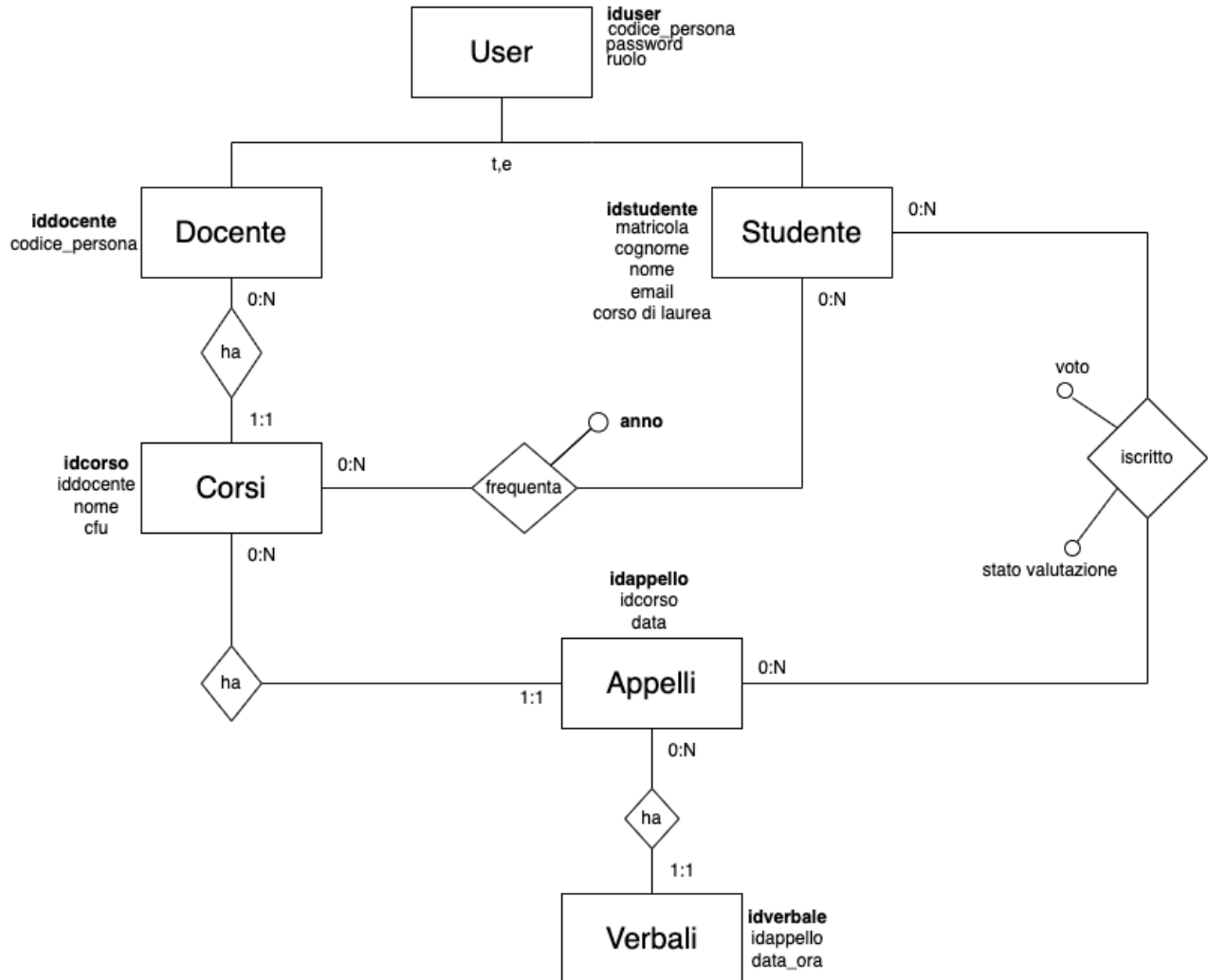


Basi di Dati - Analisi

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 in modo alfabetico decrescente e poi una **data d'appello del corso** scelto selezionata 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.** 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.

Entities, attributes, relationships

Design del Database



Schema del Database

```
CREATE TABLE `user` (  
  `iduser` int NOT NULL AUTO_INCREMENT,  
  `codice_persona` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  `ruolo` varchar(45) NOT NULL,  
  PRIMARY KEY (`iduser`),  
  KEY `id_cod` (`iduser`,`codice_persona`)  
)
```

```
CREATE TABLE `docente` (  
  `iduser` int NOT NULL AUTO_INCREMENT,  
  `codice_persona` varchar(45) NOT NULL,  
  PRIMARY KEY (`iduser`),  
  UNIQUE KEY `codice_persona_UNIQUE` (`codice_persona`),  
  KEY `id_cod_idx` (`iduser`,`codice_persona`),  
  CONSTRAINT `id_cod` FOREIGN KEY (`iduser`,`codice_persona`) REFERENCES `user`  
  (`iduser`,`codice_persona`) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

```
CREATE TABLE `studente` (  
  `idstudente` int NOT NULL,  
  `matricola` varchar(45) NOT NULL,  
  `cognome` varchar(45) NOT NULL,  
  `nome` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `corsoLaurea` varchar(45) NOT NULL,  
  PRIMARY KEY (`idstudente`),  
  UNIQUE KEY `matricola_UNIQUE` (`matricola`),  
  CONSTRAINT `id` FOREIGN KEY (`idstudente`) REFERENCES `user` (`iduser`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
)
```

```
CREATE TABLE `corsi` (  
  `idcorso` int NOT NULL AUTO_INCREMENT,  
  `iddocente` int NOT NULL,  
  `nome` varchar(45) NOT NULL,  
  `cfu` int NOT NULL,  
  PRIMARY KEY (`idcorso`,`iddocente`),  
  KEY `doc_idx` (`iddocente`),  
  CONSTRAINT `doc` FOREIGN KEY (`iddocente`) REFERENCES `docente` (`iduser`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
)
```

```

CREATE TABLE `corsiStudente` (
  `idstudente` int NOT NULL,
  `idcorso` int NOT NULL,
  `anno` year NOT NULL,
  PRIMARY KEY (`idstudente`,`idcorso`,`anno`),
  KEY `idcorso_idx` (`idcorso`),
  CONSTRAINT `idcorsoo` FOREIGN KEY (`idcorso`) REFERENCES `corsi` (`idcorso`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `idstudentee` FOREIGN KEY (`idstudente`) REFERENCES `studente`
(`idstudente`) ON DELETE CASCADE ON UPDATE CASCADE
)

```

```

CREATE TABLE `appelli` (
  `idcorso` int NOT NULL,
  `idappello` int NOT NULL,
  `data` date NOT NULL,
  PRIMARY KEY (`idappello`),
  KEY `idx_appelli_idappello` (`idappello`),
  KEY `idcorso` (`idcorso`),
  CONSTRAINT `idcorso` FOREIGN KEY (`idcorso`) REFERENCES `corsi` (`idcorso`) ON
DELETE CASCADE ON UPDATE CASCADE
)

```

```

CREATE TABLE `dettaglioAppello` (
  `idappello` int NOT NULL,
  `idstudente` int NOT NULL,
  `voto`
enum('','assente','rimandato','riprovato','18','19','20','21','22','23','24','25','26','27','28','
29','30','30 e lode') NOT NULL DEFAULT '',
  `stato_valutazione` enum('non inserito','inserito','pubblicato','rifiutato','verbalizzato')
NOT NULL DEFAULT 'non inserito',
  PRIMARY KEY (`idappello`,`idstudente`),
  KEY `id_idx` (`idstudente`),
  CONSTRAINT `idapp` FOREIGN KEY (`idappello`) REFERENCES `appelli` (`idappello`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `idstud` FOREIGN KEY (`idstudente`) REFERENCES `studente`
(`idstudente`) ON DELETE CASCADE ON UPDATE CASCADE
)

```

```

CREATE TABLE `verbale` (
  `idverbale` int NOT NULL,
  `idappello` int NOT NULL,
  `data_ora` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`idverbale`),
  KEY `idappello_idx` (`idappello`),
  CONSTRAINT `idappello` FOREIGN KEY (`idappello`) REFERENCES `appelli` (`idappello`)
ON DELETE CASCADE ON UPDATE CASCADE
)

```

Analisi dei requirements

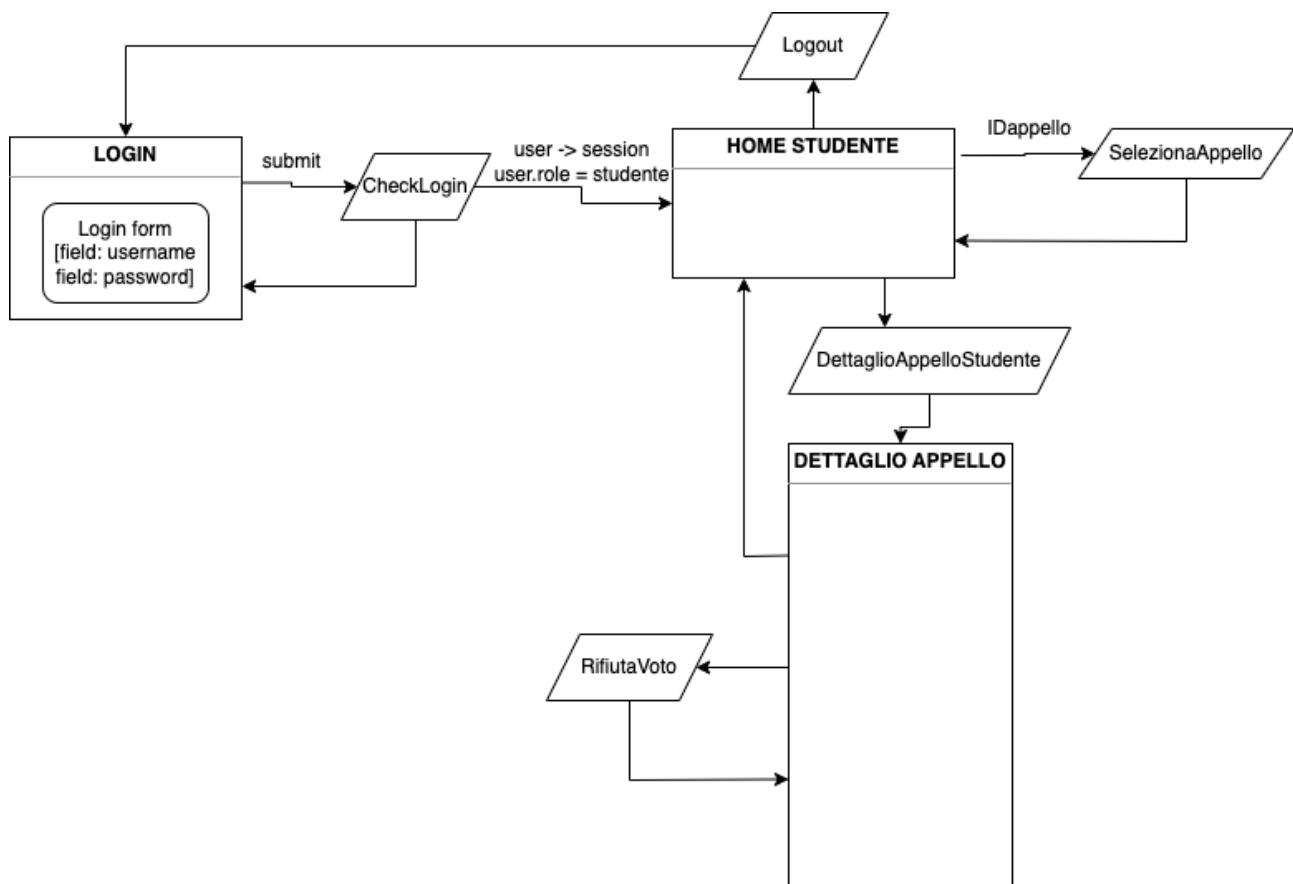
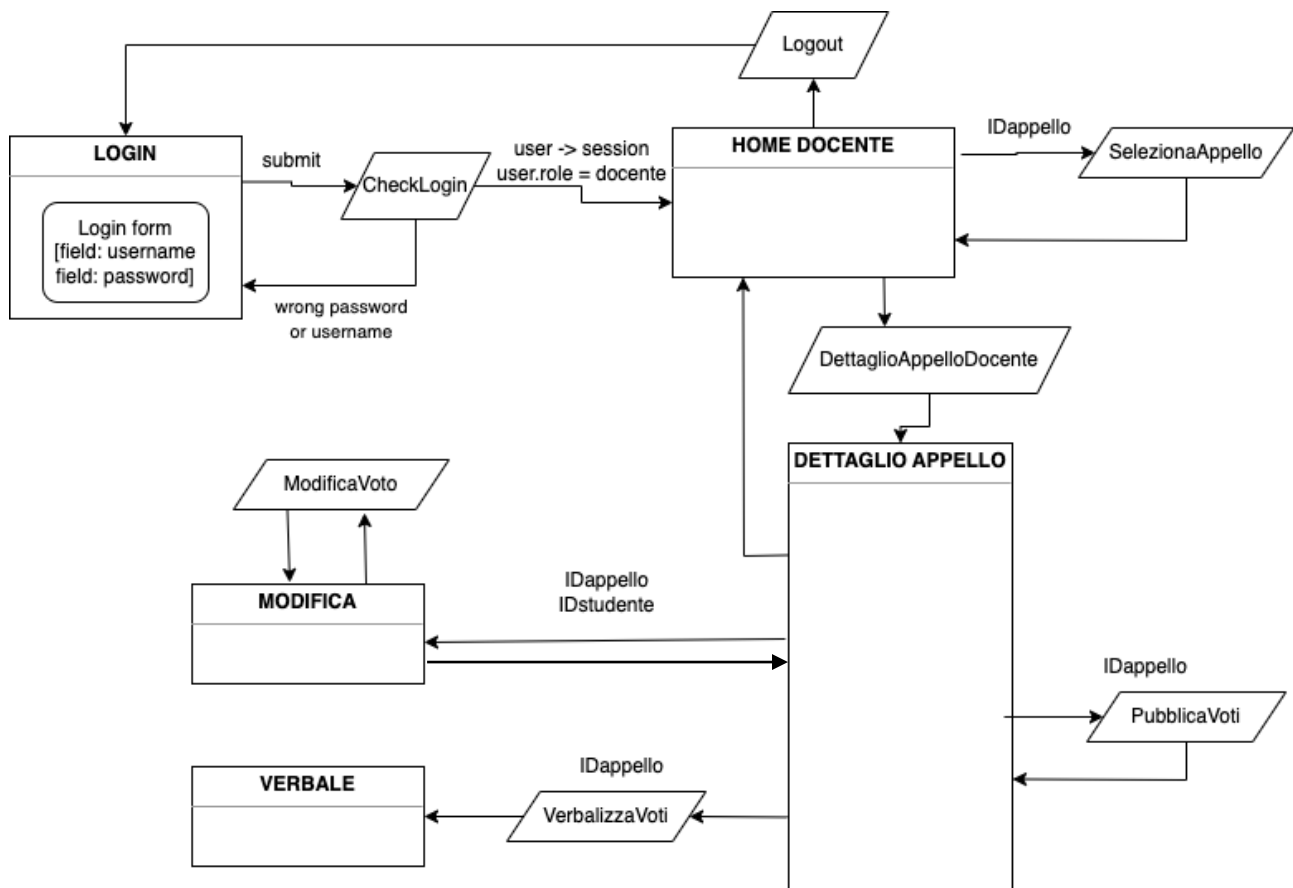
Un'applicazione permette di verbalizzare gli esiti degli esami di un appello. Il docente **accede tramite login** e **seleziona** nella **HOME page** un **corso** da **una lista dei propri corsi ordinata in modo alfabetico decrescente** e poi **una data d'appello del corso scelto selezionata da un elenco 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. **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**.

Pages (views), view components, events, actions

Completamento delle specifiche

- La pagina di default contiene il form di login
- Viene introdotta la possibilità di fare il logout
- È possibile tornare alla pagina precedente tramite un bottone “Torna indietro”
- Non si possono pubblicare voti se non è presente almeno un’entry con stato di valutazione inserito
- Non si posso verbalizzare voti se non è presente almeno un’entry con stato di valutazione pubblicato o rifiutato

Application Design



Components

Controllers (servlets) HTML

- CheckLogin
- DettaglioAppelloDocente (DOCENTE)
- DettaglioAppelloStudente (STUDENTE)
- GoToHomeDocente (DOCENTE)
- GoToHomeStudente (STUDENTE)
- Logout
- ModificaVoto (DOCENTE)
- PubblicaVoti (DOCENTE)
- RifiutaVoto (STUDENTE)
- SelezionaAppello
- SortOrder (DOCENTE)
- VerbalizzaVoti (DOCENTE)

Controllers (servlets) RIA

- CheckLogin
- DettaglioAppello
- GoToHome
- Logout
- ModificaVoto
- PubblicaVoti
- RifiutaVoto
- SelezionaAppello
- SortOrder
- VerbalizzaVoti
- InserimentoMultiplo

Views (templates) HTML

- DettaglioAppelloDocente
- DettaglioAppelloStudente
- HomeDocente
- HomeStudente
- ModificaVoto
- Verbale

Model objects (beans)

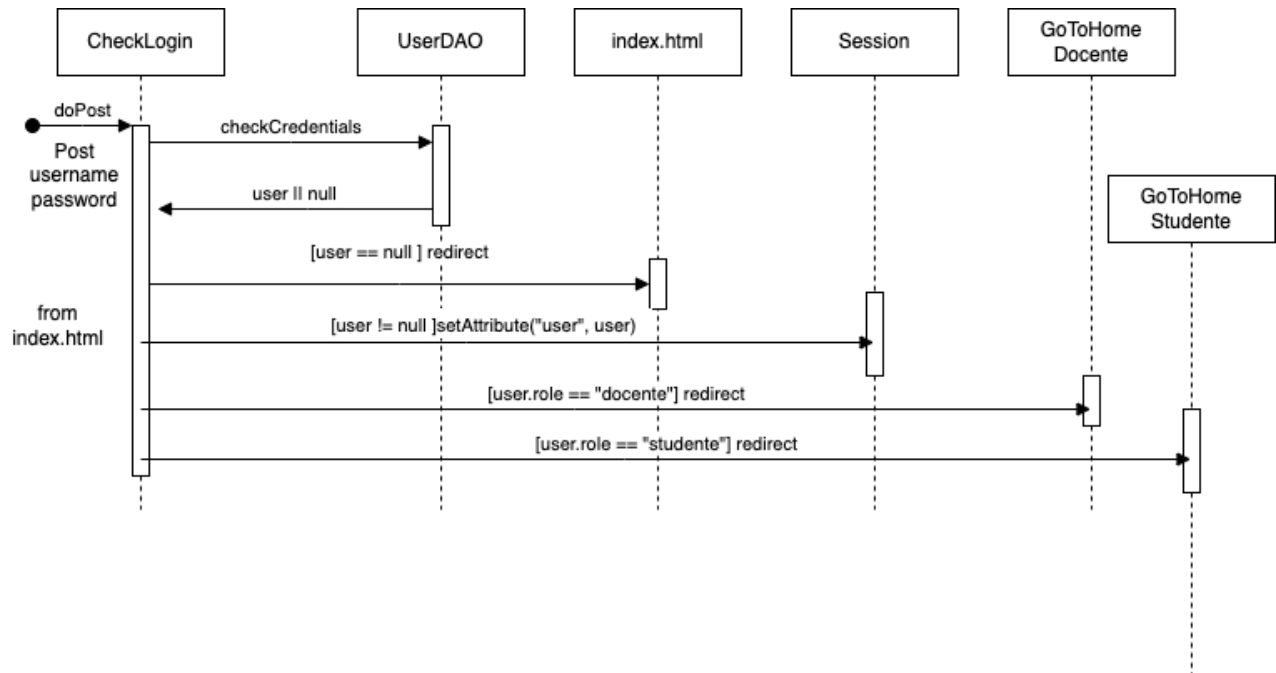
- AppelloBean
- CorsoBean
- StudentiAppelloBean
- User
- VerbaleBean

DAO (classes)

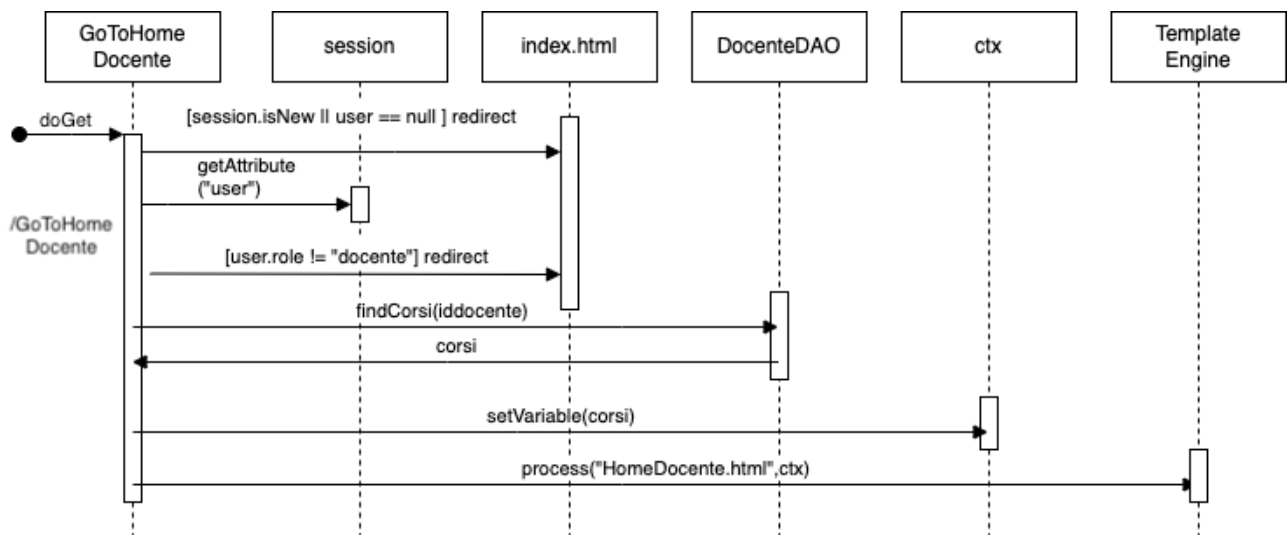
- AppelloDAO
 - findAppelli
 - hasStudente
 - findStudenti
 - findDatiStud
 - findNameDate
 - UpdateVotoStud
 - PubblicaVoti
 - VerbalizzaVoti
 - rifiutaVoto
 - inserimentoMultiplo (RIA)
 - UpdateVotoStudMultiplo (RIA)
- DocenteDAO
 - findCorsi
 - hasCorso
 - hasAppello
- StudenteDAO
 - findCorsi
 - hasCorso
- UserDAO
 - checkCredentials

Sequence Diagrams

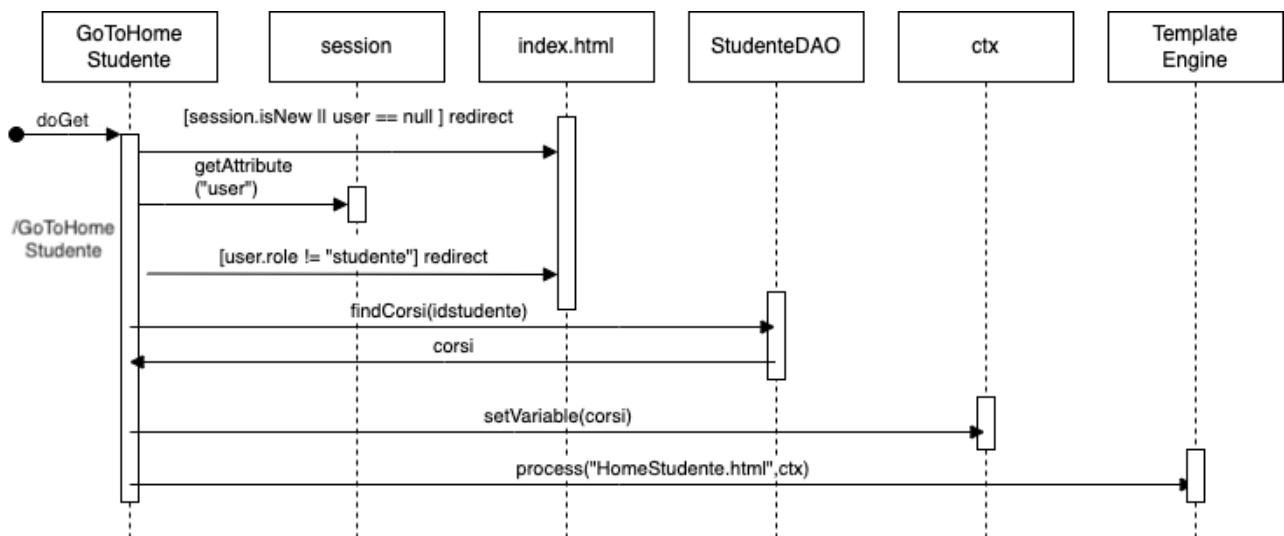
Event - Login



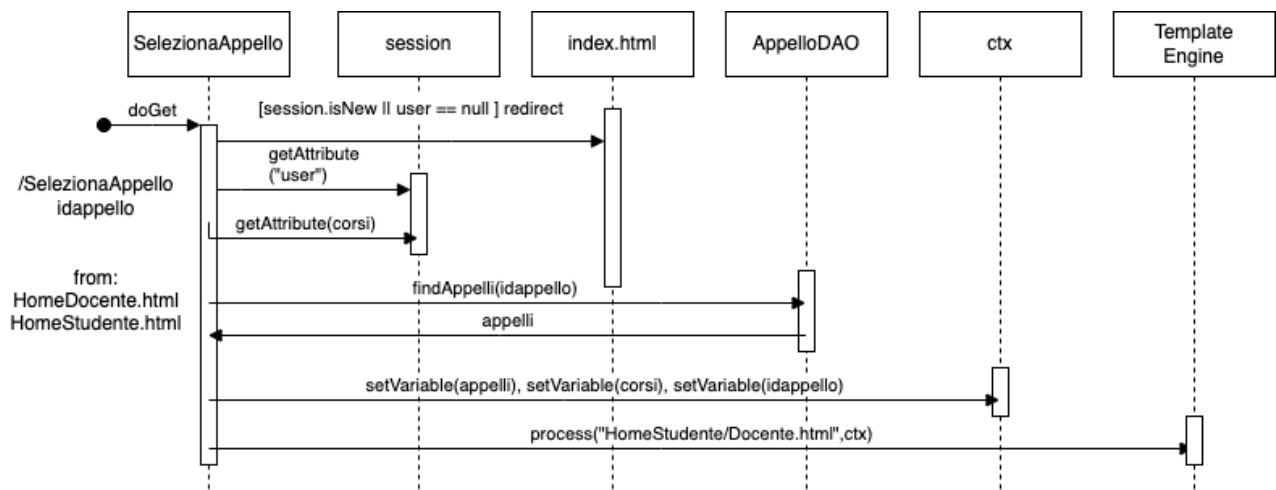
Event - GoToHomeDocente



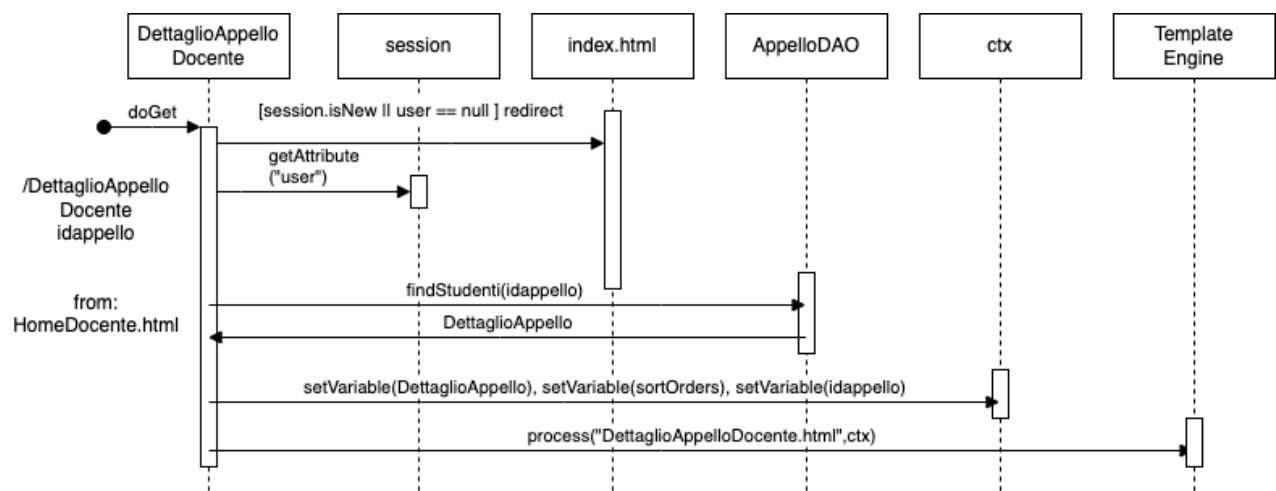
Event - GoToHomeStudiante



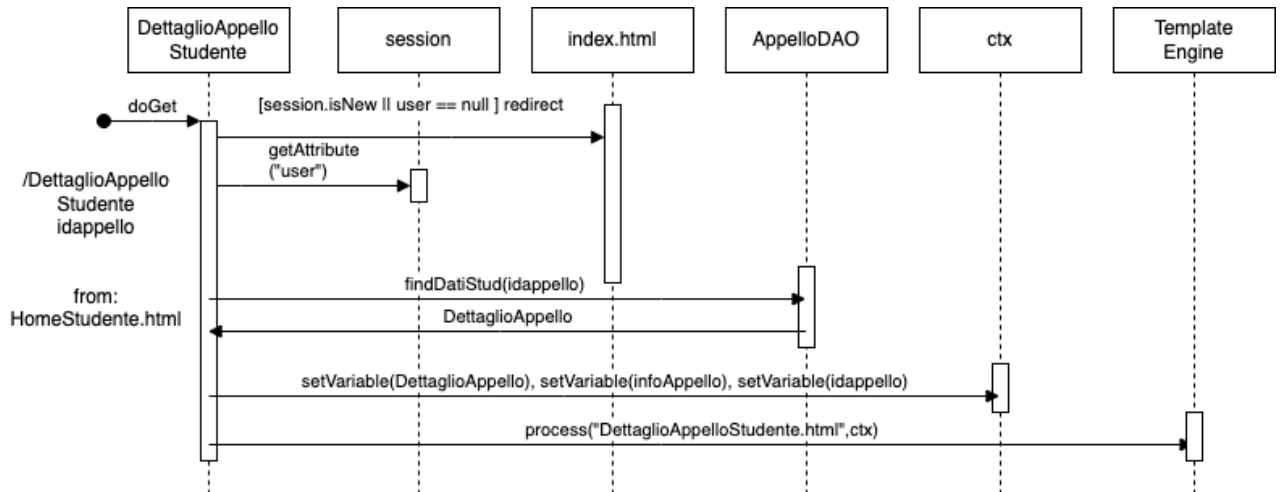
Event - SelezionaAppello



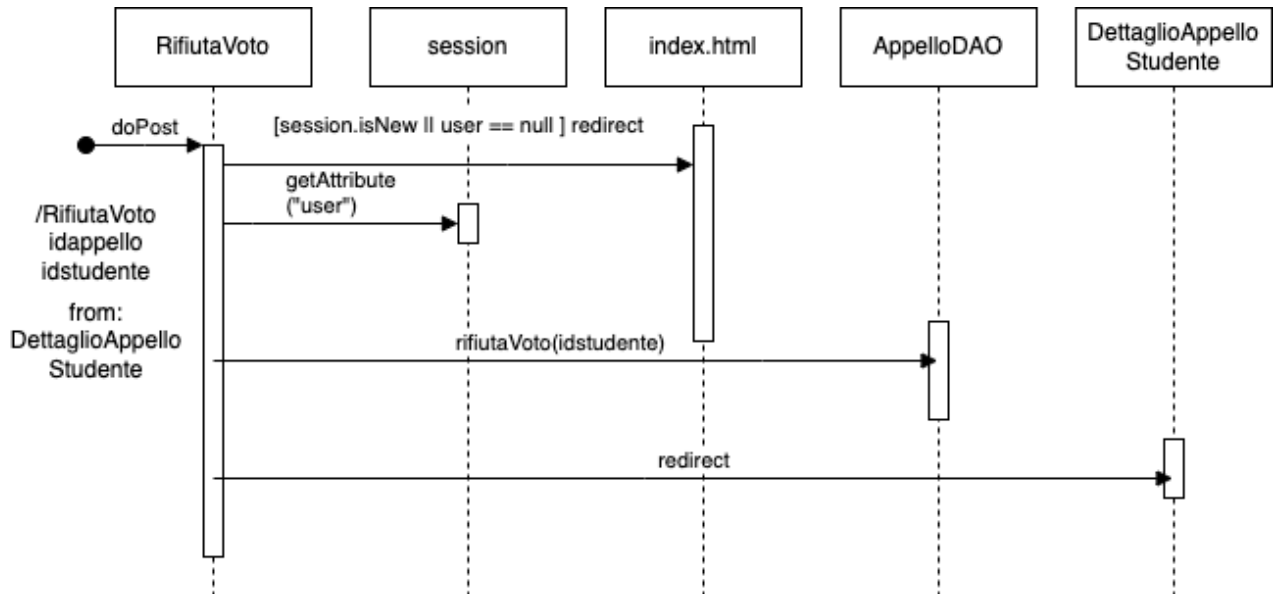
Event - DettaglioAppelloDocente



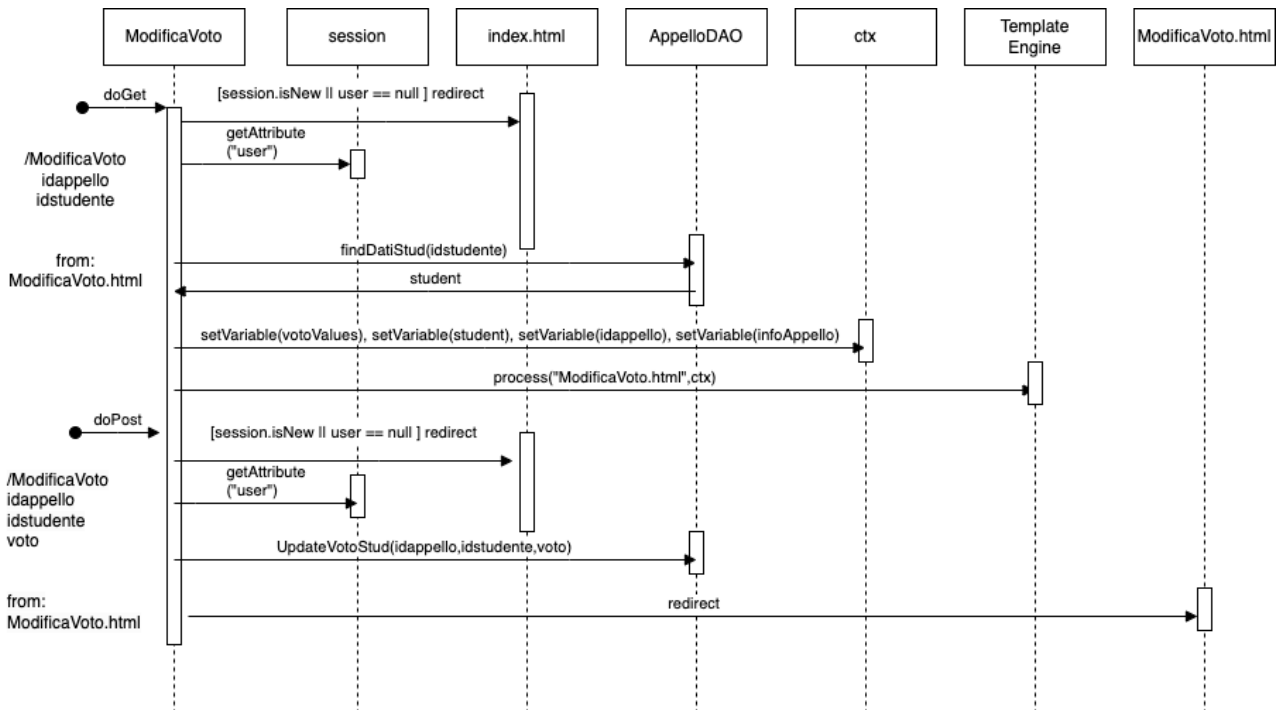
Event - DettaglioAppelloStudente



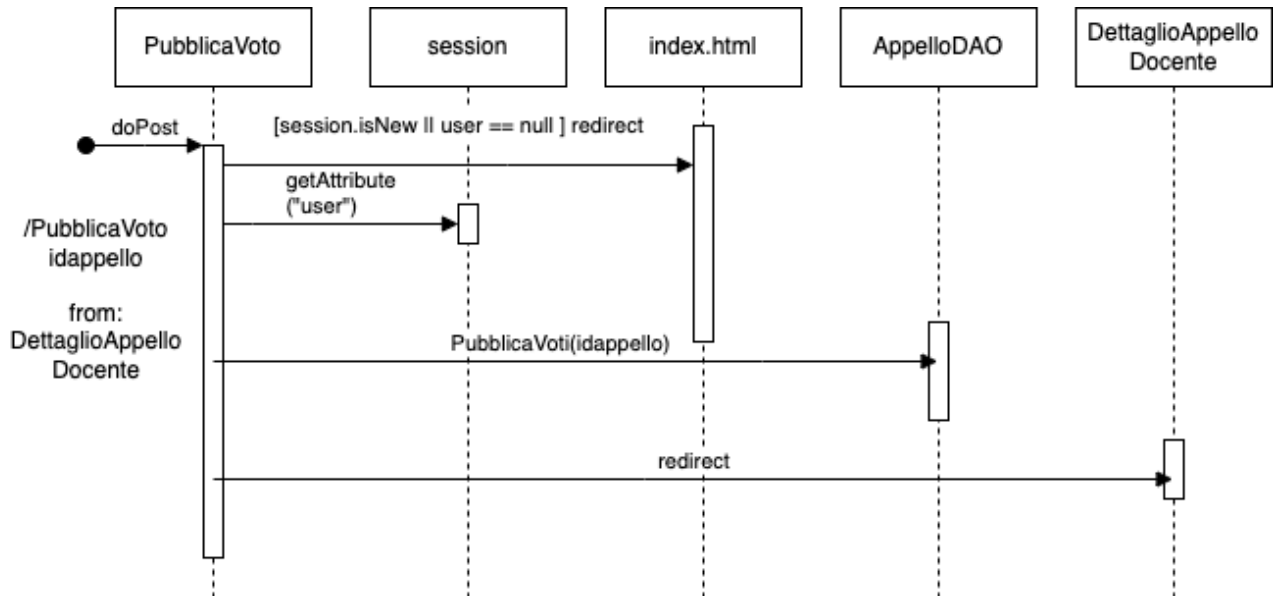
Event - RifiutaVoto



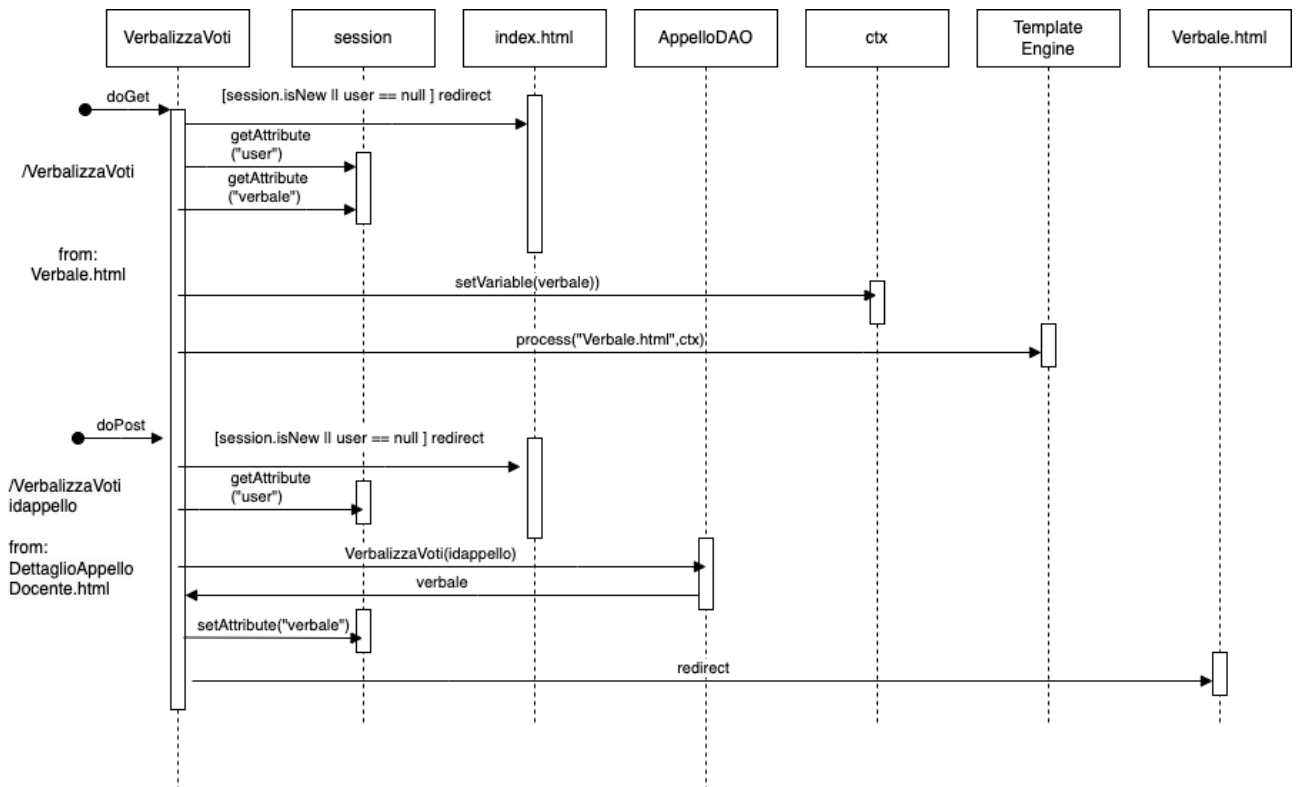
Event - ModificaVoto



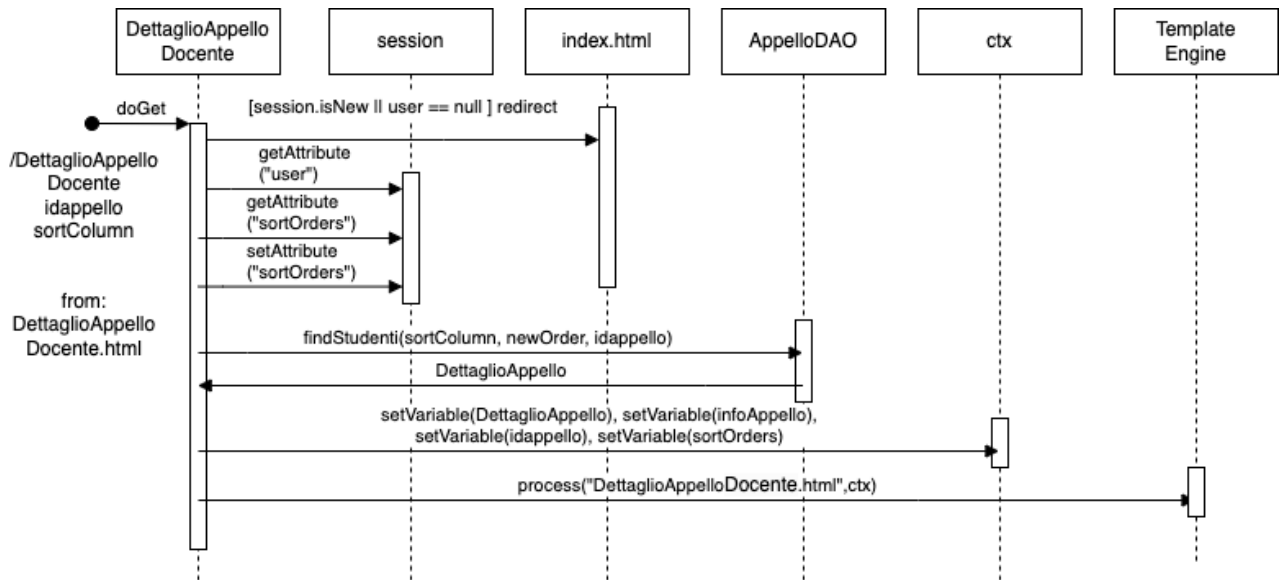
Event - PubblicaVoti



Event - VerbalizzaVoti

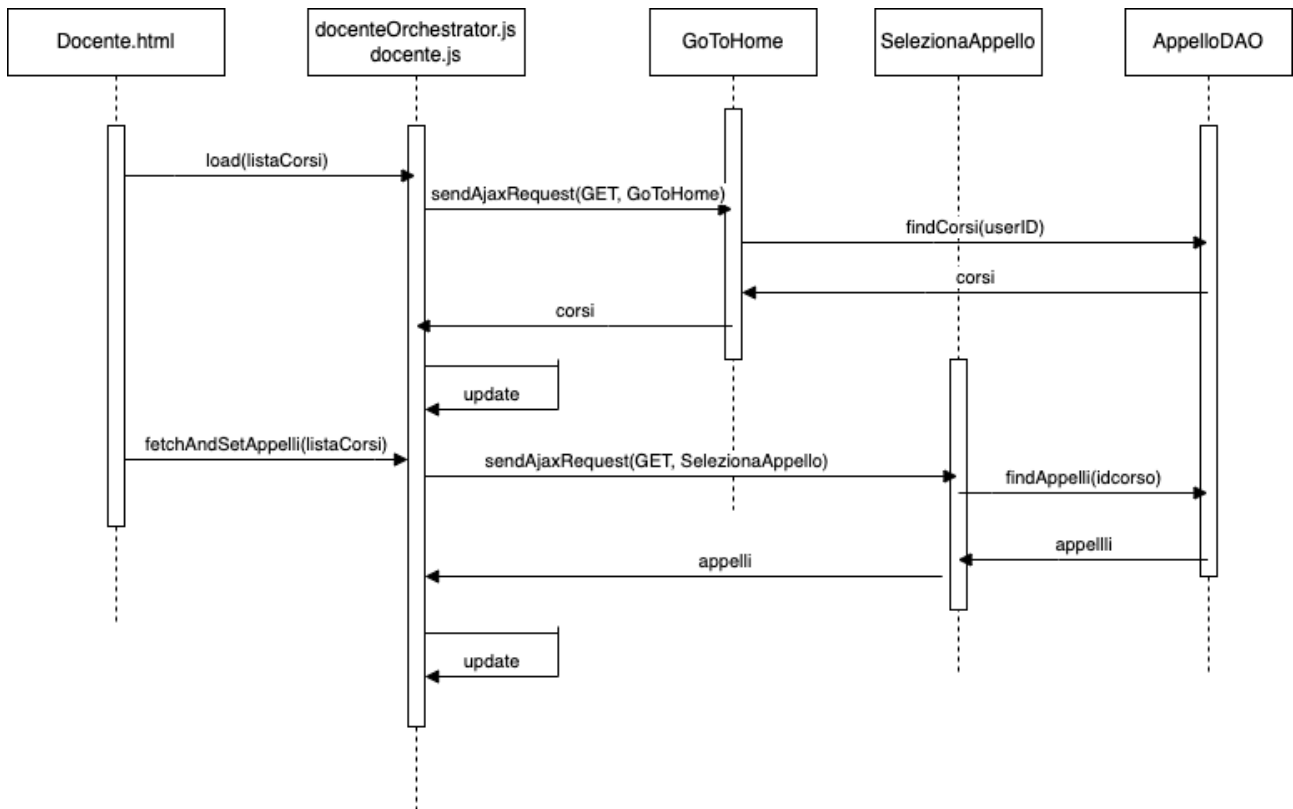


Event - SortOrder

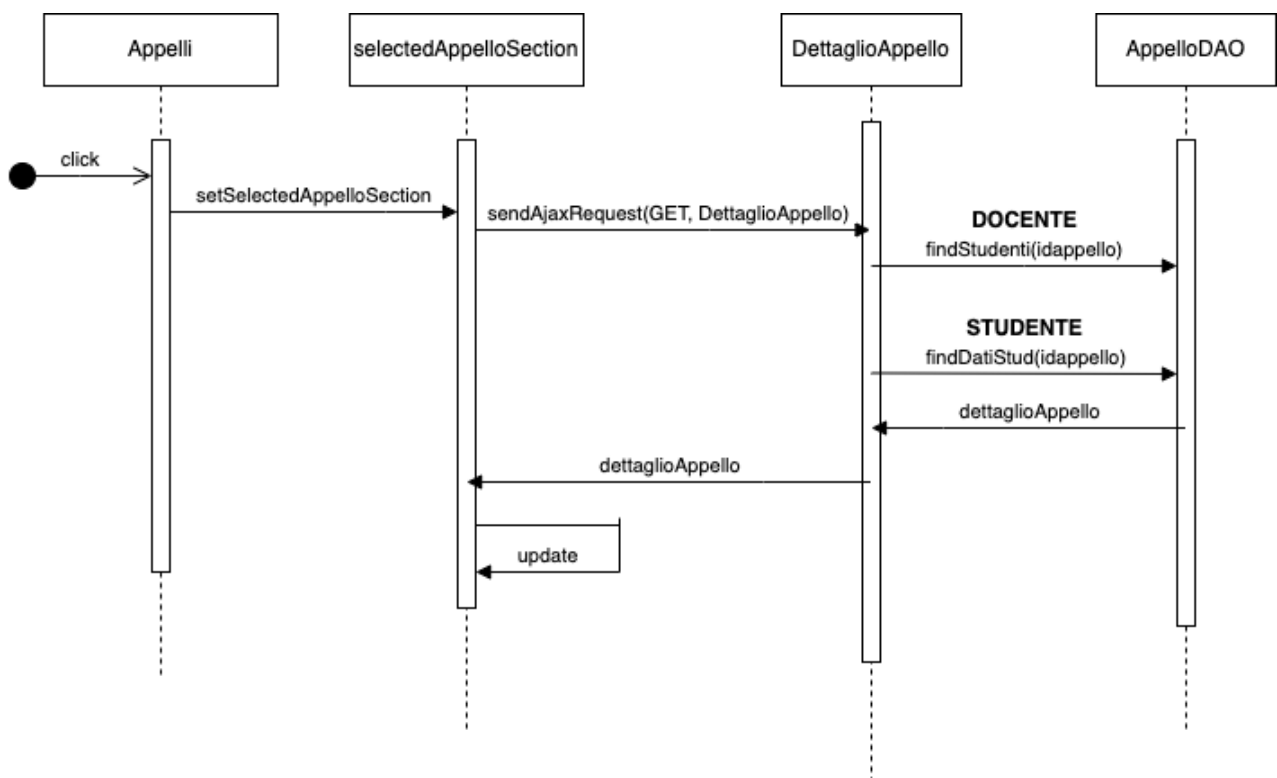


RIA

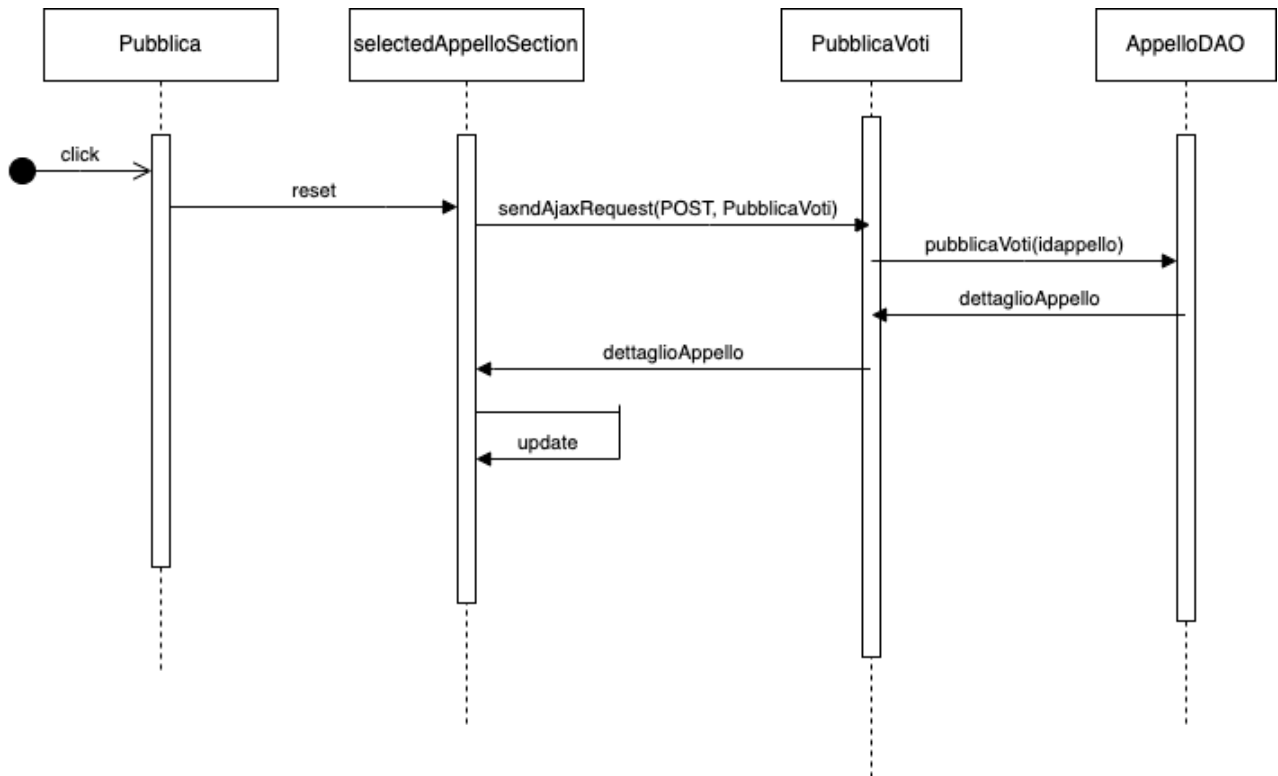
Event - Home + Seleziona



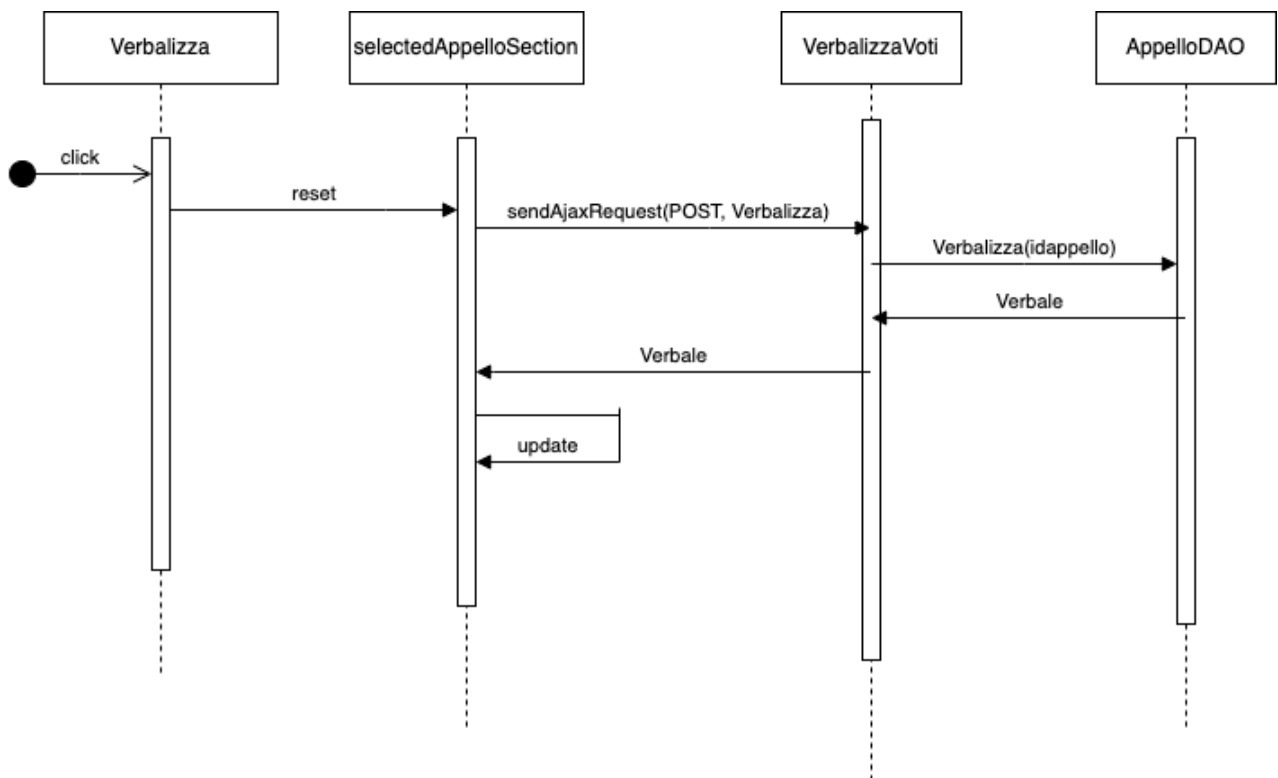
Event - DettaglioAppello



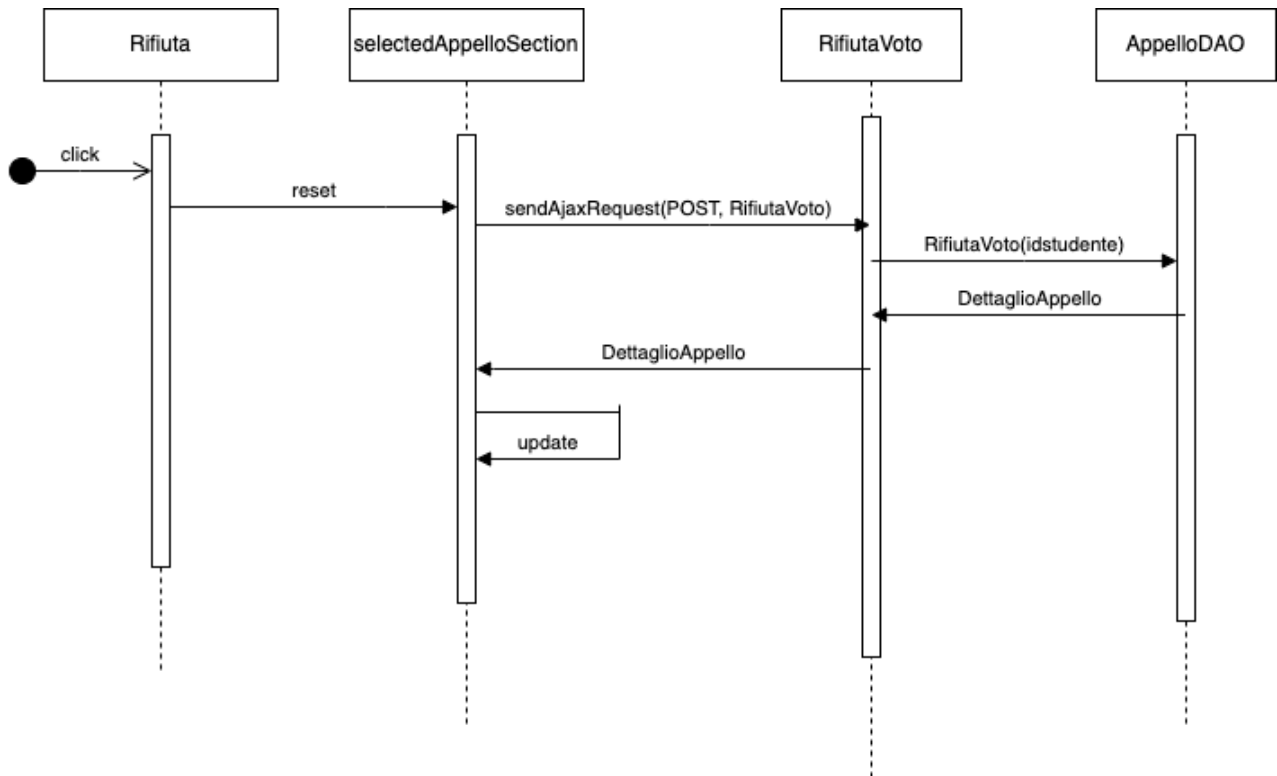
Event - Pubblica



Event - Verbalizza



Event - Rifiuta



Event - Modifica con finestra modale

