Progetto di TIW

Traccia 3: verbalizzazione esami

POLITECNICO
MILANO 1863

Stefano Bernardotto
MATRICOLA: 212980 – DOCENTE: FRATERNALI PIERO – A.A. 24/25

Sommario

1 – Specifica

1.1 Verbalizzazione degli esami – Versione HTML pura

Un'applicazione permette di verbalizzare gli esiti degli esami di un appello. Il docente accede tramite login e seleziona nella HOME page un corso da una lista dei propri corsi, ordinata per nome del corso in modo alfabetico decrescente, e poi sceglie una data d'appello del corso scelto da un elenco ordinato per data decrescente. Ogni corso ha un solo docente. La selezione dell'appello porta a una pagina ISCRITTI, che mostra una tabella con tutti gli iscritti all'appello. La tabella riporta i seguenti dati: matricola, cognome e nome, email, corso di laurea, voto e stato di valutazione. Il voto può non essere ancora definito. Lo stato di valutazione dello studente rispetto all'appello può assumere i valori: non inserito, inserito, pubblicato, rifiutato e verbalizzato. Selezionando un'etichetta nell'intestazione della tabella, l'utente ordina le righe in base al valore di tale etichetta (ad esempio, selezionando "cognome" la tabella è riordinata in base al cognome). Successive selezioni della stessa etichetta invertono l'ordinamento: si parte con l'ordinamento crescente. Il valore del voto viene considerato ordinato nel modo seguente: <vuoto>, assente, rimandato, riprovato, 18, 19, …, 30, 30 e lode. Nella tabella della pagina ISCRITTI ad ogni riga corrisponde un bottone "MODIFICA". Premendo il bottone compare una pagina con una form che mostra tutti i dati dello studente selezionato e un campo di input in cui è possibile scegliere il voto. L'invio della form provoca la modifica o l'inserimento del voto. Inizialmente le righe sono nello stato di valutazione "non inserito". L'inserimento e le successive eventuali modifiche portano la riga nello stato di valutazione "inserito". Alla tabella della pagina ISCRITTI è associato un bottone PUBBLICA che comporta la pubblicazione delle righe con lo stato di valutazione INSERITO. La pubblicazione rende il voto non più modificabile dal docente e visibile allo studente e cambia lo stato di valutazione della riga dello studente a "pubblicato". Lo studente accede tramite login e seleziona nella HOME page un corso tra quelli a cui è iscritto mediante una lista ordinata in modo alfabetico decrescente e poi una data d'appello del corso scelto selezionata da un elenco ordinato per data decrescente. Uno studente può essere iscritto a più appelli dello stesso corso. [Aggiunta da chiarimento chiesto al docente: la lista mostra tutti gli appelli disponibili per quel corso, se lo studente è iscritto all'appello è possibile visualizzare i dati come specificato in seguito, se non è iscritto, la selezione della data d'appello porta alla pagina ESITO che mostra il messaggio "Studente non iscritto all'appello"]. La selezione della data d'appello porta a una pagina ESITO che mostra il messaggio "Voto non ancora definito" se il docente non ha ancora pubblicato il risultato per quello studente in quell'appello. Altrimenti, la pagina mostra i dati dello studente, del corso, dell'appello e il voto assegnato. Se il voto è tra 18 e 30 e lode compare un bottone RIFIUTA. Premendo tale bottone la pagina mostra gli stessi dati con la dizione aggiunta "Il voto è stato rifiutato" e senza il bottone RIFIUTA. Il rifiuto del voto cambia lo stato di valutazione a "rifiutato" della riga dello studente per quell'appello nella pagina ISCRITTI del docente. Nella pagina ISCRITTI del docente la tabella degli iscritti è associata anche a un bottone VERBALIZZA. La pressione del bottone provoca il cambio di stato a "verbalizzato" per le righe nello stato "pubblicato" o "rifiutato" e comporta anche la creazione di un verbale e la disabilitazione della possibilità di rifiutare il voto. Il rifiuto implica la verbalizzazione di "rimandato" come voto. Un verbale ha un codice generato dal sistema, una data e ora di creazione ed è associato all'appello del corso a cui si riferisce e agli studenti (con nome, cognome, matricola e voto) che passano allo stato "verbalizzato". A seguito della pressione del bottone VERBALIZZA compare una pagina VERBALE che mostra i dati completi del verbale creato. Il docente dispone anche di una pagina VERBALI, in cui sono elencati tutti i verbali da lui creati, ordinati on modo alfabetico crescente per nome del corso e poi per data crescente di appello di ogni corso.

1.2 Verbalizzazione degli esami – Versione con JavaScript

- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina per il docente e un'unica pagina per lo studente.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- La funzione di riordino della tabella degli iscritti è realizzata a lato client.
- La funzione di rifiuto del voto da parte dello studente è realizzata mediante trascinamento del testo corrispondente all'esito (dati dello studente, del corso, dell'appello e voto assegnato) sopra un'icona che rappresenta il cestino. Al rilascio compare un messaggio popup che chiede di confermare l'operazione di rifiuto, con due bottoni (CANCELLA, CONFERMA). Il primo ripristina la situazione precedente al trascinamento, il secondo effettua l'operazione di rifiuto.
- Alla tabella degli iscritti è associato un bottone INSERIMENTO MULTIPLO che provoca la comparsa di una pagina modale con tutte e sole le righe nello stato "non inserito" associate a un campo di input. Il docente può inserire un voto per un insieme delle righe e premere un bottone INVIA che comporta l'invio al server dei voti, il cambio di stato delle righe coinvolte, la chiusura della finestra modale e l'aggiornamento della tabella degli iscritti.

2 – Analisi dell'applicazione

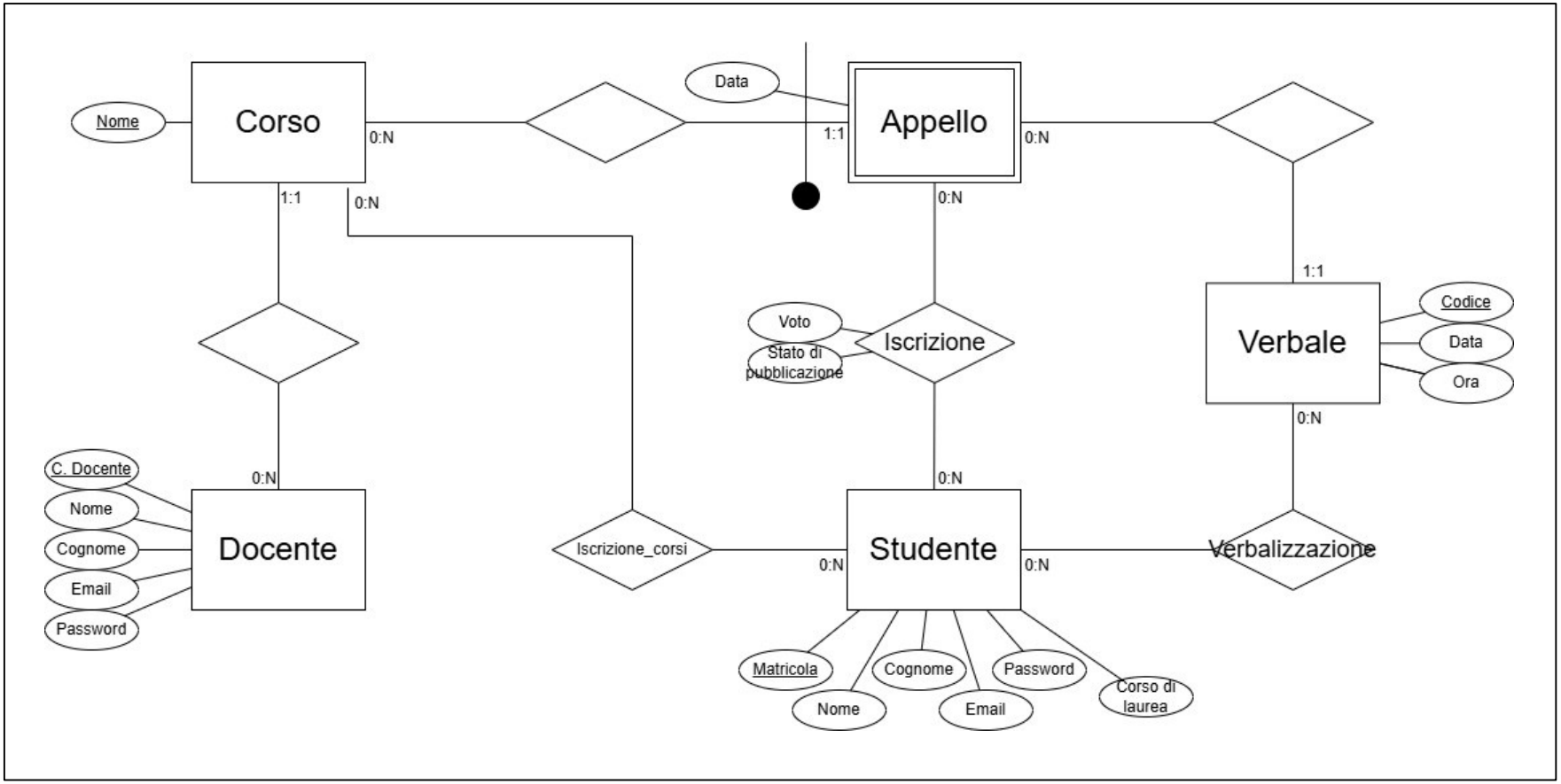
2.1 Analisi dei dati

ANALISI DATI DA SPECIFICA

Legenda: ● Entità ● Attributo ● Relazione

Un'applicazione permette di verbalizzare gli esiti degli esami di un appello. Il docente accede tramite login e seleziona nella HOME page un corso da una lista dei propri corsi, ordinata per nome del corso in modo alfabetico decrescente, e poi sceglie una data d'appello del corso scelto da un elenco ordinato per data decrescente. Ogni corso ha un solo docente. La selezione dell'appello porta a una pagina ISCRITTI, che mostra una tabella con tutti gli iscritti all'appello. La tabella riporta i seguenti dati: matricola, cognome e nome, email, corso di laurea, voto e stato di valutazione. Il voto può non essere ancora definito. Lo stato di valutazione dello studente rispetto all'appello può assumere i valori: non inserito, inserito, pubblicato, rifiutato e verbalizzato. […] Lo studente accede tramite login e seleziona nella HOME page un corso tra quelli a cui è iscritto mediante una lista ordinata in modo alfabetico decrescente e poi una data d'appello del corso scelto selezionata da un elenco ordinato per data decrescente. […] Un verbale ha un codice generato dal sistema, una data e ora di creazione ed è associato all'appello del corso a cui si riferisce e agli studenti (con nome, cognome, matricola e voto) che passano allo stato "verbalizzato". […]

DIAGRAMMA ENTITÀ-RELAZIONI



Nel passaggio dal testo allo schema del database si è ipotizzato che ogni corso abbia nome univoco, inoltre è stato introdotto un codice docente per identificare univocamente ogni docente (è infatti possibile avere docenti omonimi, così come studenti omonimi).

Siccome il popolamento del database è eseguito all'infuori dell'applicazione richiesta, per semplicità le password sono salvate in chiaro. È noto che, in un caso reale, tale pratica sarebbe assolutamente da evitare, bensì le password dovrebbero essere salvate criptate con un algoritmo di hash, applicato dall'applicazione stessa in fase di inserimento.

I codici identificativi dei docenti sono sequenze di 8 cifre decimali, quelli degli studenti sono sequenze di 6 cifre decimali, mentre quelli dei verbali sono la rappresentazione in stringa della classe Java UUID.

COMANDI PER LA CREAZIONE DELLE TABELLE

```
-- Creazione tabella "Docenti":
create table docenti(
    `codice_docente` int not null,
    `nome` varchar(256) not null,
    `cognome` varchar(256) not null,
    `email` varchar(256) not null unique,
    `password` varchar(32) not null,
    primary key(`codice_docente`),
    constraint check(codice_docente > 10000000
        and codice_docente < 99999999)
);
```

```
-- Creazione tabella "Studenti":
create table studenti(
    `matricola` int not null,
    `nome` varchar(256) not null,
    `cognome` varchar(256) not null,
    `email` varchar(256) not null unique,
    `password` varchar(32) not null,
    `corso_laurea` varchar(256) not null,
    primary key(`matricola`),
    constraint check(matricola > 100000
        and matricola < 999999)
);
```

```
-- Creazione tabella "Corsi":
create table corsi (
    `nome` varchar(256) not null,
    `codice_docente` int not null,
    primary key (`nome`),
    foreign key (`codice_docente`)
        references docenti(`codice_docente`)
        on delete no action
        on update cascade
);
```

```
-- Creazione tabella "Iscrizioni_corsi" che rappresenta l'omonima relazione
create table iscrizioni_corsi (
    `nome_corso` varchar(256) not null,
    `matricola_studente` int not null,
    primary key (`nome_corso`, `matricola_studente`),
    foreign key (`nome_corso`) references corsi(`nome`)
        on delete no action on update cascade,
    foreign key (`matricola_studente`) references studenti(`matricola`)
        on delete no action on update cascade
);
```

```
-- Creazione tabella "Appelli":
create table appelli (
    `data` date not null,
    `nome_corso` varchar(256) not null,
    primary key (`data`, `nome_corso`),
    foreign key (`nome_corso`)
        references corsi(`nome`)
        on delete no action
        on update cascade
);
```

```
-- Creazione tabella "Iscrizioni" (che rappresenta l'omonima relazione):
create table iscrizioni (
    `nome_corso` varchar(256) not null,
    `data_appello` date not null,
    `matricola_studente` int not null,
    `voto` varchar(16),
    `stato_pubblicazione` varchar(16) not null,
    primary key (`nome_corso`, `data_appello`, `matricola_studente`),
    foreign key (`nome_corso`, `data_appello`) references appelli(`nome_corso`, `data`)
        on delete no action
        on update cascade,
    foreign key (`matricola_studente`) references studenti(`matricola`)
        on delete no action
        on update cascade
);
```

```
-- Creazione tabella "Verbali":
create table verbali(
    `codice` char(36) not null,
    `data_creazione` date not null,
    `ora_creazione` time not null,
    `nome_corso` varchar(256) not null,
    `data_appello` date not null,
    primary key (`codice`),
    foreign key (`nome_corso`, `data_appello`)
    references appelli(`nome_corso`, `data`)
        on delete no action
        on update cascade
);
```

```
-- Creazione tabella "Verbalizzazioni":
create table verbalizzazioni(
    `codice_verbale` char(36) not null,
    `matricola_studente` int not null,
    primary key (`codice_verbale`, `matricola_studente`),
    foreign key (`codice_verbale`)
        references verbali(`codice`)
        on delete no action
        on update cascade,
    foreign key (`matricola_studente`)
        references studenti(`matricola`)
        on delete no action
        on update cascade
);
```

Per una maggiore integrità della base di dati, sono stati aggiunti durante l'implementazione, alcuni constraints volti ad evitare la creazione di iscrizioni ad appelli incoerenti con le effettive iscrizioni ai corsi.

2.2 Analisi funzionale – versione pure HTML

ANALISI DEL TESTO DELLA SPECIFICA

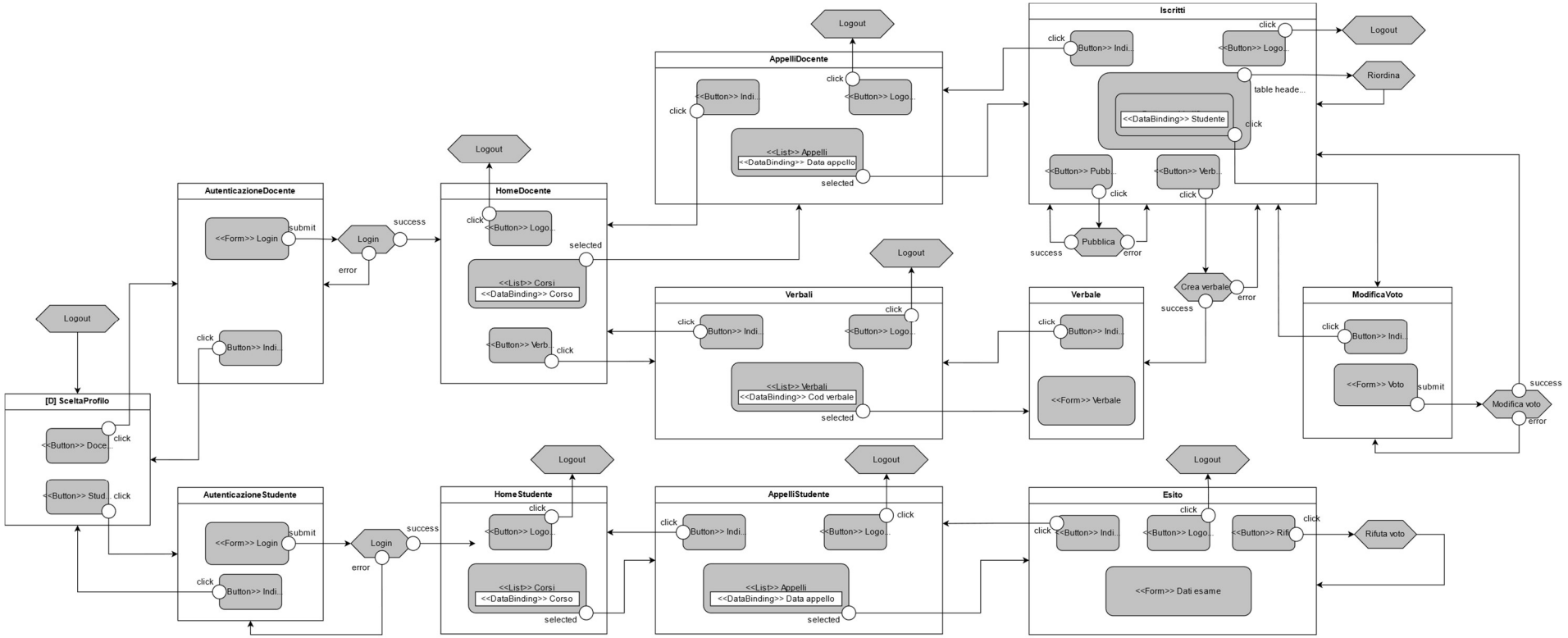
Legenda: ● Pagine ● Eventi ● View components ● Azioni

Un'applicazione permette di verbalizzare gli esiti degli esami di un appello. Il docente accede tramite login e seleziona nella HOME page un corso da una lista dei propri corsi, ordinata per nome del corso in modo alfabetico decrescente, e poi sceglie una data d'appello del corso scelto da un elenco ordinato per data decrescente. Ogni corso ha un solo docente. La selezione dell'appello porta a una pagina ISCRITTI, che mostra una tabella con tutti gli iscritti all'appello. La tabella riporta i seguenti dati: matricola, cognome e nome, email, corso di laurea, voto e stato di valutazione. Il voto può non essere ancora definito. Lo stato di valutazione dello studente rispetto all'appello può assumere i valori: non inserito, inserito, pubblicato, rifiutato e verbalizzato. Selezionando un'etichetta nell'intestazione della tabella, l'utente ordina le righe in base al valore di tale etichetta (ad esempio, selezionando "cognome" la tabella è riordinata in base al cognome). Successive selezioni della stessa etichetta invertono l'ordinamento: si parte con l'ordinamento crescente. Il valore del voto viene considerato ordinato nel modo seguente: <vuoto>, assente, rimandato, riprovato, 18, 19, …, 30, 30 e lode. Nella tabella della pagina ISCRITTI ad ogni riga corrisponde un bottone "MODIFICA". Premendo il bottone compare una pagina con una form che mostra tutti i dati dello studente selezionato e un campo di input in cui è possibile scegliere il voto. L'invio della form provoca la modifica o l'inserimento del voto. Inizialmente le righe sono nello stato di valutazione "non inserito". L'inserimento e le successive eventuali modifiche portano la riga nello stato di valutazione "inserito". Alla tabella della pagina ISCRITTI è associato un bottone PUBBLICA che comporta la pubblicazione delle righe con lo stato di valutazione INSERITO. La pubblicazione rende il voto non più modificabile dal docente e visibile allo studente e cambia lo stato di valutazione della riga dello studente a "pubblicato". Lo studente accede tramite login e seleziona nella HOME page un corso tra quelli a cui è iscritto mediante una lista ordinata in modo alfabetico decrescente e poi una data d'appello del corso scelto selezionata da un elenco ordinato per data decrescente. Uno studente può essere iscritto a più appelli dello stesso corso. [Aggiunta da chiarimento chiesto al docente: la lista mostra tutti gli appelli disponibili per quel corso, se lo studente è iscritto all'appello è possibile visualizzare i dati come specificato in seguito, se non è iscritto, la selezione della data d'appello porta alla pagina ESITO che mostra il messaggio "Studente non iscritto all'appello"] La selezione della data d'appello porta a una pagina ESITO che mostra il messaggio "Voto non ancora definito" se il docente non ha ancora pubblicato il risultato per quello studente in quell'appello. Altrimenti, la pagina mostra i dati dello studente, del corso, dell'appello e il voto assegnato. Se il voto è tra 18 e 30 e lode compare un bottone RIFIUTA. Premendo tale bottone la pagina mostra gli stessi dati con la dizione aggiunta "Il voto è stato rifiutato" e senza il bottone RIFIUTA. Il rifiuto del voto cambia lo stato di valutazione a "rifiutato" della riga dello studente per quell'appello nella pagina ISCRITTI del docente. Nella pagina ISCRITTI del docente la tabella degli iscritti è associata anche a un bottone VERBALIZZA. La pressione del bottone provoca il cambio di stato a "verbalizzato" per le righe nello stato "pubblicato" o "rifiutato" e comporta anche la creazione di un verbale e la disabilitazione della possibilità di rifiutare il voto. Il rifiuto implica la verbalizzazione di "rimandato" come voto. Un verbale ha un codice generato dal sistema, una data e ora di creazione ed è associato all'appello del corso a cui si riferisce e agli studenti (con nome, cognome, matricola e voto) che passano allo stato "verbalizzato". A seguito della pressione del bottone VERBALIZZA compare una pagina VERBALE che mostra i dati completi del verbale creato. Il docente dispone anche di una pagina VERBALI, in cui sono elencati tutti i verbali da lui creati, ordinati in modo alfabetico crescente per nome del corso e poi per data crescente di appello di ogni corso.

COMPLETAMENTO DELLA SPECIFICA

- La landing page è una pagina SceltaProfilo che porta a due pagine Autenticazione separate
- Sia per il docente che per lo studente, la lista di appelli di un corso, una volta selezionato, compare in una pagina APPELLI a parte
- La pagina verbali del docente è accessibile in ogni punto dell'applicazione tramite un apposito bottone nella barra superiore

DIAGRAMMA IFML



COMPONENTI DELL'APPLICAZIONE

Viste / Pagine:

- SceltaProfilo
- Docente:
  - LoginDocente
  - HomeDocente
  - AppelliDocente
  - Iscritti
  - ModificaVoto
  - Verbali
  - Verbale
- Studente:
  - LoginStudente
  - HomeStudente
  - AppelliStudente
  - Esito
- Logout
- Error

Controllers / Servlets:

- Index (scelta profilo)
- Docente:
  - LoginDocente
  - HomeDocente
  - AppelliDocente
  - Iscritti
  - ModificaVoto
  - Verbali
  - VisualizzaVerbale
- Studente:
  - LoginStudente
  - HomeStudente
  - AppelliStudente
  - Esito
- Logout
- Error

Beans:

- Docente
- Studente
- Corso
- Appello
- Iscrizione
- Verbale
- Verbalizzazione

Data Access Objects

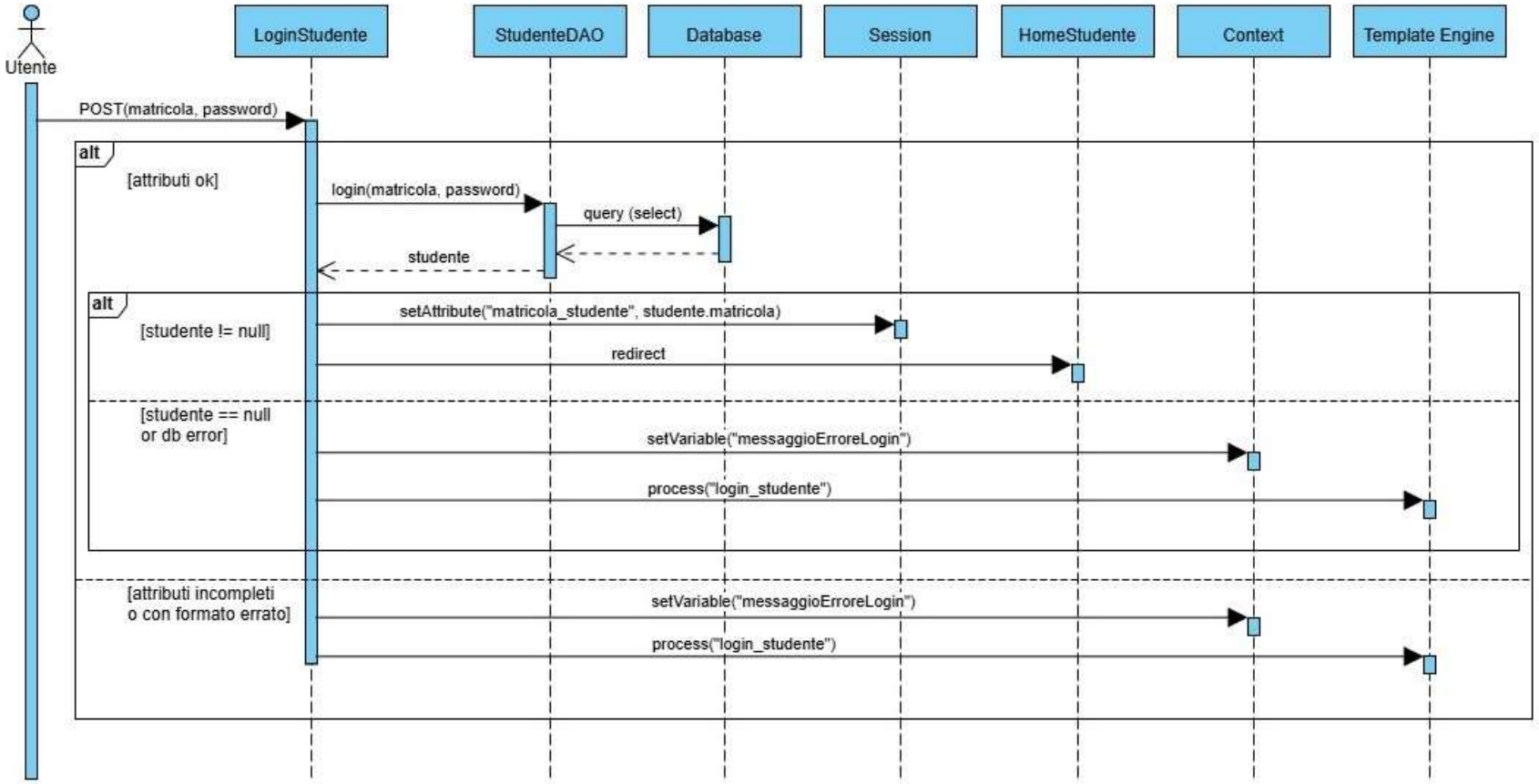
- DocenteDAO
  - login (codiceDocente, password)
- StudenteDAO
  - login (matricola, password)
  - getStudenteByMatricola (matricola)
- CorsoDAO
  - getCorsiByStudente (matricola)
  - getCorsiByDocente (codiceDocente)
- IscrizioneDAO
  - getDatiIscrizione (matricola, data, corso)
  - getDatiIscrizioni (corso, data, matricole)
  - getOrderedIscritti (data, corso, campoOrdine, desc/asc)
  - rifiutaEsito (matricola, data, corso)
  - modificaVoto (matricola, corso, voto)
  - pubblicaVoti (corso, data)
  - verbalizzavoti (corso, data)
- AppelloDAO
  - getAppelliByCorso (corso)
- VerbaleDAO
  - getVerbaleByCodice (codiceVerbale)
  - getVerbaleByDocente (codiceDocente)
  - getMatricoleByCodice (codiceVerbale)

SEQUENCE DIAGRAMS

Si riportano di seguito i sequence diagram che rappresentano la “risposta” delle servlet ai metodi GET e POST. Si è deciso di riportare soltanto quelli relativi a procedure più complesse, mentre si sono omessi quelli relativi a operazioni più semplici (come GET delle pagine di login o della pagina iniziale)

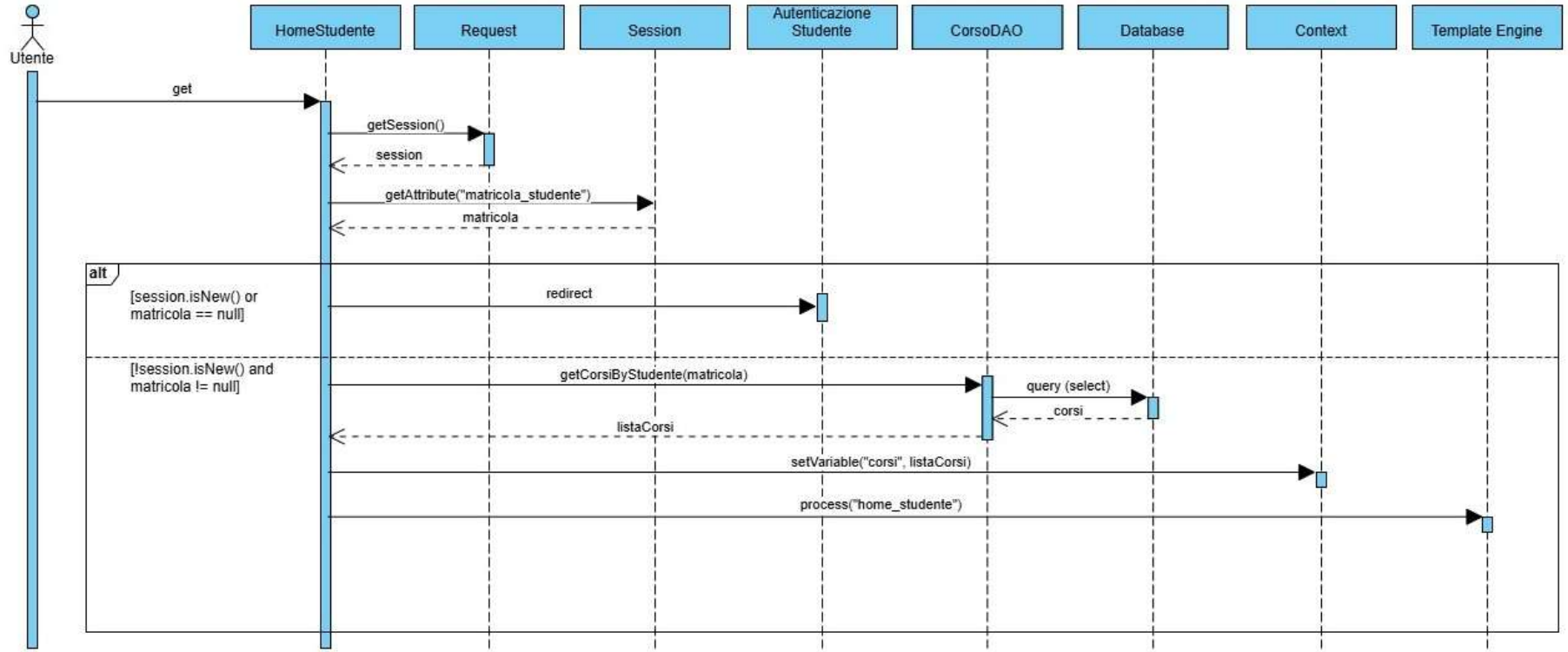
SEQUENCE DIAGRAM – AUTENTICAZIONE

Diagramma della procedura di autenticazione in risposta ad una chiamata POST alla servlet LoginStudente. Si riporta il sequence diagram del login studente, il funzionamento è analogo per il login del docente (a patto di considerare la servlet LoginDocente, il DAO DocenteDAO, la servlet di redirect HomeDocente e il template login_docente)



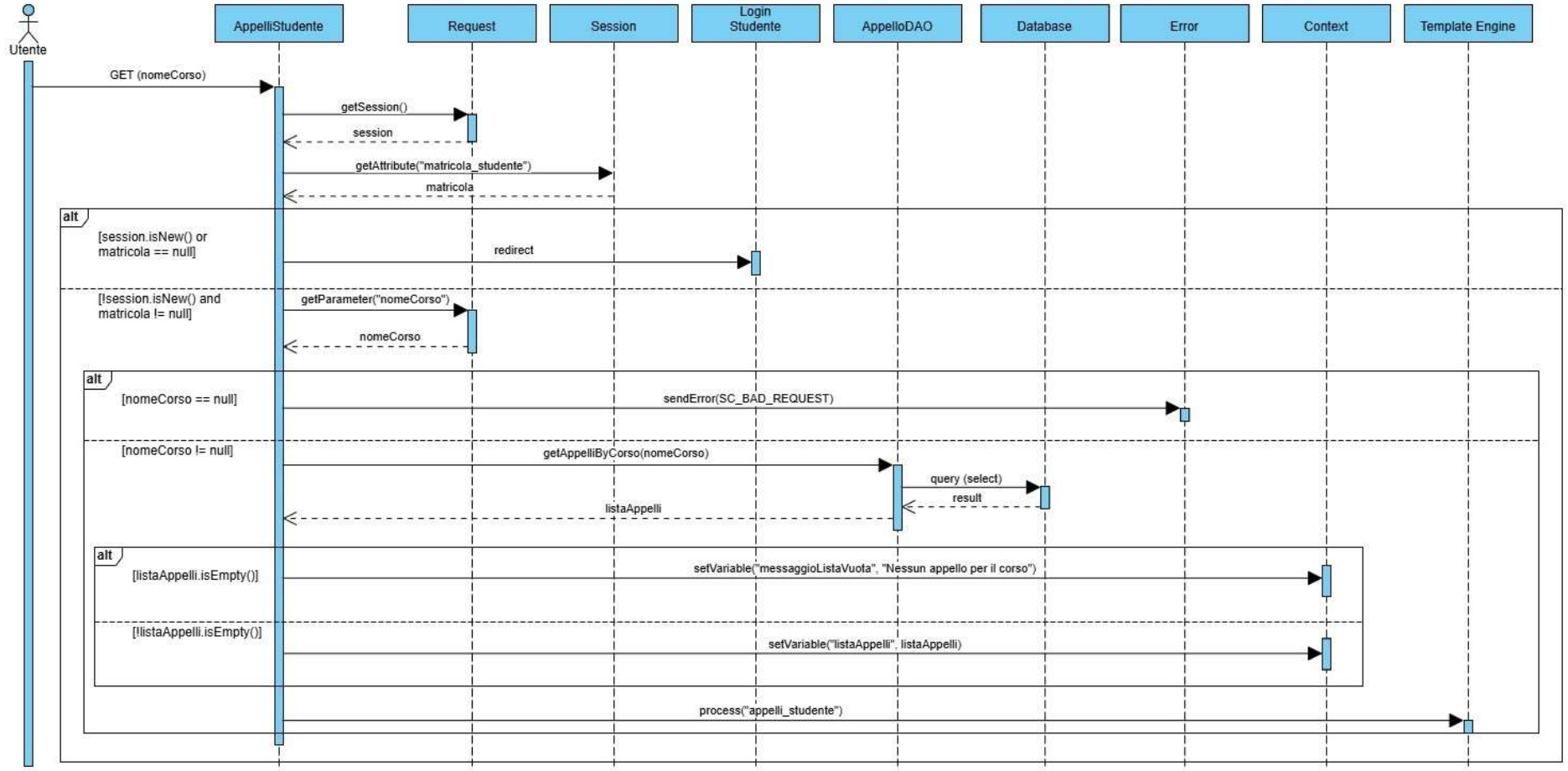
SEQUENCE DIAGRAM – HOME PAGE

Diagramma della risposta ad una richiesta GET sulla home page dello studente (analogo per il docente, a patto di considerare i controller HomeDocente e LoginDocente, il metodo “getCorsiByDocente” del DAO dei corsi e il template “home_docente”).



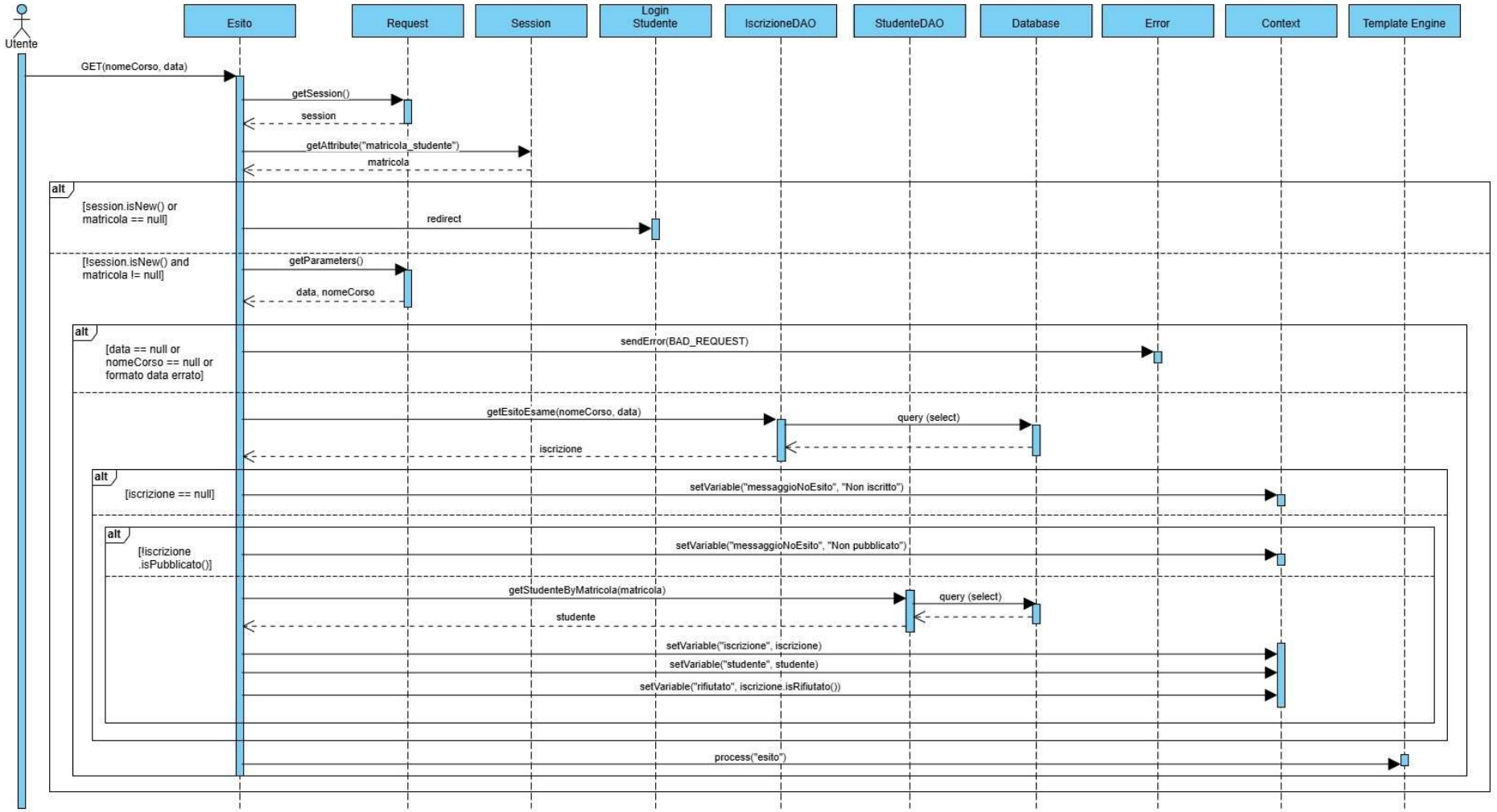
SEQUENCE DIAGRAM – APPELLI

Diagramma della reazione al click su un corso a cui è iscritto lo studente, mostra gli appelli disponibili per quel corso. È analogo per il docente, cioè come risposta al click su un corso di cui il docente è titolare, a patto di considerare i controller AppelliDocente e LoginDocente e il template “appelli_docente”



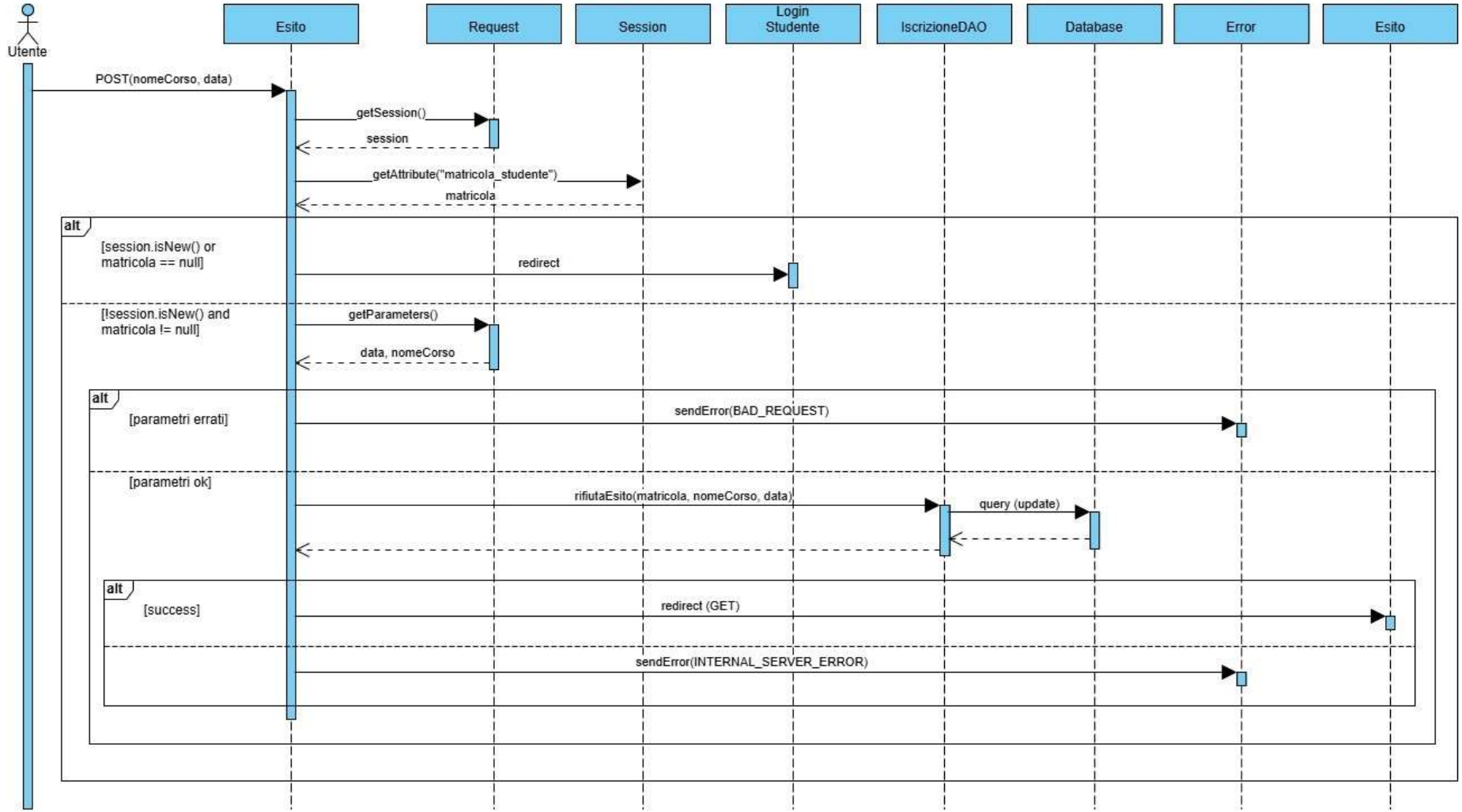
SEQUENCE DIAGRAM – ESITO

Di seguito il sequence diagram per la gestione di una GET alla pagina Esito (nella sezione per studente) legata ad un appello. Tale procedura è molto simile (seppur con qualche piccola differenza nelle variabili salvate nel context, a solo scopo di presentazione) alla procedura di risposta ad una richiesta GET alla pagina ModificaVoto della sezione docente, si omette perciò il diagramma di GET ModificaVoto.

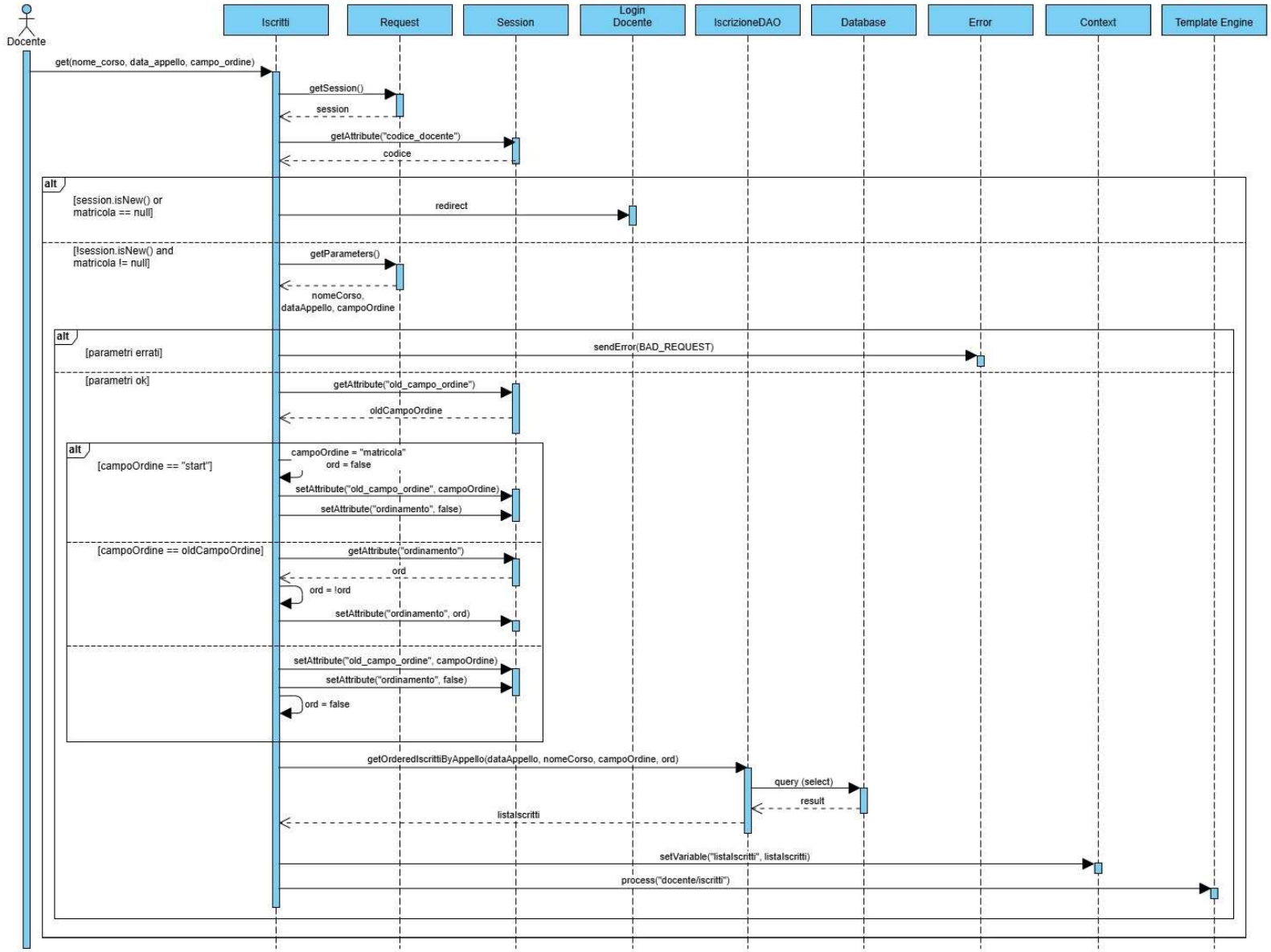


SEQUENCE DIAGRAM – RIFIUTA ESITO ESAME

Il bottone “Rifiuta” della pagina Esito genera una chiamata POST alla pagina stessa, specificando il corso e il relativo appello, di seguito il diagramma della gestione di tale richiesta alla pagina Esito della sezione studente.

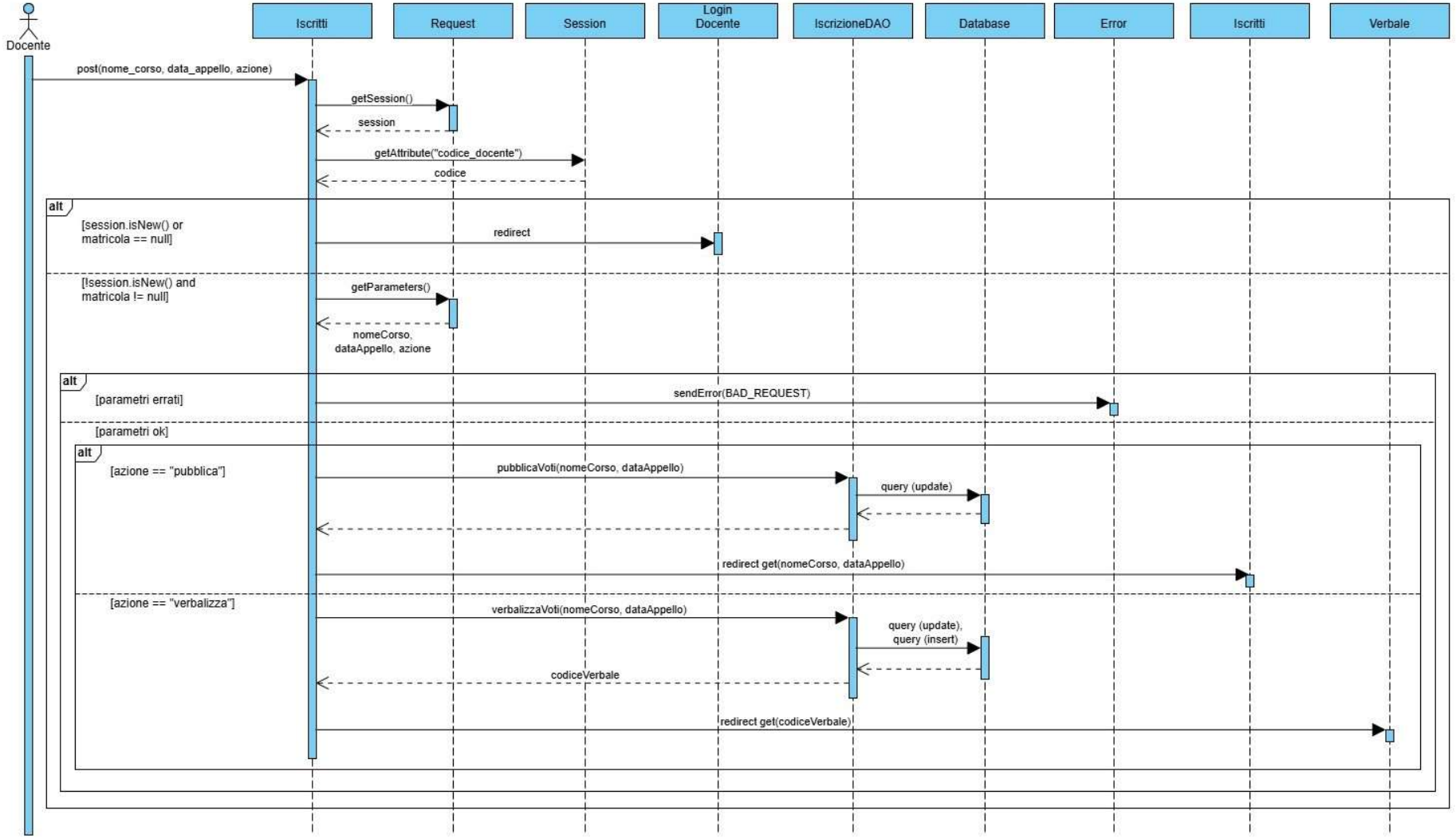


SEQUENCE DIAGRAM – ISCRITTI



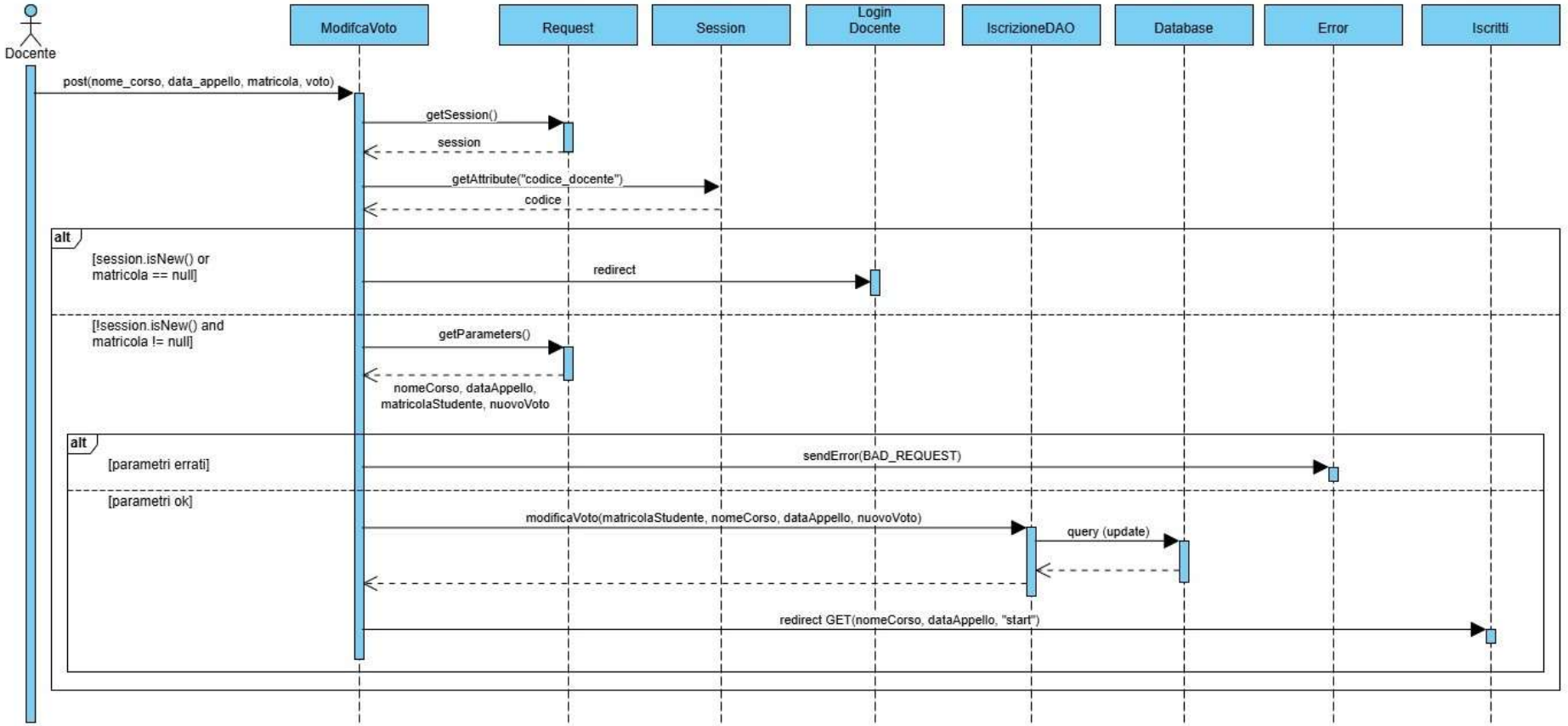
SEQUENCE DIAGRAM – PUBBLICA / VERBALIZZA VOTI

Diagramma della risposta alla pressione di uno dei bottoni “Pubblica” e “Verbalizza” della pagina Iscritti della sezione docente. Entrambi generano una chiamata POST alla pagina Iscritti che viene gestita come illustrato nel diagramma.



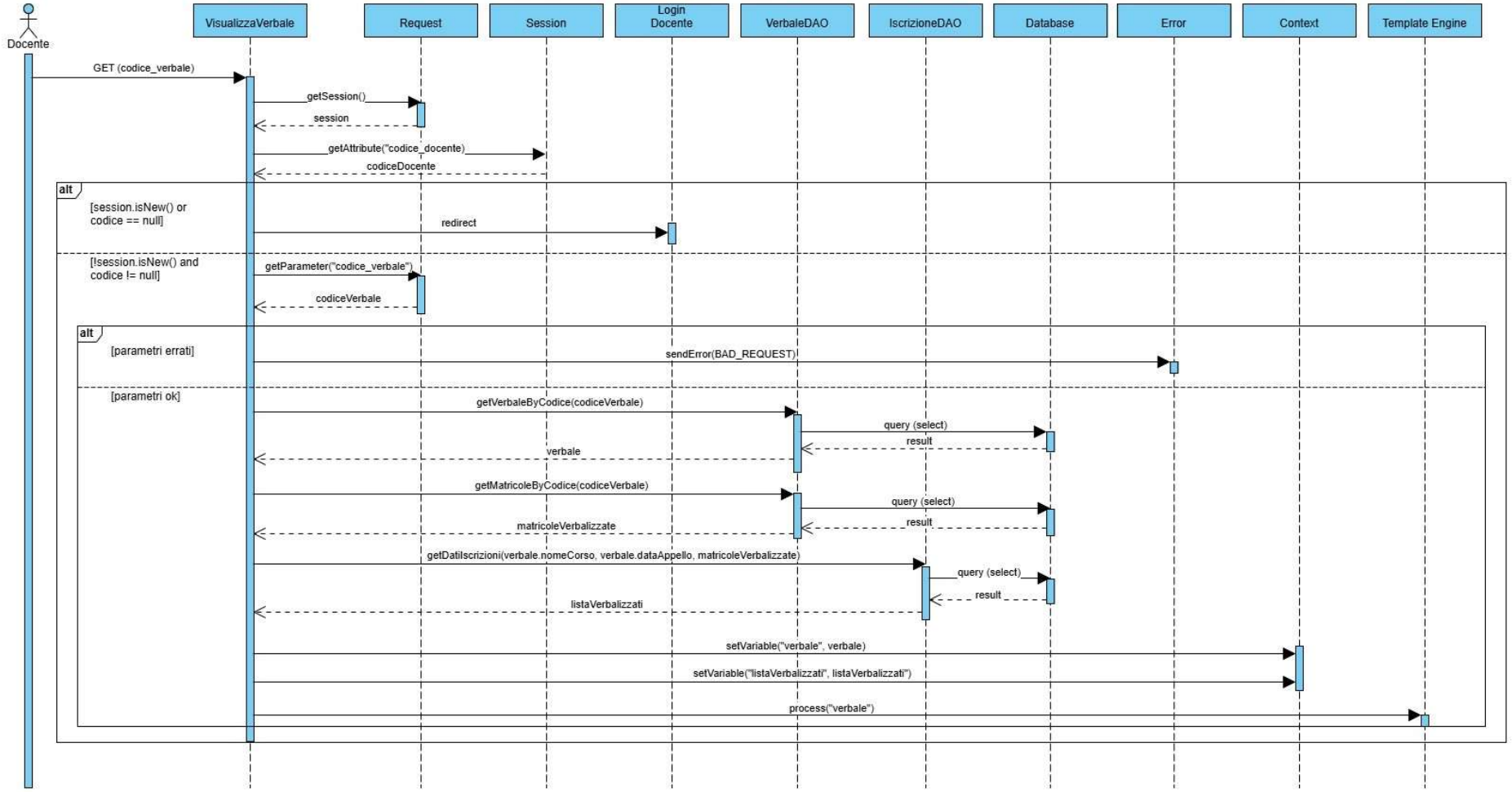
SEQUENCE DIAGRAM – MODIFICA VOTO

Diagramma della gestione di una richiesta POST alla pagina ModificaVoto del docente. Il risultato è l'inserimento del nuovo voto nel database e il cambio dello stato di valutazione.



SEQUENCE DIAGRAM – VISUALIZZA VERBALE

Diagramma della gestione di una richiesta GET alla pagina VisualizzaVerbale del docente.



2.3 Analisi funzionale – versione con JavaScript

- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina per il docente e un'unica pagina per lo studente.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- La funzione di riordino della tabella degli iscritti è realizzata a lato client.
- La funzione di rifiuto del voto da parte dello studente è realizzata mediante trascinamento del testo corrispondente all'esito (dati dello studente, del corso, dell'appello e voto assegnato) sopra un'icona che rappresenta il cestino. Al rilascio compare un messaggio popup che chiede di confermare l'operazione di rifiuto, con due bottoni (CANCELLA, CONFERMA). Il primo ripristina la situazione precedente al trascinamento, il secondo effettua l'operazione di rifiuto.
- Alla tabella degli iscritti è associato un bottone INSERIMENTO MULTIPLO che provoca la comparsa di una pagina modale con tutte e sole le righe nello stato "non inserito" associate a un campo di input. Il docente può inserire un voto per un insieme delle righe e premere un bottone INVIA che comporta l'invio al server dei voti, il cambio di stato delle righe coinvolte, la chiusura della finestra modale e l'aggiornamento della tabella degli iscritti.

SOMMARIO – VISTE E VIEW COMPONENTS

Pagina "**Scelta profilo**": bottone Docente e bottone Studente

Pagina "**Login Docente/Studente**": form di login

Home Docente

- Barra superiore:
  - Bottoni "Verbali", "Home" e "Logout"
- Vista Corsi:
  - Tabella corsi
- Vista Appelli:
  - Tabella appelli
- Vista Iscritti:
  - Tabella iscritti (riordinabile)
  - Bottoni "Pubblica" e "Verbalizza
- Vista ModificaVoto:
  - Form dati studente
  - Scelta voto e bottone "Salva"
- Vista Verbali:
  - Tabella verbali
- Vista VisualizzaVerbale:
  - Informazioni verbale

Home Studente

- Barra superiore
  - Bottoni "Home" e "Logout"
- Vista Corsi:
  - Tabella corsi
- Vista Appelli:
  - Tabella appelli
- Vista Esito:
  - Form dati studente ed esame
  - Icona cestino per rifiutare

Le viste “Corsi” e “Appelli” sono una vista unica: si parte con la sola lista dei corsi, alla selezione di un corso, compare anche la tabella di appelli alla destra, lasciando la possibilità sia di selezionare un appello, sia di selezionare un altro corso. Ciò vale sia per lo studente che per il docente.

SOMMARIO – EVENTI ED AZIONI

Pagina “Scelta profilo”: click profilo → mostra pagina login del profilo

Pagina “Login”: submit form → check credenziali e mostra home page

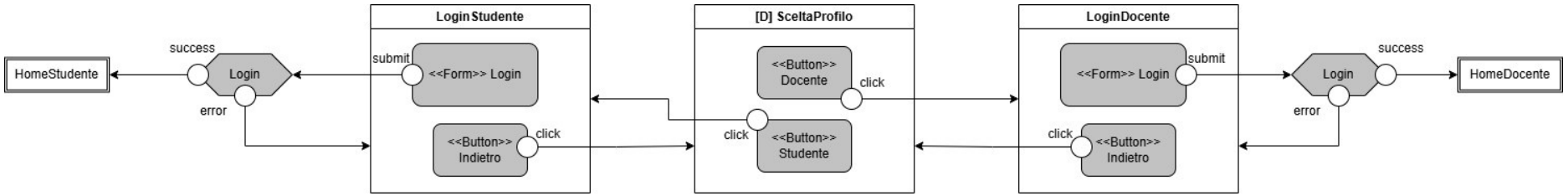
Home Docente

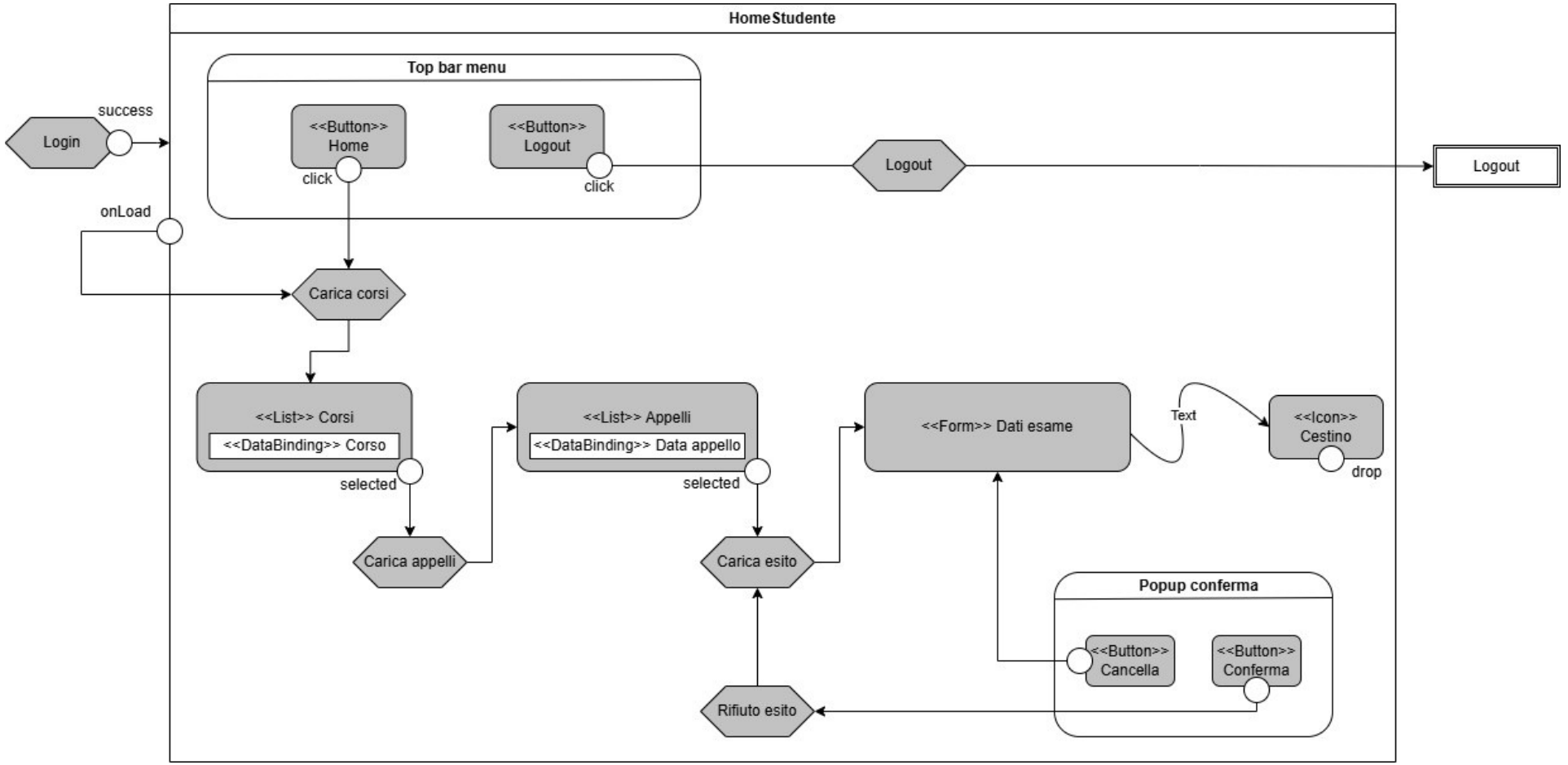
- Click su un corso → carica appelli
- Click su una data appello → carica iscritti all’appello
- Click intestazione iscritti → riordinamento tabella
- Click bottone modifica → carica dati studente e mostra form
- Submit form con voto → modifica voto
- Click bottone “Pubblica” → pubblicazione e aggiornamento tabella
- Click bottone “Verbalizza” → creazione verbale e mostra verbale
- Click su un verbale → carica verbale e mostra voto

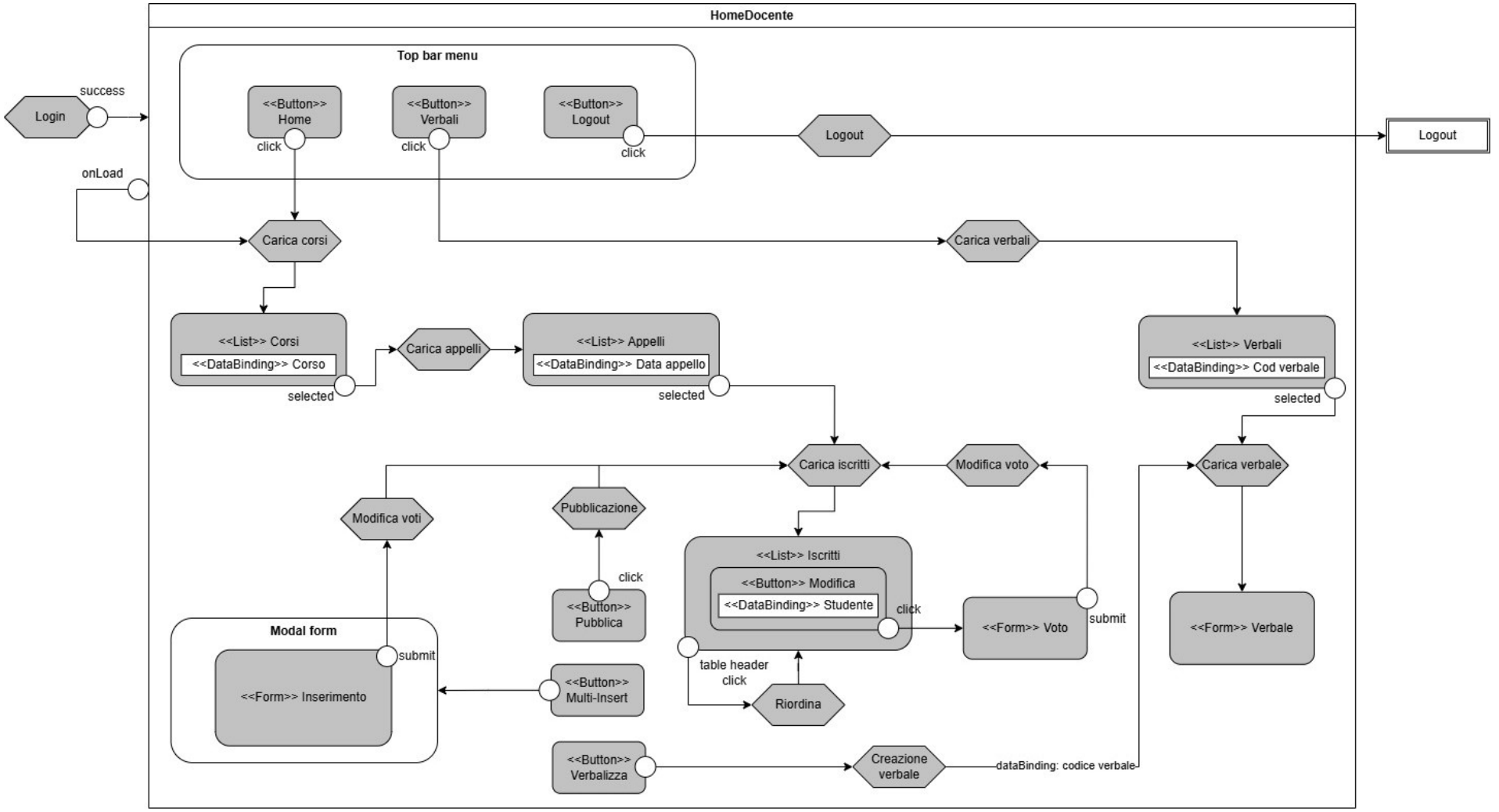
Home Studente

- Click su un corso → carica appelli
- Click su una data appello → carica esito esame
- Drag and drop dell’esito sul cestino → rifiuto esito esame

DIAGRAMMA IFML




HomeStudente
Top bar menu
<<Button>>
Home
click
<<Button>>
Logout
click
Logout
Logout
success
Login
onLoad
Carica corsi
<<List>> Corsi
<<DataBinding>> Corso
selected
Carica appelli
<<List>> Appelli
<<DataBinding>> Data appello
selected
Carica esito
<<Form>> Dati esame
Text
<<Icon>>
Cestino
drop
Popup conferma
<<Button>>
Cancella
<<Button>>
Conferma
Rifiuto esito


HomeDocente
Top bar menu
<<Button>> Home
click
<<Button>> Verbali
click
<<Button>> Logout
click
Logout
Logout
Login
success
onLoad
Carica corsi
Carica verbali
<<List>> Corsi
<<DataBinding>> Corso
selected
Carica appelli
<<List>> Appelli
<<DataBinding>> Data appello
selected
<<List>> Verbali
<<DataBinding>> Cod verbale
selected
Carica iscritti
Modifica voto
Carica verbale
Modifica voti
Pubblicazione
<<List>> Iscritti
<<Button>> Modifica
<<DataBinding>> Studente
click
<<Form>> Voto
submit
<<Form>> Verbale
click
<<Button>> Pubblica
Modal form
submit
<<Form>> Inserimento
<<Button>> Multi-Insert
table header click
Riordina
<<Button>> Verbalizza
Creazione verbale
dataBinding: codice verbale

COMPONENTI DELL'APPLICAZIONE – SERVER SIDE

Beans:

- Docente
- Studente
- Corso
- Appello
- Iscrizione
- Verbale
- Verbalizzazione

Controllori (servlets):

- VerificaLogin
- Corsi
- Appelli
- Iscritti
- EsitoEsame
- Verbali
- Logout

Data Access Objects

- DocenteDAO
  - login (codiceDocente, password)
- StudenteDAO
  - login (matricola, password)
  - getStudenteByMatricola (matricola)
- CorsoDAO
  - getCorsiByStudente (matricola)
  - getCorsiByDocente (codiceDocente)
- IscrizioneDAO
  - getDatiIscrizione (matricola, data, corso)
  - getDatiIscrizioni (corso, data, matricole)
  - getOrderedIscritti (data, corso, campoOrdine, desc/asc)
  - rifiutaEsito (matricola, data, corso)
  - modificaVoto (matricola, corso, voto)
  - modificaMultipliVoti(nomeCorso, dataAppello, mappaMatricolaVoto)
  - pubblicaVoti (corso, data)
  - verbalizzavoti (corso, data)
- AppelloDAO
  - getAppelliByCorso (corso)
- VerbaleDAO
  - getVerbaleByCodice (codiceVerbale)
  - getVerbaleByDocente (codiceDocente)
  - getMatricoleByCodice (codiceVerbale)

COMPONENTI DELL'APPLICAZIONE (SOLO HOME PAGES) – CLIENT SIDE

Docente:

- NomeDocente:
  - show(): mostra il nome del docente
- ButtonIndietro:
  - init(): imposta il ritorno al login come azione
  - evolve(): imposta il ritorno al login come azione
  - evolveEsito(): imposta il ritorno agli iscritti come azione
  - evolveVerbali(): imposta il ritorno ai verbali come azione
- AnchorLogout, AnchorHome, AnchorVerbali
- ListaCorsi
  - show(): ottiene la lista dal server e la mostra
  - updateData(lista): popola la tabella con la lista
  - reset(): svuota e nasconde il componente
- ListaAppelli
  - show(corso): ottiene la lista dal server e la mostra
  - updateData(lista): popola la tabella con la lista
  - reset(): svuota e nasconde il componente
- ListaIscritti
  - show(corso, appello): ottiene la lista dal server e la mostra
  - updatedata(lista): popola la tabella con la lista
  - reset(): svuota e nasconde il componente
  - hide(): nasconde il componente senza svuotarlo
  - refresh(): ricarica il componente con corso e data precedenti
  - pubblicaVoti(): invia al server la richiesta di pubblicare
  - verbalizzaVoti(): invia al server la richiesta di verbalizzare
  - (funzione anonima): riordina la tabella in base alla header
- ModificaEsito
  - show(): ottiene i dati dal server e li mostra
  - updateForm(iscrizione): popola il form
  - reset(): nasconde il componente
  - modificaVoto(voto): invia al server la richiesta di modificare

Studente:

- NomeStudente
  - show(): mostra il nome dello studente
- ButtonIndietro
  - init(): imposta il ritorno al login come azione
  - evolve(): imposta il ritorno alla home come azione
- AnchorLogout
- AnchorHome
- ListaCorsi
  - show(): ottiene la lista dal server e la mostra
  - updateData(lista): popola la tabella con la lista di corsi passata
  - reset(): svuota e nasconde il componente
- ListaAppelli
  - show(nomeCorso): ottiene la lista dal server e la mostra
  - updateData(lista): popola la tabella con la lista di appelli passata
  - reset(): svuota e nasconde il componente
- EsitoEsame:
  - show(nomeCorso, dataAppello): ottiene l'esito dal server e lo mostra
  - updateForm(iscrizione): popola il form vero e proprio
  - reset(): nasconde il componente
  - rifiutaEsito(): invia al server la richiesta di rifiuto, poi aggiorna la pagina
- MessaggioPopup:
  - show(messaggio, onConferma, onCancella): mostra il popup di conferma ed esegue una delle due funzioni passate
  - close(): nasconde il componente

- FinestraModale
    - show(): mostra la finestra con i suoi componenti
    - reset(): nasconde la finestra e svuota la tabella
    - modificaMultiVoto(multiVoto): invia al server la richiesta di modificare più voti
    - (funzione anonima): raccoglie i voti inseriti e chiama la funzione modificaMultiVoto
- VisualizzaVerbale
    - show(codice): ottiene dal server e mostra i dati del verbale
    - updateData(verbale, listaEsiti): popola la tabella e il form
    - reset(): svuota la tabella e nasconde il componente
- ListaVerbali
    - show(): ottiene la lista e la mostra
    - updateData(verbali): popola la tabella
    - reset(): svuota la tabella e nasconde il componente

GESTIONE DEGLI EVENTI – ENTRAMBI I PROFILI

Client side		Server side	
Evento	**Controllore**	**Evento**	**Controllore**
Index → href profilo → click	Gestore href (vai al login)	-	-
Login → form → submit (studente e docente)	Funzione makeCall	POST (username, password, profilo)	VerificaLogin
Home → load	Funzione makeCall	GET (profilo)	Corsi
Home → click anchor “indietro”	Gestore href (vai a scelta profilo)	-	-
Home → click anchor “logout”	Funzione makeCall	GET (profilo)	Logout
Home → click anchor “home”	pageOrchestrator (torna alla home)	-	-
Home → click anchor corso	Funzione makeCall	GET (profilo, corso)	Appelli

GESTIONE DEGLI EVENTI – SOLO STUDENTE

Client side		Server side	
Evento	**Controllore**	**Evento**	**Controllore**
Home → click anchor appello	Funzione makeCall	GET (profilo, corso, dataAppello)	EsitoEsame
Home (esito) → click anchor “indietro”	pageOrchestrator (refreshPage)	GET (profilo)	Corsi
Home (esito) → drag and drop esito	Funzione MessaggioPopup (show)	-	-
Popup → conferma	Funzione makeCall	POST (profile, corso, dataAppello)	EsitoEsame
Popup → cancella	Funzione MessaggioPopup (close)	-	-

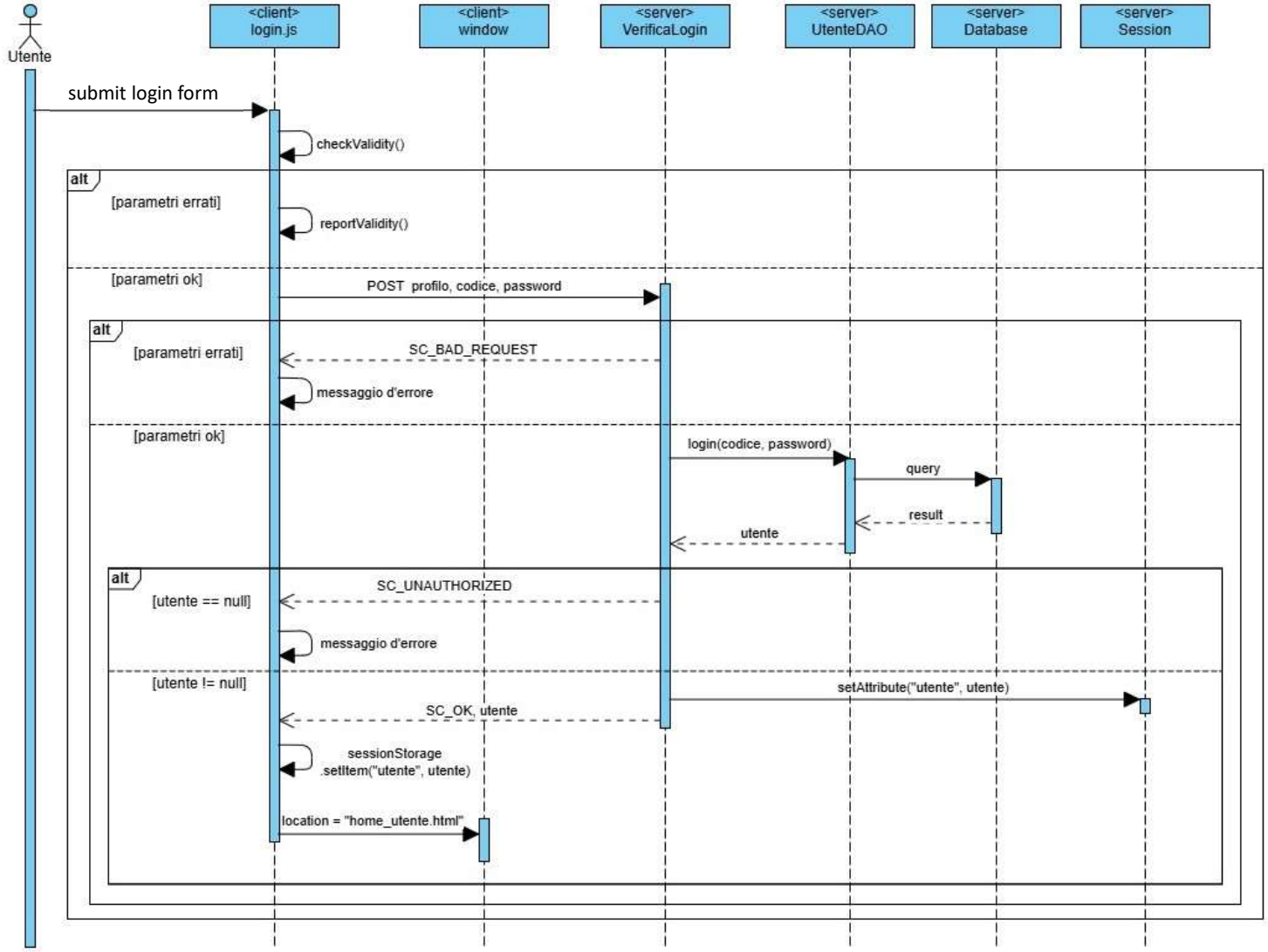
GESTIONE DEGLI EVENTI – SOLO DOCENTE

Client side		Server side	
Evento	**Controllore**	**Evento**	**Controllore**
Home → click anchor appello	Funzione makeCall	GET (corso, dataAppello)	Iscritti
Home (iscritti) → click anchor “indietro”	pageOrchestrator (refreshPage)	GET (profilo)	Corsi
Home (iscritti) → click header	ListaIscritti (riordina)	-	-
Home (iscritti) → click “Pubblica”	Funzione makeCall	POST (nomeCorso, dataAppello, azione = “pubblica”)	Iscritti
Home (iscritti) → click “Verbalizza”	Funzione makeCall	POST (nomeCorso, dataAppello, azione = “verbalizza”)	Iscritti
Home (iscritti) → click “Inserimento”	FinestraModale (show)	-	-
FinestraModale → click “Annulla”	FinestraModale (reset)	-	-
FinestraModale → click “Invia”	Funzione makeCall	POST (nomeCorso, dataAppello, azione = “modifica”, mappaMatricolaVoto)	EsitoEsame

Home (iscritti) → click anchor “edit”	Funzione makeCall	GET (nomeCorso, dataAppello, profilo, matricola)	EsitoEsame
Home (modifica voto) → click anchor “indietro”	pageOrchestrator (refresh iscritti)	GET (corso, dataAppello)	Iscritti
Home (modifica voto) → click “Salva”	Funzione makeCall	POST (nomeCorso, dataAppello, azione = “modifica”, matricola, voto)	EsitoEsame
Home (qualunque vista) → click “Verbali”	Funzione makeCall	GET (nessun parametro)	Verbali
Home (verbali) → click verbale	Funzione makeCall	GET (codiceVerbale)	Verbali

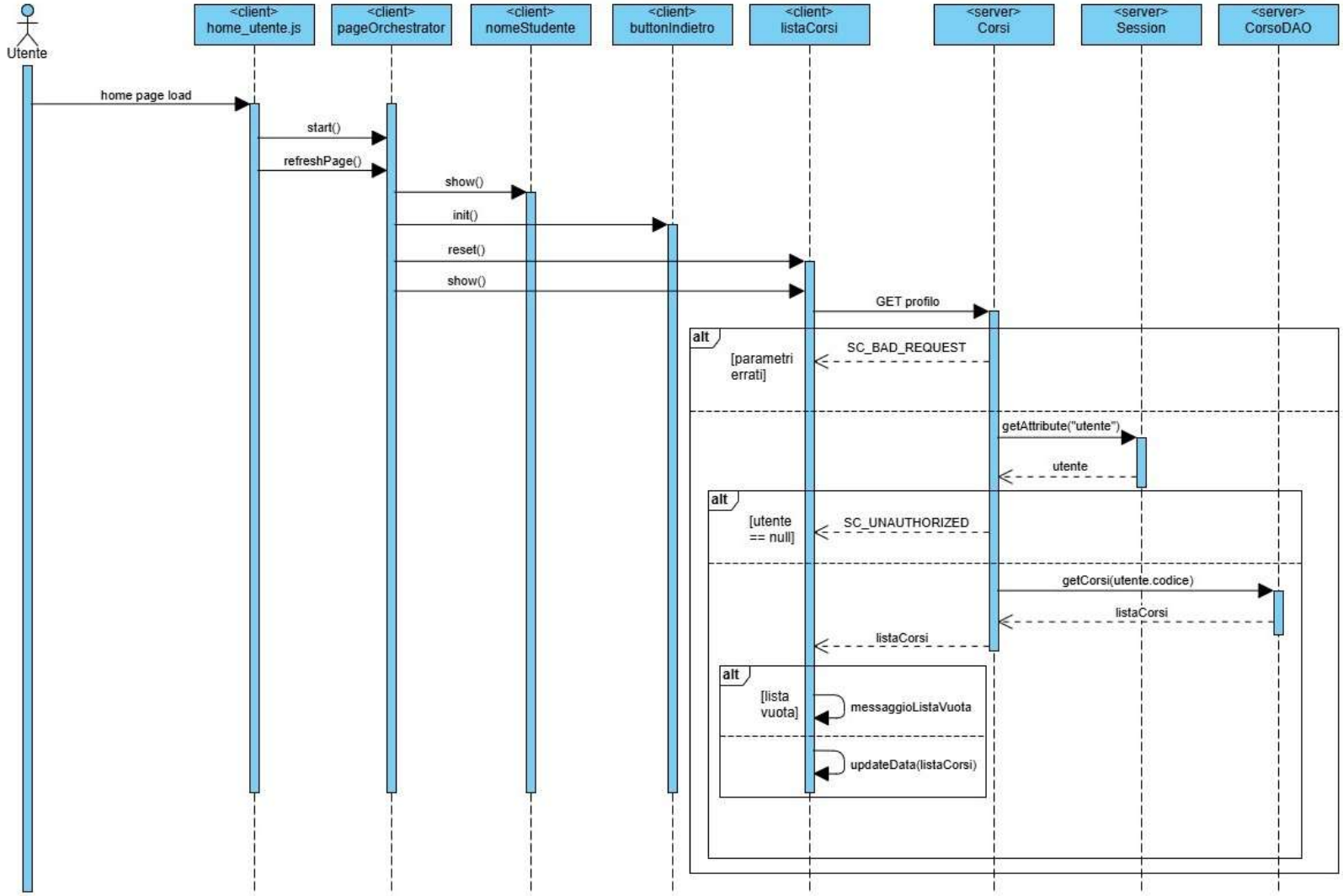
SEQUENCE DIAGRAM – LOGIN

Di seguito il sequence diagram per la gestione del login, sia del docente che dello studente, indicati come Utente.



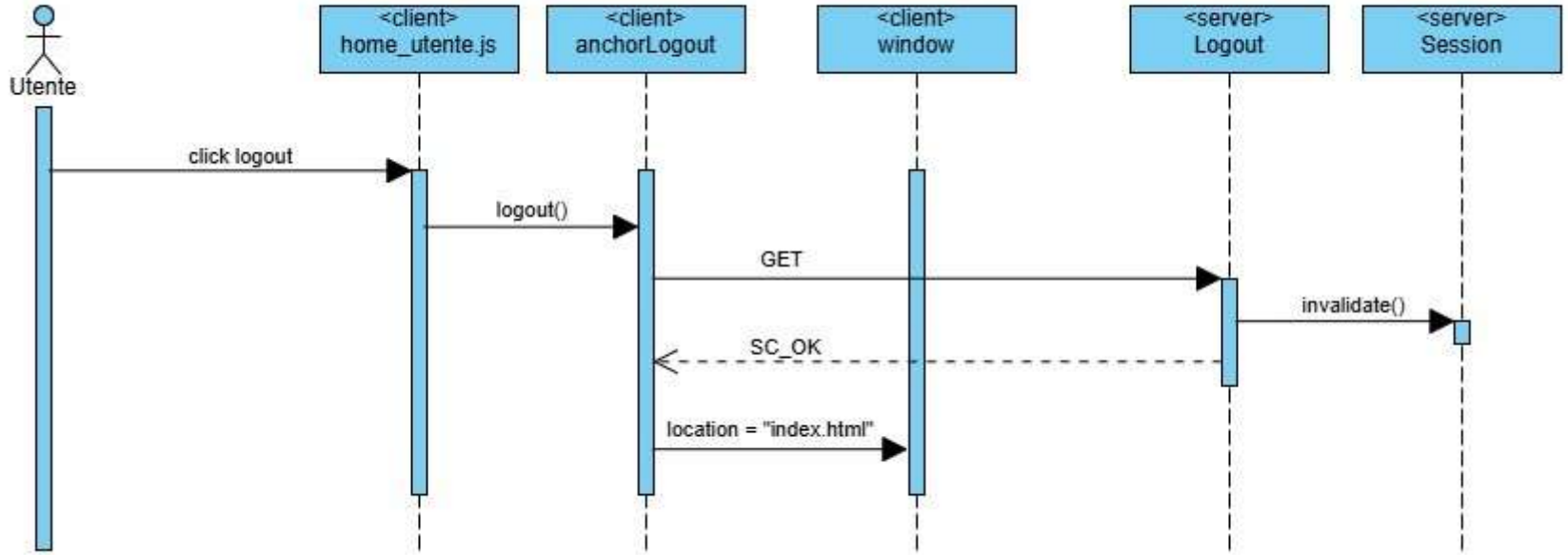
SEQUENCE DIAGRAM – CARICAMENTO HOME PAGE

Di seguito il sequence diagram del caricamento della home page, sia per lo studente che per il docente (indicati come Utente)



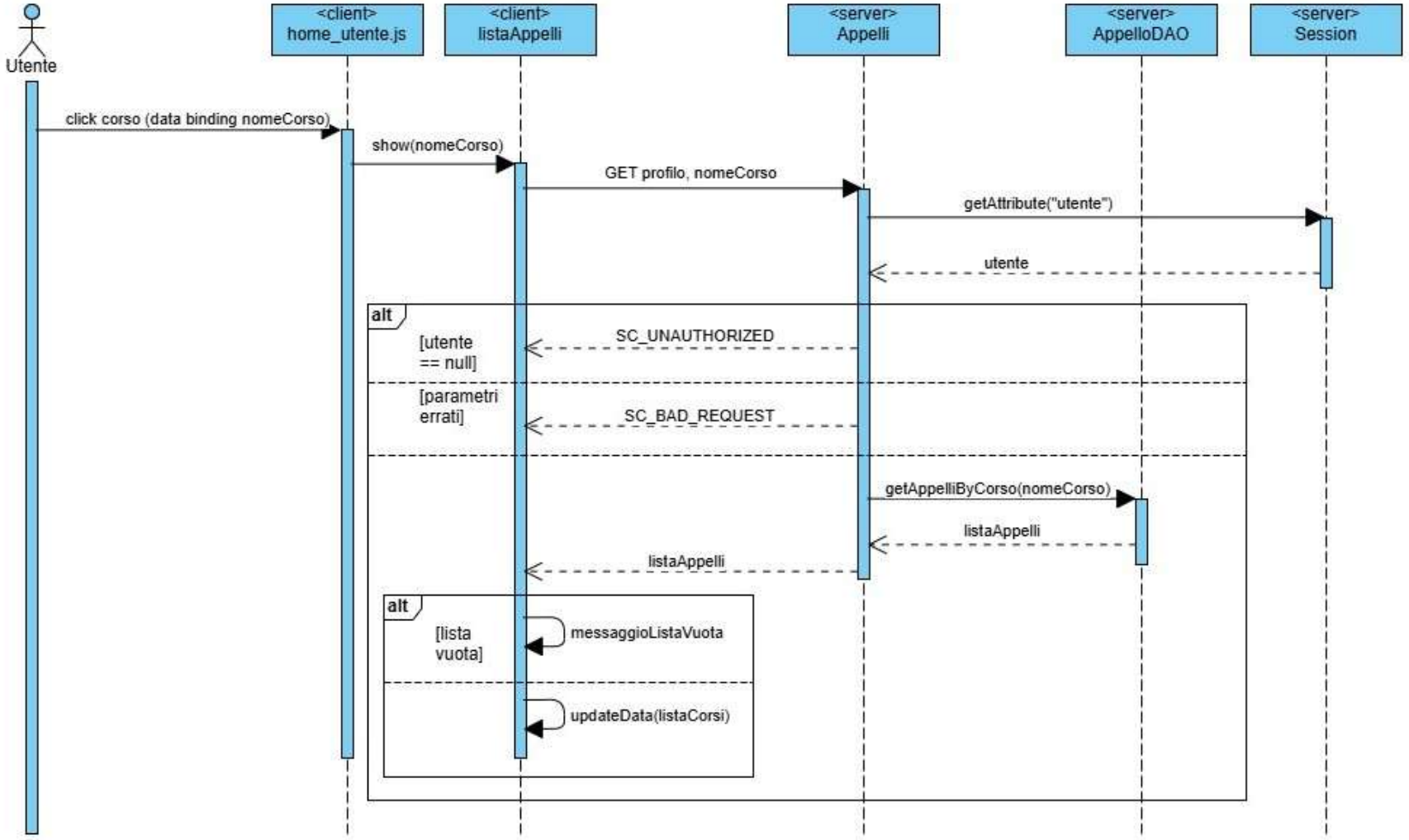
SEQUENCE DIAGRAM – LOGOUT

Di seguito il sequence diagram per il logout del docente e dello studente, indicati come Utente.



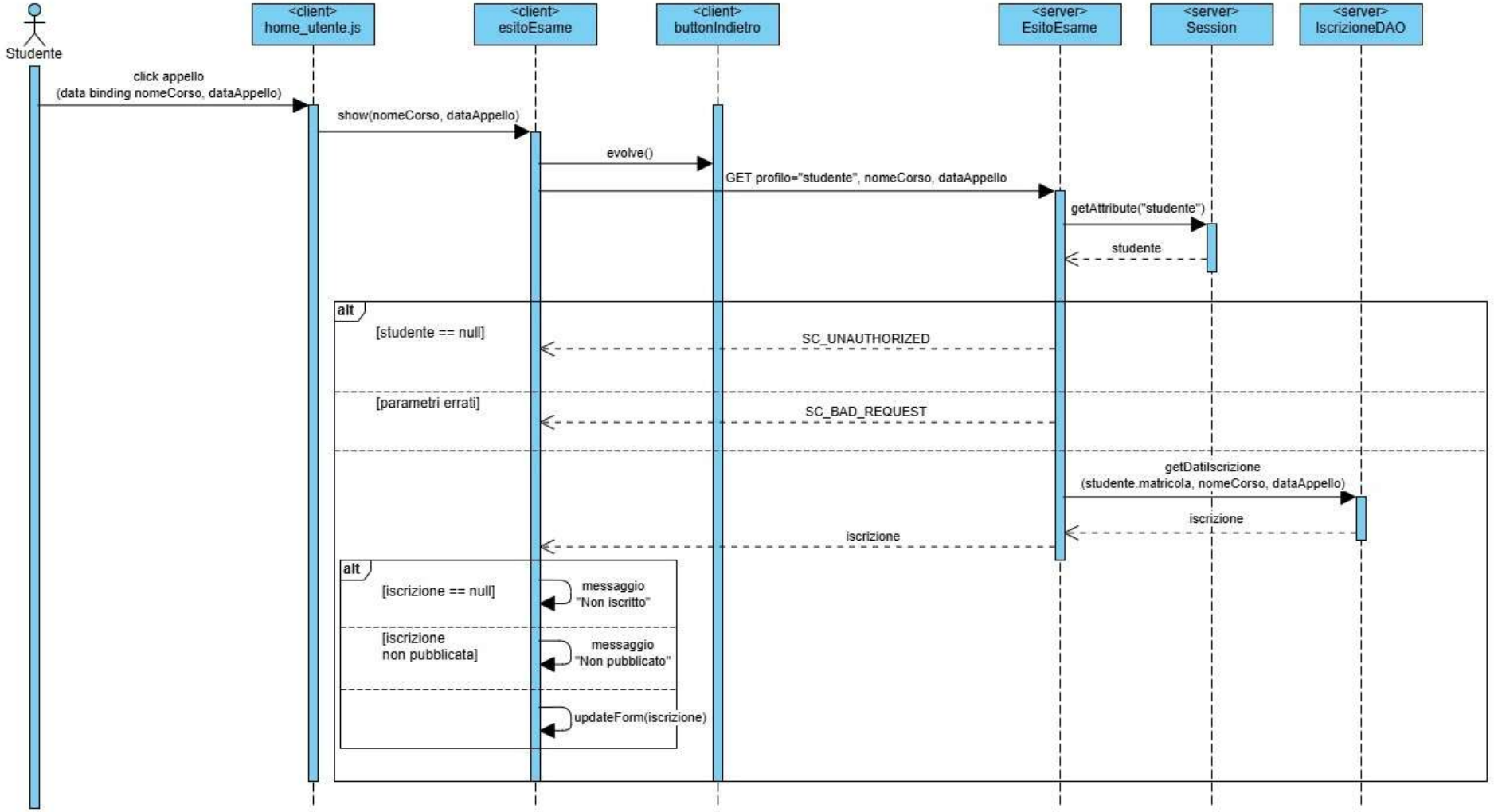
SEQUENCE DIAGRAM – CLICK ANCHOR CORSO

Di seguito il sequence diagram della risposta all'evento di click su un corso sia per il docente che per lo studente, indicati come Utente.



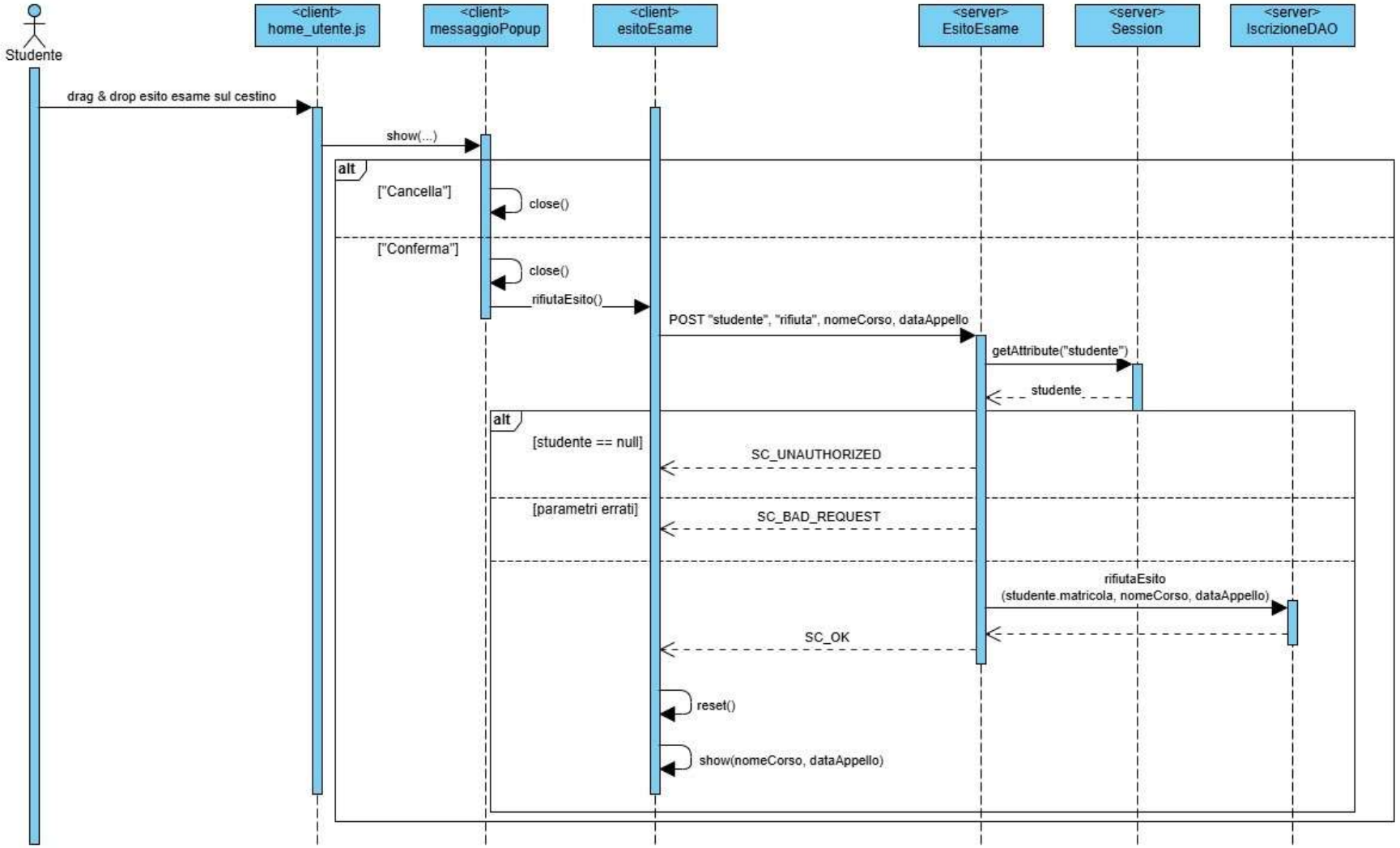
SEQUENCE DIAGRAM – VISUALIZZA ESITO ESAME (STUDENTE)

Di seguito il sequence diagram, per il solo studente, della risposta al click su una data d'appello di un corso.



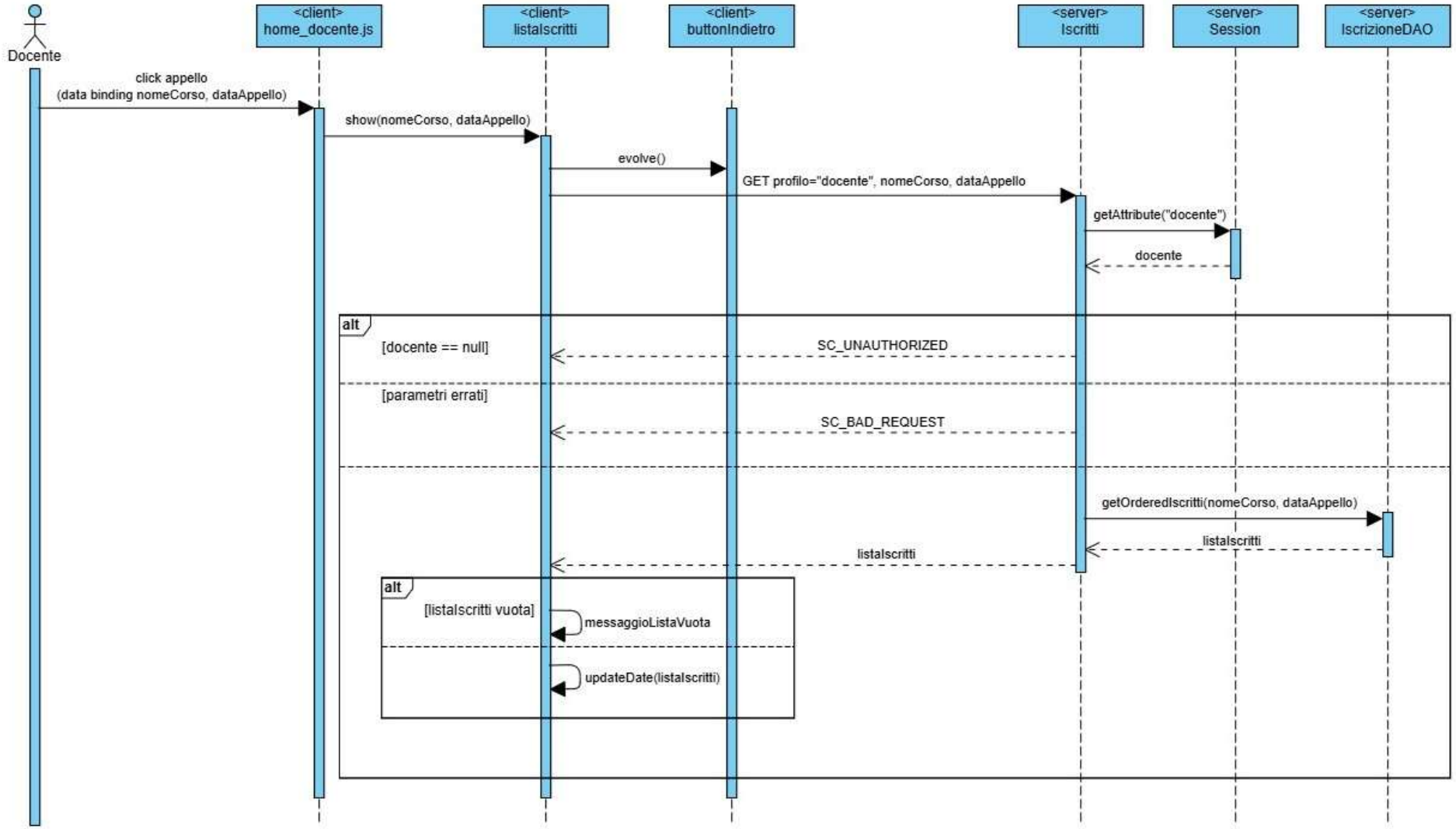
SEQUENCE DIAGRAM – RIFIUTA VOTO (STUDENTE)

Di seguito il sequence diagram per la risposta al drag and drop dell'esito esame sul cestino, da parte dello studente. Al termine si esegue esitoEsame.show(), per maggiori dettagli a riguardo si faccia riferimento al sequence diagram precedente.



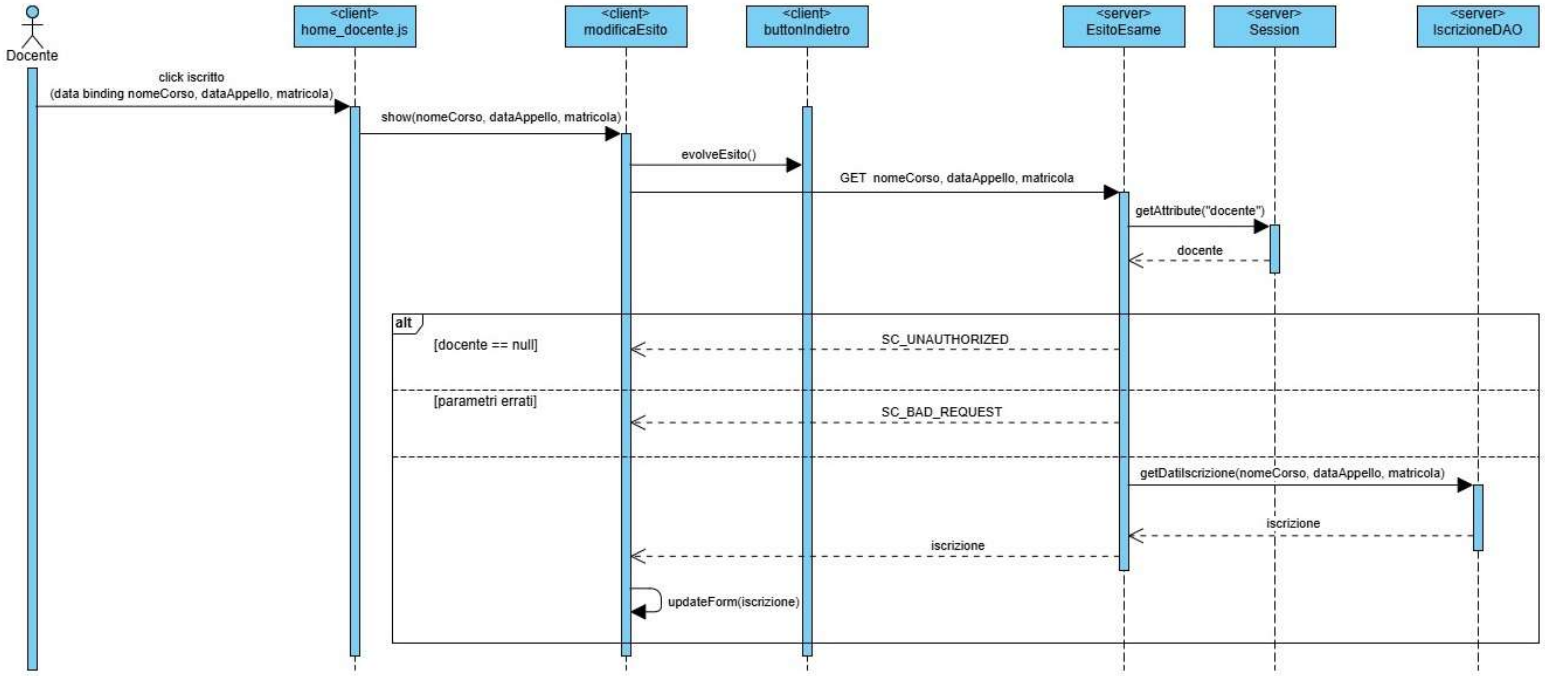
SEQUENCE DIAGRAM – CLICK ANCHOR APPELLO (DOCENTE)

Di seguito il sequence diagram della risposta al click su una data d'appello, per il solo docente, esso porta alla visualizzazione della lista degli iscritti a quell'appello.



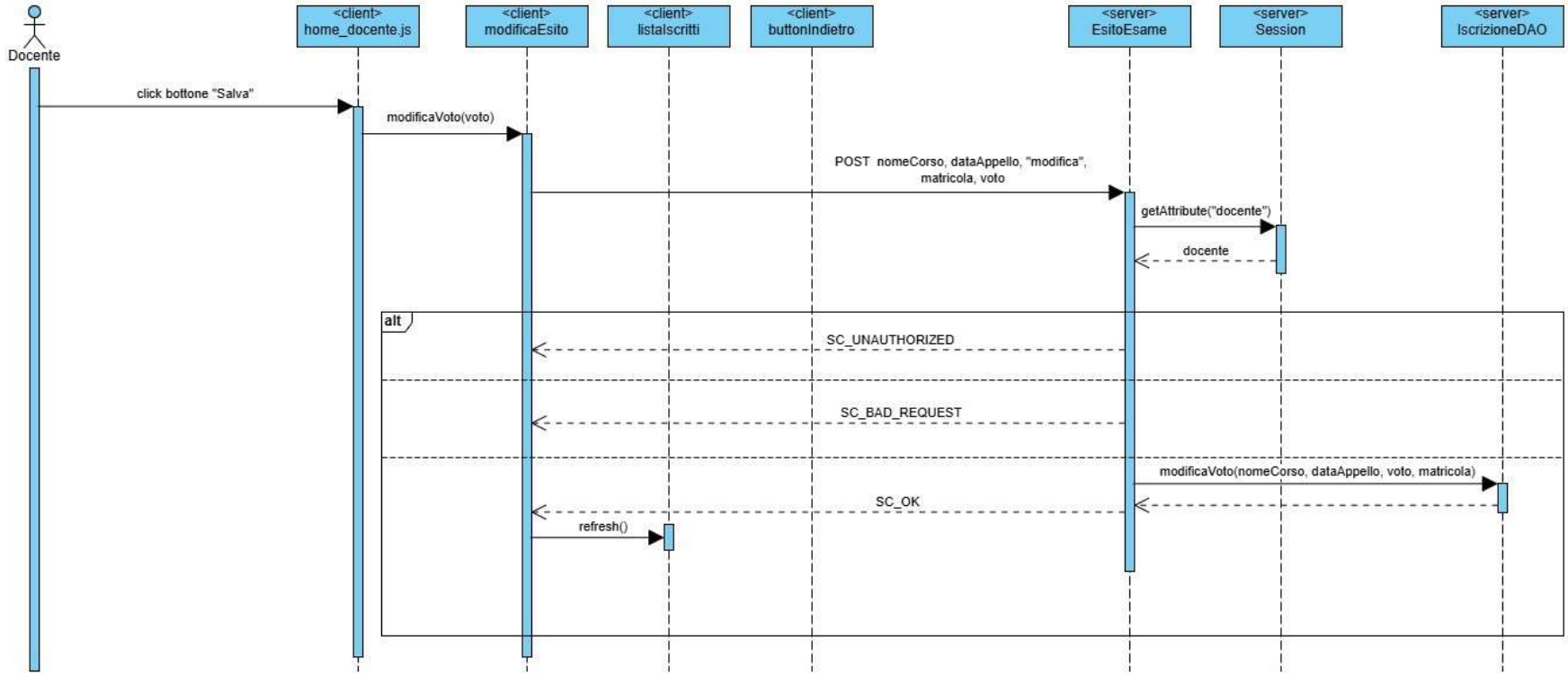
SEQUENCE DIAGRAM – CLICK ANCHOR ISCRITTO (DOCENTE)

Di seguito il sequence diagram della risposta all'evento di click sull'icona di modifica nella vista iscritti per il solo docente. Esso porta alla vista del form per inserire il voto per lo studente selezionato.



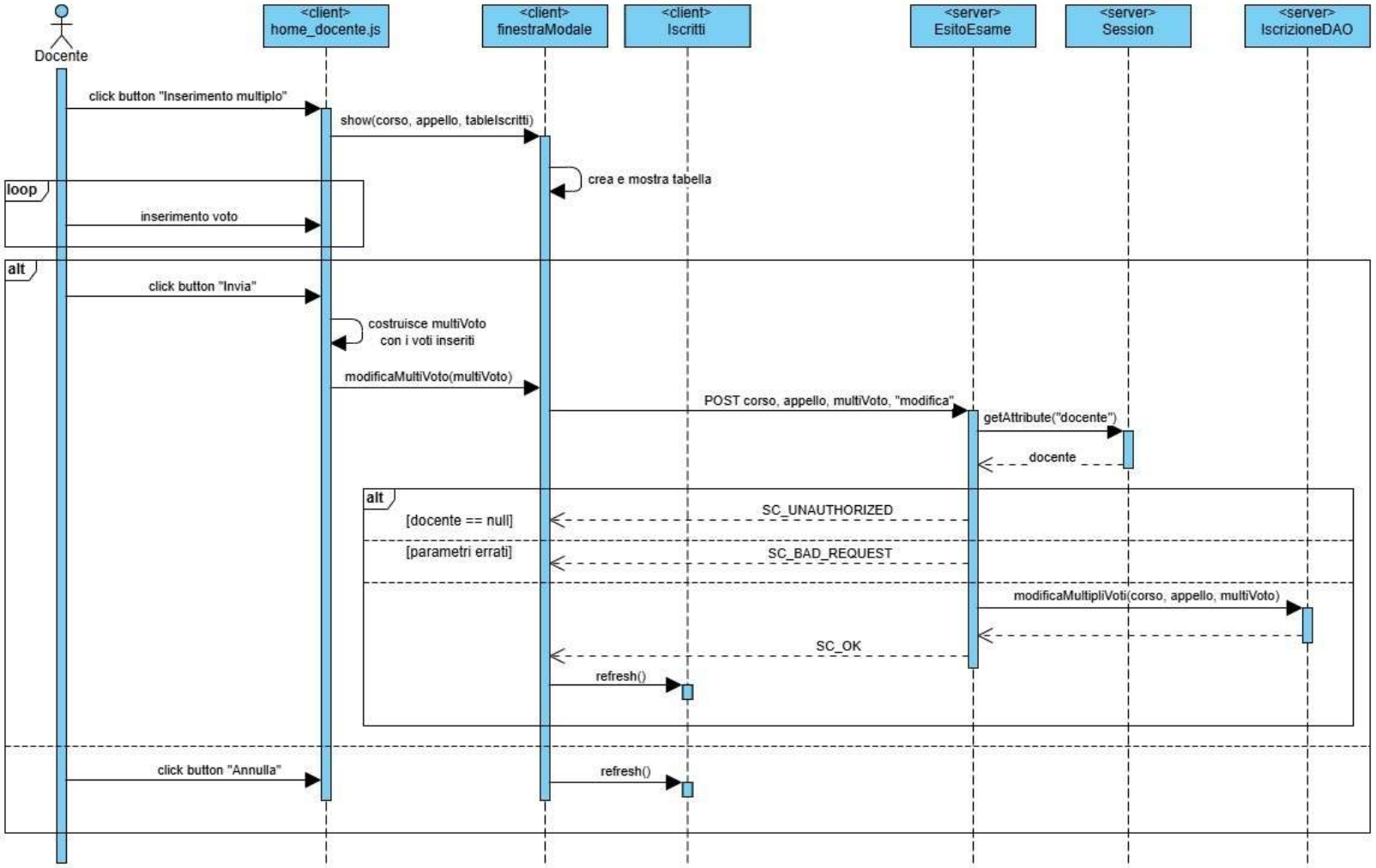
SEQUENCE DIAGRAM – MODIFICA VOTO (DOCENTE)

Di seguito il sequence diagram della risposta alla pressione del pulsante salva nella vista di modifica del voto per il solo docente.



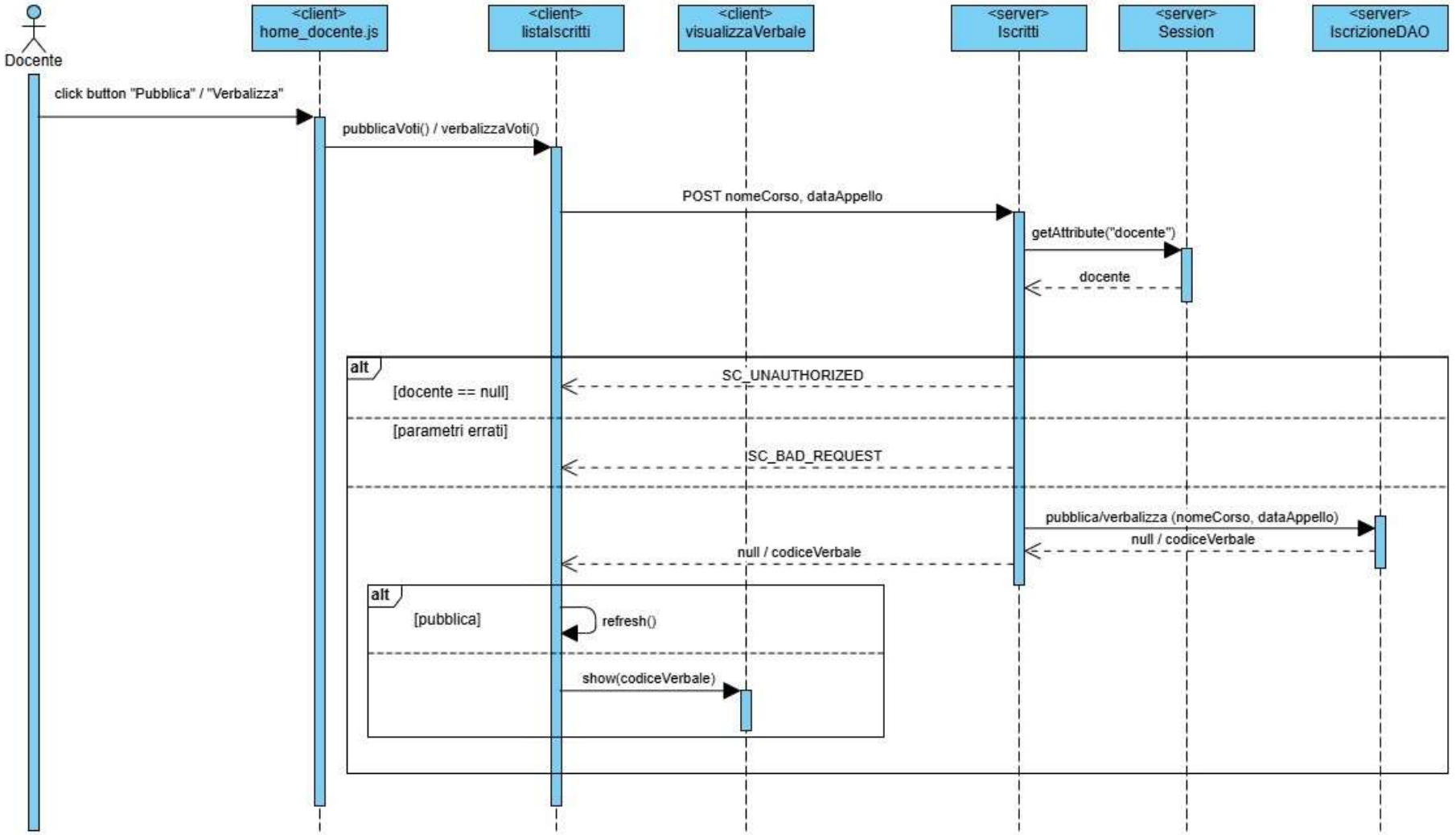
SEQUENCE DIAGRAM – INSERIMENTO MULTIPLO (DOCENTE)

Di seguito il sequence diagram della procedura di inserimento multiplo dei voti da parte del docente.



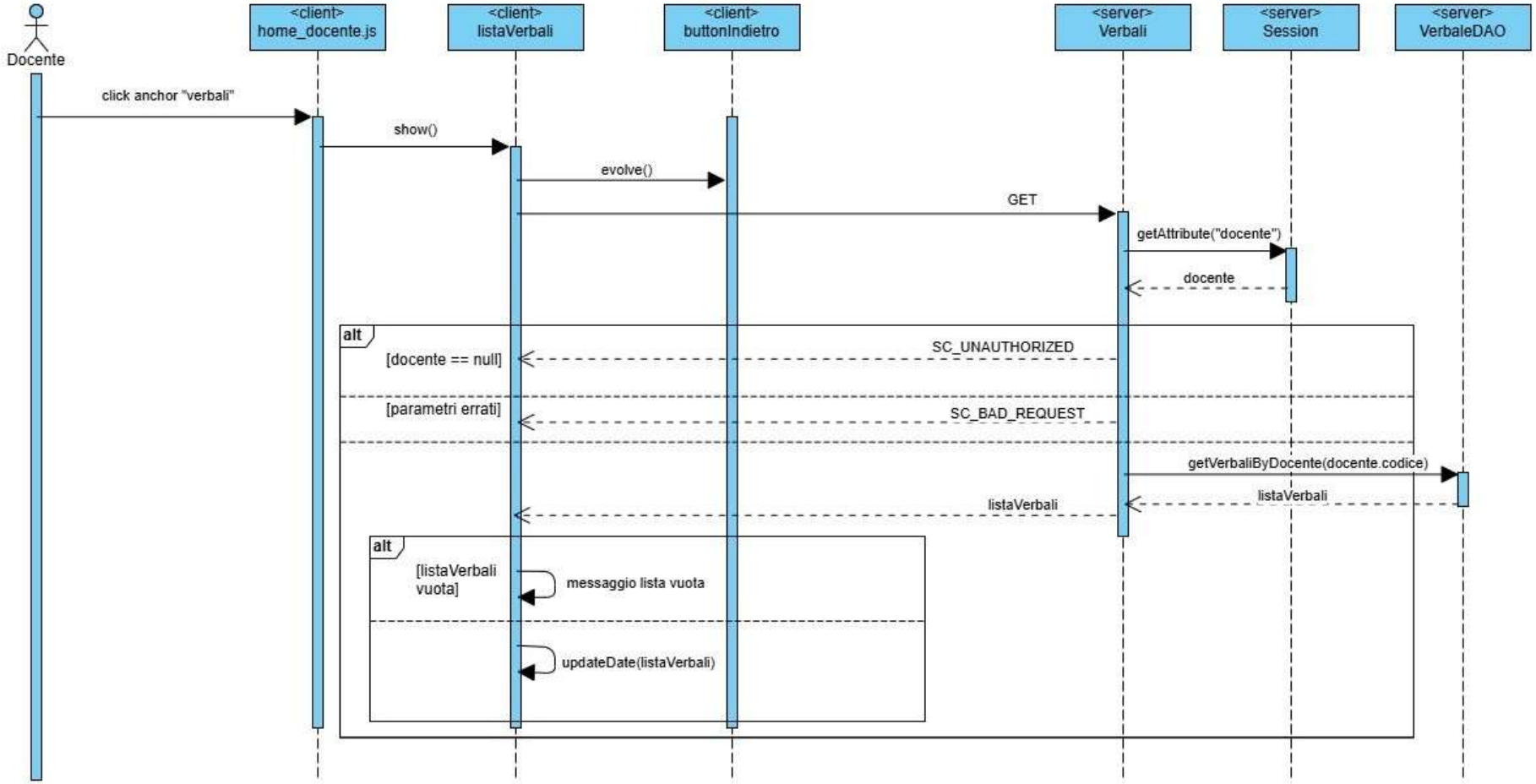
SEQUENCE DIAGRAM – PUBBLICA / VERBALIZZA (DOCENTE)

Di seguito il sequence diagram della risposta al click di uno dei due pulsanti “Pubblica” o “Verbalizza” della pagina iscritti del docente.



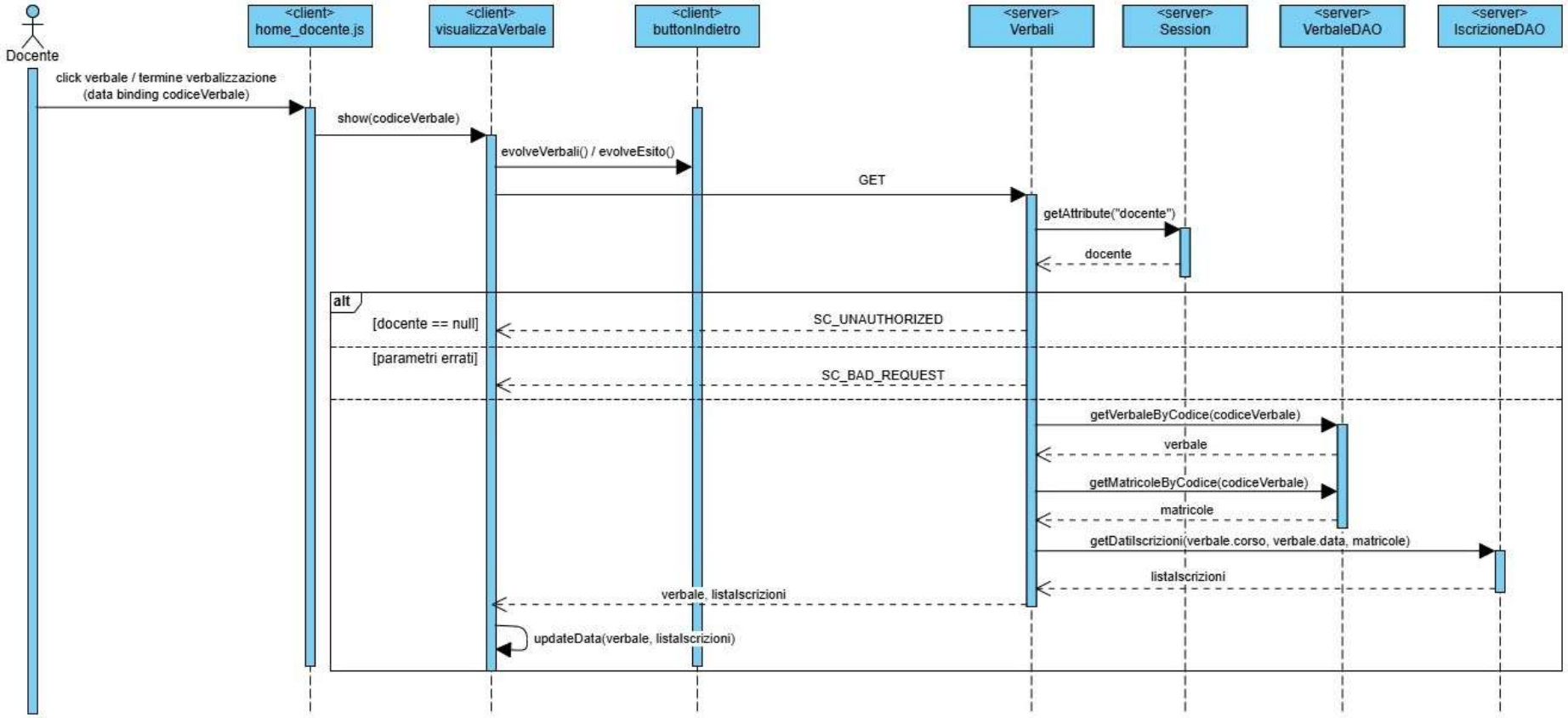
SEQUENCE DIAGRAM – CLICK ANCHOR VERBALI (DOCENTE)

Di seguito il sequence diagram della risposta al click sull'anchor "Verbali" da una qualunque vista, per il solo docente. Ciò comporta la comparsa di una nuova vista con tutti i verbali del docente.



SEQUENCE DIAGRAM – VISUALIZZA VERBALE (DOCENTE)

Di seguito il sequence diagram per la visualizzazione di un verbale del docente. Tale vista è mostrata sia in seguito al click del verbale corrispondente nella tabella dei verbali, sia al termine della sequenza di verbalizzazione.



3 – Strumenti di sviluppo

Per la realizzazione del progetto sono stati utilizzati i seguenti strumenti:

- **Motore database**: MySQL
- **Workbench database**: MySQL WorkBench 8.0, JetBrains DataGrip 2025
- **Ambiente di sviluppo**: Eclipse for Enterprise Java and Web Developers 2025
- **Version control:** Git, GitHub
- **Diagrammi IFML ed entità-relazione:** draw.io
- **Diagrammi di sequenza:** Visual Paradigm Online
- **Relazione documentazione:** Microsoft Word

Nell'applicazione sono presenti icone scaricate dal sito flaticon.com