Progetto di Tecnologie Informatiche per il web Anno 2020 / 2021

Versione PureHTML

Professore Piero Fraternali

Alessandro Mazza

Cod. Persona 10670112 Matricola 908651

Davide Marinotto

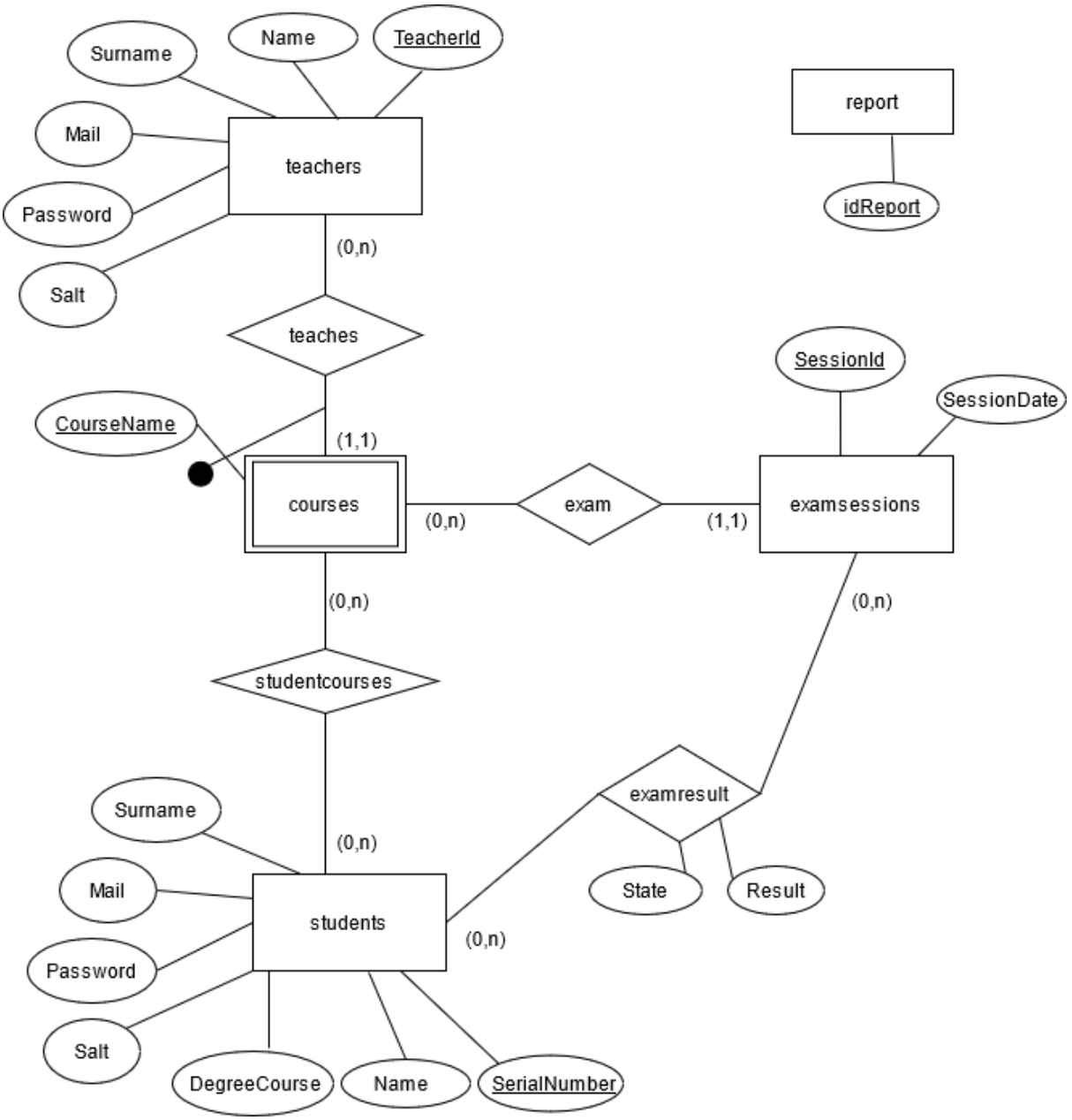
Cod. Persona 10675966 Matricola 909291

Analisi dei Dati

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: , assente, rimandato, riprovato, 18, 19, …, 30, 30 e lode. 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 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 iscriversi 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.

Legenda: entità, attributi, relazioni

Schema ER


Surname
Name
TeacherId
Mail
teachers
Password
Salt
(0,n)
teaches
report
idReport
SessionId
SessionDate
CourseName
(1,1)
courses
(0,n)
exam
(1,1)
examsessions
(0,n)
(0,n)
studentcourses
examresult
Surname
(0,n)
Mail
students
(0,n)
State
Result
Password
Salt
DegreeCourse
Name
SerialNumber

Local Database Schema

- **teachers**
  ```
  CREATE TABLE `teachers` (
  `TeacherId` int NOT NULL,
  `Name` varchar(45) NOT NULL,
  `Surname` varchar(45) NOT NULL,
  `Mail` varchar(320) NOT NULL,
  `Password` varchar(64) NOT NULL,
  `Salt` varchar(12) NOT NULL,
  PRIMARY KEY (`TeacherId`)
   )
  ```
- **students**
  ```
  CREATE TABLE `students` (
    `SerialNumber` int NOT NULL,
    `Name` varchar(45) NOT NULL,
    `Surname` varchar(45) NOT NULL,
    `Mail` varchar(320) NOT NULL,
    `Password` varchar(64) NOT NULL,
    `DegreeCourse` varchar(45) NOT NULL,
    `Salt` varchar(12) NOT NULL,
    PRIMARY KEY (`SerialNumber`)
  )
  ```
- **studentcourses**
  ```
  CREATE TABLE `studentcourses` (
    `StudentSerialNumber` int NOT NULL,
    `CourseName` varchar(45) NOT NULL,
    PRIMARY KEY (`CourseName`,`StudentSerialNumber`)
  )
  ```
- **report**
  ```
  CREATE TABLE `report` (
    `idReport` int NOT NULL DEFAULT '0',
    PRIMARY KEY (`idReport`)
  )
  ```
- **examsessions**
  ```
  CREATE TABLE `examsessions` (
    `SessionId` int NOT NULL,
    `CourseName` varchar(45) NOT NULL,
    `SessionDate` date NOT NULL,
    PRIMARY KEY (`SessionId`)
  )
  ```

- **examresult**

```
CREATE TABLE `examresult` (
  `SessionId` int NOT NULL,
  `StudentSerialNumber` int NOT NULL,
  `State` varchar(45) NOT NULL DEFAULT 'NON INSERITO',
  `Result` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`SessionId`,`StudentSerialNumber`)
)
```

- **courses**

```
CREATE TABLE `courses` (
  `CourseName` varchar(45) NOT NULL,
  `TeacherId` int NOT NULL,
  PRIMARY KEY (`CourseName`,`TeacherId`)
)
```

Azioni

Elenchiamo e descriviamo brevemente le azioni che si possono fare sulle interfacce. Suddividiamo quest'ultime in base al contesto in cui possono essere invocate,definiamo quindi tre gruppi Azioni in Comune, Azioni Insegnante, Azioni Studente

Azioni in Comune

Login
- Verifica che le credenziali appartengono a un insegnante o a uno studente

Logout
- Cancella la sessione dell utente

Azioni Insegnante

Selezione **data appello** di un corso
- Mostra l'elenco degli iscritti a tale appello, e tutte le azioni che scatenano gli eventi successivi

Click su intestazione **elenco iscritti**
- Riordinamento

Click su bottone **Modifica**
- Assegna Valutazione a uno studente
- Imposta lo stato ad INSERITO

Click su bottone **Pubblica**
- Pubblica le valutazioni registrate degli studenti iscritti
- Imposta lo stato a PUBBLICATO

Click su bottone **Verbalizza**
- Verbalizza le valutazioni registrate degli studenti iscritti
- Imposta lo stato a VERBALIZZATO

Click su bottone **Ritorna alla home**
- Riporta alla pagina con l'elenco dei corsi

Azioni Studente

Selezione **data appello** di un corso
- Mostra la valutazione se disponibili se disponibile oppure un messaggio di cortesia

Click su bottone **Rifiuta**
- Verbalizza le valutazioni assegnata dall'insegnante
- Imposta lo stato a RIFIUTATO

Click su bottone **Ritorna alla home**
- Riporta alla pagina con l'elenco dei corsi

Gestione azioni Client Server

Client side	Server side	
Azione	**Controller**	**Azione**
Login	/LoginAction POST username password	Controllo credenziali
Logout	/LogoutAction GET	Fine Sessione
Ritorna alla home	/HomeTeacher GET	Ritorno alla Home, con estrazione elenco corsi
Home	/HomeTeacher GET	Estrazione elenco corsi tenuti l'insegnante e i rispettivi appelli
Seleziona data appello	/Iscritti GET examSessionID, tableOrder, ascending	Estrazione elenco studenti iscritti all'appello examSessionID
Click intestazione tabella iscritti	/Iscritti GET examSessionID, tableOrder, ascending	Riordino dell'elenco studenti iscritti tramite nuovo tableOrder
Modifica al voto	/ExamSessionResult GET studentId, examSessionID	Estrazione i dati dello studente con matricola studentId iscritto all'appello examSessionID
Invia modifica	/ExamSessionResult POST setResultSessionID setResultStudentID result	Inserimento della valutazione result allo studente setResultStudentID nell'appello setResultSessionID
Pubblica risultato	/Iscritti POST examSessionId	Pubblica le valutazioni della sessione
Verbalizza risultato	/Verbale GET examSessionId	Verbalizza le valutazioni della sessione e restituisce la pagina del verbale
Home	/HomeStudent GET	Estrazione elenco corsi e appelli a cui lo studente è iscritto
Ritorna alla home	/HomeStudent GET	Ritorno alla Home, con estrazione elenco corsi

Seleziona appello	/Esito GET examSessionID	Estrazione dati studente riferiti all'appello identificato da examSessionID
Rifiuto valutazione	/Esito POST sessionId, refuseState	Rifiuta la valutazione assegnata all'appello sessionId

DAO & model objects

Controllers

- GoToEsito
- GoToHomeStudent
- GoToHomeTeacher
- GoToIscritti
- GoToLogin
- GoToVerbale
- LoginAction
- LogoutAction
- SetExamSessionResult

Model objects (Beans)

- Course
- ExamSession
- Student
- Teacher

View

- Esito
- ExamSessionResult
- HomeStudent
- HomeTeacher
- Iscritti
- Login
- Verbale

Data Access Objects (Classes)