

UNINOTES

*Progetto Ingegneria
del Software*

Il Sistema da realizzare consiste in una applicazione software per lo scambio di appunti e materiali didattici. Su questa applicazione si potranno caricare appunti, scaricarne/acquistarne altri ed effettuare altre operazioni.

Sommario

1.	Ideazione	4
*	Posizionamento	4
	Opportunità di business	4
	Formulazione del problema	4
	Formulazione della posizione del prodotto.....	4
*	Descrizione delle parti interessate.....	4
*	Riepilogo delle caratteristiche del sistema	5
*	Elenco regole.....	5
*	Definizioni del glossario.....	6
*	Requisiti	7
*	Obiettivi e casi d'uso	9
*	Modello dei casi d'uso.....	11
	UC3: Autenticarsi al servizio come utente	11
	UC5: Acquistare un appunto.....	13
	UC9. Accettare le richieste di appunti.....	14
	UC1: Visualizzare il catalogo degli appunti.....	15
	UC2: Registrazione nel sistema	15
	UC4: Visualizzare informazioni su un appunto	16
	UC6. Inserire la recensione di un appunto.....	16
	UC7. Inserire un nuovo appunto	16
	UC8. Autenticarsi al servizio come amministratore di sistema.....	16
	UC10: Eliminare una recensione	17
	UC11: Approvare/Disapprovare una recensione	17
	UC12: Gestire le materie/docenti.....	18
2.	Iterazione 1 – UniNotes	19
*	Introduzione	19
*	Aggiornamento del caso d'uso UC7	19
	UC7: Inserisci nuovo appunto	19
*	Analisi orientata agli oggetti.....	20
	Modello di Dominio	21
	Diagramma di sequenza di sistema	22
	Contratti delle operazioni	23
*	Progettazione.....	25

Diagramma delle classi.....	26
Diagrammi di Sequenza.....	27
* Refactoring.....	29
3. Iterazione 2 – UniNotes.....	30
* Introduzione	30
* Aggiornamento del caso d'uso UC5	30
UC5: Acquistare un appunto.....	30
* Analisi orientata agli Oggetti	31
Modello di Dominio	31
Diagramma di sequenza di sistema	32
Contratti delle operazioni	33
* Progettazione.....	35
Diagramma delle classi.....	36
Diagrammi di Sequenza.....	37
* Refactoring.....	39
4. Iterazione 3 – UniNotes.....	41
* Introduzione	41
* UC9. Accettare le richieste di appunti.....	41
* Analisi orientata agli Oggetti	43
Modello di dominio	43
Diagrammi di sequenza di sistema.....	44
Contratti delle operazioni	45
* Progettazione.....	46
Diagramma delle classi.....	46
Diagrammi di sequenza	47
* Refactoring.....	49
5. Iterazione 4 – UniNotes.....	51
* Introduzione	51
* Analisi orientata agli Oggetti	52
Modello di dominio	52
Diagrammi di sequenza di sistema.....	53
Contratti delle operazioni	54
* Progettazione.....	54
Diagramma delle classi.....	55
Diagrammi di sequenza	56

* Refactoring.....	57
6. Iterazione 5 – UniNotes.....	59
* Introduzione	59
* Analisi orientata agli Oggetti	59
Modello di dominio	59
Diagrammi di sequenza di sistema.....	60
Contratti delle operazioni	61
* Progettazione.....	61
Diagramma delle classi.....	62
Diagrammi di sequenza	63
* Refactoring.....	63
7. TESTING.....	64
Test Uninotes	64
Test Amministratore.....	64
Test Facoltà.....	64
Test Utente.....	65
Test Materia	65
Test Gestione Pagamento	65
Test Login	65

Applicazione UniNotes

1. Ideazione

Il Sistema da realizzare consiste in una applicazione software per lo scambio di appunti e materiali didattici. Su questa applicazione si potranno caricare appunti, riscattarne/acquistarne altri ed effettuare altre operazioni. L'obiettivo è quello di facilitare lo studio degli studenti universitari iscritti a questa applicazione, gestendo il caricamento e l'acquisto dei materiali di cui sopra.

Questa applicazione è pensata per tutti gli studenti di tutte le facoltà dell'Università degli studi di Catania.

* *Posizionamento*

Opportunità di business

Il software UniNotes si pone l'obiettivo di agevolare le operazioni di acquisto di appunti e/o dispense didattiche, sostituendo lo scambio fisico di questi materiali con uno scambio digitale. In questo modo sarà possibile agevolare gli studenti che si trovano lontano dalla sede universitaria di appartenenza.

Formulazione del problema

A causa delle condizioni pandemiche in cui versa il nostro Paese, diventa difficoltoso spostarsi da una città all'altra per reperire il materiale didattico. In questo modo gli studenti avranno accesso alle risorse universitarie semplicemente acquistandole online.

Formulazione della posizione del prodotto

Il software è rivolto a tutti gli studenti universitari che vogliono usufruire di questo servizio.

* *Descrizione delle parti interessate*

- I. Amministratore di sistema
- II. Utente non registrato
- III. Utente registrato

L'amministratore di sistema si occupa dell'approvazione/rifiuto del materiale caricato. Gli appunti/dispense dovranno essere inoltrati all'Amministratore che provvederà al caricamento o all'eliminazione degli stessi.

L'utente non registrato deve effettuare la registrazione ad UniNotes per poter usufruire dei servizi offerti.

L'**utente registrato** è l'utilizzatore principale dell'applicazione. Quest'ultimo può accedere a tutti i servizi: oltre ai servizi di acquisto di appunti/dispense, l'utente registrato può caricare i suoi materiali, recensire appunti/dispense acquistati, accumulare punti bonus per l'acquisto di ulteriori materiali.

* **Riepilogo delle caratteristiche del sistema**

Il sistema deve possedere le seguenti caratteristiche e permettere di:

- Visualizzare il catalogo degli appunti disponibili
- Registrare l'utente al servizio
- Autenticare al servizio come studente registrato
- Autenticare al servizio come amministratore di sistema
- Acquistare/riscattare appunti/dispense
- Scrivere recensioni a file di appunti
- Accettare o rifiutare le richieste di appunti
- Accettare o rifiutare le recensioni
- Inserire un nuovo appunto
- Inserire una nuova materia/docente

* **Elenco regole**

ID	REGOLA	MODIFICABILITÀ	SORGENTE
R1	Ogni utente può recensire ogni appunto acquistato al massimo una volta.	Bassa	Politica di UniNotes
R2	Il numero di punti bonus ottenuti al caricamento di un documento è proporzionale al numero di stelle medio delle recensioni fatte dagli altri utenti sui propri appunti. Se si tratta della prima pubblicazione, si riceveranno solo 2 punti bonus. Se invece si tratta di pubblicazioni successive (e l'utente ha ricevuto dei feedback: 1/5 stelle) ai punti bonus base verranno aggiunti un numero	Bassa	Politica di UniNotes

	<p>di punti relativo al numero di stelle:</p> <ul style="list-style-type: none"> • 1 stella: 1 punto • 2 stelle: 3 punti • 3 stelle: 5 punti • 4 stelle: 8 punti • 5 stelle: 12 punti 		
R3	Ogni utente può caricare un determinato blocco di appunti una sola volta.	Bassa	Politica di UniNotes

* ***Definizioni del glossario***

Termine	Definizione
<i>Amministratore</i>	Gestore della pubblicazione degli appunti
<i>Appunto</i>	Documento digitale contenente gli appunti riguardanti determinati argomenti di una certa materia.
<i>Materia</i>	Per materia si intende un qualsiasi settore di appartenenza di un determinato appunto.
<i>Recensione</i>	Feedback correlato alla recensione (1 stella – pessimo, 5 stelle – eccellente)
<i>Punti Bonus</i>	Punti assegnati ogni volta che avviene il caricamento di un documento. I punti bonus si possono utilizzare per acquisire gratuitamente dei file di appunti.
<i>Utente registrato</i>	Persona registrata al servizio: può visualizzare il catalogo appunti, caricare appunti ricevendo punti bonus, acquistare/riscattare appunti e recensire appunti scaricati
<i>Utente non registrato</i>	Persona non registrata al servizio
<i>Servizio mail</i>	Servizio che si occupa di gestire la posta elettronica inviando mail contenenti: <ul style="list-style-type: none"> - Link appunto acquistato - Conferma registrazione - Aggiornamento punti utente - Approvazione appunto

* ***Requisiti***

Il Sistema da realizzare consiste in una applicazione software per lo scambio di appunti e materiali didattici creata per l’Università degli Studi di Catania. Su questa applicazione si potranno caricare appunti, acquistarne/riscattarne altri ed effettuare altre operazioni.

Ogni file di appunti deve avere indicato il nome della materia, e il titolo dell’argomento: per ogni file è indicato il numero delle recensioni di quell’appunto e la sua valutazione media.

Ogni utente può acquistare appunti e recensire appunti da lui comprati/riscattati.

Caricando degli appunti si ricevono dei punti: in base alle recensioni che si otterranno in seguito alla pubblicazione, l’utente riceverà più o meno punti. Se ad esempio si tratta della prima pubblicazione, si riceveranno solo 2 punti bonus. Se invece si tratta di pubblicazioni successive (e l’utente ha ricevuto dei feedback: 1/5 stelle) ai punti bonus base verranno aggiunti un numero di punti relativo al numero di stelle:

- 1 stella: 1 punto
- 2 stelle: 3 punti
- 3 stelle: 5 punti
- 4 stelle: 8 punti
- 5 stelle: 12 punti

Ogni file di appunti richiede un numero di punti pari a 5 per il riscatto (se non si possiedono l’appunto viene pagato con la carta di credito registrata a nome dell’utente, per un importo pari a 5,00 euro): caricando appunti (in base ai feedback ricevuti) si potranno acquistare altri appunti gratuitamente.

Il sistema deve possedere:

- Un’area di autenticazione al sistema (login).
- Una pagina di conferma della richiesta di appunti, contenente i form di sottomissione
- Una pagina per il caricamento di appunti, contenente i form di sottomissione
- Una pagina di scrittura delle recensioni ad appunti (per gli studenti che abbiano ricevuto i corrispondenti file di appunti).
- Una pagina per la gestione (accettazione o rifiuto) delle richieste di appunti (per l’amministratore del sistema).

L'applicazione deve permettere di:

- Visualizzare il catalogo degli appunti disponibili
- Registrare l'utente al servizio
- Accedere alla pagina di login
- Autenticare al servizio come studente registrato
- Autenticare al servizio come amministratore di sistema
- Visualizzare, per ogni file di appunti materia, argomento e recensioni
- Acquistare/richiedere appunti/dispense
- Scrivere recensioni a file di appunti che abbiano effettivamente scaricato/acquistato
- Essere avvisato automaticamente dal sistema per e-mail quando viene accettata la richiesta di caricamento appunti
- Essere avvisato automaticamente dal sistema per e-mail quando viene inserita una recensione su un proprio appunto
- Essere avvisato automaticamente dal sistema per e-mail quando avviene la registrazione dell'utente interessato ad iscriversi al servizio
- Ricevere per e-mail il link dell'appunto acquistato (se l'acquisto è andato a buon fine)
- Accettare o rifiutare le richieste di appunti
- Inserire un nuovo appunto
- Inserire una nuova materia/docente

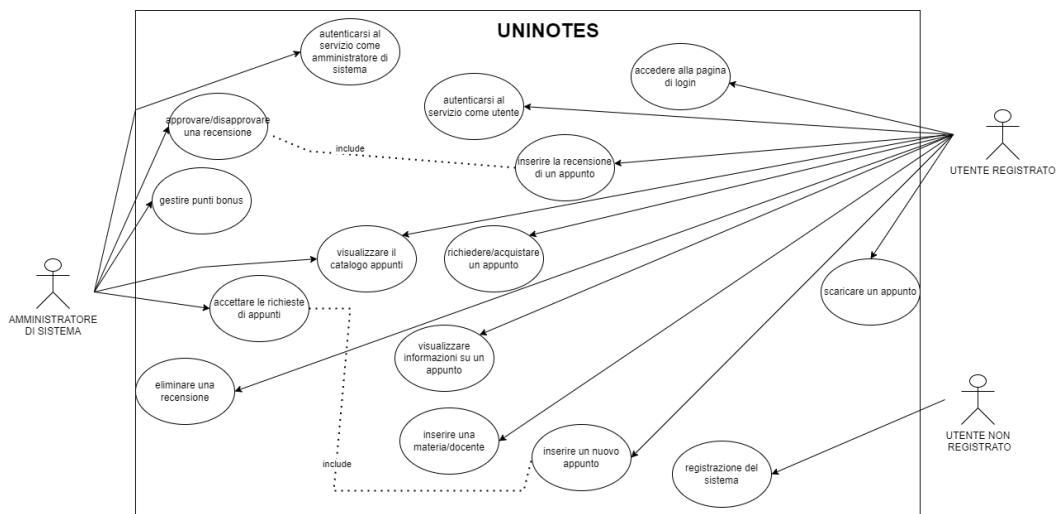
* ***Obiettivi e casi d'uso***

Analizzando i requisiti riportati nel paragrafo precedente, sono stati individuati gli attori principali a cui è destinato il sistema e gli obiettivi che si intende portare a termine; da queste informazioni sono stati ricavati i casi d'uso principali.

Attore	Obiettivo	Caso d'uso
<i>Utente registrato</i>	Gestire la visualizzazione degli appunti offerti suddivisi per materia, mostrando come informazioni sull'appunto: il nome della materia, il titolo dell'argomento e le recensioni relative.	UC1. Visualizzare il catalogo degli appunti
<i>Utente non registrato</i>	Gestire la registrazione al sistema da parte di utenti non registrati.	UC2. Registrazione nel Sistema
<i>Utente registrato</i>	Permette solo agli utenti registrati di accedere alla area personale.	UC3. Autenticarsi al servizio come utente
<i>Utente registrato</i>	Permette all'utente di visualizzare le informazioni relative ad un appunto.	UC4. Visualizzare informazioni su un appunto
<i>Utente registrato</i>	Permette all'utente di scambiare punti bonus/pagare per ottenere un appunto.	UC5. Richiedere/acquistare un appunto
<i>Utente registrato</i>	Permette all'utente di recensire un appunto che ha già scaricato.	UC6. Inserire la recensione di un appunto

<i>Utente registrato</i>	Permette all'utente registrato di caricare un file, inviandolo all'amministratore di sistema.	UC7. Inserire un nuovo appunto
<i>Amministratore di sistema</i>	Permettere all'amministratore di accedere alla propria area personale.	UC8. Autenticarsi al servizio come amministratore di sistema
<i>Amministratore di sistema</i>	Permette all'amministratore di accettare gli appunti, evitando il caricamento di file vuoti o inadeguati.	UC9. Accettare le richieste di appunti
<i>Utente registrato</i>	Permette all'utente di eliminare una recensione precedentemente inserita dallo stesso.	UC10. Eliminare una recensione
<i>Amministratore di sistema</i>	Permette all'amministratore di accettare le recensioni, evitando il caricamento contenuti inadeguati.	UC11. Approvare/Disapprovare una recensione
<i>Amministratore di sistema</i>	Permette all'utente di inserire una materia/docente, non ancora esistente.	UC12. Gestire una materia/docente

I casi d'uso appena descritti si traducono graficamente nel seguente diagramma UML:



* Modello dei casi d'uso

Tra tutti i casi d'uso individuati, si è scelto di fornire una descrizione in formato dettagliato per i seguenti casi d'uso:

- UC3: Autenticarsi al servizio come utente
- UC5: Acquistare un appunto
- UC9. Accettare le richieste di appunti

Per i restanti casi d'uso si fornisce una descrizione in formato breve.

UC3: Autenticarsi al servizio come utente

NOME DEL CASO D'USO	UC3. Autenticarsi al servizio come utente
Portata	Applicazione UniNotes
Livello	Obiettivo utente
Attore primario	Utente registrato

Parti interessate e interessi	Utente registrato: vuole accedere alla sua area personale.
Precondizioni	L'utente deve essere registrato.
Garanzia di successo	La pagina di login è funzionante e l'utente inserisce correttamente i dati.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'utente vuole accedere alla pagina personale 2. L'utente seleziona dal menù la voce "login" 3. Il sistema fornisce un form per il login 4. L'utente compila il form inserendo username e password 5. Il sistema acquisisce i dati e ne verifica la correttezza 6. L'utente viene reindirizzato alla sua area personale
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa</p> <ol style="list-style-type: none"> 1. Viene visualizzato un messaggio di errore 2. Viene chiesto all'utente il riavvio dell'applicazione <p>5a. Username e password errati:</p> <ol style="list-style-type: none"> 1. Il sistema visualizza un messaggio di errore 2. L'utente viene reindirizzato al form 3. L'utente ripete il passo 4 finché i dati non sono validi
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di volte che in cui un utente vuole effettuare un accesso
Varie	

UC5: Acquistare un appunto

NOME DEL CASO D'USO	UC5. Acquistare un appunto
Portata	Applicazione UniNotes
Livello	Obiettivo utente
Attore primario	Utente registrato
Parti interessate e interessi	Utente: vuole acquistare un appunto.
Precondizioni	Per ottenere l'appunto con i punti bonus, occorre aver effettuato l'accesso.
Garanzia di successo	L'acquisto viene effettuato con successo e il link relativo all'appunto acquistato è esistente e funzionante.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'utente vuole acquistare un appunto 2. L'utente sceglie l'appunto di suo interesse, tra quelli disponibili in catalogo 3. L'utente digita il codice univoco dell'appunto e preme "Invio" 4. L'utente riscatta l'appunto con i punti bonus 5. L'utente completa l'acquisto 6. L'utente viene reindirizzato alla pagina contenente il link per il download 7. L'utente clicca sul link per il download 8. L'utente ottiene il documento richiesto
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa</p> <ol style="list-style-type: none"> 1. Viene visualizzato un messaggio di errore 2. Viene chiesto all'utente il riavvio dell'applicazione <p>4a. Se l'utente non possiede abbastanza punti bonus, effettuerà il pagamento con la carta di credito registrata nell'account.</p>
Requisiti speciali	

Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di acquisti che gli utenti intendono effettuare
Varie	

UC9. Accettare le richieste di appunti

NOME DEL CASO D'USO	UC9. Accettare le richieste di appunti
Portata	Applicazione UniNotes
Livello	Obiettivo amministratore
Attore primario	Amministratore di sistema
Parti interessate e interessi	Amministratore: vuole approvare/non approvare l'inserimento di un nuovo appunto
Precondizioni	L'amministratore deve essere autenticato al sistema
Garanzia di successo	L'autenticazione dell'amministratore avviene con successo e l'appunto presenta i requisiti corretti per l'accettazione
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema 2. L'amministratore accede alla sezione "richieste approvazione" 3. Il sistema visualizza le richieste di approvazione 4. L'amministratore visualizza un appunto da approvare 5. L'amministratore approva l'appunto premendo su "approva richiesta" 6. Il sistema acquisisce l'approvazione 7. Il sistema registra il link relativo all'appunto approvato 8. Il sistema invia per email il messaggio "richiesta accettata/rifiutata (Titolo appunto)" all'utente proprietario dell'appunto approvato/rifiutato

	9. L'amministratore viene rimandato al punto 3 fino all'accettazione/rifiuto di tutti gli appunti in sospeso
Estensioni	<ul style="list-style-type: none"> *a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa <ul style="list-style-type: none"> 1. Viene visualizzato un messaggio di errore 2. Viene chiesto all'utente il riavvio dell'applicazione 3a. Non sono presenti richieste da approvare <ul style="list-style-type: none"> • Il sistema visualizza un messaggio: “nessuna richiesta da approvare”
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di approvazioni che l'amministratore deve effettuare
Varie	

UC1: Visualizzare il catalogo degli appunti

Scenario di successo

1. L'utente seleziona dalla homepage la voce “Catalogo appunti”
2. Il sistema mostra tutti gli appunti disponibili suddivisi per materia

UC2: Registrazione nel sistema

Scenario di successo

1. Seleziona dall'homepage la voce “Registrazione”
2. Il sistema visualizza un form per la registrazione
3. L'utente compila il form (e-mail, nome, cognome, nickname, password, carta di credito, scadenza carta di credito) in tutti i campi
4. L'utente clicca “Conferma”
5. Il sistema acquisisce i dati
6. Il sistema mostra il messaggio “Registrazione effettuata con successo”
7. L'utente riceve per email le credenziali dell'account dopo l'avvenuta registrazione

8. L'utente viene reindirizzato al menù utente

UC4: Visualizzare informazioni su un appunto

Scenario di successo

1. L'utente seleziona dalla homepage la voce "Catalogo appunti"
2. Il sistema mostra tutti gli appunti disponibili suddivisi per materia
3. L'utente digita il codice dell'appunto di suo interesse
4. L'utente visualizza la scheda descrittiva dell'appunto
5. L'utente clicca "Visualizza caratteristiche"
6. Il sistema mostra quanto richiesto

UC6. Inserire la recensione di un appunto

Scenario di successo

1. L'utente sceglie nella propria area personale la voce "recensisci appunti"
2. Il sistema mostra tutti gli appunti acquistati
3. L'utente sceglie l'appunto da recensire tra quelli acquistati
4. L'utente valuta l'appunto inserendo un numero di stelle da 1 a 5 (1 - pessimo, 5 - eccellente)
5. Il sistema invia una mail al proprietario dell'appunto con la sua nuova valutazione
6. Il sistema registra la recensione

UC7. Inserire un nuovo appunto

Scenario di successo

1. L'utente accede al sistema
2. L'utente clicca su "inserisci nuovo appunto"
3. Il sistema fornisce un form per il caricamento del nuovo appunto
4. L'utente compila il form inserendo il titolo dell'appunto, la materia relativa all'appunto, il docente di riferimento, la facoltà dell'appunto e il link contenente lo stesso
5. Il sistema acquisisce i dati relativi all'appunto
6. Il sistema invia la richiesta di approvazione alla sezione "richieste approvazione" nella area personale dell'amministratore
7. Il sistema genera il messaggio "richiesta inviata - in attesa di approvazione" nell'area personale dell'utente che ha scritto l'appunto

UC8. Autenticarsi al servizio come amministratore di sistema

Scenario di successo

1. L'amministratore vuole accedere alla sua area personale

2. L'amministratore seleziona dal menù la voce “admin login”
3. Il sistema fornisce un form per il login
4. L'amministratore compila il form inserendo password e email (univoco)
5. Il sistema acquisisce i dati e ne verifica la correttezza
6. L'amministratore viene reindirizzato alla sua “area personale”

UC10: Eliminare una recensione

Scenario di successo

1. L'utente accede alla sua Area personale
2. L'utente accede alla sezione dedicata alla visualizzazione degli appunti acquistati
3. Il sistema mostra tutti gli appunti recensiti
4. L'utente seleziona la recensione da eliminare
5. Il sistema acquisisce l'eliminazione
6. Il sistema invia all'utente il messaggio "recensione eliminata con successo"
7. Il sistema aggiorna le recensioni disponibili per quel determinato appunto

UC11: Approvare/Disapprovare una recensione

Scenario di successo

1. L'amministratore accede alla sua area personale
2. L'amministratore accede alla sezione dedicata alla visualizzazione delle recensioni in sospeso
3. Il sistema visualizza tutte le recensioni in sospeso
4. L'amministratore clicca sul tasto di approvazione di una recensione
5. Il sistema acquisisce la scelta effettuata
6. Il sistema invia all'utente il messaggio "recensione aggiunta con successo"
7. Il sistema aggiorna le recensioni disponibili per quel determinato appunto
8. Il sistema rimanda al punto 3

4a. L'amministratore sceglie di non approvare la recensione (la recensione non rispetta determinati requisiti):

1. Il sistema elimina la recensione
2. Il sistema genera il messaggio “recensione rifiutata - (Titolo appunto)” nell’area personale dell’utente autore della recensione
3. L'amministratore viene rimandato al punto 3

UC12: Gestire le materie/docenti

Scenario di successo

1. L'amministratore accede alla sua area personale
2. L'amministratore clicca “inserire una nuova materia/docente”
3. Il sistema visualizza un modulo per inserire tutte le informazioni relative alla materia o al docente
4. L'amministratore inserisce i dati
5. Il sistema acquisisce i dati inseriti
6. Il sistema visualizza un messaggio di corretto inserimento dei dati
7. Il sistema aggiorna il catalogo degli appunti

2a. L'amministratore vuole modificare una materia/docente

1. L'amministratore clicca per la modifica di una nuova materia/docente
2. Il sistema visualizza un elenco di tutte le materie/docenti disponibili
3. Il sistema visualizza un bottone per la modifica
4. Il sistema visualizza tutte le informazioni relative alla materia/docente permettendo la modifica
5. L'amministratore aggiorna i campi di interesse e conferma.
6. Il sistema acquisisce i dati inseriti
7. Il sistema visualizza un messaggio di corretto inserimento dei dati

2b. L'amministratore vuole eliminare una materia/docente

1. L'amministratore clicca per l'eliminazione di una nuova materia/docente
2. Il sistema visualizza un elenco di tutte le materie/docenti disponibili
3. Il sistema visualizza un bottone per l'eliminazione
4. L'amministratore conferma
5. Il sistema elimina la materia/docente dal sistema

2. Iterazione 1 – UniNotes

* *Introduzione*

Dopo la fase di ideazione, si procede con l'elaborazione e le varie iterazioni.

Con le varie iterazioni si intende raffinare il modello di Visione, implementare l'architettura del software proposto in maniera iterativa, risolvere le problematiche legate ai rischi maggiori, definire in maniera più chiara i requisiti e la portata, fornendo inoltre stime più realistiche del piano di lavoro e delle risorse necessarie.

Nella prima iterazione si è scelto di concentrarsi sul seguente caso d'uso:

- UC7: Permettere all'utente registrato di caricare un file.

Per permettere la realizzazione del caso d'uso sopra indicato, è stato sviluppato anche il caso d'uso UC3 (login Utente), necessario per l'inserimento di un nuovo appunto.

Per questa prima iterazione, assumiamo che tutti gli appunti vengano accettati. Dato che per questa prima fase non viene considerato il punteggio dell'utente, per ogni appunto caricato viene assegnato un numero standard di punti (5 punti: il costo di un appunto).

* *Aggiornamento del caso d'uso UC7*

In merito al caso d'uso preso in esame, UC7, sono state apportate alcune modifiche a fronte di alcuni passaggi risultati poco chiari emersi durante gli incontri preliminari di revisione. Per rendere i casi d'uso più comprensibili e semplici dal punto di vista dell'implementazione si è proceduto come segue:

UC7: Inserisci nuovo appunto

NOME DEL CASO D'USO	UC7: Inserisci nuovo appunto
Portata	Applicazione UniNotes

Livello	Obiettivo utente
Attore primario	Utente registrato
Parti interessate e interessi	Utente: vuole inserire un nuovo appunto.
Precondizioni	L'utente deve aver effettuato il login.
Garanzia di successo	Le informazioni relative all'appunto sono state inserite con successo.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'utente vuole inserire un appunto 2. L'utente sceglie l'attività “Inserisci nuovo appunto” 3. L'utente inserisce il link drive dell'appunto che vuole inserire e il titolo dell'argomento dell'appunto. Il Sistema registra le informazioni inserite. 4. Il Sistema mostra tutte le facoltà presenti nello stesso. 5. L'utente seleziona tramite il <i>codice univoco facoltà</i>, la facoltà di riferimento per l'appunto. 6. L'utente inserisce il nome della materia e il nome del docente di riferimento. Il Sistema registra le informazioni inserite. 7. L'utente visualizza un messaggio che indica il corretto inserimento dei dati dell'appunto.
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa:</p> <ol style="list-style-type: none"> 1. Viene visualizzato un messaggio di errore 2. Viene chiesto all'utente il riavvio dell'applicazione <p>5a. Se l'utente inserisce un codice non esistente:</p> <ul style="list-style-type: none"> - Il Sistema genera un messaggio di errore - L'utente ripete il passaggio 5 inserendo un codice differente.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di appunti che l'utente vuole caricare.
Varie	

* *Analisi orientata agli oggetti*

L'analisi orientata agli oggetti verte sulla generazione di una descrizione del dominio da un punto di vista ad oggetti. A questo proposito, vengono presentati vari strumenti:

- Modello di dominio

- Sequence System Diagram (SSD)
- Contratti delle operazioni

Modello di Dominio

La modellazione del business è una disciplina che, in termini di UP (processo unificato), si occupa di fornire dettagli sul dominio. Nello specifico, essa è costituita da:

- Modello di dominio
- Elaborato grafico dei concetti
- Attributi e associazioni significative

Relativamente al caso d'uso scelto (UC7), dopo un'attenta valutazione dello scenario principale di successo è stato possibile identificare le seguenti classi concettuali:

- ✚ UniNotes
- ✚ Utente Registrato
- ✚ Appunto
- ✚ Facoltà
- ✚ Materia

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:

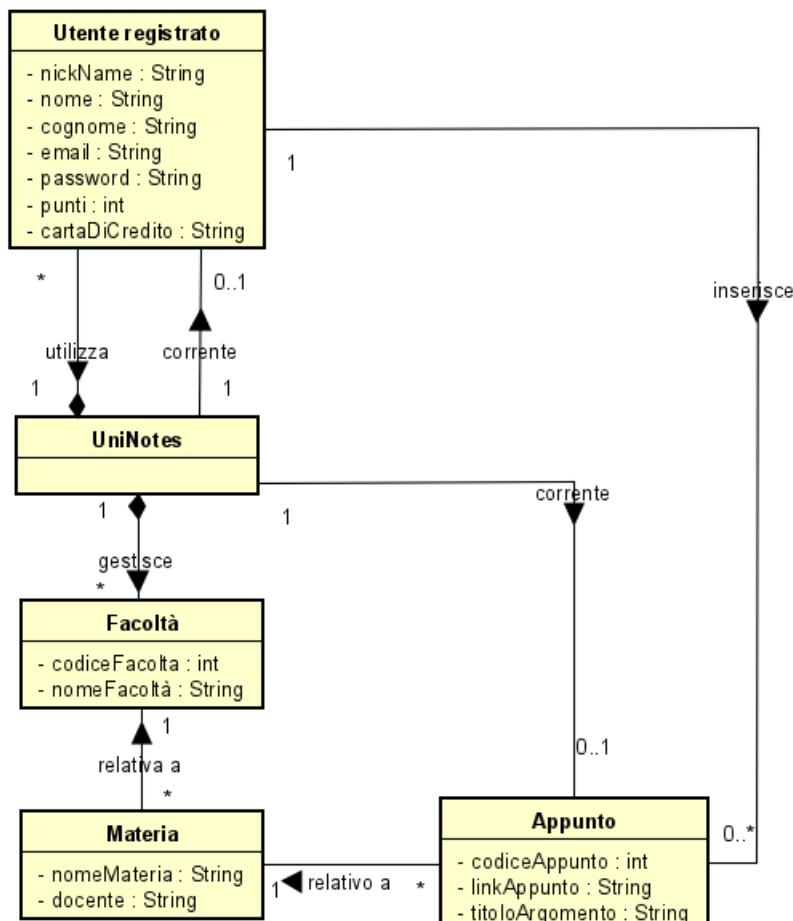
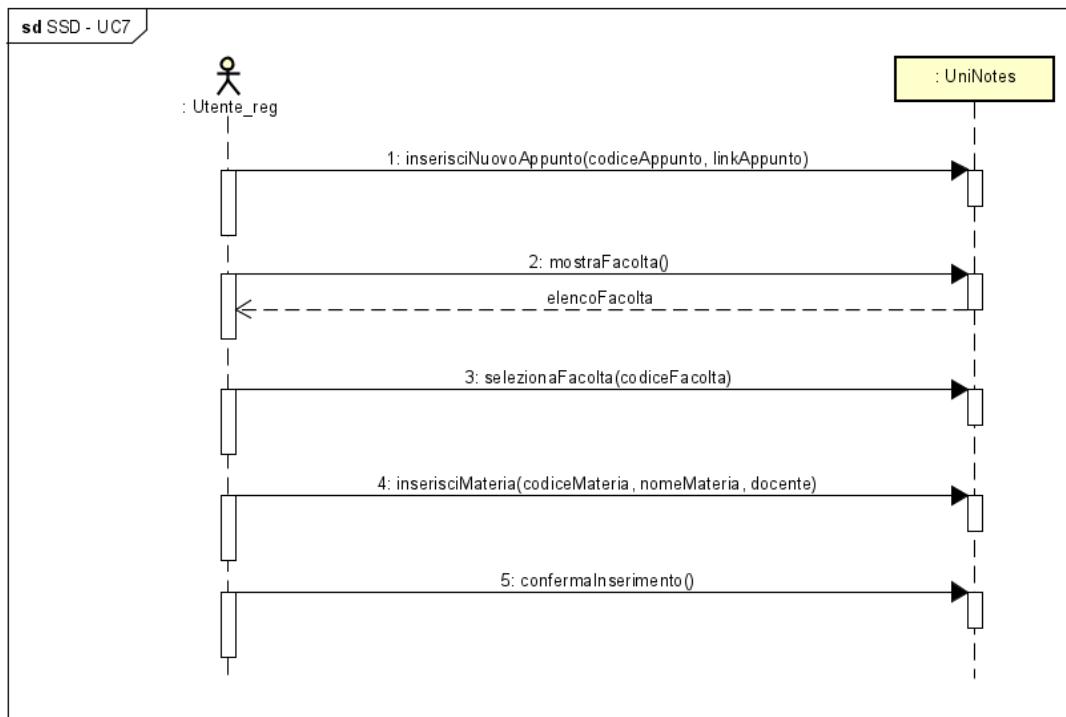
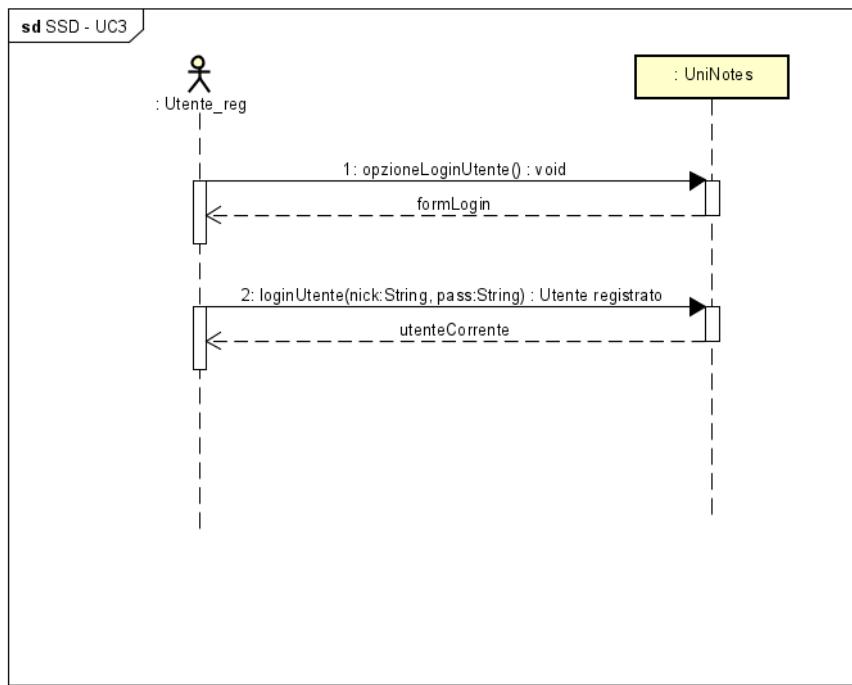


Diagramma di sequenza di sistema

Il passo successivo all'analisi OO è la creazione del diagramma di sequenza (SSD) con lo scopo di mostrare il succedersi degli eventi di input e output per lo scenario principale di successo del caso d'uso UC7. Si avrà dunque:



Per quanto riguarda il caso d'uso UC3, necessario per lo svolgimento di UC7, il diagramma di sequenza di sistema è il seguente:



Contratti delle operazioni

Attraverso i Contratti vengono adesso illustrate le principali operazioni di sistema che si occupano di gestire gli eventi di sistema individuati negli SSD.

CONTRATTO CO1 - Opzione Login Utente

<i>Operazione</i>	OpzioneLoginUtente()
<i>Riferimenti</i>	UC3
<i>Pre-condizioni</i>	L'utente è registrato.
<i>Post-condizioni</i>	-

CONTRATTO CO2 - Login Utente

<i>Operazione</i>	LoginUtente(nickname : String, password : String)
<i>Riferimenti</i>	UC3
<i>Pre-condizioni</i>	L'utente inserisce correttamente le credenziali.
<i>Post-condizioni</i>	L'utente viene reindirizzato al menù utente

CONTRATTO CO3 - Inserisci nuovo appunto

<i>Operazione</i>	InserisciNuovoAppunto(codiceAppunto, linkAppunto)
<i>Riferimenti</i>	UC7

<i>Pre-condizioni</i>	L'utente ha effettuato il login.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata creata un'istanza a di appunto - Gli attributi di a sono stati inizializzati - a è stata associata a UniNotes tramite l'associazione "Corrente"

CONTRATTO CO4 - Mostra Facoltà	
<i>Operazione</i>	mostraFacolta()
<i>Riferimenti</i>	UC7
<i>Pre-condizioni</i>	È in corso l'inserimento dell'appunto a.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - Viene stampata a video la lista delle Facoltà presenti su UniNotes.

CONTRATTO CO5 - Seleziona Facoltà	
<i>Operazione</i>	SelezionaFacolta(codiceFacolta)
<i>Riferimenti</i>	UC7
<i>Pre-condizioni</i>	È in corso l'inserimento dell'appunto a. Il codice della facoltà inserita è esistente.
<i>Post-condizioni</i>	-

CONTRATTO CO6 - Inserisci materia	
<i>Operazione</i>	InserisciMateria(nomeMateria, docente)
<i>Riferimenti</i>	UC7
<i>Pre-condizioni</i>	È in corso l'inserimento dell'appunto a.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata creata un'istanza m di Materia - Gli attributi di m sono stati inizializzati - m è associata ad a tramite l'associazione "relativo a"

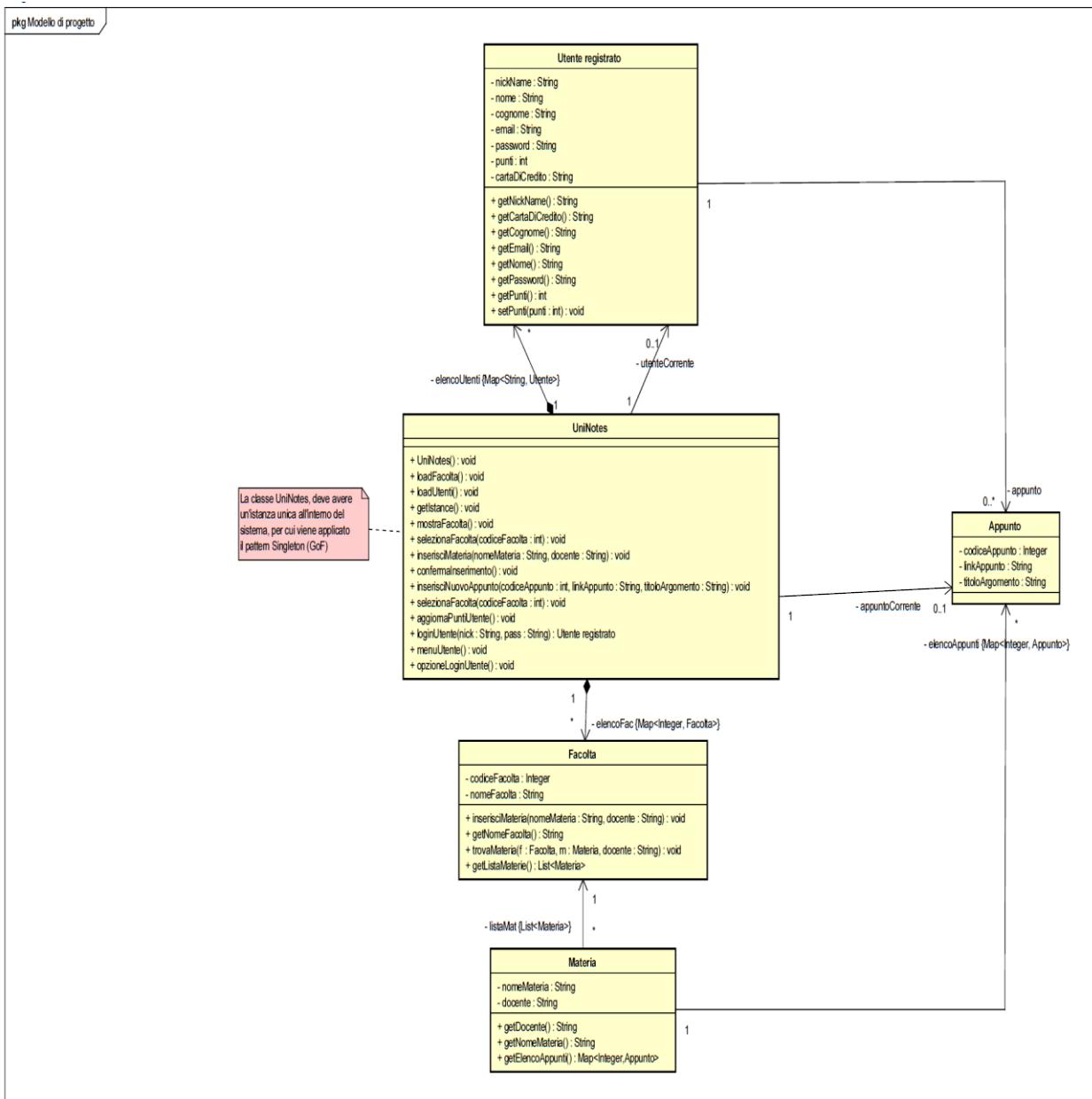
CONTRATTO CO7 - Conferma inserimento	
<i>Operazione</i>	ConfermaInserimento()
<i>Riferimenti</i>	UC7
<i>Pre-condizioni</i>	È in corso l'inserimento dell'appunto a.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata associata l'istanza a di appunto a UniNotes tramite l'associazione "Gestisce" - Setta i punti dell'utente corrente, aggiungendo 5 punti riscatto.

* **Progettazione**

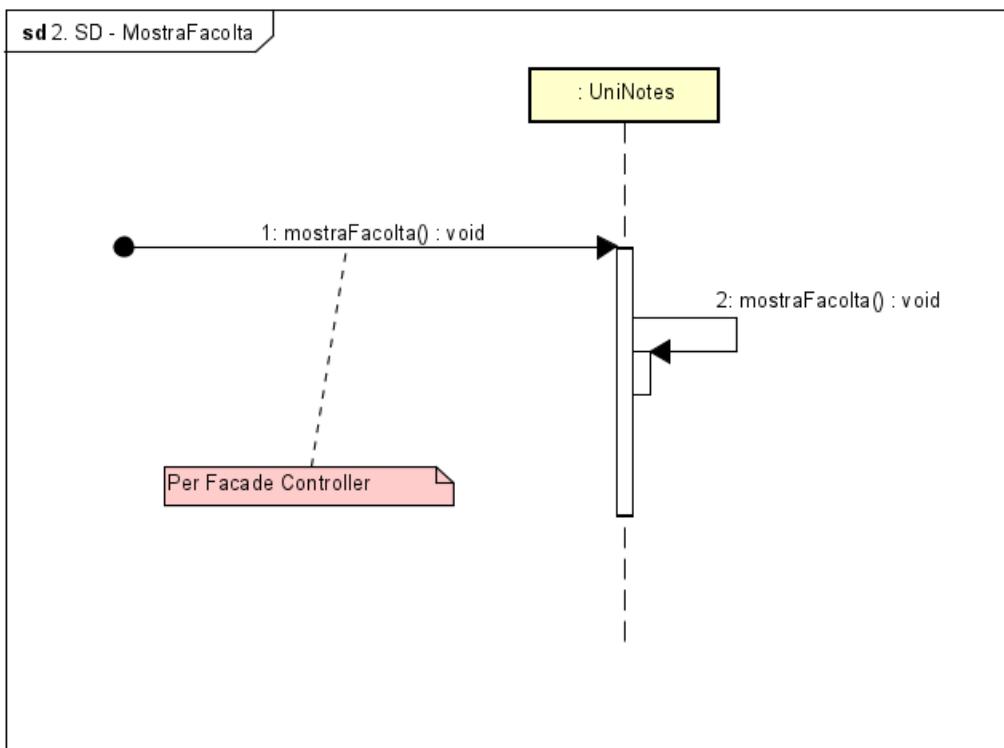
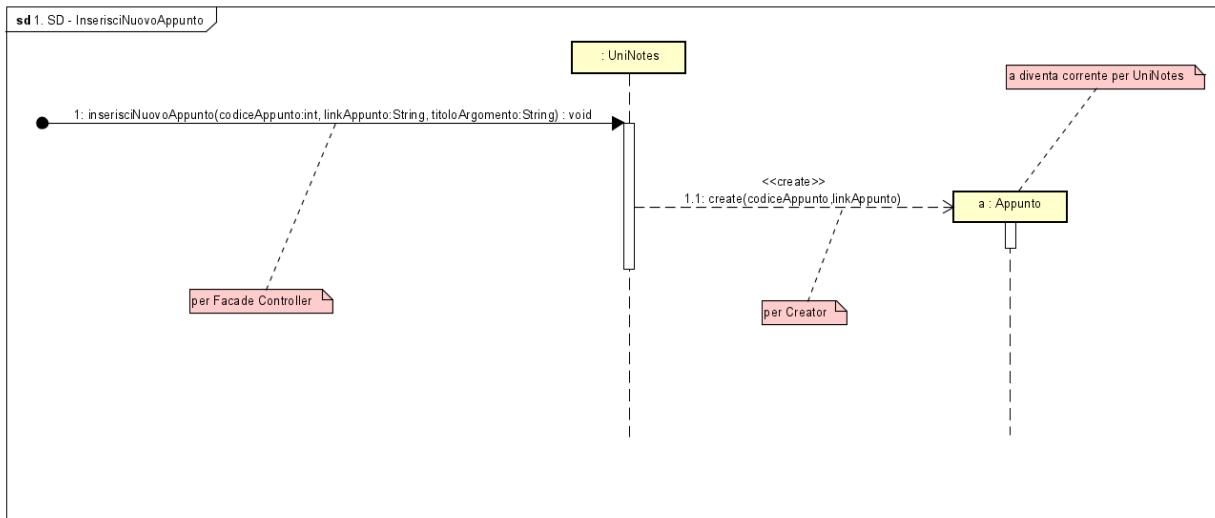
La disciplina di up che si occupa della definizione degli oggetti software, delle loro responsabilità e del modo in cui questi agiscono per soddisfare i requisiti individuati nei punti precedenti, è la progettazione orientata agli oggetti. Il modello di progetto è l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di interazione) che statico (Diagramma delle Classi). A seguire, figurano i diagrammi di iterazione più significativi e il diagramma delle classi relativi al caso d'uso UC7 realizzato dopo uno studio scrupoloso degli elaborati composti in precedenza.

UniNotes

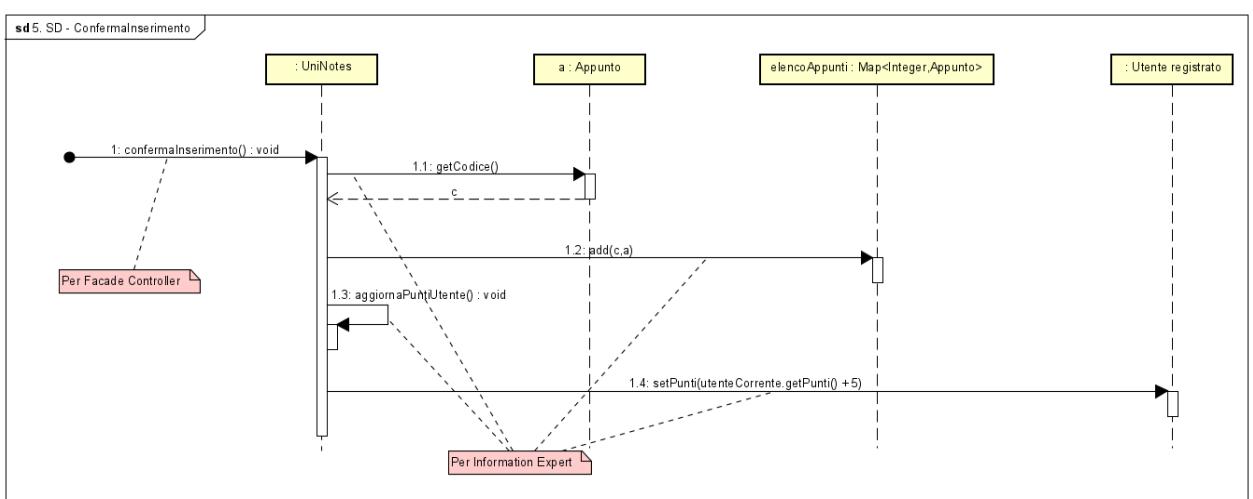
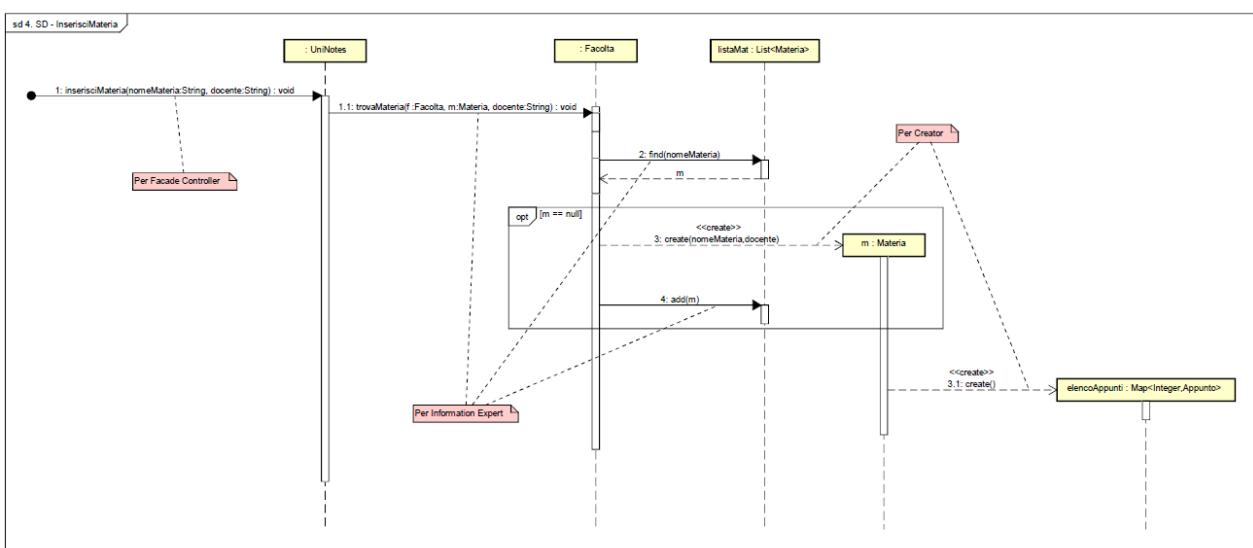
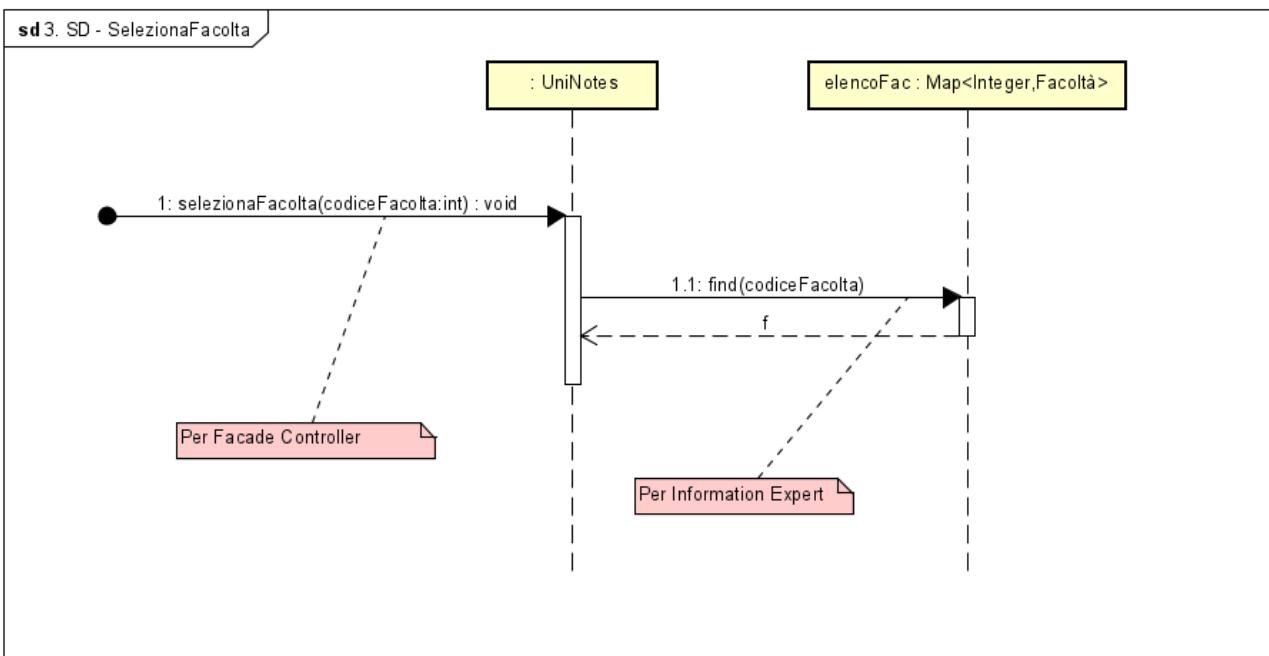
Diagramma delle classi



Diagrammi di Sequenza



UniNotes



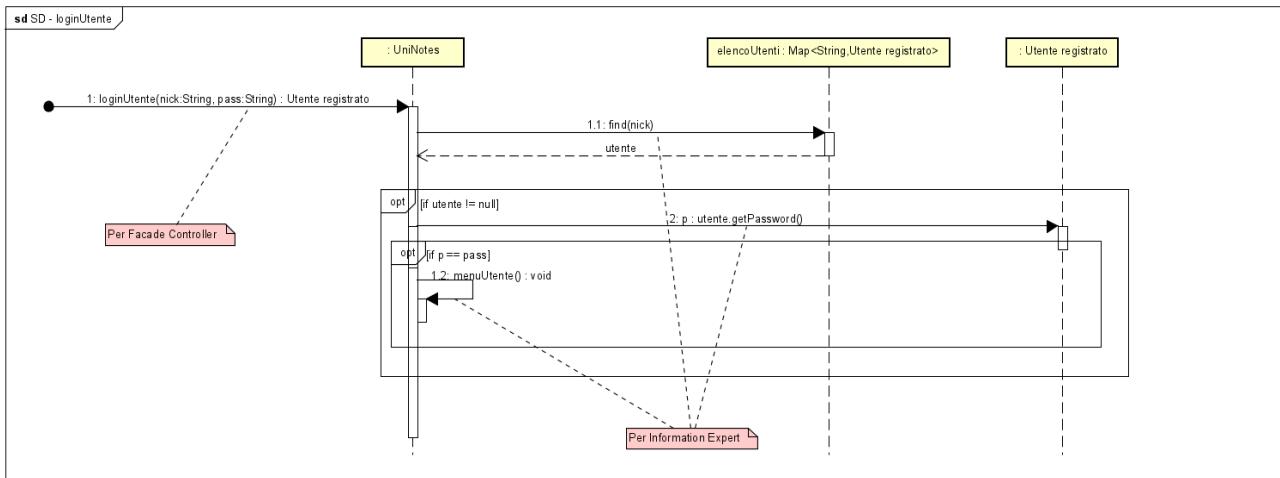
Per quanto riguarda il caso d'uso UC3, il diagramma SD è il seguente:

Dario Sapienza 1000043697

Ivan Scandura 1000043696

Enrica Spataro 1000042244

UniNotes



* Refactoring

In fase di refactoring non sono state apportate modifiche al modello di dominio. È stata tuttavia implementata una miglioria a livello di codice del programma, gestendo la sessione utente tramite un attributo “*UtenteCorrente*”.

3. Iterazione 2 – UniNotes

* Introduzione

Dopo la prima iterazione, si procede con la seconda iterazione. Durante questa seconda iterazione il focus sarà incentrato sullo scenario di successo del caso d'uso UC5:

- Acquistare un appunto: questa iterazione si concentrerà sullo scenario che prevede il riscatto dell'appunto tramite i punti posseduti dall'utente.

È stato deciso di rimuovere la classe catalogo pensata durante l'ideazione poiché la visualizzazione degli appunti è gestita tramite le funzioni Java per le liste/mappe.

* Aggiornamento del caso d'uso UC5

UC5: Acquistare un appunto

NOME DEL CASO D'USO	UC5. Acquistare un appunto
Portata	Applicazione UniNotes
Livello	Obiettivo utente
Attore primario	Utente registrato
Parti interessate e interessi	Utente: vuole riscattare un appunto con i punti.
Precondizioni	L'utente deve aver effettuato l'accesso.
Garanzia di successo	L'acquisto viene effettuato con successo e il link relativo all'appunto acquistato è esistente e funzionante.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'utente vuole acquistare un appunto 2. L'utente seleziona la facoltà di suo interesse. 3. L'utente sceglie l'appunto di suo interesse, tra quelli disponibili in catalogo 4. L'utente digita il codice univoco dell'appunto e preme "Invio" 5. L'utente riscatta l'appunto con i punti bonus 6. L'utente completa l'acquisto 7. L'utente riceve una mail contenente il link per il download 8. L'utente ottiene il documento richiesto
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa</p> <ol style="list-style-type: none"> 3. Viene visualizzato un messaggio di errore 4. Viene chiesto all'utente il riavvio dell'applicazione

	4a. Se l'utente non possiede abbastanza punti bonus, effettuerà il pagamento con la carta di credito registrata nell'account.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di acquisti che gli utenti intendono effettuare
Varie	

* *Analisi orientata agli Oggetti*

In questa iterazione, come nella precedente saranno utilizzati strumenti come: Modello di Dominio, Sequence System Diagram (SSD), e contratti delle operazioni. Verranno analizzati anche i cambiamenti rispetto alla fase precedente.

Modello di Dominio

Analizzando il caso d'uso UC5 emergono nuove classi concettuali, rispetto all'iterazione precedente:

- ✚ Amministratore di sistema: si occupa della conferma degli acquisti e della gestione dell'invio delle email con il link degli appunti acquistati (abbiamo utilizzato il pattern GoF Singleton per l'amministratore)
- ✚ Acquisto: contiene le informazioni relative all'acquisto di un appunto da parte di un utente

Dall'integrazione di queste nuove classi, tenendo conto di associazioni e attributi, è stato ricavato il seguente modello di dominio.

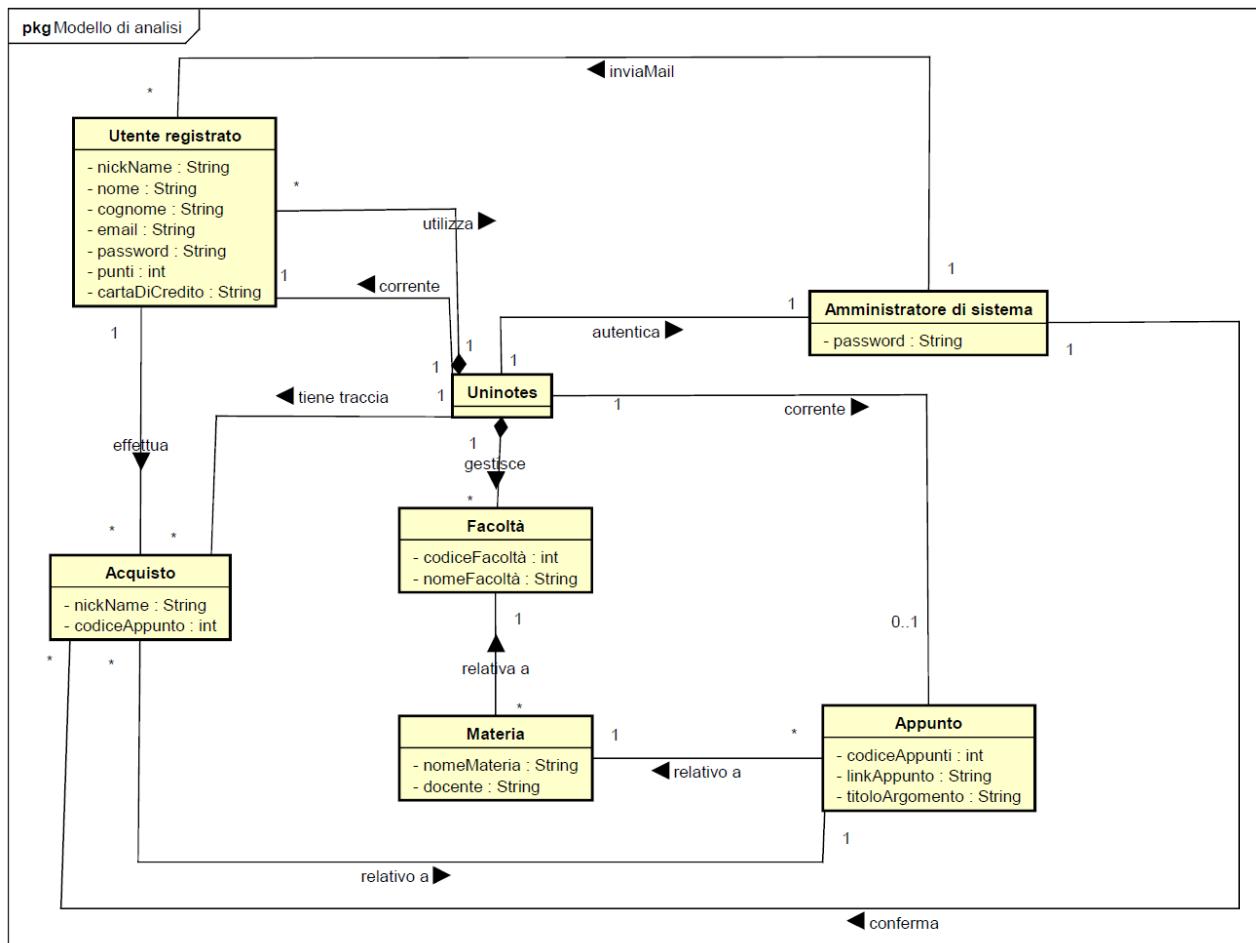
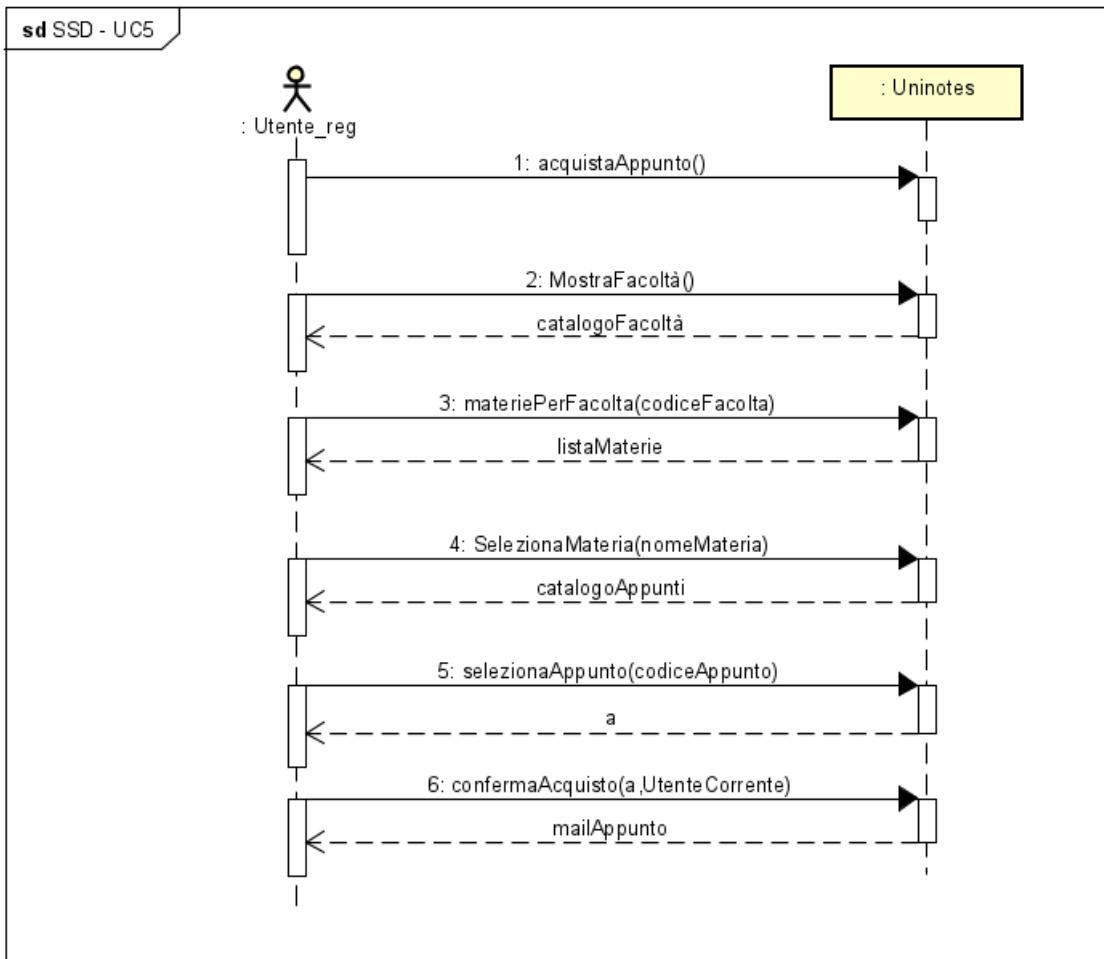


Diagramma di sequenza di sistema

In seguito, è stato creato il diagramma di sequenza di sistema (SSD) per il caso d'uso UC5.



Contratti delle operazioni

Attraverso i Contratti vengono adesso illustrate le principali operazioni di sistema che si occupano di gestire gli eventi di sistema individuati nell'SSD.

CONTRATTO CO1 - Acquista Appunto	
<i>Operazione</i>	acquistaAppunto()
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	L'utente ha effettuato l'accesso
<i>Post-condizioni</i>	-

CONTRATTO CO2 - Mostra Facoltà	
<i>Operazione</i>	mostraFacolta()
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	L'utente ha effettuato l'accesso È in corso l'acquisto di un appunto
<i>Post-condizioni</i>	- Viene stampata a video la lista delle Facoltà presenti su UniNotes.

CONTRATTO CO3 – Materie Per Facoltà

<i>Operazione</i>	MateriePerFacoltà (codiceFacoltà)
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	È in corso l'acquisto di un appunto. Il codice selezionato corrisponde ad una facoltà esistente.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata recuperata l'istanza a di f di Facoltà dall'elenco delle facoltà sulla base del codiceFacoltà - È stata recuperata la lista delle materie relative alla facoltà f (la lista viene ritornata se è diversa da null)

CONTRATTO CO4 – Seleziona Materia

<i>Operazione</i>	SelezioneMateria(nomeMateria, f)
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	È in corso l'acquisto di un appunto. Il nome della materia corrisponde al nome di una delle materie presenti nella facoltà f selezionata.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata recuperata l'istanza m di Materia dalla lista delle materie relative alla facoltà f sulla base del nomeMateria - È stata recuperata la lista degli appunti relative alla materia m.

CONTRATTO CO5 – Seleziona Appunto

<i>Operazione</i>	SelezioneAppunto(codiceAppunto,m)
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	È in corso l'acquisto di un appunto. Il codice dell'appunto corrisponde ad un appunto esistente nella mappa appunti relativo alla materia m
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - È stata recuperata l'istanza a di Appunto dalla mappa degli appunti relativa alla materia m - Viene recuperata l'istanza Utente Corrente tramite "getUtenteCorrente"

	<ul style="list-style-type: none"> - Viene verificato il numero di punti posseduti dall'utente corrente - Viene settato il nuovo valore dei punti dell'utente corrente (viene sottratto il valore di un appunto, cioè 5 punti) - È stata creata un'istanza ac di Acquisto - È stata associata l'istanza ac ad UniNotes tramite l'associazione “tiene traccia”
--	---

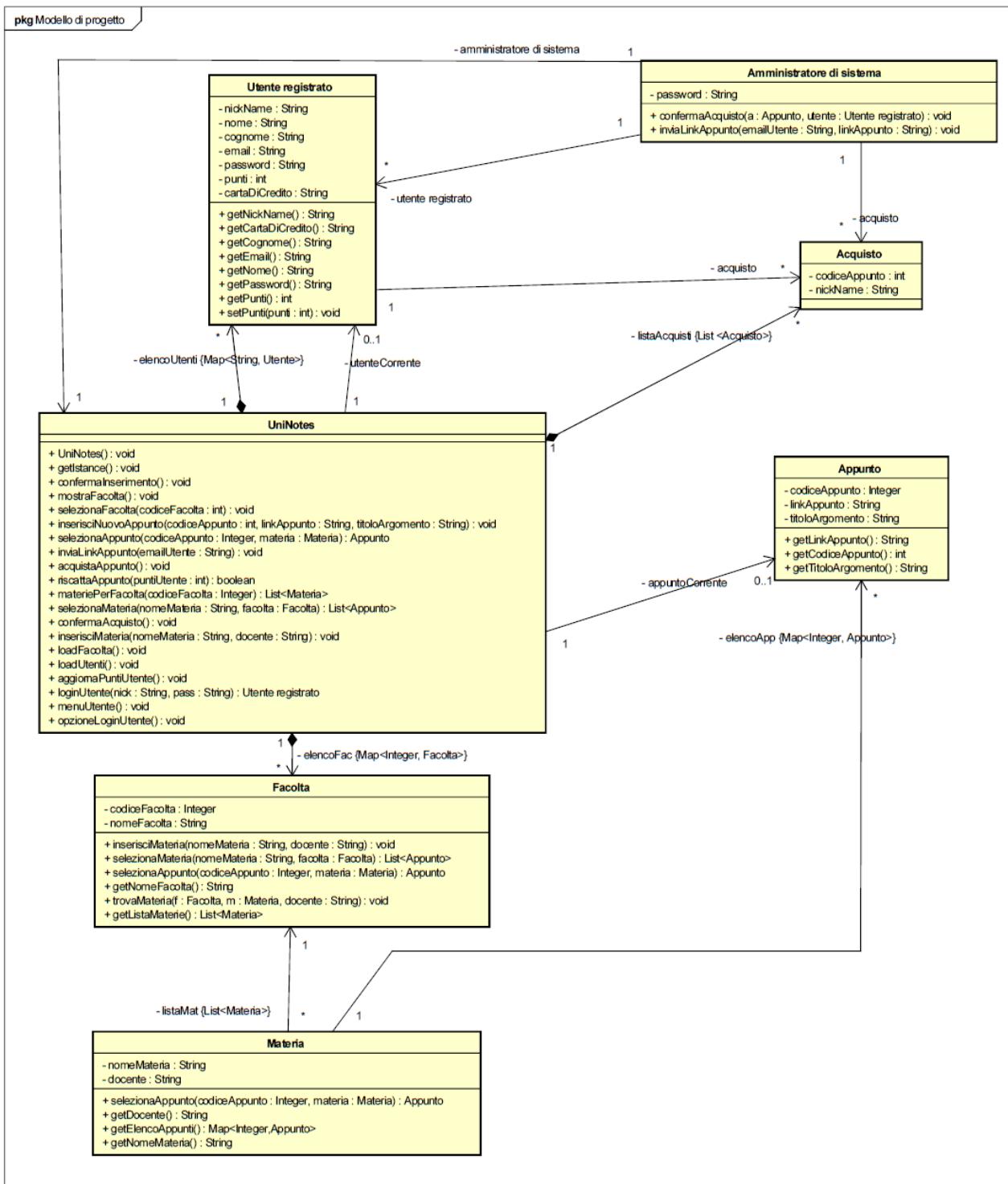
CONTRATTO CO6 – Conferma Acquisto

<i>Operazione</i>	ConfermaAcquisto(a, UtenteCorrente,)
<i>Riferimenti</i>	UC5
<i>Pre-condizioni</i>	È in corso l'acquisto di un appunto
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - Sono stati inizializzati gli attributi dell'istanza ac di Acquisto - L'acquisto ac è stato aggiunto alla lista acquisti - È stata recapitata una email all'utente contenente il link dell'appunto acquistato

* Progettazione

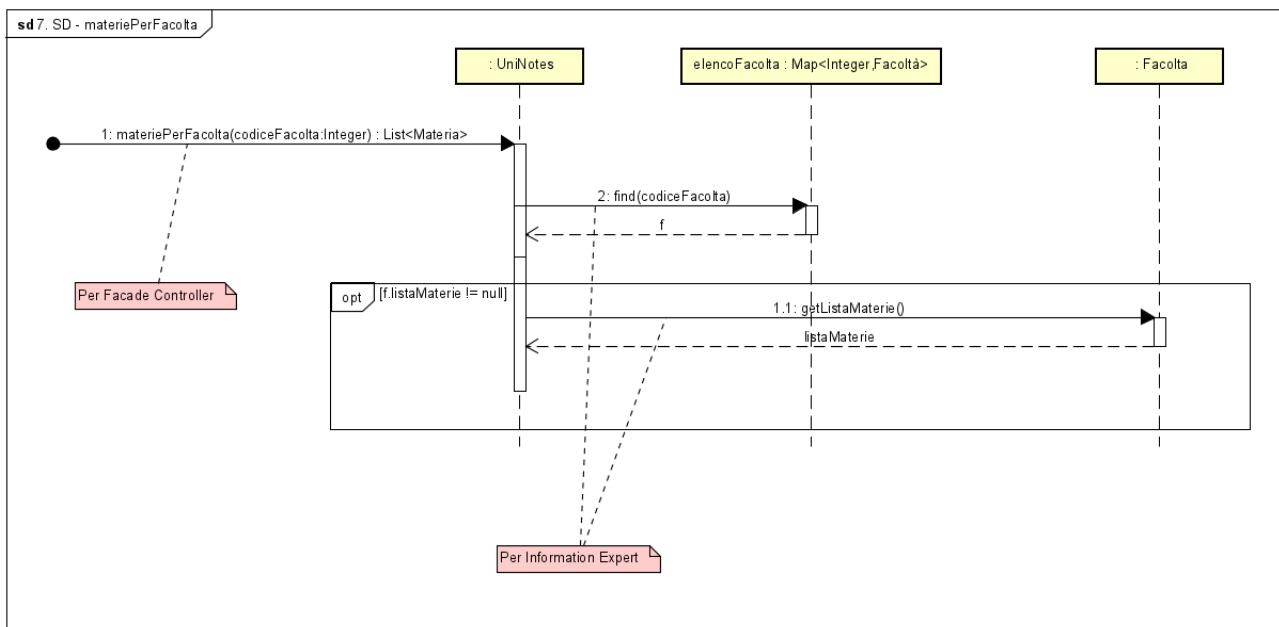
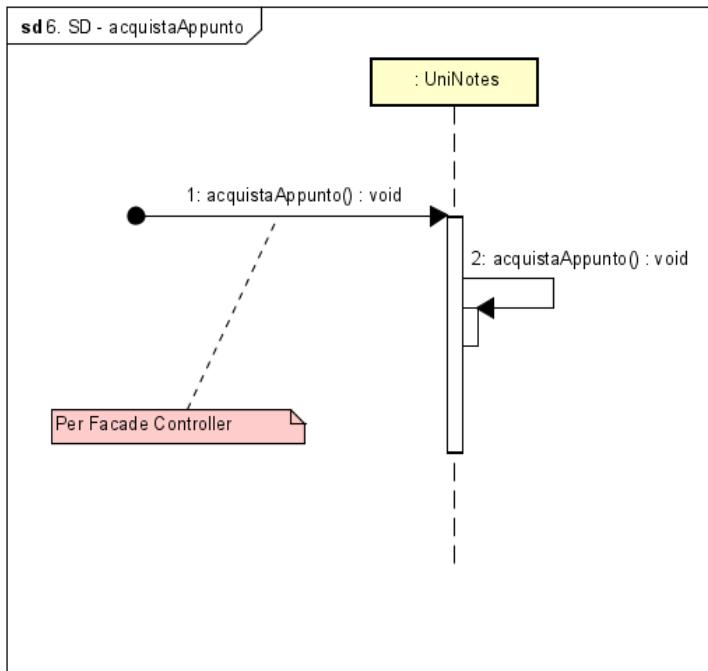
In questa sezione, come nell'iterazione precedente, vengono mostrati tutti i diagrammi che servono a descrivere la progettazione logica da un punto di vista dinamico (diagrammi di interazione) e da un punto di vista statico (diagramma delle classi).

Diagramma delle classi

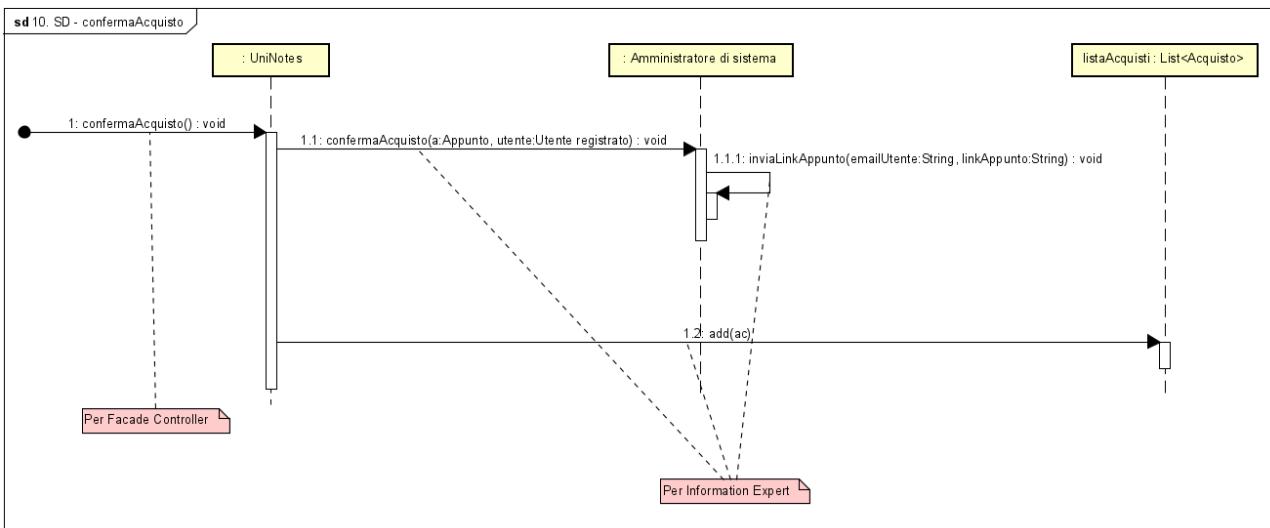
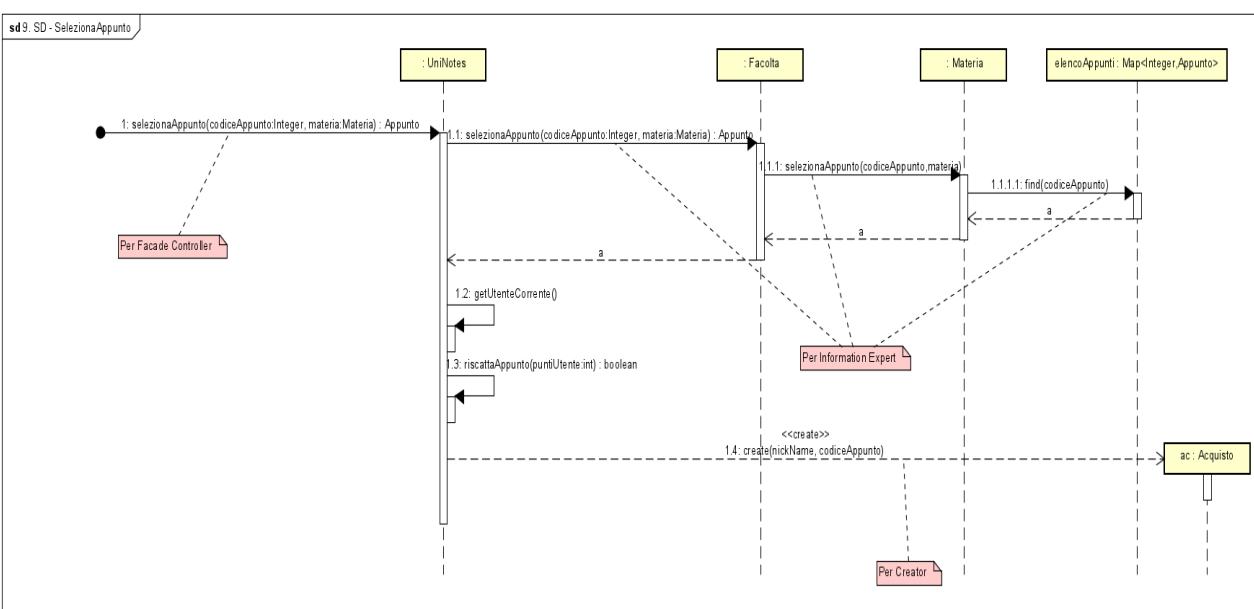
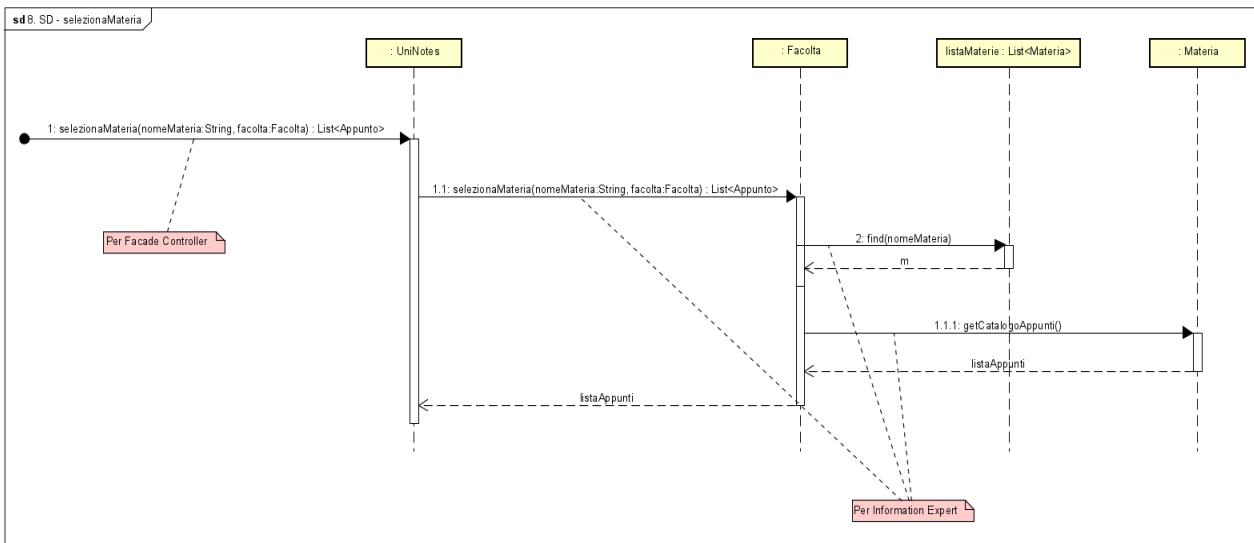


Diagrammi di Sequenza

In questa sezione vengono mostrati gli sd per il caso d'uso UC5. Non vengono mostrati quelli già presentati nell'iterazione 1.



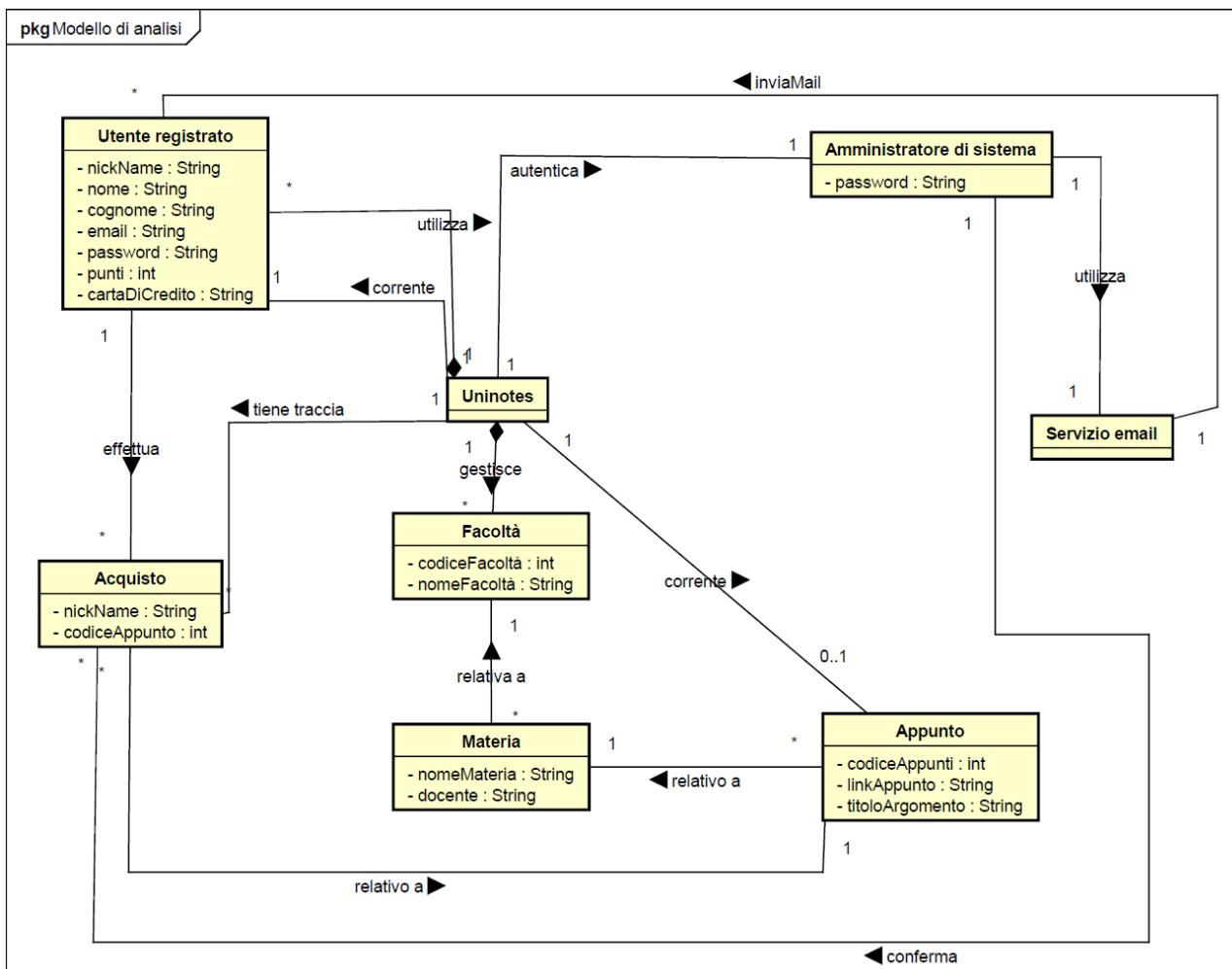
UniNotes



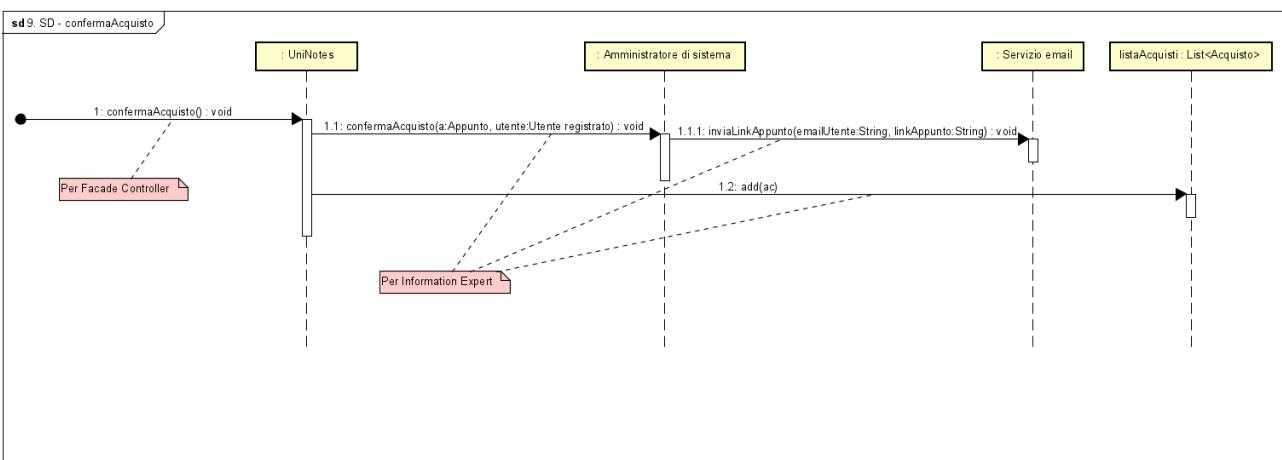
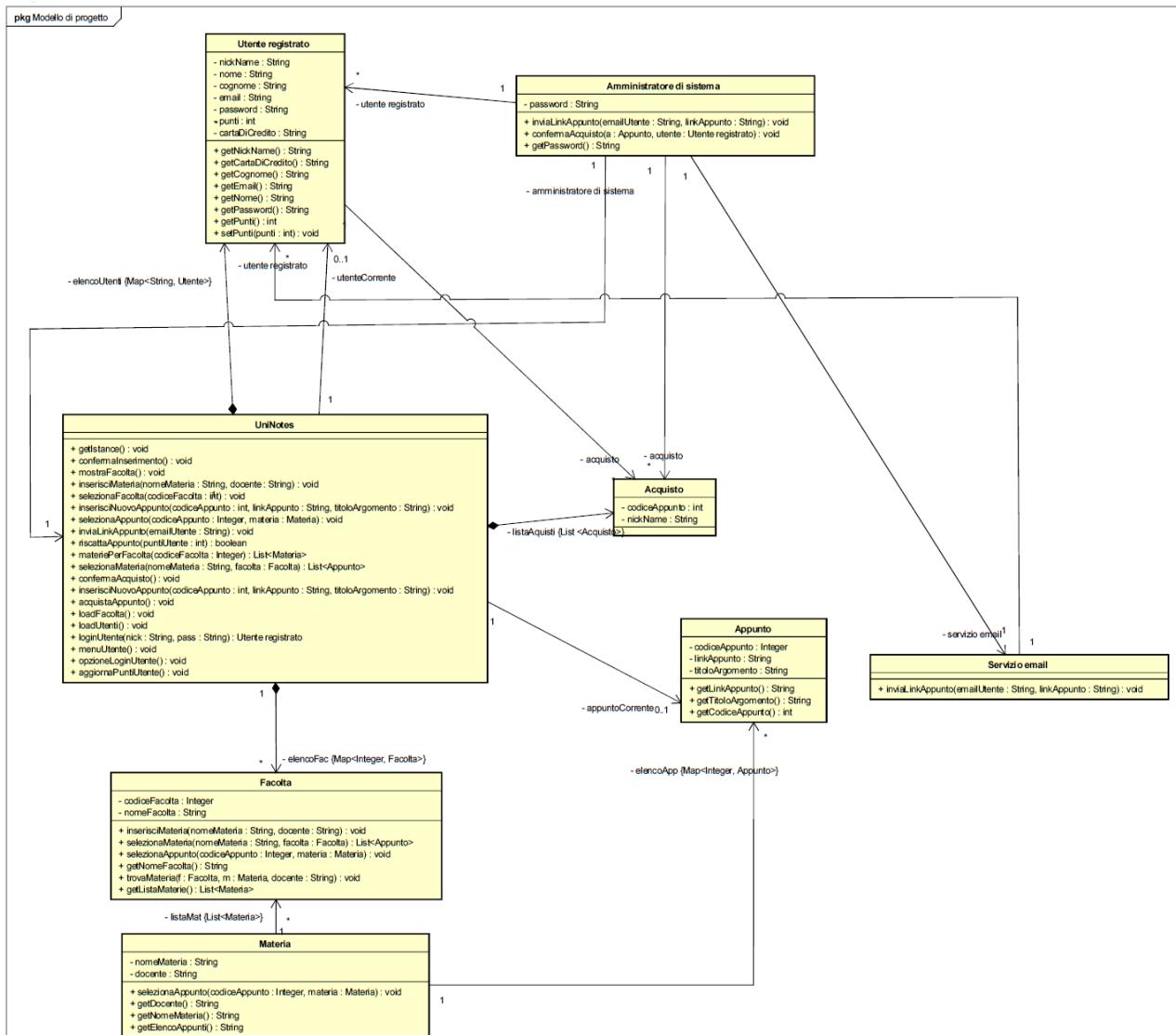
* Refactoring

In fase di refactoring, dopo un'attenta analisi, è stato deciso di apportare alcune modifiche al modello di dominio. È stata introdotta la classe “*ServizioEmail*”, per gestire l’invio del link relativo all’appunto acquistato. La necessità di questa operazione nasce dal voler sollevare l’amministratore di sistema da alcune responsabilità, per non sovraccaricarlo. L’introduzione di questa classe ha introdotto modifiche a:

- Modello di Dominio
- Modello delle classi di progetto
- Diagrammi di Sequenza (SD) : sono indicati con “ * ” nel file astah
- Codice Java



UniNotes



4. Iterazione 3 – UniNotes

* Introduzione

Durante questa terza iterazione, ci si concentrerà sull’analisi e l’implementazione del caso d’uso UC9: Approvare le richieste di appunti:

- ✚ UC9: Permette all’amministratore di accettare gli appunti, evitando il caricamento di file vuoti o inadeguati.

In questa iterazione non viene più assunto che gli appunti vengano tutti approvati, ma tutti gli appunti caricati vengono inseriti in una mappa di appunti in sospeso e solo dopo l’approvazione l’utente potrà riscattare i suoi punti o visualizzare questo appunto nel catalogo.

Per permettere lo svolgimento di questa iterazione, è stato anche analizzato ed implementato il caso d’uso UC8: Autenticarsi al servizio come Amministratore.

In questa iterazione si è analizzato ed implementato anche lo scenario alternativo del caso d’uso UC5, che permetteva il pagamento di un appunto per riscattarlo, nel caso in cui non fosse possibile richiederlo a causa del numero insufficiente di punti riscatto.

Nel caso in cui infatti i punti non siano sufficienti, viene verificata la validità della carta di credito dell’utente (si è ipotizzato che la carta risulti valida se non è scaduta); in caso di successo, l’utente riceverà il link tramite mail, in caso contrario verrà visualizzato un messaggio di errore.

* UC9. Accettare le richieste di appunti

NOME DEL CASO D’USO		UC9. Accettare le richieste di appunti
Portata	Applicazione UniNotes	
Livello	Obiettivo amministratore	
Attore primario	Amministratore di sistema	
Parti interessate e interessi	Amministratore: vuole approvare/non approvare l’inserimento di un nuovo appunto	
Precondizioni	L’amministratore deve essere autenticato al sistema	
Garanzia di successo	L’autenticazione dell’amministratore avviene con successo e l’appunto presenta i requisiti corretti per l’accettazione	
Scenario principale di successo	10. L’amministratore accede al sistema 11. L’amministratore accede alla sezione “richieste approvazione” 12. Il sistema visualizza le richieste di approvazione	

	<p>13. L'amministratore visualizza un appunto da approvare</p> <p>14. L'amministratore approva l'appunto premendo su "approva richiesta"</p> <p>15. Il sistema acquisisce l'approvazione</p> <p>16. Il sistema registra il link relativo all'appunto approvato</p> <p>17. Il sistema invia per email il messaggio "richiesta accettata/rifiutata (Titolo appunto)" all'utente proprietario dell'appunto approvato/rifiutato</p> <p>18. L'amministratore viene rimandato al punto 3 fino all'accettazione/rifiuto di tutti gli appunti in sospeso</p>
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa</p> <p>3. Viene visualizzato un messaggio di errore</p> <p>4. Viene chiesto all'utente il riavvio dell'applicazione</p> <p>3a. Non sono presenti richieste da approvare</p> <ul style="list-style-type: none"> • Il sistema visualizza un messaggio: "nessuna richiesta da approvare"
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di approvazioni che l'amministratore deve effettuare
Varie	

* Analisi orientata agli Oggetti

Al fine di descrivere il dominio da un punto di vista ad oggetti e gestire ulteriori requisiti, saranno utilizzati nuovamente gli stessi strumenti delle iterazioni precedenti. Tuttavia, esaminando il caso d'uso *UC9*, si è deciso di modificare il Modello di Dominio aggiungendo una classe “Carta di Credito” e una relazione “gestisce” tra “Amministratore” e “Appunto”.

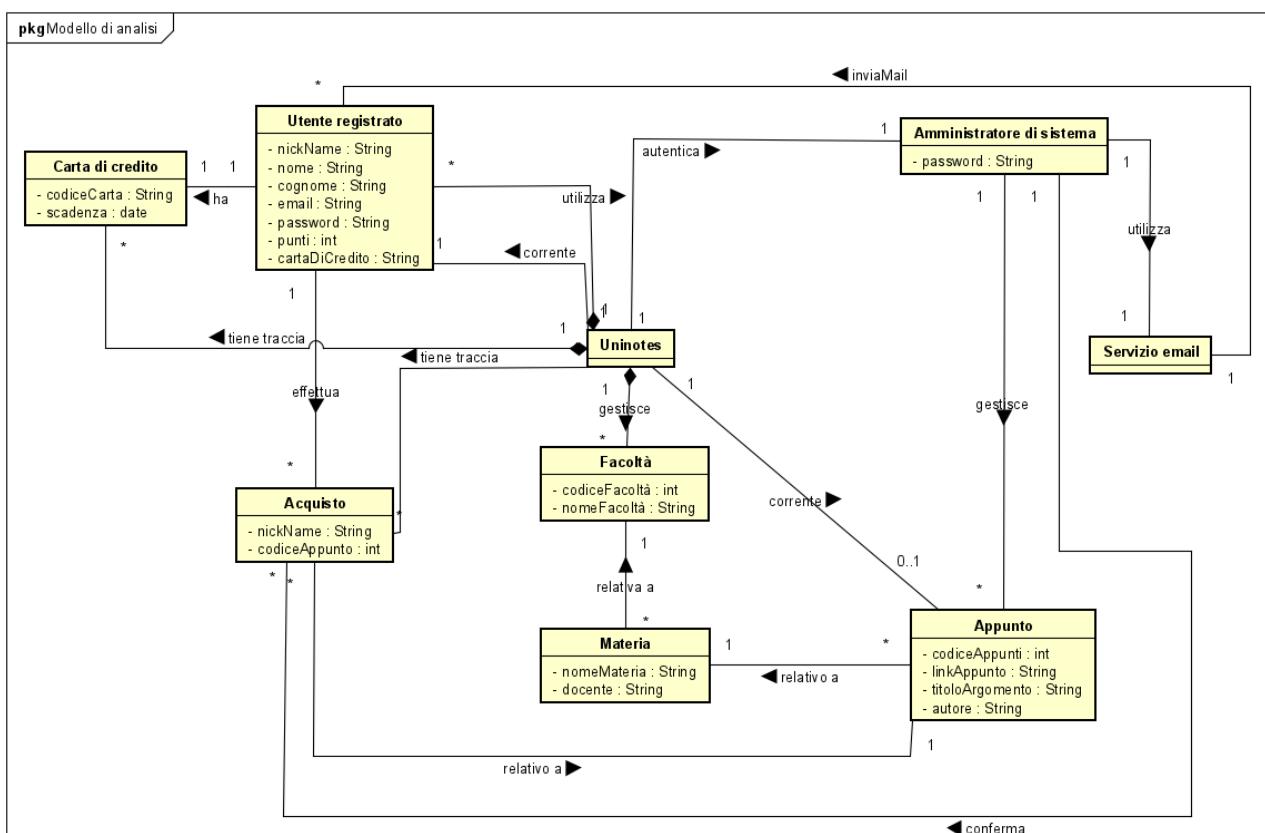
Dato che non esiste più l’ipotesi che ogni appunto viene approvato, ma che l’approvazione viene fatta in un momento successivo all’inserimento dell’appunto, è stato necessario aggiungere un attributo “autore” all’appunto, per poter attribuire a quest’ultimo i punti una volta che il suo appunto venga approvato.

Modello di dominio

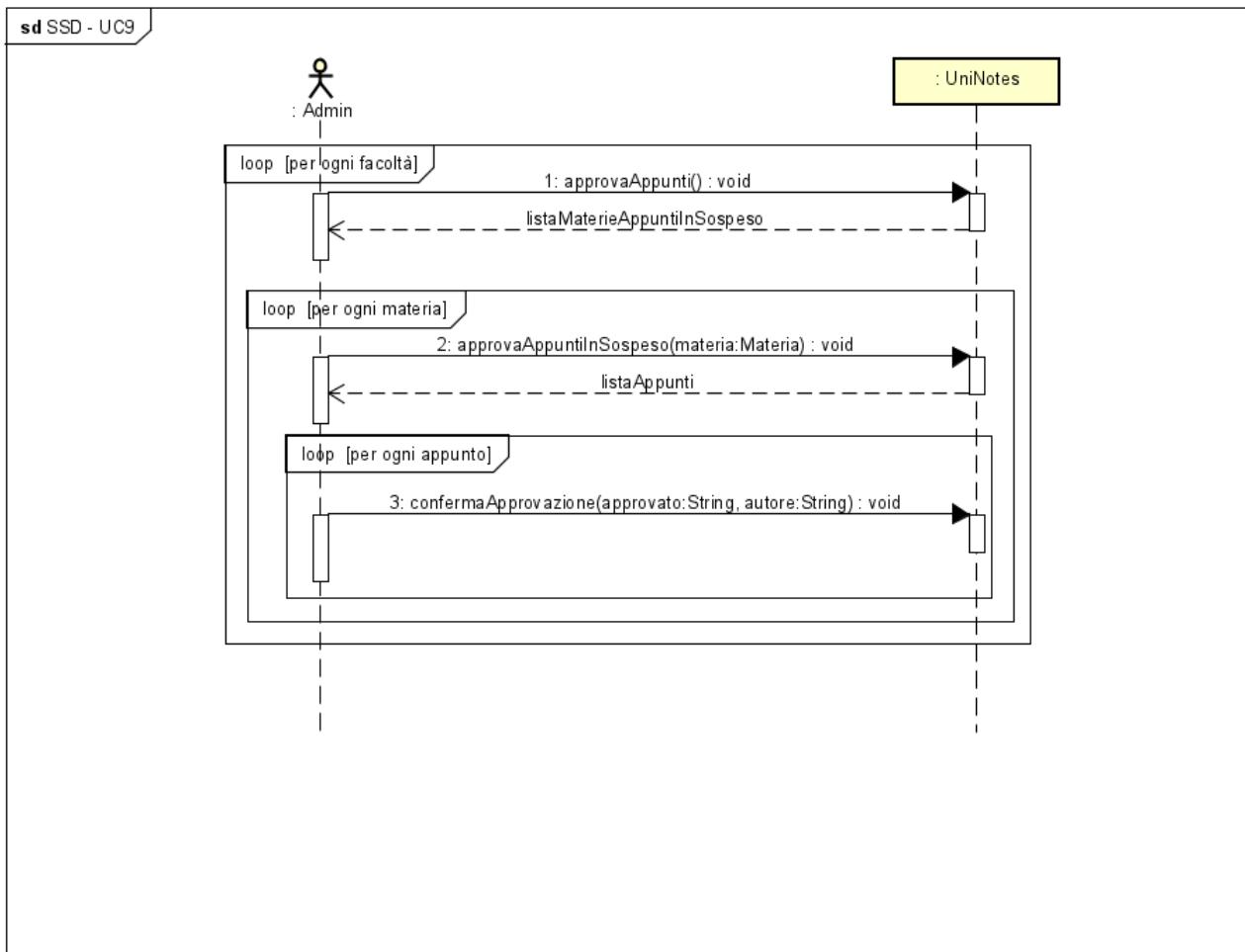
Analizzando il caso d’uso *UC9* emergono nuove classi concettuali, rispetto all’iterazione precedente:

- Carta di credito: contiene le informazioni relative alla carta di ciascun utente.

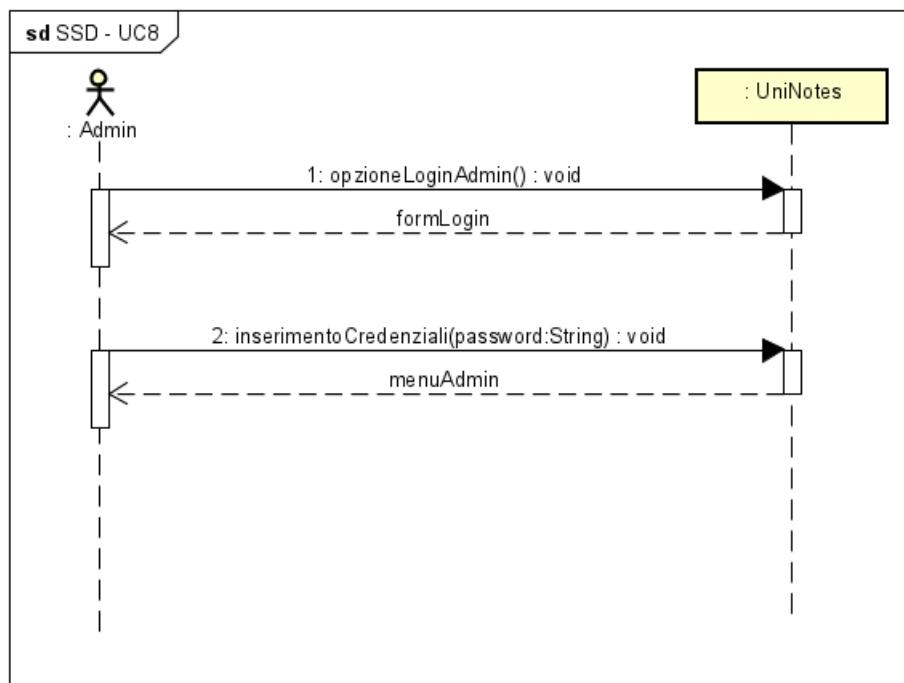
Dall’integrazione di questa nuova classe, tenendo conto di associazioni e attributi, è stato ricavato il seguente modello di dominio.



Diagrammi di sequenza di sistema



Di seguito il diagramma relativo al caso d'uso UC8.



Contratti delle operazioni

CONTRATTO CO1 - Opzione Login Admin

<i>Operazione</i>	OpzioneLoginAdmin()
<i>Riferimenti</i>	UC8
<i>Pre-condizioni</i>	-
<i>Post-condizioni</i>	- L'admin viene reindirizzato al form di login

CONTRATTO CO2 – Inserimento Credenziali

<i>Operazione</i>	inserimentoCredenziali(password)
<i>Riferimenti</i>	UC8
<i>Pre-condizioni</i>	L'admin inserisce correttamente le credenziali.
<i>Post-condizioni</i>	L'utente viene reindirizzato al menù utente

CONTRATTO CO3 - Approva Appunti

<i>Operazione</i>	approvaAppunti()
<i>Riferimenti</i>	UC9
<i>Pre-condizioni</i>	L'admin ha effettuato l'accesso
<i>Post-condizioni</i>	- Per ogni Facoltà f viene ritornata una lista di materie in cui sono presenti appunti in sospeso

CONTRATTO CO4 - Approva Appunti in sospeso

<i>Operazione</i>	approvaAppuntiInSospeso(materia)
<i>Riferimenti</i>	UC9
<i>Pre-condizioni</i>	È in corso l'approvazione di un appunto
<i>Post-condizioni</i>	- Per ogni materia viene ritornata la lista di appunti in sospeso

CONTRATTO CO5 - Conferma approvazione

<i>Operazione</i>	confermaApprovazione(approvato,autore)
<i>Riferimenti</i>	UC9
<i>Pre-condizioni</i>	-
<i>Post-condizioni</i>	- Per ogni appunto a della lista ritornata, l'admin sceglie se approvare o meno l'appunto

- Se a viene approvato, vengono incrementati i punti dell'autore
 - Viene inviata una mail all'autore dell'appunto con l'esito dell'approvazione

* *Progettazione*

Nuovamente, gli elaborati principali presi in considerazione sono i diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi).

Si osservi in particolare che per gestire gli scenari di fallimento del caso d'uso UC5, si è scelto di sfruttare il pattern GoF *Strategy*.

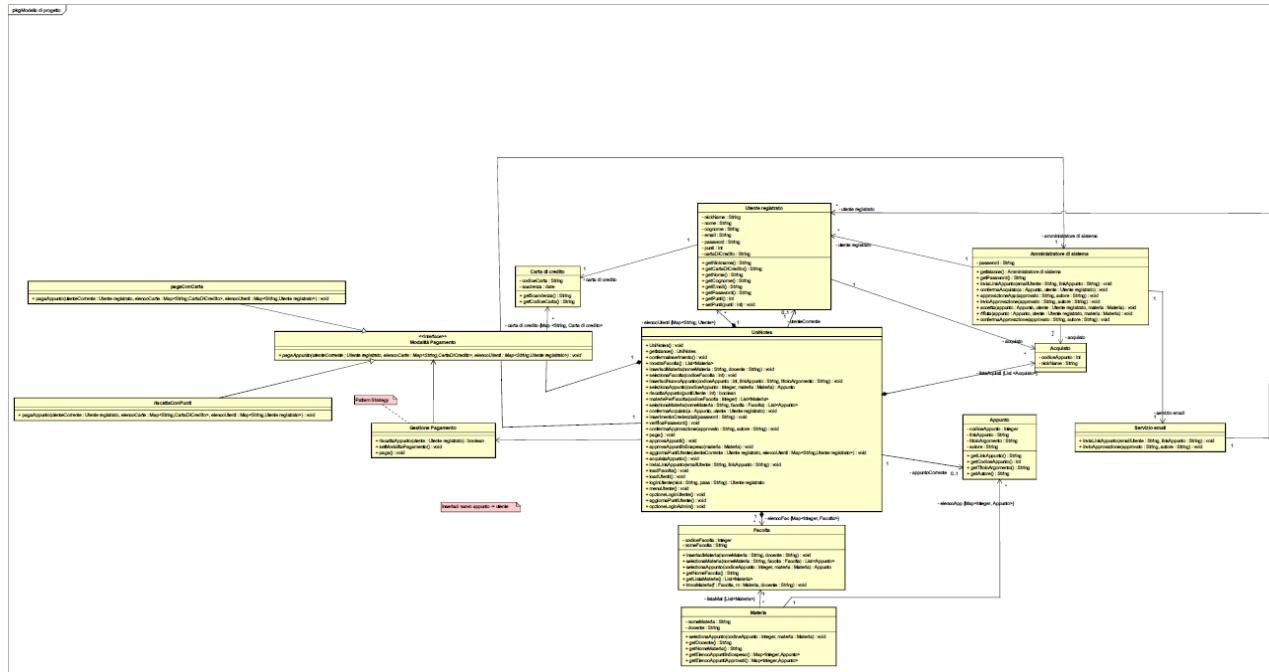
Esso permette di definire una varietà di algoritmi da applicare a seconda della necessità.

La Strategia può essere scelta direttamente a *runtime* perché sfrutta il polimorfismo.

Ogni algoritmo viene incapsulato all'interno di una classe, in modo da permettere la sua modifica indipendentemente da chi lo utilizza. Il Pattern suggerisce, quindi, l'incapsulamento del meccanismo di ciascun algoritmo, in apposite Classi che implementano un'interfaccia comune, la quale fornisce un accesso efficiente ai dati.

Tale Pattern, nel caso di studio proposto, si adegua all'esigenza di differenziare le modalità di pagamento degli appunti: la gestione del pagamento è affidata ad una classe che si chiama “Gestione pagamento” che setta la modalità di pagamento (punti o carta di credito) in base al ritorno di una funzione booleana che verifica la quantità di punti dell'utente che sta effettuando l'acquisto.

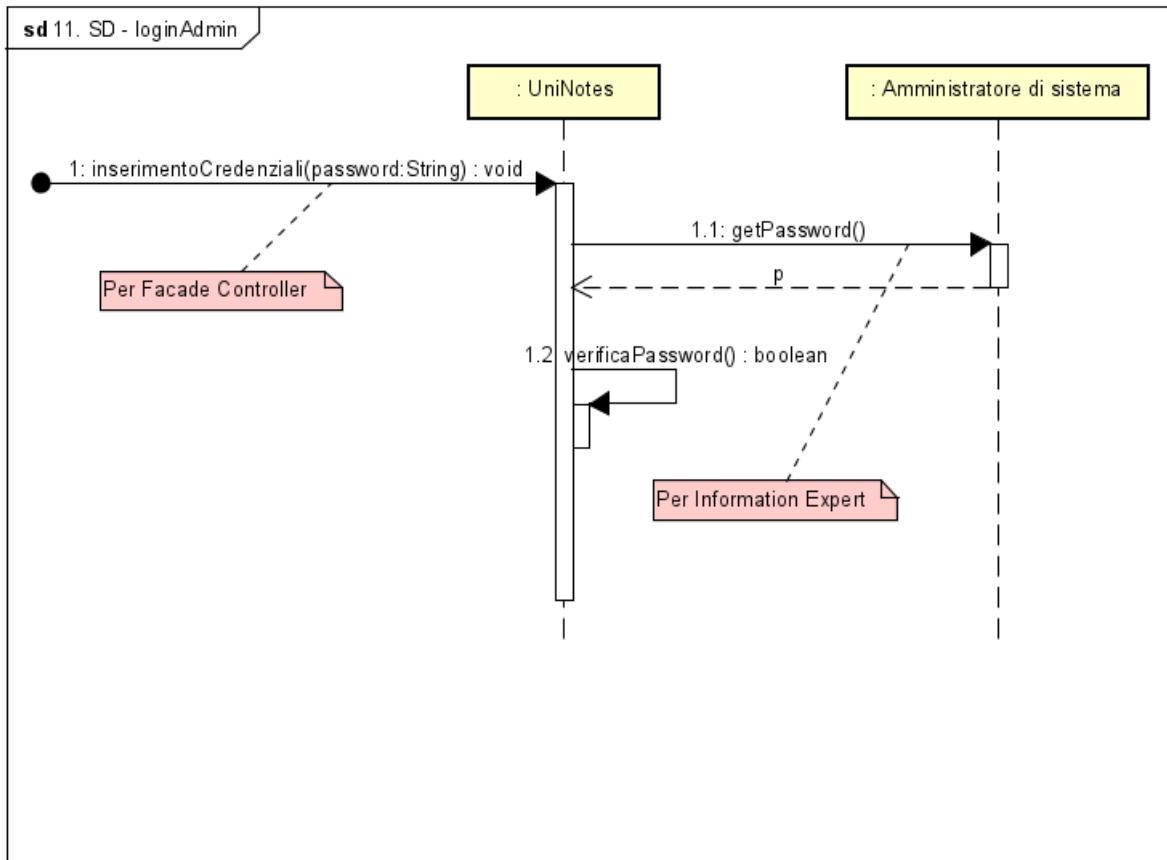
Diagramma delle classi



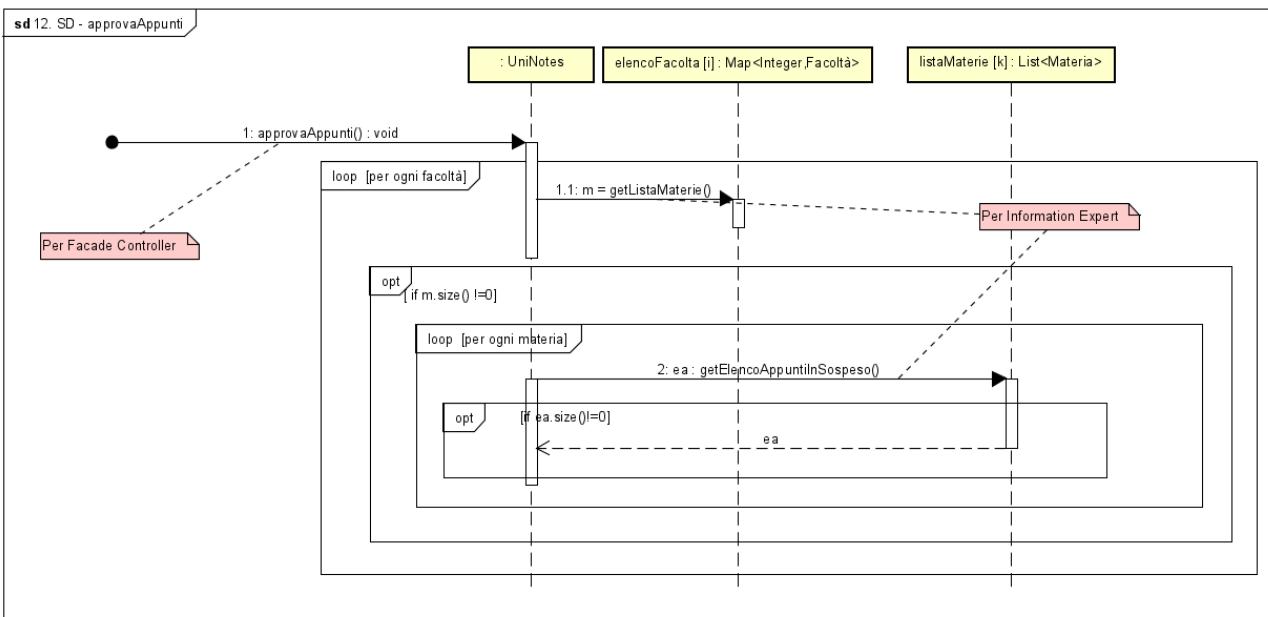
Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: in Iterazione 3 > UniNotes – IT3.astah.

Diagrammi di sequenza

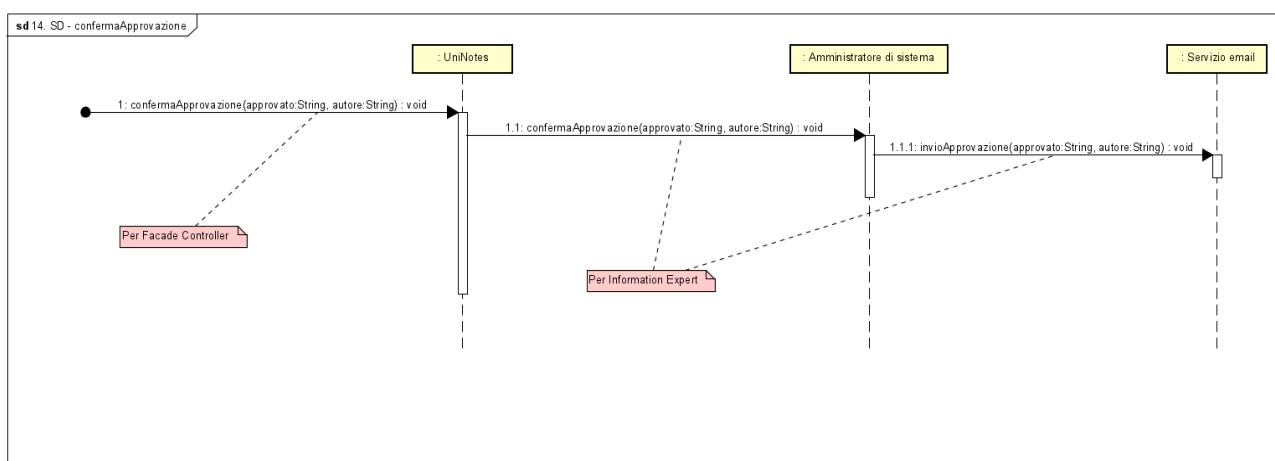
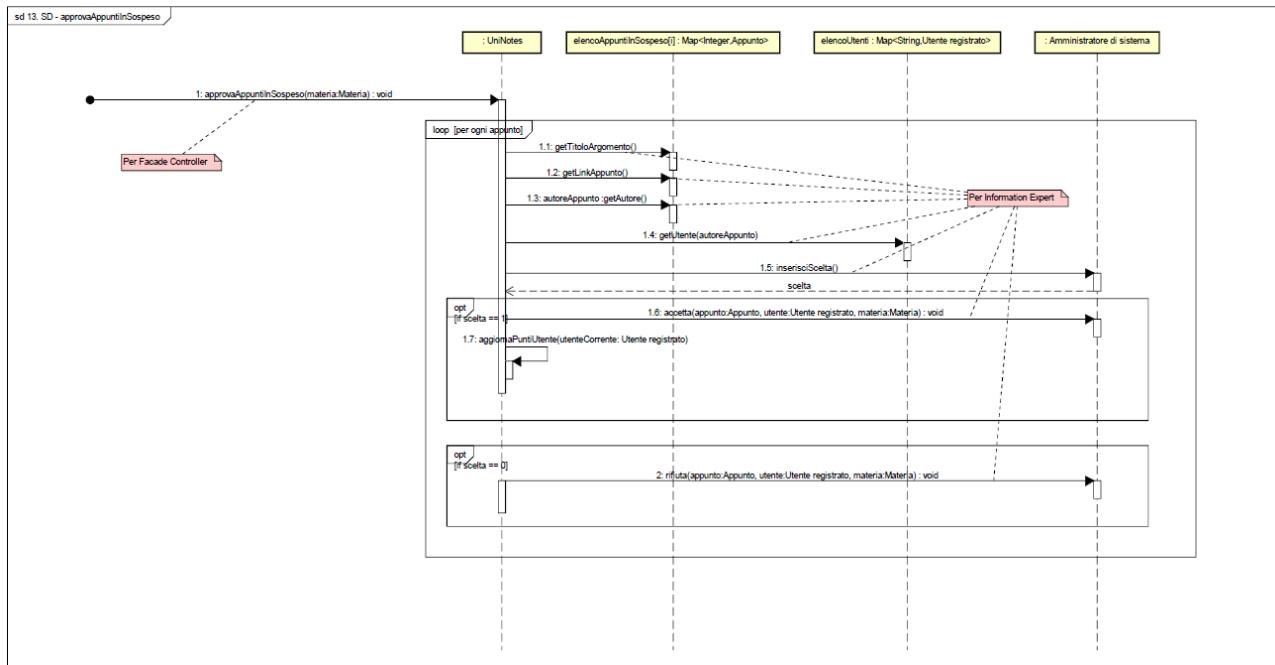
In questa sezione vengono mostrati gli sd per il caso d'uso UC5. Non vengono mostrati quelli già presentati nelle iterazioni precedenti.



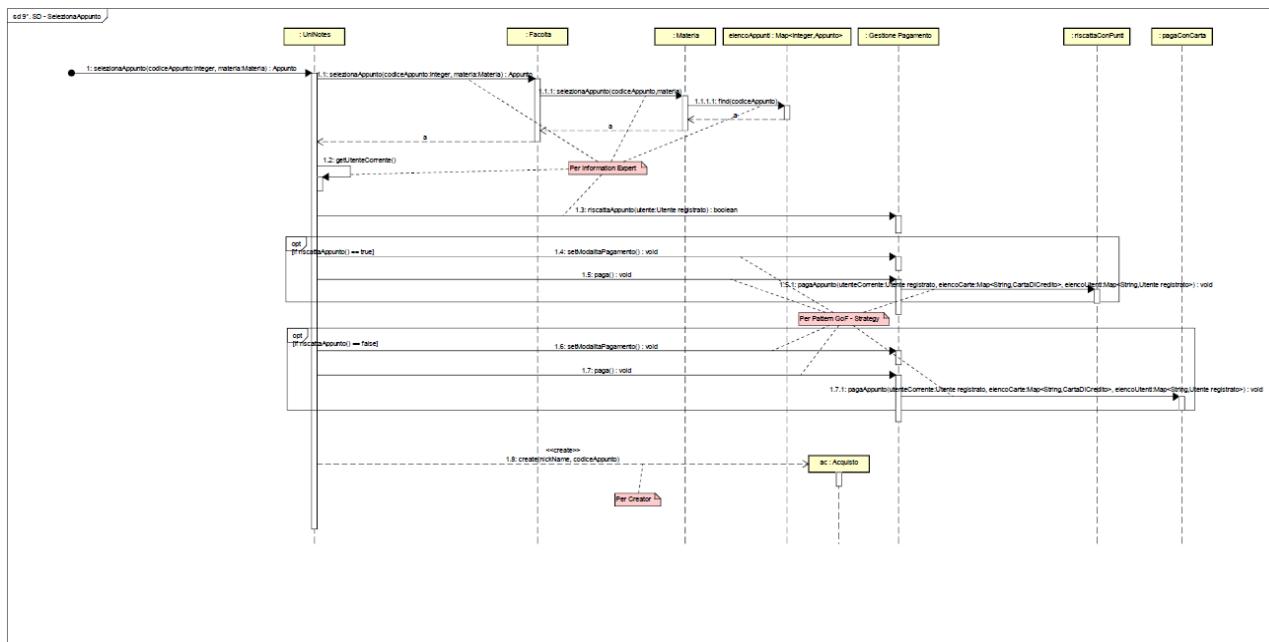
Viene supposto che l'Amministratore, una volta scelto di approvare gli appunti, debba approvare/rifiutare tutti quelli presenti in coda.



UniNotes



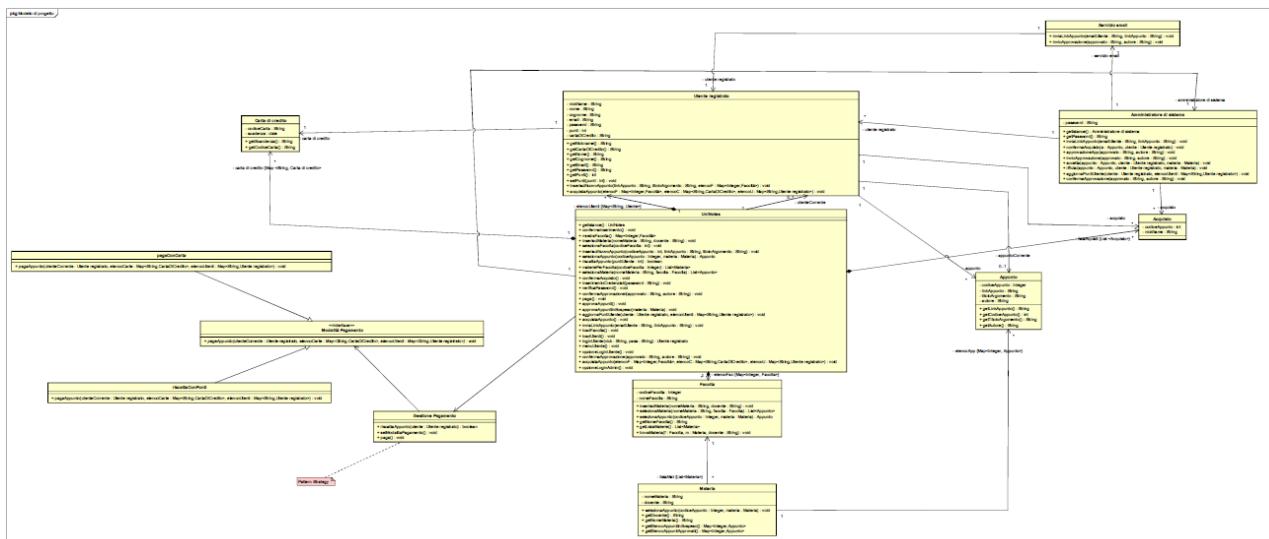
Di seguito il diagramma di interazione SD9 – Seleziona appunto, modificato rispetto all'iterazione precedente per l'aggiunta dell'utilizzo del pattern GoF **Strategy**.



* Refactoring

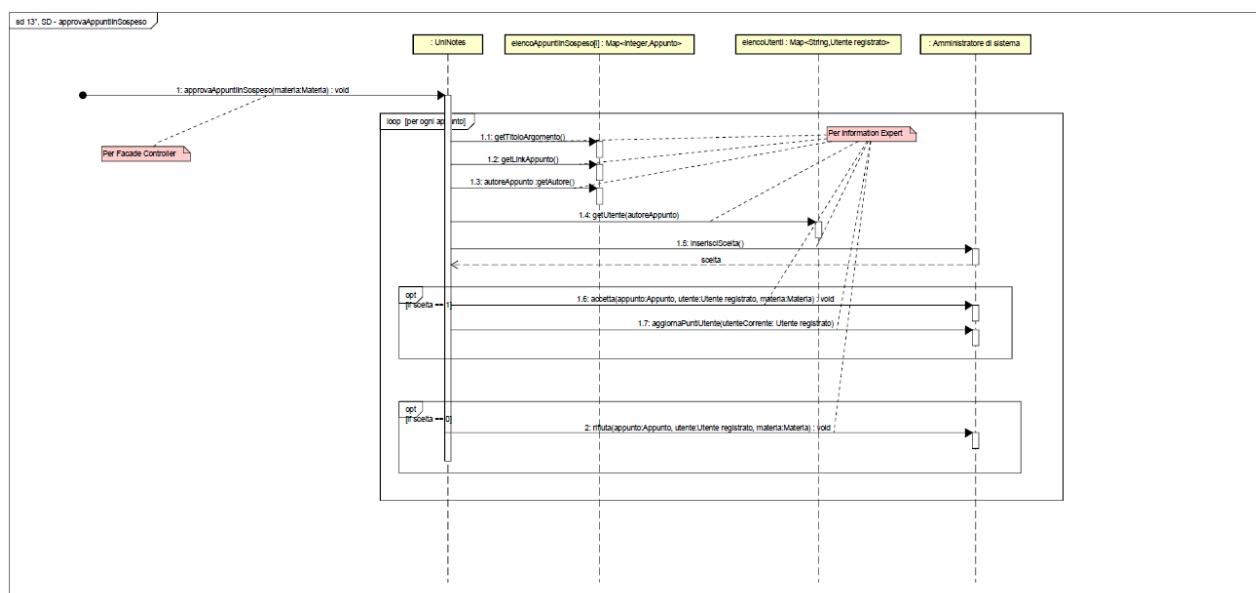
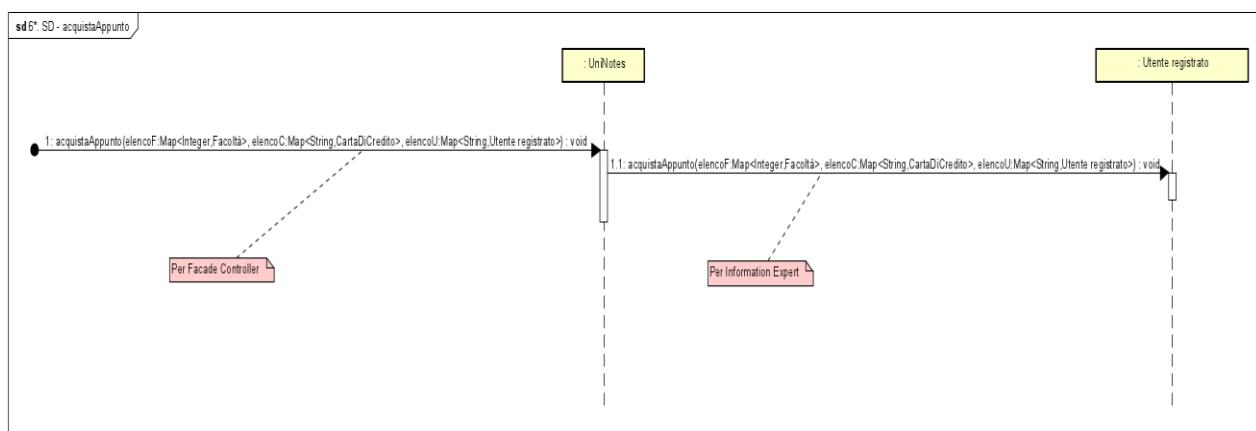
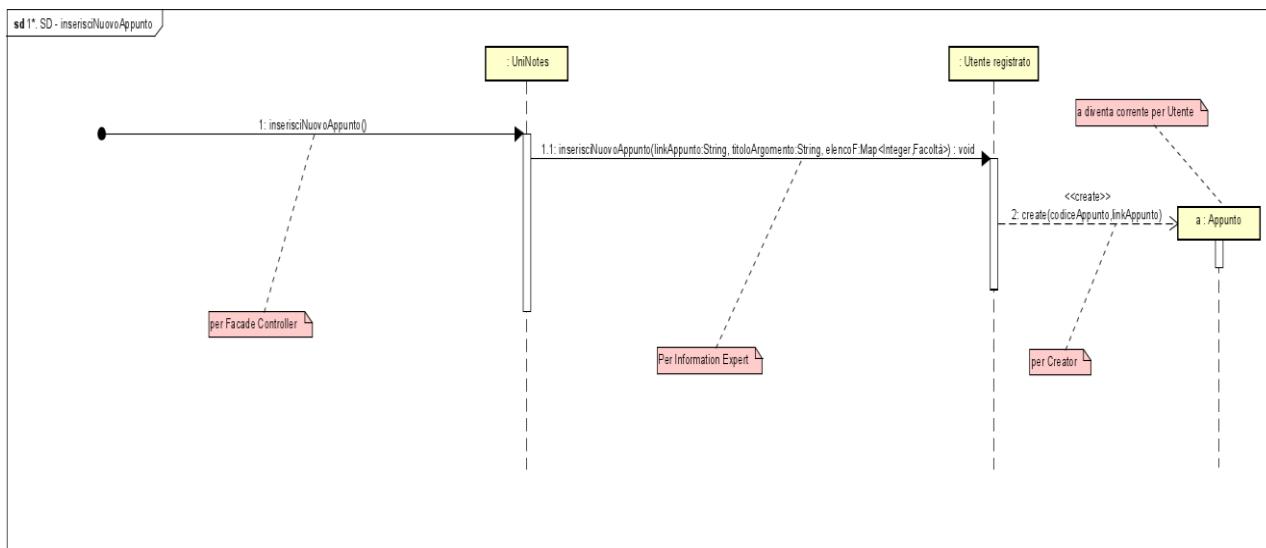
In fase di refactoring, dopo un'attenta analisi, è stato deciso di apportare alcune modifiche al sistema: osservando il Modello di Dominio, la Classe UniNotes risulta essere molto accoppiata e poco coesa; volendo sollevare il sistema da alcune responsabilità, per non sovraccaricarlo e rendere più omogenea la distribuzione dei metodi, sono state introdotte modifiche a:

- Modello delle classi di progetto
- Diagrammi di Sequenza (SD)
- Codice Java



Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 3 > UniNotes – IT3 – Refactoring.astah.

UniNotes



5. Iterazione 4 – UniNotes

* *Introduzione*

Durante questa quarta iterazione, ci si concentrerà sull'analisi e l'implementazione del caso d'uso UC6: inserire la recensione di un appunto.

- ✿ UC6: Permette all'utente di inserire una recensione di un appunto.

In questa iterazione viene presa in considerazione solo l'inserimento di una votazione relativa all'appunto senza un commento; per questo motivo viene assunto che tutte le recensioni siano automaticamente approvate, senza chiamare in causa l'amministratore. Tutte le recensioni vengono aggiunte in una lista recensioni.

Nel caso in cui la votazione non rientri all'interno di dei parametri stabiliti (1 – 5), viene stampato un messaggio di errore e viene chiesto di reinserire il voto; in caso di successo, per l'utente autore dell'appunto verrà aggiornato il campo “*valutazione*” aggiunto in questa iterazione.

NOME DEL CASO D'USO	UC6: Inserisci una recensione di un appunto.
Portata	Applicazione UniNotes
Livello	Obiettivo utente
Attore primario	Utente registrato
Parti interessate e interessi	Utente registrato: vuole valutare un appunto
Precondizioni	L'utente deve aver acquistato l'appunto che vuole recensire.
Garanzia di successo	L'appunto esiste
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'utente vuole recensire un appunto 2. L'utente seleziona dal menù la voce “Inserisci recensione” 3. Il sistema stampa l'elenco degli appunti acquistati dall'utente 4. L'utente sceglie l'appunto 5. L'utente compila il form inserendo il voto 6. Il sistema acquisisce i dati e ne verifica la correttezza

	7. L'utente autore riceve una mail con la nuova valutazione
Estensioni	<p>*a. In qualsiasi momento il sistema fallisce e si arresta in maniera improvvisa</p> <ol style="list-style-type: none"> 1. Viene visualizzato un messaggio di errore 2. Viene chiesto all'utente il riavvio dell'applicazione <p>2a. Non è stato effettuato alcun acquisto:</p> <ul style="list-style-type: none"> - L'utente viene reindirizzato al menù
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Legato al numero di volte che in cui un utente vuole recensire un appunto
Varie	

* *Analisi orientata agli Oggetti*

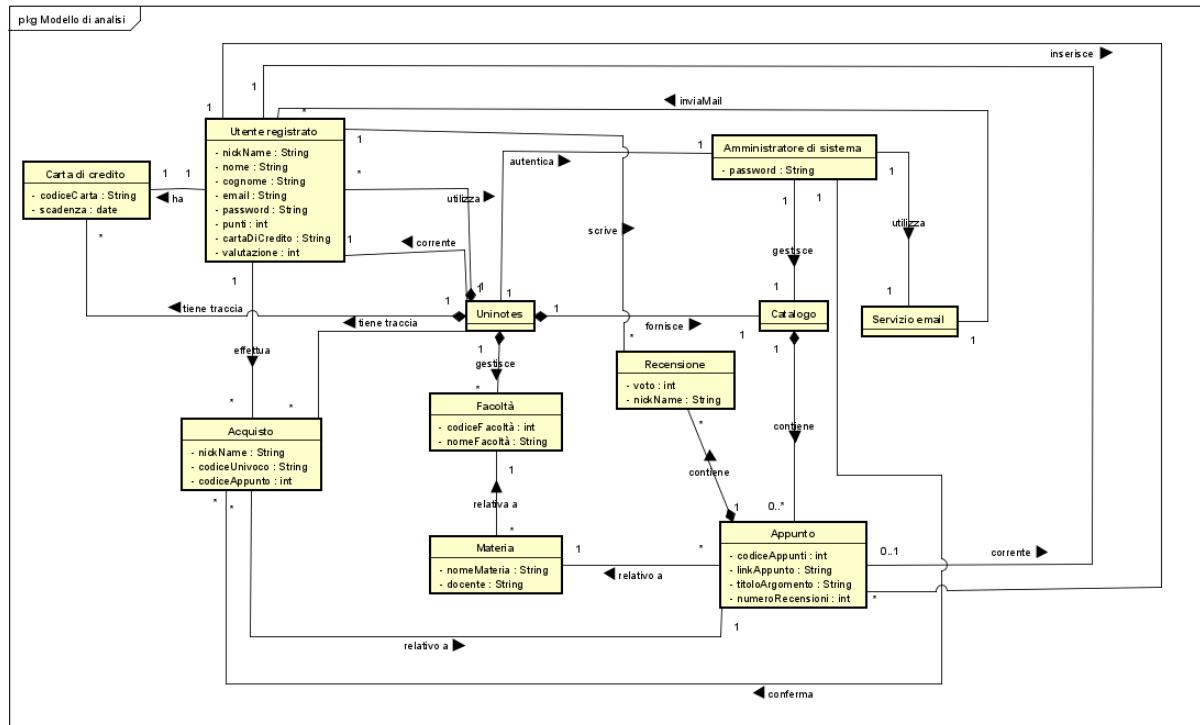
Al fine di descrivere il dominio da un punto di vista ad oggetti e gestire ulteriori requisiti, saranno utilizzati nuovamente gli stessi strumenti delle iterazioni precedenti. Tuttavia, esaminando il caso d'uso *UC6*, si è deciso di modificare il Modello di Dominio aggiungendo una classe “Recensione” e due relazioni: “scrive” tra “Recensione” e “Utente Registrato”, e “contiene” tra “Recensione” e “Appunto”. È stato inoltre aggiunto un attributo “valutazione” per Utente registrato, necessario per il calcolo dei punti al caricamento di nuovo appunto.

Modello di dominio

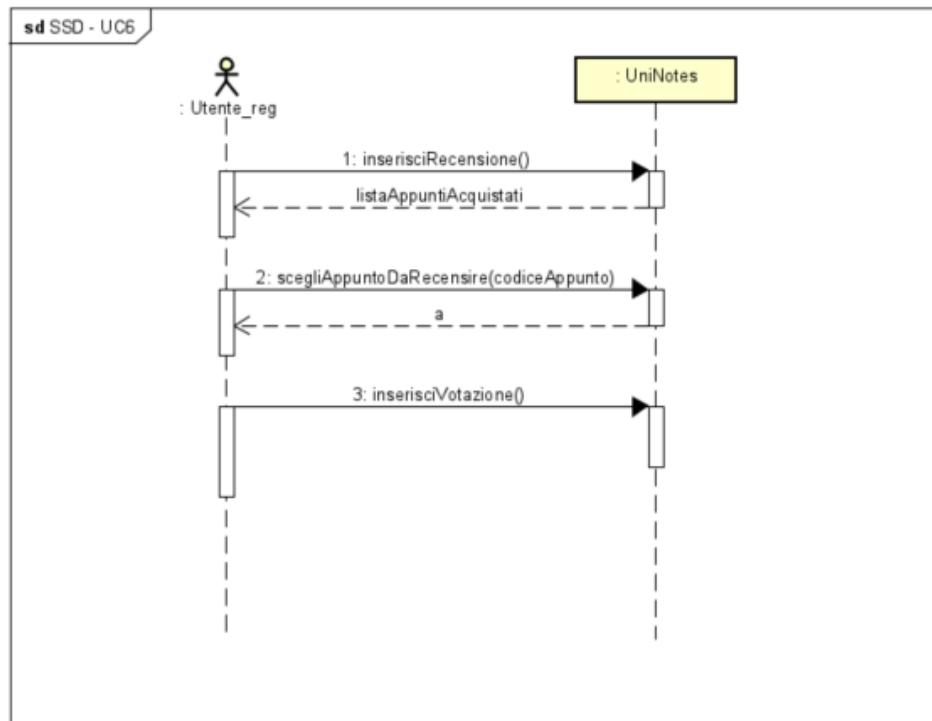
Analizzando il caso d'uso UC6 emergono nuove classi concettuali, rispetto all'iterazione precedente:

- ✚ Recensione: contiene le informazioni relative alla recensione di un appunto (nickName dell'utente che recensione e la votazione assegnata).

Dall'integrazione di questa nuova classe, tenendo conto di associazioni e attributi, è stato ricavato il seguente modello di dominio.



Diagrammi di sequenza di sistema



Contratti delle operazioni

CONTRATTO CO1 – Inserisci recensione	
<i>Operazione</i>	inserisciRecensione()
<i>Riferimenti</i>	UC6
<i>Pre-condizioni</i>	L’utente ha effettuato l’accesso. È in corso l’inserimento di una recensione.
<i>Post-condizioni</i>	- È stata recuperata la lista degli appunti acquistati e non recensiti da un utente sulla base del nickName dell’utente

CONTRATTO CO2 – Scegli appunto da recensire	
<i>Operazione</i>	scegliAppuntoDaRecensire(codiceAppunto)
<i>Riferimenti</i>	UC6
<i>Pre-condizioni</i>	È in corso l’inserimento di una recensione.
<i>Post-condizioni</i>	- È stata recuperata l’istanza a di appunto sulla base del codiceAppunto

CONTRATTO CO3 – Inserisci Votazione	
<i>Operazione</i>	inserisciVotazione ()
<i>Riferimenti</i>	UC6
<i>Pre-condizioni</i>	È in corso l’inserimento di una recensione.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - Viene creata l’istanza r di Recensione - r viene aggiunta alla lista di recensioni - viene incrementato il valore numeroRecensioni su appunto - viene invocata la funzione update()

* *Progettazione*

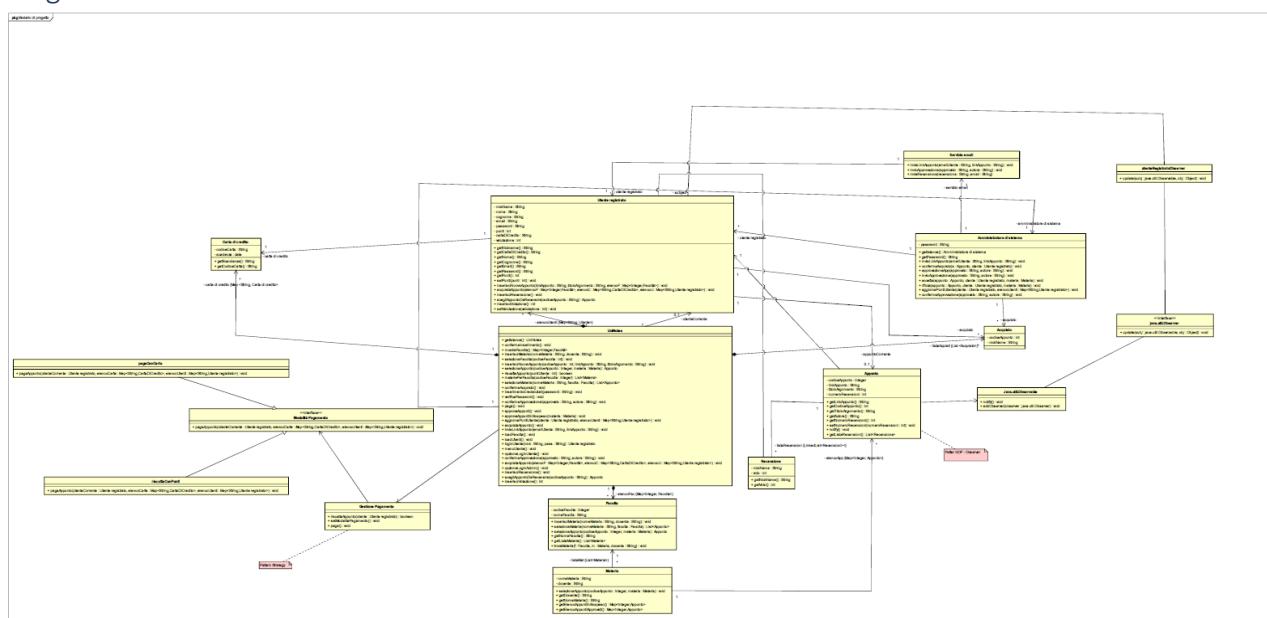
Nuovamente, gli elaborati principali presi in considerazione sono i diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi).

Si osservi in particolare che per gestire l’aggiornamento dell’attributo valutazione dell’autore dell’appunto appena recensito, si è scelto di sfruttare il **pattern GoF Observer**: esso si occupa di osservare un oggetto e compiere delle azioni quando viene modificato. In generale questa tipologia di Pattern può “osservare” anche più di un unico oggetto, senza necessariamente averne il riferimento; esso, tecnicamente, non osserva (come si potrebbe pensare dal nome), ma rimane in attesa di essere notificato dall’oggetto in questione.

Un gruppo di oggetti *Observer* viene istanziato a *runtime* e viene associato all'oggetto da osservare, esso, tuttavia non conosce l'identità esatta degli *Observer*, in quanto si occupa solamente di notificare il gruppo intero.

Per istanziare un oggetto *Observer* si utilizza una classe (nel nostro caso abbiamo implementato la classe *utenteRegistratoObserver*) che implementa l'interfaccia “*Observer*”, che avrà il compito di svolgere delle operazioni quando lo stato dell'oggetto osservato (nel caso di questa implementazione, l'appunto) varia; in particolare quando un appunto viene recensito la classe Appunto dovrà quindi estendere la classe *Observable* in modo da poter notificare (tramite la funzione *notify*) tutti gli *Observer* quando varia lo stato di un proprio oggetto, in questo caso il numero delle recensioni. Ogni recensione inserita fa variare la “valutazione” di un utente, mediando il suo punteggio per ottenere in seguito un numero di punti adeguati.

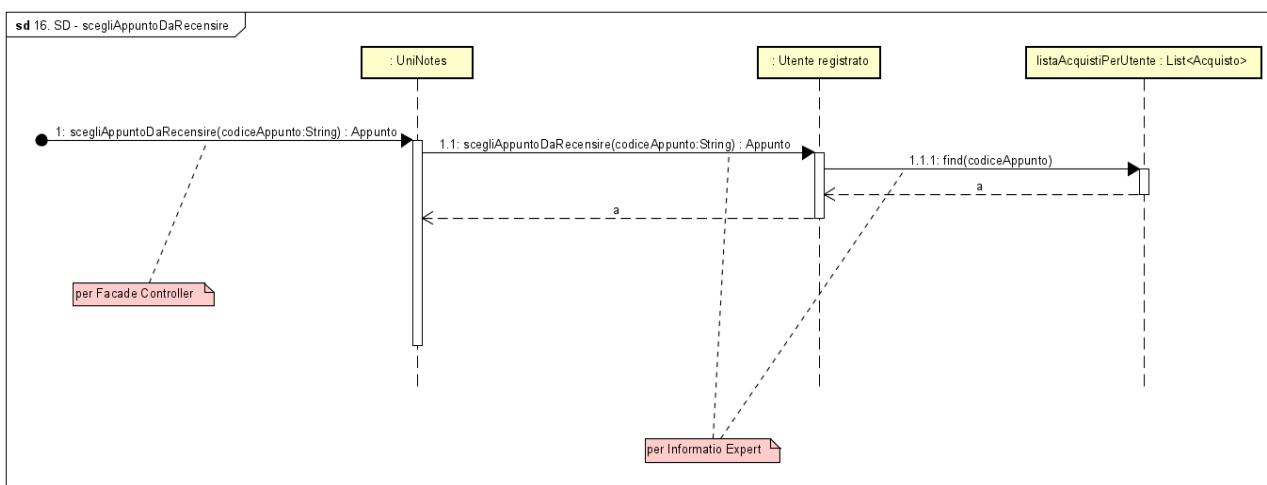
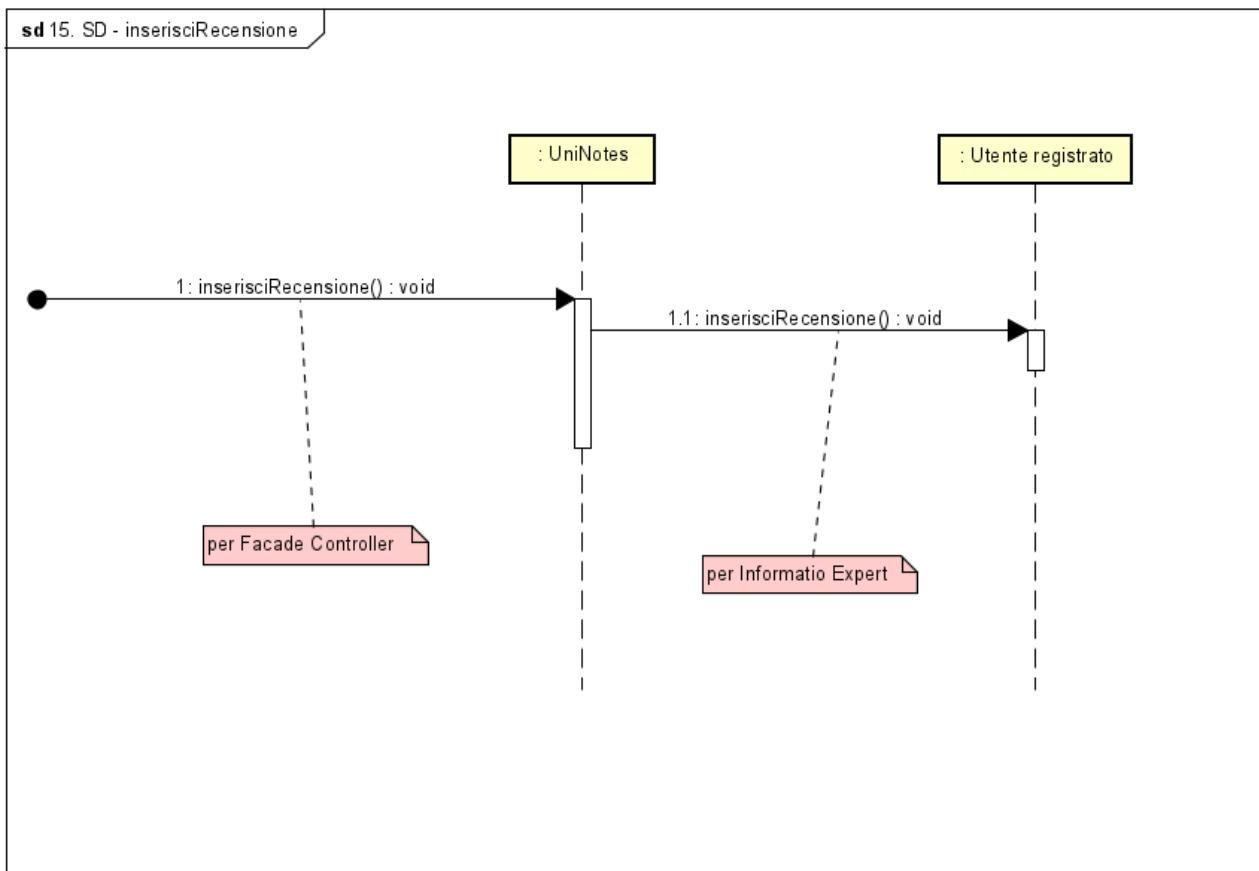
Diagramma delle classi

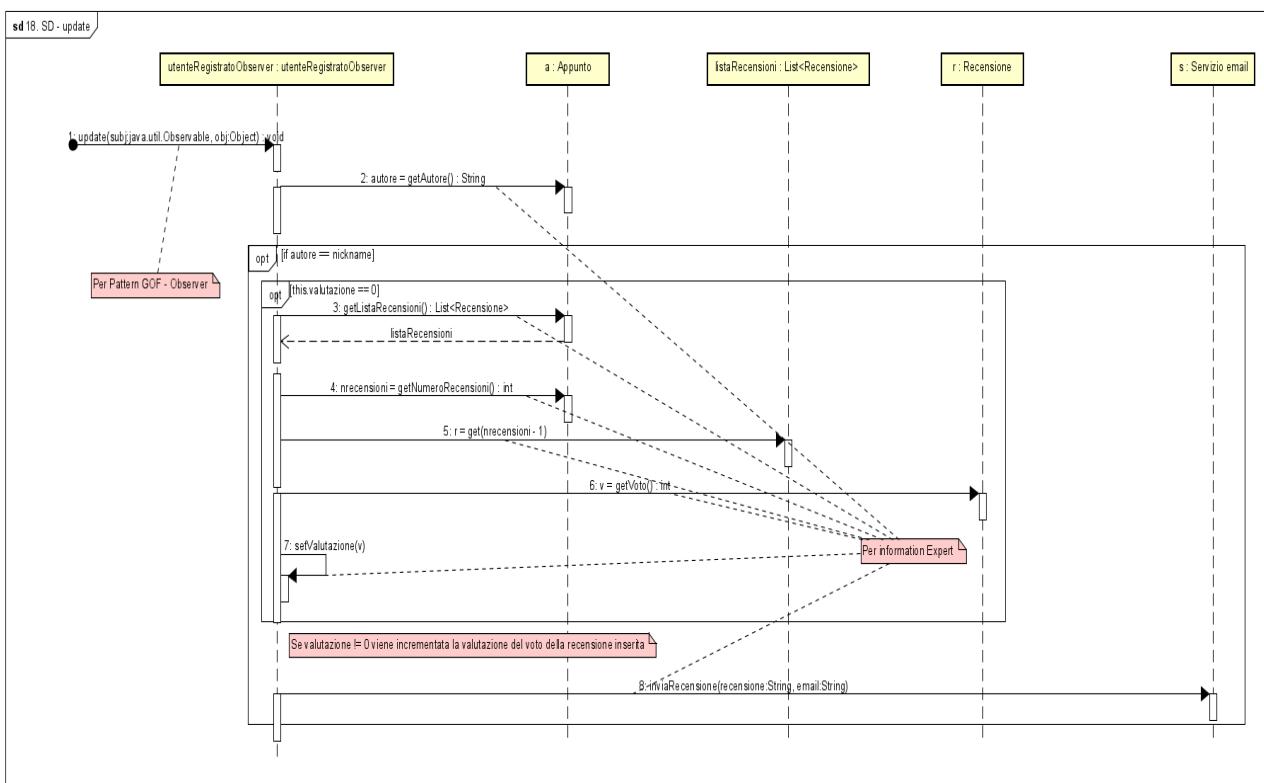
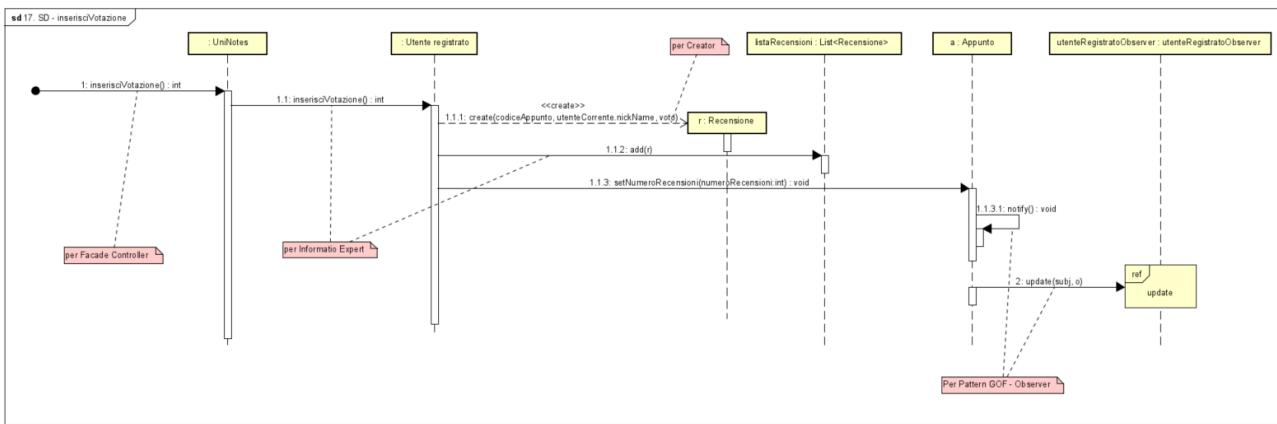


Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 4 > UniNotes – IT4.

Diagrammi di sequenza

In questa sezione vengono mostrati gli sd per il caso d'uso UC6. Non vengono mostrati quelli già presentati nelle iterazioni precedenti.





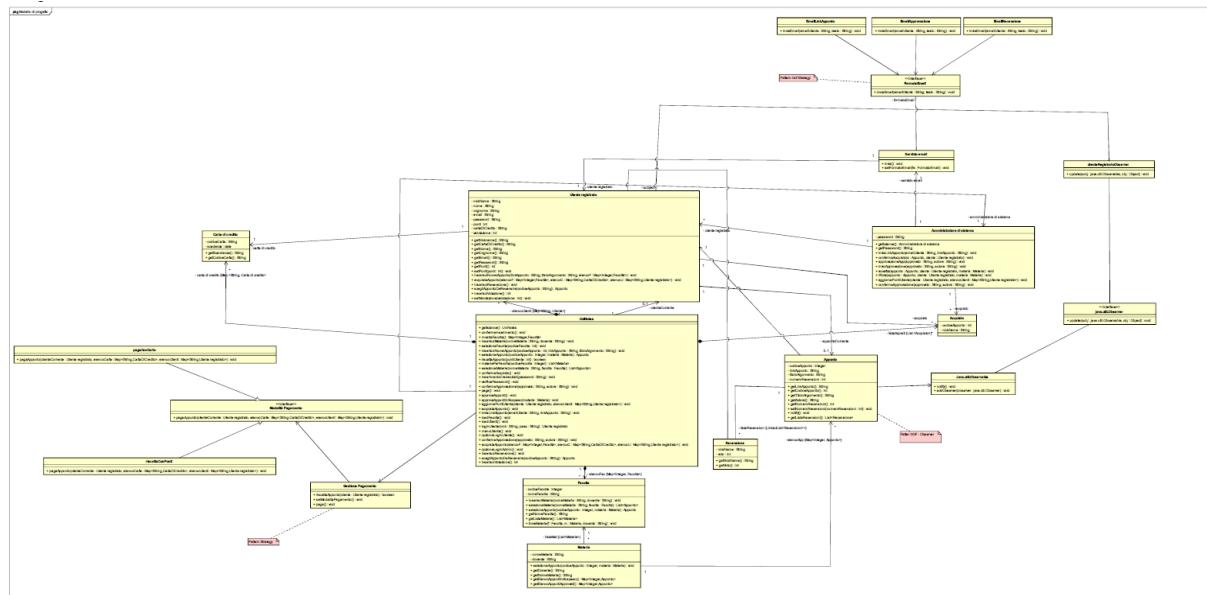
Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 4 > UniNotes – IT4.astah.

* Refactoring

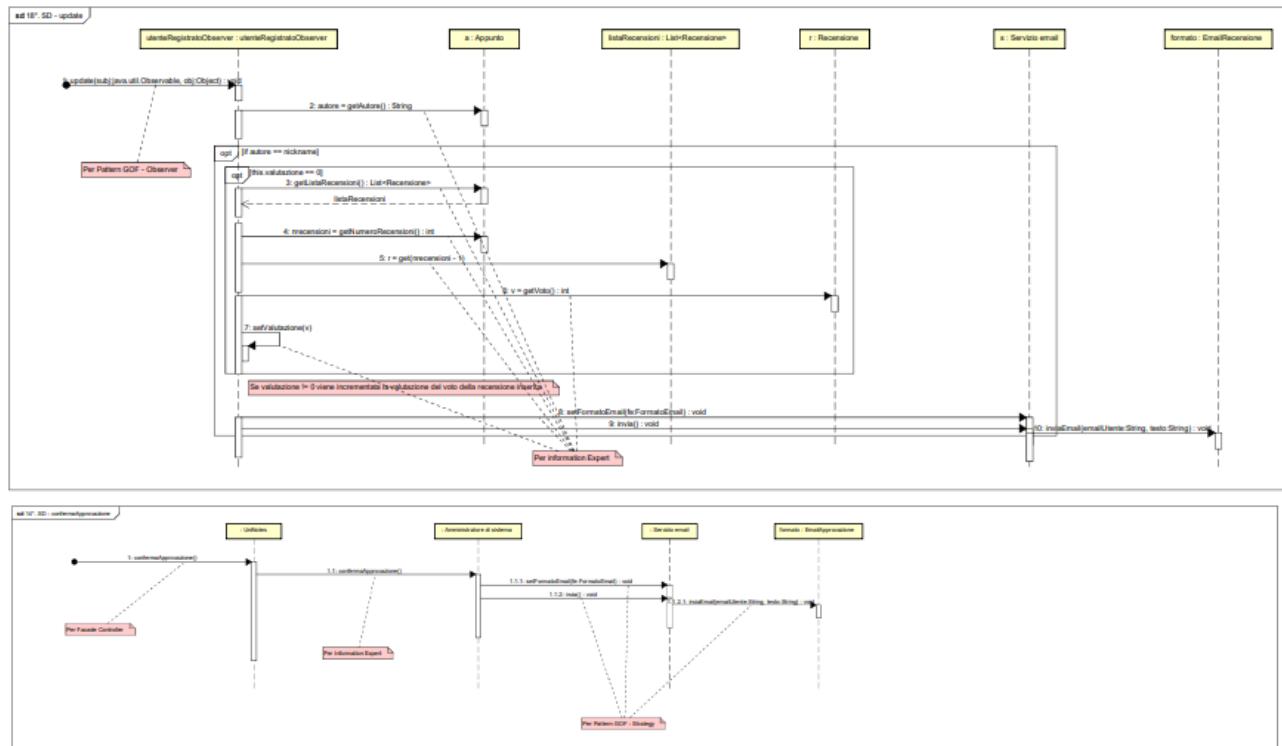
In fase di refactoring, dopo un'attenta analisi, è stato deciso di apportare alcune modifiche al sistema: volendo gestire in maniera più efficace il servizio di invio email, si è deciso di implementarlo utilizzando il pattern GoF **Strategy**: in particolare, la gestione dell'invio email è affidata ad una classe che si chiama “Servizio email” che setta la modalità di invio (EmailApprovazione, EmailLinkAppunto o EmailRecensione) in base al ritorno di una funzione che ritorna un valore in base a l'operazione da eseguire.

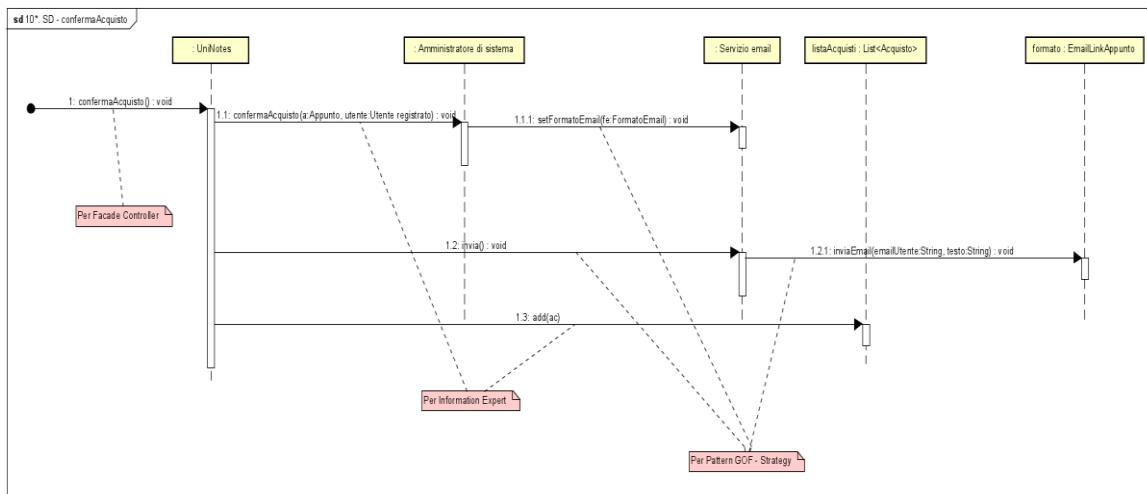
Volendo ottimizzare la gestione dell'invio delle e-mail sono state introdotte modifiche a:

- Modello delle classi di progetto
- Diagrammi di Sequenza (SD)
- Codice Java



Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 4 > UniNotes – IT4 - Refactoring.astah.





Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 4 > UniNotes – IT4 – Refactoring.astah.

6. Iterazione 5 – UniNotes

* *Introduzione*

Durante questa terza iterazione, ci si concentrerà sull’analisi e l’implementazione del caso d’uso UC2: Registrazione nel sistema.

- ✚ UC2: Permette all’utente di registrarsi al sistema.

In questa iterazione viene presa in considerazione la registrazione al sistema di un utente non registrato ed inoltre quando si procede all’acquisto di un appunto vengono stampate a video le relative recensioni e valutazioni (il caso d’uso UC1 non è stato approfondito separatamente perché è stato implementato per permettere l’utilizzo delle altre funzioni del Sistema).

* *Analisi orientata agli Oggetti*

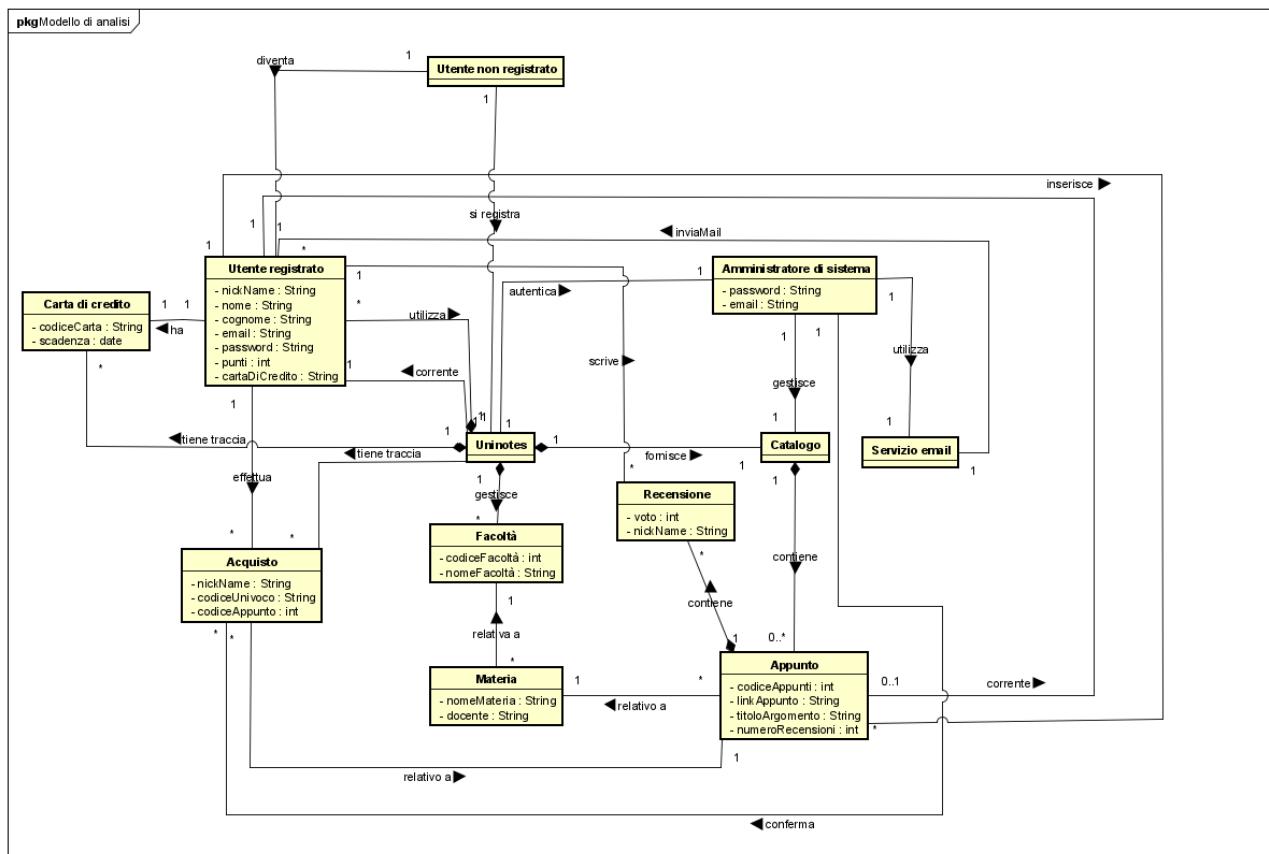
Al fine di descrivere il dominio da un punto di vista ad oggetti e gestire ulteriori requisiti, saranno utilizzati nuovamente gli stessi strumenti delle iterazioni precedenti. Tuttavia, esaminando il caso d’uso *UC2*, si è deciso di modificare il Modello di Dominio aggiungendo una classe “Utente non registrato” e due relazioni : “diventa” tra “Utente registrato” e “Utente non Registrato”, e “si registra” tra “Uninotes” e “Utente non registrato”. È stato inoltre aggiunto un attributo “email” per Amministratore, necessario il login gestito con il Pattern GoF.

Modello di dominio

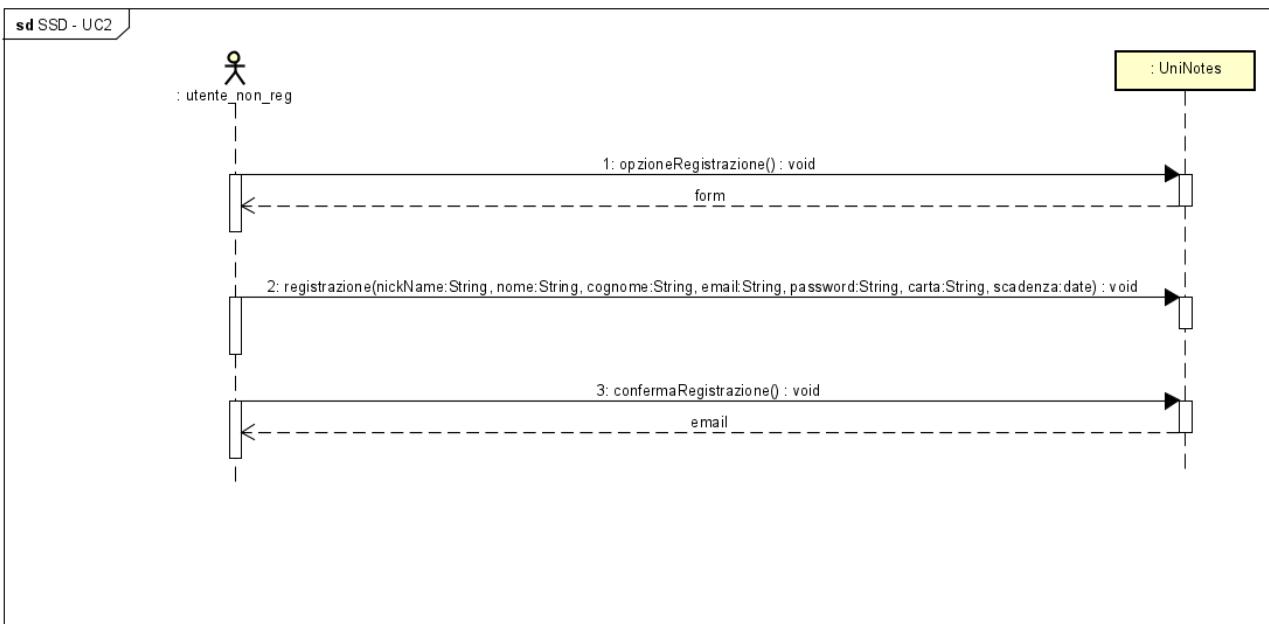
Analizzando il caso d’uso UC2 emergono nuove classi concettuali, rispetto all’iterazione precedente:

- ✚ Utente non registrato

Dall’integrazione di questa nuova classe, tenendo conto delle sue associazioni, è stato ricavato il seguente modello di dominio.



Diagrammi di sequenza di sistema



Contratti delle operazioni

CONTRATTO CO1 – opzioneRegistrazione	
<i>Operazione</i>	opzioneRegistrazione()
<i>Riferimenti</i>	UC2
<i>Pre-condizioni</i>	-
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - L'utente non registrato viene reindirizzato al form di registrazione

CONTRATTO CO2 – registrazione	
<i>Operazione</i>	registrazione(nickname, nome, cognome, email, password, carta, scadenza)
<i>Riferimenti</i>	UC2
<i>Pre-condizioni</i>	L'utente inserisce correttamente le credenziali.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - L'utente si è registrato - Viene creata un'istanza u di utente con i nuovi attributi - Viene reindirizzato al menù utente - Viene settata la variabile utenteCorrente uguale a u

CONTRATTO CO3 - confermaRegistrazione	
<i>Operazione</i>	confermaRegistrazione()
<i>Riferimenti</i>	UC2
<i>Pre-condizioni</i>	La registrazione è avvenuta con successo.
<i>Post-condizioni</i>	<ul style="list-style-type: none"> - Viene inviata una email all'utente contenente le sue credenziali

* **Progettazione**

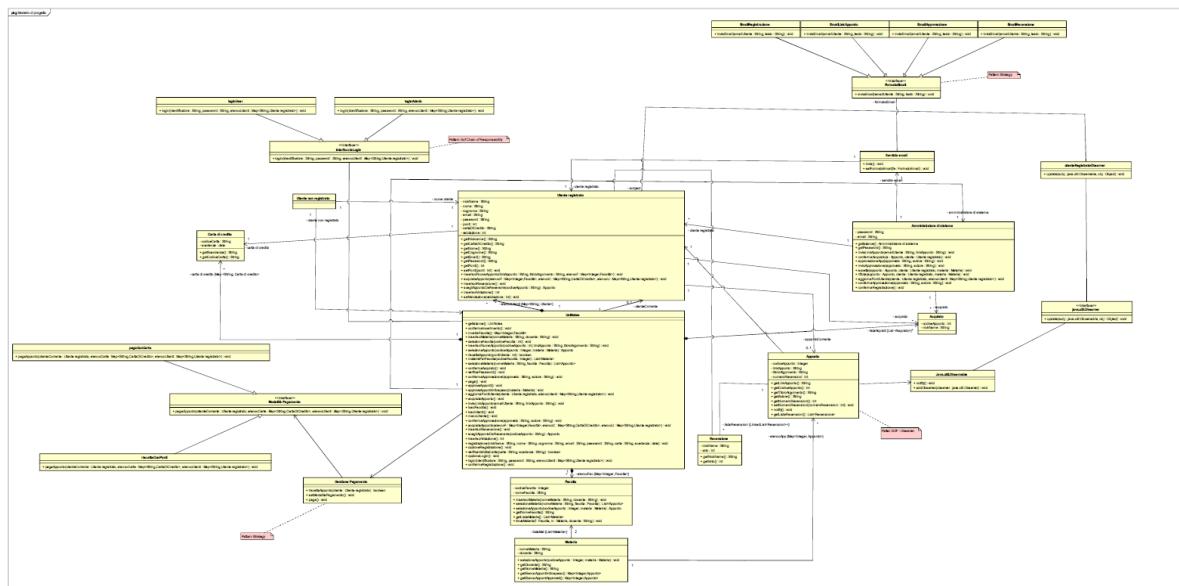
Nuovamente, gli elaborati principali presi in considerazione sono i diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi).

In questa operazione, per ottimizzare la gestione del login (amministratore o utente), è stato utilizzato il pattern GoF: **Chain of Responsibility**. Questo pattern consente di separare il mittente di una richiesta dal destinatario, in modo di consentire a più di un oggetto di gestire la richiesta. Gli oggetti destinatari vengono messi in catena, e la richiesta trasmessa dentro questa catena fino a trovare un oggetto che la gestisca. Il “Chain of responsibility” propone la costruzione di una catena di oggetti responsabili della gestione del login. Quando un oggetto della catena riceve una richiesta, analizza se corrisponde a lui gestirla, o, altrimenti, inoltrarla al seguente oggetto

dentro la catena. Nel nostro caso la prima richiesta viene inoltrata al login amministratore (verifica se l'accesso è stato effettuato tramite e-mail o tramite nickname e ne verifica la correttezza insieme alla password) e la seconda al login utente.

Inoltre, è stata effettuata una modifica al diagramma delle classi aggiungendo EmailRegistrazione che implementa l'interfaccia FormatoEmail (pattern GoF *Strategy*) per gestire l'invio delle credenziali del nuovo utente.

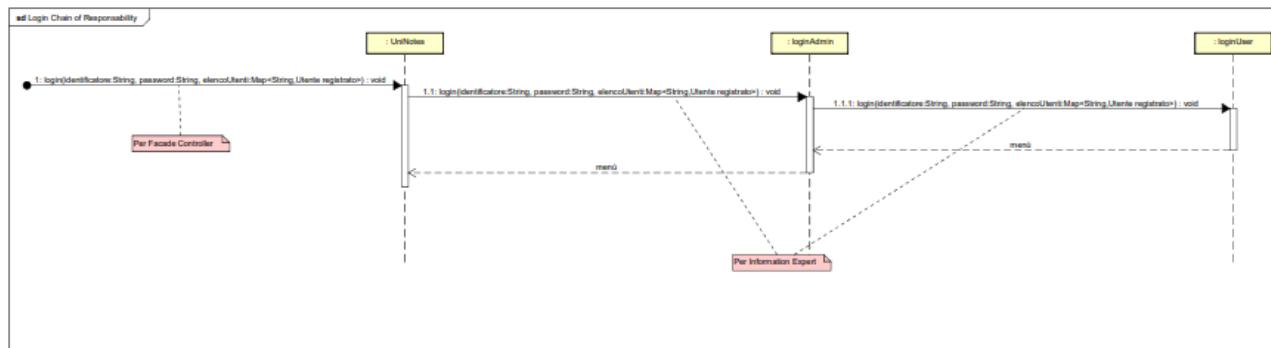
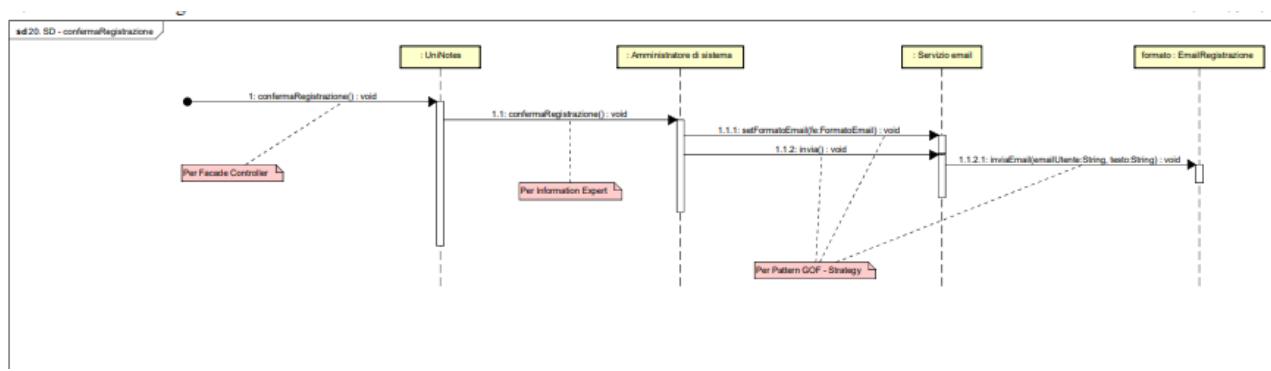
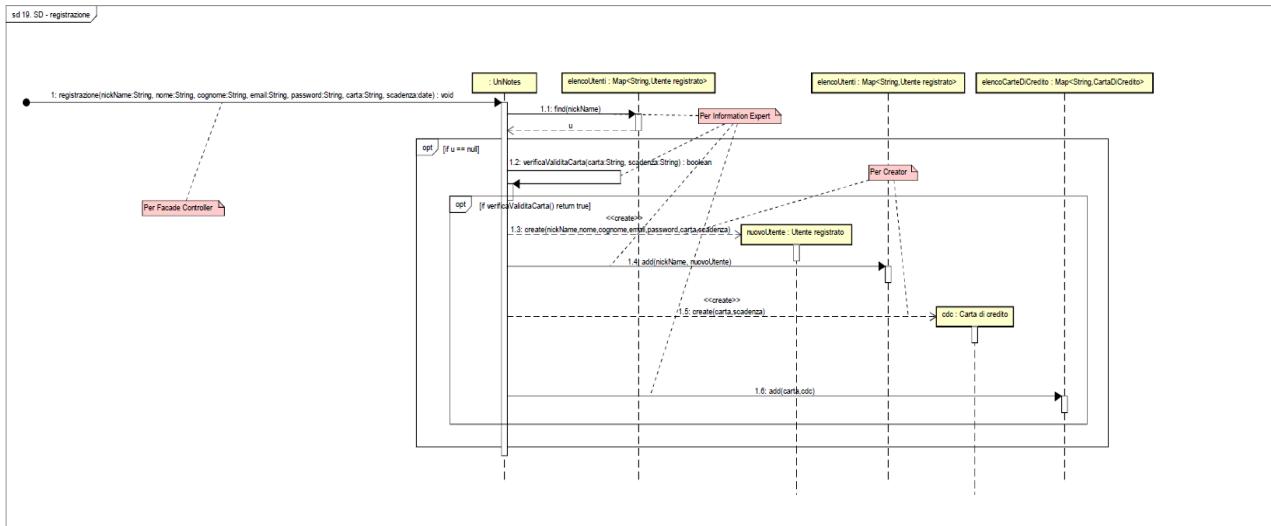
Diagramma delle classi



Per visualizzarlo nel dettaglio in maniera chiara, è stato inserito tra gli allegati: Iterazione 5 > UniNotes – IT5.astah.

Diagrammi di sequenza

In questa sezione vengono mostrati gli sd per il caso d'uso UC2. Non vengono mostrati quelli già presentati nelle iterazioni precedenti.



* Refactoring

In fase di refactoring non sono state apportate modifiche al modello di dominio. È stata tuttavia implementata una miglioria a livello di codice del programma, gestendo l'invio delle e-mail tramite multi-Threading.

7. TESTING

Il **testing** è una fase essenziale del processo di sviluppo di un programma solido ed efficiente. Grazie all'utilizzo del testing è possibile ridurre notevolmente i costi di manutenzione di un'applicazione. Per rilevare un maggior numero di malfunzionamenti è necessario eseguire un numero considerevole di test; tuttavia, è difficile testare un programma completamente. Un testing progettato accuratamente può rivelare anomalie nell'applicazione che una volta risolte rendono il software realizzato funzionante e funzionale secondo le specifiche del committente.

I test si dividono principalmente in due famiglie:

- **I test unitari (*Unit Test*)**: Sono semplici test/prove che vanno a verificare la correttezza direttamente del codice, in ogni sua piccola parte. L'idea dello Unit Test in Java è quella di valutare ogni singolo metodo in funzione dei valori attesi.
- **I test funzionali**: Sono dei test che vanno a verificare che il sistema software nella sua completezza funzioni correttamente. Questi test trattano il sistema come se fosse una scatola nera alla quale danno degli input e verificano la correttezza degli output.

Per eseguire il testing dell'applicazione realizzata, si è scelto di concentrarsi principalmente su test unitari eseguiti tramite l'ausilio di JUnit e quindi di seguire un approccio **BottomUp** collaudando prima le piccole unità che comporranno poi il programma completo.

Per ciascuno sono state individuate le **classi di equivalenza** tali da poter ridurre il numero di test al minimo insieme necessario, in particolare:

Test Uninotes

- **Test Load Utenti**: verifica che una volta caricato dal file (non vuoto) l'elenco degli utenti , le istanze di utenti non risultino “NULL”
- **Test Load Facoltà**: verifica che una volta caricato dal file (non vuoto) l'elenco delle facoltà, le istanze di facoltà non risultino “NULL”
- **Test Materia per facoltà**: verifica che una volta inserita una nuova materia per una determinate facoltà, l'elenco delle materie ritornate dal metodo *MateriePerFacolta* non risulti “NULL”
- **Test Set Utente corrente**: verifica che la variabile *utenteCorrente* non risulti “NULL”
- **Test Load Carta**: verifica che una volta caricato dal file (non vuoto) l'elenco delle carte di credito, le istanze di *CartaDiCredito* non risultino “NULL”
- **Test Registrazione**: verifica che una volta inseriti i dati per la registrazione:
 - Se la carta non è valida si generi l'eccezione *carta non valida*
 - Se tutti i dati sono corretti viene aggiunto correttamente all'elenco utenti l'utente registrato

Test Amministratore

- **Test Aggiorna Punti Utente**: verifica che, una volta invocato il metoto *AggiornaPuntiUtente* i punti dell'utente subiscono effettivamente una modifica

Test Facoltà

- **Test Trova Materia**: verifica che una volta invocato il metodo *TrovaMateria* la materia ritornata non sia “NULL” (se la materia non esiste viene creata)
- **Test Selezione Materia**: verifica che se sono inseriti appunti, la lista di appunti ritornata non sia “NULL”

Test Utente

- **Test Inserisci Appunto:** verifica che una volta creato un nuovo appunto, quest'ultimo sia presente nell'elenco degli appunti in sospeso
- **Test Acquista Appunto:** verifica che una volta acquistato una appunto, viene create la nuova istanza di acquisto
- **Test Inserisci Recensione:** verifica che inserita una recensione, venga create l'istanza di recensione

Test Materia

- **Test Seleziona Appunto:** verifica che se inserito il codice di un appunto esistente nell'elenco di appunti approvati, quest'ultimo venga ritornato

Test Gestione Pagamento

- **Test Paga con Punti:** verifica che una volta effettuato l'acquisto con I punti, quelli dell'utente siano effettivamente diminuiti di 5
- **Test Paga con Carta:** verifica che se la carta non è valida l'acquisto fallisce

Test Login

I test seguenti verificano il corretto inserimento delle credenziali:

- **Test Login Admin Password**
- **Test Login Admin E-mail**
- **Test Login User Password**
- **Test Login User Nickname**