

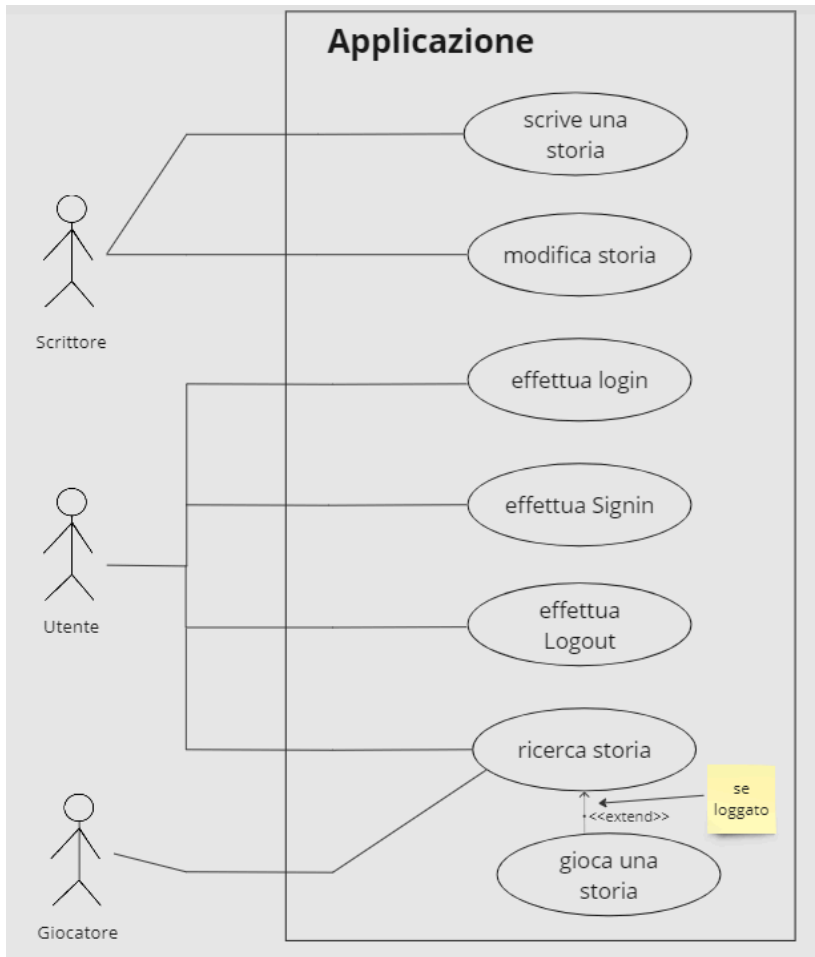
Indice

Indice.....	1
Modello dei casi d'uso.....	2
Modello di dominio.....	4
Glossario.....	5
Burn Down Chart.....	6
Manuale dello sviluppatore.....	8
Manuale dell'utente.....	14
Diario del Progetto.....	17

Modello dei casi d'uso

Attenzione: una sintesi di tutti i modelli realizzati è inoltre presente come pdf distinto, così da facilitarne la visualizzazione

Diagramma:



Descrizione testuale:

Use Case: **Effettua Login**

Actor: Utente

Pre-Conditions: Utente Registrato

Main Sequence:

1. l'utente accede al menù iniziale
 2. l'app mostra il menù iniziale
 3. l'utente clicca su "Login"
 4. l'app mostra il menù di login
 5. l'utente inserisce le credenziali
 6. L'app consente l'accesso alla home page
- Post-Condition: Login effettuato

Use Case: **Effettua Sign In**

Actor: Utente

Pre-Conditions: Utente non registrato

Main Sequence:

1. l'utente accede al menù iniziale
 2. l'app mostra il menù iniziale
 3. l'utente clicca su "SignIn"
 4. L'app richiede le credenziali per la registrazione
 5. l'utente inserisce le credenziali
 6. L'app invia un messaggio di conferma all'utente
 7. L'app mostra all'utente la home page
- Post-Condition: nuovo utente registrato

Use Case: **Effettua LogOut**

Actor: Utente

Pre-Conditions: Utente loggato

Main Sequence:

1. l'utente accede alla home page
 2. l'app mostra la home page
 3. l'utente clicca sul bottone "logout"
 4. Effettua Log Out"
 6. L'app reindirizza l'utente al menù iniziale
- Post-Condition: Log Out effettuato

miro

Use Case: **Ricerca storia**

Actor: Utente, Giocatore

Pre-Conditions: Accesso al sito

Main Sequence:

1. l'utente accede al menù principale
2. l'app mostra il menù principale
3. l'utente clicca su "visualizza catalogo"
4. l'app mostra la lista delle storie e la barra di ricerca
5. l'utente inserisce i dati per la ricerca
6. l'app reindirizza l'utente alle storie con le informazioni desiderate

Alternative path:

1. l'utente accede alla home page
2. l'app mostra la home page
3. l'utente clicca su "visualizza catalogo"
4. l'app mostra la lista delle storie e la barra di ricerca
5. l'utente inserisce i dati per la ricerca
6. l'app reindirizza l'utente alle storie con le informazioni desiderate

Post-Condition: Storia cercata

Use Case: **Gioca ad una storia**

Actor: Giocatore

Pre-Condition: utente loggato

Main Sequence:

1. il giocatore accede alla home page
 2. l'app mostra la home page
 3. il giocatore clicca su "Visualizza Catalogo"
 4. L'app mostra le storie disponibili
 5. Il giocatore sceglie la storia e preme "Gioca"
 6. L'app consente l'accesso al giocatore
 7. Il giocatore inizia a giocare
- Post-Condition: Giocatore inizia/prosegue la partita

miro

Use Case: **Scrivi una storia**

Actor: Scrittore

Pre-Conditions: Utente Registrato

Main Sequence:

1. utente accede alla home page
2. l'app mostra la home page
3. lo scrittore clicca su "Scrivi una storia"
4. L'app mostra la schermata di scrittura
5. l'utente scrive il nome della storia
6. l'app mostra la schermata di creazione degli scenari
7. l'utente crea gli scenari
8. l'app mostra la schermata per i collegamenti
9. utente setta i collegamenti
10. utente conferma la creazione
11. app crea la storia

Post Condition: storia creata

Use Case: **Modifica una storia**

Actor: Scrittore

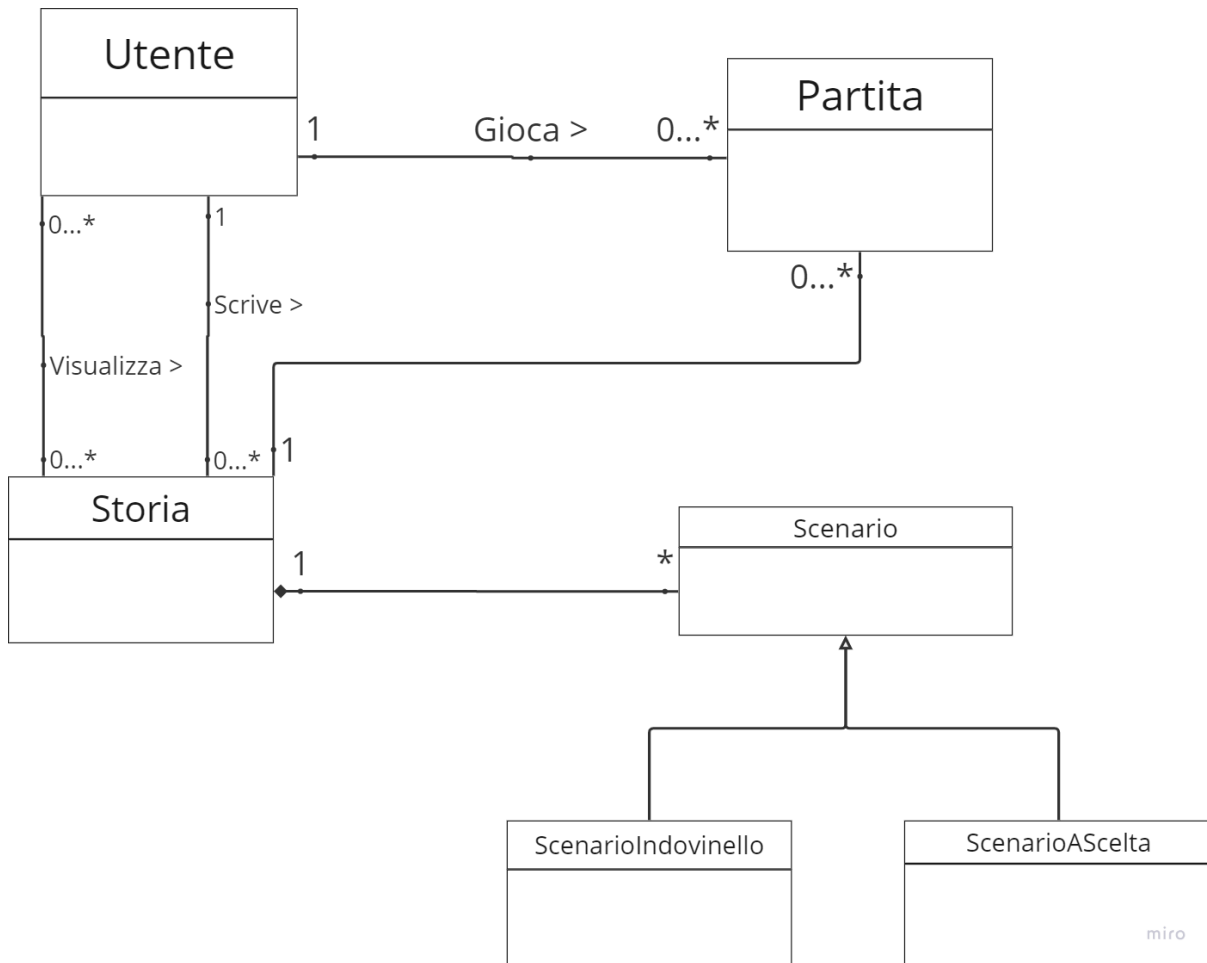
Pre-Conditions: Utente loggato, almeno una storia scritta dall'utente

Main Sequence:

1. l'utente accede alla home page
 2. l'app mostra la home page
 3. l'utente clicca su "Modifica una storia"
 4. L'app mostra le storie scritte dall'utente
 5. l'utente seleziona la storia da modificare
 6. L'app mostra la schermata di modifica
 7. lo scrittore modifica la storia
 8. Lo scrittore clicca su "salva la storia"
 9. L'app mostra il salvataggio completato
 10. L'app reindirizza alla schermata di scelta della storia da modificare
- Post-Condition: Storia modificata

miro

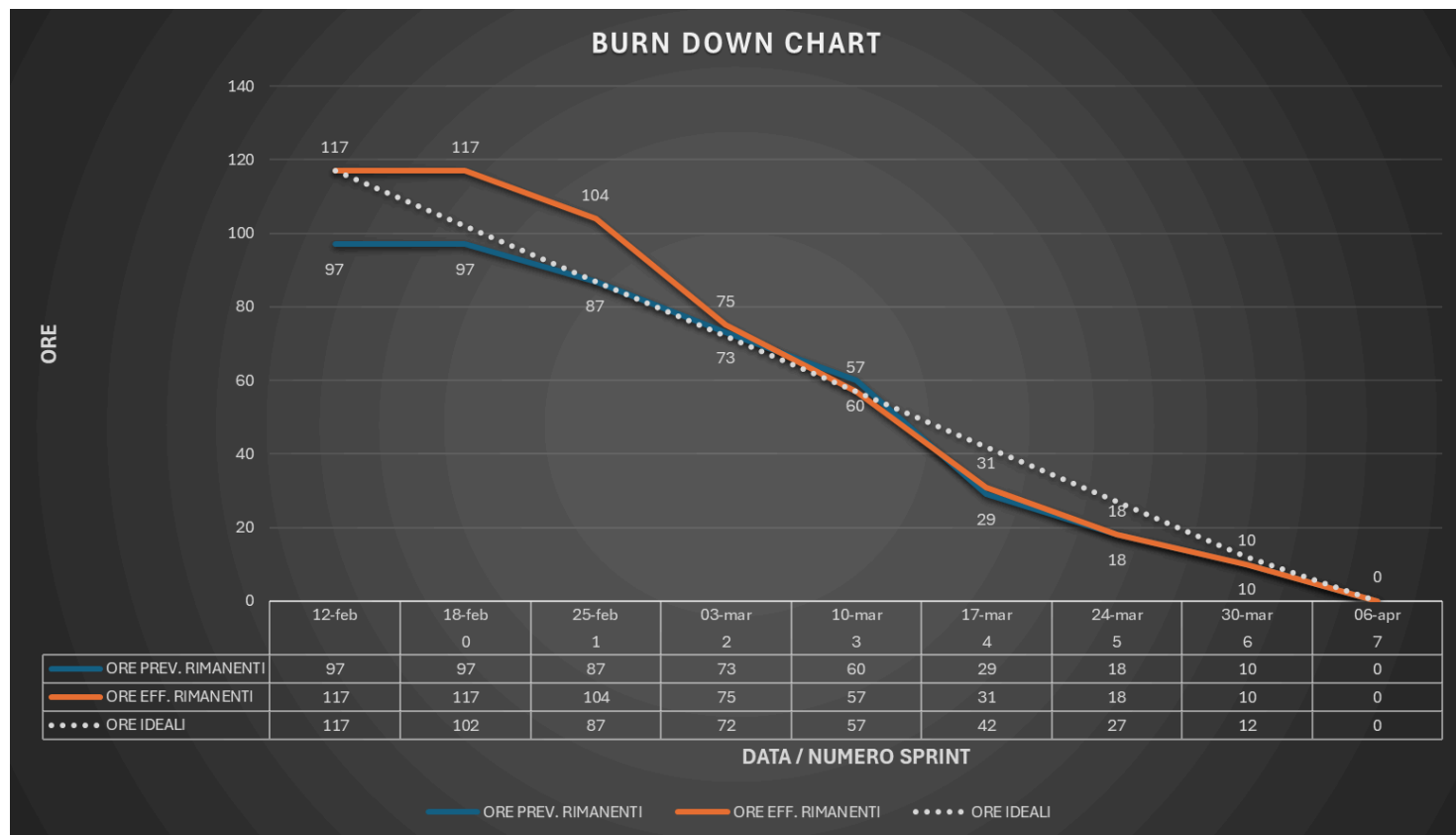
Modello di dominio



Glossario

- **Utente:** Può essere un semplice visitatore che entrando nel sito senza effettuare il login visualizza le storie non potendo però giocare. Oppure può essere un giocatore e/o uno scrittore di storie qualora decida di accedere.
- **Catalogo delle storie:** Una raccolta di storie disponibili per la lettura e il gioco all'interno dell'applicazione.
- **Storia:** Una narrazione in cui il giocatore può prendere decisioni che influenzano lo sviluppo della storia.
- **Scenario:** Una situazione o un evento specifico all'interno di una storia interattiva, che presenta una o più scelte per il giocatore.
 - *Scenario a scelta:* l'utente ha opzioni date tra cui scegliere (alcune di queste vengono precluse dal possesso di oggetti raccolti in scenari precedenti) che porteranno a scenari successivi diversi.
 - *Scenario a indovinello:* il giocatore dovrà rispondere ad un indovinello per poter proseguire, in base a come risponde cambierà scenario.
 - *Scenario finale:* è lo scenario che comparirà una volta finito di giocare alla storia, raggiunto attraverso una serie di decisioni del giocatore. Può essere diverso in base alle decisioni prese nella storia.
- **Oggetto:** Un oggetto che si può raccogliere in uno scenario per poi usarlo per prendere alcune decisioni negli scenari successivi. Questi limitano quindi la possibilità di fare delle scelte.
- **Inventario:** Una raccolta di oggetti che il protagonista può raccogliere durante lo svolgimento della storia interattiva e che possono essere utilizzati per risolvere determinate situazioni.

Burn Down Chart



N° SPRINT	DATA FINE SPRINT	ORE PREVISTE	ORE EFFETTIVE	ORE PREV. RIMANENTI	ORE EFF. RIMANENTI	ORE IDEALI
	12-feb	0	0	97	117	117
0	18-feb	0	0	97	117	102
1	25-feb	10	13	87	104	87
2	03-mar	14	29	73	75	72
3	10-mar	13	18	60	57	57
4	17-mar	31	26	29	31	42
5	24-mar	11	13	18	18	27
6	30-mar	8	8	10	10	12
7	06-apr	10	10	0	0	0
	ORE TOTALI	97	117			

RESUME:

- 10 issue
- 26 task
- Totale ore previste: 97
- Totale ore effettive: 117
 - Previsione s.0 = 0h - effettivo sprint s.0 = 0h

- Previsione s.1 = 10h - effettivo sprint s.1 = 13h
 - Previsione s. 2 = 14h - effettivo sprint s.1 = 29h
 - Previsione s. 3 = 13h - effettivo sprint s.1 = 18h
 - Previsione s. 4 = 31h - effettivo sprint s.1 = 26h
 - Previsione s. 5 = 11h - effettivo sprint s.1 = 13h
 - Previsione s. 6 = 8h - effettivo sprint s.1 = 8h
 - Previsione s. 7 = 10h - effettivo sprint s.1 = 10h
 - 7 sprint (+ sprint 0) da 7 giorni ciascuno
-
- Tempo impiegato: dal 12/02 al 06/04
 - Stack tecnologico: gwt, maven, jetty
 - Linguaggi utilizzati: java, xml, css
 - Software usato per i diagrammi: miro
 - Software usato per il testing: junit, jacoco

Manuale dello sviluppatore

Per iniziare

Per avviare il progetto:

- installare Java (versione meno recente utilizzata: 10.9.2)
- installare Maven (versione meno recente utilizzata: 3.9.6)
- da terminale aprire due schede,
 - in una inserire il comando:

```
mvn gwt:codeserver -pl *-client -am,
```

- nell'altra:

```
mvn jetty:run -pl *-server -am -Denv=dev
```

Attenzione: se il server non si avvia correttamente controllare che la porta sia libera!

Se così non fosse, da terminale inserire il seguente comando:

```
netstat -ano | find "9876"
```

per individuare il processo che occupa la porta.

Per terminare il processo inserire nel prompt:

```
taskkill /F /PID nprocesso
```

Infine aprire il browser e nella barra di ricerca digitare:

```
http://localhost:8080
```

Sulla pagina apparirà lo starter dell'applicazione.

Struttura generale del progetto

Il progetto è articolato in tre sezioni: Client, Server e Shared.

Il **Client** contiene le classi per gestire l'interfaccia dell'utente e per richiamare le operazioni ad essa connesse. Ogni classe .java ha il suo ui.xml per la realizzazione degli elementi grafici.

Nello **Shared** sono presenti le classi condivise tra il Client e il Server che permettono alle due parti di comunicare.

Nel **Server** ci sono tutti i metodi necessari per effettuare le operazioni richieste.

La comunicazione Client-Server è gestita da tre file principali:

- interfaccia **GreetingService** (Shared): contiene i metodi che possono essere chiamati da client diversi per interagire con il server. Implementa l'interfaccia **RemoteService** che è l'interfaccia base per i servizi remoti GWT (Google Web Toolkit);
- interfaccia **GreetingServiceAsync** (Shared): viene generata automaticamente da GWT durante la compilazione del codice client. Fornisce i metodi asincroni corrispondenti a quelli definiti nell'interfaccia **GreetingService**. Questi metodi consentono al client di eseguire chiamate RPC (Remote Procedure Call) al server in modo asincrono, evitando di bloccare l'interfaccia utente durante le operazioni di rete;
- classe **GreetingServiceImpl** (Server): implementazione concreta dell'interfaccia **GreetingService**. Estende **RemoteServiceServlet**, che è una classe base fornita da GWT per semplificare l'implementazione dei servizi remoti. In questo caso, implementando l'interfaccia **GreetingService**, fornisce l'implementazione dei metodi definiti in **GreetingService**. Quando il client chiama uno dei metodi definiti in **GreetingService**, la richiesta viene gestita da questa classe sul server.

Nel dettaglio

Il **Client** contiene le seguenti classi:

- **App**: classe di accesso che implementa l'EntryPoint.
- **Starter**: rappresenta la schermata iniziale costituita dal messaggio di benvenuto e dai bottoni per la registrazione, il login e la visualizzazione delle storie nel catalogo.
- **SignIn**: classe per la registrazione di nuovi utenti.
- **Login**: classe per effettuare l'accesso. Rimanda, in caso di successo, alla HomePage.
- **HomePage**: schermata da cui è possibile scrivere storie, visualizzare il catalogo, modificare storie ed effettuare il logout.
- **ScriviStoria**: permette all'utente di creare Storie. E' connessa alla classe Collegamenti.
- **Collegamenti**: permette di collegare i vari scenari di una storia.
- **StorieCreateDaUtente**: mostra tutte le storie create da un utente così da poterle selezionare ed eventualmente modificare.
- **ModificaStoria**: permette di modificare i dati testuali degli scenari appartenenti ad una storia.
- **VisualizzaCatalogo**: mostra tutte le storie del sito e permette, se l'utente ha effettuato l'accesso, di giocarele.
- **SalvataggiPartita**: prima di giocare permette all'utente di scegliere se ricominciare la storia o continuarla da dove l'aveva interrotta, nel caso in cui sia già stata iniziata

- **GiocaStoria**: consente di giocare alla storia. Si scorrono gli scenari e vengono mostrati all'utente.

Lo **Shared** contiene le seguenti classi:

- **GreetingService**: interfaccia che contiene i metodi per interagire con il server.
- **GreetingServiceAsync**: interfaccia con i metodi asincroni (corrispondono a quelli di GreetingService).
- **Utentel**: interfaccia che contiene le operazioni che possono essere svolte dall'utente.
- **Utente**: implementazione dell'utente.
- **Storial**: interfaccia che contiene le operazioni per le storie.
- **Storia**: implementazione della storia.
- **ScenarioI**: interfaccia che contiene le operazioni per gli scenari.
- **Scenario**: implementazione dello scenario (rappresenta lo scenario finale) e i metodi comuni a ScenarioAScelta e ScenarioIndovinello.
- **ScenarioAScelta**: estende Scenario, in più consente di poter associare ad uno scenario più scelte con eventuali oggetti necessari per compierle.
- **ScenarioIndovinello**: estende Scenario, in più consente di poter superare lo scenario rispondendo ad un indovinello.
- **TipologiaScenario**: enum con i tipi di scenario, DEFAULT (finale), ASCELTA, INDOVINELLO.
- **Partital**: interfaccia che contiene le operazioni per le partite.
- **Partita**: implementazione della partita con tutti i dati associati all'utente che la sta giocando.

Il **Server** contiene la classe **GreetingServiceImpl**, che implementa tutti i metodi di GreetingService e GreetingServiceAsync, **Database**, che implementa i metodi relativi alla gestione dei dati, e **ListenerImpl** che implementa i metodi all'avvio e alla chiusura del server.

Il Database è stato realizzato attraverso un **Singleton** così da far riferimento sempre allo stesso oggetto e da facilitarne l'accesso limitandone gli errori.

I metodi di **Database** sono:

- **getInstance**: ritorna il database.
- **openDb**: apre il database e gli hashMap contenenti tutti i dati necessari (storie, scenari, partite, utenti).
- i metodi **get** che ritornano le strutture dati (utenti, scenari, partite, db)
- **closedb()**: per la chiusura del database.

I metodi nel **GreetingServiceImpl** sono molteplici e riguardano la gestione di sezioni differenti dell'app.

Per la scrittura su **file**:

- **public void initData()**: richiama i getInstance per popolare le mappe.

- private void **convertToJsonUtenti()**: converte le informazioni presenti nell'hashMap relativo agli utenti in un file.json.
- private void **convertToJsonStorie()**: converte le informazioni presenti nell'hashMap relativo alle storie in un file.json.
- public boolean **salvaSuFileScenari(String nomeStoria)**: converte le informazioni presenti nell'hashMap relativo agli scenari in un file.json.
- private void **convertToJsonPartite()**: converte le informazioni presenti nell'hashMap relativo alle partite in un file.json.

Per gli **utenti**:

- public boolean **signIn(Utente utente)**: verifica se un username è già presente in piattaforma, se non lo è registra il nuovo utente.
- public boolean **login(Utente utente)**: logga l'utente se le credenziali sono corrette.
- public boolean **logout(Utente utente)**: slogga l'utente dal sito.

Per le **storie** e gli **scenari**:

- public boolean **creaNuovaStoria(Storia nuovaStoria)**: permette di creare una nuova storia se il titolo non è già presente nel sito e l'aggiunge alle storie presenti.
- public ArrayList<Storia> **ottieniStorie()**: restituisce tutte le storie presenti nel sito.
- public Storia **ottieniStoria(String nomeStoria)**: restituisce la storia con il titolo passato in input.
- public boolean **eliminaStoria(String nomeStoria)**: elimina la storia con il titolo passato in input.
- public boolean **aggiungiScenarioAScelta(String id, Scenario scenario)**: aggiunge lo scenario passato negli scenari presenti nel sito.
- public boolean **aggiungiScenarioFinale(String id, Scenario scenario)**: aggiunge lo scenario passato negli scenari presenti nel sito.
- public boolean **aggiungiScenarioIndovinello(String id, Scenario scenario)**: aggiunge lo scenario passato negli scenari presenti nel sito.
- public String **contaScenari()**: conta il numero di scenari, necessario per stabilire l'id da assegnare ad uno scenario per la creazione.
- public boolean **settaScenarioIniziale(Scenario scenario)**: setta lo scenario come scenario attuale della storia in cui è presente.
- public boolean **settaCollegamentoSuccessivo(Scenario attuale, String opzione, Scenario daCollegare)**: setta lo scenario successivo di una scelta collegata ad un altro scenario.
- private String **trovaChiavePerScenario(Scenario scenario)**: trova la chiave (id) associata allo scenario.
- public boolean **controlloCollegamenti(ArrayList<Scenario> temp)**: controlla se tutti gli scenari passati in input hanno almeno un precedente.

- private void **salvaNumeroScenariPerStoria(String nomeStoria, boolean verifica)**: salva il numero di scenari nella storia passata in input.
- public String **prossimold()**: restituisce l'id da settare al nuovo scenario che si vuole creare.
- public ArrayList<Scenario> **ottieniScenariStoria(String nomeStoria)**: restituisce tutti gli scenari di una storia passata in input.
- public boolean **modificaScenario(String nomeStoria, Scenario scenarioDaModificare, Scenario scenarioModificato)**: sostituisce, in una storia passata in input, lo scenario vecchio con lo scenario aggiornato.
- public void **eliminaScenari(String nomeStoria)**: elimina tutti gli scenari di una storia.
- public ArrayList<Scenario> **ottieniCollegamentiMancanti(String nomeStoria)**: restituisce tutti gli scenari di una storia che non hanno ancora un precedente.
- public String **contaPartite()**: conta il numero di partite (per settare l'id della prossima partita da creare).
- public Partita **caricaPartita(Storia storia, Utente giocatore, boolean nuovoGioco)**: carica una partita esistente o ne crea una nuova. Utilizza il metodo datiPartita per ottenere tutte le informazioni.
- public Partita **datiPartita(String storia, String utente)**: restituisce tutti i dati di una partita riguardante una specifica storia ed utente.
- public void **eliminaPartita(Storia storia, Utente utente)**: elimina la partita di una specifica storia ed utente.
- public Partita **caricaSuccessivoIndovinello(Partita partita, String risposta)**: carica lo scenario successivo quando si sta giocando una partita e lo scenario successivo è di tipo INDOVINELLO.
- public Partita **caricaSuccessivoScelta(Partita partita, String scelta)**: carica lo scenario successivo quando si sta giocando una partita e lo scenario successivo è di tipo ASCELTA.

Per la **chiusura del server**:

- public void **logoutUtenti()**: slogga tutti gli utenti del sito.

Per i **test**:

- public void **eliminaUtente(String username)**: elimina l'utente con l'username passato in input.

Il **ListenerImpl** contiene due metodi:

- contextInitialized(ServletContextEvent sce): richiama initData all'apertura del server;
- contextDestroyed(ServletContextEvent sce): slogga gli utenti e chiude il db allo spegnimento del Server.

E' possibile definire la gestione dello Shared attraverso il seguente diagramma delle classi:



Manuale dell'utente

Benvenuto in hALA!

Questa applicazione offre l'opportunità di immergersi in avventure narrative coinvolgenti, dove sarai tu a prendere le decisioni chiave che influenzano lo sviluppo della storia.

L'applicazione consente agli utenti di creare, esplorare e giocare una vasta gamma di storie interattive, ognuna con la propria trama e molteplici finali possibili. Sei tu il protagonista delle tue scelte, che determineranno il corso degli eventi e il destino dei personaggi.

Che tu sia un aspirante scrittore desideroso di condividere le tue creazioni con altre persone o un appassionato di narrazioni interattive in cerca di nuove avventure, hALA ti offre un ambiente per esprimere la tua creatività e vivere emozionanti esperienze di gioco.

Di seguito ti spiegheremo rapidamente come sfruttare al meglio hALA e le sue numerose funzionalità.

1. Registrazione e Login

Per utilizzare l'applicazione, è necessario registrarsi creando un account. Ecco come fare:

1. Registrazione

- Per usufruire di tutti i servizi della nostra app è necessario registrarsi.
- Una volta avviata l'app, clicca su "Registrati".
- Inserisci il tuo username e scegli una password.
- Clicca su "SignIn"
- Se l'username non è già in uso, comparirà una scritta di benvenuto e un tasto di reindirizzamento al Login.
- Clicca sul tasto e inserisci di nuovo le credenziali appena create. Verrai portato al Menu Principale.

2. Login

- Se hai già creato un account e vuoi accedere all'app con questo, effettua il login.
- Una volta avviata l'app, clicca su "Accedi".
- Inserisci username e password e clicca su Login. Verrai portato al menu principale.

3. Logout

- Se ti trovi nel menu principale con un account, ma vuoi uscire per cambiarlo, effettua il Logout.
- Clicca sul tasto in basso con su scritto “LogOut”.
- Clicca di nuovo sul tasto “Esegui Logout”. Verrai reindirizzato al menu iniziale.

2. Esplorazione delle Storie

In hALA ci sono due modi di visionare le storie. Dopo aver effettuato l'accesso, puoi iniziare a esplorare quelle disponibili nel catalogo e giocarci dopo averle selezionate. Puoi anche visualizzare solamente le storie senza parteciparvi se non hai eseguito il login.

Visualizzazione:

- Dal menu iniziale clicca su “Visualizza Storie”.
- Potrai visionare il catalogo con tutte le storie presenti al momento in hALA, l'autore, il numero di scenari presenti, ed ispezionare lo scenario iniziale della storia. Tuttavia, non avendo effettuato il login, non potrai iniziare a giocarci.
- Puoi anche ricercare le storie secondo alcuni filtri nel menu a tendina: cerca per nome autore, numero scenari, o titolo della storia. Una volta inserita la parola chiave nella casella di testo, clicca su “Ricerca”, e ti verrà mostrato il risultato.

Visualizzazione e Gioco:

- Accedi al menu iniziale ed esegui il Login.
- Una volta eseguito l'accesso, verrai portato al menu principale.
- Da qui, cliccando sul tasto “Visualizza le Storie” accederai alla stessa schermata descritta in precedenza, ma cliccando su una storia ed in seguito sul tasto “Gioca”, inizierai l'avventura! Leggi i dettagli sul gioco in fondo al manuale.

3. Scrittura di una Storia

Se desideri creare la tua storia, segui questi passaggi:

- Dalla tua home page, fai clic su “Scrivi una Storia”.
- Segui le istruzioni per creare gli scenari e le scelte che formeranno la trama della tua storia: Prima di tutto sceglierai il titolo della storia, poi scriverai gli scenari (decidendo tipo, descrizione, domanda da porre al giocatore, eventuali oggetti raccogliibili nello scenario, e scelte da compiere), e infine li potrai collegare tra loro cliccando su “Crea Collegamenti”, tasto all'estrema destra dello schermo. In

questa schermata sceglierai prima di tutto lo scenario iniziale, dopodiché potrai creare i vari collegamenti tra gli scenari, fino ad arrivare agli scenari finali.

- Esistono tre tipi di scenari: a scelta, ad indovinello e scenari finali. Nei primi non servirà altro che porre una scelta da compiere al giocatore, in base alla quale entrerà in un diverso scenario. Puoi scegliere in fase di creazione delle scelte se per eseguirle è necessario un oggetto che dovrà aver raccolto precedentemente. Nei secondi, andrai a scrivere un indovinello al quale il player dovrà rispondere, quindi andrà in uno scenario se risponde correttamente, o in un altro se risponde in modo errato. In base alle sue scelte arriverà ad uno scenario finale dove si concluderà il suo percorso.

4. Modifica di una Storia

- È possibile modificare solo il testo degli scenari, non la struttura delle scelte.
- Una volta loggato, dalla home page, clicca su “Modifica una Storia” e apporta le modifiche necessarie ai testi nelle storie che hai scritto tu. Non è possibile modificare storie scritte da altri utenti.

5. Giocare ad una Storia

Una volta che hai scelto una storia da giocare, segui questi passaggi:

1. Inizia a giocare:

- Fai clic sulla storia selezionata dal catalogo, poi clicca su “Gioca”.
- Leggi il testo dello scenario iniziale e scegli una delle opzioni disponibili.
- In alcuni scenari avrai la possibilità di raccogliere degli oggetti che andranno nel tuo inventario. L’inventario è unico per ogni storia.
- In base alle tue scelte, verrai portato a nuovi scenari. Se si tratta di uno scenario a indovinello dovrai dare una risposta e la trama si muoverà in base alla correttezza o meno della risposta. Se si tratta di uno scenario a scelta dovrai solo prendere una delle decisioni possibili. Ricorda, ci sono delle scelte che potrai prendere solo se sei in possesso di un oggetto che potresti aver raccolto in precedenza!
- Continua a prendere decisioni fino a raggiungere uno dei finali della storia.

2. Interrompi e ricomincia:

- Puoi interrompere una partita in qualsiasi momento e riprenderla più tardi. Quando sceglierai di giocare di nuovo, ti verrà chiesto se riprendere dall’ultimo salvataggio oppure ricominciare da capo. In questo ultimo caso non potrai più recuperare i dati del salvataggio precedente, in quanto sarà sovrascritto.

Ora sei pronto, Buon divertimento!

Alessia, Lorenzo, Altea

Diario del Progetto

SPRINT 0

Dal 12/02 al 18/02

Ruoli: Alessia (Developer), Altea (Scrum Master + dev), Lorenzo (Product Owner)

In quanto Sprint 0 e fase di Inception, i Ruoli non sono strettamente rispettati, poiché lavoriamo insieme per allinearci con la strumentazione e per pensare a come muoverci.

12/02/2024

Daily Scrum

Incontro in presenza tra tutti i membri del gruppo.

Configurazione Eclipse con GitHub, GWT, e MAVEN.

Fine Daily Scrum

14/02/2024

Daily Scrum

Creiamo i diagrammi dei casi d'uso e delle classi e le descrizioni testuali dei casi d'uso.

Fine Daily Scrum

16/02/2024

Daily Scrum

Siccome lavoreremo a uno stack nuovo per tutti, iniziamo a leggere la documentazione di gwt per capire i concetti generali e di conseguenza come iniziare a lavorare.

Fine Daily Scrum

SPRINT 1 - Dal 19/02 al 25/02

Ruoli: Lorenzo (Developer), Alessia (Scrum Master + dev), Altea (Product Owner)

19/02/2024

Sprint Planning

Si vogliono realizzare i seguenti casi d'uso: Login, Logout e Signin.

Primo giorno del primo sprint, viene compilato il product backlog e lo sprint backlog.

I due developer (Alessia e Lorenzo) decidono di procedere realizzando Task per gli UC.

Issue: Use Case 5 - Sign In

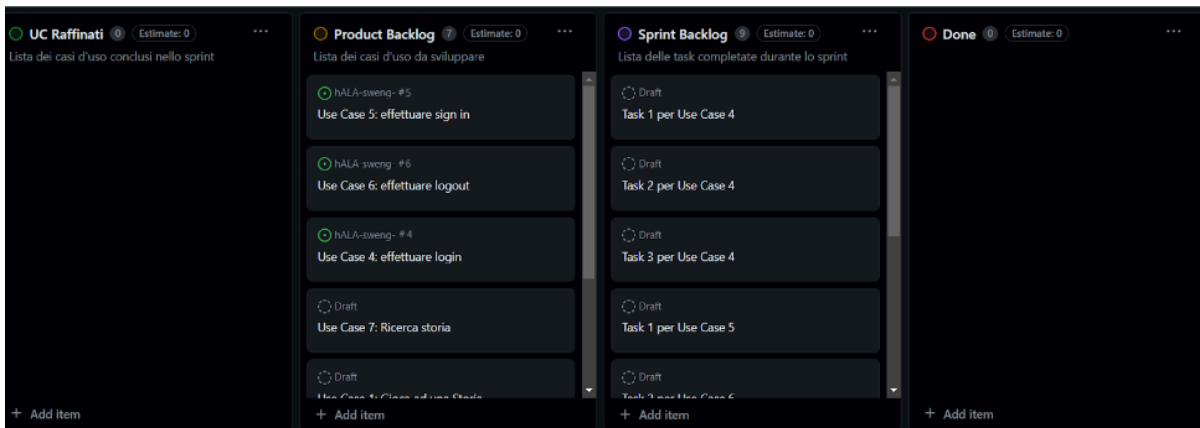
- Task 1: realizzazione schermata grafica; Ore previste: 1
- Task 2: realizzazione parte logica tra client/server; Ore previste: 1
- Task 3: gestione dei dati degli utenti; Ore previste: 1

Issue: Use Case 4 - Login

- Task 1: realizzazione schermata grafica; Ore previste: 1
- Task 2: realizzazione parte logica tra client/server; Ore previste: 2
- Task 3: gestione dei dati degli utenti; Ore previste: 1

Issue: Use Case 6 - Logout

- Task 1: realizzazione schermata grafica; Ore previste: 1
- Task 2: realizzazione parte logica tra client/server; Ore previste: 1
- Task 3: gestione dei dati degli utenti; Ore previste: 1



Fine Sprint Planning

Daily Scrum

Si iniziano a realizzare le classi base del progetto seguendo la documentazione aggiornata di GWT su MyWebApp.

In seguito si riscontrano numerose problematiche in quanto non capiamo bene come ragionare e come costruire la base del progetto (soprattutto la comunicazione tra client e server attraverso lo shared).

Fine Daily Scrum

20/02/2024

Previsto incontro col tutor, presenti tutti i membri del gruppo. Dalla conversazione vengono chiariti alcuni dubbi.

Daily Scrum

I developer dopo aver discusso con il tutor e comprese le problematiche che stavano affrontando riescono ad iniziare a lavorare al progetto in modo più consapevole.

Si dividono le Task successive:

- Signin (T1, T2, T3)
- Login(T1, T2, T3)

Fine Daily Scrum

21/02/2024

Daily Scrum

I developer si aggiornano sul lavoro svolto il giorno precedente e sulle problematiche riscontrate, ovvero l'errore dovuto all'apertura del File e una

piccola complicazione grafica per il tasto di Ritorno che si prevede di risolvere nel giorno successivo (la dicitura sopra il bottone).

Fine Daily Scrum

22/02/2024

Daily Scrum

Alessia e Lorenzo decidono di risolvere il problema del tasto di Ritorno.

Si discute inoltre un nuovo problema dovuto alla gestione dei dati.

E' opportuno trasformare l'HashMap<String,String> in <String,Utente> così da mantenere tutti i dati relativi all'utente e di poterli costantemente aggiornare nel file json e nel database.

Fine Daily Scrum

23/02/2024

Daily Scrum

I due developer si aggiornano sul lavoro del giorno precedente. La modifica sui dati, invece, risulta complessa in quanto è necessaria una Serializzazione di Utente e, nonostante i tentativi, continuano a risultare errori. Le prove non sono state committate per mantenere su github un codice funzionante. Ci si pone come obiettivo quello di risolvere il problema della gestione dei dati, svolgere Logout (T1, T2, T3) e terminare Login (T3).

Fine Daily Scrum

24/02/2024

Daily Scrum

I developer si aggiornano sul lavoro svolto nella giornata precedente.

Lorenzo aveva realizzato delle modifiche sui metodi di Signin e Login in GreetingServiceImpl, ora prendono in input, invece di (username, password), (Utente).

Alessia aveva risolto il problema dell'HashMap trasformandolo in un Map<String, Utente> rendendo Utente Serializable (implements Serializable) e creando un costruttore vuoto (prova già svolta ieri ma senza successo). In seguito apporta le modifiche necessarie a SignIn, Login e a ConvertToJson così da mantenere i dati aggiornati come si era stabilito.

Fine Daily Scrum

25/02/2024

Sprint Review

Inizialmente Alessia e Lorenzo si incontrano per valutare il lavoro svolto durante lo Sprint. In seguito Alessia e Lorenzo aggiornano Altea (product owner) sul lavoro svolto durante lo sprint.

Issue: Use Case 5 - Sign In

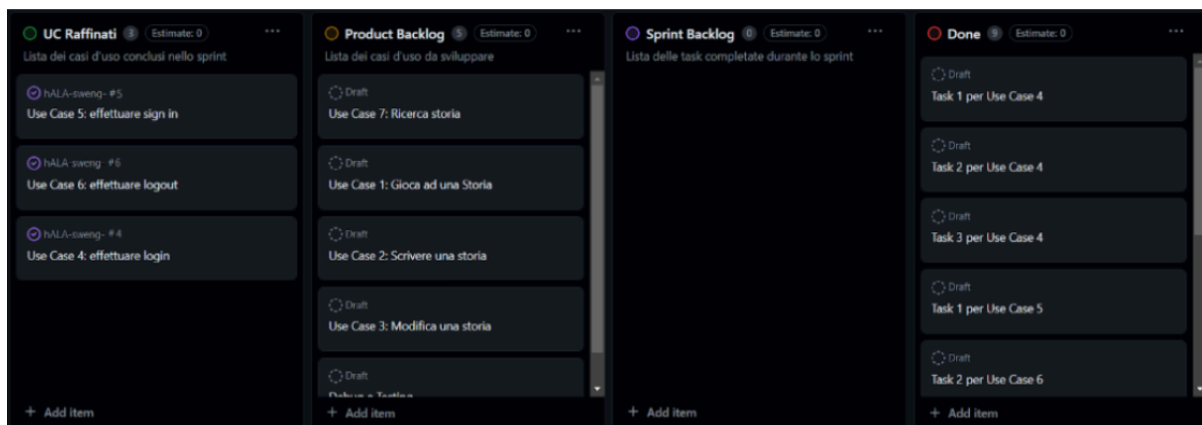
- Task 1: realizzazione schermata grafica; Ore effettive: 1
- Task 2: realizzazione parte logica tra client/server; Ore effettive: 2
- Task 3: gestione dei dati degli utenti; Ore effettive: 1

Issue: Use Case 4 - Login

- Task 1: realizzazione schermata grafica; Ore effettive: 1
- Task 2: realizzazione parte logica tra client/server; Ore effettive: 3
- Task 3: gestione dei dati degli utenti; Ore effettive: 1

Issue: Use Case 6 - Logout

- Task 1: realizzazione schermata grafica; Ore effettive: 1
- Task 2: realizzazione parte logica tra client/server; Ore effettive: 1
- Task 3: gestione dei dati degli utenti; Ore effettive: 1



Fine Sprint Review

Sprint retrospective

Partecipano tutti i membri del gruppo, lo scopo è trattare dei seguenti argomenti:

- Cosa è andato bene durante lo Sprint precedente: la scrittura del codice è stata efficiente e veloce
- Cosa possiamo migliorare: affrontare problemi soprattutto migliorando il coordinamento del gruppo e della scrittura del codice, gestire commit e push
- Cosa ci impegniamo a migliorare nel prossimo Sprint: lavoro di gruppo in generale

Fine Sprint retrospective

SPRINT 2 - Dal 26/02 al 3/03

Ruoli: Altea (Scrum Master + dev), Lorenzo (Developer), Alessia (Product Owner)

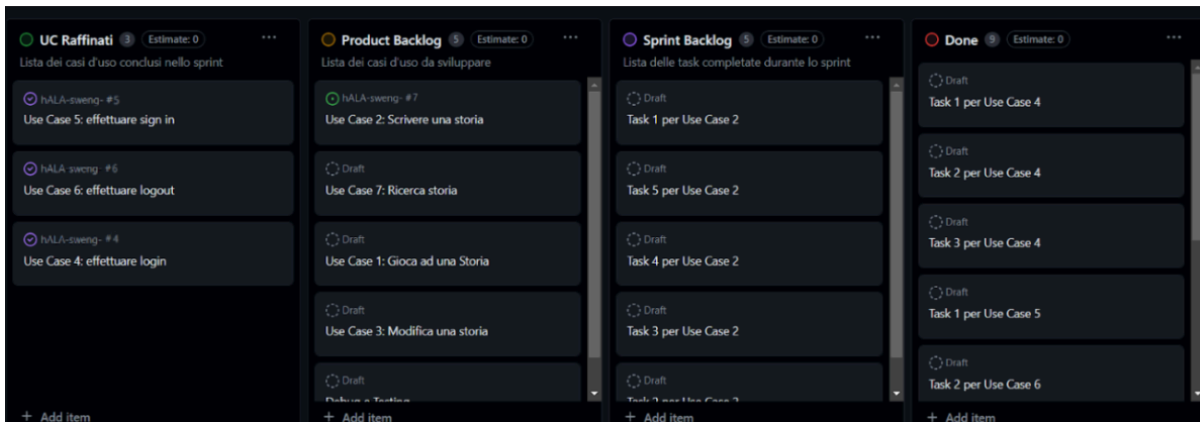
26/02/2024

Sprint Planning

UC 2: Creazione Storia

- Task 1: bottone ScriviStoria + gestione ScriviStoria.java (si attiva solo se è loggato); ore previste: 2
- Task 2: shared creazione classe Storia + interfaccia; ore previste: 1
- Task 3: shared creazione classe Scenario + interfaccia; ore previste: 1
- Task 4: gestione dati (creazione database, json "Storie", "Scenari"); ore previste: 5

- Task 5: gestione metodi nel server per la scrittura della storia; ore previste: 5



Fine Sprint Planning

27/02/2024

Daily Scrum

Si continua a lavorare al progetto. Prima di farlo ci coordiniamo per dividerci il lavoro come accordato nell'ultimo sprint retrospective.

A fine giornata abbiamo creato le classi storia e scenario e le due relative interfacce. Inoltre lavoriamo ai file html delle classi per gestire l'aspetto estetico.

Fine Daily Scrum

28/02/2024

Daily Scrum

Riunione tra i developer. Si inizia a pensare alla grafica del progetto. Optiamo per non modificare la grafica direttamente dalle classi java per rispettare i principi SOLID e quindi lasciare le classi con meno assi di cambiamento possibili, quindi iniziamo a testare l'aggiunta di classi aggiuntive (class.ui.xml) collegate alle classi java per gestire la grafica. Inoltre iniziamo a gestire la scrittura delle storie ed il salvataggio. Vengono aggiornate le interfacce di logout e signin, ora sono quasi tutte uniformate.

Fine Daily Scrum

29/02/2024

Daily Scrum

Si continua a lavorare parallelamente alla gestione grafica poiché utile anche per la logica del progetto e anche alla logica delle storie.

Fine Daily Scrum

01/03/2024

Daily Scrum

Modifichiamo le classi scenario per migliorare la gestione delle storie.

Ipotizziamo un metodo per creare le storie e per salvarle con gli scenari in ordine. Troviamo qualche difficoltà nella modifica degli scenari, in quanto le modifiche che abbiamo in mente sono abbastanza grosse.

Troviamo il modo di salvare i json degli scenari in un file.

Fine Daily Scrum

02/03/2024

Daily Scrum

Altea lavora alla grafica della facciata per fare i collegamenti dopo aver chiarito con Lorenzo come gestire la cosa;

Altea e Lorenzo si allineano sul lavoro che ha fatto Altea e prosegue Lorenzo cercando di sincronizzare i collegamenti col server. Purtroppo si è fatta fatica a trovare una soluzione per cui bisognerà lavorarci anche il giorno dopo.

Si continua a lavorare sui collegamenti e come gestirli tra client, server e json, anche oggi si è dovuta concludere la giornata senza una soluzione definitiva, ma comunque è stato portato avanti e migliorato il lavoro.

Fine Daily Scrum

03/03/2024

Sprint review:

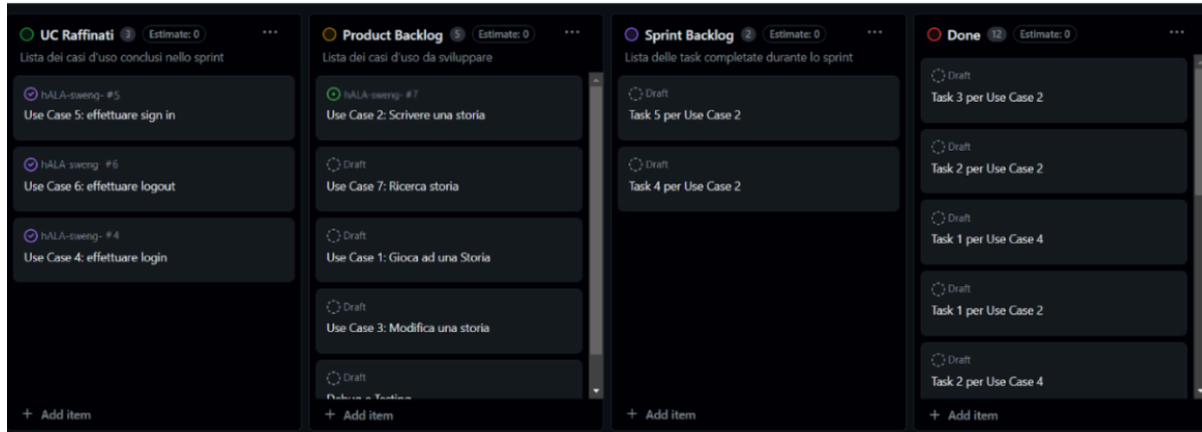
Altea e Lorenzo mostrano ad Alessia il lavoro svolto.

Non è stato possibile concludere tutte le task.

Issue: Use Case 2: Creazione Storia

- Task 1: bottone ScriviStoria + gestione ScriviStoria.java(si attiva solo se è loggato); ore effettive: 5
- Task 2: shared creazione classe Storia e interfaccia; ore effettive: 3
- Task 3: shared creazione classe Scenario + interfaccia; ore effettive: 5

- Task 4: gestione dati (creazione database, json “Storie”, “Scenari”); ore effettive: 8 (non concluso)
- Task 5: gestione metodi nel server per la scrittura della storia; ore effettive: 8 (non concluso)



Fine Sprint Review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: lavoro di gruppo, pair programming
- Cosa possiamo migliorare: affrontare problemi
- Cosa ci impegniamo a migliorare nel prossimo Sprint: velocità nella risoluzione dei problemi

Fine Sprint retrospective

SPRINT 3 - dal 4/03 al 10/03

Ruoli: Alessia (developer e scrum master), Lorenzo (Product owner), Altea (developer)

04/03/2024

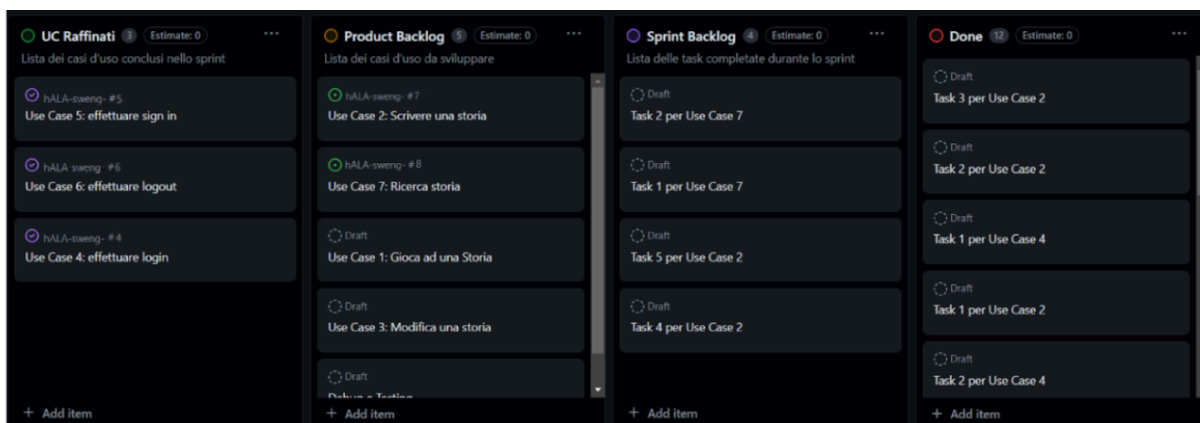
Sprint Planning

Issue: Use Case 2: Creazione Storia

- Task 4: gestione dati (creazione database, json “Storie”, “Scenari”); ore previste: 8 (+ rimanenti 2 = 10 totali)
- Task 5: gestione metodi nel server per la scrittura della storia; ore previste: 8 (+ rimanenti 2 = 10 totali)

Issue: Use Case 7: Ricerca Storia

- Task 1: Creazione Interfaccia Ricerca; ore previste: 1
- Task 2: Creazione Logica di ricerca nel database; ore previste: 8



Fine Sprint Planning

Daily Scrum

Bisogna risolvere qualche problema nella scrittura nei json, perché i metodi funzionano correttamente ma manca la scrittura degli scenari precedenti e successivi.

Fine Daily Scrum

05/03/2024

Daily Scrum

Vengono riscontrati numerosi problemi di comunicazione tra il Client e il Server, sia riguardante la creazione della storia sia degli scenari. Questi

problemi sono risultati solo successivamente nel momento in cui si è cercato di prelevare i dati per fare i collegamenti (ma i dati non c'erano!!!).

Dopo tante ore riusciamo a risolvere i problemi grazie a stampe di debug.

Aggiunti gli ID, il settaggio di IDPrecedenti e IdSuccessivi.

Problematica riscontrata: apertura/chiusura del database. (Risolta in serata)

Obiettivi per oggi/domani: eliminare stampe di debug, sistemare la grafica di Collegamenti, effettuare controlli sul codice, iniziare UC Visualizza Storia

Fine Daily Scrum

06/03/2024

Daily Scrum

Eliminate stampe di debug.

Iniziato VisualizzaStoria (Task 1): realizzazione tabella + aggiunta campo

NumeroScenari nelle storie.

Fine Daily Scrum

07/03/2024

Daily Scrum

Si continua a lavorare alla connessione degli scenari.

Realizzato il filtro nella tabella per la ricerca delle storie.

Fine Daily Scrum

08/03/2024

Daily Scrum

Si lavora alla risoluzione di alcuni bug sullo scenario iniziale.

Realizzata la colonna con il testo dello scenario nella tabella per la ricerca delle storie. Realizzazione dei sort per i campi nome autore e numero scenari.

Fine Daily Scrum

09/03/2024

Daily Scrum

Ci siamo occupati della sistemazione dell'estetica della applicazione web e iniziato la logica per la creazione dello scenario finale, avendo avuto poi dei problemi si è proseguito il giorno successivo.

Fine Daily Scrum

10/03/2024

Daily Scrum

Risolti problemi sui collegamenti, gestione scenario finale, gestione mappa {scelta, indice successivo}.

Fine Daily Scrum

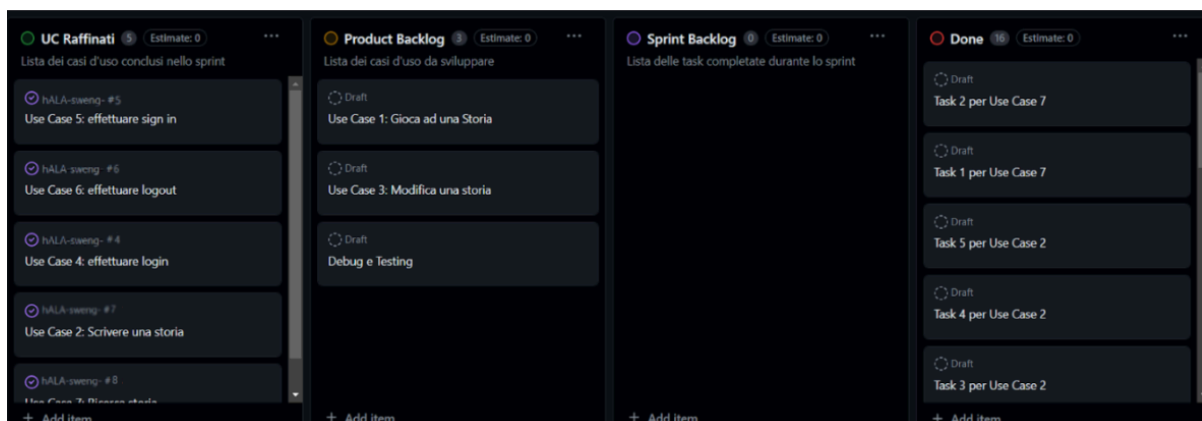
Sprint review

Issue: Use Case 2: Creazione Storia

- Task 4: gestione dati (creazione database, json “Storie”, “Scenari”); ore effettive: 10
- Task 5: gestione metodi nel server per la scrittura della storia; ore effettive: 10

Issue: Use Case 7: Ricerca Storia

- Task 1: Creazione Interfaccia Ricerca; ore effettive: 2
- Task 2: Creazione Logica di ricerca nel database; ore effettive: 12



Fine sprint review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: lavoro di gruppo, pair programming
- Cosa possiamo migliorare: coordinamento
- Cosa ci impegniamo a migliorare nel prossimo Sprint: coordinamento scrivendoci più spesso anche per problemi minori

Fine sprint retrospective

SPRINT 4 - dal 11/03 al 17/03

Ruoli: Lorenzo (developer e scrum master), Altea (Product owner), Alessia (developer)

11/03/24

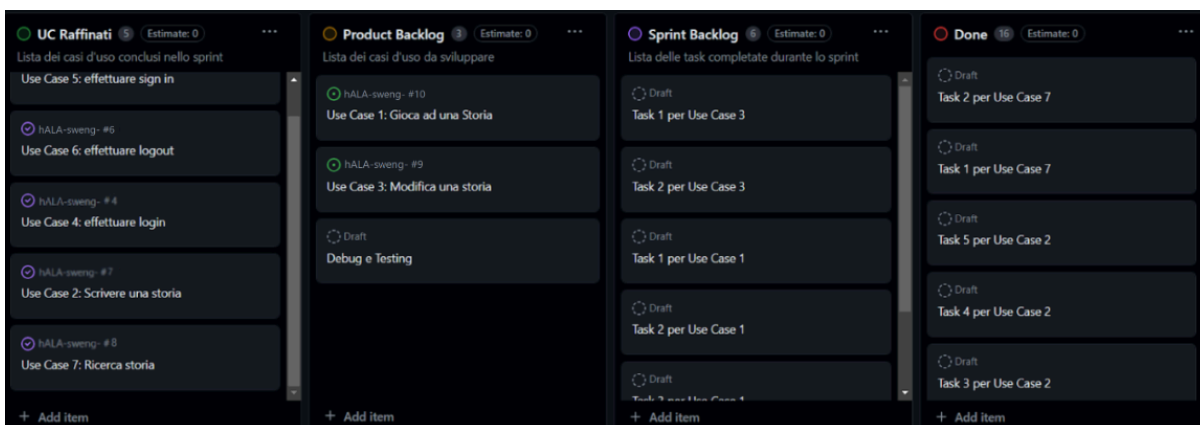
Sprint Planning

Issue: Use Case 3 - Modifica Storia

- Task 1: Realizzazione parte grafica che permette di modificare le storie; ore previste: 2
- Task 2: logica del server per modificare le storie; ore previste: 10

Issue: Use Case 1 - Gioca Storia

- Task 1: Implementazione parte grafica; ore previste: 6
- Task 2: Metodi lato Client; ore previste: 8
- Task 3: Metodi lato Shared; ore previste: 1
- Task 4: Metodi lato Server; ore previste: 4



Fine Sprint Planning

Daily scrum

Obiettivo: debug della parte grafica. In particolare:

- Togliere schermata oggetti raccogliibili dalla creazione dello scenario finale.
- Settare il default del Listbox all'apertura della schermata dei collegamenti.
- non ha dato errore dopo aver provato a creare uno scenario a scelta senza scelta e provato a creare una scelta con opzioni vuote, controllare.
- gestire il caso se, nei collegamenti, viene impostato come primo scenario uno finale.

Si inizia a lavorare alle task dello Sprint 4 da domani.

Fine Daily Scrum

12/03/24

Daily scrum

Vengono effettuate le seguenti modifiche: aggiunto controllo sull'inserimento degli oggetti raccogliibili nello scenario, iniziata la logica e scritta la schermata di modifica storia, gestione logout migliorata, modificata grafica e messaggio della schermata ricerca storie, modificata la grafica per rendere l'applicazione più bella.

Trovato bug nel tasto back di ricerca storie: funziona solo la prima volta dopo riporta a home page anche se non si ha già fatto il logout.

Iniziata modifica storia.

Fine Daily Scrum

13/03/2024

Daily Scrum

Si conclude la modifica delle storie e degli scenari. Sono stati realizzati dei test. Questo ci ha fatto trovare dei bug, come un errore di gestione della domanda dell'indovinello che causava il non salvataggio di quella, quindi si è proceduto subito a risolverlo.

Sono proseguiti i test con risultati positivi.

A questo punto si è iniziato a discutere su come implementare le prime task dello Use Case "gioca storia".

Fine Daily Scrum

14/03/2024

Daily Scrum

I due developer hanno iniziato ad implementare la gestione di gioca storia e hanno notato piccoli errori nella sezione collegamenti per i dati relativi al settaggio dello scenario iniziale della Storia. Si lavora ai nuovi task di use case 1 “gioca storia”. Viene realizzata una parte della prima schermata.

Fine Daily Scrum

15/03/2024

Daily Scrum

Realizzazione metodi per gestire il caricamento della partita, ma si pensa che sarebbe opportuno dividere nel server il caricamento dei dati con l'aggiornamento della partita.

Fine Daily Scrum

16/03/2024

Daily Scrum

Creazione metodi per eliminare le partite, caricarle e ottenere i dati. Gestione degli oggetti sbloccabili corretta. Correzioni grafiche. Correzione sezione vecchia dei collegamenti per non permettere collegamenti tra uno scenario e se stesso.

Fine Daily Scrum

17/03/2024

Daily Scrum

Viene ultimata l'implementazione dello Use Case Gioca Storia. Le parti funzionali del progetto sono state completate. Nel prossimo sprint si faranno test per provare a migliorare l'usabilità e la facilità di comprensione del progetto.

Fine Daily Scrum

Sprint review

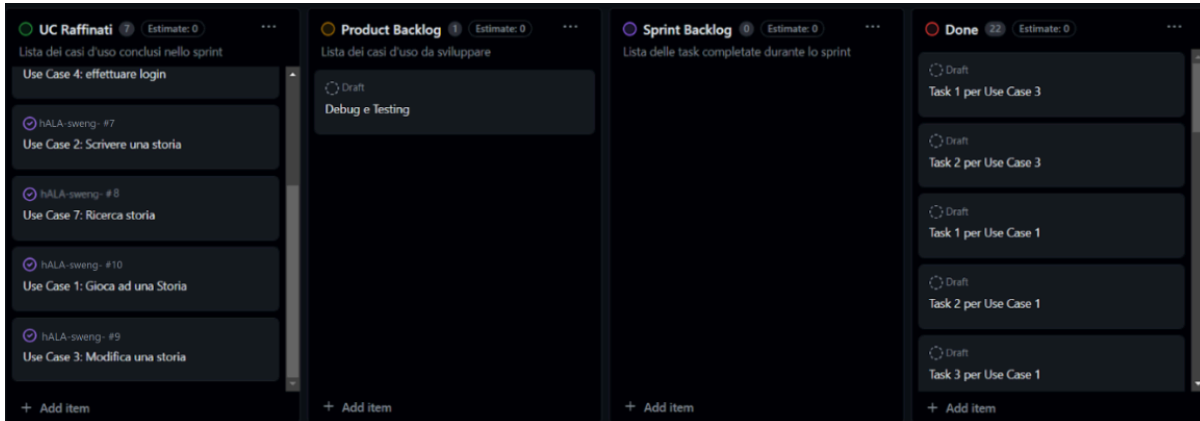
Issue: Use Case 3 - Modifica Storia

- Task 1: Realizzazione parte grafica che permette di modificare le storie; ore effettive: 2
- Task 2: logica del server per modificare le storie; ore effettive: 6

Issue: Use Case 1 - Gioca Storia

- Task 1: Implementazione parte grafica; ore effettive: 5

- Task 2: Metodi lato Client; ore effettive: 8
- Task 3: Metodi lato Shared; ore effettive: 1
- Task 4: Metodi lato Server; ore effettive: 4



Fine sprint review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: abbiamo diviso bene gli incarichi e i momenti di lavoro alternandoci in modo efficiente nella scrittura del codice, e di conseguenza abbiamo concluso rapidamente le task da fare.
- Cosa possiamo migliorare: Condivisione idee su risoluzione problemi
- Cosa ci impegniamo a migliorare nel prossimo Sprint:
Comunicazione per condivisione di idee

Fine sprint retrospective

SPRINT 5 - dal 18/03 al 24/03

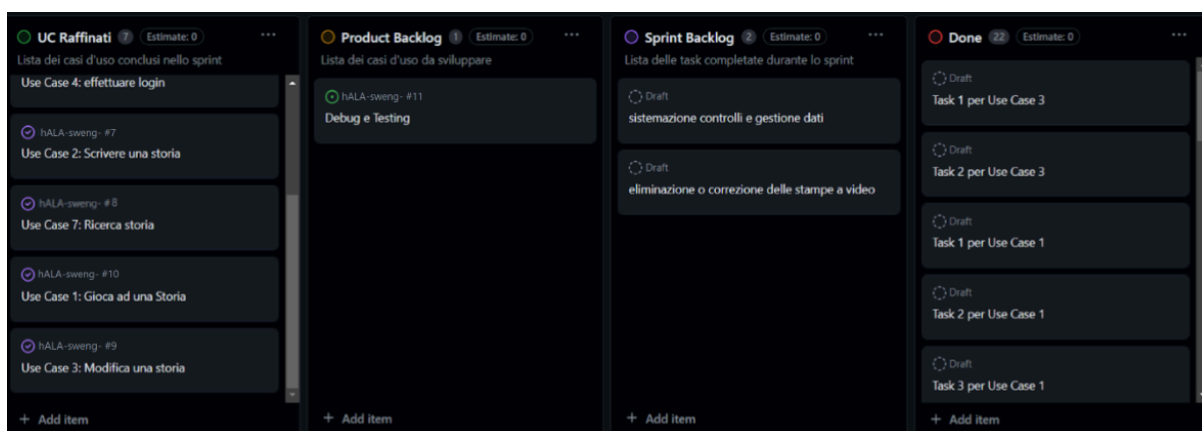
Ruoli: Altea (developer e scrum master), Alessia (Product owner), Lorenzo (developer)

18/03/2024

Sprint Planning

Issue: Pulizia Codice e Testing

- Task 1: eliminazione e/o correzione delle stampe a video; ore previste: 5
- Task 2: sistemazione controlli e gestione dati; ore previste: 6



Fine Sprint Planning

20/03/2024

Daily Scrum

Iniziamo ad effettuare i test con Jacoco. Ci rendiamo conto di un problema di gestione della sessione causato dall'uso di una variabile Utente nel server. Dobbiamo sistemare questo problema che porterà a diversi cambiamenti nelle classi.

Fine Daily Scrum

22/03/2024

Daily Scrum

Viene conclusa una prima fase di testing e di modifiche al codice per rimediare ad alcuni errori emersi grazie al testing. Riprendiamo quindi a lavorare sul main branch, chiudendo la issue precedente ed aprendone una nuova sempre per fare test e debug.

In seguito a questo avvenimento ci rendiamo conto dell'importanza di eseguire test man mano che si scrive codice e non solo alla fine, in modo da riscontrare gradualmente degli errori se sono presenti nel codice.

Fine Daily Scrum

23/03/2024

Daily Scrum

Si prosegue il testing e il debug del codice parallelamente ad alcune modifiche grafiche.

Fine Daily Scrum

24/03/2024

Daily Scrum

mywebapp-server												
mywebapp-server												
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.hala.mywebapp	<div><div></div></div>	95%	<div><div></div></div>	77%	50	151	32	425	3	42	1	2
Total	75 of 1.785	95%	48 of 218	77%	50	151	32	425	3	42	1	2

Si conclude il codice del testing ottenendo una buona coverage (95%). Per concludere questa fase vengono modificati alcuni metodi nel server. Inoltre si continua la fase di pulizia del codice.

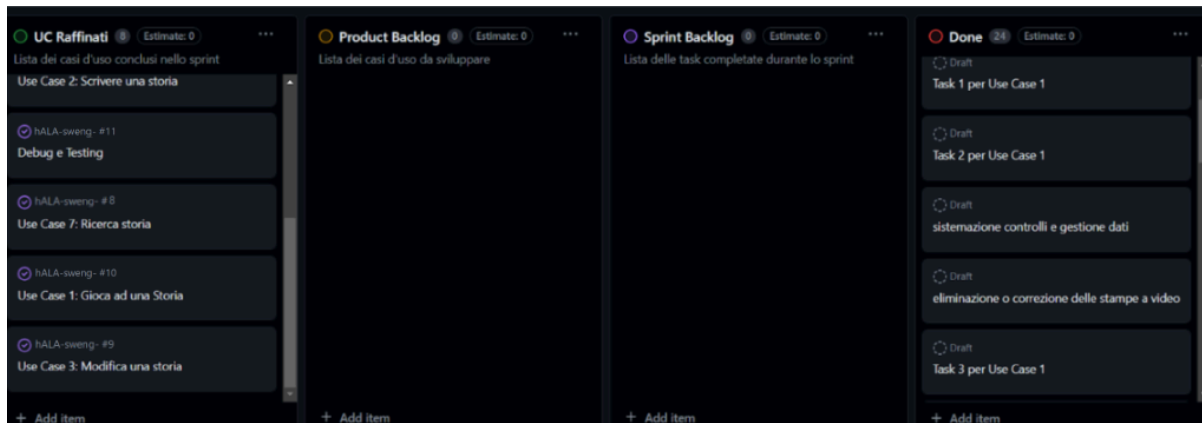
Essendo l'ultimo giorno dello sprint si pensa a organizzare la prossima iterazione. Inoltre, si raccolgono le restanti task da fare in vista della consegna, da cui mancano due settimane, ovvero due sprint.

Fine Daily Scrum

Sprint review

Issue: Pulizia Codice e Testing

- Task 1: eliminazione e/o correzione delle stampe a video; ore effettive: 5
- Task 2: sistemazione controlli e gestione dati; ore effettive: 8



Fine sprint review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: i ruoli sono stati rispettati, si procede bene col tempo a disposizione
- Cosa possiamo migliorare: continuare con lo stesso ritmo per concludere anticipatamente per la consegna
- Cosa ci impegniamo a migliorare nel prossimo Sprint: proseguire con le task

Fine sprint retrospective

SPRINT 6 - dal 25/03 al 30/03

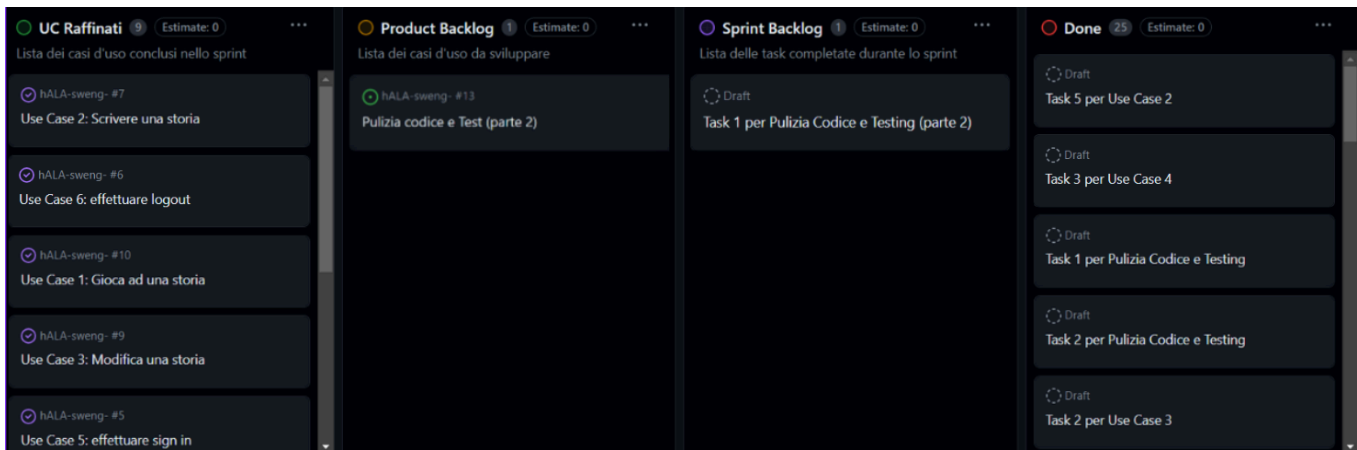
Ruoli: Alessia (developer e scrum master), Lorenzo (Product owner), Altea (developer)

25/03/2024

Sprint Planning

Issue: Debug e Pulizia Codice (Parte 2)

- Task 1: Pulizia del Codice (eliminare le classi inutili, eliminare le stampe di debug, sistemare leggibilità del codice); ore previste: 8




Fine Sprint Planning

26/03/2024

Daily Scrum

Si prosegue il testing e il debug del codice. La modifica attuale del testing porta alla seguente coverage:

mywebapp-server

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 com.hala.mywebapp	<div><div></div></div>	94%	<div><div></div></div>	81%	29	130	33	430	2	47	0	2
Total	94 of 1.731	94%	31 of 166	81%	29	130	33	430	2	47	0	2

Fine Daily Scrum

27/03/2024

Daily Scrum

Si continua a testare l'app.

Fine Daily Scrum

28/03/2024

Daily Scrum

Si prosegue la scrittura della documentazione finale in parallelo alle ultime modifiche al codice.

Fine Daily Scrum

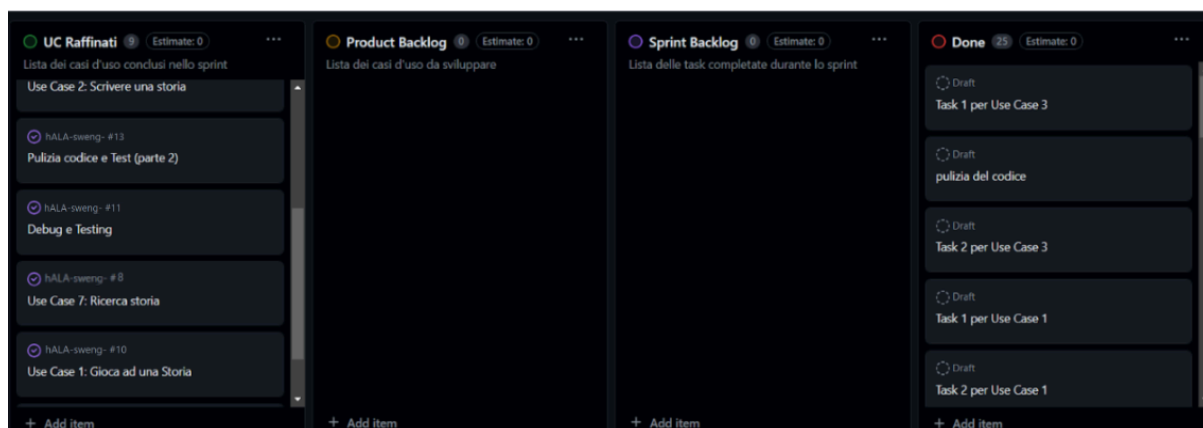
30/03/2024

Si chiude lo sprint il giorno prima di Pasqua in quanto non si lavorerà al codice nel giorno di Festa.

Sprint review

Issue: Pulizia Codice e Testing (parte 2)

- Task 1: Pulizia del Codice (eliminare le classi inutili, eliminare le stampe di debug, sistemare leggibilità del codice); ore effettive: 8



Fine sprint review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: i tempi sono stati rispettati
- Cosa possiamo migliorare: controllare più a fondo possibile il codice per fare debug
- Cosa ci impegniamo a migliorare nel prossimo Sprint: divisione delle parti del codice per debug

Fine sprint retrospective

SPRINT 7 - dal 01/04 al 06/04

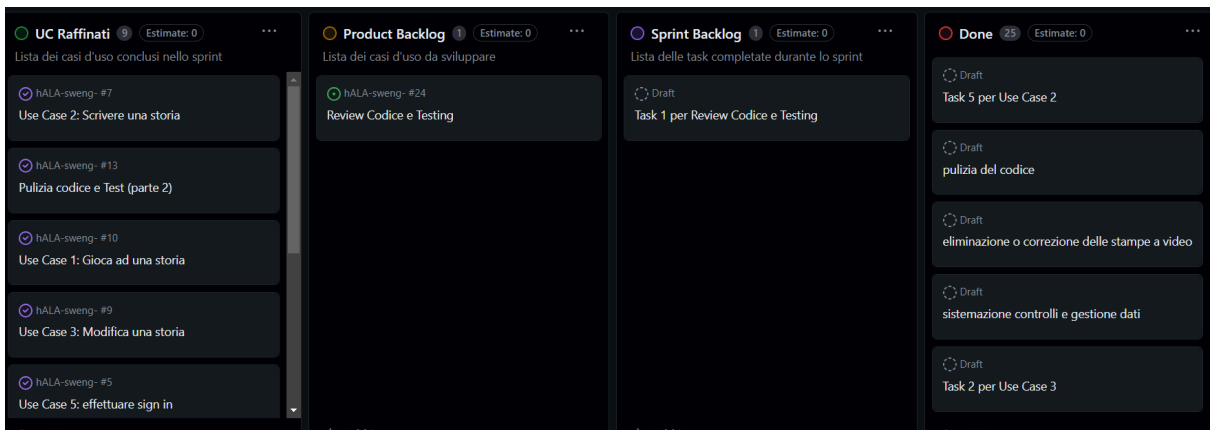
Ruoli: Lorenzo (developer e scrum master), Altea (Product owner), Alessia (developer)

02/04/2024

Sprint Planning

Issue: Review Codice e Testing

- Task 1: Review Codice e Testing; ore previste: 10



Fine Sprint Planning

03/04/2024

Daily Scrum

Si prosegue con l’aggiornamento della classe di testing e con la review finale del codice.

La modifica attuale del testing porta alla seguente coverage:

mywebapp-server												
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.hala.mywebapp	<div><div></div></div>	95%	<div><div></div></div>	83%	26	128	30	434	1	47	0	3
Total	72 of 1.730	95%	26 of 162	83%	26	128	30	434	1	47	0	3

Fine Daily Scrum

04/04/2024

Daily Scrum

Si prosegue con la review finale del codice.

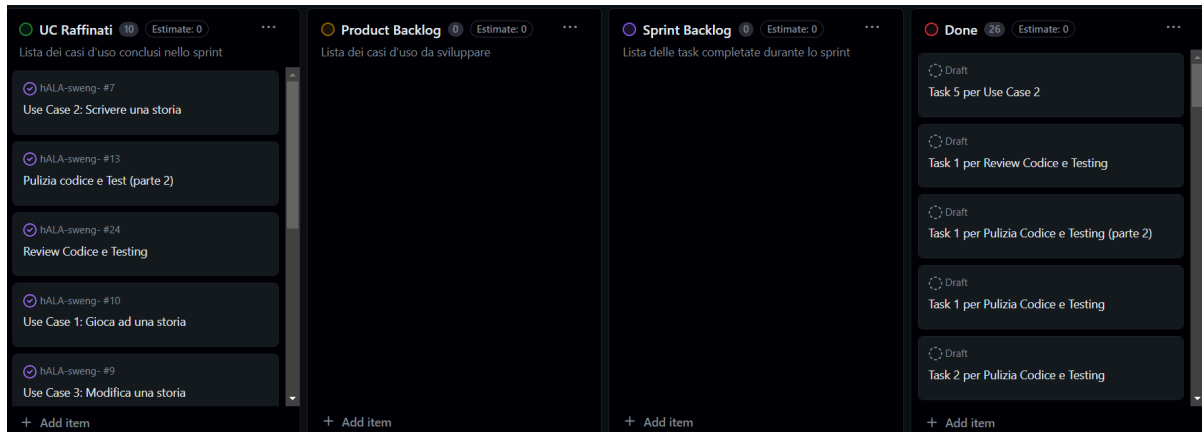
Fine Daily Scrum

06/04/2024

Sprint review

Issue: Review Codice e Testing

- Task 1: Review Codice e Testing; ore effettive: 10



Fine sprint review

Sprint retrospective

- Cosa è andato bene durante lo Sprint precedente: divisione lavoro
- Cosa possiamo migliorare: nd
- Cosa ci impegniamo a migliorare nel prossimo Sprint: nd

Fine sprint retrospective