

Nota: questa documentazione non va a sostituire quella precedente ma ad integrarla con i nuovi documenti creati nel primo sprint

- 10. Retrospective
 - 10.1 Preparazione
 - 10.2 Cosa è andato bene?
 - 10.3 Cosa può essere migliorato?
 - 10.4 Action Point
- 11. Ruoli
- 12. Backlog aggiornato
 - 12.1 Sprint Planning meeting
 - 12.2 User Stories aggiornate
- 13. Modelli
 - 13.1 Diagramma delle classi
- 14. Definition of Ready
- 15. Definition of Done
- 16. Test Strategy

10 Retrospective

10.1 Preparazione

Esistono svariati metodi per condurre un Retrospective meeting, quello più tipico ed è quello che abbiamo usato prevede di tracciare su una lavagna, o su una superficie preparata appositamente, tre colonne. Visto i tempi in cui viviamo, abbiamo optato di usare un software online:

<https://ideaboardz.com/>

Come intestazione per la prima colonna abbiamo messo : Cosa è andato bene. Nella seconda colonna: Cosa si può migliorare. La terza colonna è riservata agli Action Point, ovvero ai punti su cui si decide di intervenire nel prossimo Sprint.

10.2 Cos'è andato bene ?



Il Retrospective meeting inizia con l'inserimento delle proprie osservazioni in maniera libera e obiettiva nei post-it. In questa fase si dà un intervallo di tempo (10 minuti), in cui ogni partecipante riflette in silenzio e inizia a inserire i post-it.

Come si può notare, tutto il team è stato abbastanza contento dell'attività svolte, sia proprie e sia degli altri membri.

Specialmente la comunicazione tra il team, ci si aiuta a vicenda per garantire uno stesso livello di conoscenza tra il team e una continua attività di produzione.

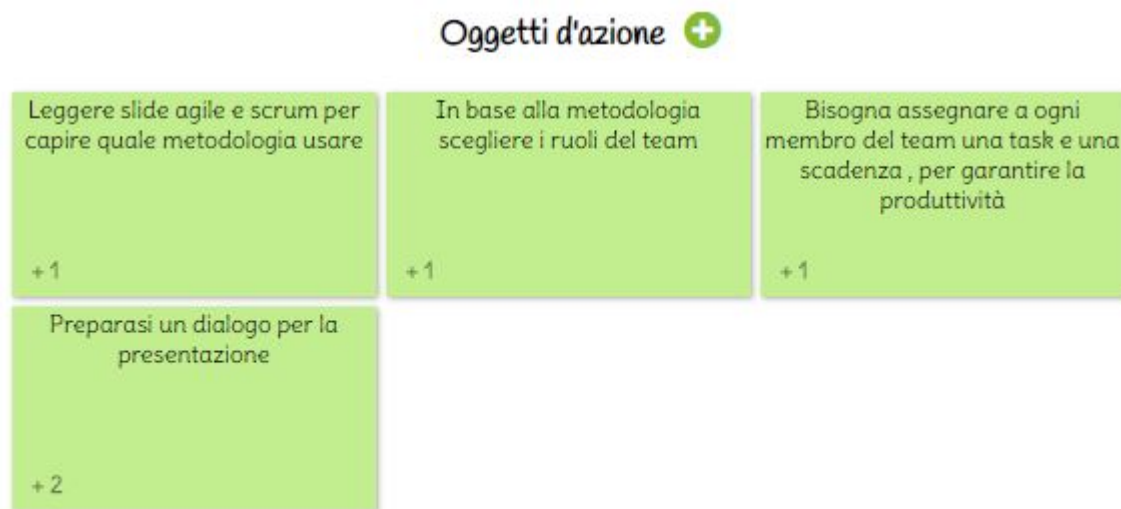
10.3 Cosa può essere migliorato ?



Come si può notare , abbiamo notato che, come in tutti i team di un progetto , abbiamo percepito della lacune da colmare o da sistemare. Il team si è sentito di giudicare questi punti , che secondo loro , sono punti importanti per la continuità del progetto. I punti che abbiamo considerato più critici sono:

- 1) Gestione del tempo e organizzazione dei task individuali. (Cercare di non perdere molto tempo e produrre di più).
- 2) Gestione dei ruoli del team. (Avere in modo chiaro ogni membro team che cosa deve fare.)

10.4 Action Point



Solo a questo punto siamo arrivati alla terza e ultima colonna. Qui la procedura è stata la stessa: post-it per tutti, opinioni diverse, discussione, e sono saltati fuori questi post indicati sopra.

Dalla votazione il team ha deciso di cercare di presentarsi meglio con il cliente, cercando di essere più dettagliati e meno imprecisi, contiamo nel dimostrare la nostra professionalità e di garantire le tempistiche richieste dal cliente.

La seconda criticità che vogliamo assolutamente risolvere l'assegnazione dei ruoli e la metodologia più adatta a noi.

11 Ruoli

L'assegnazione dei ruoli è una criticità che è stata rilevata durante la retrospective. Come team abbiamo deciso di utilizzare la metodologia scrum da cui abbiamo deciso i ruoli di ogni componente del team

Product Owner: Jihed Fatnassi
Scrum Master: Ivan Cotti
Developer Team: Federico Tani,
Andrea Donato,
Filippo Principato,
Matteo Cremonini

12 Backlog Aggiornato

12.1 Sprint Planning Meeting

Il team ha pianificato il lavoro che deve essere svolto e portato a termine durante lo Sprint. Sono stati evidenziati due obiettivi generali da realizzare durante lo sprint, calcolando le ore di sviluppo ideali.

Sprint Goal: Autenticazione (~2 / 3 ore)

Sprint Planning:

Guest deve poter registrarsi

- inserimento mail
- inserimento nome utente
- inserimento e verifica password
- autenticazione tramite OAuth

User deve poter fare il login:

- inserimento nome utente o mail
- inserimento password
- controllo se l'utente è registrato nel DB
- autenticazione tramite OAuth

Sprint Goal: features necessarie per utilizzare il software (~14 ore)

Sprint Planning:

Creazione Inserimento (~ 6 ore):

- inserimento di un titolo
- inserimento di un testo
 - permettere incolla con tasto dx (◦ CTRL+V)
- utente clicca "OK" quando è pronto a far processare
- testo processato
 - salvare parole e la loro frequenza
 - dare punteggio al testo

Visualizzare Dettagli di un articolo (~ 2 ore):

- utente deve poter selezionare l'articolo che vuole visualizzare
- mostrare titolo, indice di complessità, parole + occorrenze

User deve poter visionare Lista dei suoi articoli (~2 ore):

- utente vede i titoli degli articoli in base alla data di creazione
- permettere di selezionare un articolo per vederne i dettagli
- permettere di cancellare un articolo
 - messaggio di conferma
- permettere di modificare il titolo di un articolo
 - messaggio di conferma

Visionare gli articoli degli altri utenti (~ 4 ore):

- necessità di una bacheca con gli articoli ordinati in base alla data di creazione
- di ogni articolo si deve poter vedere il titolo, l'utente creatore e l'indice di complessità
- permettere di selezionare un articolo

12.2 User Stories Aggiornate

Le seguenti User Stories sono state spostate dallo Sprint 2 allo Sprint 1 perchè ci sono sembrate delle features necessarie per realizzare il software. Sono già state stimate precedentemente dal team e qua sotto sono in ordine di priorità.

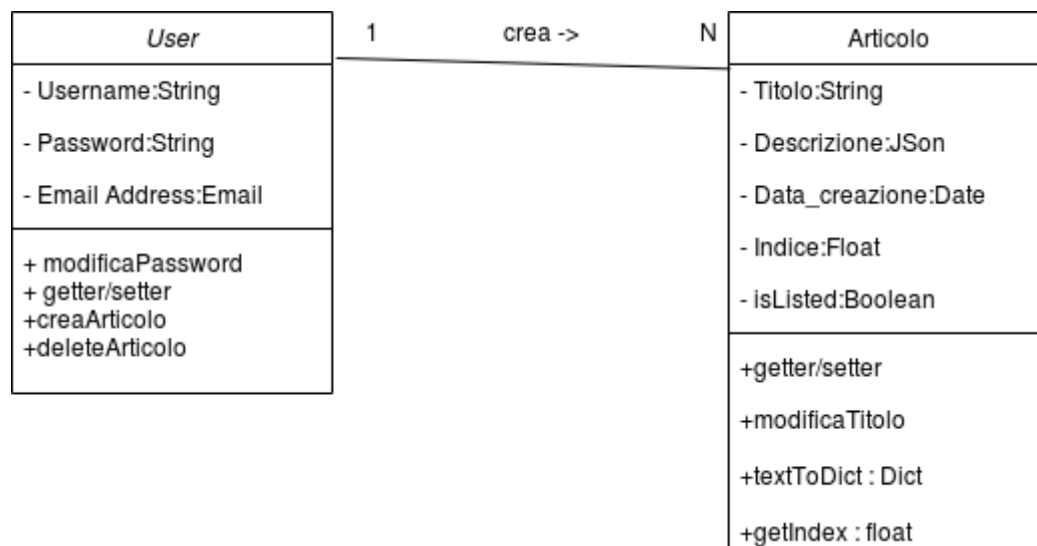
1. come utente voglio poter aggiornare e quindi modificare la password del mio account
2. come utente voglio poter cercare testi processati

Inoltre per lo sprint 2 il team ha deciso di aggiungere una nuova UserStory stimata e ordinata. La stima utilizzata è la stessa utilizzata per le UserStories precedenti.

[20] come utente voglio poter ordinare la bacheca secondo vari criteri di ordinamento

13 Modelli

13.1 Diagramma delle classi



Spiegazione

Le classi utilizzate sono due : User e Articolo. La relazione tra le due classi è una associazione semplice 1 ad N , visto che uno User può creare N Articoli ed un Articolo è creato da un solo User.

Per quanto riguarda i metodi di Articolo, oltre a quelli di default, sono presenti textToDict e getIndex. Il primo serve a

processare il testo in un dizionario in modo che vengano salvate solo le parole con le loro occorrenze e non tutto il testo integrale. Il dizionario viene poi convertito a JSON per poterlo salvare nel modello. `GetIndex` è un metodo che viene chiamato dopo che il testo è stato processato in un dizionario, tiene conto della lunghezza delle parole e delle occorrenze e serve per poter assegnare un indice di complessità all'articolo.

14 Definition of Ready

Per considerare una storia Ready, tutti i membri del team dovevano essere d'accordo sulla sua priorità e sul contenuto. Inoltre durante lo sprint planning meeting abbiamo diviso ogni US in passi necessari (To-Do) alla sua creazione. Le storie per essere considerate Ready infine devono poter essere indipendenti dalle altre, in modo da poterle sviluppare separatamente in team più piccoli supportati dalla divisione del codice in app di django. Durante lo sprint planning meeting abbiamo modificato le storie in modo che potessero essere considerate Ready

15 Definition of Done

Per considerare una storia Done abbiamo deciso che devono essere state completate la codifica della funzionalità richiesta e i test di unità. Per quanto riguarda la codifica, tutte le To-Do di una storia devono essere state risolte e il codice deve essere integrato in modo ottimale con il codice già esistente. Dopo aver verificato l'integrazione, le funzionalità devono essere testate. Anche qui django viene in aiuto visto che permette nel suo framework di creare dei test sia per i modelli sia per le view.

16 Test Strategy

Il framework scelto per la creazione del sito web (Django) ha implementato di default una metodologia per applicare i test. Il sito è diviso in 3 applicazioni: accounts, core e forum. Quando queste app vengono create dal framework viene inserito automaticamente un file chiamato "tests.py" che permette di scrivere test specifici per quella applicazione. La nostra strategia per i test è stata di analizzare ogni singola view poiché venga testata ogni singola azione possibile dall'utente. In particolare abbiamo utilizzato gli status code per testare i permessi di visualizzazione degli utenti, visto che il sito non accetta utenti non registrati. Partendo da questo presupposto abbiamo verificato che in caso di utente non registrato ogni sua richiesta venisse reindirizzata alla pagina del login. Un altro test che abbiamo eseguito è stato il controllo sulla modifica del titolo di un articolo e sull'eliminazione dell'articolo. Grazie a quest'ultimo test ci siamo resi conto di una vulnerabilità del sito, che sarebbe potuta essere sfruttata da utenti registrati con cattive intenzioni, sfruttando questo exploit si sarebbe potuto modificare o cancellare un articolo che non era dell'utente creatore. Per testare questo exploit non abbiamo potuto utilizzare i redirect siccome l'utente malintenzionato è registrato e quindi ha i permessi per visualizzare l'articolo. Abbiamo verificato che i template utilizzati fossero quelli che consentivano solo la visualizzazione dei dettagli dell'articolo.