Test log

Date	(Class) Test Name	Test type	Test Objective	Category	Condition to Verify	Expected Results	Actual Results	Comments
18/05	python manage.py test	Già scritti in precedenza.	Controllare il successo di tutti i test, per poter chiudere lo sprint 3 e consegnare il progetto			59 test passati (True), 0 errori	57 test passati, 2 errori	In UpdateText e DeleteText, se l'utente non era autenticato avevamo un redirect (302), ma per evitare problemi di sicurezza abbiamo cambiato in NotPermitted (403) così da evitare l'accesso alla modifica ad altri utenti.
19/05	user_management/ test_view. test_view_delete_user	Appena scritto.	Controllare che la view "DeleteUserView" cancelli effettivamente l'utente autenticato e tutti i suoi testi/analisi.	Django test	Se si accede a DeleteUserView, l'utente e i suoi dati/testi/analisi non devono essere più presenti.	True (nessun dato dell'utente presente) e re-direct al login.	True	
20/05	text_management/ test_urls	Già scritti in precedenza, testati nuovamente	Controllare che la risoluzione degli urls dell'app	Django test	Ad ogni url deve corrisponder e una	Dato un url in input la classe deve ritornare True la view	Ogni sotto-test della classe ritorna True,	

		in quanto file urls.py leggermente modificato.	text_management richieda le giuste Views.		specifica View	ottenuta è uguale a quella attesa, diversamente false.	quindi ogni url è collegato al template corretto.	
20/05	text_management/ test_models. test_blacklist_user	Appena scritto.	Controllare che dopo la creazione di un utente, vengano create in automatico le sue impostazioni, e che la @property blacklist dell'utente funzioni.	Django test	Che esistano le impostazioni di un utente dopo la sua creazione. La property blacklist ritorni ['prova', 'ciao', 'di'] se la blacklist dell'utente è 'prova; ciao;di'	True	True	
20/05	text_management/ test_templatetags. test_get_user_popular_words	Appena scritto, con tecnica TDD.	Controllare che, applicando il filtro per filtrare le parole nella blacklist dell'utente ad una lista di testi, la lista di	Django test	Che dalla lista delle parole dei testi salvati dall'utente non vengano ritornate le parole della sua blacklist.	True	True	Ovviamente, essendo stata applicata la tecnica TDD il test inizialmente failava, per poi ritornare True dopo aver scritto nella maniera giusta il filtro.

			parole e frequenze ritornata non contenga parole della blacklist.					
22/05	text_management/test_urls	Aggiunta dei test sulle nuove url, testati nuovamente tutti	Controllare che la risoluzione degli urls dell'app text_management richieda le giuste Views.	Django test	Ad ogni url deve corrisponder e una specifica View	Dato un url in input la classe deve ritornare True la view ottenuta è uguale a quella attesa, diversamente false.	Ogni sotto-test della classe ritorna True, quindi ogni url è collegato al template corretto.	
22/05	text_management/test_models	Aggiunta test sul model “UserTextSettings”	Controllare che le property e i campi riguardanti la blacklist e il minimo n° di caratteri per salvare le parole settati dall’utente funzionino.	Django test	Settando adeguatamente i settings dell’utente, le proprietà di Text devono restituire i giusti risultati filtrati	[‘prova’, ‘ciao’, ‘di] per la blacklist, {‘ciao’: 2, ‘come’: 4}	Tutti i test (anche i precedenti) sono OK.	
22/05	text_management/test_views	Aggiunta dei test sulle nuove view inserite	Controllare che non si possa accedere alle View se non si	Django test	Impedire a utenti non loggati di accedere alle	Risposta 302 (reindirizzamento al login) per il primo tentativo	Tutti i test (anche i precedenti) sono OK.	

			è autenticati. Controllare che la risposta alla richiesta ritorni il giusto template della View corrispondente.		View del sito. Restituire i template giusti.	senza autenticazione. Risposta 200 dopo essersi loggati, e template ritornato corrispondente a quello della View.		
22/05	text_management/ test_views	Aggiunta di test per controllare l'effettivo filtraggio della blacklist durante la visualizzazione dell'analisi di un testo e l'eliminazione di parole troppo corte dopo l'inserimento di un nuovo testo.	Controllare che la blacklist venga applicata all'analisi di un testo da visualizzare e che, dopo aver inserito un nuovo testo, vengano eliminate le parole troppo corte per l'utente.	Django test	Data una testo di partenza, una blacklist ed un certo numero di caratteri l'analisi non deve mostrare parole contenute nella blacklist, e dopo l'inserimento di un nuovo testo non devono essere presenti	Dopo l'inserimento del testo specificato nel test la visualizzazione dell'analisi deve ritornare solo {'prova': 1, 'testo': 2, 'qualunque': 1}, eliminando le parole < 4 caratteri. Per la blacklist, la visualizzazione deve ritornare solo [('come', 4)].	Tutti i test passati.	Difficoltà nel capire come passare i dati del testo e dell'utente alla view/template e salvare i settings dell'utente nel test.

					parole troppo corte.			
22/05	text_management/ text_views	Aggiunta test per controllare la modifica dei settings di un utente (blacklist e min_characters).	Provata la view sia in GET (visualizzazione) che POST (salvataggio effettivo delle impostazioni).	Django test	Impedire l'accesso a utenti non autenticati, ritornare il giusto template e salvare effettivamente le nuove impostazioni.	Dopo la richiesta in POST, l'utente deve avere salvato come impostazioni ['ciao', 'prova] come blacklist e 5 come min_characters	Passati.	Difficoltà nel capire come passare i dati del testo e dell'utente alla view/template e salvare i settings dell'utente nel test.
24/05	cruscanalyzer/ test_acceptance	Completamento test di accettazione automatizzato	Controllare la corretta registrazione di un utente, il login, l'inserimento di testi, il loro salvataggio, la loro modifica ed eliminazione e il confronto di più testi.	Django test	Impedire accesso alle pagine dell'app ad utenti non autenticati/non proprietari dei testi, controllare il corretto funzionamento di registrazione, login, inserimento, salvataggio, modifica ed	Tutti gli assert effettuati (su template, valori, response…) siano giusti.	Tutti gli assert sono corretti.	Essendo un test di accettazione molto grande, difficoltà iniziale nel integrare il tutto nel modo corretto.

					eliminazioni di testi e il loro confronto.			
24/05	python manage.py test	Tutti i test creati in precedenza.	Controllo che, con le modifiche al codice effettuate, tutti i test producano ancora esito positivo	Django test	Controllare che le modifiche effettuate non abbiano creato bug/problemi che portano al fallimento dei test.	75 test corretti.	75 test corretti.	
25/05	python manage.py test	Tutti i test creati in precedenza.	Controllo che, con le modifiche al codice effettuate, tutti i test producano ancora esito positivo, per poter consegnare il progetto.	Django test	Controllare che le modifiche effettuate non abbiano creato bug/problemi che portano al fallimento dei test.	75 test corretti.	75 test corretti.	