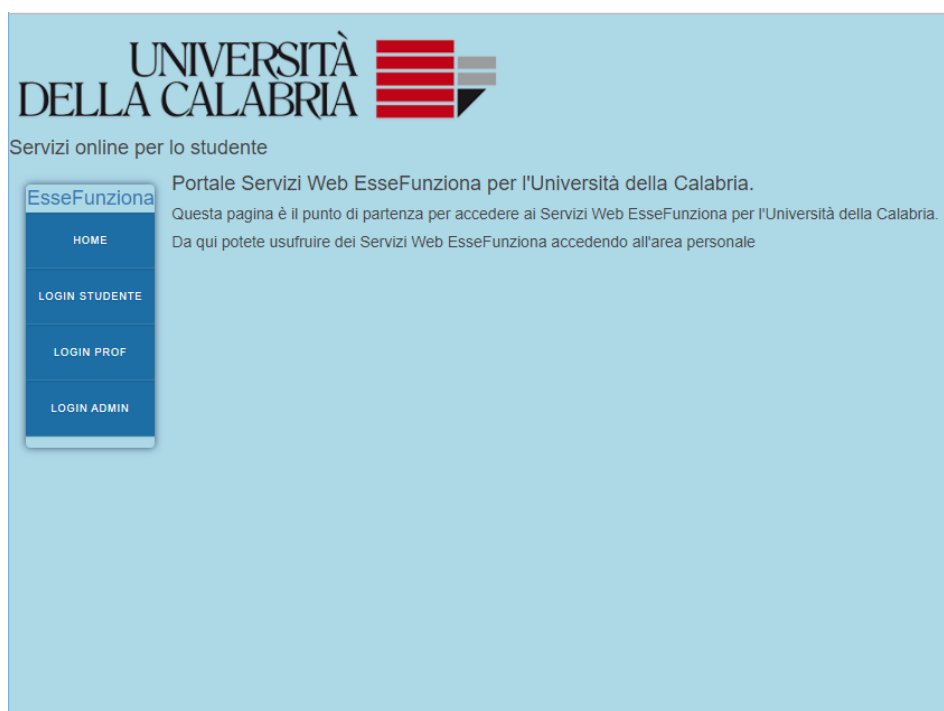




## PROGETTO INGSW-SIW 2017/18

### ESSE-Funziona

Di  
Quarta Luca - Mattia Cava



# Sommario

ESSEFUNZIONA - DESCRIZIONE .....	3
ANALISI DEI REQUISITI INFORMALE .....	3
MODELLO DI DOMINIO .....	5
ARCHITETTURA DI SISTEMA .....	6
CASI D'USO .....	6
SCHEDE CASI D'USO .....	7
REGISTRARE UN UTENTE .....	7
GESTIRE ESAME .....	10
MODIFICARE IL PIANO DI STUDI .....	16
GESTIRE TASSE .....	20
AGGIUNGERE MATERIALE .....	23
CREARE UN RICEVIMENTO .....	27
GESTIRE CORSI DI LAUREA .....	30
CLASSI CONCETTUALI .....	35
APPENDICE .....	36

## ESSEFUNZIONA - DESCRIZIONE

Esse Funziona è una piattaforma Web che ripropone i servizi offerti dalla piattaforma Esse3 dell'Università della Calabria.

La piattaforma permette la registrazione di tutti gli studenti e di tutti i professori dell'università e gli dà la possibilità di usufruire dei servizi loro dedicati. In questa piattaforma sono presenti corsi e corsi di laurea dell'università della Calabria. Sono accessibili bandi e materiali dei corsi e le tasse che gli studenti devono pagare.

Gli studenti possono per esempio prenotarsi agli appelli degli esami, pagare le tasse universitarie, modificare il proprio piano di studi, chiedere ricevimenti ai professori, accettare o rifiutare gli esiti degli esami sostenuti, visualizzare la propria carriera e i propri dati.

I professori invece possono creare nuovi appelli per i corsi da loro insegnati, aggiungere gli esiti degli esami per gli studenti prenotati all'appello selezionato, rispondere alle richieste dei ricevimenti, aggiungere documenti relativi ai corsi.

## ANALISI DEI REQUISITI INFORMALE

Il sistema è un sistema di gestione dei servizi web dell'università.

Un qualsiasi utente per accedere al proprio account immette la propria matricola e la propria password.

L'Utente può essere:

- Studente
- Professore
- Admin (gestore sistema)

Il sistema contiene i vari corsi di laurea e dipartimenti a cui appartengono.

Il sistema permette all'utente admin di registrare studenti e professori.

L'admin può registrare corsi di laurea e corsi, pubblicare bandi e caricare le tasse per gli studenti.

Uno studente ha la possibilità di prenotarsi agli appelli d'esame disponibili.

Un professore può registrare l'esito degli esami per ogni studente.

Il sistema contiene una pagina per la visualizzazione della carriera dello studente, il piano di studi per il quale dà la possibilità di effettuare delle modifiche.

Il sistema permette l'aggiunta di appelli d'esame.

Il sistema consente la visione delle tasse.

Il sistema contiene una pagina per la visualizzazione di bandi e news e una per la visione dei documenti relativi ai corsi.

Si possono visualizzare varie informazioni e corsi di studi specifici.

Il sistema dà uno spazio dove caricare e/o scaricare documentazioni per ogni corso.

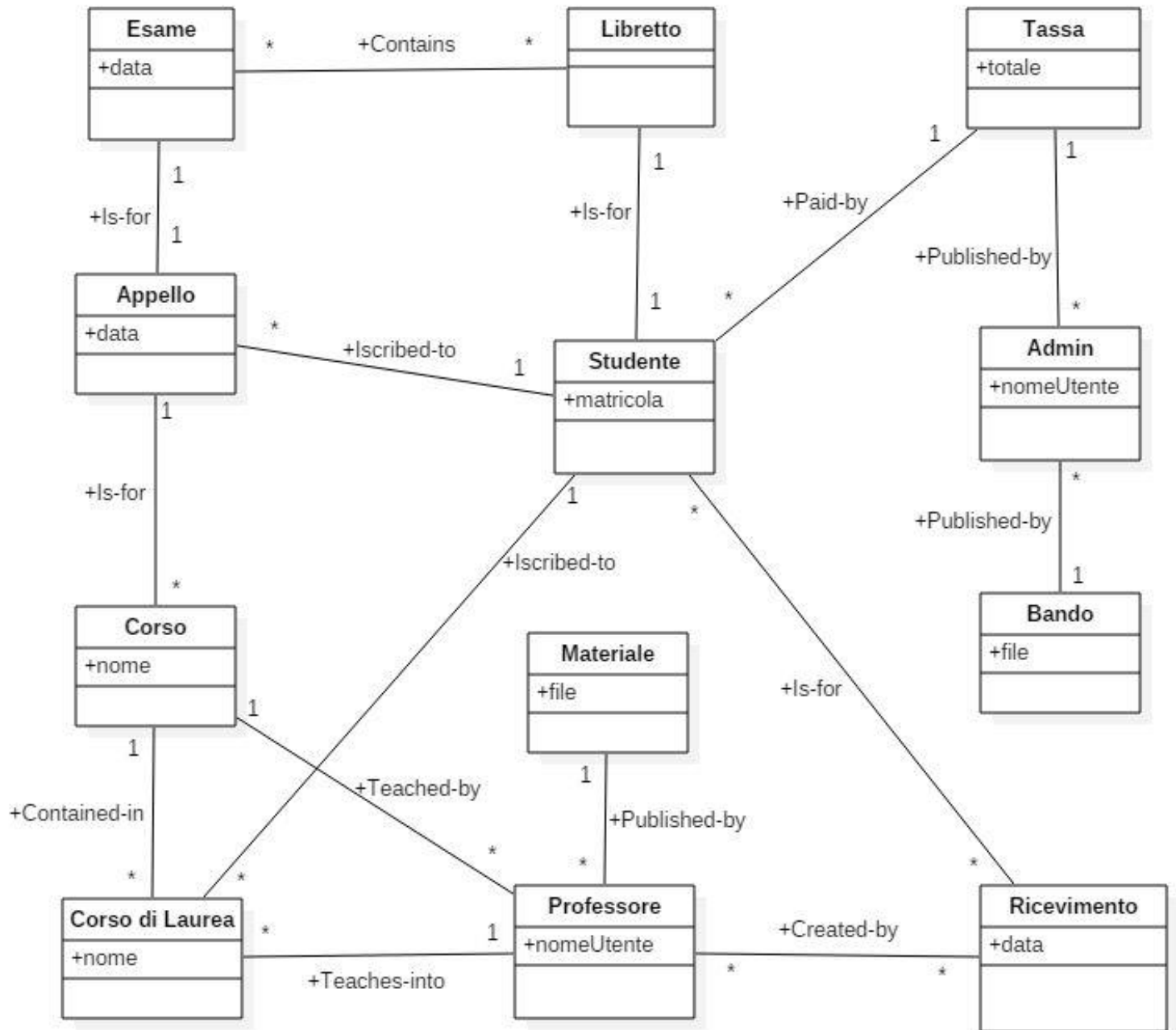
Il sistema fornisce un servizio che permette di richiedere ricevimento ai professori.

Il sistema permette di modificare il corso di laurea aggiungendo o rimuovendo corsi e cambiando il nome.

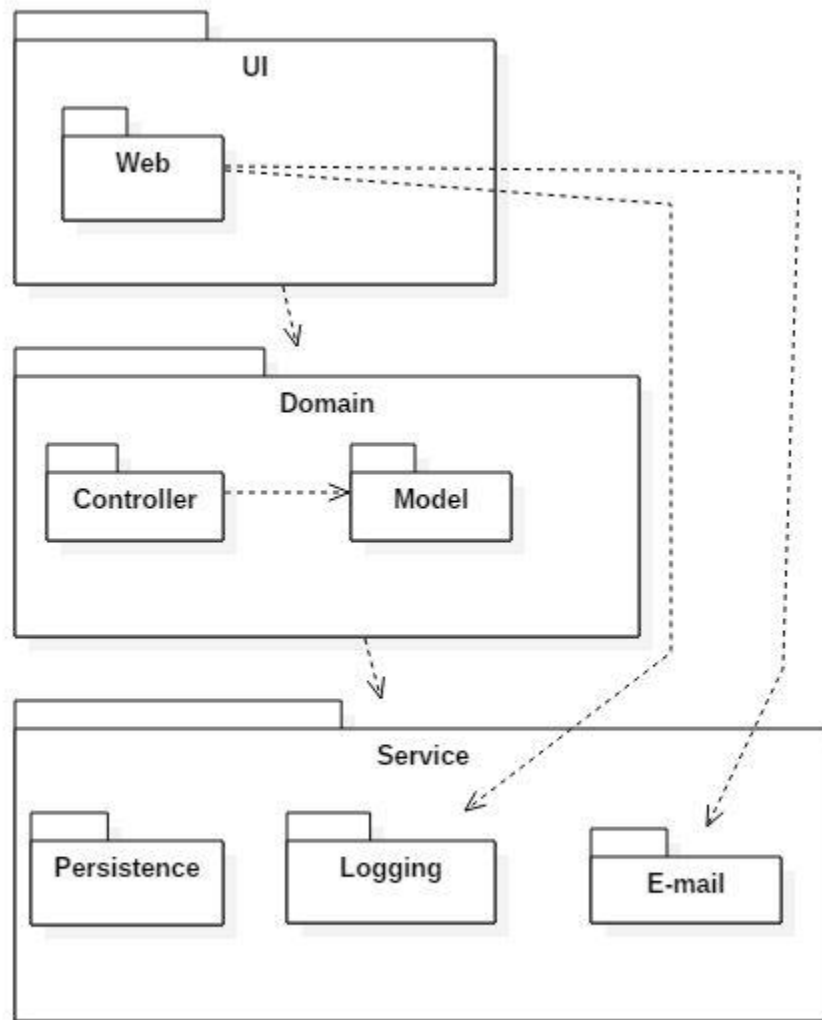
Lo studente può vedere gli esiti degli esami a cui si è prenotato e scegliere se accettare o meno.

Il Sistema utilizza anche Web API esterne quali Google Maps e Facebook.

# MODELLO DI DOMINIO



## ARCHITETTURA DI SISTEMA



## CASI D'USO

- Registrare un utente
- Gestire Esame
- Modificare il Piano Di Studi
- Gestire Tasse
- Aggiungere Materiale
- Creare un Ricevimento
- Gestire Corsi Di Laurea

## SCHEDA CASI D'USO

### REGISTRARE UN UTENTE

**Nome del caso d'uso:** Registrare un Utente

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Admin

#### **Parti interessate e interessi**

- Studente: richiede di registrarsi al sistema
- Professore: richiede di registrarsi al sistema
- Admin: richiede la possibilità di registrare un utente che può essere sia uno studente che un professore
- Università: vuole un sistema per registrare nel sito tutti gli studenti e i professori che fanno parte dell'università

#### **Pre-condizioni**

- Lo studente deve essere iscritto all'università.
- Il professore deve insegnare nella università.

#### **Garanzia di successo**

La registrazione è completata. L'account dell'utente viene creato.

#### **Scenario principale di successo**

1. L'admin apre il menu per registrare gli utenti
2. L'admin sceglie di registrare un professore
3. L'admin accede all'area per registrare un professore
4. L'admin compila tutti i campi della registrazione
5. L'admin salva e invia i dati
6. Il sistema notifica il caricamento del professore
7. Il sistema invia una mail di notifica al professore

8. Il professore riceve la notifica della registrazione

## **Estensioni**

\*1-5 L'admin può annullare il processo di registrazione in qualsiasi momento

2a. L'admin sceglie di registrare uno studente

1. L'admin accede all'area per registrare uno studente
2. L'admin compila tutti i campi della registrazione
3. L'admin salva e invia i dati
4. Il sistema notifica il caricamento dello studente
5. Il sistema invia una mail di notifica allo studente
6. Lo studente riceve la notifica della registrazione

5a. L'admin cancella tutti i campi

1. L'admin ricompila tutti i campi della registrazione
2. Il caso d'uso riparte dal passo 5

5b. L'admin annulla la registrazione

1. Il sistema annulla la creazione dell'utente
2. Il caso d'uso termina

6c. Il sistema fallisce il salvataggio/invio

1. Il sistema non notifica il caricamento del nuovo utente
2. L'admin ricompila tutti i campi per la registrazione
3. L'admin salva e invia i dati
4. Il sistema notifica il caricamento dell'utente
5. Il sistema invia l'e-mail di notifica all'utente
6. L'utente riceve la notifica della registrazione

6d. Il sistema trova un errore di compilazione di uno o più campi

1. Il sistema si comporta come nel caso 6c

## **Requisiti speciali**

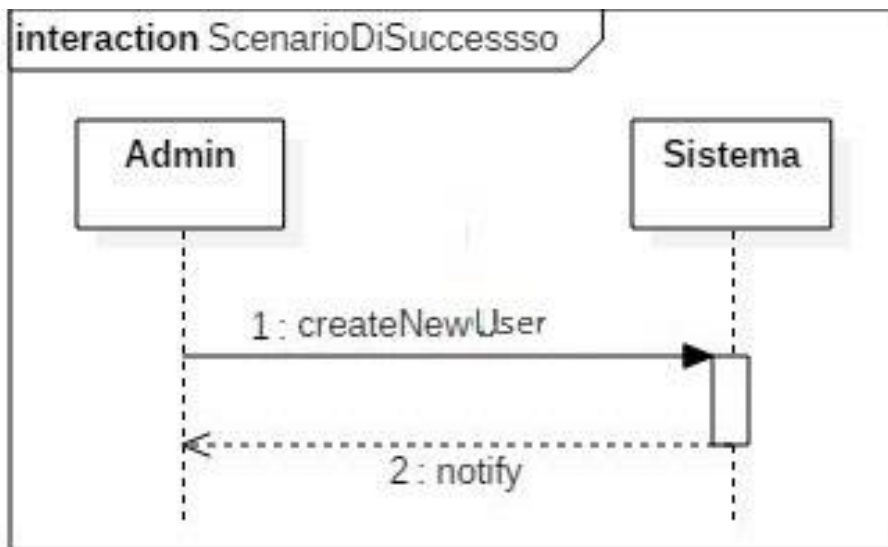
Interfaccia web responsive

## **Frequenza di ripetizione**

Quasi ininterrotta



## Diagramma di sequenza



## Contratti

**Operazione:** createUser()

**Riferimenti:** casi d'uso: Registrare un utente

**Pre-condizioni:** è in corso una registrazione

**Post-condizioni:**

1. È stata creata un'istanza u di utente

## **GESTIRE ESAME**

**Nome del caso d'uso:** Gestire Esame

**Portata:** Piattaforma Web EsseFunziona

**Livello** Obiettivo utente

**Attore primario:** Professore, Studente

### **Parti interessate e interessi**

- Professore: desidera caricare un appello per un esame e caricare l'esito per ogni studente iscritto dopo lo svolgimento di esso
- Studente: desidera prenotarsi ad un appello e poter decidere se accettare o meno l'esito dell'esame ed in caso di conferma del voto poter vedere l'esame caricato sul proprio libretto.
- Università: vuole un sistema per gestire gli esami dalla creazione al caricamento nel libretto degli studenti

### **Pre-condizioni**

- Il professore deve essere identificato e autenticato nel sistema
- Il professore deve avere tra i suoi corsi l'esame per il quale vuole caricare un appello
- Lo studente deve essere identificato e autenticato nel sistema
- Lo studente deve avere nel suo piano di studi l'esame che vuole sostenere

### **Garanzia di successo**

L'esame deve essere presente nel libretto dello studente che lo ha sostenuto con relativa votazione e data di superamento, lo studente deve aver ricevuto una e-mail di conferma del caricamento dell'esame

### **Scenario principale di successo**

1. Il professore attraverso il menu accede all'area per la creazione di nuovi appelli
2. Il professore sceglie l'esame per cui caricare il nuovo appello
3. Il professore sceglie la data per l'appello
4. Il professore salva e invia i dati
5. Il sistema carica l'appello
6. Il sistema notifica il caricamento
7. Il professore riceve la notifica

1-7 Il professore ripete il processo per ogni esame per il quale vuole caricare un appello

8. Lo studente attraverso il menu accede all'area prenotazione esame
9. Lo studente seleziona il corso per il quale ha intenzione di prenotarsi
10. Lo studente sceglie l'appello a cui prenotarsi
11. Lo studente invia i dati al sistema
12. Il sistema effettua la prenotazione e la carica nelle prenotazioni
13. Il sistema notifica la prenotazione allo studente
14. Lo studente riceve la notifica

8-14 Lo studente ripete il processo per ogni appello a cui vuole prenotarsi

15. Il professore tramite il menu accede all'area di registrazione esame
16. Il professore seleziona l'esame da registrare
17. Il professore seleziona lo studente
18. Il professore inserisce il voto
19. Il professore salva e invia i dati
20. Il sistema carica l'esito
21. Il sistema notifica al professore ed allo studente l'avvenuta registrazione
22. Il professore riceve la notifica

15-22 Il professore ripete il processo per ogni studente che ha partecipato all'esame

23. Lo studente riceve la notifica

24. Lo studente accede dal menu alla bacheca dei risultati esami

25. Lo studente seleziona l'esame da verbalizzare

26. Lo studente accetta il risultato dell'esame

27. Il sistema carica l'esame nel libretto dello studente

28. Il sistema notifica allo studente la verbalizzazione

29. Lo studente riceve la notifica

23-29 Lo studente ripete il processo per ogni esame che deve verbalizzare

### **Estensioni**

\*1-7 in qualsiasi momento il professore può decidere di non continuare e il caso d'uso riparte dal passo 8

\*8-14 Lo studente in qualsiasi momento può non prenotarsi ad alcun appello il caso d'uso riparte dal passo 15

\*15-22 in qualsiasi momento il professore può decidere di non effettuare registrazione di alcun esame e il caso d'uso riparte dal passo 23

\*23-29 Lo studente in qualsiasi momento può non verbalizzare alcun esame e il caso d'uso termina

4a. Il professore decide di non completare la creazione del nuovo appello

1. Il caso d'uso riparte dal passo 2

4b. Il sistema riscontra un errore nel salvataggio/invio

1. Il sistema non carica l'appello

2. Il sistema non notifica il caricamento

3. Il professore riavvia la procedura di caricamento nuovo appello

4. Il caso d'uso riparte dal passo 2

5a. Il sistema non riesce a caricare un nuovo appello

1. Il sistema non notifica il caricamento

## 2. Il caso d'uso riparte dal passo 1

### 12a. Il sistema riscontra errori nella prenotazione

1. Il sistema non effettua la prenotazione
2. Lo sistema non notifica il la prenotazione
3. Il caso d'uso riparte dal passo 9

### 12b. Il sistema non riesce ad effettuare la prenotazione

1. Lo sistema non notifica il la prenotazione
2. Il caso d'uso riparte dal passo 9

### 12c. Lo studente è già prenotato all'appello

1. Il sistema non effettua la prenotazione
2. Il sistema non notifica la prenotazione
3. Il caso d'uso riparte dal passo 9

### 12d. Lo studente ha già sostenuto l'esame

3. Il sistema non effettua la prenotazione
4. Il sistema non notifica la prenotazione
5. Il caso d'uso riparte dal passo 9

### 20a. Il sistema riscontra errori nei dati

1. Il sistema non carica l'esito
2. Il sistema non notifica il caricamento
3. Il caso d'uso riparte dal passo 2.

### 23a. Lo studente non riceve alcuna notifica

1. Il caso d'uso termina

### 26a. Lo studente rifiuta il risultato dell'esame

1. Il sistema invia un'e-mail al professore che conferma il rifiuto dello studente
2. L'esito viene eliminato
3. Il caso d'uso termina

### 27a. Il sistema non riesce a caricare l'esame

1. Lo sistema non notifica il caricamento
2. Il caso d'uso termina

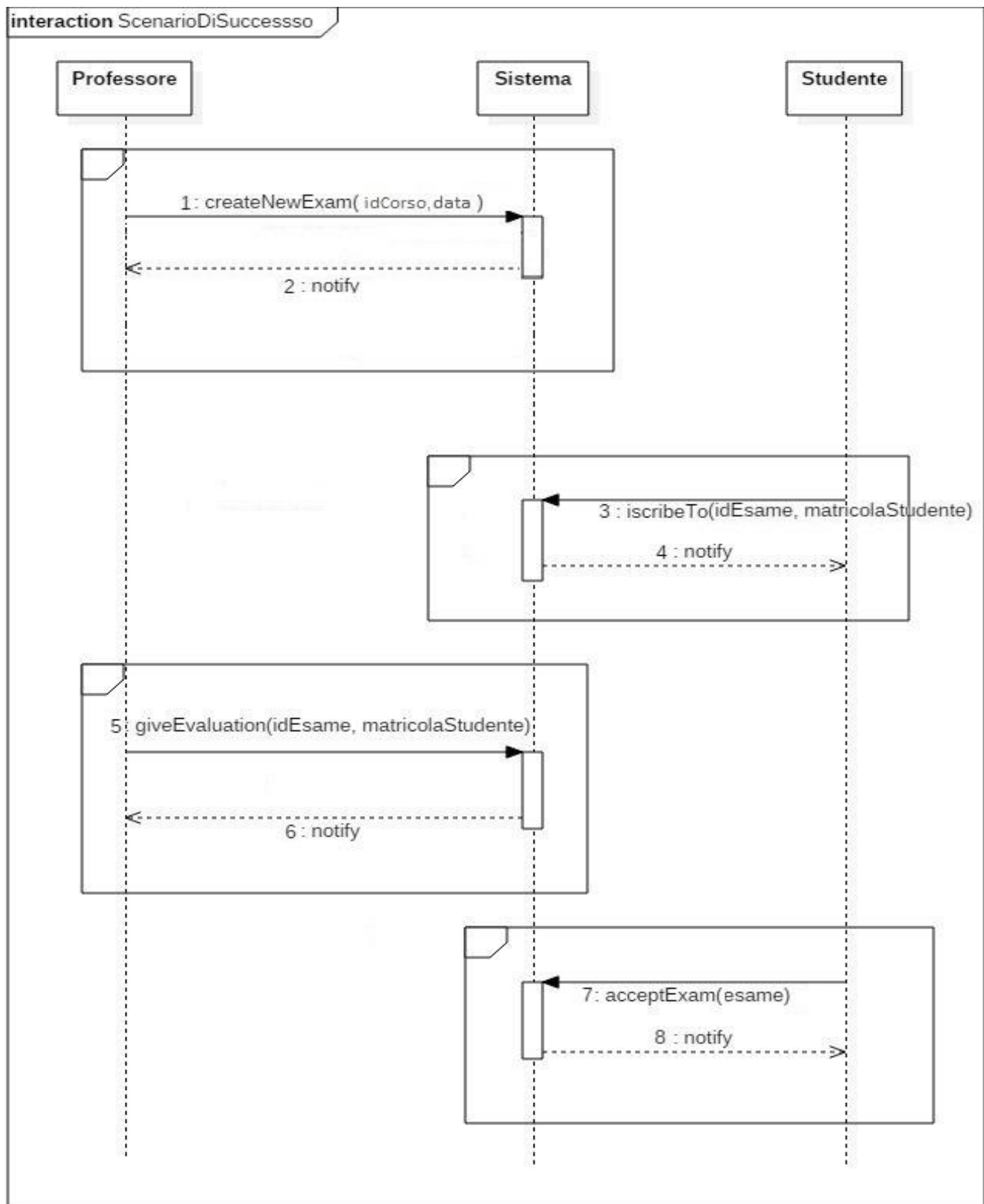
## Requisiti speciali

Interfaccia web responsive

## Frequenza di ripetizione

Quasi ininterrotta durante la sessione d'esame

## Diagramma di sequenza



## Contratti

**Operazione:** createNewExam(idCorso: integer, data:Date)

**Riferimenti:** casi d'uso: Gestire Esame

**Pre-condizioni:** Deve esistere il corso per il quale creare un esame

**Post-condizioni:**

1. È stata creata un'istanza a di Appello
2. a è stata associata con il Corso corrente
3. a.idCorso è diventata idCorso
4. a.data è diventato data

**Operazione:**iscibeTo(idAppello: integer, matricolaStudiante: String)

**Riferimenti:** casi d'uso: Gestire Esame

**Pre-condizioni:** Deve esistere un'istanza a di Appello e un'istanza s di Studente

**Post-condizioni:**

1. È stata creata un'istanza e di Esame
2. e è stata associata con l'Appello corrente
3. e.matricolaStudiante è diventata matricolaStudiante

**Operazione:**giveEvaluation(idAppello: integer, matricolaStudiante: String, voto: integer)

**Riferimenti:** casi d'uso: Gestire Esame

**Pre-condizioni:** Deve esistere un'istanza a di Appello e un'istanza s di Studente

**Post-condizioni:**

1. e.voto è diventata voto

**Operazione:**accept(esame:Esame)

**Riferimenti:** casi d'uso: Gestire Esame

**Pre-condizioni:** Deve esistere un'istanza di Esame

**Post-condizioni:**

1. È stata creata un'istanza I di libretto
2. I.idAppello è diventato esame.idAppello
3. I.matricolaStudente è diventata esame.matricolaStudente
4. I.voto è diventata esame.voto

## MODIFICARE IL PIANO DI STUDI

**Nome del caso d'uso:** Modificare piano di studi

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Studente, admin

### **Parti interessate e interessi**

- Studente: Vuole un sistema semplice ed intuitivo per modificare il piano di studi quando è possibile e la possibilità entro breve tempo di rimediare ad un errore di commesso in fase di compilazione del modulo.
- Università: Vuole registrare il piano di studi completo dello studente.
- Admin: Vuole visionare le scelte dello studente e approvarle o rifiutarle.
- Università: vuole un sistema che permetta di modificarne i corsi dei piani di studi degli studenti

### **Pre-condizioni**

- Lo studente deve essere iscritto all'università.
- Lo studente deve essere identificato e autenticato nel sistema.
- L'università ha permesso allo studente di modificare il piano di studi.



- L'admin deve essere identificato e autenticato nel sistema.

## **Garanzia di successo**

Il piano di studi viene modificato con successo ed approvato dalla commissione. Le informazioni aggiornate all'ultima modifica vengono salvate. Viene inviata allo studente una e-mail che conferma l'avvenuta modifica.

## **Scenario principale di successo**

1. Lo studente tramite il menu accede all'area per la modifica del piano di studi
2. Lo studente visualizzerà ciò che può essere modificato del suo piano di studi
3. Lo studente decide se rimuovere o aggiungere corsi nel suo piano
4. Lo studente decide di aggiungere un esame seleziona il corso nell'apposita sezione

Lo studente ripete i passi 3 e 4 fino a quando non ha finito

5. Lo studente decide di inviare il suo piano alla commissione
6. Il sistema chiede una conferma per l'invio del piano di studi
7. Lo studente conferma
8. Il sistema invia il piano di studi alla commissione
9. Il sistema invia una e-mail allo studente che conferma l'invio con successo del piano di studi all'admin
10. L'admin accetta il piano di studi proposto dallo studente
11. Allo studente viene notificata la decisione dell'admin.

## **Estensioni**

- 4a. Lo studente decide di rimuovere un corso dal suo piano
  1. Lo studente seleziona il corso da rimuovere
    - a. L'esame scelto è obbligatorio e lo studente non può rimuoverlo
    - b. L'esame scelto viene rimosso con successo

7a. Lo studente si accorge di un errore o di una dimenticanza e decide di non confermare l'invio

1. Il caso d'uso riparte dal passo 4

8a. A causa di qualche inconveniente il sistema non riesce ad inviare il piano di studi alla commissione

1. Appare un messaggio di errore e viene inviata una e-mail allo studente, che notifica l'errato invio del piano di studi
2. Il caso d'uso riparte dal passo 5

10a. L'admin rifiuta il piano di studi proposto dallo studente

1. L'admin informa lo studente e lo invita a ricompilare il piano di studi
2. Il caso d'uso riparte dal passo 1

\*1-6 in qualsiasi momento durante la modifica del piano di studi lo studente può decidere di non continuare e rimodificare da capo un'altra volta

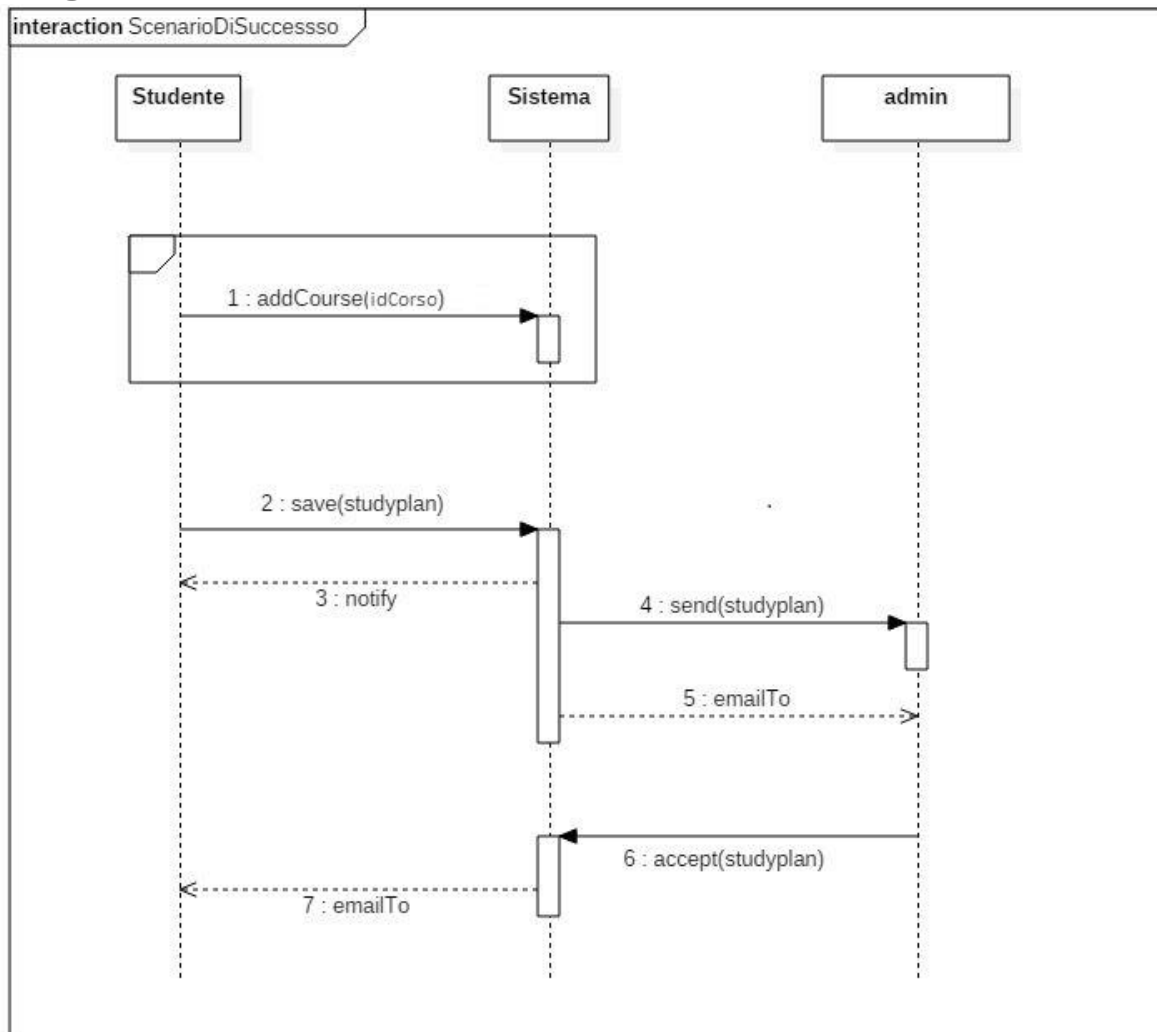
## **Requisiti speciali**

Interfaccia web responsive

## **Frequenza di ripetizione**

Numero di ripetizioni consentiti dalla commissione

## Diagramma di sequenza



## Contratti

**Operazione:** addCourse(idCourse:integer)

**Riferimenti:** casi d'uso: modificare piano di studi

**Pre-condizioni:** istanza p di piano di studi già esistente

**Post-condizioni:**

- È stata creata un'istanza c di Corso(creazione istanza)
- s viene inizializzata(modifica attributo)
- s.id è diventata idCourse(modifica attributo)
- s è stata associata a p(associazione formata)

**Operazione:** push(studyPlan:PianoDiStudi)

**Riferimenti:** casi d'uso: modificare piano di studi

**Pre-condizioni:** istanza p di piano di studi e a di admin già esistenti, è incorso la modifica del piano di studi

**Post-condizioni:**

1. p è stata associata a a(associazione formata)

**Operazione:** accept(studyPlan:PianoDiStudi)

**Riferimenti:** casi d'uso: modificare piano di studi

**Pre-condizioni:** istanza p di piano di studi e a di admin già esistenti, è incorso la modifica del piano di studi

**Post-condizioni:**

1. s.accepted è diventata vera(modifica attributo)

## GESTIRE TASSE

**Nome del caso d'uso:** Gestire Tasse

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Admin

**Parti interessate e interessi**

- Admin: Vuole caricare e pubblicare le tasse per gli studenti
- Studente: Vuole visionare l'importo delle tasse
- Università: Vuole un sistema per la gestione delle tasse universitarie

**Pre-condizioni**

- L' Admin deve essere registrato ed identificato

## **Garanzia di successo**

La Tassa viene caricata con successo sulla piattaforma per ogni studente ed è accessibile.

## **Scenario principale di successo**

1. L' Admin entra nell'area apposita per gestire le tasse
2. L' Admin decide di creare una nuova tassa
3. L' Admin determina il relativo importo
4. L' Admin scrive una descrizione della tassa
5. L' Admin pubblica la tassa
6. Il sistema carica la tassa
7. Il sistema notifica il caricamento la tassa

Admin ripete i passi 2-7 fino a quando ha finito le tasse da aggiungere

8. Ogni studente paga l'ammontare della tassa
9. L'admin accede all'area per modificare le tasse
10. L'admin segna come pagate le tasse che lo studente ha pagato
11. L'admin salva

## **Estensioni**

1-5 in qualsiasi momento l'admin può annullare il processo di creazione tassa

6a. Il sistema riscontra errori nei campi

1. Il sistema non carica la tassa
2. Il caso d'uso riparte dal passo 2

6b. Il sistema non riesce a caricare la tassa

1. Il sistema non carica la tassa
2. Il caso d'uso riparte dal passo 2

9a. Nessuno studente ha pagato delle tasse

1. Il caso d'uso termina

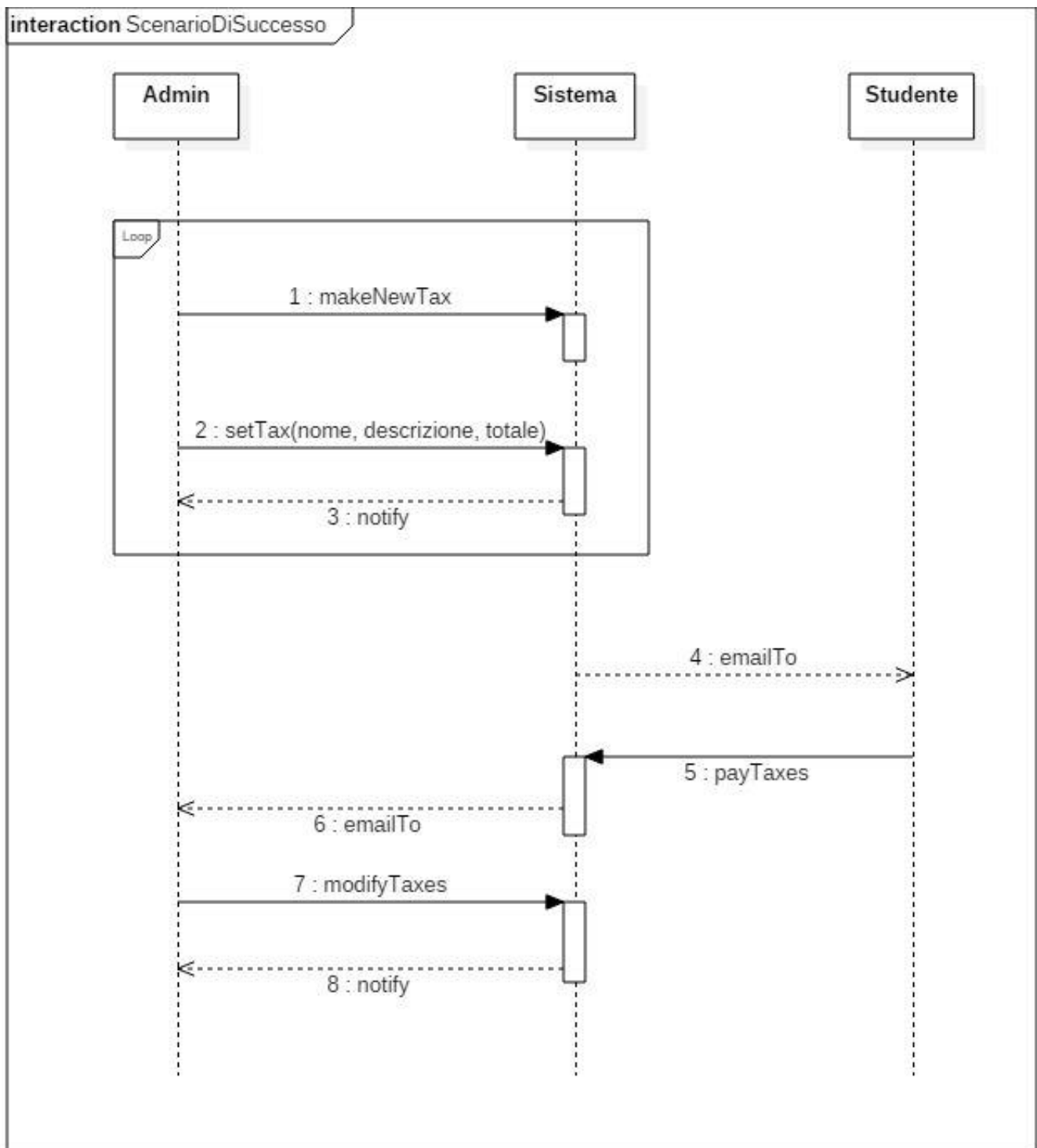
## Requisiti speciali

Interfaccia web responsive

## Frequenza di ripetizione

Anche ininterrotta

## Diagramma di sequenza



## Contratti

**Operazione:**makeNewTax()

**Riferimenti:** caso d'uso: Gestire Tasse

**Pre-condizioni:** nessuna

**Post-condizioni:**

1. È stata creata un'istanza t di Tassa

**Operazione:**setTax(nome: string, descrizione: string, totale: int)

**Riferimenti:** caso d'uso: Gestire Tasse

**Pre-condizioni:** deve esistere un'istanza t di Tassa, è in corso la creazione di una tassa

**Post-condizioni:**

1. t.nome è diventato nome
2. t.descrizione è diventata descrizione
3. t.totale è diventato totale
4. t è stata associata con Studente

## AGGIUNGERE MATERIALE

**Nome del caso d'uso:** Aggiungere Materiale

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Admin, Professore

**Parti interessate e interessi**

- Admin: vuole pubblicare un bando sulla piattaforma

- Professore: vuole caricare il materiale dei suoi corsi sulla piattaforma
- Studente: vuole accedere al materiale e scaricarlo
- Università: vuole un sistema per gestire il caricamento di materiali rendendoli disponibili agli studenti

### **Pre-condizioni**

- L' Admin deve essere registrato ed identificato
- Lo studente deve essere identificato e autenticato nel sistema.
- Il professore deve essere identificato e autenticato nel sistema.

### **Garanzia di successo**

Il bando viene caricato con successo sulla piattaforma ed è accessibile su una apposita pagina, il materiale viene caricato con successo sulla piattaforma ed è accessibile su una apposita pagina

### **Scenario principale di successo**

1. L'admin accede all'area bandi
2. L'admin avvia la procedura per il caricamento di un nuovo bando
3. L'admin carica il file del bando
4. Il L'admin conferma la pubblicazione
5. Il sistema crea e carica il bando
6. Il sistema notifica la creazione

1-6 L'admin ripete il processo per ogni bando che vuole pubblicare

7. Il professore accede all'area materiale
8. Il professore avvia la procedura per il caricamento di un nuovo bando
9. Il professore carica il file del materiale
10. Il professore conferma la pubblicazione
11. Il sistema crea e carica il materiale
12. Il sistema notifica il caricamento del materiale



6-12 Il professore ripete il processo per ogni materiale che vuole pubblicare

## **Estensioni**

1-6\* L'admin in qualsiasi momento decide di non pubblicare bandi e si riparte dal passo 7

3a. L'admin non trova il file del bando

1. L'admin annulla la pubblicazione
2. Il caso d'uso riparte dal passo 2

5a. Il sistema non riesce a caricare il bando sulla piattaforma

1. Il sistema non notifica il caricamento del bando
2. Il caso d'uso riparte dal passo 2

5b. il sistema riscontra errori nel file

1. Il sistema non notifica il caricamento del bando
2. Il caso d'uso riparte dal passo 2

7-12\* Il professore decide di non pubblicare alcun materiale e il caso d'uso termina

8a. Il professore non trova il file del materiale

1. Il professore annulla la pubblicazione
2. Il caso d'uso riparte dal passo 8

10a. Il sistema non riesce a caricare il materiale sulla piattaforma

1. Il caso d'uso riparte dal passo 8

10b. il sistema riscontra errori nel file

1. Il sistema non notifica il caricamento del bando
2. Il caso d'uso riparte dal passo 8

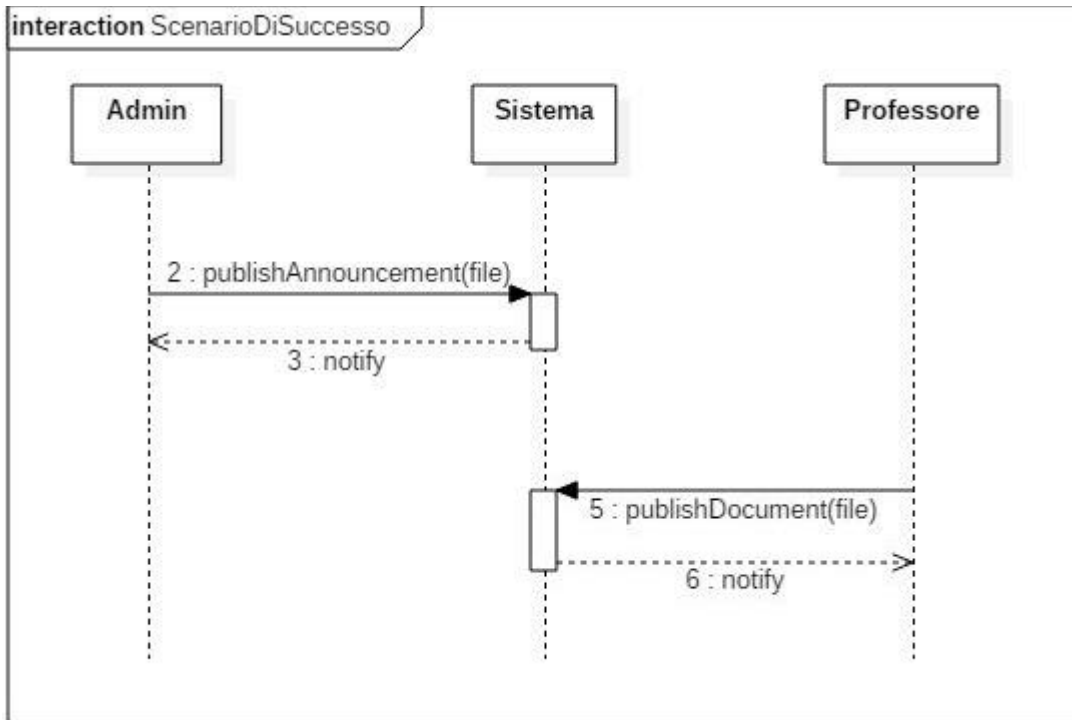
## **Requisiti speciali**

Interfaccia web responsive

## Frequenza di ripetizione

Anche ininterrotta

## Diagramma di sequenza



## Contratti

**Operazione:**publishAnnouncement(file: File)

**Riferimenti:** casi d'uso: Aggiungere Materiale

**Pre-condizioni:** è in corso la creazione di un Materiale o di un Bando

**Post-condizioni:**

1. È stata creata un'istanza b di Bando
2. B.file è diventato file

**Operazione:**publishDocument(file: File)

**Riferimenti:** casi d'uso: Aggiungere Materiale

**Pre-condizioni:** è in corso la creazione di un Materiale o di un Bando

**Post-condizioni:**

1. È stata creata un'istanza m di Materiale
2. m.file è diventato file

## CREARE UN RICEVIMENTO

**Nome del caso d'uso:** Creare un Ricevimento

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Studente

### **Parti interessate e interessi**

- Studente: vuole richiedere un ricevimento al professore per discutere di alcuni argomenti
- Professore: vuole vedere tutte le richieste ricevute per dare loro successivamente risposta
- Università: vuole un sistema per gestire i ricevimenti tra studenti e professori

### **Pre-condizioni**

- Lo studente deve essere identificato e autenticato nel sistema.
- Il professore deve essere identificato e autenticato nel sistema.
- Lo studente deve essere iscritto ad un corso di laurea dove è presente almeno un corso del professore

### **Garanzia di successo**

La richiesta è stata inviata correttamente al professore che successivamente potrà creare o meno il ricevimento che poi sarà visibile tra le richieste

### **Scenario principale di successo**

1. Lo studente accede all'area relativa alla richiesta di ricevimenti
2. Lo studente avvia la procedura per inviare la richiesta di ricevimento
3. Lo studente seleziona il professore tra quelli disponibili

4. Lo studente invia la richiesta
5. Il sistema notifica il corretto invio della richiesta
6. Il sistema invia al una e-mail professore

1-6 Lo studente ripete i passi per tutti i professori a cui vuole chiedere ricevimento

7. Il professore accede all'area relativa alle richieste di ricevimento
8. Il professore seleziona lo studente per il quale creare un ricevimento
9. Il professore seleziona la data del ricevimento
10. Il professore conferma
11. Il sistema notifica la creazione del ricevimento
12. Il sistema invia una e-mail allo studente per informarlo della creazione del ricevimento

7-12 Il professore ripete i passi per tutte le richieste di ricevimento pendenti

## **Estensioni**

1-4\* Lo studente può annullare il processo in qualsiasi momento ed il caso d'uso riparte dal passo 7

6-9\* Il professore può annullare il processo in qualsiasi momento

3a. Lo studente non trova il professore desiderato

1. Lo studente annulla il processo
2. Il caso d'uso riparte dal passo 7

4a. Lo studente decide non inviare più la richiesta

1. Il caso d'uso riparte dal passo 7

5a. Il sistema non riesce ad inviare la richiesta

1. Il sistema non notifica il caricamento della richiesta
2. Il caso d'uso riparte dal passo 2

8a. Il professore non trova nessuna richiesta e annulla il processo

1. Il caso d'uso termina

11a. Il sistema riscontra degli errori nei campi

1. Il sistema non carica il ricevimento

2. Si riparte dal passo 7

11b. Il sistema non riesce a caricare il ricevimento

1. Il sistema non carica il ricevimento

2. Si riparte dal passo 7

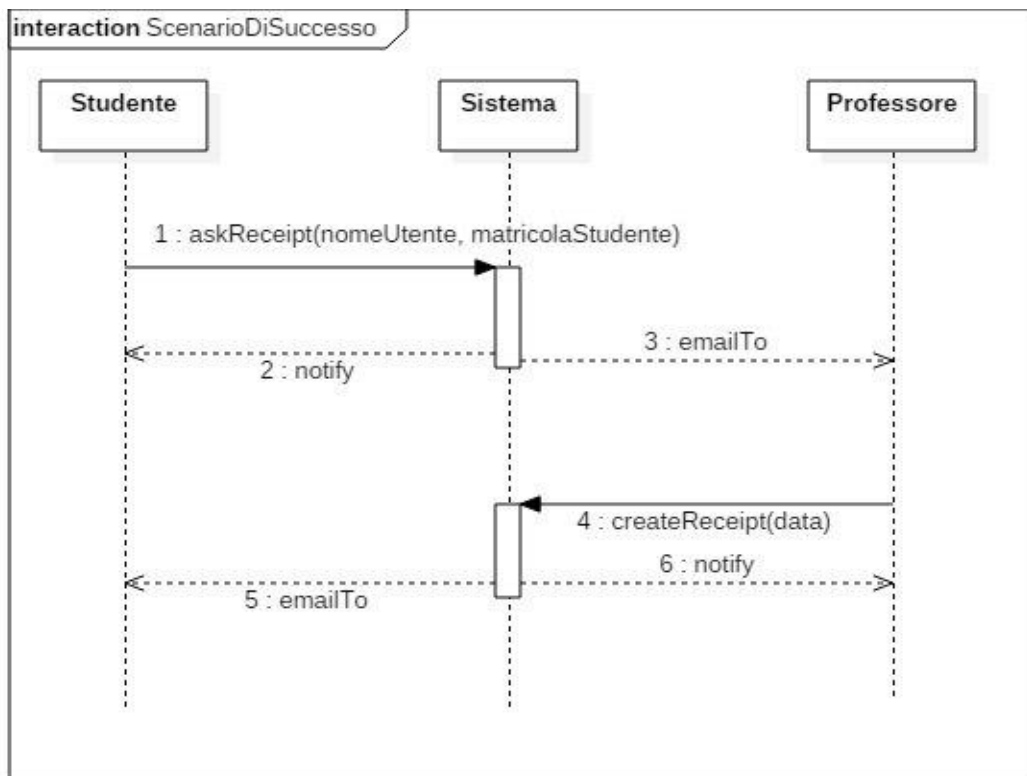
## Requisiti speciali

Interfaccia web responsive

## Frequenza di ripetizione

Anche ininterrotta

## Diagramma di sequenza



## Contratti

**Operazione:**askReceipt(nomeUtente : string ,matricola : string)

**Riferimenti:** caso d'uso: Creare Ricevimento

**Pre-condizioni:** è in corso la creazione di un ricevimento

**Post-condizioni:**

1. È stata creata un'istanza r di Riceve
2. r.nomeUtente è diventato nomeUtente
3. r.matricola è diventata matricola

**Operazione:**askReceipt(data)

**Riferimenti:** caso d'uso: Creare Ricevimento

**Pre-condizioni:** è in corso la creazione di un ricevimento ed esiste un'istanza r di Riceve

**Post-condizioni:**

1. r.data è diventata data

## GESTIRE CORSI DI LAUREA

**Nome del caso d'uso:** Gestire Corsi Di Laurea

**Portata:** Piattaforma Web EsseFunziona

**Livello:** Obiettivo utente

**Attore primario:** Admin

**Parti interessate e interessi**

- Admin: vuole creare un corso di laurea e gestire i corsi che lo riguardano.
- Professore: vuole poter vedere i corsi di laurea dell'università e i corsi di laurea che lo riguardano
- Studente: vuole poter vedere i corsi di laurea dell'università e i corsi di laurea che lo riguardano

- Università: vuole un sistema per gestire tutti i corsi di laurea dell'università e che permetta di modificarne i corsi

## **Pre-condizioni**

- L'admin deve essere registrato e autenticato nel sistema

## **Garanzia di successo**

Il corso di laurea deve essere caricato nel sistema e visibile nella pagina dei corsi di laurea

## **Scenario principale di successo**

1. L'admin accede all'area per aggiungere un corso
2. L'admin avvia la procedura per caricare un nuovo corso
3. L'admin dà un nome al corso
4. L'admin conferma la creazione del corso
5. Il sistema notifica l'effettuata creazione del corso

1-5 L'admin ripete il processo per tutti i corsi che intende creare

6. L'admin accede all'area per aggiungere un corso di laurea
7. L'admin avvia la procedura per caricare un nuovo corso di laurea
8. L'admin dà un nome al corso di laurea
9. L'admin conferma la creazione del corso di laurea
10. Il sistema notifica l'effettuata creazione del corso di laurea

6-10 L'admin ripete il processo per tutti i corsi di laurea che intende creare

11. L'admin accede alla modifica del corso di laurea
12. L'admin seleziona il corso di laurea da modificare
13. L'admin seleziona i corsi da aggiungere al corso di laurea
14. L'admin seleziona i corsi da rimuovere dal corso di laurea
15. L'admin salva le scelte

16. Il sistema effettua la modifica

11-16 L'admin ripete il processo per tutti i corsi di laurea che intende modificare

## **Estensioni**

1-16\* L'admin può annullare il processo in ogni momento

1a. L'admin non crea alcun nuovo corso

1. Il caso d'uso riparte dal passo 6

5a. Il sistema non riesce a completare la creazione del corso

1. Il sistema non notifica il caricamento

2. Il caso d'uso riparte dal passo 1

6a. L'admin non crea alcun nuovo corso di laurea

1. Il caso d'uso riparte dal passo 11

10a. Il sistema non riesce a completare la creazione

1. Il sistema non notifica il caricamento

2. Il caso d'uso riparte dal passo 6

11a. L'admin non crea alcun nuovo corso

1. Il caso d'uso termina

13a. Non ci sono corsi da aggiungere

1. Il caso d'uso termina

14a. Non ci sono corsi da rimuovere

1. Il caso d'uso termina

16a. Il sistema non salva correttamente i corsi da aggiungere o da rimuovere

1. Il sistema non notifica il caricamento

2. Il caso d'uso riparte dal passo 11

## **Requisiti speciali**

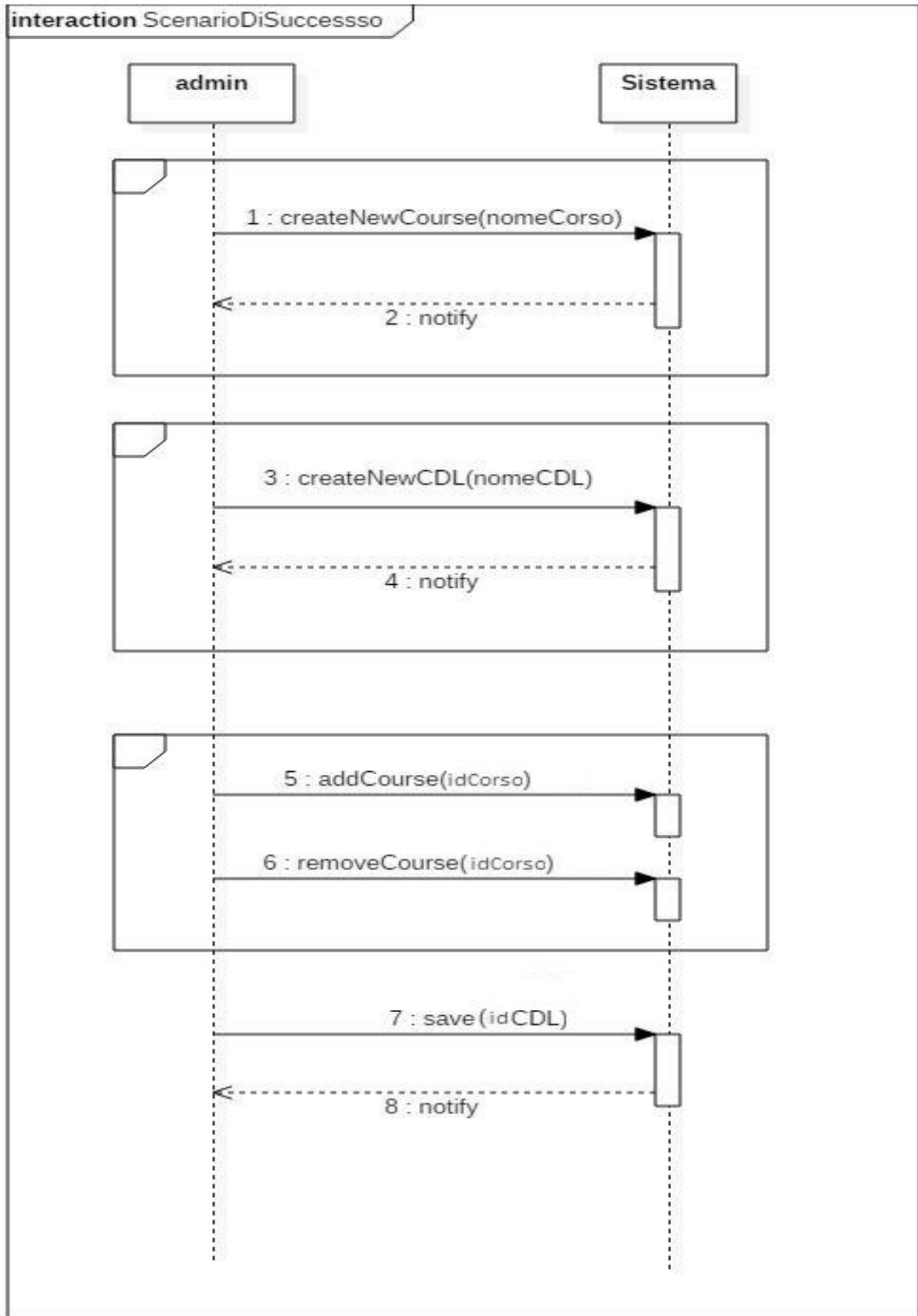
Interfaccia web responsive



## Frequenza di ripetizione

Anche ininterrotta

## Diagramma di sequenza



## Contratti

**Operazione:**createNewCourse(nomeCorso: string)

**Riferimenti:** caso d'uso: Gestire corsi di laurea

**Pre-condizioni:** nessuna

**Post-condizioni:**

1. È stata creata un'istanza c di Corso
2. c.nomeCorso è diventato nomeCorso

**Operazione:**createNewCDL(nomeCDL: string)

**Riferimenti:** caso d'uso: Gestire corsi di laurea

**Pre-condizioni:** nessuna

**Post-condizioni:**

1. È stata creata un'istanza cdl di Corso di Laurea
2. cdl.nomeCorso è diventato nomeCorso

**Operazione:**addCourse(idCorso : integer)

**Riferimenti:** caso d'uso: Gestire corsi di laurea

**Pre-condizioni:** devono esistere istanze cdl di Corso di Laurea e c di Corso

**Post-condizioni:**

1. c è stato associato a cdl.corsi

**Operazione:**removeCourse(idCorso : integer)

**Riferimenti:** caso d'uso: Gestire corsi di laurea

**Pre-condizioni:** devono esistere istanze cdl di Corso di Laurea e c di Corso

**Post-condizioni:**

1. c è stato rimosso da cdl.corsi

**Operazione:**save(idCDL : integer)

**Riferimenti:** caso d'uso: Gestire corsi di laurea

**Pre-condizioni:**deve esistere un'istanza cdl di corso di laurea

**Post-condizioni:**

1. cdl.corsi è stato modificato

## CLASSI CONCETTUALI

### **Studente**

Rappresenta gli studenti iscritti all'università

### **Professore**

Rappresenta i professori che insegnano all'università

### **Admin**

Rappresenta l'admin che deve gestire l'applicazione

### **Corso Di Laurea**

Rappresenta i corsi di laurea presenti nei dipartimenti dell'università

### **Corso**

Rappresenta i corsi presenti nei corsi di laurea dell'università

### **Appello**

Rappresenta gli appelli d'esame per ogni corso

### **Esame**

Rappresenta gli appelli d'esame dopo che è stato aggiunto un voto

### **Libretto**

Rappresenta gli esami che sono stati caricati sulla carriera degli studenti dopo che sono stati superati

### **Materiale**

Rappresenta i documenti che un professore carica per i corsi

## **Bandi**

Rappresenta i bandi e le news dell'università

## **Tassa**

Rappresenta le tasse universitarie per ogni studente

## **Ricevimento**

Rappresenta i ricevimenti tra professori e studenti

# **APPENDICE**

## **Tecnologie e programmi utilizzati**

Linguaggi Usati per lo svolgimento del progetto:

- Java
- JavaScript
- Html5
- HTTP
- CSS

Altre tecnologie usate:

- BootStrap
- Ajax
- Json
- Gson
- Google Maps Api
- Facebook API

La realizzazione di questo progetto ci è stata resa possibile anche grazie all'aiuto di diversi strumenti di sviluppo come:

- Star UML
- Eclipse
- Postgres DataBase