCOLI RELATIVI	Scelte per C1 : <ul style="list-style-type: none">● PP1: Id presente● PP2: Id NON presente
TEST FRAME	<ul style="list-style-type: none">● TF1: TC1 ORACOLO: Visualizzazione post● TF1: TC2 ORACOLO: Messaggio di errore: "Post non trovato"
TEST CASE	<ul style="list-style-type: none">● TC1: id=3 – Su DB presente contenuto associato – ORACOLO: Visualizzazione post con id = 3● TC2: id = 55 – Su DB NON presente contenuto associato – ORACOLO: Messaggio di errore: "Post non trovato"

10 TESTING SCHEDULE

Al momento non prevediamo un'agenda per lo svolgimento del testing che verrà eseguito al termine dello sviluppo del software o delle componenti da sottoporre ai singoli test cases.