"Software Engineering" Course a.a. 2019-2020

Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)

Progetto3: Gestionale per lo sharing online di appunti e materiale di studio

Date	5/12/2019
Deliverable	D1
Team (Name)	LAG

Team Members		
Name & Surname	**Matriculation Number**	**E-mail address**
Leonardo Serilli	*252256*	*leonardo.serilli@studentunivaq.it*
Andrea Formichetti	*255086*	*andrea.formichetti@studentunivaq.it*
Gabriele Colapelle	*252262*	*gabriele.colapelle@studentunivaq.it*

Table of content of this deliverable

List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task	Date the chall	Explanation on how the challenge has been managed

	is identified	enge is resolved	
Sistema di guadagno Token	25/11	2/12	• I token si guadagnano dall'acquisto e non dagli upvotes per evitare il lucro tramite conoscenti • Servono più caricamenti per guadagare token da questi ultimi, altrimenti potrebbero essere caricati file inutili e poi rimossi (prima che vengano segnalati) sempre al fine di lucro
Riconoscimento dell'appartenenza di un nuovo utente ad un universitò • Il sistema deve in automatico riconoscere sia se lo studente appartiene a un'Università che l'appartenenza ad un corso di laurea dell'università indicata, tramite la sua mail (vedi NFR: interface) • Il tempo di risposta al momento della registrazione deve essere estremamente basso	25/11	13/01/20	• Abbiamo risolto la challenge aggiungendo nel database il dominio delle università in modo tale che al momento della registrazione, se il dominio specificato nella mail inserita coincide con uno di quelli presenti nella base di dati, all'ora verrà inviata automaticamente una mail di conferma della registrazione. In questo modo è verificata l'appartenenza ad un università; il corso di laurea verra specificato dall'utente. • Quindi occorrerà una semplice e veloce verifica sule università nel database che non saranno cosi elevate da allungare pesantemente I tempi di risposta • Purtroppo ancora non è possibile ottenere automaticamente il corso di laurea di tale utente tramite la conoscenza della sua universita • Inoltre utilizzeremo una mail di conferma per l'autenticita di tale mail
Velocità di risposta della ricerca di appunti	2/12		• L'organizzazione dei dati in pagine e appunti ci permette una ricerca su due livelli, cioè: la ricerca in base ai campi della pagina (utente e materia) e un filtraggio avanzato su quelli degli appunti che permetta di velocizzare le query sul database. In più le varie scelte di ordinamento danno all'utente

			la possibilità di trovare gli appunti più appropriati relativamente a ciò che cerca
Conversione di immagini multiple in un singolo file pdf: ● Con alcune ricerche siamo venuti a conoscenza di alcune funzioni di php che permeterebbero di convertire un array di immagini in un singolo pdf, ma non avendole ancora testate non si può ancora dire risolta questa task	15/12		

A. Stato dell'Arte

DOCSITY

Docsity è una piattaforma per la condivisione di documenti scolastici.
Anche qui, come nel nostro caso è stato integrato un sistema di gamification: ogni utente può comprare i documenti tramite gettoni che si guadagnano all'interno del sito.
E' possibile guadagnare gettoni(token) tramite caricamento di documenti o tramite download dei propri documenti da parte di altri utenti (stesso sistema di guadagno implementato nella nostra piattaforma).
Inoltre in docsity è permesso di guadagnare punti attraverso la recensione della propria università , attraverso la risposta alle domande poste da altri utenti; e se la propria risposta è stata votata come migliore riceverete dei punti bonus.
PIU PRECISAMENTE IL GUADAGNO DI TOKEN AVVIENE ATTRAVERSO LE SUEGENTI AZIONI:

Completa il tuo profilo	20 punti	Una sola volta
Condivisione appunti	10 punti	Ogni documento caricato
Download ricevuti	Fino a 990 punti	In base a quante volte il tuo documento viene scaricato
Recensisci la tua università	20 punti	Una sola vota
Rispondi alle domande	20 punti	Ogni risposta max 3 al giorno
Risposta votata come migliore	40 punti	

Il sistema di ricerca fornisce una prima ricerca per materia, che può essere filtrata in un secondo momento (molto simile al sistema di ricerca appunti implementato nel nostro sistema)
Altre funzionalità aggiuntive di docsity:

- Sistema di tutoraggio interno per gli utenti in difficoltà (a pagamento)
- Possibilità di fare domande su determinati argomenti
- Possibilità di comprare appunti in € se non si hanno punti download disponibili.
- Possibilità di visualizzare una graduatoria delle università in base a come sono state recensite dagli utenti
- Blog interno al sistema dove vengono esposti consigli per lo studio,

notizie, proposte di lavoro ed altre piccole funzionalità

- Possibilità di esercitarsi su una determinata materia attraverso delle domande, si può visualizzare la risposta sole una volta effettuato l'accesso nel sistema
- Sezione dedicata alle proposte di lavoro che vengono pubblicate

STUDOCU

Nella home page viene subito offerta una possibilità di ricerca di testi e appunti (tramite materia); solo dopo aver effettuato la ricerca preliminare c'è la possibilità di effettuare un filtraggio per università, corso, categoria e lingua. Questo sistema di ricerca ci è stato utile per definire una gerarchia di ricerca per rendere le operazioni sui dati più veloci. Inoltre abbiamo aggiunti la ricerca in base alle segnalazioni per gli scopi del moderatore. Una volta effettuala la ricerca possiamo visualizzare un'anteprima dell'appunto oppure visualizzarlo completo, il download di un appunto è possibile solo se l'utente ha effettuato il login.

Anche qui è presente un sistema di gamification implementato nel seguente modo:

Caricamento di un documento	50 punti
Ogni commento ricevuto	50 punti
Ogni unpvote ricevuto su un documento	25 punti
Ogni volta che il proprio documento viene visualizzato	1 punto

I punti ottenuti sono utili solo al fine di ricevere ticket della lotteria del costo di 50 punti.
E' possibile visualizzare gli appunti che ci interessano solo dopo aver caricato un nostro appunto, in questo modo possiamo visualizzare tutti i testi che vogliamo per i prossimi 14 giorni, dopodichè sarà necessario caricare un altro appunto. Se non vogliamo pubblicare i nostri documenti, è possibile acquistare un account premium dal costo di 3€.

Altre funzionalità aggiuntive di studocu:

- L'utente che ha effettuato il login può accedere alla dashboard dove visualizza gli appunti caricati ed eventualmente modificarli, inoltre può visualizzare i punti ottenuti grazie al caricamento, commento, upvote di un appunto oppure ogni volta che un proprio appunto viene visualizzato da un altro utente.
- E' possibile spendere i propri punti per comprare ticket di una lotteria che effettua estrazioni mensilmente, questa ci è sembrata un idea carina anche di facile implementazione.

- Possibilità di vedere le statistiche di ogni università all'interno del sistema es. totale documenti caricati e corsi ma che ci è sembrata carina come idea ma abbastanza inutile nella pratica.
- Blog dove sono presenti una vasta gamma di categorie, le più importanti sono job, lifestyle e student life.

APPUNTICONDIVISI

Molto simile ai due sistemi citati in precedenza, anche qui si incentiva il caricamento di appunti per guadagnare punti che potranno essere spesi per scaricare altri documenti presenti all'interno del sistema. Ogni volta che viene caricato un documento l'utente riceve un punto, ogni documento da scaricare costa un punto(prendo 1,pago1), in questo modo è stato creata una gestione della gamification semplice ed efficiente.
Il feedback su un appunto è garantito da un sistema di rating, ogni utente può dare un "like" ad un appunto se è stato di suo gradimento.

PUNTI DI INTERSEZIONE

- Questi sistemi ci hanno dato una grande mano sull'implementazione della gamification, molto importante per invogliare gli utenti ad usufruire i servizi offerti.
- Struttura della ricerca degli appunti, operazione fondamentale per offrire all'utente un sistema facile ed efficace, sono stati anche presi spunti per l'implementazione del database in modo da poter soddisfare i requisiti non funzionali.
- Sistema di feedback interno al sistema, ogni utente può recensire positivamente un appunto se è stato considerato utile, se vengono pubblicati appunti inappropriati si ricorre ad una segnalazione.

ALTRI POSSIBILI SERVIZI CHE POTRANNO FACILMENTE ESSERE INSERITI NEL SISTEMA

- Visualizzazione graduatoria delle università in base a come vengono recensite
- Lotteria mensile con ticket acquistabili attraverso token
- Blog interno all'ateneo

B. Raffinamento dei Requisiti

A.1 Servizi (con prioritizzazione)

- *Importanza e Complessità: A, M, B (alta, media, bassa)*

- Il sistema prevede 4 attori e I requisiti funzionali sono organizzati associandoli all'attore che riguardano

Requisiti Funzionali (organizzati per attore)

- Un **utente non registrato**:
  - Registrarsi nel sistema se è iscritto ad un università indicando mail, universita e e il corso di laurea ottenendo dei token di partenza (vedi NFR: Interface) **AA**
  - Può visualizzare solo anteprime degli appunti (ad esempio solo un frammento di un pdf o una parte dei video) **AB**
  - Ricercare tramite parola chiave: materia o utente **AB**
  - Effettuare un filtraggio avanzato, dopo la ricerca, sulle caratteristiche degli appunti tramite un form in cui può indicare corso di laurea, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti), argomento e tipo di file(video, testo, audio) **AM**
  - visualizzare la graduatoria degli studenti che hanno totalizzato più token (nell'anno, nel mese, nella settimana o in totale):Best Collaborative Students **MB**

- Un **utente registrato** può:
  - Visualizzare la graduatoria degli studenti che hanno totalizzato più token (nel mese)::Best Collaborative Students **MB**
  - Ricercare tramite parola chiave: materia o utente **AB**
  - Effettuare un filtraggio avanzato, dopo la ricerca, sulle caratteristiche degli appunti tramite un form in cui puoò indicare corso di laurea, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti), argomento, e tipo di file(video, testo, audio) **AM**
  - Ordinare I risultati per Data e upvotes (crescente o decrescente) **BB**
  - Effettuare log in/out **AM**
  - Creare una pagina inserendo il suo nome e la materia che tratta**AB**
  - Inserire appunti in una pagina tramite caricamento o link (pdf, immagini, audio, video) **AA**
    - Le immagini cosi caricate possono essere convertite in un pdf, inoltre il sistema offre la possibilità di mergare in un unico pdf una collezione di immagini prima di caricarlo, dando più valore all'appunto **AA**
  - Valutare l'utilità del materiale altrui (tramite un upvote non modificabile); **BB**
  - Ottenere token dal caricamento di appunti (dopo un numero ancora non definito di caricamenti) **AB**

- Ottenere token in base ai suoi appunti comprati dagli altri con il vincolo: il numero di token guadagnati è sempre inferiore al prezzo di vendita dell'appunto **AB**
- Aggiornare I propri appunti sostituendo un file con la versione aggiornata
- **MB**
- Rimuovere I propri appunti: la rimozione di un appunto entro un certo lasso di tempo dal suo caricamento comporta la perdita dei token guadagnati grazie ad esso**AB**
- Rimuovere una propria pagina con le stesse ripercussioni date dalla rimozione degli appunti **MB**
- Modificare il nome di una propria pagina **BB**
- Modificare il proprio profilo **MB**
- Navigare tra I propri appunti allo stesso modo di una ricerca avanzata **MB**
- Segnalare appunti inappropriati segnalanadone il motivo **BB**
- Visualizzare le anteprime degli appunti (qualche pagina per I pdf e qualche secondo per audio e video)**AB**
- Scaricare appunti in cambio di token **AB**

- Un **moderatore** può:
  - Effettuare il log-in con le credenziali fornite dall'amministratore **AB**
  - Visualizzare la graduatoria degli studenti che hanno totalizzato più token (nell'anno, nel mese, nella settimana o in totale)::Best Collaborative Students **MB**
  - Ricerca tramite materia o utente o semplicemente senza campi per ottenere solo I risultati più segnalati **(I risultati verrano di base ordinati in base al numero di segnalazioni)AB**
  - Effettuare un filtraggio avanzato, dopo la ricerca, sulle caratteristiche degli appunti tramite un form in cui può indicare corso di laurea, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti) e tipo di file(video, testo, audio) **(inoltre il moderatore può filtrare I risultati anche per segnalazioni, ovvero fare in modo che gli vengano mostrati solo gli appunti segnalati)AM**
  - Ordinare I risultati per Data, upvotes (crescente o decrescente) e segnalazioni **BB**
  - Rimuovere appunti segnalati o l'intera pagina pagina: la rimozione di una pagina o di un appunto entro un certo lasso di tempo dalla loro creazione comporta la rimozione dei token guadagnati dagli appunti **AB**
  - Bannare un utente per un periodo che può essere illimitato, aggiungendo la sua mail a una black list che non gli permetterà più di accedere al sistema: gli appunti caricati da un utente che viene bannato per un periodo illimitato sono rimossi;

Al momento del Login il sistema controllora se l'utente è nella blacklist e nel caso effettuerà un controllo sulle informzaioni relative all'utene per rispondere con un promp il tempo mancante alla fine del ban **AM**
  - Scaricare appunti gratuitamente **AB**

- Un **amministratore** può:
  - Fare tutto ciò che può fare un moderatore
  - Aaccedere al backend del sistema da cui può:
    - Creare o rimuovere un account moderatore **AB**
    - Promuovere un moderatore ad amministratore
    - Vedere i dati analitici del sistema (tramite api di google o simili, perciò a costo quasi 0 di complessità) **BB**

A.2 Requisiti non Funzionali

- **Tutelability:** non essendo possibile prevedere e impedire ciò che un utente può fare con il materiale ottenuto dal nostro sistema, abbiamo dovuto escludere l'NFR riguardante le politiche di gestione della proprietà intellettuale, in alternativa inseriremo nel sistema un disclamer per avvertire l'utente a cosa va incontro diffondendo, senza citare il creatore dei contenuti, questi ultimi.
- **Tutelability**: il sistema prevede la registrazione solo da studenti universitari, la loro appartenenza ad un università andrà controllata

- **Performance**: Il software deve poter gestire almeno 200.000 studenti che dovranno potersi scambiare almeno 5.000.000 di appunti e documenti.

- **Usability(** intuitivness) Il software deve essere di facile utilizzo (anche con grandi moli di contenuti), la ricerca e il filtraggio devono produrre risultati in tempo lineare con il crescere dei campi inseriti, questo presuppone delle query ben formate e un organizzazione dei contenuti ben definita

- **Adaptabilty**: Il software dovrà essere modulare e permettere l'integrazione con altri sistemi

- **Interface**: il sistema deve interagire con altri sistemi per diminuire il data-entry (ad esempio trovare il corso di laurea di un utente che si sta registrando) ma per il momento questà funzionalità non è implementabile, quindi tutto viene fornito dall'utente **(vedi assunzioni importanti)**

- **Reliabilty**: il sistema dovra assicurare un backup delle informazioni presenti eseguendo un Dump del database almeno una volta al giorno in modo che un utente nel caso peggiore perda solamente gli ultimi progressi

- **Gamification**: un utente non deve essere in grado di ottenere un quantità spropositata di credito virtuale tramite scappatoie del sistema (ad esempio tramite ripetuti caricamenti di file inutili), deve invece avere un credito proporzionale alla sua reale attività nel sistema

A.3 Scenari d'uso dettagliati

name: primo appunto online
attori: utente non registrato: mario
flow of events:

1. Mario viene a conoscenza del sistema di condivisione di appunti usato da un suo compagnio di corso e decide di dare un occhiata
2. Effettua una ricerca e filtra I risultati sul sito indicando la categoria "riassunto" e corso di laurea "lettere", poi ordina la ricerca per "data" e naviga tra i contenuti, si accorge di poter solo visualizzare frammenti dei contenuti senza avere modo di scaricarli, perciò decide di iscriversi
3. Inserisce la propria mail universitaria e una password, attende che il sistema rilevi il suo corso di laurea
    a. **NOTA**: per il momento inserisce anche il suo corso di laurea (vedi NFR inteface)

4. A operazione avvenuta con successo gli mostra una pagina di registrazione confermata e viene reindirazzato alla pagina di spiegazioni sul funzionamento dei token e del download, ottenendo il suo primo token per la registrazione
5. Crea una pagina con il nome della materia e altre informazioni e vi carica cosi gli appunti della sua ultima lezione in formato pdf, inoltre aggiunge anche un link a un video che la riguarda.
6. Si accorge, dopo il caricamento, di essere a una piccola percentuale dei caricamenti necessari a ottenere il suo prossimo token
7.

name: primi token passivi
attori: mario
flow of events:

1. Mario dopo due giorni torna nel sistema rieffettuando il login
2. Nota che il suo credito token è aumentato per i download dei suoi appunti ricevuti dagli altri utenti
3. Va nella pagina degli appunti a cui era interessato e li scarica spendendo i suoi token
4. Ritenendo gli appunti utili torna nella pagina da dove li ha scaricati e mette un upvote

name: creazione profilo moderatore
attori: Luigi(Amministratore), Gianni(moderatore)
flow of events:

1. Luigi è incaricato di creare un account modratore
2. Va nella sezione crea Moderatore, inserisce la mail di Gianni,una password e un nome Utente generale “moderatore”, salva il profilo.
3. Contatta Gianni fornendogli le credenziali

name: ricerca e appunti inappropriati
attori: Gianni (moderatore)
flow of events:

1. Gianni esegue un ricerca degli appunti filtrando I risultati in base al numero di segnalazioni degli utenti
2. Apre il primo risultato e scarica il contenuto
3. Verifica che il linguaggio usato dall’utente Luca è inappropriato ed elimina il risultato
4. Notando che ha ricevuto molte segnalazioni decide di bannarlo momentaneamente
5. Aggiunge l"’account di Luca a una black list che gli impedira di accedere al sistema

name: caricamento immagini multiple
attori: Luca(utente registrato)
flow of events:

1. Luca soddisfato degli ottimi appunti presi alla lezione di anatomia decide di renderli disponibile nel sistema di gestione appunti della sua università
2. Effettua il log in, crea una pagina e clicca sul bottone 'carica nuovo appunto', seleziona il tipo di appunto 'immagini into pdf'
3. Scatta diverse foto con il proprio telefono al suo quaderno
4. Tramite il sistema aggiunge le foto nella gui di caricamento appunti e preme invio
5. Attente qualche secondo che il sistema raccolga le sue foto in un unico pdf, e gli notifichi il caricamento avvenuto con successo

A.4 Excluded Requirements

- **Traceabilty:** abbiamo escluso questo NFR perchè le informazioni di traceability non sono necessarie al suo funzionamento, ad esempio ci interessa solo tener conto del numero di segnalazioni e non chi le ha fatte e quando

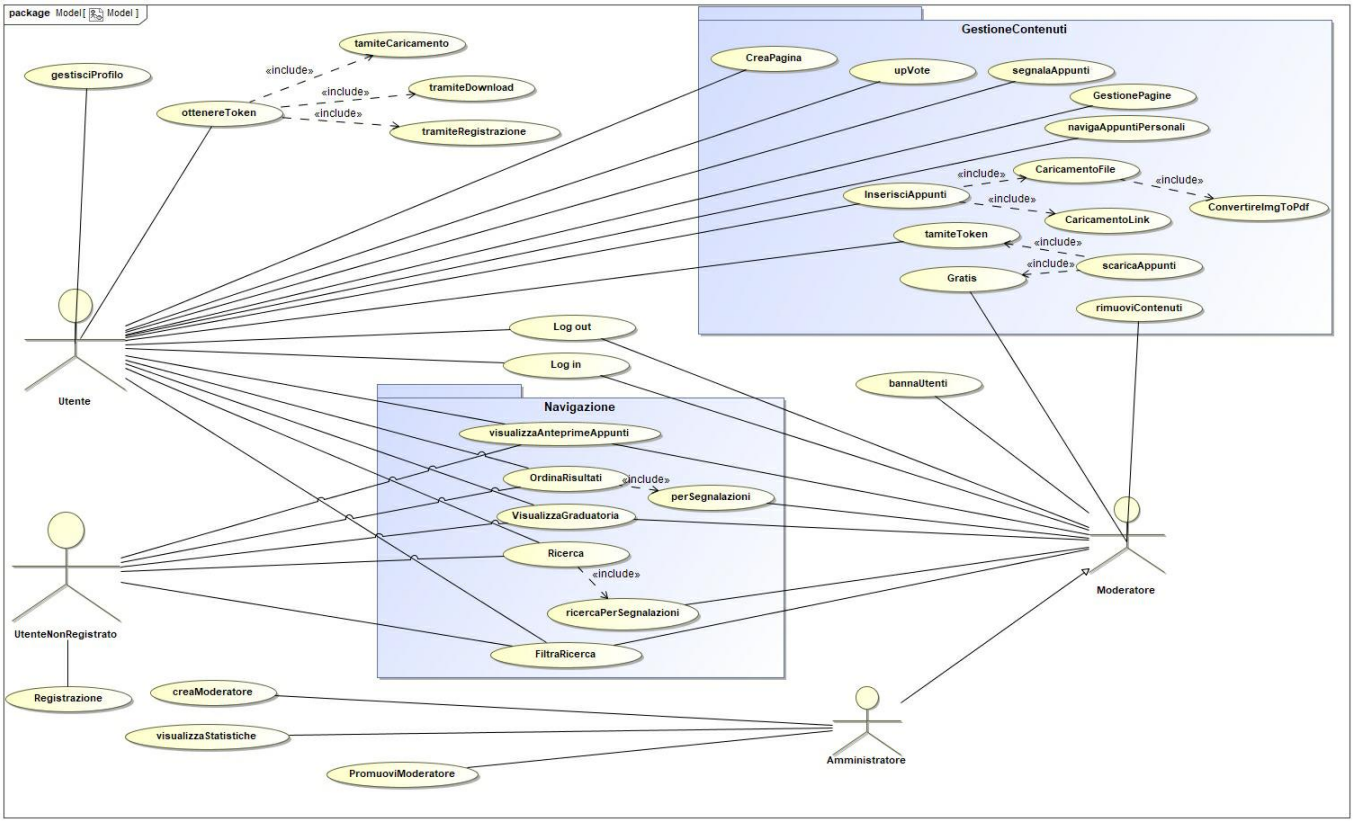
- **Availability:** gli utenti non hanno particolari restrizioni di permessi per la visualizzazione degli appunti, ogni utente può usufruire di ogni appunto indipendentemente da attributi come 'corso di laurea'

- **Tutelability:** Il software potrà prevedere diverse modalità di gestione della "proprietà intellettuale di quanto caricato", realizzeremo in parte questo requisito tramite un disclamer per avvisare gli utenti dei rischi che corrono utilizzando il materiale presente sul sito senza autorizzazione

A.5 Assunzioni

- **Assumiamo che ogni università possieda un dominio di posta elettronica rivolto agli studenti, altrimenti dovremmo creare una specifica funzionalià per il check dell'appartenenza di alcune altre università che andremmo adinserire**

A.6 Use Case Diagram



Cockburn Tables

USE CASE #	Ricerca	
Goal in Context	Orientarsi tra gli appunti del sistema	
Scope & Level	Primary Task: cercare tra gli appunti presenti nel sistema Subtask: filtrare I risultati	
Preconditions		
Success End Condition	Stampa a schermo una lista di risultati pertinenti alla ricerca	
Failed End Condition	Nessun risultato	
Primary, Secondary Actors	Attore1 : utente (anche non registrato) Attore 2: moderatore	
Trigger	Un utente cerca qualcolsa nel sistema	
DESCRIPTION	Step	Action
	1	L'utente scrive una frase indicando la materia o l'utente ricercata/o
	2	cerca
	3	Visualizza I risultati prodotti
	4	Ordina e naviga tra I risultati
EXTENSIONS	Step	Branching Action
	1	

	1	L'utente che ha effettuato la ricerca è un moderatore: I risultati sono automaticamente ordinati per segnalazioni
	2	La ricerca ha prodotto un numero eccessivo di risultati, l'utente compila a form per il filtraggio avanzato riducendo le entry in base alle caratteristiche degli appunti che ricerca

RELATED INFORMATION	ricerca
Priority:	Alta, impatta notevolmente sull'usability
Performance	I tempi di risposta devono essere lineari e crescere il meno possibile con il crescere il meno possibile con il crescere dei campi del filtraggio avanzato
Frequency	Si ci aspetta una notevole quantità di ricerche soprattutto durante il giorno
Channels to actors	<e.g. interactive, static files, database, timeouts>
OPEN ISSUES	Potrebbe richiedere troppo tempo se il traffico è intenso
Due Date	Fine gennaio
...any other management information...	<...as needed>
Superordinates	Ricerca

USE CASE #	Registrazione	
Goal in Context	Creare un account Utente	
Scope & Level	SubFunction: verrifica dell'appartenza a un corso di laurea dell'università indicata (per ora non disponiamo di un modo per verificare tale informazione perciò verra fornità dall'utente)	
Preconditions	L'utente è iscritto a un corso di laurea	
Success End Condition	L'utente sarà registrato nel sistema	
Failed End Condition	1 Account relativo alla mail gia esistente 2 la mail non corrisponde a nessuno studente dell'università indicata	
Primary, Secondary Actors	Utente non Registrato	
Trigger	L'utente avvia la Registrazione	
DESCRIPTION	Step	Action
	1	L'utente fornisce: NomeUtente, email Universitaria (e per il momento corso di laurea)
	2	Check appartenenza ad un università ramie mail di conferma se il dominio indicato è presente nella base di dati
	3	Il sistema controlla il corso di laurea a cui appartiene lo studente
		la registrazione è avvenuta con successo e vengono forniti alcuni token iniziali al nuovo utente

	2	la mail non è stata verificata: lo studente non frequenta alcun corso di laurea nell'università indicata, e viene notificato
SUB-VARIATIONS		Branching Action

RELATED INFORMATION	<Use case name>
Priority:	E critico controllare che un utente sia anche uno studente per rivolgere il sistema a questi ultimi
Performance	Il controllo del corso di laurea deve avvenire in modo rapido, per questo lo studente fornisce la propria università, altrimenti si incontra il problema che ogni università ha il proprio sistema di gestione profili, e effettuare un controllo per ognuno graverebbe sull'efficenza del sistema
Frequency	Si ci aspetta una grande affluenza al rilascio del sistema che diminuirà sempre di più
Channels to actors	<e.g. interactive, static files, database, timeouts>
OPEN ISSUES	Conrollare l'appartenenza a un corso di laurea
Due Date	Fine gennaio
...any other management information...	<...as needed>
Superordinates	<optional, name of use case(s) that includes this one>
Subordinates	<optional, depending on tools, links to sub.use cases>

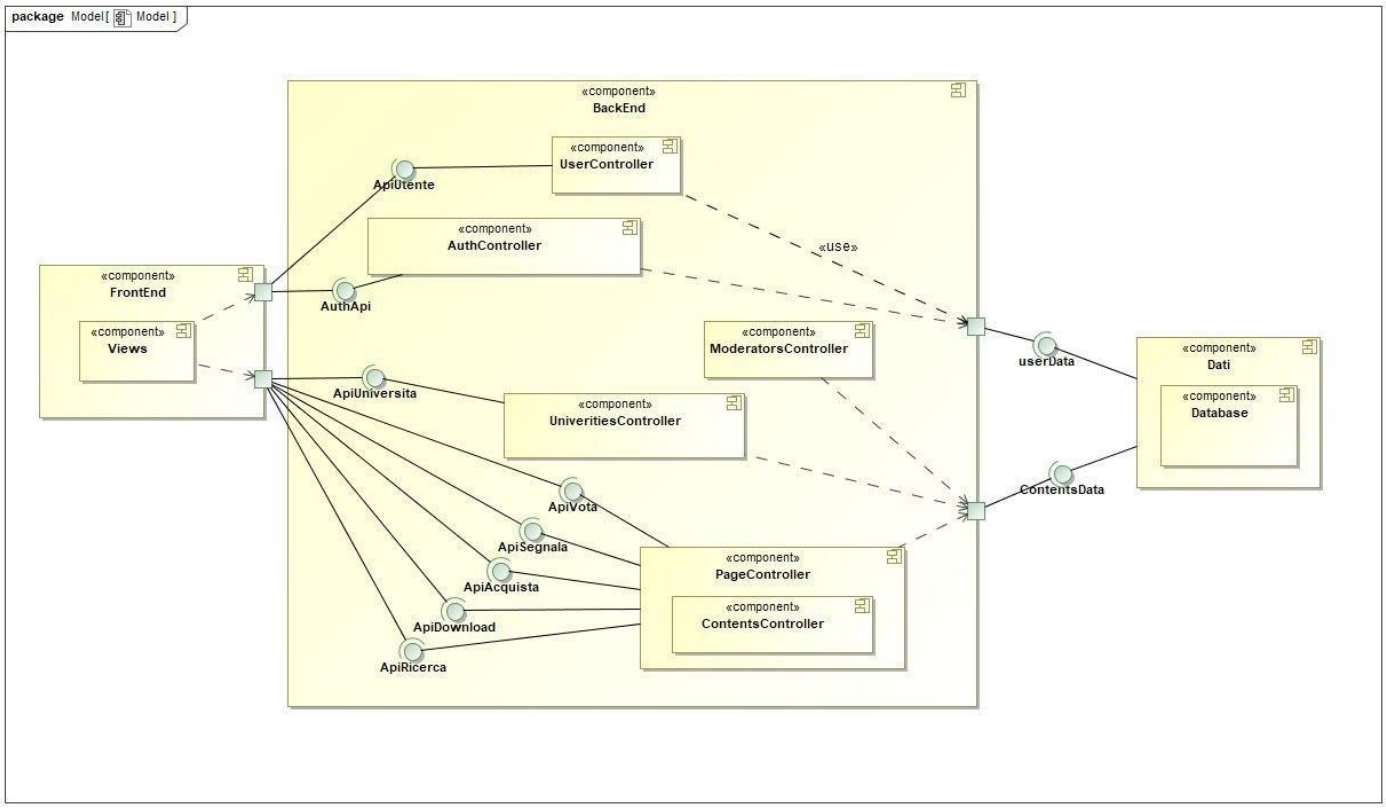
USE CASE #	Carica Appunti
Goal in Context	Caricare un appunto (file o link) all'interno di una pagina
Scope & Level	Primary Task: caricare appunti in una pagina
Preconditions	La pagina in cui caricare gli appunti è stata creata con successo
Success End Condition	Appunto caricato correttamente in una pagina
Failed End Condition	Caricamento fallito
Primary, Secondary Actors	Attore1: utente
Trigger	Un utente carica un appunto

DESCRIPTION	Step	Action
	1	L’utente seleziona il file da caricare
	2	Clicca su ‘carica’
	3	Verifica che il file è stato caricato
	4	Ottiene Token per il caricamento del file
EXTENSIONS	Step	Branching Action
	1	L'appunto non è stato caricato correttamente: rimanda alla pagina
	2	
SUB-VARIATIONS		Branching Action

RELATED INFORMATION	Carica Appunti
Priority:	Tra gli UC fondamentali, è la modalità tramite la quale il sistema ricava le informazioni alla base del suo funzionamento
Performance	Deve funzianare 24 ore su 24 e deve garantire la presenza di 5 milioni di documenti
Frequency	
Channels to actors	<e.g. interactive, static files, database, timeouts>
OPEN ISSUES	Potrebbe richiedere troppo tempo se il traffico è intenso
Due Date	Fine gennaio
...any other management information...	<...as needed>
Superordinates	Ricerca Ricerca Avanzata Ordina Risultati
Subordinates	<optional, depending on tools, links to sub.use cases>

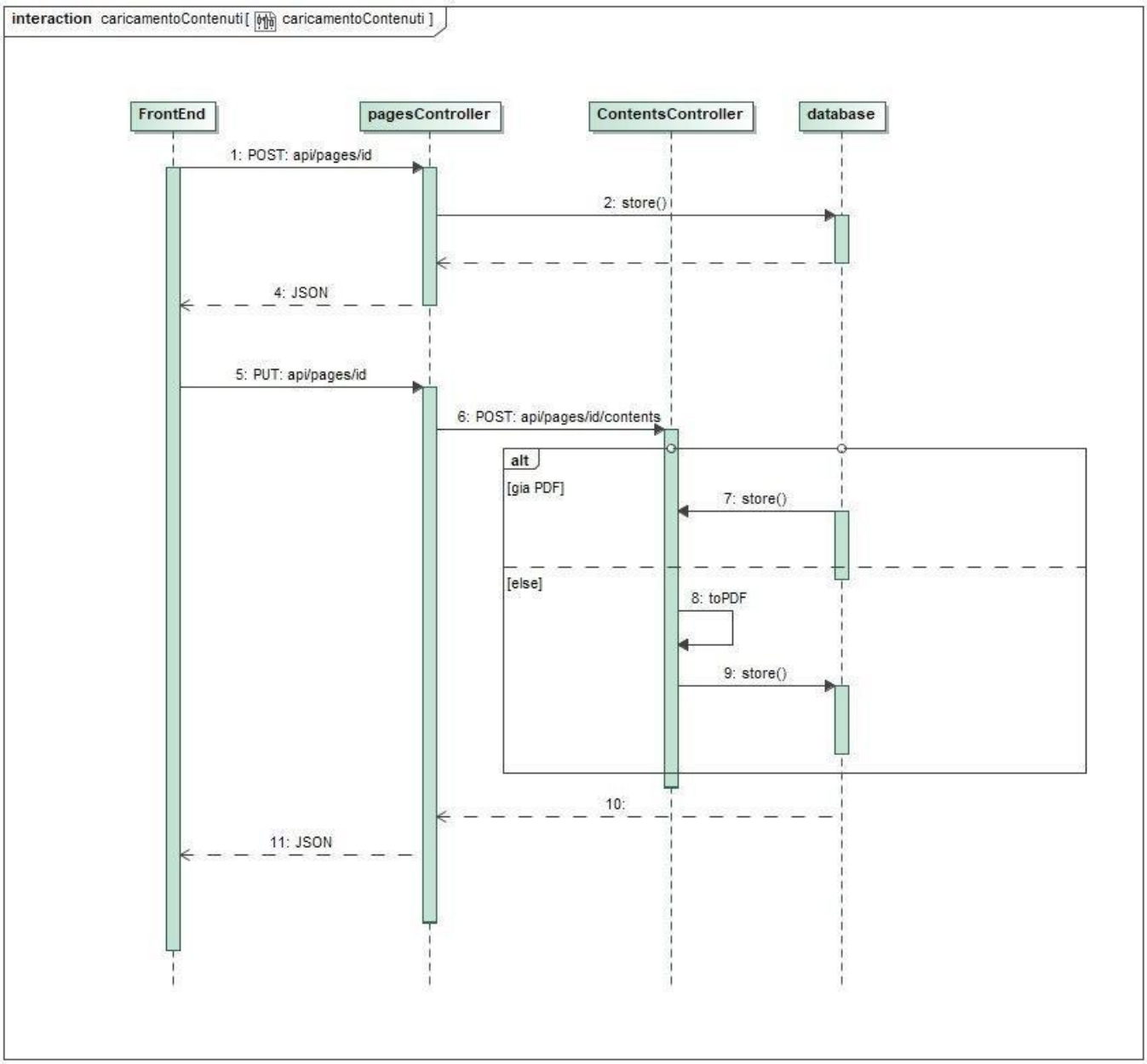
C. Architettura Software

C.1The static view of the system: Component Diagram



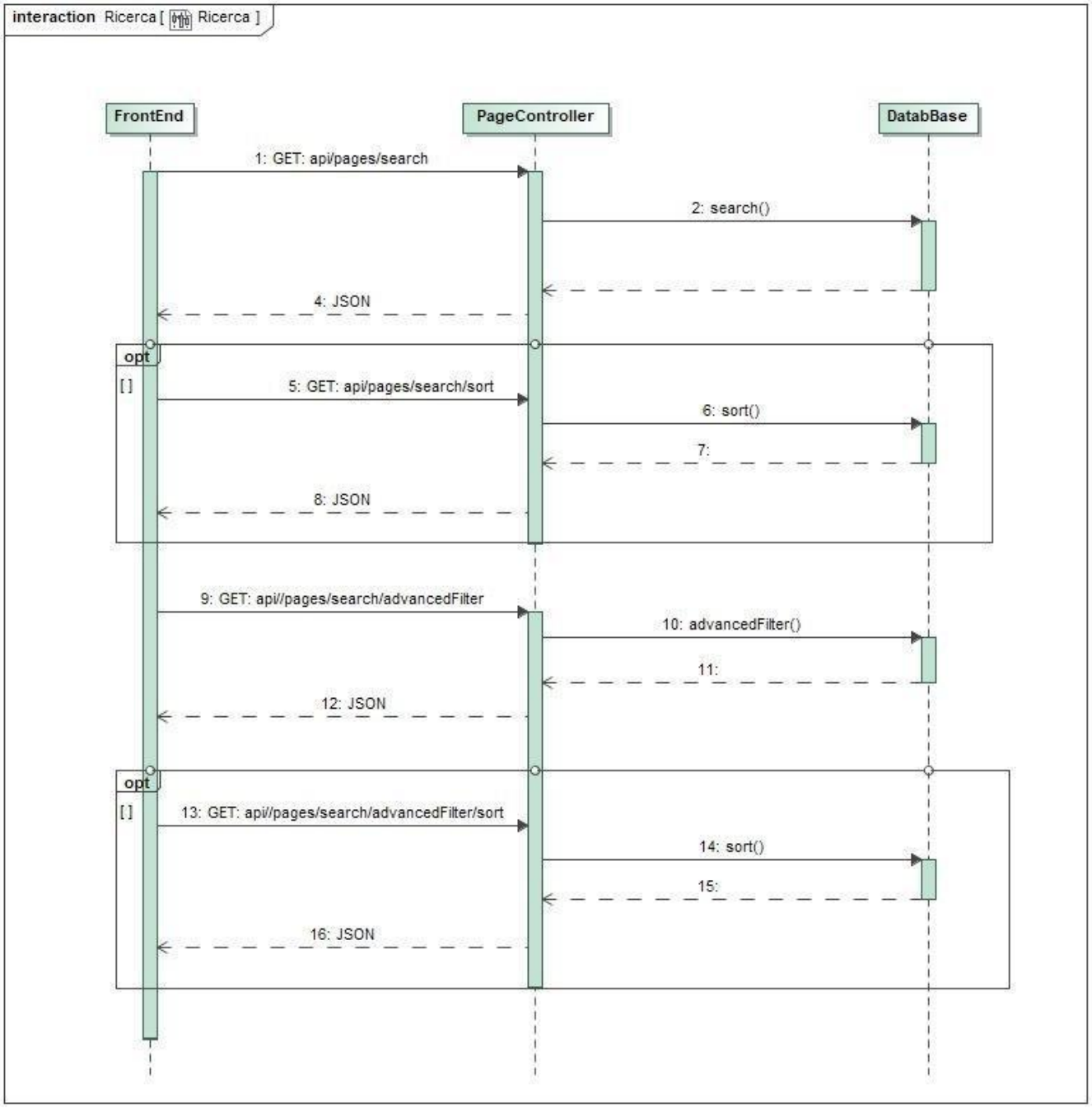
- **userController:** permette le operazioni di modifica su un profilo utente: ovvero la modifica di nome utente e password;
- Il **PageController** racchiude tutte le api per creazione e rimozione di una pagina e quelle per il caricamento, aggiornamento e rimozione dei propri appunti sfruttando il **ComponentController,** inoltre tramite le api per l'Interazione con I contentuti offrono le operazione di download, upvote e segnalazione sui contenuti ottenuti tramite ricerca (effettuata sempre dal page controller);

C.2 The dynamic view of the software architecture: Sequence Diagram 1:Caricamento Contenuti



- Il sistema permette di caricare vari tipi di appunto e di convertire più immagini in un singolo file pdf, ovviamente le api rest, per la loro natura stateless non possono richiedere o inviare file, perciò il sequence esplica il funzionamento delle route per l'interfaccia grafica corrispondenti alle api, che ovviamente restituiranno view anziche json.

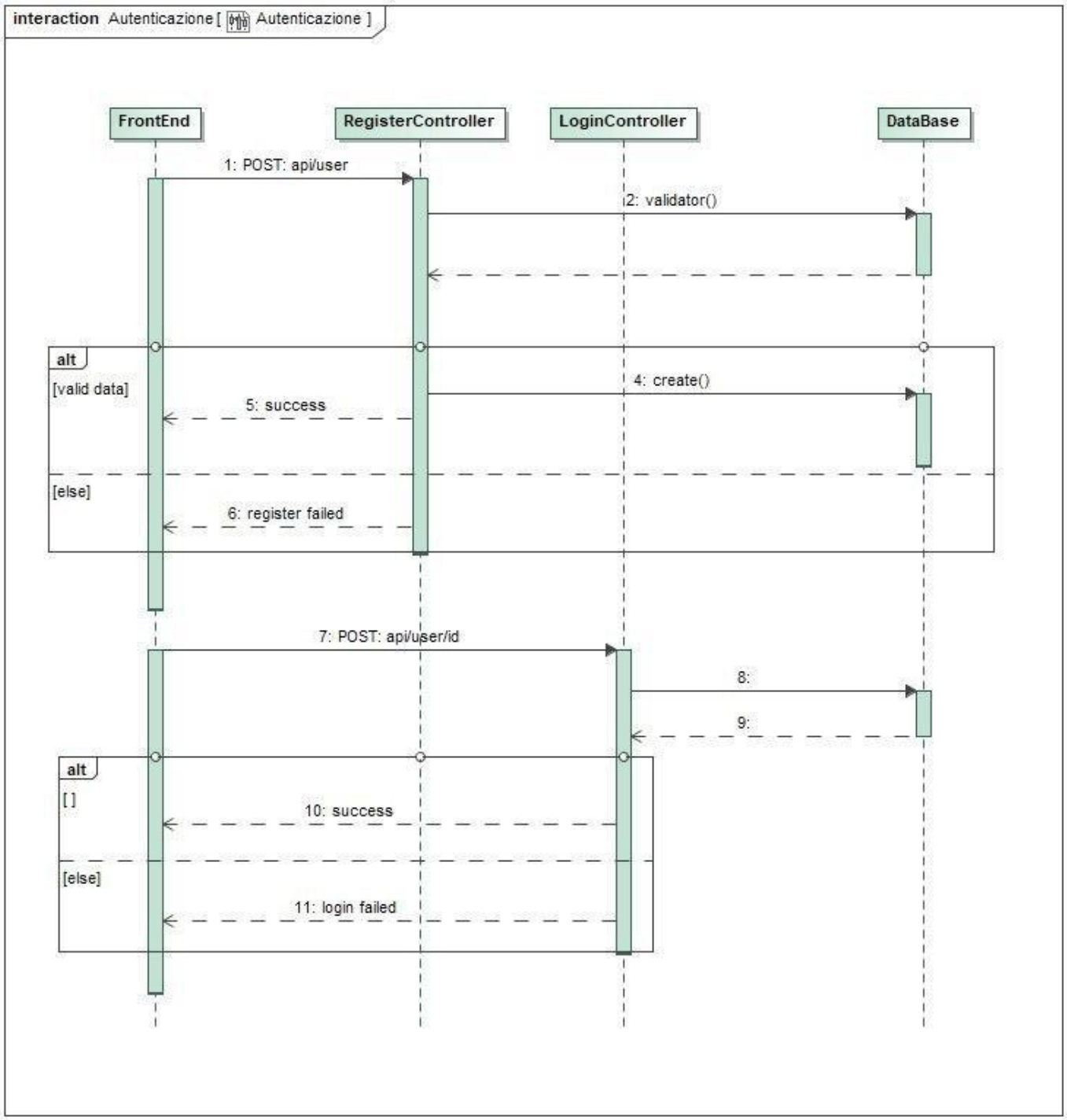
2: Ricerca



- La ricerca punta ad aumentare la velocita e la navigabilità del sito. Per questo abbiamo esplicato I due piani su cui avviene: dalla ricerca standard al filtraggio avanzato;
- Nell'mplementazione dell'interfaccia è semplice passare al metodo per il filtraggio I risultati della ricerca normale che andranno filtrati, ma, per le api rest,essendo stateless, occorrera

inviare I parametri ‘materia’ e ‘utente’ per effettuare un ricerca normale e poi filtrarla nel metodo di Filtraggio Avanzato. Per questo motivo la sequenzialità mostrata nel Sequence soprastante è valida solo nel caso si parli di interfaccia grafica e non di richieste api;

3: Autenticazione

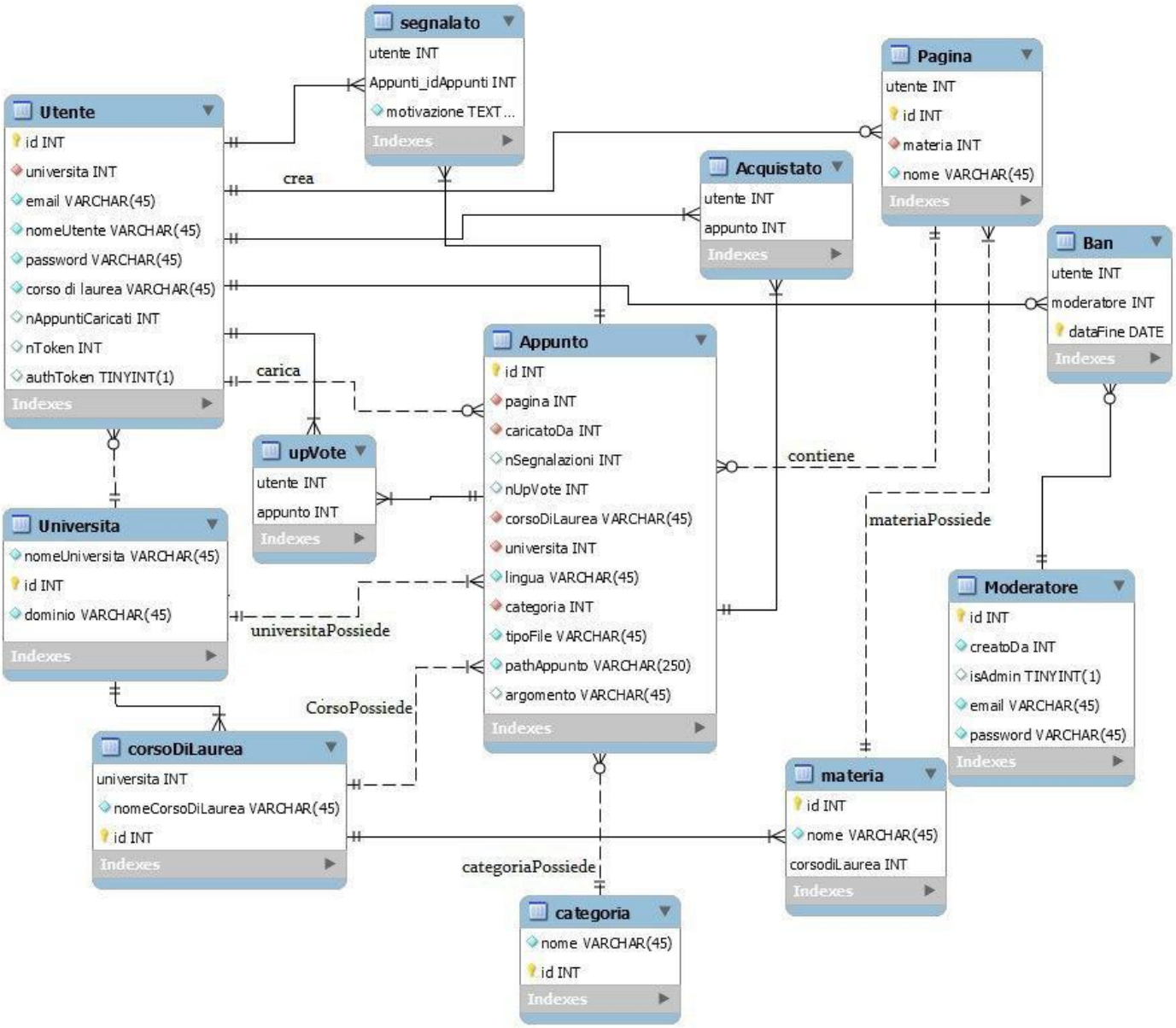


- Tra le validazioni effettuate alla registrazione va sottolineata la verifica della mail, poichè il sistema andra a controllare che il dominio indicato corrisponda a un dominio universtario

esistente nel sistema, e successivamente inviera una mail di conferma per verificare che la mail inserita, oltre ad essere una mail universitaria, esista davvero

(si rimanda al paragrafo su come abbiamo soddisfatto i requisiti non funzionali)

D. Dati e loro modellazione



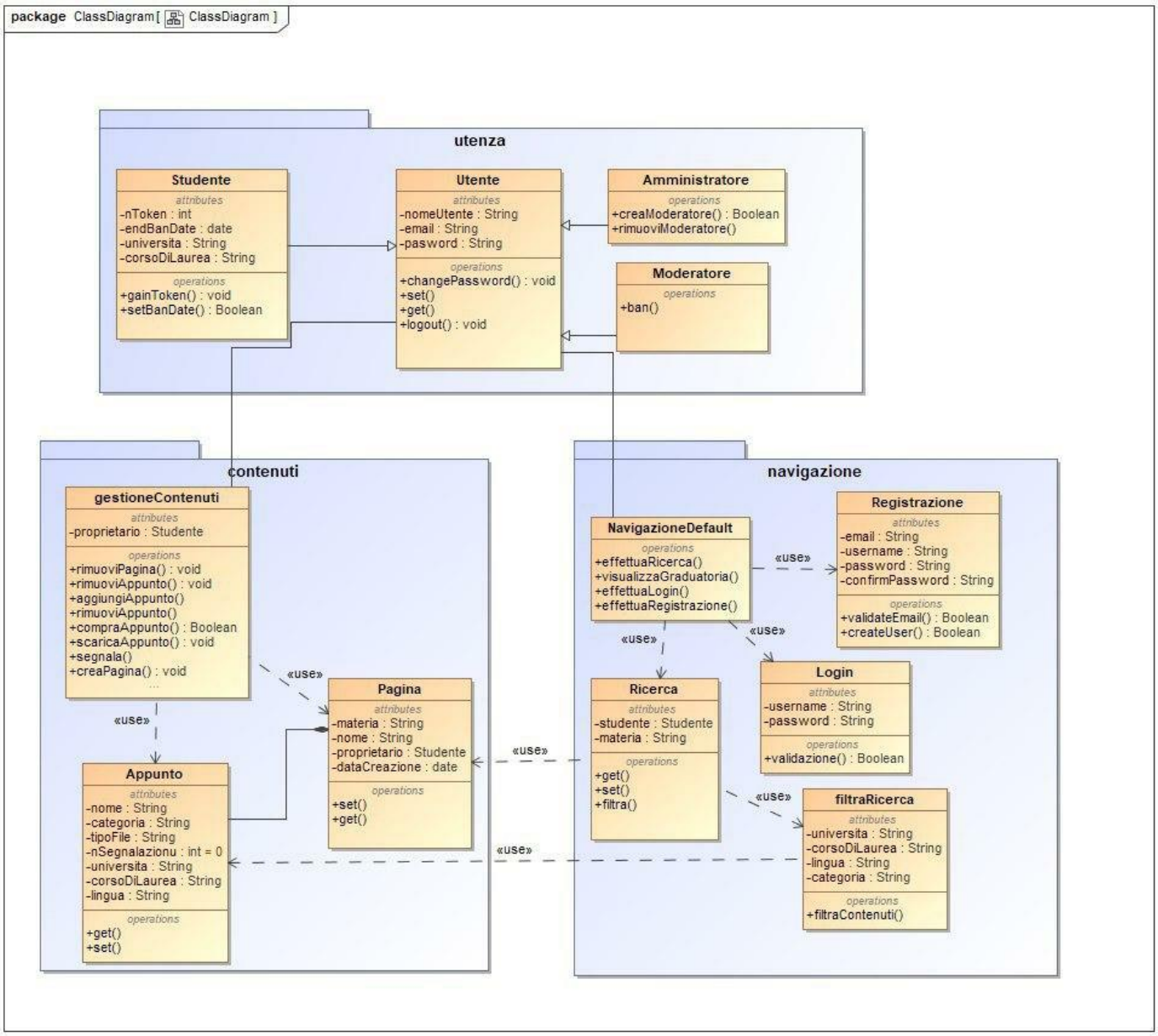
- **NFR: Diminuire Data-Entry**: lo studente al momento della registrazione si limita a fornire la propria mail universitaria e l'università che sostiene di frequentare. Il sistema tenterà il match di queste informazioni con dei sistemi universitari gia esistenti ricavandosi anche il corso di laurea e le materie trattate in esso (vedi NFR: interface).
- Nello schema sono mostrate le diverse informazioni che mirano a facilitare la **funzionalità di ricerca dei contenuti** e di conseguenza ad alleggerire le query (che andranno fatte una sulle pagine e una sui relativi appunti).
- Per la somiglianza delle informazioni del moderatore con quelle dell'**amministratore** abbiamo deciso di unirle in una singola tabella contenente l'attributo '**is admin**' per differenziare I due.
- La tabella contenente una lista di **utenti bannati** è utile al fine di controllare ogni utente al login, mentre l'attributo **'data di fine ban'** è presente allo scopo di notificare all'utente il tempo rimanente alla fine del suo divieto di autenticazione.
- La tabella **'categorie'** è utile per identificare le varie categorie disponibili per gli appunti sia al momento della **creazione di un appunto** (quando vai a inserire I suoi campi nella form) sia nel **filtraggio avanzato,** poichè deve essere una **scelta finita**: un utente non puo specificare una qualsiasi stringa come categoria di un appunto allo stesso modo in cui non può che scegliere tra una materia, poichè anche un **errore di typing**, impossibilitando il match, renderebbe introvabile il suo appunto durante un filtraggio avanzato (a meno di onerose espressioni regolari)

E. Design Decisions

1. Abbiamo deciso di implementare il sistema secondo una classica architettura a tre livelli: Client, Server, DataBase perchè la riteniamo la più adatta a un sitema di condivisione di appunti
2. Il meccanismo di gamification da noi pensato impone che I Token si guadagnino dagli aquisti e non dagli upvote per evitare il guadagno grazie agli upvote dei propri conoscenti. Inoltre, per evitare il caricamento di file inutili solo al fine di guadagnare token, aumentiamo il numero di caricamenti necessari per acquisire token e infine imponiamo che il numero di token guadagnato dall'aquisto dei propri appunti da parte di altri utenti sia inferirore ai token che essi utilizzano per comprarlo (es costa 4, ma ne guadagni solo 2) poichè altrimenti, se prezzo e guadagno coincidessero, ogni aquisto ricevuto pemetterebbe di comprare un appunto
3. La ricerca è effettuata su due livelli, prima per materia (o utente) e successivamente sulle caratteristiche degli appunti

4. Il sistema è implementato sfruttando la tecnologia delle api rest in modo da facilitare la sua modularità e futura integrazione con altri sistemi

F. Design di Basso Livello



- Il Package ‘navigazione’ tramite la classe ‘navigazione default’ offre all’utente le varie funzionalita navigazionali: in particolare la ricerca (effettuata sulle pagine) e il filtraggio (sugli

appunti contenui nelle pagine trovare con la ricerca) e presenta solo due relazioni di lettura con I contenuti

- Il package contenuti invece permette di fare ogni operazione che va a modificare pagine o appunti nel sistema, esplicate nei metodi del 'gestore contenuti'
- I permessi sulle varie operazioni (come ad esempio) che un utente non registrato non può operare sui contenuti) non sono esplicati (per via della generalizzazione utente) ma sono documentati accuratamente nei requisiti del sistema. Al fine di non rendere una ragnatela di relazioni il class diagram abbassando più del necessario il suo livello di astrazione

G. Explain how the FRs and the NFRs are satisfied by design

Usability: intuitivness

Il sistema offre all'utente un interfaccia che racchude tutte le operazioni principali che può fare quali la gestione degli appunti o del proprio account, inoltre la pagina tramite degli array, incorpora tutte le informazioni degli appunti che contiene per diminuire la complessità della ricerca anche con grandi moli di dati: infatti cercando una particalare categoria di appunto (come ad esempio "esercitazione esame" o "saggio breve") il motore di ricerca non dovrà controllare cosi che ogni appunto presente in ogni pagina matchi con le altre informazioni inserite.

Tutelability

Il controllo dell'appartenenza di uno studente a un università viene effettuato richiedendo all'utente (alla registrazione) sia la mail che la propria università per diminuire il numero di controlli effettuati (altrimenti per ogni studente andrebbero controllati tutti gli studenti di tutte le università conosciute dal sistema), il sitema tenterà un match con il dominio corrispndente e manderà, in caso di successo, una mail di conferma

Adaptabilty

Come esplicitato nel class diagram un utente comunica con I suoi contenuti tramite delle interfacce che ne racchiudono le funzionalità, facilitando sia la modifica di tali funzionalità sia l'itegrazione con altri sistemi essendo l'utenza staccata dalle funzionalità, inoltre le interfacce sono implementate sotto forma di api rest, di conseguenza restituiscono I contenuti richiesti sotto forma di Json

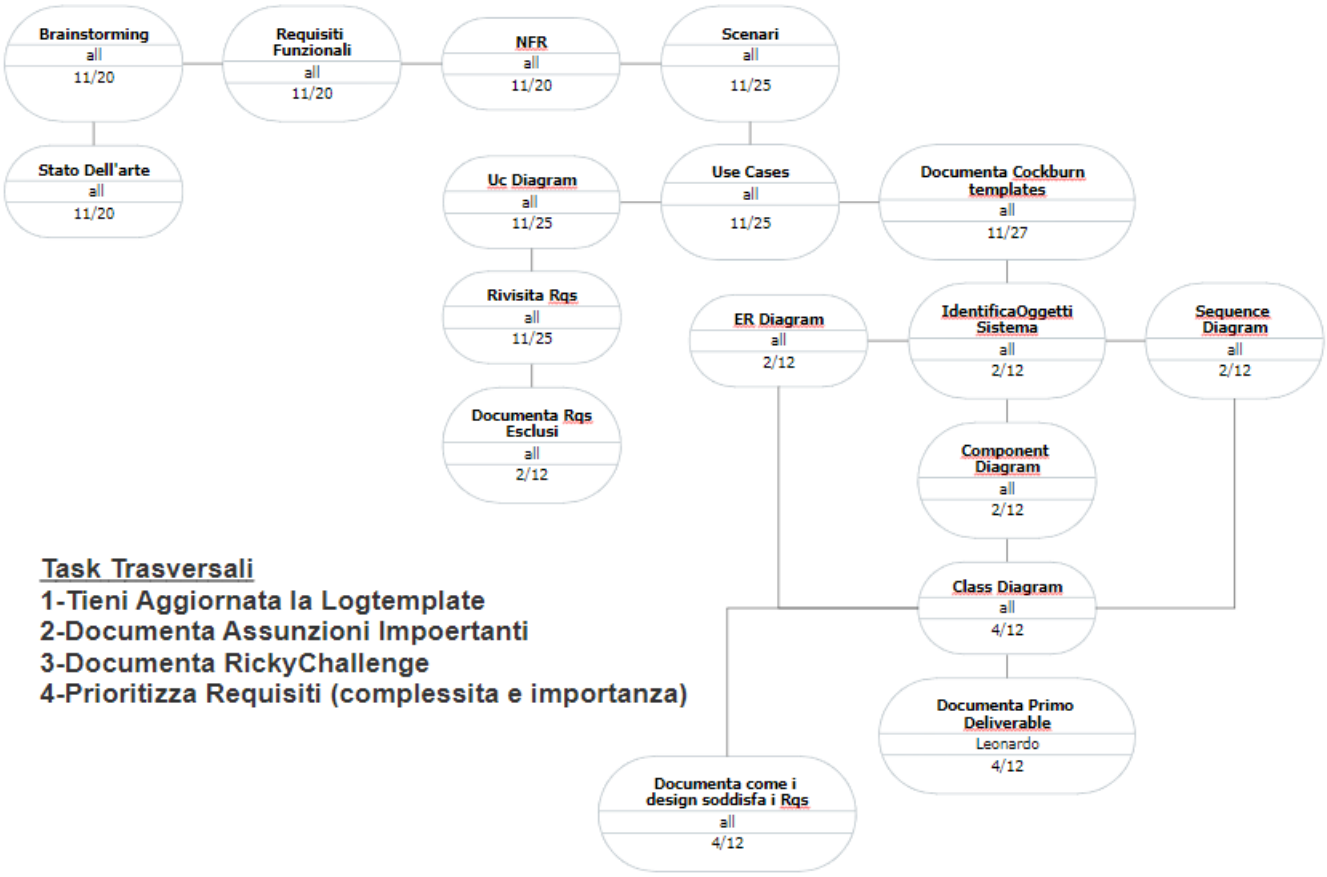
Interface (*)
Soddisfatto in parte tramite l'inserimento del campo 'dominio' all'interno della tabella 'università', come spiegto nel paragrafo relativo alle risky challenge;
Non abbiamo pero implementato il controllo della mail tramite 'mail di conferma'

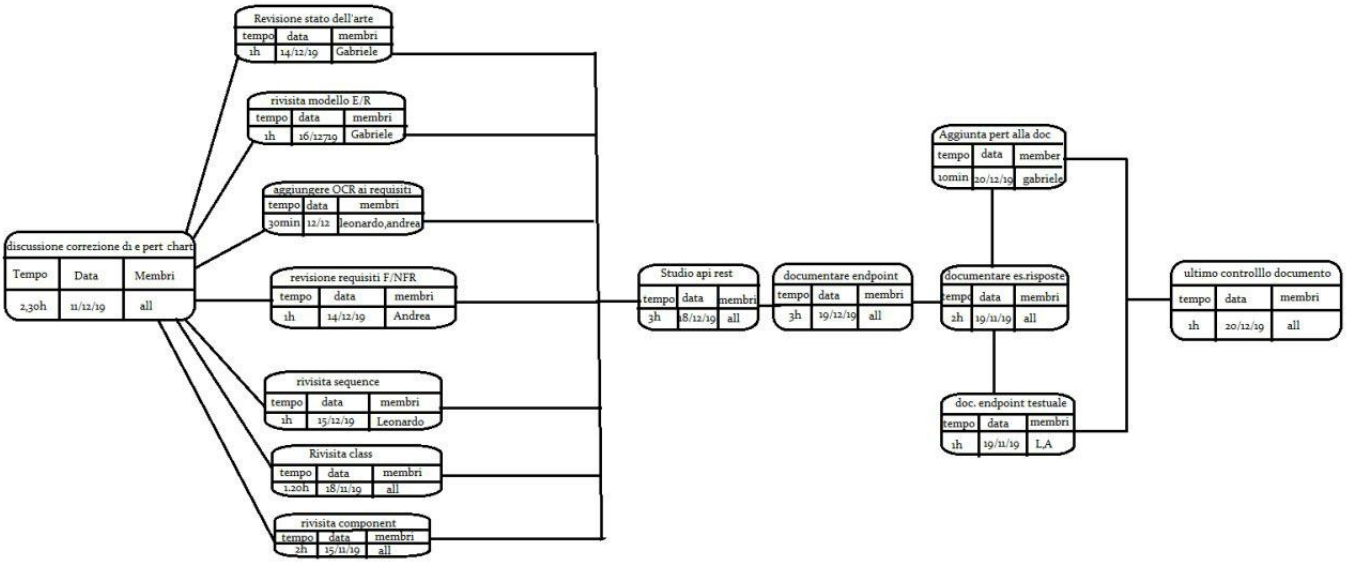
Reliability
Un dump del database ogni giorno è sufficente a limitare la perdita di dati agli ultimi caricati, che può essere considerata accettabile poichè è possibile effettuarlo nelle ore di traffico meno intense (notte) in modo da non influire troppo sull'**usability** come potrebbero fare soluzioni più pesanti applicate a dati più sensibili di quelli contenuti nel nostro sistema

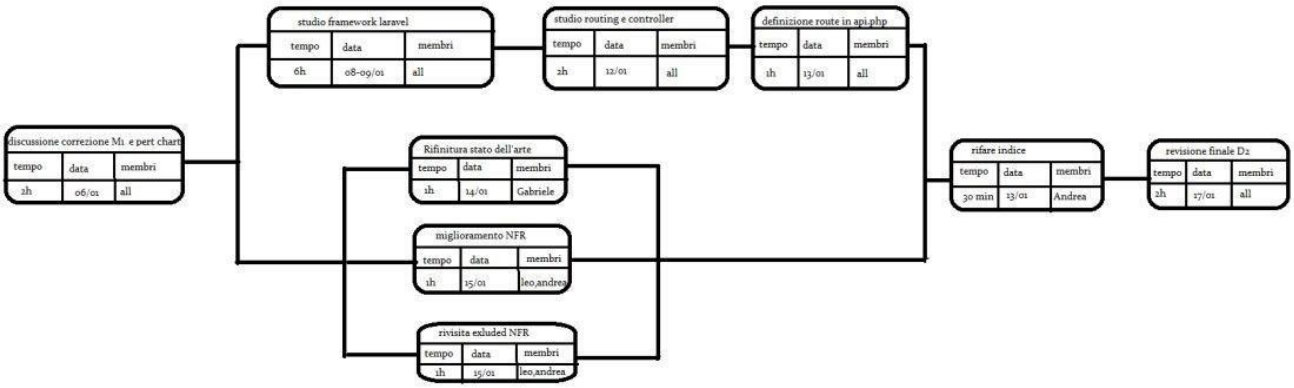
Requisito di ricerca
Soddisfatto dalla presenza di pagine e appunti e dei relativi campi; una ricerca standard andra a fare confronti con I campi delle pagine, mentre un filtraggio avanzato con I campi degli appunti delle pagine trovate dalla prima, cosi facendo il tempo di attesa sara ripartitò tra le due cosi verrà percepità una maggiore fluidità di navigazione

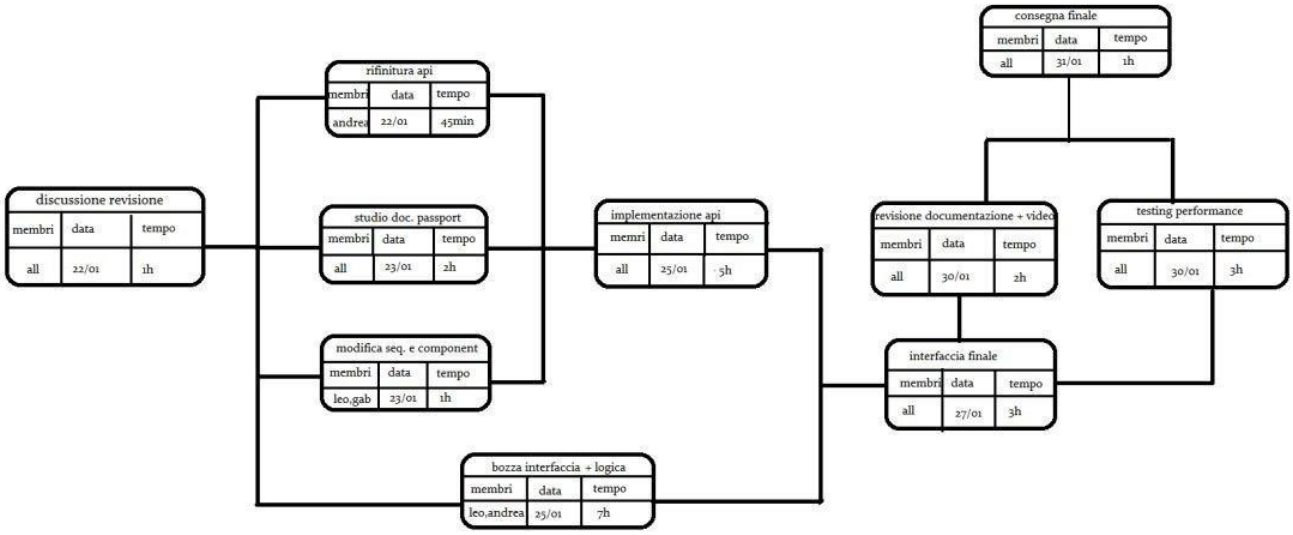
G. Effort Recording

PERT


Brainstorming
all
11/20
Requisiti Funzionali
all
11/20
NFR
all
11/20
Scenari
all
11/25
Stato Dell'arte
all
11/20
Uc Diagram
all
11/25
Use Cases
all
11/25
Documenta Cockburn templates
all
11/27
Rivisita Rqs
all
11/25
ER Diagram
all
2/12
IdentificaOggetti Sistema
all
2/12
Sequence Diagram
all
2/12
Documenta Rqs Esclusi
all
2/12
Component Diagram
all
2/12
Task Trasversali
1-Tieni Aggiornata la Logtemplate
2-Documenta Assunzioni Impoertanti
3-Documenta RickyChallenge
4-Prioritizza Requisiti (complessita e importanza)
Class Diagram
all
4/12
Documenta Primo Deliverable
Leonardo
4/12
Documenta come i design soddisfa i Rqs
all
4/12


Revisione stato dell'arte
tempo data membri
1h 14/12/19 Gabriele
rivisita modello E/R
tempo data membri
1h 16/12719 Gabriele
aggiungere OCR ai requisiti
tempo data membri
30min 12/12 leonardo,andrea
discussione correzione d1 e pert chart
Tempo Data Membri
2,30h 11/12/19 all
revisione requisiti F/NFR
tempo data membri
1h 14/12/19 Andrea
rivisita sequence
tempo data membri
1h 15/12/19 Leonardo
Rivisita class
tempo data membri
1.20h 18/11/19 all
rivisita component
tempo data membri
2h 15/11/19 all
Studio api rest
tempo data membri
3h 18/12/19 all
documentare endpoint
tempo data membri
3h 19/12/19 all
Aggiunta pert alla doc
tempo data member
10min 20/12/19 gabriele
documentare es.risposte
tempo data membri
2h 19/11/19 all
doc. endpoint testuale
tempo data membri
1h 19/11/19 L,A
ultimo controlllo documento
tempo data membri
1h 20/12/19 all


discussione correzione M1 e pert chart
tempo data membri
2h 06/01 all
studio framework laravel
tempo data membri
6h 08-09/01 all
studio routing e controller
tempo data membri
2h 12/01 all
definizione route in api.php
tempo data membri
1h 13/01 all
Rifinitura stato dell'arte
tempo data membri
1h 14/01 Gabriele
miglioramento NFR
tempo data membri
1h 15/01 leo,andrea
rivisita exluded NFR
tempo data membri
1h 15/01 leo,andrea
rifare indice
tempo data membri
30 min 13/01 Andrea
revisione finale D2
tempo data membri
2h 17/01 all



Logging

When (Month/Day)	Time spent	Partners *(please report how many people have been working)*	Brief Description of the performed task	Category	SubCategory
11 20	2h	all	Analisi delle specifiche e brainstorming	doing	brainstorming
11 20	1,5h	all	Ricerca Attori del sistema	doing	Elicitation
11 20	1h	all	Requisiti funzionali	doing	Elicitation
11 20	1h	all	Requisiti funzionali	learning	Elicitation
11 20	2h	all	Requisiti funzionali	doing	Elicitation
11 20	1h	all	Ricerca online delle varie categorie appunti	learning	Stato dell'arte
11 20	1,5h	all	Requisiti non funzionali	doing	Elicitation
11 20	30m	L	creazione pert-chart	doing	PERT
11 25	2h	all	Scritti Scenari	doing	Elicitation
11 25	1,5h	all	Rivisitato I Rqs scritti	learning	D1 - Documentazione
11 25	1,5h	all	identificato Ucs	doing	Elicitation
11 25	4.5h	all	Documenta use cases su cockburn template	doing	Elicitation
11 27	4.5h	all	Creazione UC diagram	doing	Elicitation
11 27	2h	all	esaminato stato dell'arte	learning	Stato dell'arte
11 27	1.5h	all	rivisitato NFR: aggiunti di nuovi e esclusi altri	doing	Elicitation
12 2	2.5h	L	Documenta Deliverable 1	doing	D1 - documentazione
12 2	1h	all	sequence diagram	learning	Analysis
12 2	5h	all	sequence diagram per UC principali	doing	Analysis
12 2	6h	all	strutturao una base di dati	doing	DataBase
12 4	3h	all	class diagram	Learning	Analysis
12 2	15h	all	class diagram	Doing	Analysis
12 4	2h	all	component	Learning	Analysis
12 4	12h	all	component	doing	Analysis
12 5	6h	all	conclusa Documentazione primo deliverable	doing	D1 -Documentazione
12 11	1h	G	creata pert chartt	doing	PERT
12 11	1.5h	all	discusso documentazione e assegnato task di correzione	learning	D1-risultati
12 14	1h	A	diminuito astrazione Rqs	doing	Elicitation

12	15	1h	G	approfondito stato dell'arte	doing	Stato dell'arte
12	15	30m	L	aggiunto OCR nei requisiti	doing	Elicitation
12	15	1h	L	rivisita Sequence	doing	Analysis
12	15	1.5h	all	rivisita component	doing	Analysis
12	16	1h	G	aggiornato ER diagram	doing	ER-diagram
12	16	1h	all	rivisitato class diagram	doing	System Design
12	18	2h	all	rivistazione documentazione e approfondita descrizione dei diagrammi	doing	Documentazione
12	18	9h	all	studiato desing di api rest	learning	rest api
12	18	3h	all	documentate endpoint su postman	doing	rest api
12	19	2h	all	studio request e response http	learning	Postman
12	19	2h	all	documentato su postman degli esempi di response	doing	postman
12	19	1h	all	aggiunto descrizione testuale endpoint	doing	Postman
12	19	10m	all	aggiunta pert a D2	doing	Documentazione
44	1	6	4.5h	all	discussione correzzioni deliverable 2	learning	D2
45	1	8	10h	all	studiato Laravel Framework	learning	Laravel
46	1	8	8h	all	studiato Laravel Framework	learning	Laravel
47	1	12	4.5	all	Studio Routing e Controller	learning	Laravel
48	1	13	4h	all	Implementato nel file api.h le nostre Route	Doing	Laravel
49	1	13	1.5h	all	Studio Routing e Controller	learning	Laravel
50	1	13	1.5h	all	Implementato nel file api.h le nostre Route	learning	Laravel
51	1	14	1h	G	rifinitura Stato dell'arte	doing	Stato dell'arte
52	1	14	2h	L,A	Rivisitazione NFR	doing	Elicitation
53	1	15	2h	L,A	Rivisitazione Requisiti esclusi	doing	Elicitation
54	1	15	1h	all	rielaborato indice del deliverable	doing	D3

1	22	1h	all	revisione D3	doing	rest api
1	22	2h	all	rifinitura API	doing	laravel
1	23	3h	all	studio PASSPORT	learning	laravel
1	24	2h	all	studio PASSPORT	learning	D4
1	24	1.5h	g	modifica sequence e component	doing	D4
1	25	30m	l	documentazione sequence e component	doing	GUI
1	25	4h	a g	bozza interfaccia	doing	GUI
1	25	3h	l	controller GUI	doing	GUI
1	25	2h	all	implementazione API su laravel	doing	api rest
1	27	3h	all	implementazione API su laravel	doing	api rest
1	27	3h	g	controller GUI	doing	GUI
1	28	2h	a	bozza interfaccia	doing	GUI
1	29	1h	l g	controller GUI	doing	GUI
1	30	6h	all	controller GUI	doing	GUI
1	30	4.5h	all	bozza interfaccia	doing	D4
1	31	1h	all	rivisitazione Documentazione	doing	GUI
1	31	4h	l	rifinitura finale inerfaccia	doing	GUI
1	31	3h	a	rivisitazione metodi di ricerca e filtraggio	doing	prototype
1	31	5h	all	rivisitazione metodo di inserimento appunti	doing	D4
1	25	2h	all	video funzionamento	doing	api rest

Categorization

- ***Doing***
    - ***Brainstorming***
    - ***Elicitation***
    - ***Analysis***
    - ***Laravel***
    - ***D1-Documentazione***
    - ***Pert***
    - ***Database***

  - ***System design***
  - ***Rest API***
  - ***Postman***
  - ***Documentazione deliverable***
  - ***D2***
  - ***D4***
  - ***GUI***

- ***Learning***
  - ***Elicitation***
  - ***Analysis***
  - ***Stato dell'arte***
  - ***Discussione deliverable***
  - ***Rest API***
  - ***Postman***
  - ***Summary Statistics***
  - ***Laravel***
- 5/12 Doing: 112 h +42h (D4)
- 5/12 Learning 37 + 5h (D4)

Appendix. Prototype

Descrizione testuale end point (progettati nel D2/D3)

https://documenter.getpostman.com/view/9762972/SWECXvC6?version=latest

- GET Api/users/:id
  - Se sei autenticato con il profilo corrispondente all'id ricevi le informazioni di quel profilo
- POST api/users/
- se appartieni ad una università e un corso di laurea presenti nel sistema e le informazioni inserite non corrispondono a nessun profilo presente nel sistema viene creato un nuovo profilo
- PUT api/users/:id
  - aggiorna il tuo profilo con le informazioni inserite
- GET api/users?sort=bcs
  - restituisce la graduatoria degli studenti che hanno guadagnato più token
- DELETE api/users/:id
  - Cancella il profilo, funziona solo se sei autenticato con quel profilo
- GET api/pages/:idPagina/contents/:id
  - avvia il download di un appunto se l'hai acquistato
- PUT api/pages/:idPagina/contents/:id
  - invia una segnalazione ad un appunto se non l'hai già segnalato e sei autenticato
- PUT api/pages/:idPagina/contents/:id
- aggiorna un appunto con le informazioni inserite se quell'appunto è del profilo che richiede l'aggiornamento
- DELETE api/pages/:idPagina/contents/:id
- cancella un appunto se lo possiedi ed eventualmente rimuove i token guadagnati con questo appunto
- POST api/pages/:idPagina/contents
  - carica nel sistema l'appunto inviato su una pagina che hai creato
- POST api/pages/:idPagina/contents/:id

    - se hai abbastanza token acquista l'appunto e quindi potrai scaricarlo per intero
- PUT api/contents/:id
    - ti permette di dare un solo upvote all'appunto
- DELETE api/pages/:id
    - cancella una pagina che hai creato
- POST api/pages
    - crea una pagina
- GET api/pages?subject={subject}&user={username}&course={course}&lang={lang}&uni={uni}&cat={cat}&file={filetype}&sort={order}&startIndex={index}
- ricerca tra gli appunti filtrando per materia, utente, corso di laurea, lingua, università, categoria, tipo di file e ordinare per like o data, inoltre gli amministratori possono ordinare per segnalazioni
- GET api/pages?subject={subject}&user={username}
    - ricerca tra le pagine per materia e utente
- GET api/pages/bought?idUtente
    - ricevi le pagine acquistate con il tuo profilo
- PUT api/pages/:id
    - puoi aggiornare una tua pagina inseredo le informazioni da sostituire
- GET /api/:idUtente/pages
    - puoi vedere le pagine create dal tuo profilo
- POST api/universities/courses
    - aggiunge al sistema un nuovo corso di laurea
- POST api/universities
    - aggiunge al sistema una nuova università
- PUT api/users/:id
    - banna un utente se non è già bannato
- POST api/moderators
    - crea un nuovo account moderatore
- PUT api/moderators/:id
    - promuove un account moderatore ad amministratore
- DELETE api/moderators/:id
    - cancella un account moderatore

Api realizzate nel prototipo su Laravel

https://documenter.getpostman.com/view/9765754/SWTD8H1T?version=latest