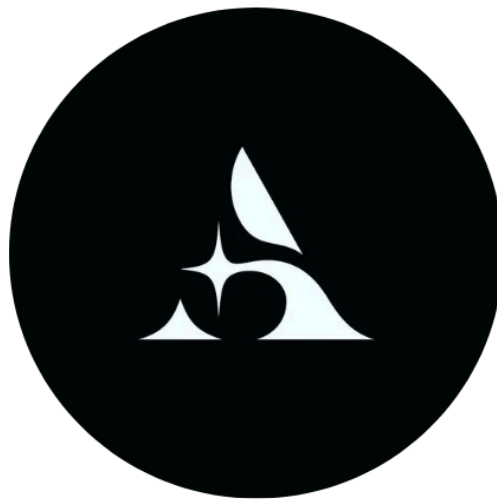




Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof. C. Gravino

Test Plan

AstroVerse





Riferimento	NC31 Test Plan
Versione	2.0
Data	17/12/2024
Destinatario	Professore C. Gravino
Presentato da	Gruppo NC31



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software* - Prof. C. Gravino

Revision History

Data	Versione	Descrizione	Autori
17/12/2024	0.1	Prima stesura	Christian Fontana, Pellegrino Piccolo
17/12/2024	0.2	Test Cases	Christian Bianco
19/12/2024	1.0	Prima Revisione	Adriano De Vita
21/12/2024	2.0	Revisione Pre-release	Tutto il team



Team members

Nome	Ruolo progettuale	Contatto
Pellegrino Piccolo	Membro del team	p.piccolo4@studenti.unisa.it
Adriano De Vita	Membro del team	a.devita40@studenti.unisa.it
Christian Fontana	Membro del team	c.fontana7@studenti.unisa.it
Christian Bianco	Membro del team	c.bianco9@studenti.unisa.it



Sommario

Revision History.....	3
Team members.....	4
Sommario.....	5
1. Introduzione.....	6
2. Relazione con altri documenti.....	6
3. Panoramica del sistema.....	7
4. Features da testare/non testare.....	7
5. Pass/Fail Criteria.....	8
6. Approccio.....	8
7. Sospensione e ripristino.....	10
8. Materiale di testing.....	11
9. Test Cases.....	11
9.1 Gestione Utente.....	11
9.1.1 Modifica dati utente.....	11
9.1.2 Login.....	14
9.1.3 Registrazione.....	15
9.2 Gestione Spazio.....	18
9.2.1 Creazione Spazio.....	18
9.2.2 Modifica Spazio.....	20
9.3 Gestione Post.....	22
9.3.1 Creazione Post.....	22
9.3.2 Modifica Post.....	23



1. Introduzione

Il Test Plan ha lo scopo di definire e analizzare le attività di testing per la piattaforma AstroVerse. L'obiettivo principale è assicurare che ogni componente della piattaforma operi correttamente. Il documento descrive le strategie di testing utilizzate, le funzionalità sottoposte a verifica e gli strumenti adottati per individuare eventuali errori, con l'intento di consegnare al cliente finale una piattaforma completamente funzionante e priva di anomalie.

Sono state pianificate le attività di testing per le seguenti gestioni (quelle effettivamente in fase di sviluppo):

- Gestione Utente
- Gestione Spazio
- Gestione Post

2. Relazione con altri documenti

Per identificare correttamente i test case, si fa riferimento ad altri documenti sviluppati:

Collegamenti con il Requirements Analysis Document (RAD)

I test case previsti nel Test Plan sono formulati sulla base dei requisiti funzionali e non funzionali definiti nel RAD.

Collegamenti con il System Design Document (SDD)

I test case previsti nel Test Plan devono seguire la suddivisione in sottosistemi delineata nel SDD.



3. Panoramica del sistema

Il sistema proposto basa la sua architettura sulla tipologia Client-Server in particolare, verrà utilizzato Vue per la parte di front-end mentre per la logica applicativa e quindi il backend sarà utilizzato il framework di Java, Spring Boot. Per la gestione del database e dei dati persistenti verranno utilizzate le Spring JPA.

4. Features da testare/non testare

Di seguito si riporta la lista delle features da implementare di cui si effettuerà il testing per varie gestioni:

- Gestione Utente
 - Modifica Dati Utente
 - Login
 - Registrazione
- Gestione Spazio
 - Creazione Spazio
 - Modifica Spazio
- Gestione Post
 - Creazione Post
 - Modifica Proprio Post

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano requisiti funzionali di bassa o media priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'utente, ad esempio le attività riguardanti esclusivamente la visualizzazione di dati.



5. Pass/Fail Criteria

Le attività di testing sono finalizzate a individuare eventuali errori (faults) nel sistema, così da poter procedere alla loro successiva correzione. L'esito di un test case viene determinato utilizzando un oracolo, ossia il risultato previsto per l'esecuzione del test, basato sui requisiti. Un test si considera **superato (pass)** se, dato un input al sistema, l'output prodotto coincide con quello previsto dall'oracolo. Un test si considera **fallito (fail)** se, dato un input al sistema, l'output prodotto differisce da quello previsto dall'oracolo. Il testing complessivo sarà considerato valido solo se tutti i seguenti criteri saranno soddisfatti:

- Verifica di tutti i requisiti funzionali prioritari;
- Esecuzione di test di regressione ogni volta che vengono introdotte nuove funzionalità o modificate quelle esistenti.

6. Approccio

Il testing dell'intero sistema si articola in due fasi principali: test di unità e test di sistema. I casi di test di sistema saranno progettati prima della fase di implementazione e perfezionati durante la loro esecuzione. Nel corso dello sviluppo, saranno condotte periodiche revisioni del codice da parte di **tutti i membri coinvolti** per garantirne la qualità.

Poiché la progettazione segue un modello iterativo e incrementale, le attività di testing di sistema sono pianificate in parallelo alla definizione e revisione dei requisiti documentati nel **Requirements Analysis Document (RAD)**.



Testing Di Sistema

Il server, per la fase di testing, verrà deployato in localhost e sarà accessibile alla porta 8080 a meno di diversa configurazione da parte dell'utente.

Functional Testing

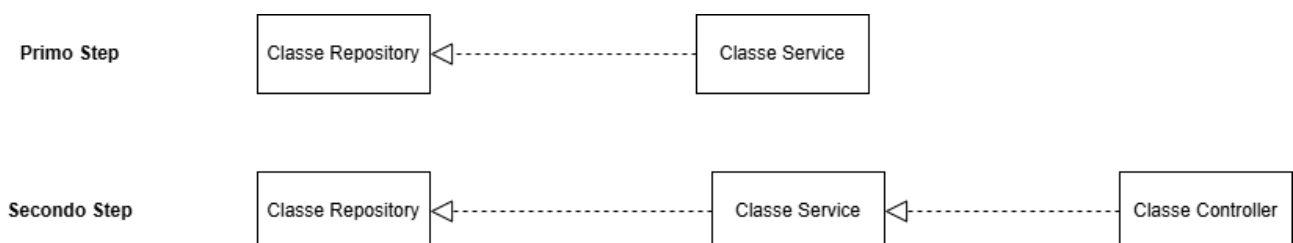
Il functional testing ha lo scopo di verificare i requisiti funzionali prioritari descritti nel documento RAD e si concentra sull'identificazione dei possibili errori derivanti dagli input degli utenti.

Testing Di Integrazione

Sarà effettuato il testing di integrazione direttamente sul codice sorgente adottando un approccio **bottom-up**, considerato particolarmente adatto per il nostro sviluppo client-server che parte prima dalla costruzione e dal testing dei sottosistemi back-end. I test case saranno definiti utilizzando il framework **JUnit**, mentre per il mocking verrà impiegato **Mockito**.

Per la Continuous Integration e il Continuous Development (CD/CI) verrà usato **Travis CI**.

Il test di integrazione seguirà lo stesso approccio per tutte le componenti da verificare. In particolare, si inizierà testando le classi **Service**, per poi passare al test delle classi **Controller**. Durante il test delle classi Controller, le chiamate ai controller saranno simulate utilizzando **Mockito**.





Testing Di Unità

La strategia prevista per il testing di unità prevede di testare ogni metodo delle classi del sistema, escludendo le interfacce e le classi entity. Queste ultime sono escluse in quanto contengono unicamente metodi getters e setters e costruttori. I casi di test saranno definiti utilizzando un approccio black-box. Per ogni classe di produzione (Production Class) verrà creata una corrispondente classe di test (Test Class) con il formato **ProductionClassTest**. Le classi di test saranno sottoposte a revisione e modifica per garantire maggiore qualità e affidabilità.

Altre tecnologie usate in tale fase saranno:

- **Junit:** per effettuare il testing dei singoli metodi;
- **Mockito:** per la costruzione degli stub e l'isolamento della componente testata;
- **Maven:** per la build e l'esecuzione automatica dei tests.

7. Sospensione e ripristino

Questa sezione definirà i criteri per la sospensione dei test e le attività che dovranno essere ripetute al momento della ripresa dei test.

Criteri di sospensione

Il testing proseguirà senza interruzioni fino alla sua conclusione, anche in presenza di una failure. Tuttavia, potrà essere temporaneamente sospeso nel caso in cui, durante l'esecuzione, venga rilevato un errore nella definizione di uno dei test.

Criteri di ripristino

Il testing verrà ripreso solo dopo aver risolto i fault individuati, oppure nel caso di un errore nella definizione di un test, solo dopo la revisione e la correzione di tale errore.



8. Materiale di testing

Per l'attività di test sarà sufficiente un computer.

9. Test Cases

9.1 Gestione Utente

9.1.1 Modifica dati utente

Parametro: Nome	
Formato	
^[A-Za-zÀ-ÿ\s]{2,30}\$	
Nome Categoria	Scelte per la categoria
Formato [FN]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FN_OK]
Parametro: Cognome	
Formato	
^[A-Za-zÀ-ÿ\s]{2,30}\$	
Nome Categoria	Scelte per la categoria
Formato [FC]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FC_OK]



Parametro: Email	
Formato	
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$	
Nome Categoria	Scelte per la categoria
Formato [FE]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FE_OK]
Parametro: Password	
Formato	
^(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{6,}\$	
Nome Categoria	Scelte per la categoria
Formato [FP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FP_OK]
Parametro: Conferma Password	
Formato	
^(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{6,}\$	
Nome Categoria	Scelte per la categoria



Match [MCP]	<ol style="list-style-type: none"> 1. match con password = false[error] 2. match con password = true[PROPERTY MCP_OK]
Parametro: Username	
Formato $^{\wedge}[A-Za-z0-9._-\backslash s]\{3,20\}$$	
Nome Categoria	Scelte per la categoria
Formato [FU]	<ol style="list-style-type: none"> 1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FU_OK]



9.1.2 Login

Parametro: Email	
Formato $^{\wedge}[a-zA-Z0-9._ \% + -] + @ [a-zA-Z0-9. -] + \. [a-zA-Z] \{ 2, \} \$$	
Nome Categoria	Scelte per la categoria
Formato [FE]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FE_OK]
Parametro: Password	
Formato $^{\wedge} (? = . * [A-Z]) (? = . * \backslash d) (? = . * [@ \$! \% * ? \&]) [A-Za-z \backslash d @ \$! \% * ? \&] \{ 6, \} \$$	
Nome Categoria	Scelte per la categoria
Formato [FP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FP_OK]



9.1.3 Registrazione

Parametro: Nome	
Formato $^{\wedge}[A-Za-z\grave{A}-\ddot{y}\backslash s]\{2,30\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FN]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FN_OK]
Parametro: Cognome	
Formato $^{\wedge}[A-Za-z\grave{A}-\ddot{y}\backslash s]\{2,30\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FC]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FC_OK]
Parametro: Email	
Formato $^{\wedge}[a-zA-Z0-9._ \% + -] + @ [a-zA-Z0-9. -] + \backslash . [a-zA-Z]\{2, \}\$$	
Nome Categoria	Scelte per la categoria



Formato [FE]	<ol style="list-style-type: none"> rispettare il formato = false[error] rispettare il formato = true[PROPERTY FE_OK]
Parametro: Password	
Formato $^(\?=.*[A-Z])(\?=.*\d)(\?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{6,}\$$	
Nome Categoria	Scelte per la categoria
Formato [FP]	<ol style="list-style-type: none"> rispettare il formato = false[error] rispettare il formato = true[PROPERTY FP_OK]
Parametro: Conferma Password	
Formato $^(\?=.*[A-Z])(\?=.*\d)(\?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{6,}\$$	
Nome Categoria	Scelte per la categoria
Match [MCP]	<ol style="list-style-type: none"> match con password = false[error] match con password = true[PROPERTY MCP_OK]



Parametro: Username	
Formato	
^[A-Za-z0-9._\-\s]{3,20}\$	
Nome Categoria	Scelte per la categoria
Formato [FU]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FU_OK]

Test Case ID	Test frame	Esito
TC1.1	MCP	Errato: Match con password errato
TC1.2	FP	Errato: Password non rispetta il formato
TC1.3	FE	Errato: email non rispetta il formato
TC1.4	FN	Errato: Nome non rispetta il formato
TC1.5	FC	Errato: Cognome non rispetta il formato
TC1.6	FU	Errato: Username non rispetta il formato



9.2 Gestione Spazio

9.2.1 Creazione Spazio

Parametro: Titolo Spazio	
Formato $^[A-Za-zÀ-ù0-9, '\- _]{2,50}\$$	
Nome Categoria	Scelte per la categoria
Formato [FTS]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FTS_OK]
Parametro: Descrizione	
Formato $^[\\w\\s\\p{Punct}]{1,200}\$$	
Nome Categoria	Scelte per la categoria
Formato [FDS]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FDS_OK]



Parametro: Argomento	
Formato ^[A-Za-zÀ-ÿ\s]{2,30}\$	
Nome Categoria	Scelte per la categoria
Formato [FAS]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FAS_OK]
Parametro: Immagine	
Formato "jpg", "png"	
Nome Categoria	Scelte per la categoria
Formato [FIS]	<ol style="list-style-type: none">1. rispettare il formato = false[error]2. rispettare il formato = true[PROPERTY FIS_OK]



9.2.2 Modifica Spazio

Parametro: Titolo Spazio	
Formato $^[A-Za-z\À-\grave{0}-9, '\- \s]{2,50}\$$	
Nome Categoria	Scelte per la categoria
Formato [FTS]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FTS_OK]
Parametro: Descrizione	
Formato $^[\w \s \p{Punct}] \{ 1, 200 \} \$$	
Nome Categoria	Scelte per la categoria
Formato [FDS]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FDS_OK]
Parametro: Argomento	
Formato $^[A-Za-z\À-\grave{\s}]{2,30}\$$	



Nome Categoria	Scelte per la categoria
Formato [FAS]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FAS_OK]
Parametro: Immagine	
Formato "jpg", "png"	
Nome Categoria	Scelte per la categoria
Formato [FIS]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FIS_OK]

Test Case ID	Test frame	Esito
TC2.1	FTS	Errato: Titolo non conforme al formato
TC2.2	FDS	Errato: Descrizione non conforme al formato
TC2.3	FAS	Errato: Argomento non conforme al formato



TC2.4	FIS	Errato: Immagine dello spazio non conforme al formato
-------	-----	---

9.3 Gestione Post

9.3.1 Creazione Post

Parametro: Immagine	
Formato "jpg", "png"	
Nome Categoria	Scelte per la categoria
Formato [FIP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FIP_OK]
Parametro: Testo	
Formato $^[\backslash w\backslash s\backslash p\{Punct\}]{1,400}\$$	
Nome Categoria	Scelte per la categoria
Formato [FTP]	3. rispettare il formato = false[error] 4. rispettare il formato = true[PROPERTY FTP_OK]



9.3.2 Modifica Post

Parametro: Immagine	
Formato "jpg", "png"	
Nome Categoria	Scelte per la categoria
Formato [FIP]	5. rispettare il formato = false[error] 6. rispettare il formato = true[PROPERTY FIP_OK]
Parametro: Testo	
Formato $^[\backslash w\backslash s\backslash p\{Punct\}]{1,400}\$$	
Nome Categoria	Scelte per la categoria
Formato [FTP]	7. rispettare il formato = false[error] 8. rispettare il formato = true[PROPERTY FTP_OK]



Test Case ID	Test frame	Esito
TC3.1	FIP	Errore: Il formato dell'immagine non è conforme
TC3.2	FTP	Errore: Testo non conforme al formato