



**UNIVERSITÀ
DEL SALENTO**

Appunti online

PROGETTO DI INGEGNERIA DEL SOFTWARE



The Apache Software

Realizzato da:

Montanaro Teodoro (10091047)

SOMMARIO

1. Requisiti informali	4
2. Analisi e specifica dei requisiti.....	5
2.1 Stake holder principali (utenti):.....	5
2.2 Specifica dei requisiti funzionali	5
2.3 Specifica dei requisiti NON funzionali	7
3. Progettazione UML dell'architettura software	8
3.1. CASI D'USO	8
Attore principale : Utente generico.....	8
Attore principale : Studente	9
Attore principale : Gestore del servizio di scambio.....	15
3.2. DIAGRAMMI DEI CASI D'USO.....	25
Diagramma dei casi d'uso : Utente generico.....	25
Diagramma dei casi d'uso : Studente	26
Diagramma dei casi d'uso : Gestore del servizio di scambio.....	26
3.3. DIAGRAMMA DELLE CLASSI	27
3.4. DIAGRAMMI di ATTIVITA'	31
1) Registrazione al servizio	32
2) Richiedere un appunto	33
3) Confermare il carrello della spesa	34
4) Inserire la recensione di un appunto.....	35
5) Accettare le richieste di appunti.....	36
6) Modificare le informazioni varie relative ad un appunto.....	37
7) Modifica o eliminazione di una materia/docente	38
3.5. DIAGRAMMI di SEQUENZA	39
1) Login di uno studente	39
2) Registrazione di uno studente al servizio	40
3) Richiedere un appunto	41

4)	Inserire la recensione di un appunto.....	42
5)	Download di un appunto.....	43
6)	Eliminare una recensione.....	44
7)	Approvare la registrazione di uno studente.....	45
8)	Confermare l'avvenuta consegna di un appunto cartaceo.....	46
9)	Inserire un nuovo appunto.....	47
3.6.	DIAGRAMMA di PACKAGE.....	48
4.	Database.....	49
4.1.	Diagramma E-R.....	49
4.2.	Dizionario dei dati.....	50
4.3.	Tipi di relazioni.....	51
5.	Collaudo del sistema.....	53
1)	Metodi di collaudo usati.....	53
2)	Test di unità.....	53
3)	Test funzionali: test case.....	55
6.	Tecnologie usate.....	58
7.	Scrum "sprint backlog" e "burndown chart".....	59
	Sprint backlog 1.....	59
	Burndown chart 1.....	60
	Sprint backlog 2.....	68
	Burndown chart 2.....	69

1. Requisiti informali

Si realizzi un sistema software che consenta agli studenti iscritti a un corso di laurea magistrale in ingegneria informatica/gestionale lo scambio di appunti delle lezioni. Il sistema deve consentire agli utenti generici di prendere visione del catalogo degli appunti disponibili, suddivisi per materie di insegnamento (utilizzando gli stessi nomi dell'offerta formativa della facoltà), oppure suddivisi per autore (lo studente che li ha redatti), oppure suddivisi per docente titolare dell'insegnamento al quale si riferiscono. In aggiunta a ciò, il sistema deve consentire agli studenti iscritti al corso di laurea di scambiarsi online uno o più appunti (se gli appunti sono in formato cartaceo, dovranno essere preventivamente digitalizzati). Lo studente deve avere la possibilità di sfogliare il catalogo degli appunti, visualizzando, per ogni file di appunti, una preview parziale (per valutarne la leggibilità), una scheda descrittiva, una o più recensioni scritte online da chi ha già ricevuto il medesimo file completo degli appunti. All'atto della richiesta di un insieme di appunti, dopo aver esplicitamente confermato il contenuto del "carrello della spesa", lo studente deve avere a disposizione una form in cui confermare le proprie generalità (nome, cognome, matricola, facoltà, email, nickname), le preferenze di consegna degli appunti (ritiro dei file da parte dello studente oppure invio online); se lo studente opterà per l'invio online, il sistema gli trasmetterà un messaggio di email con indicazione della URL da cui scaricare gli appunti, opportunamente codificata così che solamente lui li potrà scaricare da detta URL. Il sistema deve consentire agli studenti iscritti di prendere visione dello stato delle richieste di appunti. Il sistema deve consentire agli studenti iscritti di scrivere recensioni a file di appunti che abbiano effettivamente richiesto (non è sufficiente che ne abbiano visionata la sola preview). Il sistema deve consentire al gestore del servizio di scambio di appunti di accedere all'elenco delle richieste e di accettarle o rifiutarle. Quando una richiesta è accettata o rifiutata, lo studente che l'ha fatta è avvisato automaticamente dal sistema per email.

Il sistema deve contenere almeno:

- Una pagina di presentazione del servizio di scambio di appunti delle lezioni, con le regole di utilizzo, da cui sia possibile accedere al catalogo degli appunti disponibili.
- Una pagina contenente il carrello delle richieste di appunti.
- Un'area di autenticazione al sistema (login).
- Una pagina di conferma della richiesta di appunti, contenente le form di sottomissione dei dati necessari all'effettuazione della richiesta e trasmissione degli appunti (per gli studenti iscritti).
- Una pagina di scrittura delle recensioni ad appunti (per gli studenti iscritti e che abbiano ricevuto i corrispondenti file di appunti).
- Una pagina di visione dello stato delle richieste di appunti (per gli studenti iscritti).
- Una pagina per la gestione (accettazione o rifiuto) delle richieste di appunti (per il gestore del servizio).
- Una pagina per la creazione e aggiornamento del catalogo degli appunti (per il gestore del servizio).

2. Analisi e specifica dei requisiti

Con l'analisi e la specifica dei requisiti si definiscono le proprietà di una applicazione.

I requisiti sono caratteristiche, proprietà e comportamenti che l'applicazione deve avere al termine dello sviluppo.

Per effettuare l'analisi dei requisiti si individuano i casi d'uso, gli attori e le relazioni che intercorrono tra di essi.

I casi d'uso descrivono le interazioni tra il sistema e l'utente, mentre gli attori rappresentano i ruoli di utenti che interagiscono con il sistema; Essi possono anche essere dei sottosistemi.

Un attore svolge uno o più casi d'uso e allo stesso modo un caso d'uso coinvolge uno o più attori.

2.1 Stake holder principali (utenti):

- Utente generico: colui che accede al servizio per visionare le funzionalità e le caratteristiche disponibili, ma che, non essendosi autenticato, non può utilizzare nessuna delle caratteristiche che necessitano di autenticazione.
- Studente: è il principale utilizzatore del servizio. Accede a tutte le funzionalità e a tutti i servizi disponibili per gli utenti registrati: visualizzare il catalogo degli appunti, richiedere e condividere appunti con altri utenti, visionare lo stato delle richieste di appunti e scrivere recensioni sugli appunti scaricati.

Si prevede che gli studenti inviino gli appunti al gestore tramite email.

- Gestore del servizio di scambio: si occupa della gestione del servizio accettando o rifiutando le richieste di appunti. Inoltre elimina vecchie voci e inserisce nuove voci al catalogo.

2.2 Specifica dei requisiti funzionali

Di seguito è riportato l'elenco dei requisiti funzionali divisi per attore.

- Utente generico:
 - ✓ Visualizzare il catalogo degli appunti disponibili
 - ✓ Visualizzare le regole di utilizzo del servizio
 - ✓ Registrarsi al servizio

- ✓ Accedere alla pagina di login
- Studente:
 - ✓ Autenticarsi al servizio
 - ✓ Sfogliare il catalogo degli appunti
 - ✓ Visualizzare, per ogni file di appunti, una preview parziale
 - ✓ Visualizzare una scheda descrittiva
 - ✓ Visualizzare una o più recensioni su un articolo scritte online da chi ha già ricevuto il medesimo file completo
 - ✓ Richiedere un insieme di appunti
 - ✓ Confermare il contenuto del “carrello della spesa”
 - ✓ Confermare le proprie generalità dopo aver confermato il contenuto del carrello
 - ✓ Confermare le preferenze di consegna degli appunti (ritiro dei file da parte dello studente oppure invio online)
 - ✓ Prendere visione dello stato delle richieste di appunti
 - ✓ Scrivere recensioni a file di appunti che abbiano effettivamente scaricato
 - ✓ Essere avvisato automaticamente dal sistema per email quando viene accettata la richiesta di appunti
 - ✓ Scaricare un appunto dopo essere stato autorizzato
- Gestore del servizio di scambio:
 - ✓ Autenticarsi al servizio
 - ✓ Accedere all’elenco delle richieste di appunti
 - ✓ Accettare o rifiutare le richieste di appunti
 - ✓ Eliminare un appunto o modificare le informazioni varie (preview, scheda descrittiva)
 - ✓ Inserire un nuovo appunto
 - ✓ Eliminare una recensione
 - ✓ Inserire una nuova materia/docente
 - ✓ Modificare o eliminare una materia/docente esistente
 - ✓ Confermare o ritirare una registrazione

- ✓ Gestire il ritiro cartaceo dell'appunto

2.3 Specifica dei requisiti NON funzionali

Di seguito è riportato l'elenco dei requisiti non funzionali del sistema.

- Usabilità: l'interfaccia utente del sistema è stata implementata cercando di garantire la massima operabilità, un veloce apprendimento e una facile localizzazione dei comandi da utilizzare. Viene garantita inoltre un'interfaccia coerente in tutte le sezioni dell'applicazione.
- Sicurezza: l'applicazione gestisce informazioni sensibili, pertanto deve garantire un determinato livello di sicurezza per preservarle. E' stata perciò implementata una procedura di autenticazione che permette di separare i diversi profili utente garantendo in questo modo diversi livelli di privilegi e di funzioni utilizzabili.

Tutte le chiavi di accesso sono crittografate e vengono generati link diversi a seconda degli utenti.

- Ambiente: Application server: Apache 2.2.19; Database Server: MySQL 5.5.15; PHP 4; Framework: PHP Codeigniter versione 2.0.3

3. Progettazione UML dell'architettura software

3.1.CASI D'USO

Attore principale : **Utente generico**

1. Visualizzare il catalogo degli appunti

PASSI

1. Seleziona dal menu la funzionalità **catalogo**
2. Il sistema visualizza l'elenco degli appunti disponibili ordinati per materia.

ESTENSIONI

2a. Catalogo vuoto

- Il sistema visualizza una pagina di avviso

2b. Cambio di ordinamento (ordinamento per materia/docente/autore/titolo/descrizione)

- L'utente seleziona il tipo di ordinamento (per materia, per docente, per autore, per titolo o per descrizione)
- Il sistema visualizza la lista ordinata.

2. Visualizzare le regole di utilizzo del servizio

PASSI

1. Seleziona dal menu la funzionalità **regole di utilizzo**
2. Il sistema visualizza una pagina dedicata alle regole di utilizzo.

3. Registrazione al servizio

PASSI

1. Seleziona dal menu la funzionalità di **registrazione**

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

2. Il sistema visualizza un modulo per la registrazione
3. L'utente compila il modulo per la registrazione in tutti i campi e poi preme sul tasto di conferma
4. Il sistema acquisisce i dati
5. Il sistema invia un'email di conferma
6. Il sistema mostra un messaggio di procedura terminata

ESTENSIONI

- 4a. Dati inseriti non corretti
- Il sistema visualizza un errore
 - Il sistema ritorna al passo 2

Attore principale : **Studente**

1. Autenticarsi al servizio

PASSI

1. Seleziona dal menu la funzionalità di **login**
2. Il sistema visualizza un modulo per il login
3. L'utente compila il modulo per il login
4. Il sistema acquisisce i dati
5. Il sistema visualizza l'home page relativa allo studente

ESTENSIONI

- 4a. Dati inseriti non corretti
- Il sistema visualizza un errore
 - Il sistema ritorna al passo 2

2. Visualizzare informazioni su un appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Seleziona dal menu la funzionalità relativa al catalogo degli **appunti**
2. Il sistema visualizza l'elenco degli appunti disponibili ordinati per titolo dell'appunto
3. L'utente seleziona il pulsante per la visualizzazione della scheda descrittiva dell'appunto
4. Il sistema visualizza una pagina con:
 - informazioni relative all'appunto
 - un pulsante per la prenotazione dell'appunto
 - un pulsante per la visualizzazione di una preview iniziale
 - una lista delle recensioni relative all'appunto ordinate per data di pubblicazione

ESTENSIONI

2a. Non ci sono appunti nel catalogo

- Il sistema visualizza un avviso in cui avverte l'utente che non ci sono appunti nel catalogo

2b. L'utente seleziona il tipo di ordinamento per docente / autore / materia / descrizione / titolo

- Il sistema visualizza l'elenco degli appunti ordinati per docente / autore / materia / descrizione / titolo
- Si torna al passo 3

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

3a. L'utente seleziona il pulsante per visualizzare la preview dell'appunto

- Il sistema visualizza la preview parziale dell'appunto in un popup

3b. L'utente seleziona il pulsante per visualizzare le recensioni dell'appunto

- Il sistema visualizza le recensioni dell'appunto

3. Richiedere un appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Utente visualizza la scheda descrittiva dell'appunto
2. L'utente preme sul pulsante per aggiungere l'appunto al proprio carrello
3. Il sistema acquisisce la richiesta
4. Il sistema inserisce l'appunto nel carrello
5. Il sistema rimanda lo studente nella pagina relativa al carrello

ESTENSIONI

3a. Appunto è già nel carrello

- Il sistema non permette di premere sul pulsante di prenotazione ma visualizza al suo posto un pulsante per rimandarlo al carrello

3b. L'appunto è stato già richiesto in precedenza ed è già stato accettato

- Il sistema non permette di premere sul pulsante di prenotazione ma visualizza al suo posto un pulsante per rimandarlo alla pagina con lo stato delle richieste effettuate

3c. L'appunto è stato già richiesto ma è stata rifiutata la richiesta

- Il sistema non permette di premere sul pulsante di prenotazione ma visualizza al suo posto un pulsante per rimandarlo alla pagina con lo stato delle richieste effettuate

4. Confermare il carrello della spesa

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Utente seleziona dal menu la funzione **carrello**
2. Il sistema visualizza l'elenco degli articoli inseriti nel carrello
3. L'utente conferma la richiesta
4. Il sistema visualizza un modulo da compilare con i propri dati personali
5. L'utente inserisce tutti i dati richiesti
6. Il sistema acquisisce i dati e la richiesta
7. Il sistema visualizza una pagina di procedura avvenuta correttamente

ESTENSIONI

2a. Il carrello è vuoto

- Il sistema visualizza un avviso

3a. L'utente decide di svuotare il carrello (rimuovere tutti gli appunti presenti nel carrello)

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

- Il sistema chiede conferma
- Si va al punto 2°

6a. I dati inseriti non sono corretti

- Il sistema mostra un messaggio di errore
- Il sistema torna al punto 4

5. Visionare lo stato delle richieste

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Utente seleziona dal menu la funzione **visualizza stato delle richieste**
2. Il sistema visualizza l'elenco delle richieste effettuate con lo stato ordinandole in base a quando sono state effettuate

ESTENSIONI

- 2a. Non è stata fatta alcuna richiesta
- Il sistema visualizza un avviso

6. Inserire la recensione di un appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

1. Utente visualizza l'elenco degli stato delle richieste
2. Sistema presenta un pulsante per inserire una **recensione** per ogni appunto
3. Il sistema visualizza un modulo da compilare
4. L'utente inserisce la recensione
5. Il sistema acquisisce i dati acquisiti

ESTENSIONI

2a. La recensione è già stata inserita

- Il sistema NON presenta il pulsante per l'inserimento ma scrive **Hai già recensito l'appunto.**

2a. L'appunto non è ancora stato realmente scaricato

- Il sistema NON presenta il pulsante per l'inserimento ma scrive **Non puoi recensirlo**

2a. E' stata rifiutata la richiesta di appunto

- Il sistema NON presenta il pulsante per l'inserimento ma scrive **Non puoi recensirlo**

7. Inviare un appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Utente seleziona funzionalità per **inserimento appunto**
2. Il sistema visualizza un modulo da compilare
3. Utente inserisce le informazioni richieste
4. Utente seleziona file da allegare

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

5. Il sistema verifica la correttezza dei dati inseriti
6. Il sistema acquisisce i dati acquisiti
7. Il sistema invia un'email al gestore contenente le informazioni inserite

ESTENSIONI

5a. Manca un'informazione richiesta o qualche campo è stato compilato in maniera errata

- Il sistema visualizza un errore
- Il sistema torna al punto 2

8. Scaricare un appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Utente riceve un'email con il link dell'appunto
2. Utente clicca sul pulsante per il download presente nell'email
3. Il sistema verifica la corrispondenza studente – link
4. Il sistema fa partire il download

ESTENSIONI

3a. L'url inserito non corrisponde allo studente che ne sta facendo richiesta

- Il sistema visualizza un errore

Attore principale : **Gestore del servizio di scambio**

1. Autenticarsi al servizio

PASSI

1. Seleziona dal menu la funzionalità di **login**
2. Il sistema visualizza un modulo per il login
3. L'utente compila il modulo per il login
4. Il sistema acquisisce i dati
5. Il sistema visualizza l'home page relativa allo studente

ESTENSIONI

- 4a. Dati inseriti non corretti
- Il sistema visualizza un errore
 - Il sistema ritorna al passo 2

2. Accettare le richieste di appunti

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare le **richieste**
2. Il gestore seleziona il pulsante "In sospeso"
3. Il sistema visualizza le richieste in sospeso
4. Il gestore conferma una singola richiesta premendo sul tasto apposito
5. Il sistema acquisisce la conferma
6. Il sistema genera un link per l'appunto
7. Il sistema invia un'email allo studente con il link generato
8. il sistema rimanda al punto 3

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

ESTENSIONI

2a. Il gestore seleziona il pulsante “Non approvate”

- Il sistema visualizza l’elenco di tutte le richieste non approvate e gli permette di approvarle

2b. Il gestore seleziona il pulsante “Cartacee da confermare”

- Il sistema visualizza l’elenco di tutte le richieste che chiedevano la consegna cartacea e gli permette di confermare che l’appunto è stato consegnato allo studente

3a. Non ci sono richieste in sospeso

- Il sistema visualizza un avviso

4a. Il gestore rifiuta una richiesta

- Il sistema acquisisce il rifiuto
- Il sistema invia un’email allo studente dicendogli che la sua richiesta non è stata accettata
- Il sistema rimanda al punto 2

6a. Lo studente aveva scelto di ritirare l’appunto per via cartacea

- Il sistema invia un’email allo studente invitandolo a contattare il gestore per mettersi d’accordo sul giorno, ora e luogo dell’incontro
- Il sistema rimanda al punto 2

3. Visualizzare gli appunti disponibili

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare tutti gli appunti disponibili
2. Il sistema visualizza tutti gli appunti disponibili

ESTENSIONI

2a. Non ci sono appunti disponibili

- Il sistema visualizza un avviso

4. Modificare le informazioni varie relative ad un appunto (preview, scheda descrittiva)

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare tutti gli appunti disponibili
2. Il sistema visualizza tutti gli appunti disponibili
3. Il gestore seleziona un appunto disponibile
4. Il sistema visualizza tutte le informazioni relative all'appunto (scheda descrittiva, link al file pdf dell'appunto)
5. Il gestore modifica qualsiasi informazione
6. Il gestore preme sul tasto di conferma delle modifiche
7. Il sistema acquisisce le modifiche
8. il sistema rimanda al punto 2

ESTENSIONI

2a. Non ci sono appunti disponibili

- Il sistema visualizza un avviso

7a. I dati inseriti non sono corretti (file non nel formato corretto o modifiche errate)

- Il sistema visualizza un avviso
- Il sistema rimanda al punto 4

5. Inserire un nuovo appunto

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare tutti gli appunti disponibili
2. Il gestore il tasto per inserire un nuovo appunto
3. Il sistema visualizza un form per inserire tutte le informazioni relative all'appunto (scheda descrittiva, link al file pdf dell'appunto, tasto per l'eliminazione)
4. Il gestore modifica qualsiasi informazione
5. Il gestore preme sul tasto di conferma delle modifiche
6. Il sistema acquisisce le modifiche
7. il sistema mostra un messaggio di inserimento avvenuto correttamente
8. Il sistema visualizza l'elenco degli appunti disponibili

ESTENSIONI

5a. I dati inseriti non sono corretti (file non nel formato corretto o modifiche errate)

- Il sistema visualizza un avviso
- Il sistema rimanda al punto 2

6. Eliminare una recensione

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare gli appunti disponibili
2. Il sistema visualizza tutti gli appunti disponibili
3. Il gestore seleziona il tasto **recensioni**
4. Il sistema visualizza un elenco con tutte le recensioni disponibili
5. Il gestore preme sul tasto di eliminazione di una recensione
6. Il sistema acquisisce l'eliminazione
7. Il sistema rimanda al punto 4

ESTENSIONI

2a. Non ci sono appunti disponibili

- Il sistema visualizza un avviso

4a. Non ci sono recensioni disponibili

- Il sistema visualizza un avviso

7. Approvare/Disapprovare una recensione

PRECONDIZIONI

1. Utente si è autenticato al servizio

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

PASSI

1. Il gestore seleziona la funzionalità per visualizzare gli appunti disponibili
2. Il sistema visualizza tutti gli appunti disponibili
3. Il gestore seleziona il tasto **recensioni**
4. Il sistema visualizza un elenco con tutte le recensioni disponibili con un pulsante per l'approvazione
5. Il gestore preme sul tasto di approvazione / disapprovazione di una recensione
6. Il sistema acquisisce la modifica
7. Il sistema rimanda al punto 4

ESTENSIONI

2a. Non ci sono appunti disponibili

- Il sistema visualizza un avviso

4a. Non ci sono recensioni disponibili

- Il sistema visualizza un avviso

4b. La recensione è già stata approvata

- Il sistema visualizza un pulsante per la disapprovazione

ALTERNATIVA

1. Il gestore seleziona la funzionalità per **visualizzare le recensioni in sospeso**
2. Il sistema visualizza un elenco con tutte le recensioni in attesa di approvazione con un pulsante per l'approvazione
3. Il gestore preme sul tasto di approvazione di una recensione

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

4. Il sistema acquisisce la modifica
5. Il sistema rimanda al punto 4

ESTENSIONI

2a. Non ci sono recensioni disponibili

- Il sistema visualizza un avviso

8. Modifica o eliminazione di una materia/docente

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per modificare una materia/docente
2. Il sistema visualizza un elenco di tutte le materie/docenti disponibili
3. Il sistema visualizza un pulsante per la modifica e uno per l'eliminazione
4. Il gestore seleziona il pulsante per la modifica
5. Il sistema visualizza tutte le informazioni relative alla materia/docente permettendo la modifica
6. Il gestore modifica qualsiasi informazione
7. Il sistema acquisisce i dati inseriti
8. Il sistema visualizza un messaggio di corretto inserimento dei dati

ESTENSIONI

2a. non ci sono materie/docenti disponibili

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

- il sistema visualizza un avviso

3a. La materia/docente è associata a qualche appunto presente nel database

- il sistema visualizza un avviso: “Non si può eliminare” ed accanto il pulsante per la modifica

4a. il gestore seleziona la funzionalità di **eliminazione**

- Il sistema chiede conferma
- Il sistema elimina tutte le informazioni relative alla materia/docente selezionata
- il sistema rimanda al punto 2

7a. i dati non sono corretti

- Il sistema visualizza un avviso
- Il sistema rimanda al punto 4

9. Inserire una nuova materia/docente

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per modificare una materia/docente
2. Il gestore preme sul pulsante per l’inserimento di una nuova materia/docente
3. Il sistema visualizza un modulo per inserire tutte le informazioni relative alla materia o al docente
4. Il gestore compila il modulo

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

5. Il sistema acquisisce i dati inseriti
6. Il sistema visualizza un messaggio di corretto inserimento dei dati

ESTENSIONI

5a. i dati non sono corretti

- Il sistema visualizza un avviso
- Il sistema rimanda al punto 2

10. Confermare o ritirare una registrazione

PRECONDIZIONI

1. Utente si è autenticato al servizio

PASSI

1. Il gestore seleziona la funzionalità per visualizzare le registrazioni
2. Il gestore seleziona il pulsante "In sospeso"
3. Il sistema visualizza tutti gli utenti registrati al servizio non ancora autorizzati ad accedere al servizio
4. Il gestore seleziona il tasto per l'approvazione
5. Il sistema acquisisce la modifica
6. Il sistema invia un'email allo studente per avvertirlo che è stato autorizzato
7. Il sistema rimanda al punto 2

ESTENSIONI

2a. Il gestore seleziona il tasto "Non approvate"

- il sistema visualizza tutte le registrazioni non approvate e gli permette di approvarle

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

3a. non ci sono utenti disponibili

- il sistema visualizza un avviso

4a. il gestore seleziona la funzionalità di **disapprovazione**

- Il sistema chiede conferma
- Il sistema invia un'email all'utente selezionato
- il sistema rimanda al punto 2

6a. Il sistema non riesce ad inviare l'email

- il sistema visualizza un avviso di errore

3.2. DIAGRAMMI DEI CASI D'USO

I diagrammi dei casi d'uso riportati rappresentano la mappa riassuntiva dei casi d'uso e riportano tutti gli attori (stakeholder), i goal e le relazioni fra loro.

Per non appesantire la notazione non sono state riportate le relazioni di inclusione fra alcuni casi d'uso del gestore e quello di autenticazione al servizio (i casi d'uso non collegati sono "Inserire una nuova materia/docente", "Eliminare una recensione") ed alcuni casi d'uso dello studente con quello di autenticazione (in particolare l'unico a non essere collegato è il caso d'uso "inserire la recensione ad un appunto").

Le inclusioni, in ogni caso, potevano essere omesse tutte in quanto l'autenticazione al servizio è stata formalizzata come pre-condizione.

Diagramma dei casi d'uso : **Utente generico**



Diagramma dei casi d'uso : **Studente**

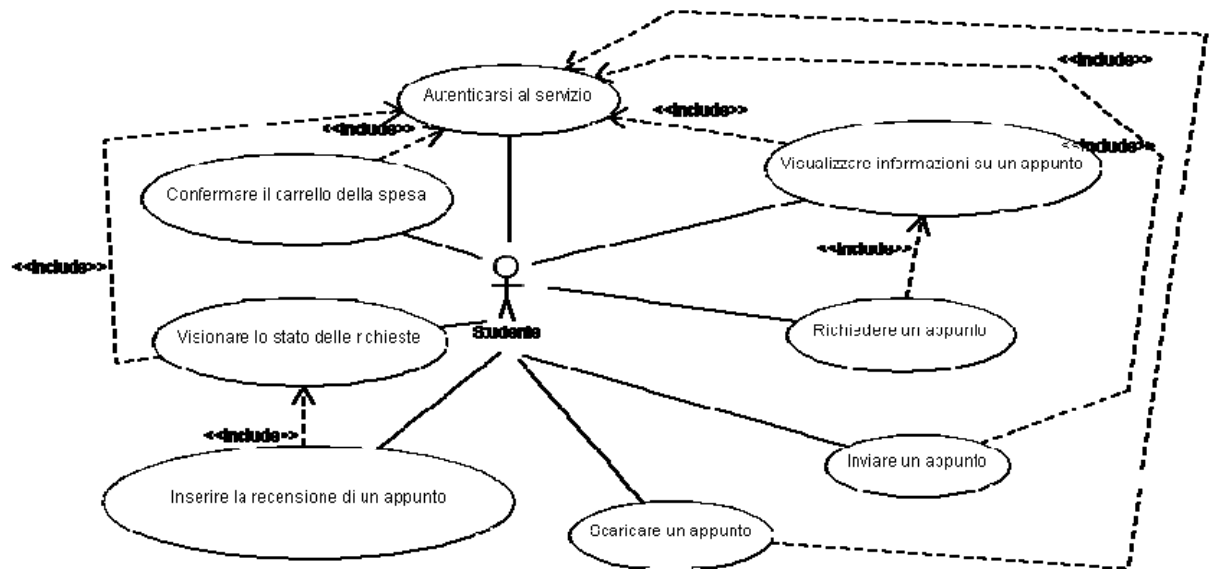
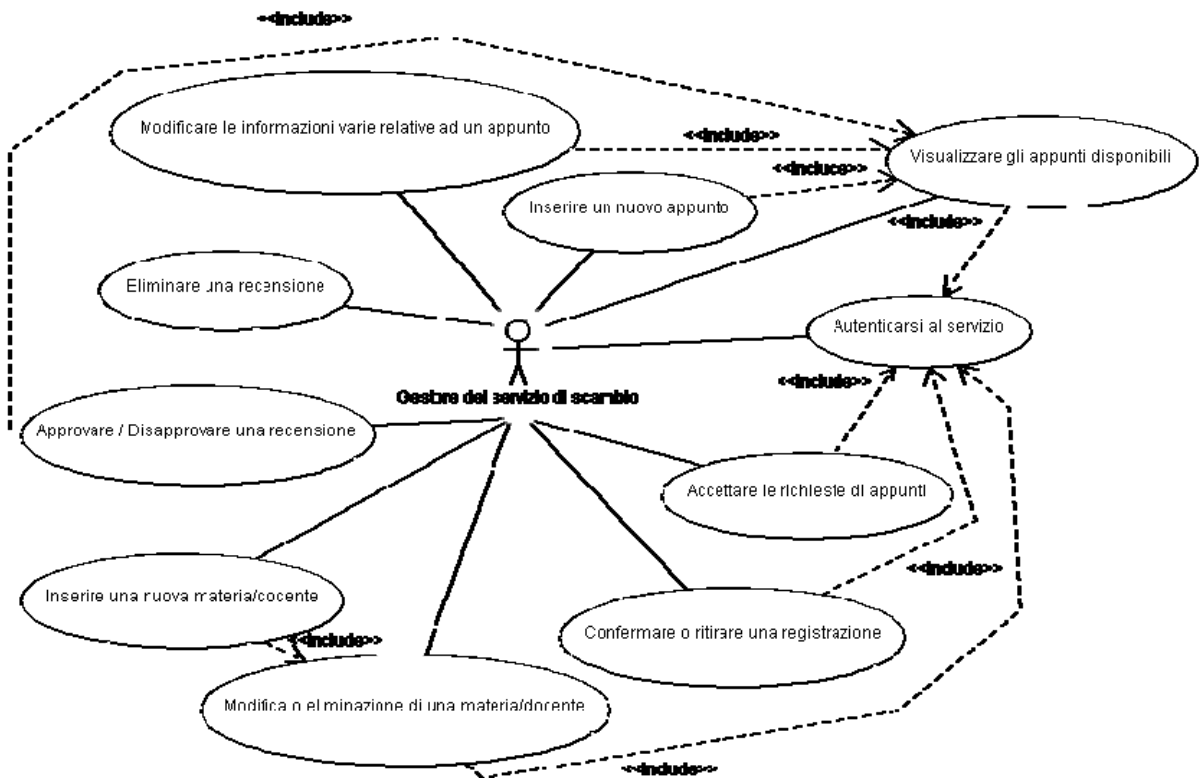
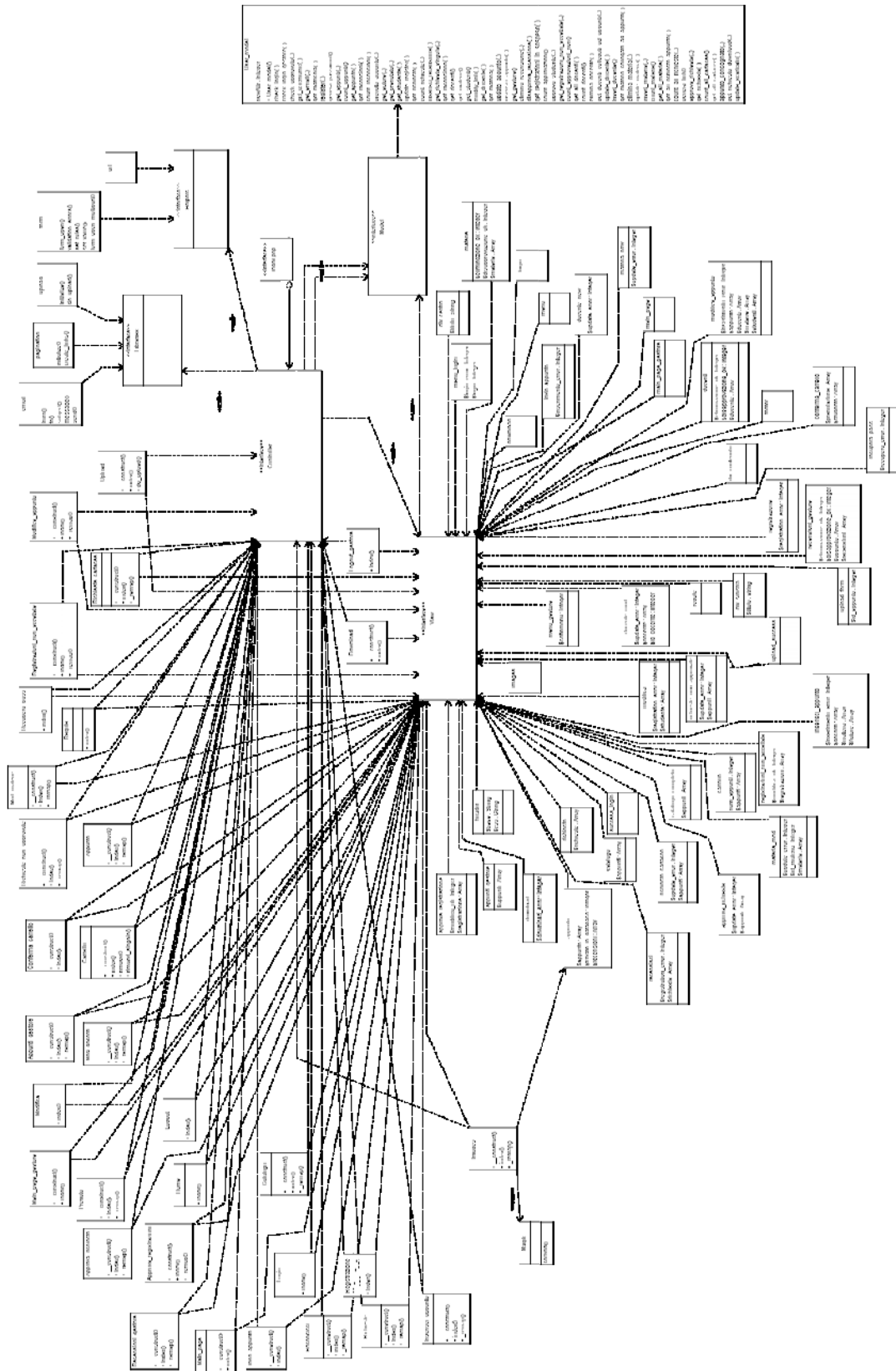


Diagramma dei casi d'uso : **Gestore del servizio di scambio**



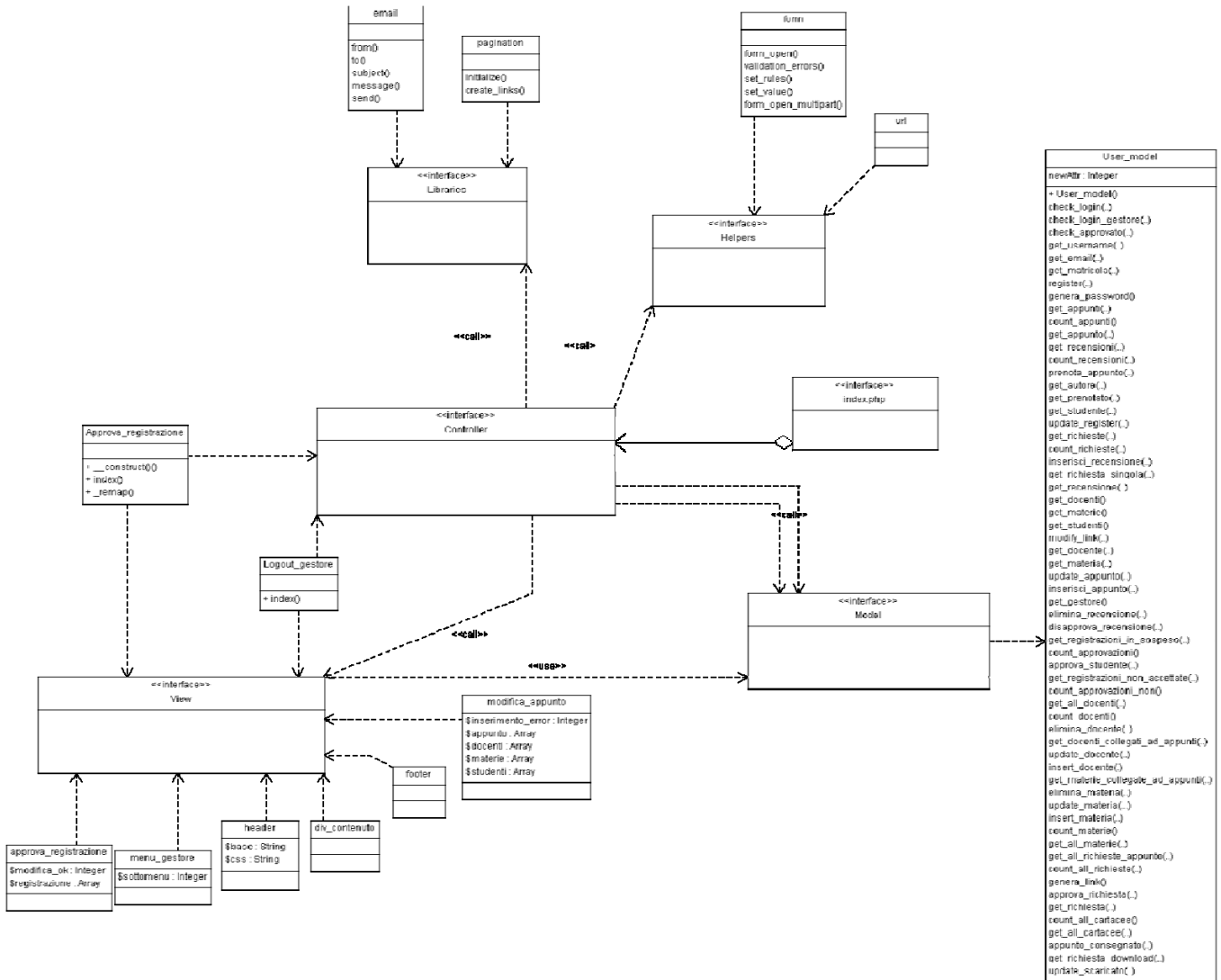
3.3. DIAGRAMMA DELLE CLASSI

Di seguito è riportato il diagramma delle classi completo.



Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Per rendere più agevole la comprensione del diagramma si è preferito isolare le interfacce al centro dell'architettura eliminando tutte le classi associate tranne una: quella relativa alla funzionalità di approvazione di una registrazione, funzionalità prevista per il solo gestore del servizio.



Il diagramma sopra riportato è una parte dell'architettura del sistema che rappresenta il pattern architetturale MVC (Model – View – Controller).

Il pattern in questione (MVC) permette di implementare applicazioni nelle quali vengono separati i compiti tra 3 diversi componenti:

- il model che fornisce all'applicazione i metodi per accedere ai dati utili;
- il view che visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- il controller che riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti

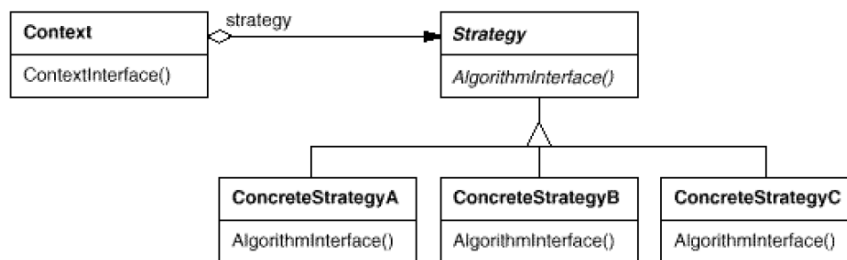
Progetto realizzato da Teodoro Montanaro (matricola 10091047)

La forza di questo pattern è appunto quella di separare la logica di business (o applicativa) dall'interfaccia utente.

L'architettura implementata dal framework PHP CodeIgniter ricalca grosso modo 2 dei più importanti design pattern: Strategy e Observer. E' inoltre possibile riconoscere il pattern Composite guardando l'implementazioni delle classi concrete di View e di Controller.

Mentre per il primo (Strategy) si ha un funzionamento identico, per il secondo (Observer) si possono notare alcune differenze rispetto all'implementazione standard. Ciò nonostante di seguito verranno riportati entrambi con l'intento di far notare le similitudini che l'architettura MVC implementata in PHP CodeIgniter ha con questi 2 design pattern.

Cominciamo con il riportare la struttura del design pattern **Strategy**:

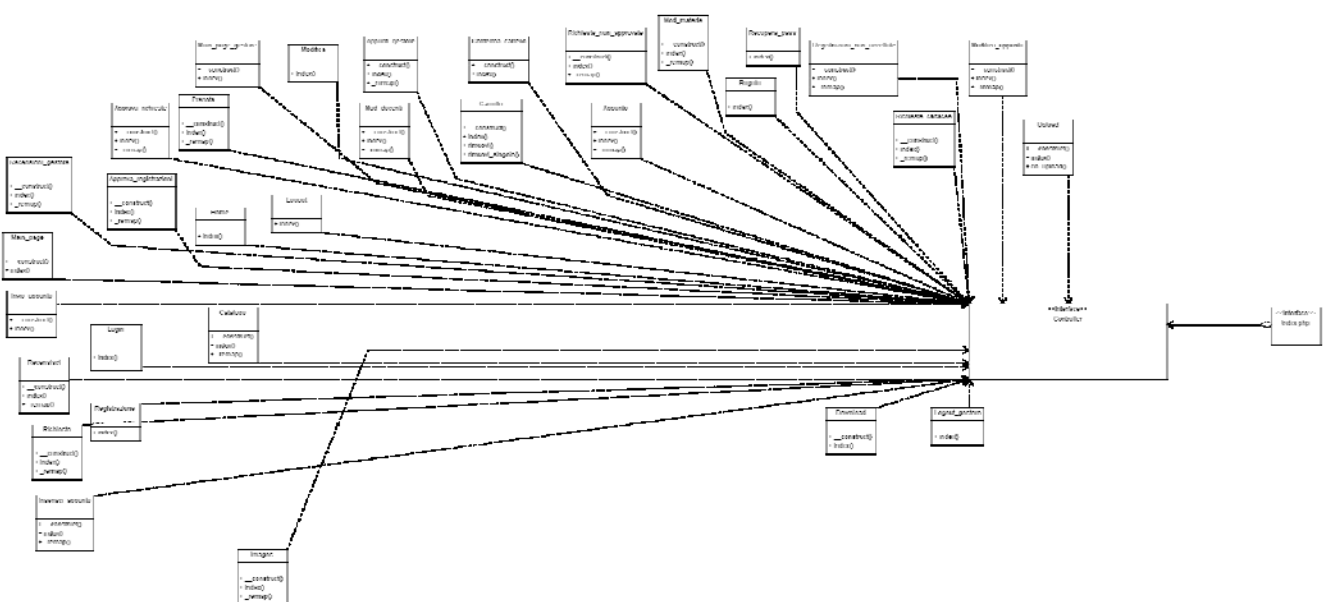


Come è possibile notare vi è un oggetto ("Context") che si preoccupa di caricare oggetti ConcreteStrategy.

Nel nostro caso, l'oggetto Context corrisponde all'interfaccia index.php che non è altro che l'oggetto che si preoccupa di caricare ogni oggetto Controller.

L'oggetto Strategy corrisponde all'interfaccia Controller e i ConcreteStrategy corrispondono alle reali implementazioni dei Controller (nella figura di sotto sono Approva_registrazione, Menu_gestore, ...)

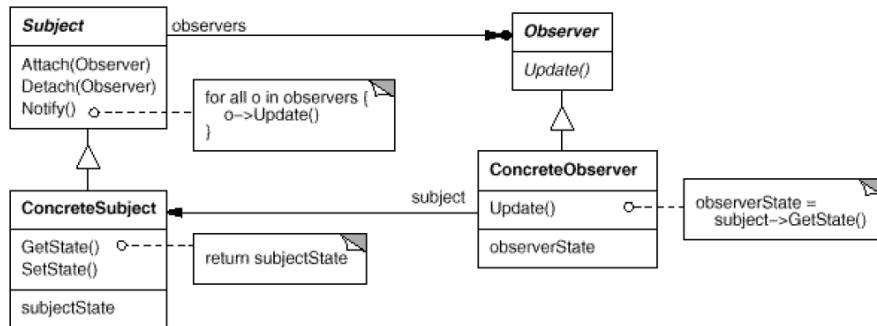
Di seguito riporto la struttura della nostra implementazione.



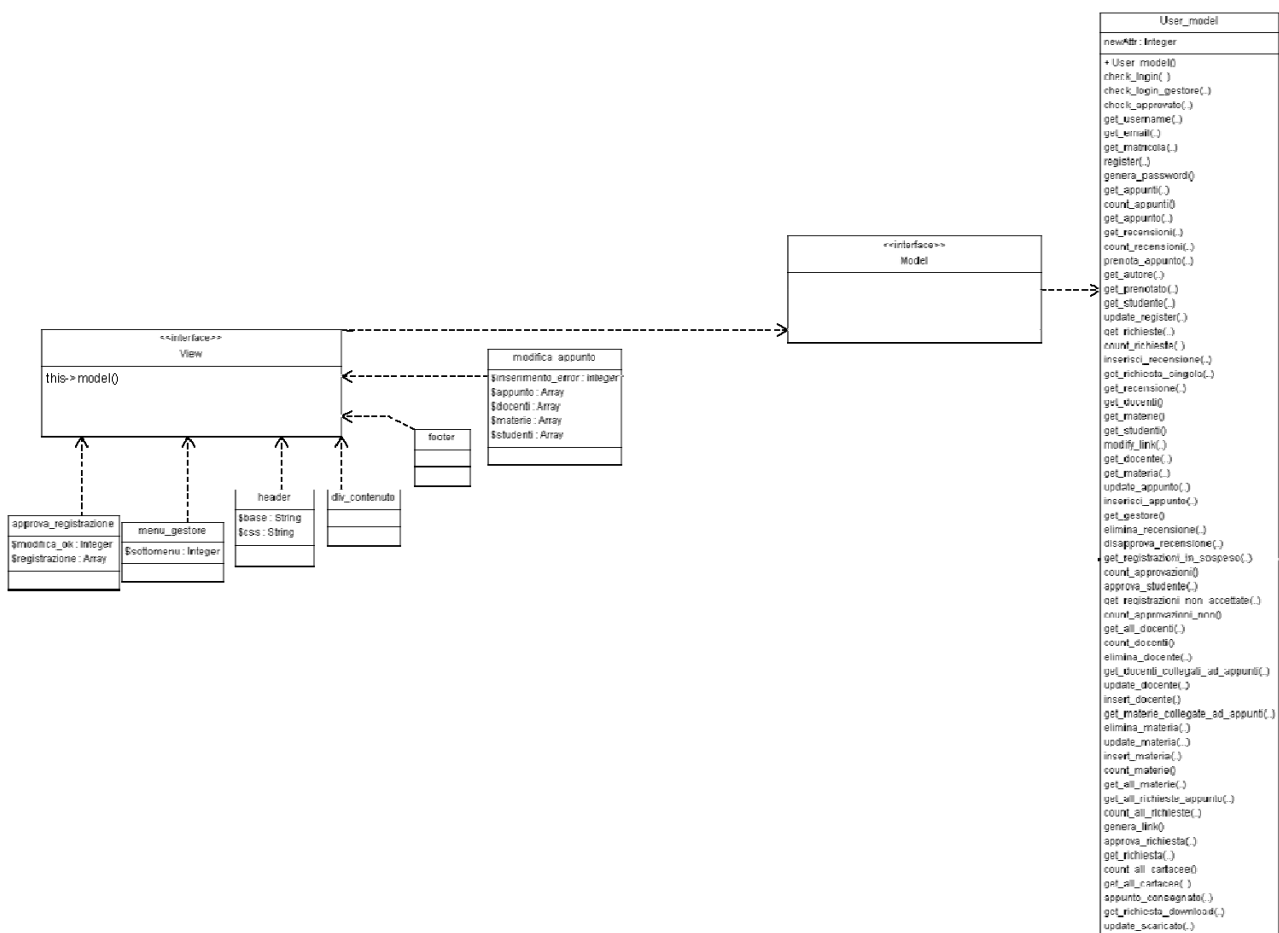
Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Analizzando invece il funzionamento e le interazioni tra View e Model si può riconoscere l'implementazione del design pattern **Observer**.

La struttura del design pattern Observer è



E la struttura delle interazioni tra model e view è molto simile:



L'oggetto **View** corrisponde all'oggetto Subject e **fornisce l'interfaccia per collegare oggetti Model**

L'oggetto **Model** invece dovrebbe corrispondere ad **Observer** e fornire una specie di interfaccia per la notifica degli oggetti a cui devono essere notificati i cambiamenti di stato di Subject.

In realtà questa **notifica non esiste**, ma dal momento che il model si preoccupa di aggiornare il database, stiamo di fatto "notificando" a tutte le altre classi che c'è stato un cambiamento. Non lo fanno subito, ma appena verranno richiamate verrà caricata dal database la nuova versione dei dati.

Se ci viene passata questa forzatura, possiamo dire che le View realizzative corrispondono alle realizzazioni dei Concrete Subject e i Model a quelle di Concrete Observer.

3.4. DIAGRAMMI di ATTIVITA'

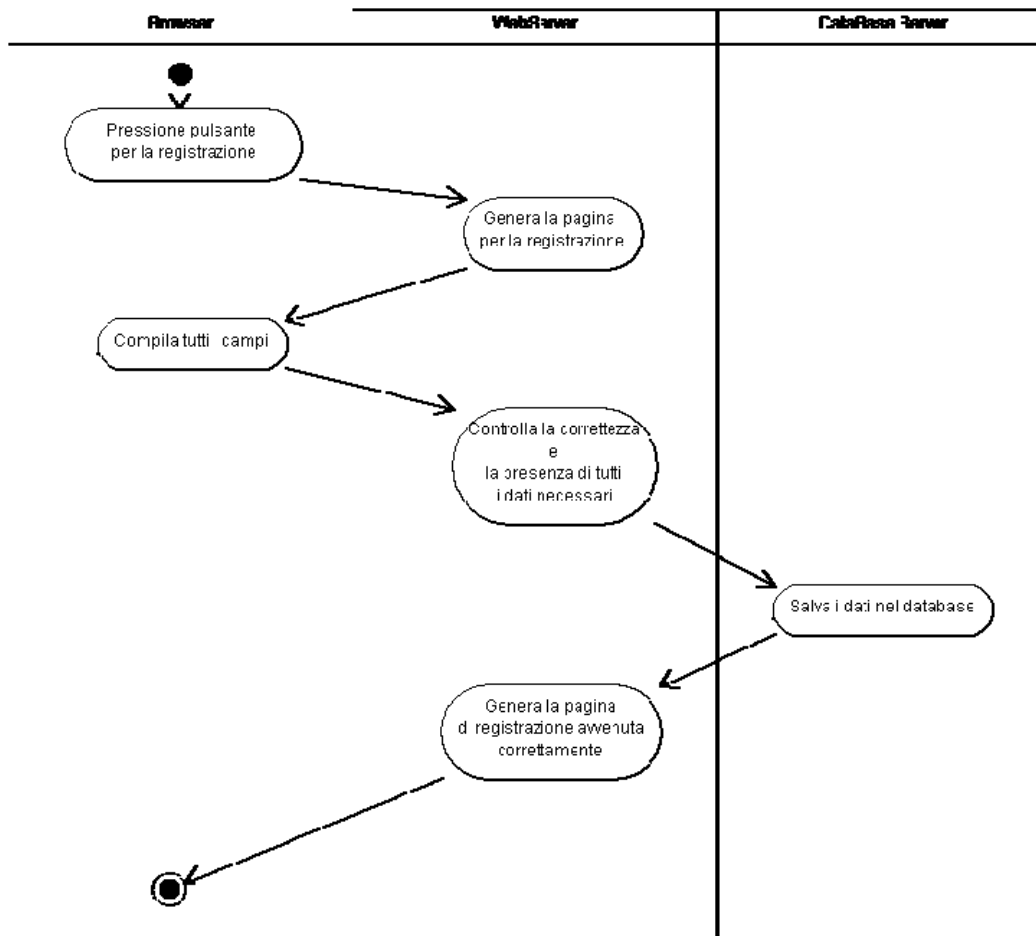
I diagrammi di attività descrivono la sequenza delle attività che intervengono nel flusso di un programma.

Essi mostrano i passi (chiamati propriamente attività), i punti decisionali e i rami (cioè le relazioni tra le attività) che intervengono nel flusso di un programma e descrivono le attività necessarie per il completamento dei casi d'uso (all'interno di un singolo caso d'uso vi possono essere diversi percorsi possibili).

Rispetto ai diagrammi di sequenza (proposti in seguito) permettono di comprendere meglio gli algoritmi usati essendo molto simili a 'flow charts'.

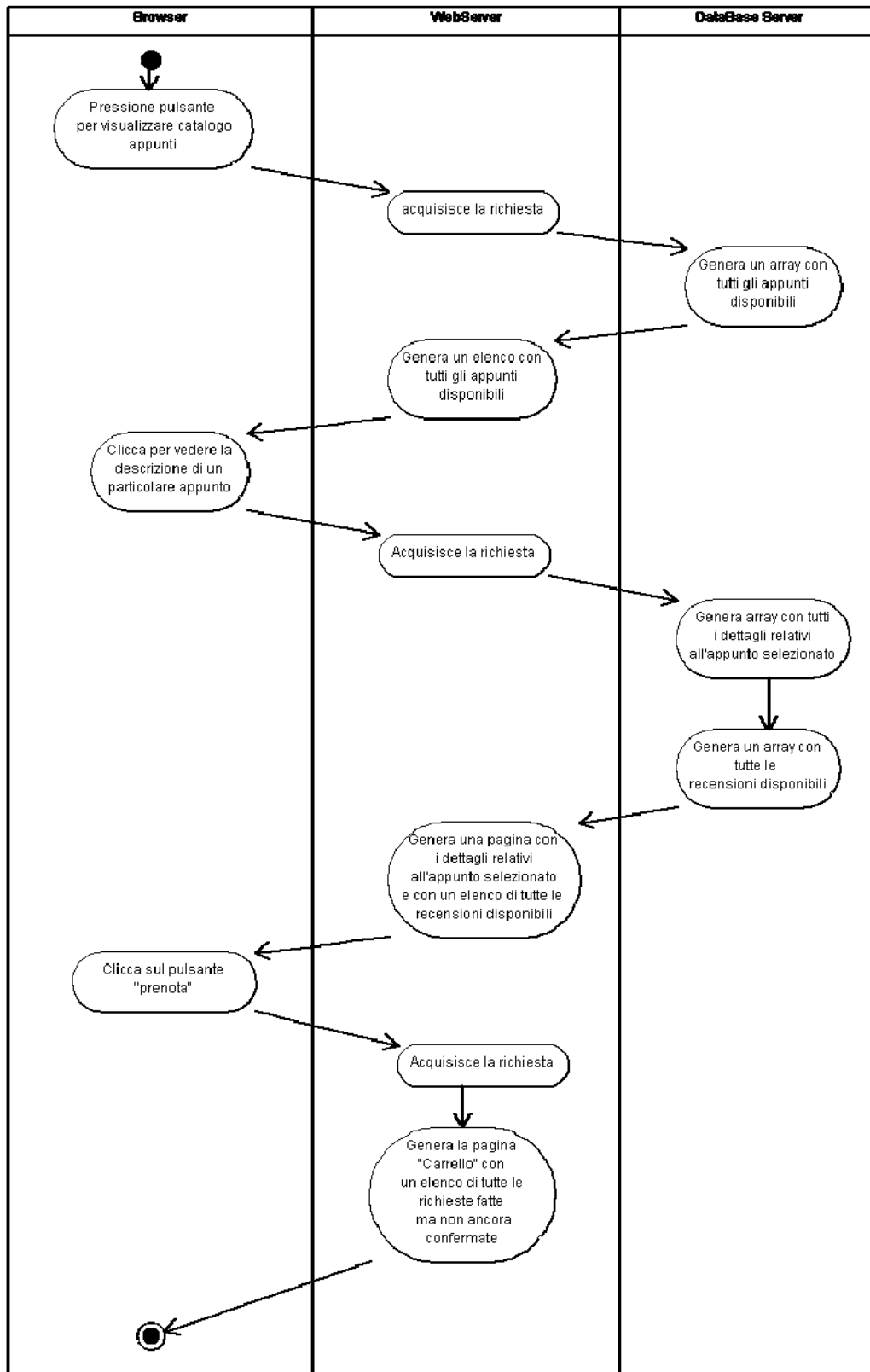
Di seguito vengono riportati i diagrammi di attività relativi ai casi d'uso di maggiore rilevanza

1) Registrazione al servizio



Questo primo caso d'uso è relativo alla registrazione al servizio di appunti online. Al termine della fase di registrazione l'utente non può ancora accedere al servizio in quanto il gestore deve ancora convalidare la richiesta.

2) Richiedere un appunto

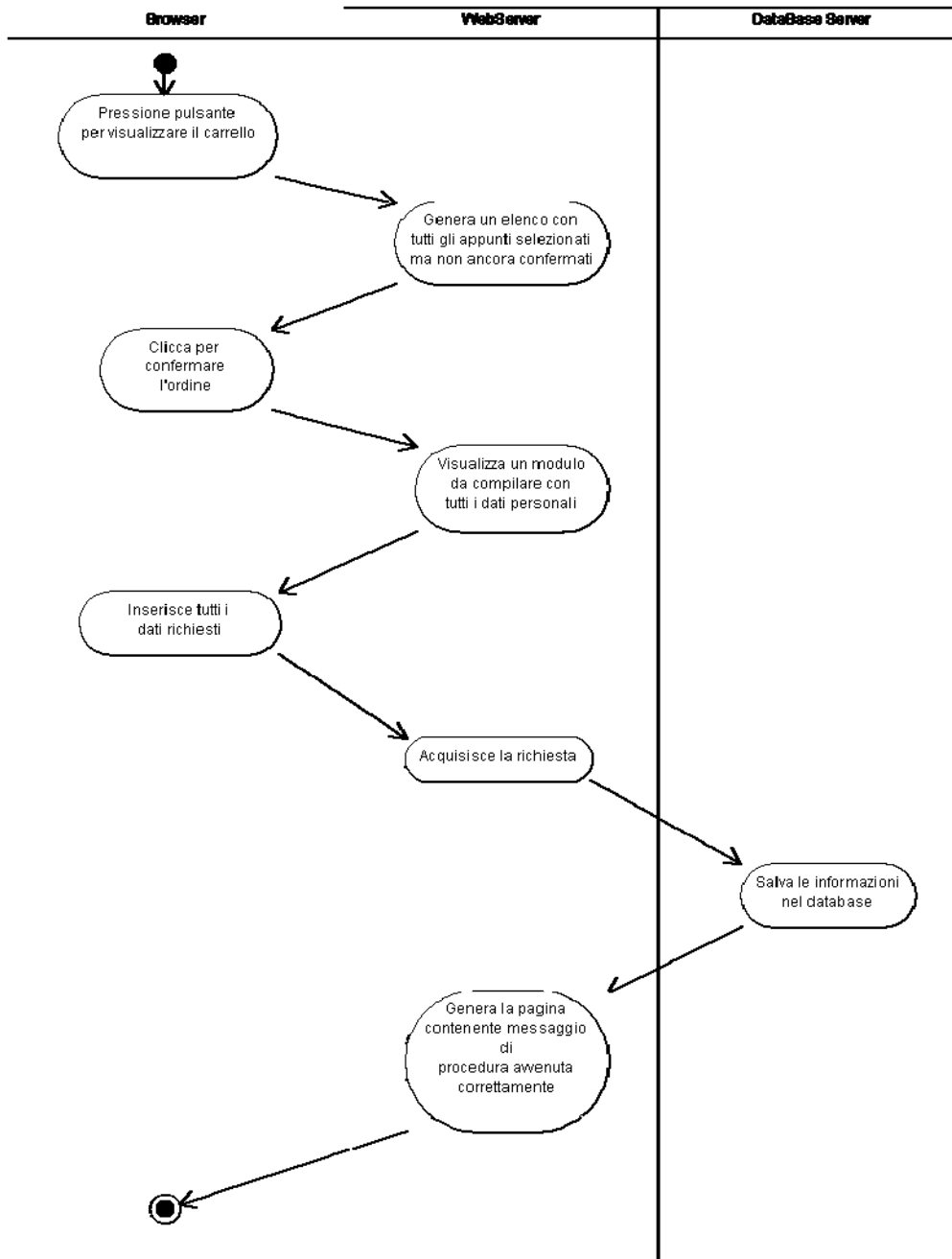


Questo diagramma di attività si riferisce al caso d'uso relativo alla richiesta di un appunto da parte di uno studente. Prima di inserire i dati nel database, lo studente deve andare nel carrello e confermare le

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

richieste inserendo i dati personali (servono qualora lo studente voglia che l'appuntamento venga recapitato ad un indirizzo email diverso da quello inserito in fase di registrazione).

3) Confermare il carrello della spesa



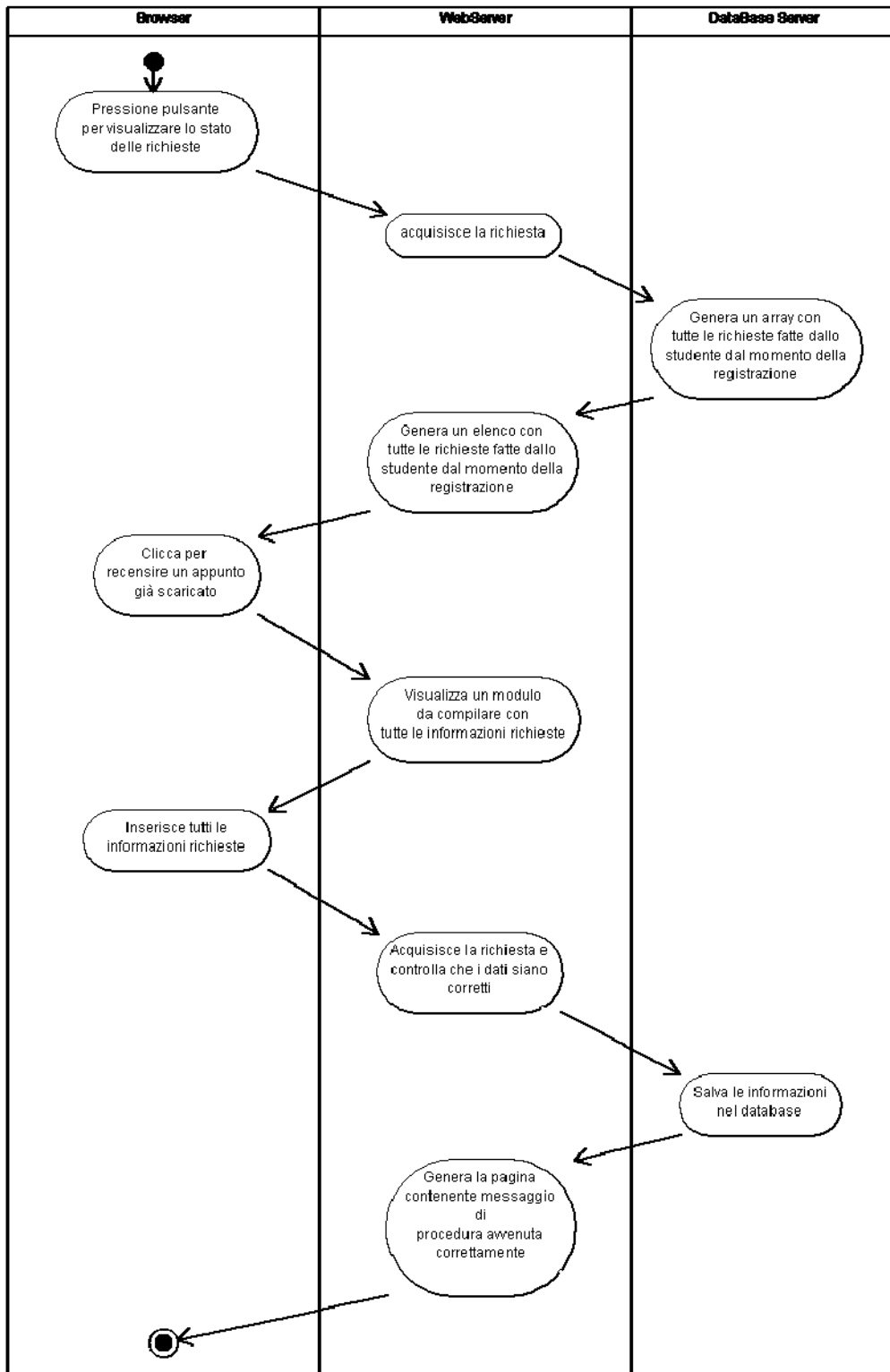
Questo diagramma delle attività è relativo al caso d’uso che riguarda la conferma del carrello della “spesa”.

Viene chiesto allo studente a quale indirizzo email vuole ricevere il link dell’appuntamento e se preferisce riceverlo in formato cartaceo o in formato digitale.

Se chiede di riceverlo in formato cartaceo il gestore gli invierà un’email per mettersi d’accordo su dove e quando incontrarsi.

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

4) Inserire la recensione di un appunto

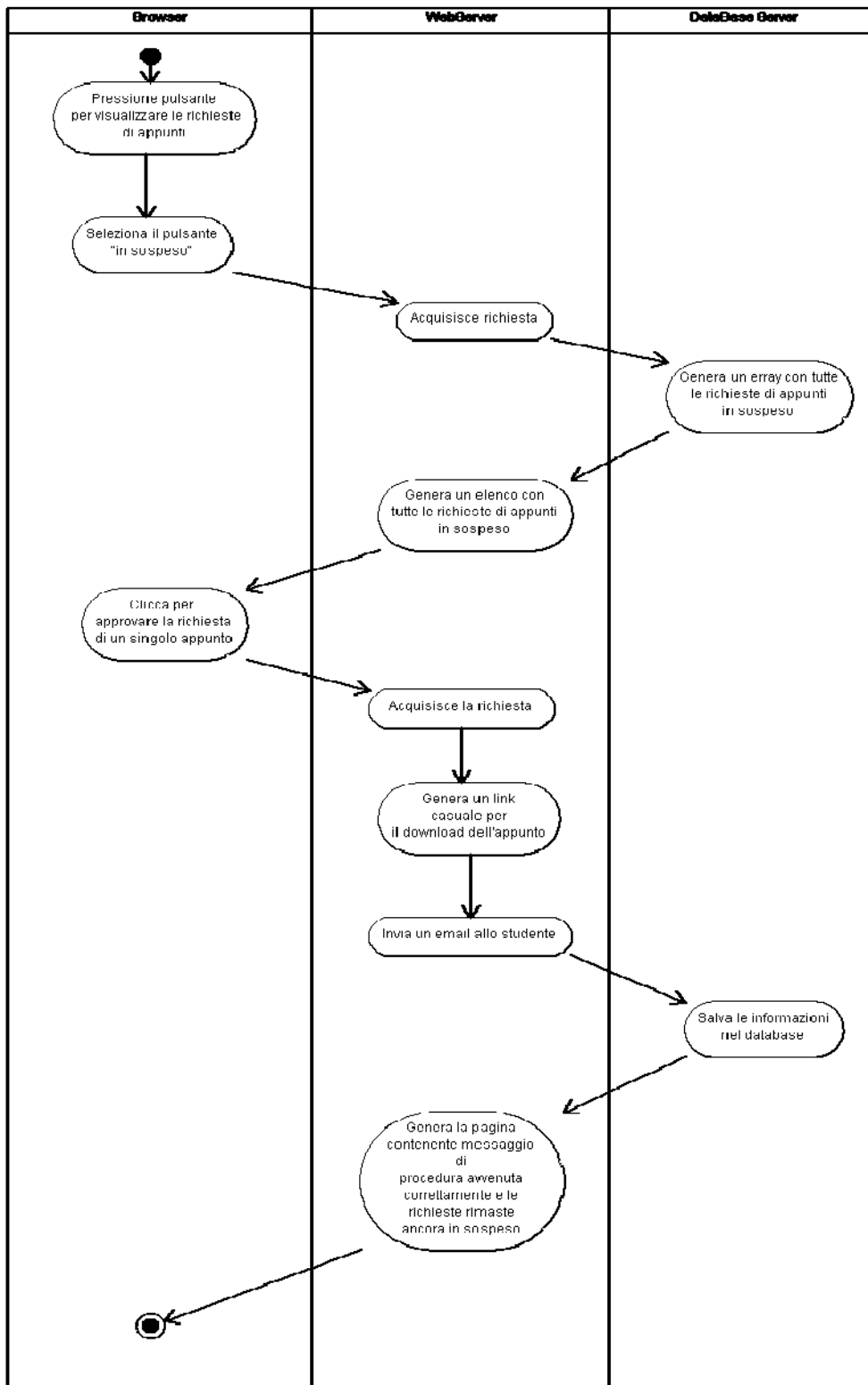


Questo diagramma si riferisce al caso d'uso relativo all'inserimento di una recensione.

Il sistema permette l'accesso a questa pagina solo qualora lo studente abbia già scaricato completamente l'appunto.

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

5) Accettare le richieste di appunti



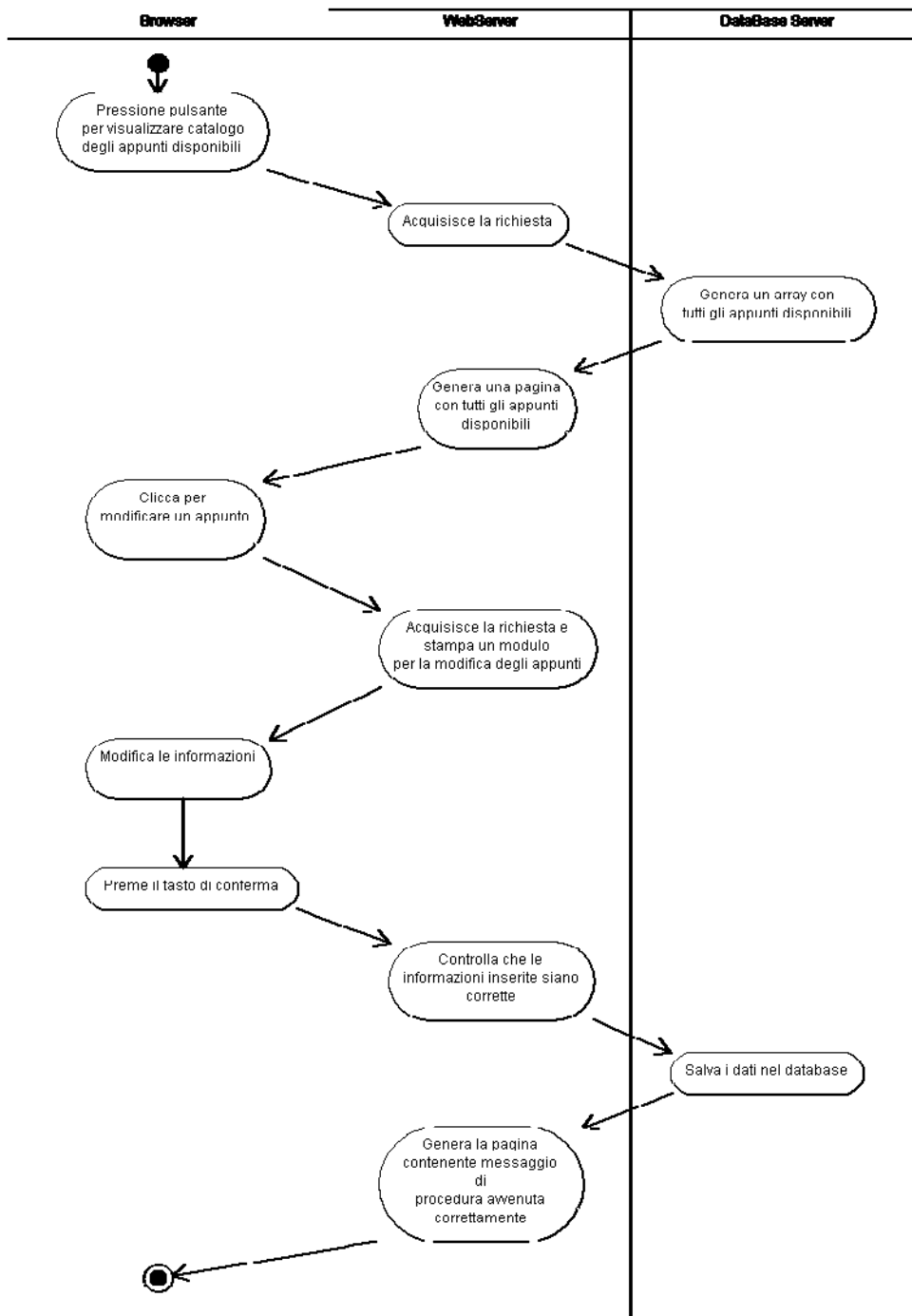
Questo diagramma è relativo al caso d'uso relativo all'accettazione di una richiesta di appunto.

Si preferisce inviare prima l'email allo studente e poi memorizzare le informazioni nel database perché il sistema, una volta memorizzato nel database la conferma non permette più di inviare l'email contenente il

link, quindi lo studente potrebbe non venire mai a conoscenza della conferma! Mentre nel caso contrario se l'invio non va a buon fine non vengono neanche salvati i dati nel database e quindi il gestore può ripetere la procedura.

Anche se lo studente ha richiesto l'appunto in forma cartacea viene inviata un'email. L'unica differenza sta nel fatto che nel secondo caso l'email conterrà solo un messaggio che invita lo studente ad inviare un'email al gestore per mettersi d'accordo su quando e dove incontrarsi per potergli dare l'appunto.

6) Modificare le informazioni varie relative ad un appunto

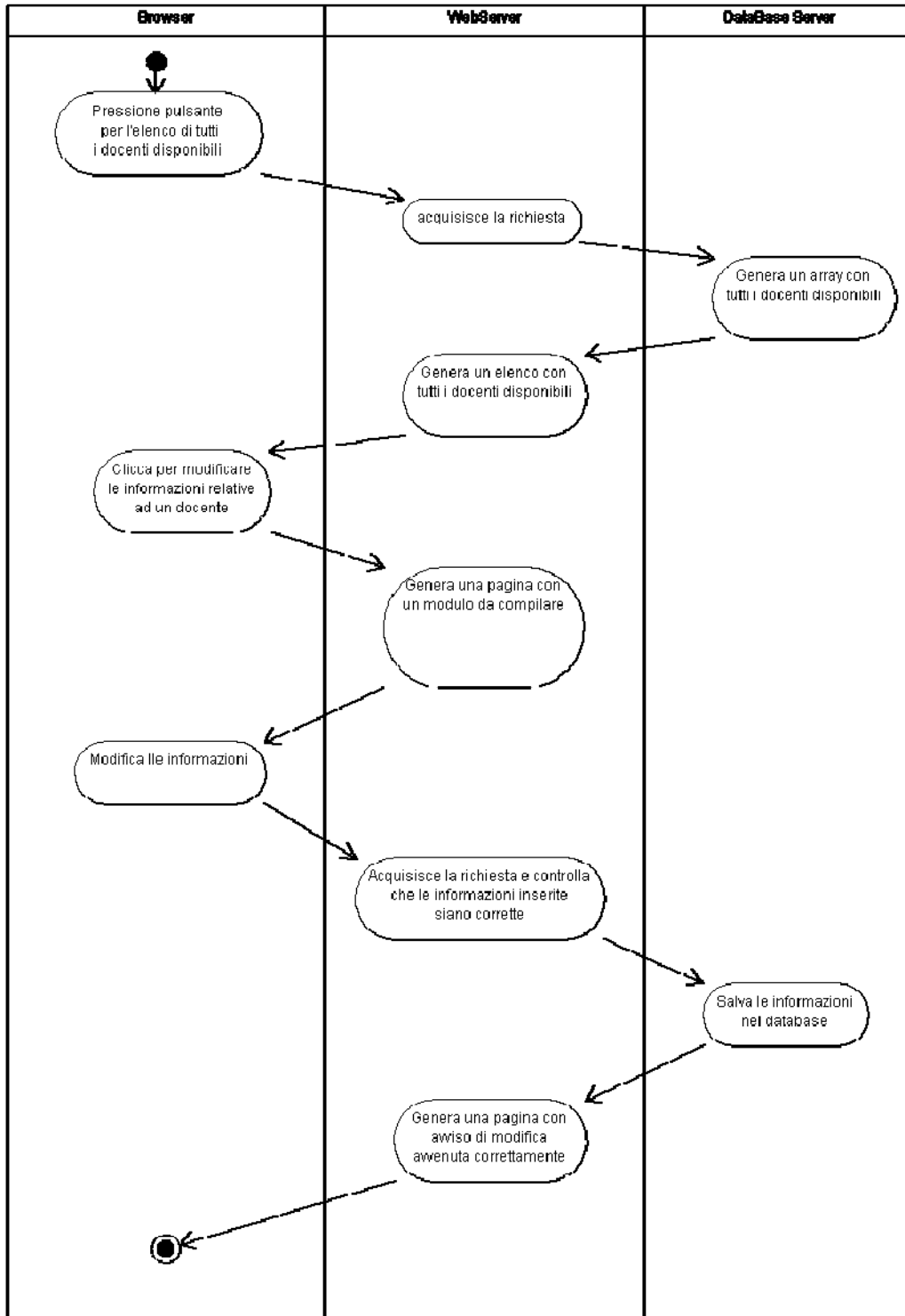


Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Questo diagramma è relativo alla modifica delle informazioni relative ad un appunto.

Qui è possibile anche caricare un pdf diverso. E' ammesso solo il formato pdf.

7) Modifica o eliminazione di una materia/docente



Questo diagramma si riferisce al caso in cui si voglia modificare le informazioni relative ad un docente. Il procedimento è identico a quello per la modifica delle informazioni relativa ad una materia.

Inoltre il diagramma relativo all'inserimento di un nuovo docente / materia è molto simile a questo, l'unica differenza è che non si devono caricare in prima istanza le informazioni già inserite.

3.5. DIAGRAMMI di SEQUENZA

I diagrammi di sequenza servono a specificare il funzionamento a basso livello della sequenza di operazioni tra le classi.

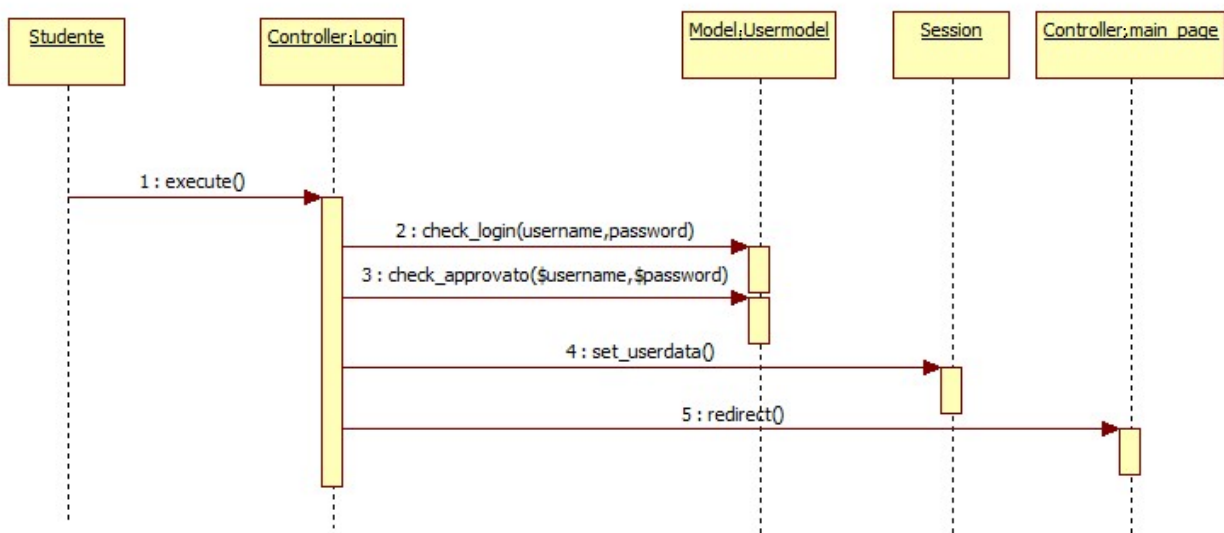
Sono molto importanti soprattutto se qualche processo ha durata più lunga rispetto ad altri, o riveste un ruolo più importante.

Essi documentano tipicamente il comportamento di un singolo scenario ed includono:

- un certo numero di oggetti
- i messaggi scambiati tra essi durante l'esecuzione del caso d'uso.

Di seguito sono rappresentati i diagrammi di sequenza relativi a:

1) Login di uno studente



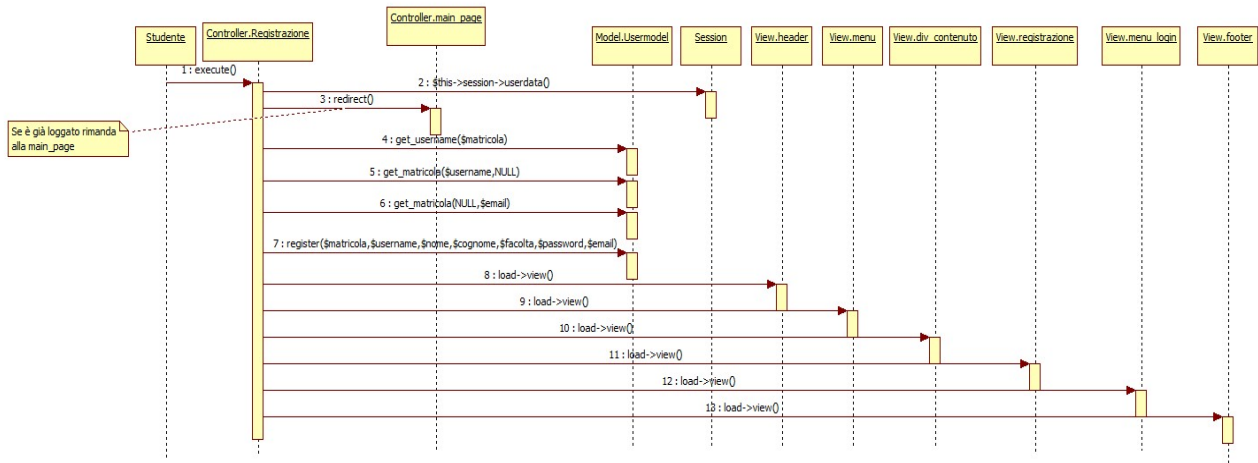
Il controller Login riceve come input i parametri username e password che il sistema verifica.

Se tutto è corretto il sistema istanzia una sessione definendo 3 variabili (differenti a seconda che si tratti di un gestore o di uno studente):

- per lo studente vengono inizializzate: logged_in, matricola_studente, num_appunti
- per il gestore vengono inizializzate: logged_in_gestore, id_gestore

Per quanto riguarda la terza variabile essa serve in un secondo momento: per il momento viene solo settata a 0.

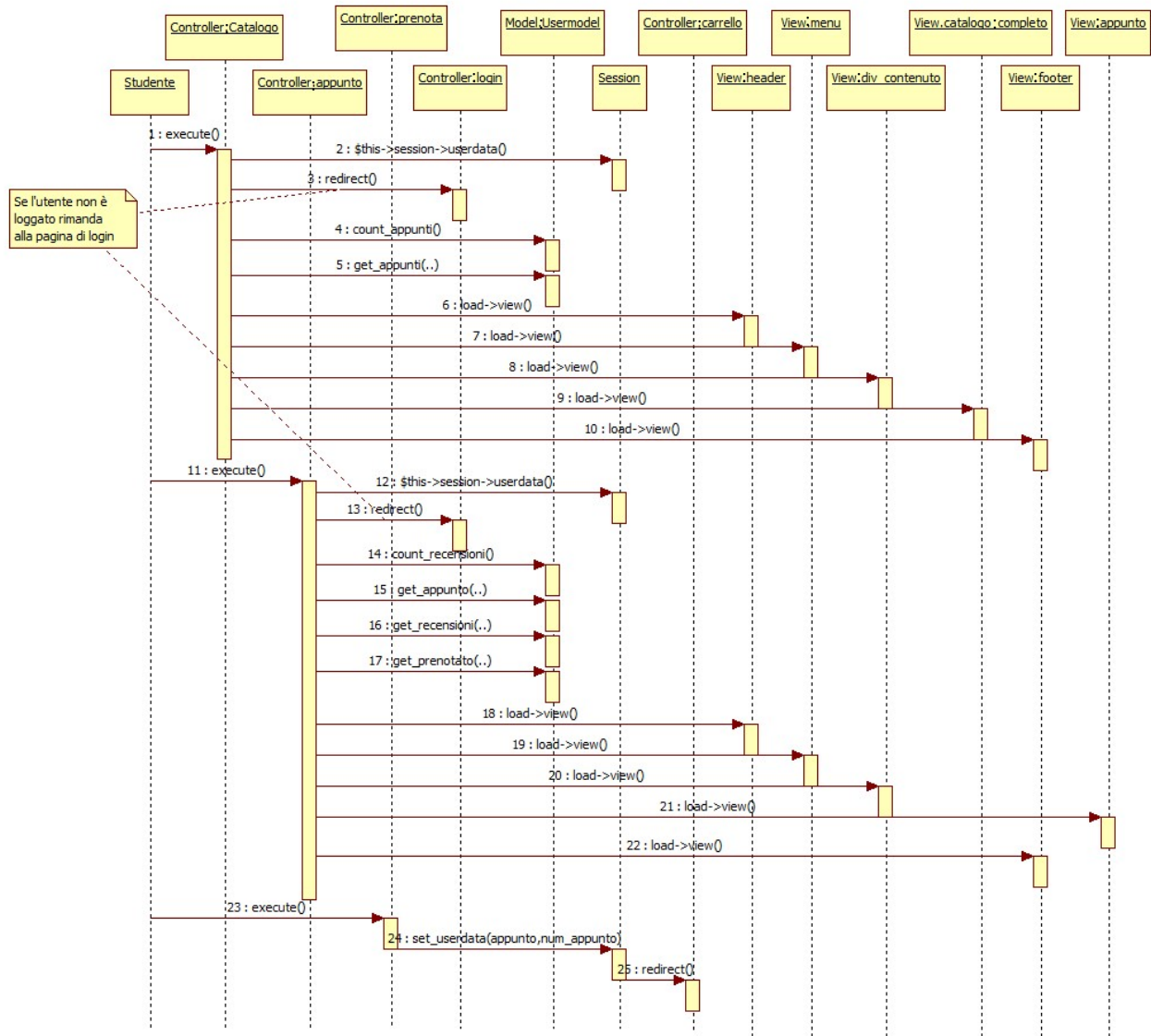
2) Registrazione di uno studente al servizio



Se l'utente si è già loggato e prova ad accedere alla pagina di registrazione viene rimandato alla propria main page (sia che si tratti di studente, sia che si tratti di gestore).

Prima di effettuare la registrazione controlliamo che non ci sia nessuno studente registrato con la stessa matricola, o con lo stesso username o con la stessa email.

3) Richiedere un appunto



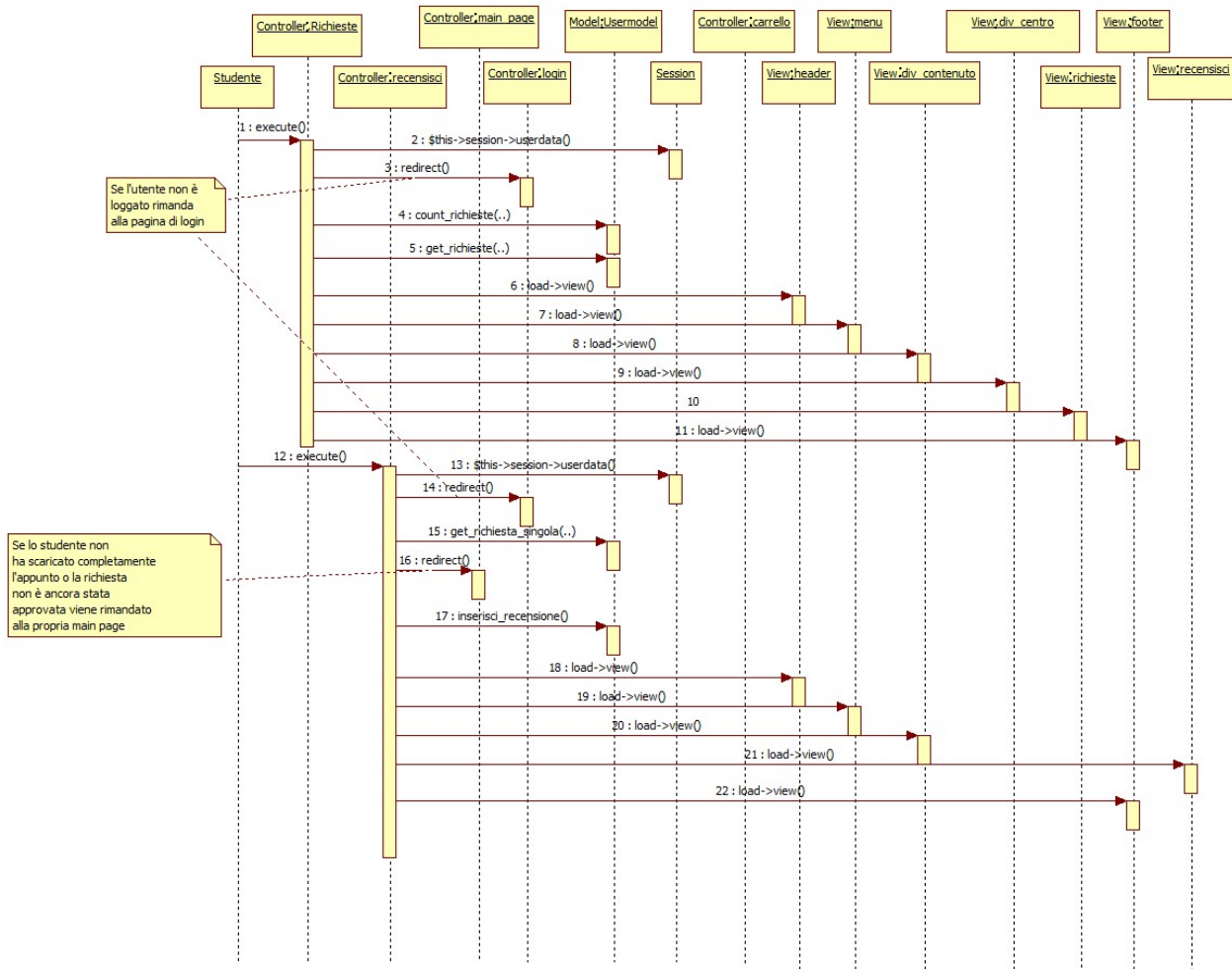
Lo studente interviene 3 volte:

- la prima volta è quando richiede di visualizzare l'elenco di tutti gli appunti disponibili
- la seconda è quando richiede di visualizzare le informazioni dettagliate relative all'appunto
- la terza è quando richiede di prenotare l'appunto.

In tutti e tre i casi il sistema controlla che l'utente sia realmente loggato, caso contrario lo rimanda alla pagina di login (nella terza interazione non si è riportato il controllo solo per non appesantire la documentazione).

Quando lo studente richiede di prenotare un appunto, il sistema salva in delle variabili di sessione quali e quanti appunti vuole prenotare e solo dopo aver confermato il carrello della spesa questi valori vengono registrati nel database.

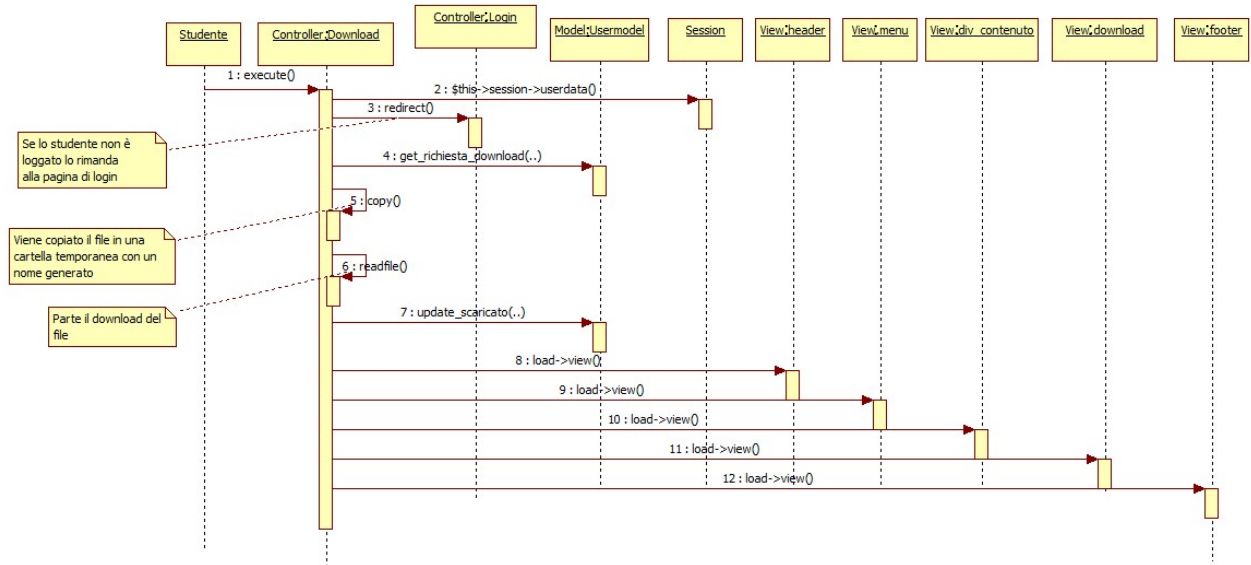
4) Inserire la recensione di un appunto



Uno studente può inserire una recensione solo se la richiesta è stata accettata e se ha scaricato completamente l'appunto. Quindi prima di rimandarlo alla pagina di inserimento di una recensione si controlla che tutte queste condizioni siano verificate e solo dopo si permette di inserirne una.

Inoltre lo studente può inserire una sola recensione per ogni appunto, quindi nella pagina con l'elenco di tutte le richieste viene stampato il pulsante per inserire una recensione solo se lo studente non l'ha già inserita.

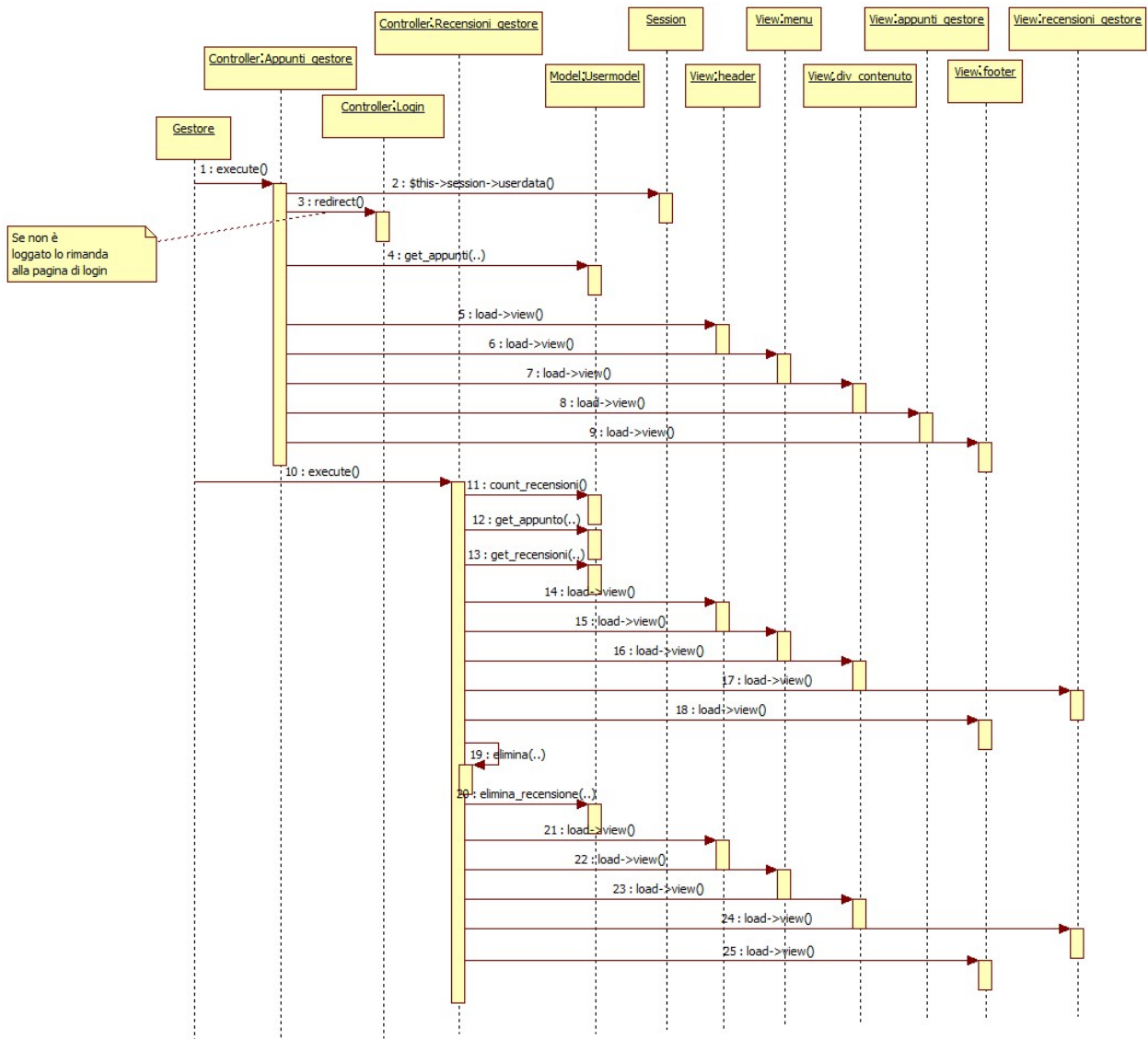
5) Download di un appunto



Prima di far partire il download il sistema controlla che l'utente abbia effettuato il login.

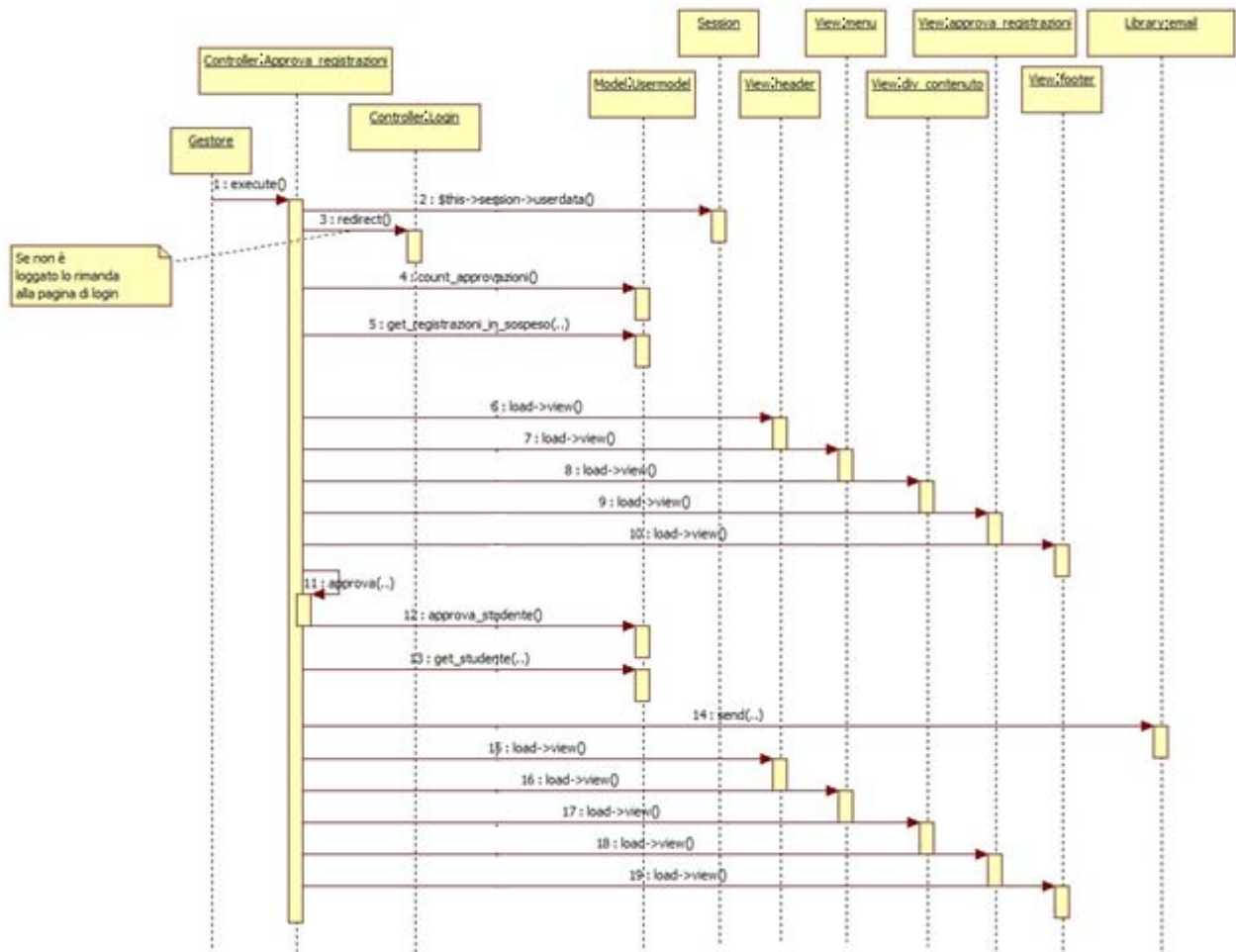
Il database viene aggiornato solo se il download viene completato, altrimenti figura che lo studente non abbia mai finito di scaricare l'appunto e di conseguenza non gli fa inserire recensioni.

6) Eliminare una recensione



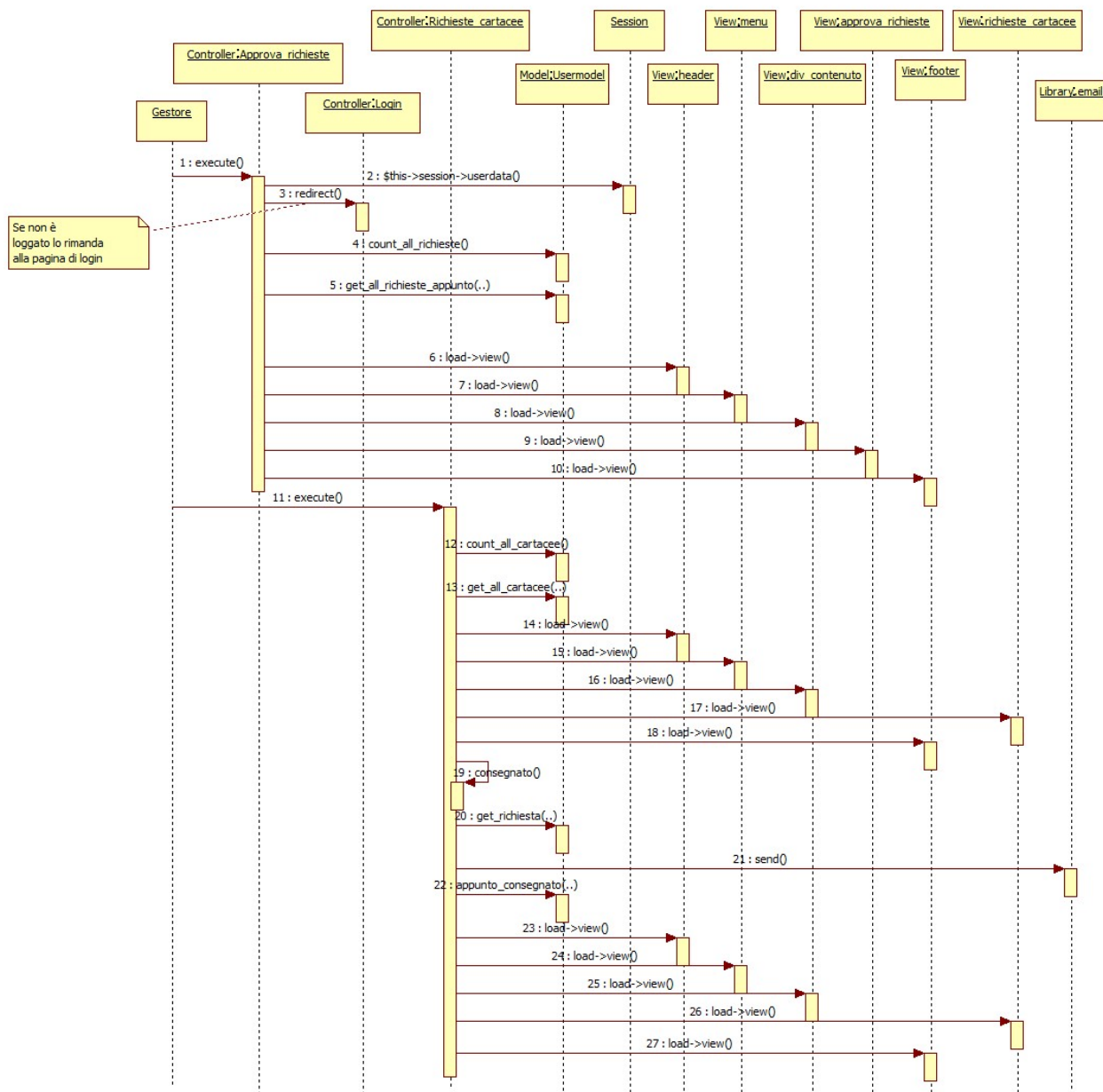
Per non appesantire troppo il diagramma si è volutamente tralasciato il controllo dell'avvenuto login nel secondo caso: quando l'utente seleziona la funzionalità "Recensioni".

7) Approvare la registrazione di uno studente



L'email viene inviata sia nel caso in cui è stata richiesta la versione cartacea dell'appunto, sia nel caso in cui si è richiesta la copia digitale. Nel primo caso, però, invece di inviare un link, si invia un'invito a contattare il gestore per ottenere un appuntamento.

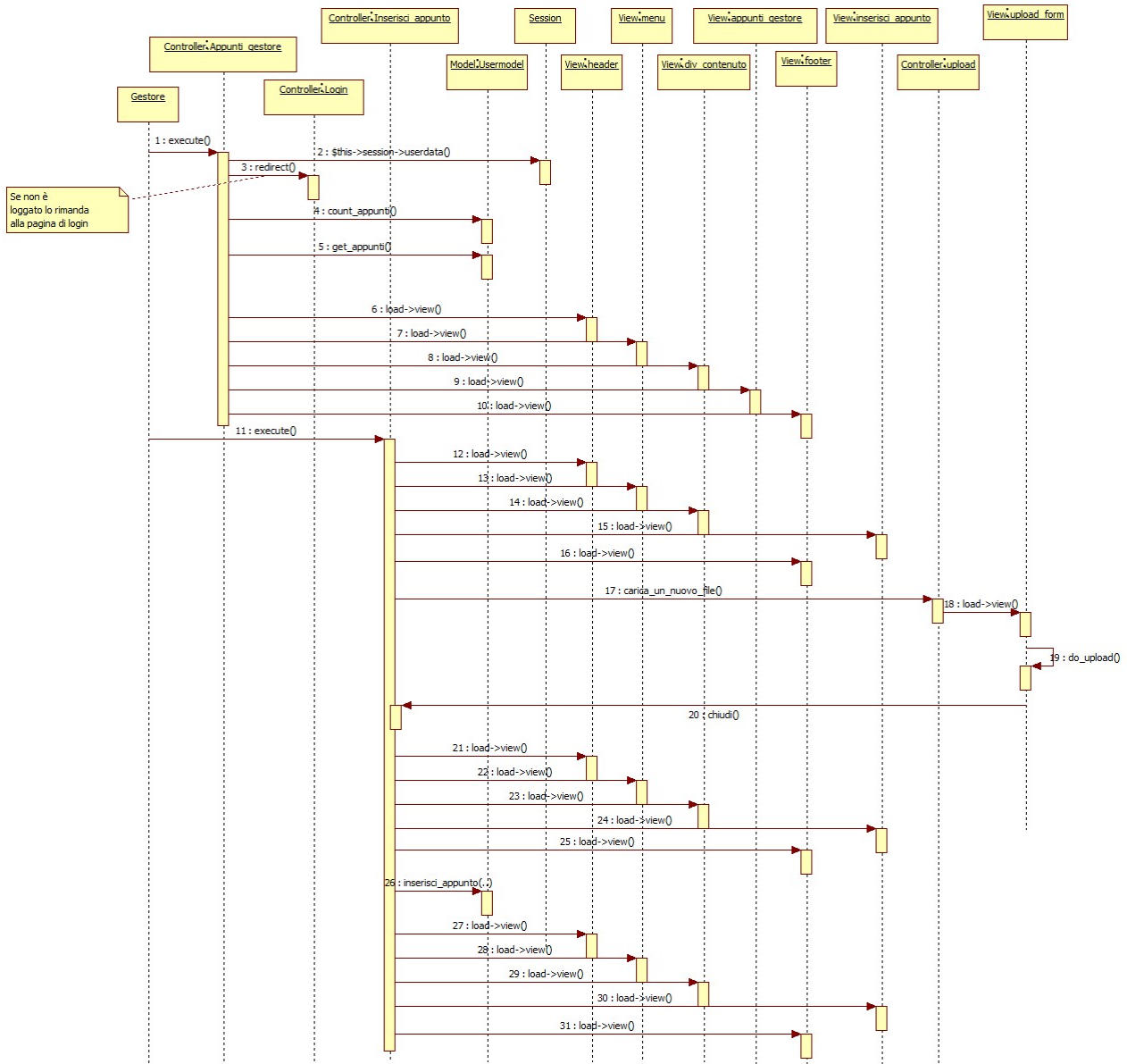
8) Confermare l'avvenuta consegna di un appunto cartaceo



Per non appesantire troppo il diagramma si è volutamente tralasciato il controllo dell'avvenuto login nel secondo caso: quando l'utente seleziona la funzionalità "Cartacee da confermare".

Si preferisce inviare l'email prima di aggiornare il database perché in questo modo se qualcosa va storto il gestore può ripetere l'operazione, mentre se prima si aggiornasse il database, il gestore non vedrebbe più la richiesta in sospeso.

9) Inserire un nuovo appunto



Per non appesantire troppo il diagramma si è volutamente tralasciato il controllo dell'avvenuto login nel secondo caso: quando l'utente seleziona la funzionalità "Inserisci nuovo appunto".

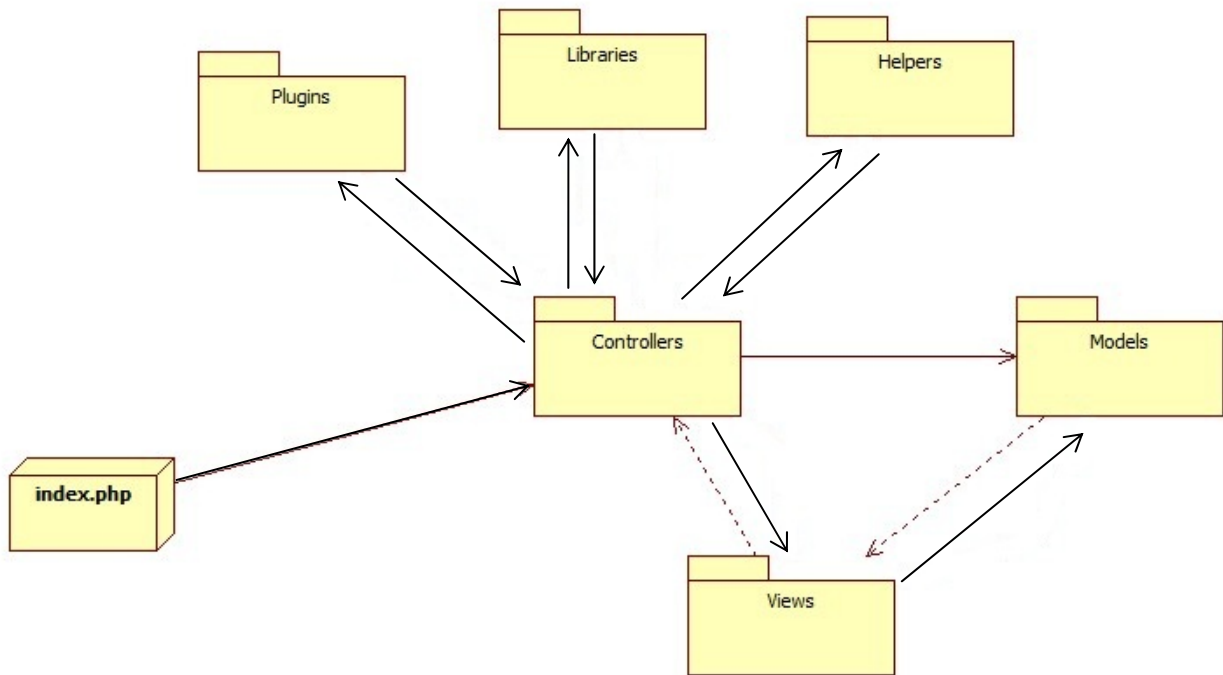
Per il caricamento di un nuovo file viene caricata un'ulteriore interfaccia in un popup che viene chiuso subito dopo aver effettuato il download del file e aver premuto il tasto Chiudi.

Se prima di confermare le modifiche si cambia file, il vecchio file viene cancellato prima di caricare il nuovo. Il nome del file viene infatti conservato in un COOKIE in modo da conoscere sempre il nome del vecchio file.

3.6. DIAGRAMMA di PACKAGE

I diagrammi di package sono utilizzati per mostrare i vari blocchi logici in cui è suddiviso il sistema e le dipendenze che relazionano questi gruppi.

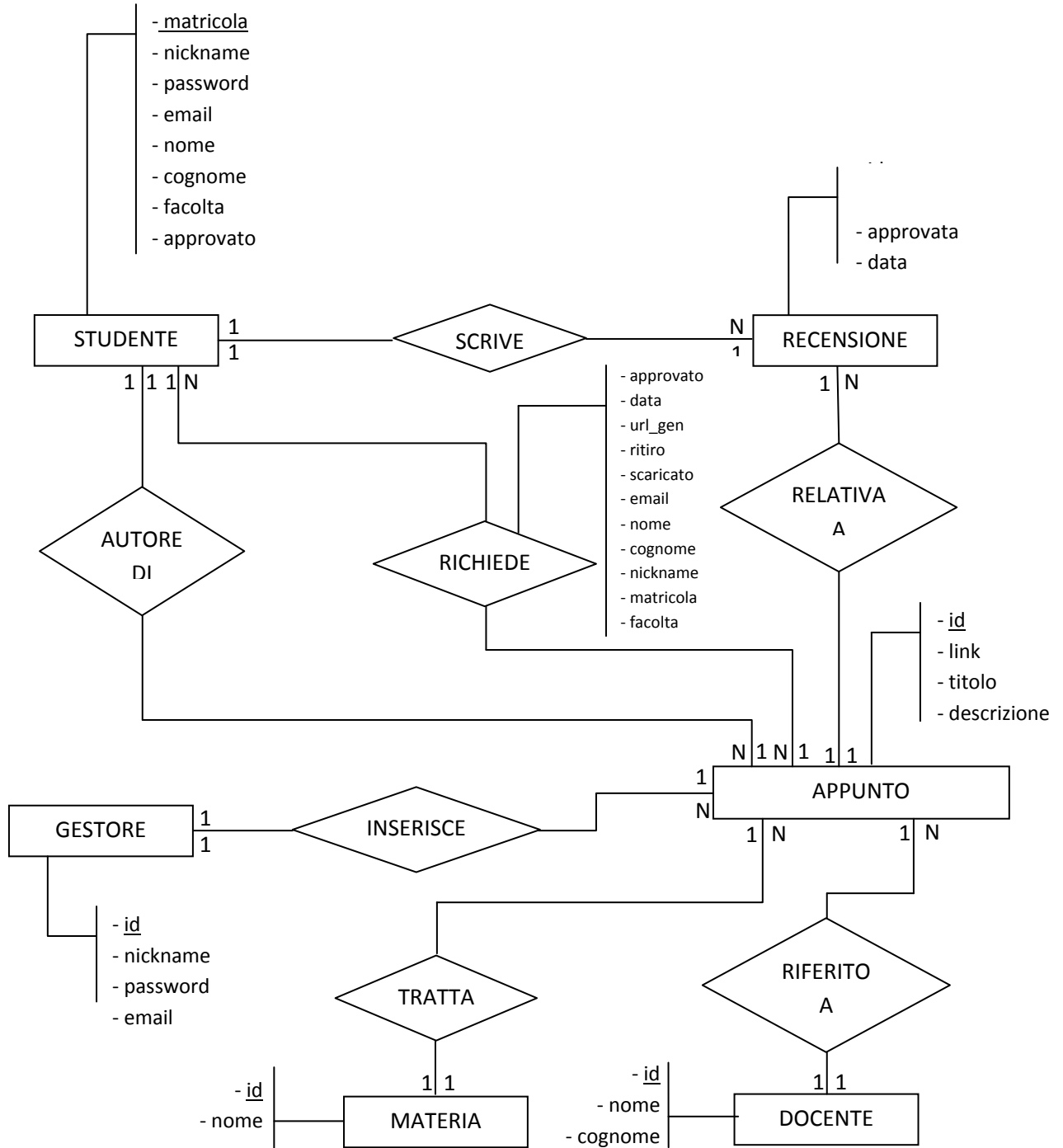
Un package in genere contiene più classi, quindi una dipendenza tra due package esiste se c'è fra le classi costituenti i package.



- `index.php` lavora come il controller principale, inizializza le risorse base necessarie per il corretto funzionamento del framework CodeIgniter.
- Il Controller è quello che carica tutti i Moduli, le Librerie, gli Helpers e qualsiasi altra risorsa necessaria per soddisfare la specifica richiesta
- I Models sono quelli che si occupano della gestione dei dati (interfacciamento con il database e tutto ciò che concerne i dati)
- Le Views sono responsabili dell'interfaccia utente. Il controller prima fa i controlli ed istanzia le risorse, poi si preoccupa di acquisire i dati necessari utilizzando i modelli ed in seguito carica le Views passandogli tutti i dati precedentemente acquisiti
- Le librerie, i plugins e gli Helpers servono come supporto al controller. Al loro interno sono implementate tutte le classi più spesso usate nella creazione di pagine web dinamiche.

4. Database

4.1. Diagramma E-R



4.2. Dizionario dei dati

STUDENTE

Indica la persona che utilizza il servizio. Gli attributi che descrivono questa entità sono:

- **matricola:** indica la matricola (codice identificativo di uno studente all'interno della facoltà) dello studente
- **nickname:** è lo pseudonimo usato dallo studente per utilizzare il servizio
- **password:** è il codice segreto necessario per accedere al servizio
- **email:** indica l'e-mail dello studente
- **nome:** indica il nome dello studente
- **cognome:** indica il cognome dello studente
- **facolta:** indica la facoltà alla quale lo studente iscritto
- **approvato:** indica se lo studente è abilitato ad utilizzare il servizio

GESTORE

Indica la persona responsabile di amministrare il sistema. Gli attributi che descrivono questa entità sono:

- **id:** indica nel numero identificativo del gestore
- **nickname:** è lo pseudonimo usato dal gestore per amministrare il sistema
- **password:** è il codice segreto necessario per accedere al servizio
- **email:** indica l'e-mail dello studente

RECENSIONE

Conterrà tutte le recensioni scritte dagli studenti dopo aver scaricato completamente l'appunto. Gli attributi che descrivono questa entità sono:

- **id:** indica il numero identificativo del gestore
- **descrizione:** indica il contenuto della recensione
- **approvata:** indica se la recensione è stata approvata dal gestore
- **data:** data di pubblicazione della recensione

APPUNTO

Contiene tutte le informazioni relative agli appunti che l'amministratore inserirà. Gli attributi che descrivono questa entità sono:

- **id:** indica il numero identificativo dell'appunto
- **link:** indica il link all'appunto
- **titolo:** indica il titolo dell'appunto

- descrizione: indica la descrizione dell'appunto

MATERIA

Indica la materia alla quale l'appunto si riferisce. Gli attributi che descrivono questa entità sono:

- id: indica il numero identificativo della materia
- nome: indica il nome (lo stesso riconosciuto nel piano di studi) della materia

DOCENTE

Indica il docente titolare del corso al quale l'appunto si riferisce. Gli attributi che descrivono questa entità sono:

- id: indica il numero identificativo della materia
- nome: indica il nome del docente
- cognome: indica il cognome del docente

4.3.Tipi di relazioni

Per la descrizione dei principali tipi di relazioni sarà utilizzata la seguente notazione:

RELAZIONE(ENTITA'->ENTITA').

Ciò evita equivoci qualora vi siano duplicazioni di nomi.

AUTORE DI (STUDENTE -> APPUNTO)

È un tipo di associazione 1:N

La relazione permette di inquadrare lo studente che redige un appunto.

RICHIEDE (STUDENTE -> APPUNTO)

È un tipo di associazione N:N

La relazione permette di inquadrare lo studente che richiede un appunto.

Ci sono inoltre degli attributi che descrivono tale relazione:

- approvato: indica se la richiesta è stata approvata
- data: indica la data in cui si è fatta richiesta
- url_gen: indica il link autogenerato che permette allo studente di scaricare l'appunto
- ritiro: indica in che modo lo studente preferisce acquisire l'appunto: in forma cartacea o in forma digitale (download)

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

- scaricato: indica se l'appunto è stato veramente scaricato
- matricola: indica la matricola specificata dallo studente al momento della prenotazione dell'appunto
- nickname: indica il nickname specificata dallo studente al momento della prenotazione dell'appunto
- nome: indica il nome specificato dallo studente al momento della prenotazione dell'appunto
- cognome: indica il cognome specificato dallo studente al momento della prenotazione dell'appunto
- email: indica l'email specificata dallo studente al momento della prenotazione dell'appunto

SCRIVE (STUDENTE -> RECENSIONE)

È un tipo di associazione 1:N

La relazione permette di inquadrare lo studente che scrive una recensione

RELATIVA A (APPUNTO -> RECENSIONE)

È un tipo di associazione 1:N

La relazione permette di capire a quale appunto si riferisce la recensione

INSERISCE (GESTORE -> APPUNTO)

È un tipo di associazione 1:N

La relazione permette di inquadrare la data di inserimento di un appunto, infatti vi è l'attributo:

- data: indica la data in cui è stato inserito l'appunto

TRATTA (APPUNTO -> MATERIA)

È un tipo di associazione N:1

La relazione permette di inquadrare la materia alla quale l'appunto si riferisce

RIFERITO A (APPUNTO -> DOCENTE)

È un tipo di associazione N:1

La relazione permette di inquadrare il docente al quale l'appunto si riferisce

5. Collaudo del sistema

In questa sezione si descrivono i procedimenti, le strategie e le metodologie usate per organizzare, pianificare, eseguire e gestire il testing del sistema software.

Gli obiettivi che si vogliono raggiungere con il collaudo del sistema sono:

1. Verificare che si siano implementate tutte le funzionalità dichiarate nella specifica dei requisiti
2. Verificare che il software soddisfi alcuni requisiti di qualità

1) Metodi di collaudo usati

Le tecniche utilizzate per collaudare il sistema sono principalmente 2:

- a) Test di unità, che ci hanno consentito di verificare il corretto funzionamento delle funzioni di interfacciamento con il database
- b) Test funzionali, che basandosi esclusivamente sulle specifiche, ci hanno permesso di verificare il corretto funzionamento del sistema e la robustezza dello stesso

I test di unità sono stati implementati man mano che venivano dichiarate le funzioni da testare, mentre i test funzionali sono stati implementati al termine della fase di sviluppo.

2) Test di unità

Per rispettare il modello MVC, le funzioni che implementano l'interfacciamento del database si preoccupano esclusivamente di salvare/modificare/cancellare dati dal/sul database.

I controlli sulla correttezza dei dati inseriti, i controlli sull'esistenza dei dati da visualizzare e tutte le altre operazioni di verifica sui dati estratti dal database vengono implementati nei controllers che, per come è progettato il framework PHP Codeigniter possono essere testati esclusivamente richiamando il controller stesso.

Per tali motivi, i test di unità sono stati implementati esclusivamente sulle classi che estraggono dati dal database.

Per non modificare in maniera errata il database o la classe di interfacciamento, è stato duplicato il database in un database appositamente creato per il test (testappunti).

E' stato inoltre creato un apposito controller che permette tramite una semplice pressione di un tasto di eseguire il test e restituire a video i risultati.

Per realizzare i test di unità, si è fatto uso della libreria messa a disposizione dal Framework PHP CodeIgniter: **"unit_test"**.

Di seguito vengono elencati i test di unità più significativi implementati nel controller di test:

- a) Test login con combinazione username-password non esistente
- b) Test login con username con caratteri non ammessi
- c) Test login con username e password con caratteri non ammessi
- d) Test login con username e password CORRETTI
- e) Test login gestore con username e password con caratteri non ammessi
- f) Test verifica utente approvato (con username e password di utente NON approvato)
- g) Test verifica utente approvato (con username e password di utente APPROVATO)
- h) Test verifica utente approvato (con username e password non esistenti)
- i) Test get_username (con matricola non esistente)
- j) Test get_username (con matricola Esistente)
- k) Test registrazione (con matricola già esistente)
- l) Test registrazione (con matricola non ancora esistente)
- m) Test get_appunto (con id appunto non esistente)
- n) Test get_appunto (con id appunto esistente)
- o) Test get_recensioni (con id appunto NON esistente)
- p) Test get_recensioni (con id appunto esistente)
- q) Test get_autore (con id appunto NON esistente)
- r) Test get_autore (con id appunto esistente)
- s) Test get_prenotato (con id appunto e matricola non corrispondenti)
- t) Test get_prenotato (con id appunto e matricola corrispondenti)
- u) Test get_studente (con matricola Esistente)
- v) Test get_richieste (con matricola NON esistente)
- w) Test get_richiesta_singola (con matricola e id_appunti non corrispondenti)
- x) Test get_richiesta_singola (con matricola e id_appunti Corrispondenti)
- y) Test get_recensione (con matricola e id_appunti NON corrispondenti)
- z) Test get_recensione (con matricola e id_appunti Corrispondenti)

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

- aa) Test get_docente (con cognome nome docente NON esistente)
- bb) Test get_docente (con cognome nome docente Esistente)
- cc) Test get_materia (con nome materia NON esistente)
- dd) Test get_materia (con nome materia Esistente)
- ee) Test get_richiesta_download (con url_gen e matricola NON corrispondenti)
- ff) Test get_richiesta_download (con url_gen e matricola Corrispondenti)

3) Test funzionali: test case

Per verificare la robustezza, l'affidabilità, la correttezza e l'usabilità dell'applicazione si sono elaborati alcuni test case.

E' scontato che allo stato attuale (stato di consegna dell'elaborato) i test case riportino tutti un riscontro positivo, ma nelle varie fasi implementative alcuni di questi hanno permesso di risolvere alcuni problemi che non avevamo notato in fase di implementazione.

Di seguito si riportano i test case di maggiore interesse.

- a) Registrazione nuovo utente con i seguenti dati:
 - Matricola già esistente
 - Username già esistente
 - Email già esistente
 - Dati mancanti nella compilazione del modulo
 - Caratteri non ammessi nello username / nome / cognome / matricola
 - Nome utente più lungo del previsto
 - Nome utente più corto del previsto
 - Password più lunga del previsto
 - Email non valida

- b) Recupero password:
 - Matricola non esistente
 - Email e matricola non corrispondenti

c) Modifica dati personali:

- Dati mancanti
- Dati non corretti
- email già usata da un altro utente

d) Dettagli appunto:

- Generazione preview
- Generazione preview di un appunto non esistente
- Prenotazione appunto non disponibile
- Prenotazione appunto già prenotato

e) Carrello

- Conferma carrello vuoto
- Provare a inserire i dati per conferma carrello anche se il carrello è vuoto
- Inserimento dati non conformi alle regole imposte

f) Richieste e recensioni

- Provare a inserire una recensione per un appunto per il quale non si è ancora ottenuta l'autorizzazione per il download
- Provare a inserire una recensione per un appunto per il quale si è ottenuta l'autorizzazione per il download ma non si è ancora realmente scaricato
- Provare a inserire una recensione per un appunto per il quale non si è ottenuta l'autorizzazione per il download: autorizzazione negata

g) Invio appunto

- Dati mancanti
- Dati non corretti
- File da inviare non nel formato pdf
- File da inviare di dimensione superiore a quella consentita

h) Logout

- Effettuare il logout anche se non si è loggati
- Verificare che le variabili di sessione siano state realmente eliminate

- i) Login Gestore (gestore)
 - Entrare nell'area riservata al gestore con dati di un utente normale
 - Password del gestore errata
- j) Inserimento nuovo appunto (gestore)
 - Informazioni mancanti nel form da compilare
 - Informazioni non corrette
 - File da caricare non nel formato pdf
- k) Modifica appunto esistente (gestore)
 - Informazioni mancanti nel form da compilare
 - Informazioni non corrette
 - File da caricare non nel formato pdf
- l) Recensioni (gestore)
 - Eliminazione recensione non esistente
 - Approvazione recensione già approvata
 - Disapprovazione recensione già disapprovata
- m) Gestione studenti (gestore)
 - Approvare studente che è già autorizzato
- n) Recensioni (gestore)
 - Approvare una recensione che è già stata approvata
 - Approvare una recensione che non esiste
 - Eliminare una recensione che non esiste
- o) Richieste (gestore)
 - Approvare recensione che non esiste
 - Disapprovare recensione che non esiste
 - Confermare recensione che non è cartacea
- p) Inserimento / modifica / eliminazione Materia (gestore)
 - Informazioni mancanti nel form da compilare
 - Informazioni non corrette
 - Eliminazione materia collegata con qualche appunto

- q) Inserimento / modifica / eliminazione Docente (gestore)
- Informazioni mancanti nel form da compilare
 - Informazioni non corrette
 - Eliminazione docente collegato con qualche appunto

6. Tecnologie usate

Per la realizzazione di questa applicazione sono stati utilizzati diversi framework e diversi strumenti software che hanno permesso una semplificazione del lavoro:

- **PHP Codeigniter:** è un framework open source ideato per lo sviluppo di applicazioni Web. Esso si basa sul design pattern MVC (Model View Controller) che permette di separare la logica di business dall'interfaccia utente.
- **PHP:** è un linguaggio di scripting opensource concepito per la progettazione di pagine web dinamiche. Attualmente la versione 5 è la più aggiornata, ma PHP Codeigniter lavora con PHP versione 4, quindi per l'intero progetto è stata utilizzata la versione 4.
- **MySQL:** E' un database di tipo relazionale, cioè che organizza i dati in maniera tabellare e usa il linguaggio SQL per operare sui dati.
- **Apache HTTP Server:** software che realizza le funzioni di trasporto delle informazioni, di internetwork e di collegamento, ha il vantaggio di offrire anche funzioni di controllo per la sicurezza come quelli che compie il proxy.
- **ImageMagick:** è una suite software che permette di creare, modificare, comporre o convertire immagini bitmap. E' in grado di leggere, e quindi successivamente convertire in moltissimi formati, compreso il formato pdf.

Proprio perchè permette di convertire pdf in immagini jpeg è stato scelto per la creazione delle preview degli appunti. Esso consente inoltre di convertire una singola pagina piuttosto che l'intero documento, e si interfaccia molto bene con il php. La maggior parte dei webserver include questa suite tra le feature preinstallate.

7. Scrum “sprint backlog” e “burndown chart”

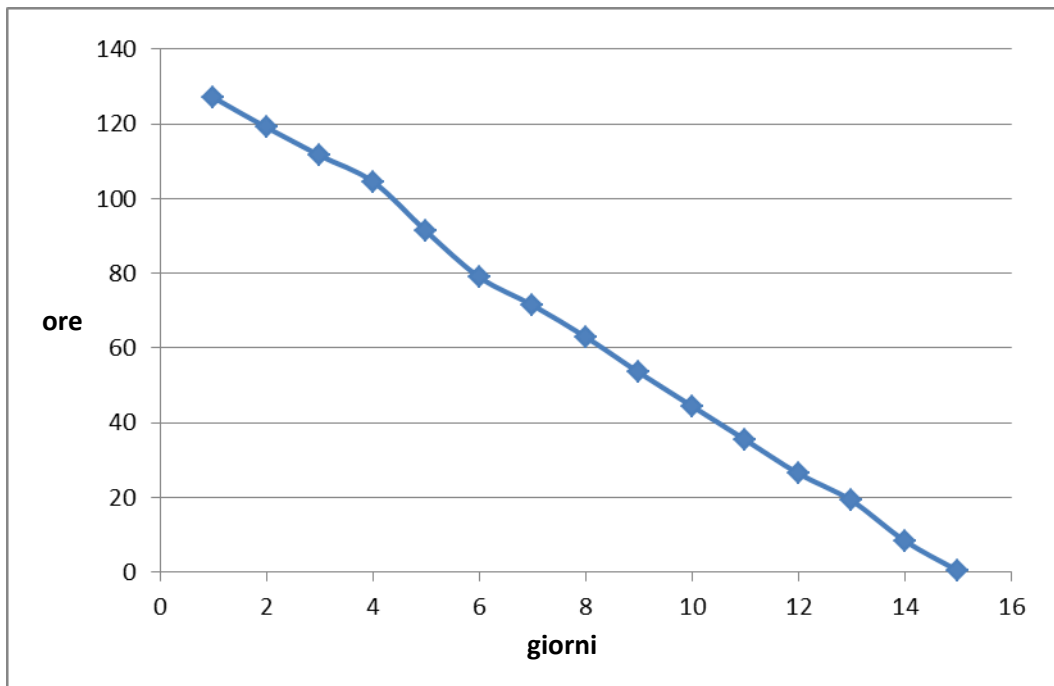
Dal momento che si prevede una durata del progetto pari circa ad un mese, si prevede uno sprint ogni 2 settimane.

Di seguito riporto lo sprint backlog relativo ai 2 sprint con le funzionalità incluse nelle varie voci.

Sprint backlog 1

Task	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no	Gi or no
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Configurazione del framework PHP CodeIgniter	2													
Implementazione del database	4													
Implementazione dell'interfaccia grafica	2	4												
Implementazione di tutti i casi d'uso relativi ad un utente generico		2	6	6	5									
Implementazione di tutti i casi d'uso relativi ad un utente studente						5	5	5	5	5	5	5	5	5
Scrittura classe interfacciamento con database		0,5	1	1	1	1	1	1	1	1	1	1	1	1
Scrittura dei commenti		1	1	1	1	1	1	1	1	1	1	1	1	1
Scrittura della documentazione					2									
Scrittura unit test		1	1	1	1	0,5	1	1	1	1	1	1	1	1
Test delle parti implementate		0,5	0,5	0,5	1		1	1	1	1	1	1	1	0,5
Correzione del codice					1									

Burndown chart 1



	totale ore giornaliere ipotizzate	totale ore giornaliere reali	differenza	totale ore mancanti
1	8	8	0	127
2	9	10,5	1,5	119
3	9,5	13,5	4	111,5
4	9,5	9,75	0,25	104,5
5	12	12	0	91,25
6	7,5	7,5	0	79
7	9	9,5	0,5	71,5
8	9	9	0	63
9	9	9	0	53,5
10	9	9	0	44,5
11	9	9	0	35,5
12	9	10,75	1,75	26,5
13	9	9	0	19,25
14	8,5	9	0,5	8,5
15				0,5

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Per realizzare il burndown chart è stato necessario tenere traccia delle attività da compiere e compiute ogni giorno.

Di seguito riporto ciò che, se avessi lavorato in team, sarebbe venuto fuori dalla riunione giornaliera (Daily scrum).

Giorno 1

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
1. Configurazione del framework PHP CodeIgniter	1. Configurazione del framework PHP CodeIgniter	2	1,5
2. Implementazione del database	1. Implementazione del database	4	4,5
3. Implementazione dell'interfaccia grafica	1. Implementazione dell'interfaccia grafica	2	2

Giorno 2

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Implementazione dell'interfaccia grafica	1. Implementazione dell'interfaccia grafica	4	5
Realizzazione della pagina con il modulo per la registrazione al servizio (+ commenti, unit test, test funzionali)	1. Implementazione di tutti i casi d'uso relativi ad un utente generico	2	2
	2. Scrittura classe interfacciamento con database	0,5	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	0,5	0,5

Giorno 3

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Implementazione pagina registrazione	1. Implementazione di tutti i casi d'uso relativi ad un utente generico	3	4,5
	2. Scrittura classe interfacciamento con database	0,5	1
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,25	0,5
Implementazione pagina richiesta nuova password	1. Implementazione di tutti i casi d'uso relativi ad un utente generico	3	4
	2. Scrittura classe interfacciamento con database	0,5	1
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,25	0,5

Giorno 4

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Implementazione pagina login	1. Implementazione di tutti i casi d'uso relativi ad un utente generico	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	0,25	0,5
Realizzazione pagina con termini di utilizzo	1. Implementazione di tutti i casi d'uso relativi ad un	1	1

	utente generico		
	2. Scrittura classe interfacciamento con database	0	0
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0	0
	5. Test delle parti implementate	0,25	0,25

Giorno 5

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione catalogo degli appunti non completo: caso in cui nn è loggato (anche qui differenziamo caso in cui è loggato da caso in cui non è loggato)	1. Implementazione di tutti i casi d'uso relativi ad un utente generico	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	1	1
Correzione del codice	1. Correzione del codice	1	1
Scrittura documentazione	1. Scrittura documentazione	2	2

Giorno 6

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione home page studente loggato	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	0,5	0,5

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Giorno 7

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione della parte di catalogo dedicata agli studenti loggati (in aggiunta ci sarà un pulsante per visionare i dettagli degli appunti)	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	3	3
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	0,5	1
Realizzazione menu per lo studente loggato	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0	0
	5. Test delle parti implementate	0,5	0,5

Giorno 8

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione Dettagli appunti	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	4	4
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	0,75	0,75

Realizzazione LOGOUT	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	1	1
	2. Scrittura classe interfacciamento con database	0	0
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0	0
	5. Test delle parti implementate	0,25	0,25

Giorno 9

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione di un popup che mostri la preview dell'appunto	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	1	1

Giorno 10

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione di parte che nei dettagli dell'appunto mostra tutte le recensioni rilasciate per l'appunto	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	1	1

Giorno 11

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione del carrello	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	1	1

Giorno 12

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Realizzazione della pagina di conferma del carrello con scelta del metodo di consegna dell'appunto	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	3	3
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0,5	1
	5. Test delle parti implementate	0,75	0,75
Realizzazione della pagina per l'invio di un appunto	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0,5	1
	5. Test delle parti implementate	0,25	1

Giorno 13

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Definizione della pagina con il riepilogo delle richieste effettuate per un appunto (sia convalidate che non convalidate)	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	5	5

	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	1	1
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	1	1

Giorno 14

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Definizione della pagina che permette di inserire delle recensioni per un appunto (con funzioni per inserimento nel database)	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	4	4
	2. Scrittura classe interfacciamento con database	0,75	0,75
	3. Scrittura dei commenti	0,75	0,75
	4. Scrittura unit test	1	1
	5. Test delle parti implementate	0,5	1
Definizione funzione per generazione link	1. Implementazione di tutti i casi d'uso relativi ad un utente studente	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0	0
	5. Test delle parti implementate	0	0

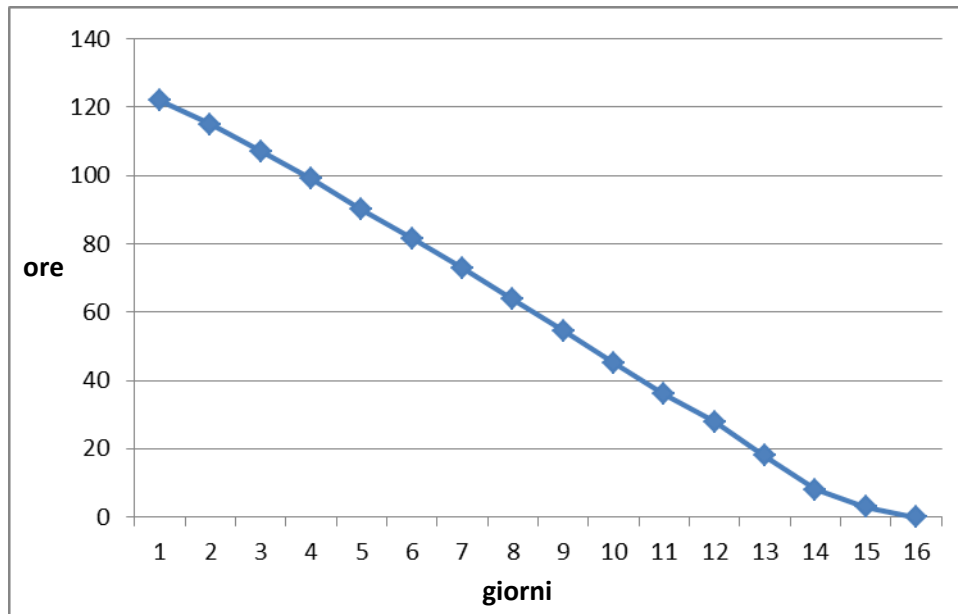
Sprint backlog 2

Task	Giorno 1	Giorno 2	Giorno 3	Giorno 4	Giorno 5	Giorno 6	Giorno 7	Giorno 8	Giorno 9	Giorno 10	Giorno 11	Giorno 12	Giorno 13	Giorno 14
Implementazione di tutti i casi d'uso relativi ad un utente gestore			2	5	5	5	5	5	5	5	5			
Scrittura classe interfacciamento con database			0,5	1	1	1	1	1	1	1	1			
Scrittura dei commenti			0,5	1	1	1	1	1	1	1	1			
Scrittura della documentazione	2		1								2	2	2	2
Scrittura unit test			0,5	1	1	1	1	1	1	1	0,5			
Test delle parti implementate (test funzionali)	5	2	5	1	1	1	1	1	1	1	1	6		7
Correzione del codice		6											7	

Per realizzare il burndown chart è stato necessario tenere traccia delle attività da compiere e compiute ogni giorno.

Di seguito riporto ciò che, se avessi lavorato in team, sarebbe venuto fuori dalla riunione giornaliera (Daily scrum).

Burndown chart 2



	totale ore giornaliere ipotizzate	totale ore giornaliere reali	differenza	totale ore mancanti
1	7	7	0	122
2	8	8	0	115
3	8,5	9	0,5	107
4	8,5	8,5	0	99
5	8,5	8,5	0	90
6	9,25	10	0,75	81,5
7	9,25	10	0,75	73
8	9,25	10	0,75	63,75
9	9,25	10	0,75	54,5
10	8,5	8,5	0	45,25
11	10	12	2	36
12	8	8	0	28
13	9	8	-1	18
14	9	12	3	8
15				3

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Per realizzare il burndown chart è stato necessario tenere traccia delle attività da compiere e compiute ogni giorno.

Di seguito riporto ciò che, se avessi lavorato in team, sarebbe venuto fuori dalla riunione giornaliera (Daily scrum).

Giorno 1

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Scrittura della documentazione relativa alla parte già implementata	1. Scrittura della documentazione	2	2
Test funzionali	1. Test delle parti implementate	5	5

Giorno 2

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Test funzionali	1. Test delle parti implementate	2	2
Correzione del codice	1. Correzione del codice	6	6

Giorno 3

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Login e autenticazione gestore	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	2
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Creazione della main page del gestore	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25

	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Test delle correzioni effettuate il giorno prima	1. Test delle parti implementate	4,5	4

Giorno 4

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di un menu per il gestore	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5
Creazione di un catalogo appunti diverso per il gestore che consenta la modifica delle voci presenti	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5

Giorno 5

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione della pagina per l'inserimento degli appunti	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5

	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5
Creazione della pagina per la modifica di un appunto	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5

Giorno 6

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di funzione per l'upload di un file	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Creazione di una pagina con tutte le recensioni relative ad appunto (approve e non approve)	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5
Creazione di una pagina dedicata alle recensioni in sospeso	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2

	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5

Giorno 7

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di una pagina con tutte le materie già inserite nel database	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Creazione della pagina che permetta la modifica delle informazioni relative ad una materia	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5
Creazione della pagina per l'inserimento di una nuova materia	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5

	5. Test delle parti implementate	0,375	0,5
--	----------------------------------	-------	-----

Giorno 8

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di una pagina con tutti i docenti già inseriti nel database	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Creazione della pagina che permetta la modifica delle informazioni relative ad un docente	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5
Creazione della pagina per l'inserimento di un nuovo docente	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5

Giorno 9

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
LOGOUT gestore	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	1	1
	2. Scrittura classe interfacciamento con database	0,25	0,25
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,25	0,25
	5. Test delle parti implementate	0,25	0,25
Creazione di una pagina contenente tutte le registrazioni di studenti in sospeso	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5
Creazione di una pagina contenente tutte le registrazioni di studenti non accettate	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2	2
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,375	0,5
	4. Scrittura unit test	0,375	0,5
	5. Test delle parti implementate	0,375	0,5

Giorno 10

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di una pagina con il riepilogo delle richieste di appunti in sospeso	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5
Creazione di una pagina con il riepilogo delle richieste di appunti rifiutate	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	2,5	2,5
	2. Scrittura classe interfacciamento con database	0,5	0,5
	3. Scrittura dei commenti	0,25	0,25
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	0,5	0,5

Giorno 11

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Creazione di una pagina con il riepilogo delle richieste di appunti che richiedono la consegna cartacea già accettate	1. Implementazione di tutti i casi d'uso relativi ad un utente gestore	5	5
	2. Scrittura classe interfacciamento con database	1	1
	3. Scrittura dei commenti	0,5	0,5
	4. Scrittura unit test	0,5	0,5
	5. Test delle parti implementate	1	1
Scrittura della documentazione	1. Scrittura documentazione	2	4

Giorno 12

Progetto realizzato da Teodoro Montanaro (matricola 10091047)

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Test funzionali	1. Test delle parti implementate	6	6
Scrittura della documentazione	1. Scrittura documentazione	2	2

Giorno 13

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Correzione del codice	1. Correzione del codice	7	6
Scrittura della documentazione	1. Scrittura documentazione	2	2

Giorno 14

Task	Task sprint coinvolti	Ore Previste	Ore impiegate
Test funzionali	1. Test delle parti implementate	7	10
Scrittura della documentazione	1. Scrittura documentazione	2	2