

# Sistemi informativi su web 2017

Progetto personalizzato :

Kayak è un framework a supporto della fase di data preparation all'interno di un data lake con primitive ad-hoc in grado di permettere alla figura del data scientist di ridurre e agevolare tutto il relativo processo mediante l'impostazione di un parametro ( tolerance dell'utente) rappresentante il tempo di attesa dell'utente rispetto al quale esso riceverà i risultati dei task sottomessi secondo una strategia di esecuzione incrementale per produrre informazioni sempre più accurate mediante previews.

Si vuole realizzare un sistema informativo che consenta ad un insieme di utenti di accedere all'applicazione.

Il dominio dell'applicazione sostanzialmente di compone di utenti in grado di sottomettere al sistema diverse pipeline custom che possono essere salvate, resumed, estese ed eseguite. Esse sono composte da un grafo aciclico orientato di task. Il sistema informativo fornisce tutte le operazioni per la loro creazione e modifica. L'utente può quindi scegliere i task da inserire da una lista e avviare la pipeline di lavoro.

Il sistema in discussione è usato principalmente da due tipi di attori : amministratori del sistema, per gestire informazioni sugli utenti e sulle relative pipeline sottomesse e gli utenti che utilizzano il sistema creando pipeline...

L'utente :

- Ha un profilo personale dove può specificare alcune preferenze. Gli utenti possono in qualsiasi momento aggiornare il loro profilo. Ogni utente possiede una lista di skill, ovvero delle label associate ad una percentuale di esperienza.

e.g. L'utente adileone ha come skill [java programming – 7.0 , software design – 5.6] .

- Può creare delle pipeline di lavoro che può sottomettere al un sistema di esecuzione esterno.

L'admin :

- Può creare utenze user.
- Può creare altre utenze admin.
- Può vedere tutte le pipeline salvate di ogni utente sul sistema.
- Può creare altre pipeline.

Il sistema è organizzato secondo un'architettura client-server composta da backend e frontend.

Link AWS :

Backend : [KayakMockBackend.eu-west-2.elasticbeanstalk.com](https://kayakmockbackend.eu-west-2.elasticbeanstalk.com)

Accessibile da qualsiasi client semplicemente aggiungendo il suffisso /rest/{operazione}  
ad esempio : [KayakMockBackend.eu-west-2.elasticbeanstalk.com/rest/list](https://kayakmockbackend.eu-west-2.elasticbeanstalk.com/rest/list)

Frontend : [KayakClient.eu-west-2.elasticbeanstalk.com](https://kayakclient.eu-west-2.elasticbeanstalk.com)

L'uso del sistema in discussione è descritto principalmente dai seguenti casi d'uso :

Caso d'uso UC1: Creazione pipeline – Attore primario : un utente del sistema.

Scenario principale di successo: [CRUD : create]

1. Un utente vuole creare una nuova pipeline di lavoro.
2. L'utente inserisce il suo username e la sua password. Il sistema verifica la correttezza dei dati inseriti, e autentica l'utente. Il sistema reindirizza l'utente all'homepage di Kayak.
3. L'utente sceglie l'attività "Pipeline". Il sistema mostra la lista delle pipeline già create dall'utente.
4. L'utente sceglie l'opzione pipeline editor dall'apposito button.
5. L'utente sceglie i nodi (stage o input) da inserire e tramite l'apposito panel definisce gli archi della pipeline tramite il panel addLink.
6. L'utente inserisce un nome e una descrizione per la pipeline, e ne conferma la creazione.
7. Il sistema registra tutte le informazioni sulla nuova pipeline, compresa la data di creazione e un identificatore univoco. Da questo momento in poi la pipeline potrà essere eseguita o ancora modificata dall'utente che l'ha creata. Un amministratore sarà in grado di visualizzarla tra la lista delle pipeline di tutti gli utenti del sistema.

Caso d'uso UC2 : Consultazione delle proprie pipeline – Attore primario : un utente del sistema.  
[CRUD : retrieve]

1. Un'utente vuole visualizzare una propria pipeline.
2. L'utente inserisce username e password per autenticarsi tramite la pagina di login.
3. L'utente sceglie la tab Pipeline.
4. Il sistema mostra l'elenco delle pipeline dell'Utente e l'Utente ne seleziona una tramite il link modify.
5. Il sistema mostra nome, descrizione della pipeline selezionata e l'elenco degli archi inseriti che la compongono.

Caso d'uso UC3: Consultazione di tutte le pipeline sottomesse al sistema – Attore primario : un amministratore del sistema.

1. L'admin inserisce il suo username e la sua password. Il sistema verifica la correttezza dei dati inseriti, e autentica l'admin. Il sistema reindirizza l'admin all'home page di kayak.
2. L'admin sceglie l'attività in alto a destra " Kayak Management" visibile solamente agli utenti admin.
3. Il sistema mostra la lista delle pipeline salvate nel sistema con descrizione, data creazione e utente creatore.
4. L'admin sceglie una determinata pipeline e vede di che archi è composta.

Caso d'uso UC4: Registrazione nuovo utente – Attore primario : un nuovo utente del sistema.  
Un nuovo utente usa il sistema attraverso l'apposita pagina per la registrazione, per registrarsi presso il sistema stesso.

Caso d'uso UC5: Cancellazione pipeline – Attore primario : un utente del sistema.  
[CRUD : delete]

1. Un'utente vuole cancellare una propria pipeline.
2. L'utente inserisce username e password per autenticarsi tramite la pagina di login.
3. L'utente sceglie la tab Pipeline.
4. Il sistema mostra l'elenco delle pipeline salvate dall'utente.
5. L'utente sceglie la pipeline da cancellare ed effettua la delete tramite il link delete.

Caso d'uso UC6: Modifica Pipeline esistente – Attore primario : un utente del sistema.  
[CRUD : Create-Retrieve-Update-Delete]

1. Un'utente vuole modificare una propria pipeline.
2. L'utente inserisce username e password per autenticarsi tramite la pagina di login.
3. Il sistema mostra l'home page di kayak.
4. L'utente sceglie la tab Pipeline.
5. Il sistema mostra l'elenco delle pipeline salvate dall'utente.
6. L'utente sceglie la pipeline da modificare ed effettua l'operazione tramite il link modify.
7. Il sistema mostra la pipeline con la lista degli archi.
8. L'utente vuole eliminare l'ultimo arco inserito, quindi inserisce l'id dell'arco nell'apposita form 'remove edge by id' e clicca sul pulsante remove.
9. Il sistema elimina l'arco selezionato.
10. L'utente ora vuole aggiungere un arco alla pipeline, quindi esegue le stesse operazioni che esegue nell'editor. Inserisce un nodo (ad esempio ciao.csv). Inserisce uno stage (ad esempio profile Complete). Poi crea il link (ciao.csv --> profile Complete).
11. Il sistema crea e registra il link.

La prima tab e in particolare le tab LIST, INSERT, REFRESH e RESET sono demo. Ad ogni operazione è associato un endpoint REST sul backend che restituisce una risposta diversa in base al tipo di operazione selezionata ... ad esempio :

Caso d'uso UC7: Submit Pipeline esistente – Attore primario : un utente del sistema.

1. Un'utente vuole sottomettere al sistema una propria pipeline.
2. L'utente inserisce username e password per autenticarsi tramite la pagina di login.
3. Il sistema mostra l'home page di kayak.
4. L'utente sceglie la tab Pipeline.
5. Il sistema mostra l'elenco delle pipeline salvate dall'utente.
6. L'utente sceglie la pipeline da sottomettere attraverso il link submit.

A questo punto Kayak Web confeziona un JSON e lo invia tramite REST al backend (anch'esso su aws). Il backend lo riceve ed eseguirà le operazioni preposte. Una volta effettuata l'operazione il backend restituisce una risposta al frontend sempre tramite REST, che stamperà sulla home in alto un Document di MongoDB "pipeline submitted".

Tecnologie proposte :

- Gestione della persistenza : Spring Data JPA su istanza RDS aws Postgresql
- Autenticazione e autorizzazione : Spring Security
- Framework sviluppo web : SpringMVC
- Web server : Amazon Web Services