CHARTA

Descrizione

Charta è un’applicazione web per la gestione di documenti digitali.

Sarà utilizzata sia da Galileo che da aziende clienti che la richiedono come servizio di archiviazione.

Deve essere utilizzata da operatori, che caricano i documenti e li taggano, e da utenti visualizzatori, che leggono e scaricano i documenti.

Il caricamento di un documento avviene tramite queste fasi:

1. Scan
2. Tag
3. Control
4. Publish

Queste fasi sono effettuate in ordine da diversi operatori in diversi momenti.

Dopo ogni fase il documento cambia stato e resta in attesa di essere gestito da un operatore nella fase successiva

1. Fase di scan

Un operatore scannerizza i vari file che compongono un documento digitale (che viene creato in questo momento), sceglie il **nome** del documento, il **realm** di appartenenza e la **classe** del documento.

Il **realm** è il dominio al quale appartiene il documento. In genere ogni azienda cliente corrisponde a un realm.

La **classe** del documento è la tipologia, ad esempio contratto, cedolino, fattura ecc.

2. Fase di tag

Un operatore tagga il documento, compilando dei campi predisposti che variano a seconda della classe del documento.

Se ad esempio il documento è un cedolino dell'azienda X, l'interfaccia fornirà all'operatore l'insieme di campi da compilare previsti per i cedolini dell'azienda X.

I valori che l'operatore dovrà inserire all'interno di questi campi sono informazioni già contenute all'interno dei file scansionati, e questa fase serve a renderli digitali per permettere agli utenti finali di fare ricerche filtrando sulla base del valore dei tag.

Bisogna inoltre implementare un sistema di OCR o comunque un intelligenza artificiale che assista

l'operatore compilando i campi in modo automatico leggendo i file scansionati.

In questa fase l'operatore deve avere da un lato la preview del documento, con pagina scorribile, e dall'altro lato i campi dei tag dentro cui inserire i valori.

3. Fase di control

Un operatore controlla i tag di un documento e verifica se combaciano con i valori del vero documento, che l'operatore può vedere tramite una preview.

L'operatore deve poter segnalare i tag che non sono stati compilati correttamente e rimandare il documento alla fase precedente, oppure correggere personalmente i tag sbagliati e mandare il documento alla fase successiva.

4. Fase di publish

Un operatore decide di approvare la pubblicazione di un documento o di mandarla alla fase precedente.

Decide anche chi potrà visualizzare il documento una volta pubblicato.

Visualizzazione

Gli utenti possono visualizzare e scaricare i documenti all'interno di una pagina che mostra i documenti in ordine inverso di caricamento e un sistema di ricerca, in grado di filtrare i documenti per categoria (è un tag) e valore dei tag o nome del documento.

Gli utenti possono vedere solo i documenti su cui hanno accesso di lettura. Se l'utente è l'**owner** del documento, può sempre vederlo, altrimenti deve essere membro di un gruppo con permesso read o admin su quella categoria di documenti

Ruoli

Ogni operatore appartiene a dei gruppi. Ogni gruppo avrà dei ruoli su delle categorie di documenti di quel realm

Un operatore quindi potrebbe avere un ruolo in un realm e un ruolo diverso in un altro realm.

Ad esempio: un operatore può scannerizzare e taggare documenti per l'azienda X e può pubblicare documenti per l'azienda Y.

Analisi dei permessi e dei gruppi

PROBLEMA: Esistono due tipi di permessi

- Permessi di tipo operativo, che vogliamo associare a cose generali come i realm e le categorie dei documenti
- Permessi di visibilità sui singoli documenti

Soluzione 2

Un utente appartiene a più ACL e più GROUP

Definizione di ACL e GROUP:

La **ACL** fornisce ai propri membri l'accesso di lettura sui singoli documenti ad esso associati

Il **GROUP** fornisce ai propri membri un ruolo su tutti i documenti appartenenti ai realm e alle classi ad esso associate

Il **ruolo** può essere scan, tag, correct, publish, read, all (comprende tutti i precedenti ruoli)

Differenza:

- Le **ACL** vengono associate ai singoli documenti
- I **group** vengono associati ai realm e alle classi

Con i group, oltre ai ruoli di scan, tag, ecc, esiste il ruolo read, che può dare ad alcuni utenti speciali (non solo agli operatori interni) i permessi di lettura a tutti i documenti di un certo realm, o di una certa classe all'interno di un realm

Con le ACL, un operatore che carica un documento può scegliere i permessi di lettura che avrà il singolo documento una volta pubblicato, associandolo a delle liste che non hanno niente a che vedere con le suddivisioni operative o gli utenti speciali

---

Soluzione 3 (scelta)

Un utente appartiene a più GROUP

- Le **categories** vengono associate ai singoli documenti
- I **group** vengono associati ai realm
- Il **role** può essere upload, tag, correct, publish, read, admin (comprende tutti i precedenti ruoli)
- Una tabella **GroupRoleCategory** assegna ad ogni gruppo di utenti dei ruoli su una categoria di documenti (esempio: il ruolo read su tutti i cedolini)

Tutti i documenti hanno una categoria più una chiamata “default” su cui un gruppo può avere dei permessi

Al momento il ruolo admin è riferito solo alla categoria default, che comprende tutti i documenti di quel realm

Esempi di permessi

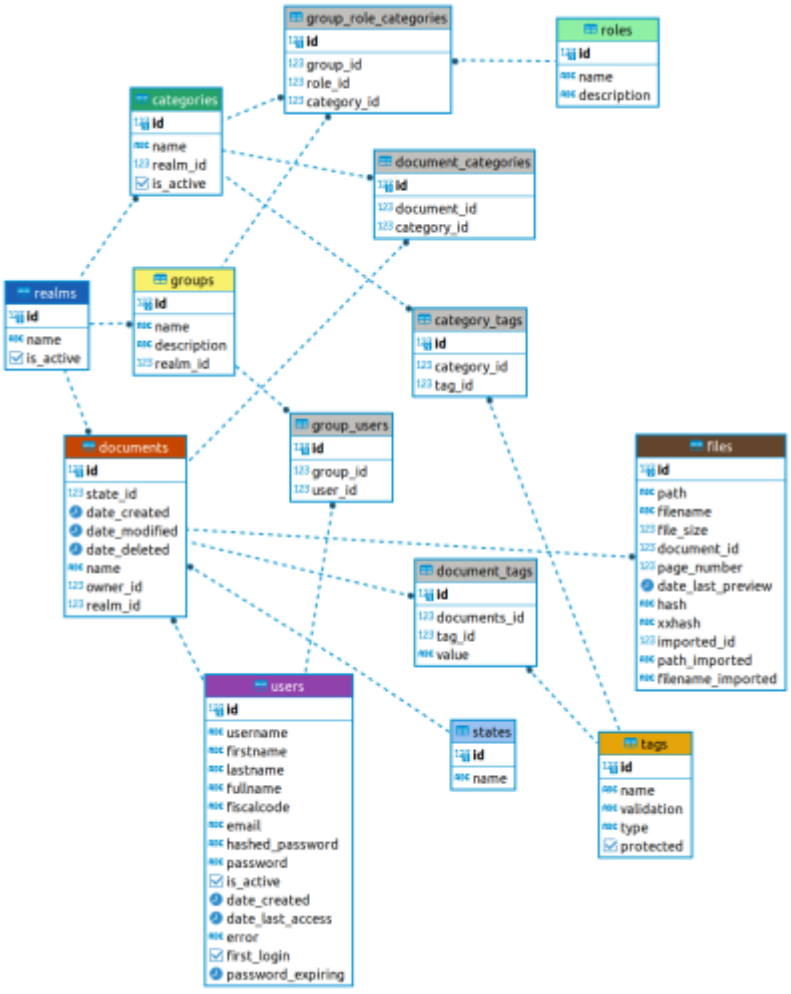
Un documento è leggibile da:

- Utente owner del documento
- Utenti membri di un supergroup con ruolo “read” o “admin” sulla categoria del documento o quella default

Un documento è scannerizzabile da:

- Utenti membri di un supergroup con ruolo “upload” (probabilmente questo ruolo sarà sempre riferito alla categoria default del realm su cui si fa l'uplaod)

Schema del db proposto



Spiegazione:

Un documento è un oggetto composto da:

- Un certo numero di **file**
- Un insieme di **tag** a cui vengono attribuiti dei valori
- Una **categoria** (tipologia, ad es. cedolino, contratto, ecc.) + la categoria default che comprende tutti i documenti di un realm

Ogni categoria di documento ha diversi tag da riempire (es. cedolino chiede il numero fiscale, contratto chiede data di scadenza), essi acquisiranno dei valori solo quando verranno associati a un documento

Ogni categoria appartiene a un realm, poiché diversi realm possono avere diverse definizioni della struttura di un cedolino o di un contratto

Gli utenti (**user**) appartengono a:

- **group**, i quali determinano il ruolo sopra ai realm e sulle categoria all'interno dei realm

I ruoli (**role**) possono essere:

- SCAN, o UPLOAD: l'utente può creare nuovi documenti, decidendone realm, categoria (e owner?)
- TAG: l'utente può aggiungere tag al documento
- CORRECT: l'utente può correggere i tag o segnalare gli errori
- PUBLISH: l'utente può rendere pubblico il documento decidendo i gruppi che ne hanno accesso
- READ: l'utente può leggere i documenti pubblicati

Utilizzo dell'applicazione

Di seguito verrano descritte le varie fasi nei dettagli concentrandosi sull'utilizzo da parte degli utenti.

NOTA IMPORTANTE: Le interfacce utente vanno disegnate tenendo conto delle **abilità degli utenti**. Per esempio, le fasi di TAG che avvengono all'interno di Galileo saranno probabilmente svolte da **operatori con disabilità**, occorre quindi semplificare le operazioni, in particolare l'input (bottoni molto grandi ecc.)

Fase di scan

Accesso

L'operatore fa **login** selezionando in alcuni casi il realm, accede alla pagina di Primo Accesso se entra per la prima volta altrimenti entra nella sezione “Home”

Scannerizzazione e upload

All'operatore si presenterà una schermata interattiva in cui può decidere se:

- **trascinare** un file all'interno di un'area
- premere il tasto “**Scansiona**” per far partire uno **scanner** connesso alla macchina client

La scansione deve bypassare il sistema operativo ed essere più intuitiva possibile e **un unico click** dovrebbe scansionare e fare l'upload del file.

Potrebbero essere aggiunte anche alcune **impostazioni** di scannerizzazione da selezionare che vengono **salvate**.

Possibile implementazione: un documento “draft” viene creato in questo momento, e i file caricati si legano al documento draft. Se l'utente abbandona l'applicazione e ritorna, deve essere caricata l'ultima draft creata, da cui l'operatore può riprendere

---

Gestione delle pagine

Dopo ogni scansione la pagina scansionata appare come piccola **preview** in un'area che contiene tutte le pagine scansionate. Queste pagine devono poter essere **cancellate**, **riordinate** e **rinominate** singolarmente. Ogni pagina deve poter essere aperta individualmente con una preview più grande per essere ispezionata.

Attenzione

In genere ogni file scansionato rappresenta una pagina di un documento. L'operatore però può trascinare i file anche manualmente, ma in questo caso potrebbe trascinare un file composto da più pagine, che va gestito in modo diverso da come gestiamo un file scansionato di una singola pagina. Si potrebbe ad esempio **dividere** (lato server) il file composto da più pagine in più file di una singola pagina.

Creazione del documento

Vicino all'area con la preview delle pagine ci devono essere **tre campi** compilabili dall'operatore:

- **Nome** del documento: è un nome generico che è slegato dal nome delle singole pagine. Si può decidere se far si che sia univoco o no (il documento viene comunque identificato dall'id)
- **Realm**: selezionabile da una select list che mostra tutti i realm su cui l'operatore ha ruolo di scan (decidere se mettere il campo qui o prima della fase di scan)
- **Class** del documento: rappresenta la tipologia del documento tra quelle previste all'interno del realm. È importante che siano selezionabili solo le class su cui l'operatore ha ruolo di scan. (questi controlli vanno implementati sia lato client che lato server)

Dato che ogni documento avrà un owner e delle ACL, va deciso in quale fase queste informazioni vanno inserite.

L'owner ad esempio potrebbe essere un campo riempibile nella fase di scan, ma facoltativo, mentre le ACL potrebbero essere decise nella fase di publish.

Fase di tag

Accesso

L'operatore fa il login ed entra nella sezione “Tag” dal menù laterale.

Selezione del documento

Gli si presenta una pagina con una **lista di realm** (tutti i realm su cui l'operatore ha ruolo di tag).

Ogni realm è affiancato da una freccetta che serve per espandere la sezione.

Cliccandoci verranno mostrate sotto al realm **tutte le class** su cui l'operatore ha ruolo di tag all'interno del realm.

Ogni realm e ogni class sono affiancati da un numero che indica la **quantità** di documenti ancora da taggare presenti in coda all'interno del realm/class.

Cliccando il realm o la class verrà visualizzata la lista di **tutti i documenti** all'interno di quel realm/class, con indicato il nome, la data di creazione e altre informazioni.

Cliccando su un documento verrà aperta la schermata di tag.

Da ogni schermata deve essere possibile **tornare indietro** alla schermata precedente

Schermata di tag

Cliccando su un documento si aprirà una schermata **divisa in due parti**.

Sulla sinistra verrà visualizzata la **preview** del documento, in cui è possibile scorrere, sfogliare le pagine e fare zoom.

Sulla destra ci saranno i **tag**, ovvero dei **campi di input** che devono essere riempiti dall'operatore.

I valori da inserire all'interno dei tag sono semplicemente dei dati contenuti nello stesso documento e visualizzabili nella preview.

Il compito dell'operatore sarà quindi di leggere il documento, trovare i dati e inserirli nei campi.

Ogni class di documenti prevede uno specifico insieme di tag che devono essere riempiti, quindi l'operatore dovrà vedere **solamente** i tag appartenenti alla class del documento che sta taggando.

Validazione

Ogni tag deve avere delle regole per **validare** i valori inseriti.

Ci sono due tipi di regole di validazione:

- Regole sul **singolo campo**: il valore del tag deve rispettare un certo formato (si pensi alle targhe, alle mail o ai codici fiscali). Queste regole possono essere implementate come stringhe **regex** salvate nel database, in una colonna apposita nella tabella dei tag.
- Regole che si riferiscono ad **altri campi**: il valore del tag deve essere coerente con il valore dato ad **altri tag**. Ad esempio, la data di scadenza deve essere "maggiore" della data di creazione. Occorre pensare a un'implementazione più generica possibile, magari anche con un linguaggio interno e un **parser** che gestisce queste regole.

Automazione

Per semplificare il lavoro dell'operatore, si dovrà creare un'**intelligenza artificiale** o un sistema di **OCR** che legga il documento e inserisca i valori all'interno dei campi. Questa elaborazione può essere fatta da un processo che gira **in parallelo**, anche a partire dalla fase di scan.

Quando l'operatore aprirà il documento in fase di tag per la prima volta, visualizzerà i campi con dei valori estratti da questo software, che l'operatore potrà modificare come vuole.

Si potrebbe anche implementare un sistema di **suggerimenti** dei valori dei campi:

Quando l'operatore clicca su un campo, compaiono diversi valori suggeriti che l'operatore può **selezionare**.

Questi valori potrebbero essere calcolati sulla base dei valori di altri campi o sulla base di valori inseriti in passato.

Esempio: Inserendo nome e cognome il sistema mi suggerisce un codice fiscale che è stato già associato a quel nome e cognome in passato.

Completamento

Una volta finito di inserire i tag, l'operatore clicca su un **pulsante** per chiudere la fase di tag su quel documento e farlo avanzare allo stato di check. La schermata si chiuderà e ritornerà sulla selezione del documento.

Visualizzazione

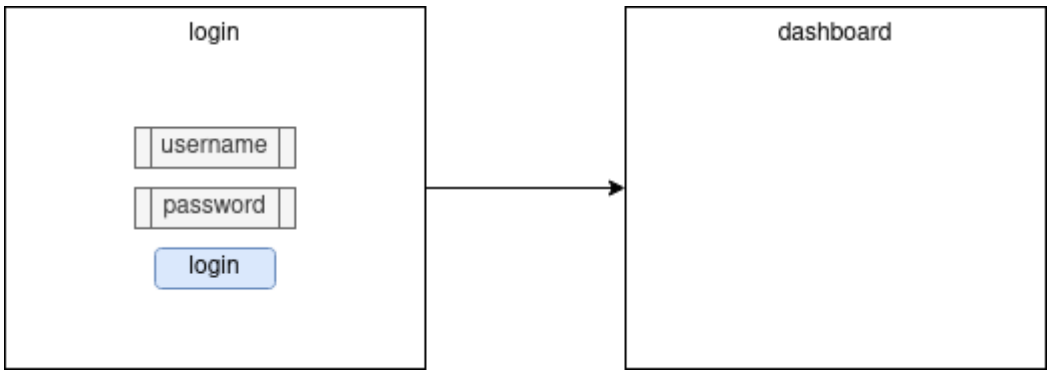
Ogni documento deve poter essere visibile dal proprio owner e dagli utenti appartenenti ai groups con READ (o ADMIN) role sul realm e sulla categoria a cui appartiene il documento

Quando un utente va nella schermata di visualizzazione, vedrà i documenti inseriti di recente nel sistema e potrà effettuare ricerche per trovare i documenti.

I documenti devono avere una thumbnail, un nome, un'indicazione del size

Da questa pagina si potrà selezionare un documento ed entrare nella relativa pagina che permetterà di visualizzare le preview dei file di quel documento, i suoi tag e di poterli scaricare

utente normale che consulta



utente operatore che inserisce

Casi particolari (e non) di utenti

- operatori (personale Galileo che scannerizza/elabora i documenti)
  - su più realm
  - più operazioni
- dipendenti serit (vedono solo i loro documenti)
- sindacalista serit (possono uploadare solo un tipo di documento)
- dipendenti serit (vedono solo una classe di documento)
- dirigenti serit (vede tutto nel real serit e creano nuovi utenti del realm)
- dirigenti galileo (vedono tutto di tutti i real e honno abilitate tutte le operazioni)

From:
https://wiki.csgalileo.org/ - **Galileo Labs**

Permanent link:
https://wiki.csgalileo.org/projects/charta/analisi

Last update: **2022/10/28 10:43**