"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	*25256*	*leonardo.serilli@studentunivaq.it*
Andrea Formichetti	*255086*	*andrea.formichetti@studentunivaq.it*
Gabriele Colapelle	*252262*	*gabriele.colapelle@studentunivaq.it*

Table of Contents of this deliverable

Non è stata trovata alcuna voce d'indice.

List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
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 corso di laurea	25/11		

A. Stato dell'Arte

- **studocu.com**

Il tipo di ricerca è stato scelto prendendo spunto da studocu.com, mimizzando a due chiavi di ricerca: utente o materia;

E per quanto riguarda la ricerca avanzata abbiamo preso le categorie usate dal sito e le abbiamo rivisitate scegliendo quelle più opporune;

Inoltre abbiamo aggiunti la ricerca in base alle segnalazioni per gli scopi del moderatore.

- **appunticondivisi.com**

Abbiamo riscontrato enormi somiglianze con il sito appunticondivisi.com in un esame dello stato dell'arte avvenuto successivamente all'identificazione degli use cases. Ad esempio il meccanismo di rating e gamification

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(tramite mail e password, anche corso di laurea per il momento, poichè solo in un implementazione futura il sistema acquisirà automaicamente questa informazione) ottenendo dei token di partenza **AA**
  - Ricercare tramite parole chiavi **AB**
  - Effettuare una ricerca avanzata tramite un form in cui puoi indicare corso di laurea, utente,data, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti, 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 materia o utente **AB**
  - Effettuare una ricerca avanzata tramite un form in cui puoi indicare corso di laurea, utente, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti, 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 **AB**
  - Inserire appunti in una pagina tramite caricamento o link (pdf, immagini, audio, video) **AA**

- Valutare l'utilità del materiale altrui (tramite un upvote non modificabile); **BB**
- Ottenere token dal caricamento di appunti **AB**
- Ottenere token in base ai suoi appunti comprati dagli altri **AB**
- Aggiornare I propri appunti **MB**
- Rimuovere I propri appunti **AB**
- Rimuovere una propria pagina **MB**
- Modificare il proprio profilo **MB**
- Navigare tra I propri appunti **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 **AB**
  - Effettuare una ricerca avanzata ricerca tramite un form in cui puoi indicare corso di laurea, lingua, università, categoria (esame, esercizi, saggi, riassunti, appunti, tipo di file(video, testo, audio)) e **Segnalazione AM**
  - Ordinare I risultati per Data, upvotes (crescente o decrescente) e segnalazioni **BB**
  - Rimuovere appunti segnalati o l'intera pagina pagina **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 **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**
    - 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:** Il software potrà prevedere diverse modalità di gestione della "proprietà intellettuale di quanto caricato"

- **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.)

- **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 **(*)**

- **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
    - **NOTA**: Con il numero di dati quanto costa in tempo farne un backup? Basterebbe farlo nelle ore notturne dove il traffico di utenti è minore?

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 avanzata 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 quanto tempo deve aspettare?
    b. **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. Mmario 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

A.4 Excluded Requirements

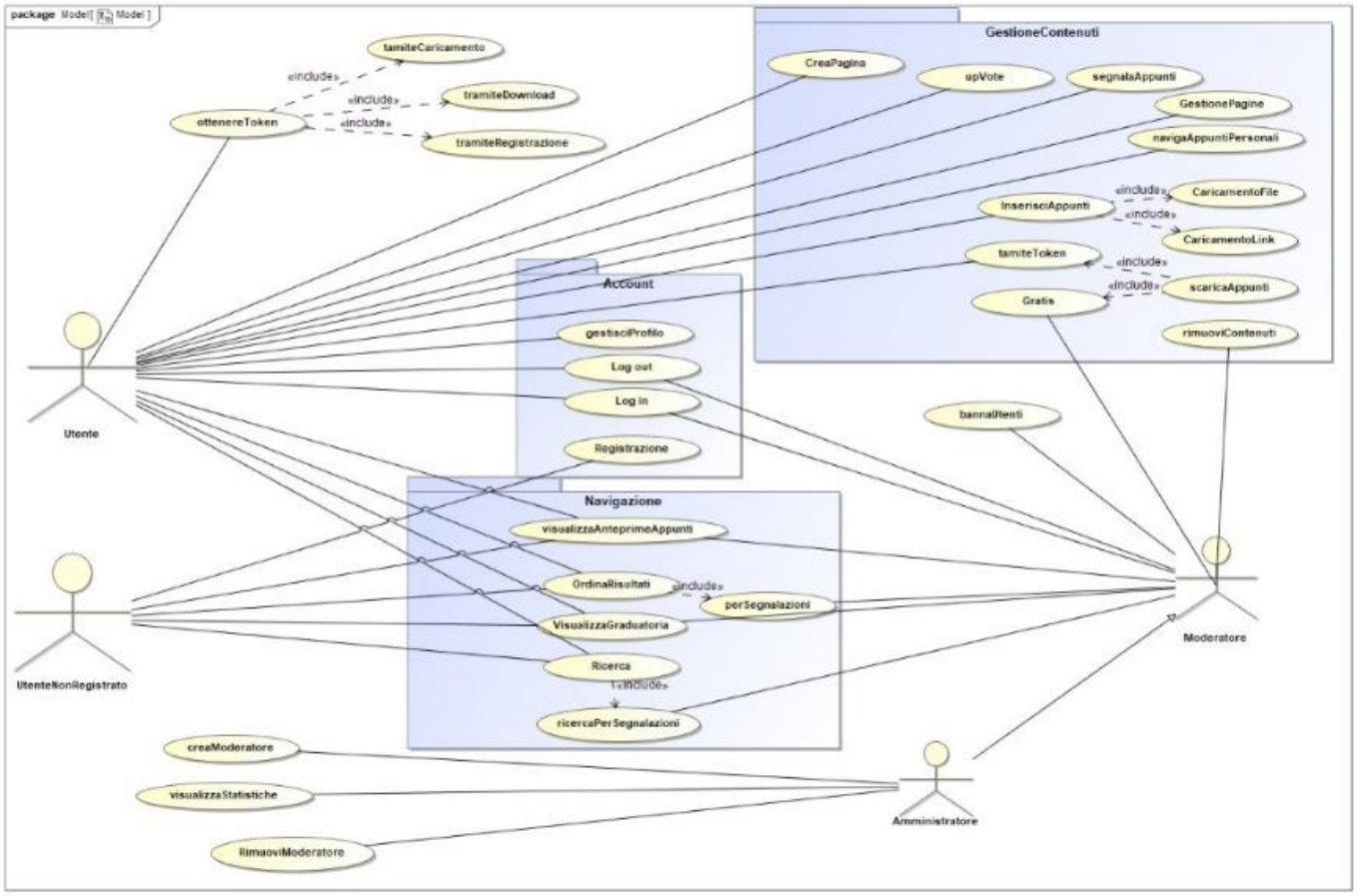
- **Traceabilty:** abbiamo escluso questo NFR perchè le informazioni di traceability non sono necessarie al suo funzionamento, ad esempio ci interessa solo tener conto del numero di segnalazioni e non chi le ha fatte e quando

- **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'

A.5 Assunzioni

- Gerarchia di utenza
- Dobbiamo assumere che la mail con la quale si registra l'utente, appartenga davvero a uno studente di una data università, in modo che ognuno possa avere un solo profilo finchè non avremmo modo di recuperare quell'informzaione (vedi NFR interface)

A.6 Use Case Diagram



USE CASE #	Ricerca	
Goal in Context	Orientarsi tra gli appunti presenti nel sistema	
Scope & Level	Primary Task: Cercare tra gli appunti presenti nel sistema Subfunction: Ordina I risultati	
Preconditions		
Success End Condition	Stampa a schermo una lista dei risultati più pertinenti alla ricerca	
Failed End Condition	1-La ricerca effettuata non ha prodotto risultati	
Primary, Secondary Actors	Attore1: utente (anche non registrato) o Attore2: moderatore	
Trigger	Un utente o un moderatore cerca qualcosa nel sistema	
DESCRIPTION	Step	Action
	1	L'utente immette una frase nella barra di ricerca
	2	Visualizza I risultati prodotti
	3	Ordina I risultati e vi naviga

EXTENSIONS	Step	Branching Action
	1	L’utente compila I campi della ricerca avanzata
SUB-VARIATIONS		Branching Action
	1	L'utente che ha effettuato la ricerca è un moderatore, e ordina I risultati per segnalazione

RELATED INFORMATION	<Use case name>
Priority:	Alta, impatta notevolmente sull'usabilità del sito
Performance	La ricerca deve produrre I risultati in tempo lineare rispetto ai campi compilati e alla lunghezza della frase chiave e deve funzianare 24 ore su 24
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 Ricerca Avanzata Ordina Risultati
Subordinates	<optional, depending on tools, links to sub.use cases>

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	Il sistema controlla il corso di laurea a cui appartiene lo studente
	3	la registrazione è avvenuta con successo e vengono forniti alcuni token iniziali al nuovo utente
EXTENSIONS	Step	Branching Action
	1	La mail è gia registrata nel sistema: rimandato alla form di log in

	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

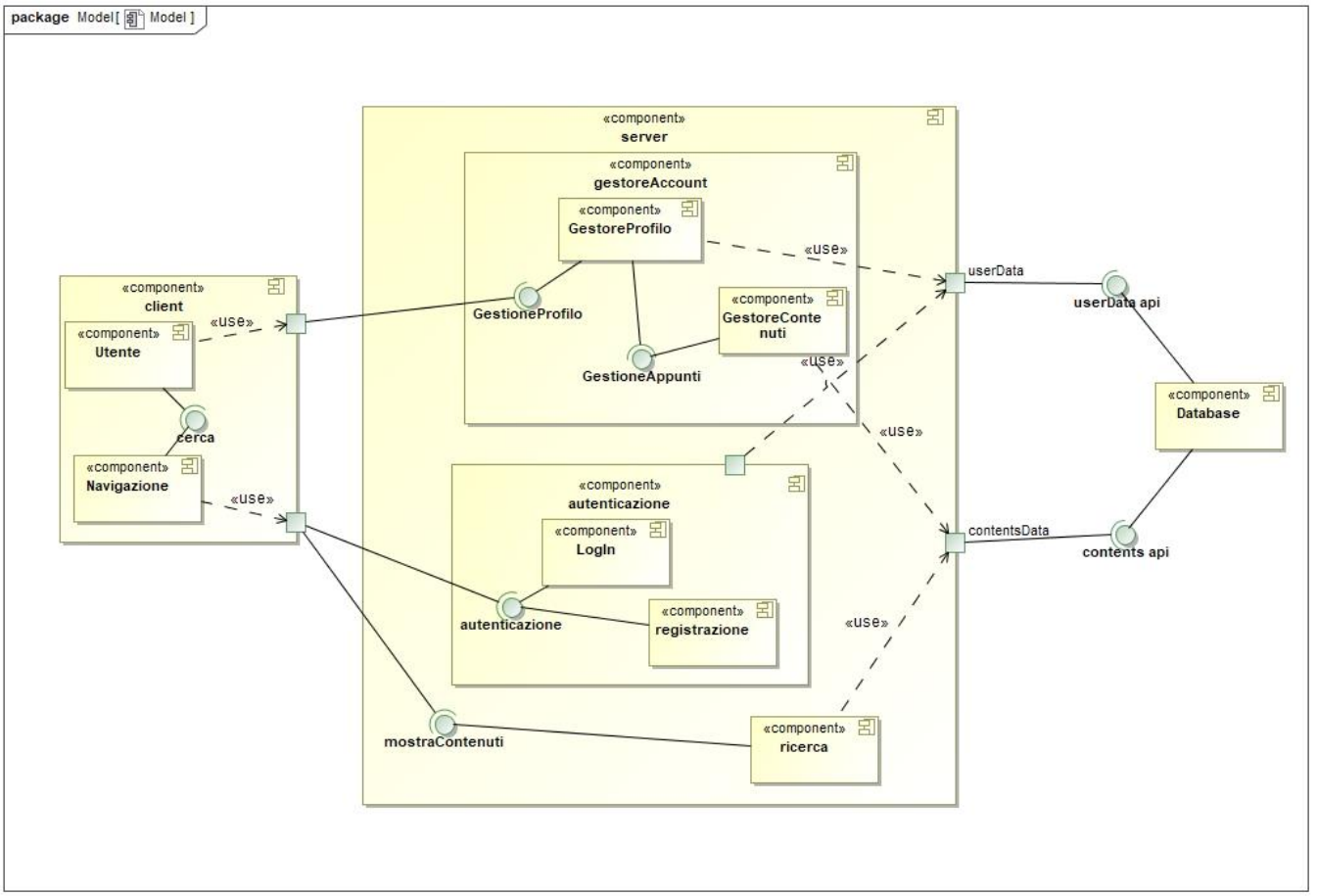
DESCRIPTION	Step	Action
	1	L'utente seleziona il file da caricare
	2	Clicca su 'carica'
	3	Verifica che il file è stato caricato
	4	Ottiene Token per il caricamento del file
EXTENSIONS	Step	Branching Action
	1	L'appunto non è stato caricato correttamente: rimanda alla pagina
	2	
SUB-VARIATIONS		Branching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

<IF RELEVANT, Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, and their related Sequence Diagrams >

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

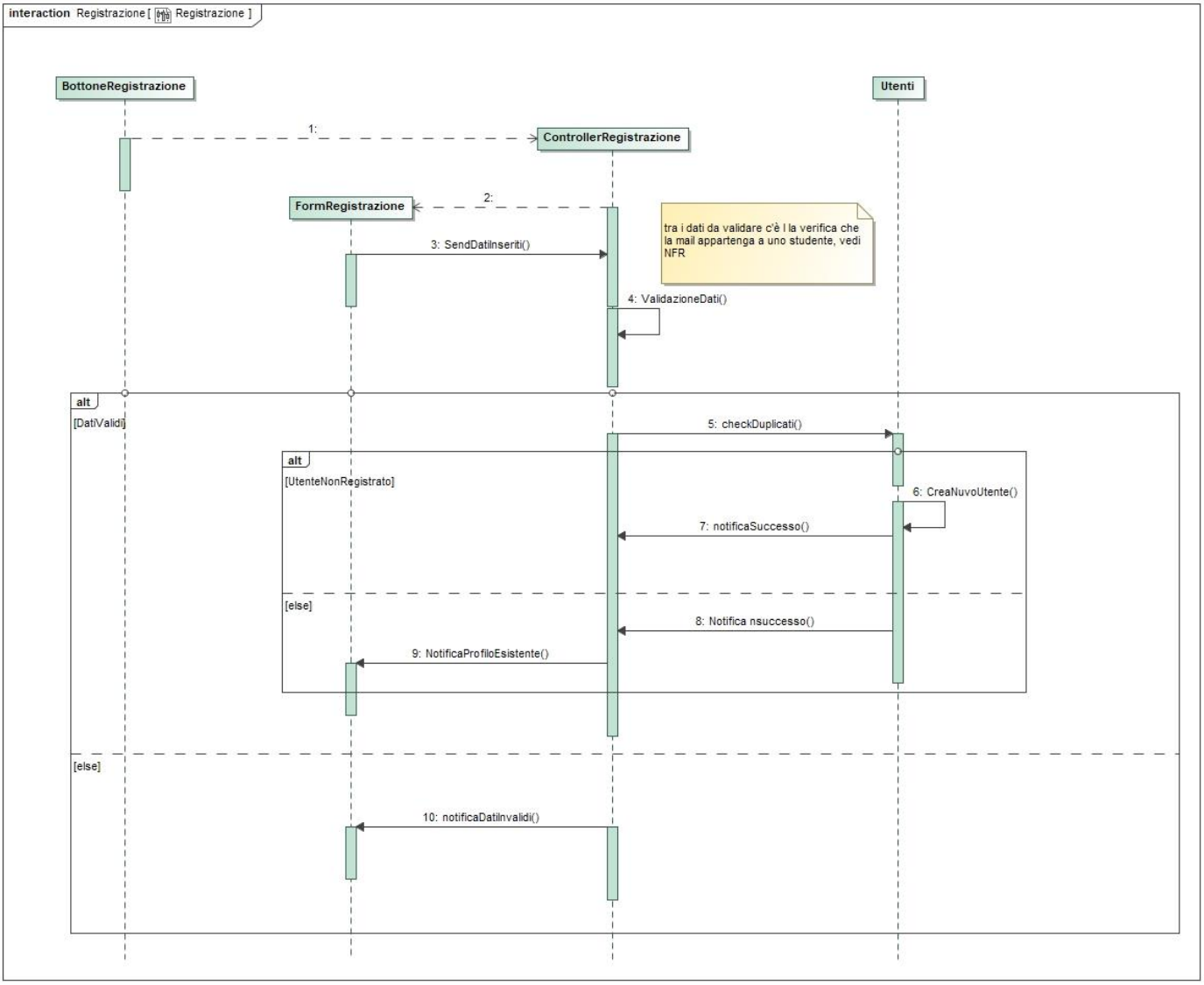


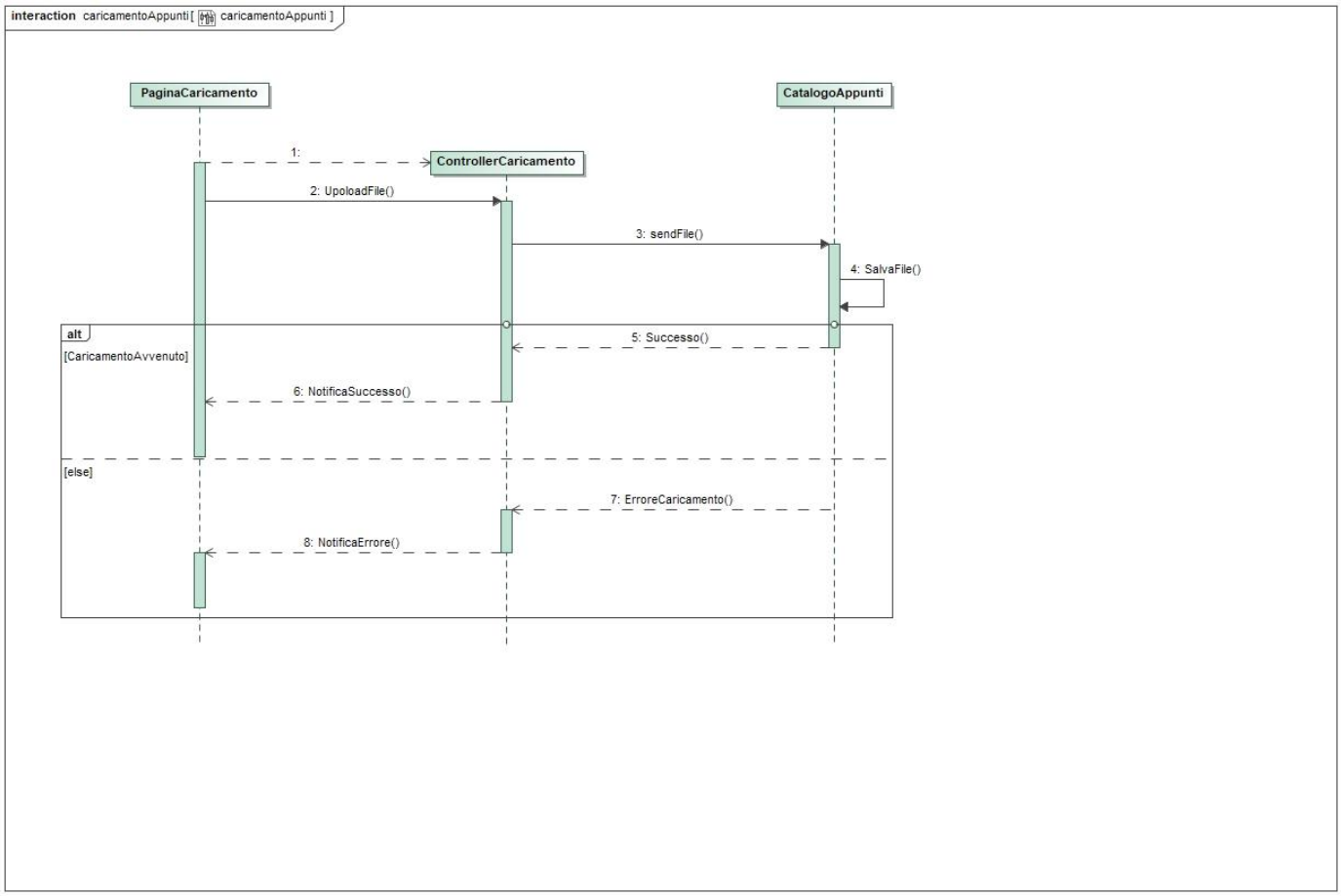
C.2 The dynamic view of the software architecture: Sequence Diagram

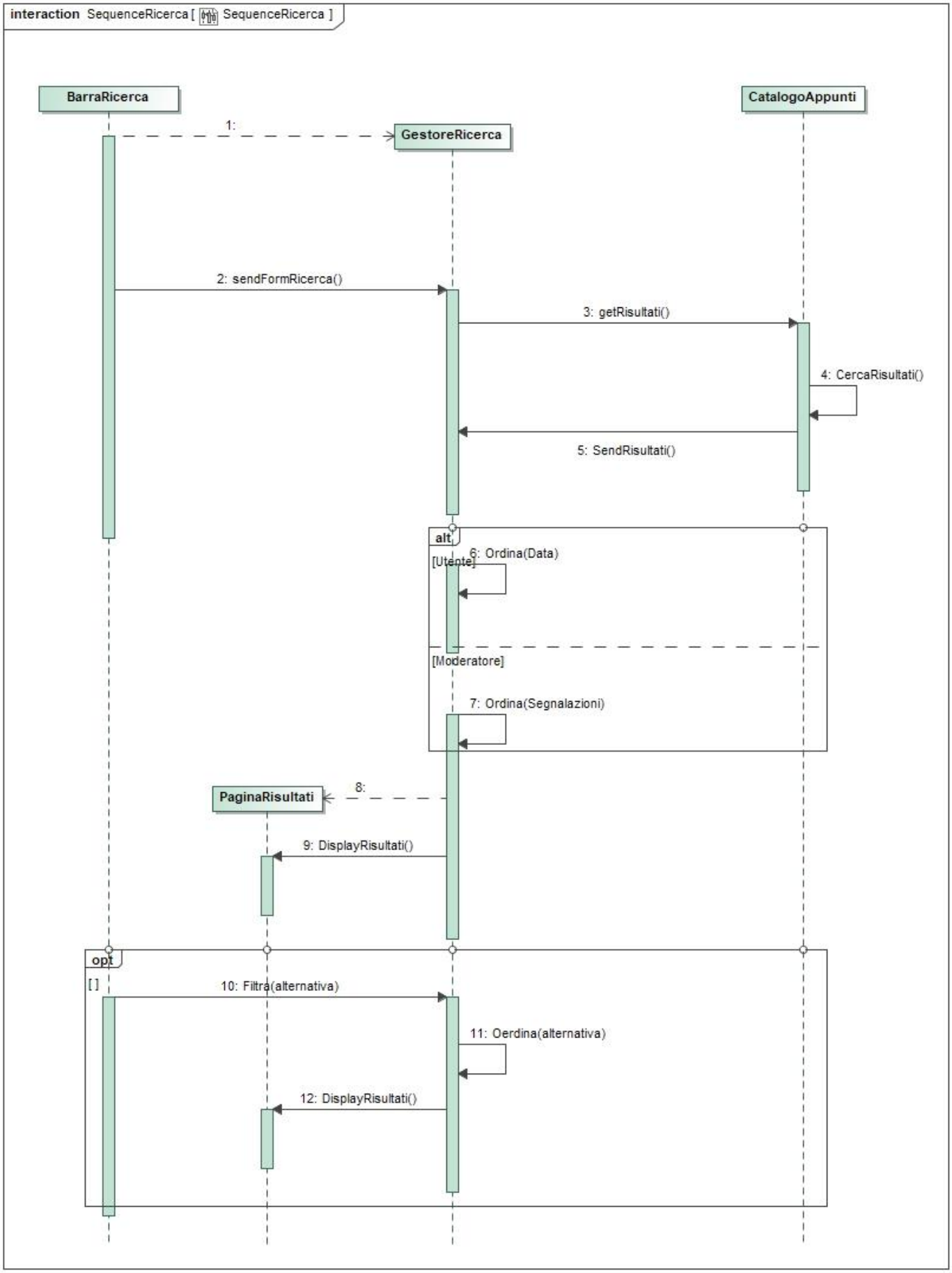
1: Registrazione

2:Caricamento Appunti

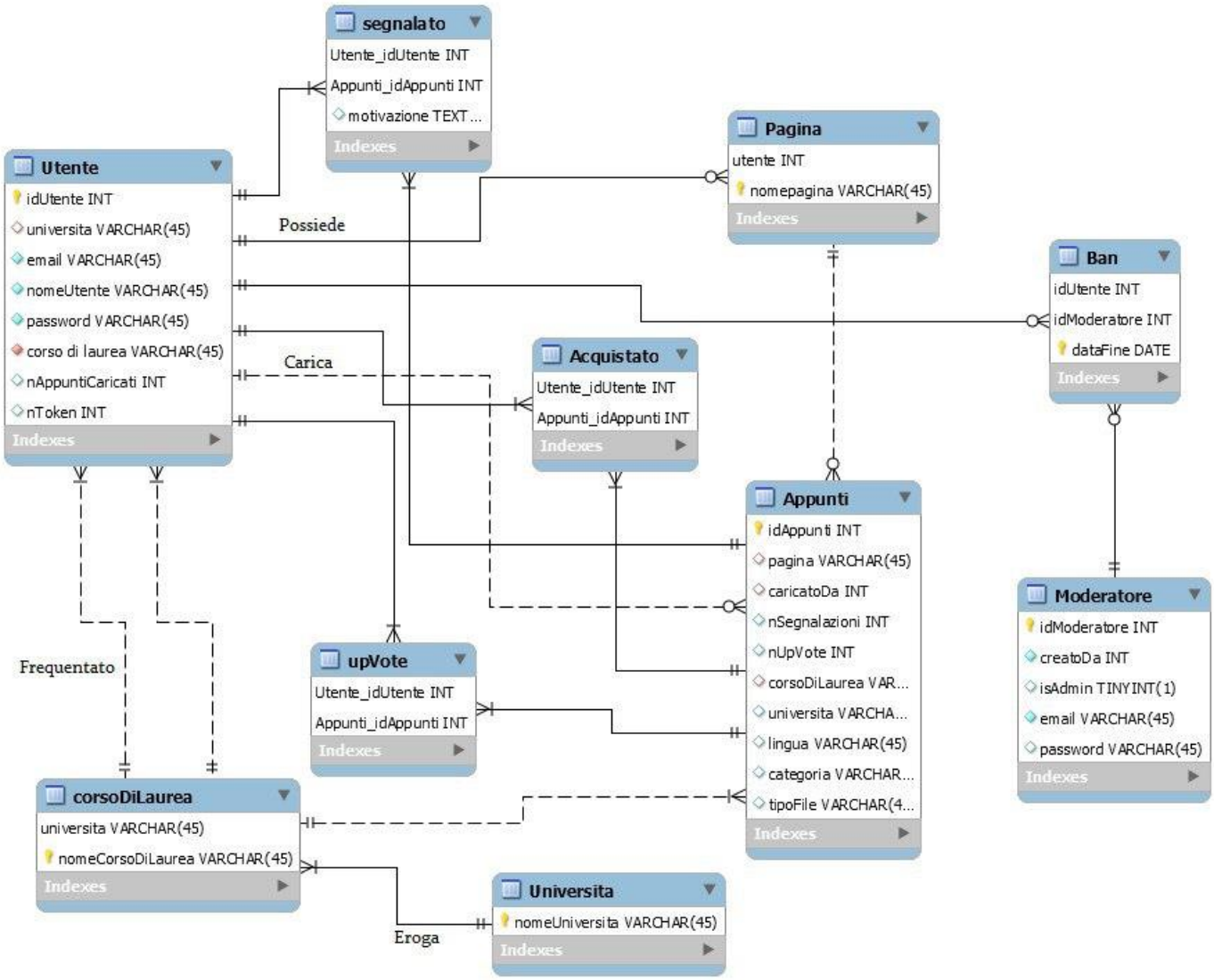
3: Ricerca


interaction Registrazione [Registrazione]
BottoneRegistrazione
ControllerRegistrazione
Utenti
FormRegistrazione
1:
2:
3: SendDatiInseriti()
tra i dati da validare c'è l la verifica che la mail appartenga a uno studente, vedi NFR
4: ValidazioneDati()
alt
[DatiValidi]
5: checkDuplicati()
alt
[UtenteNonRegistrato]
6: CreaNuovoUtente()
7: notificaSuccesso()
[else]
8: Notifica nsuccesso()
9: NotificaProfiloEsistente()
[else]
10: notificaDatiInvalidi()


interaction caricamentoAppunti [caricamentoAppunti]
PaginaCaricamento
ControllerCaricamento
CatalogoAppunti
1:
2: UpoloadFile()
3: sendFile()
4: SalvaFile()
alt
[CaricamentoAvvenuto]
5: Successo()
6: NotificaSuccesso()
[else]
7: ErroreCaricamento()
8: NotificaErrore()


interaction SequenceRicerca [SequenceRicerca]
BarraRicerca
GestoreRicerca
CatalogoAppunti
1:
2: sendFormRicerca()
3: getRisultati()
4: CercaRisultati()
5: SendRisultati()
alt
[Utente]
6: Ordina(Data)
[Moderatore]
7: Ordina(Segnalazioni)
PaginaRisultati
8:
9: DisplayRisultati()
opt
[ ]
10: Filtra(alternativa)
11: Oerdina(alternativa)
12: DisplayRisultati()

D. Dati e loro modellazione

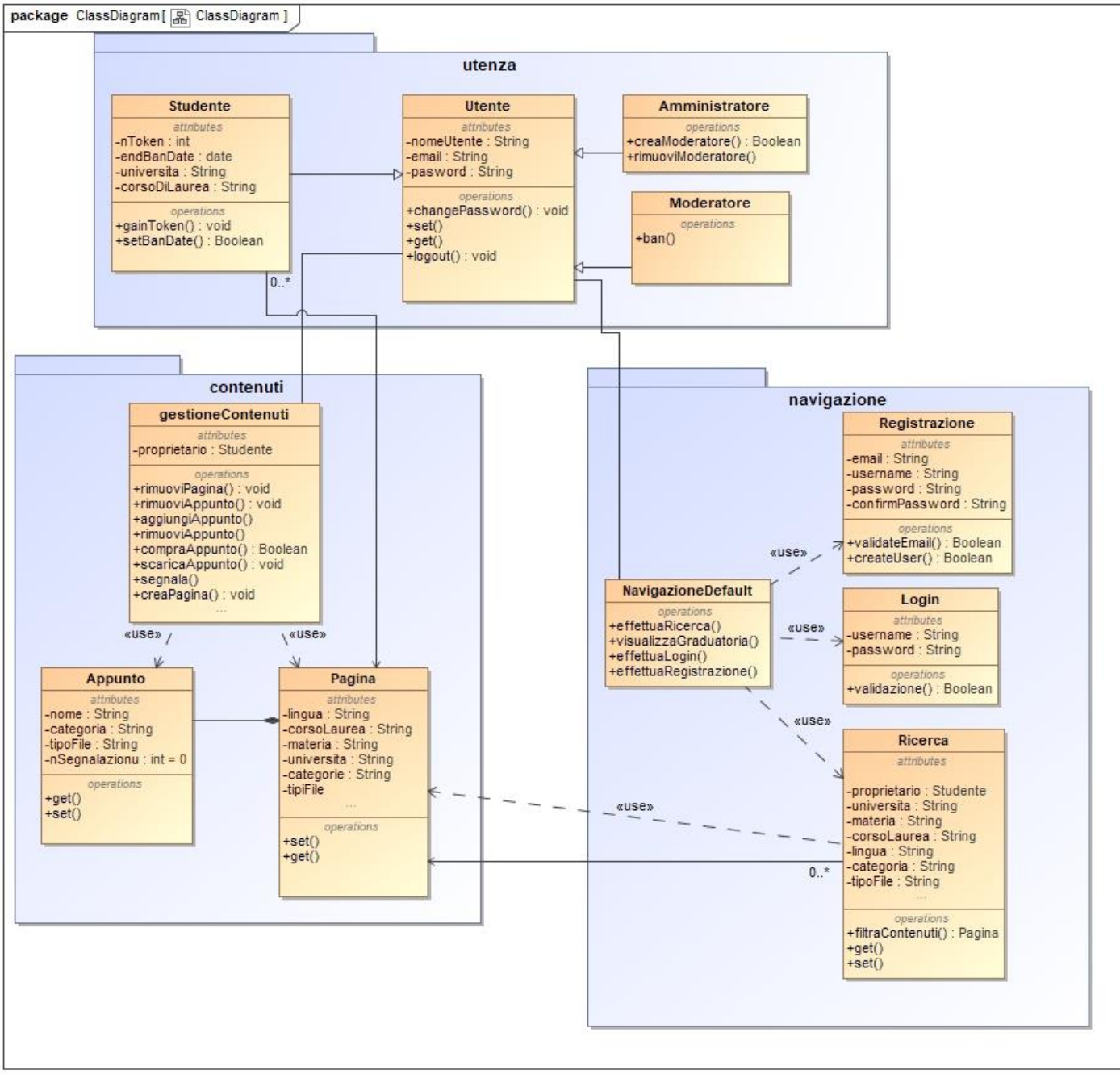


- **(*) Diminuire Data-Entry:** lo studente al momento della registrazione si limita a fornire la propria mail universitaria e l'università che sostiene di frequentare. Il sistema tenterà il match di queste informazioni con dei sistemi universitari gia esistenti ricavandosi anche il **corso di laurea.**

E. 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. 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)
4.
5.

F. Design di Basso Livello



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.

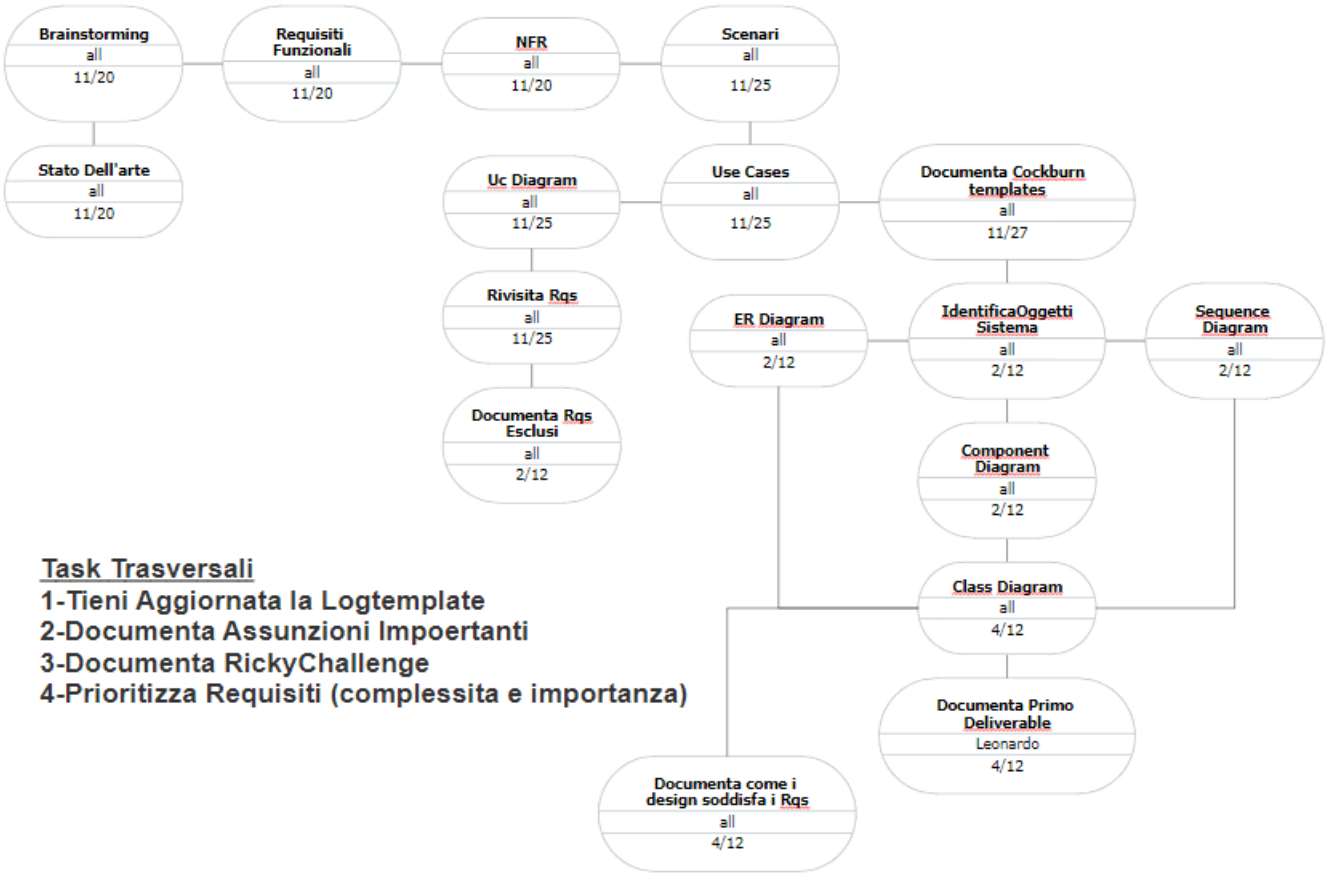
Adaptabilty
Come esplicitato nel class diagram un utente comunica con I suoi contnuti tramite delle interfacce che ne racchiudono le funzionalità, facilitando sia la modifica di tali funzionalità sia l'itegrazione con altri sistemi essendo l'utenza staccata dalle funzionalità

Interface (*)
Al momento non soddisfatto, occorre un iterfaccia per interagire con sistemi universitari per effettuare il controllo di appartenza di uno studente e ciò richiede uno studio approfondito di questi ultimi. Necessario anche per soddisfare il requisisto di **Registrazione**

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

G. Effort Recording

PERT



Logging

Personal Journal

Team (number and name): LAG

Student name:Leonardo Serilli , Andrea Formichetti , Gabriele Colapelle
Student number: 3
Email: leonardo.serilli@studentunivaq.it , andrea.formichetti@studentunivaq.it , gabriele.colapelle@studentunivaq.it

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

Categorization

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

- ***Learning***
    - ***Elicitation***
    - ***Analysis***
    - ***Stato dell'arte***

Summary Statistics

- 5/12 Doing: 66h
- 5/12 Learning: 11.5h

Appendix. Prototype

<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>