

# **Relazione Progetto**

## **Gestione Orari**

Nome: TETE SERGE

Cognome: BENISSANH DATE

Maricola : 764972

## **1 - Introduzione**

Il progetto "gestione orari" consiste nell'analisi e lo sviluppo di un sistema web distribuito per la gestione dei corsi universitari, utenti( studenti, docenti, segretari), orari e risorse dell'università.

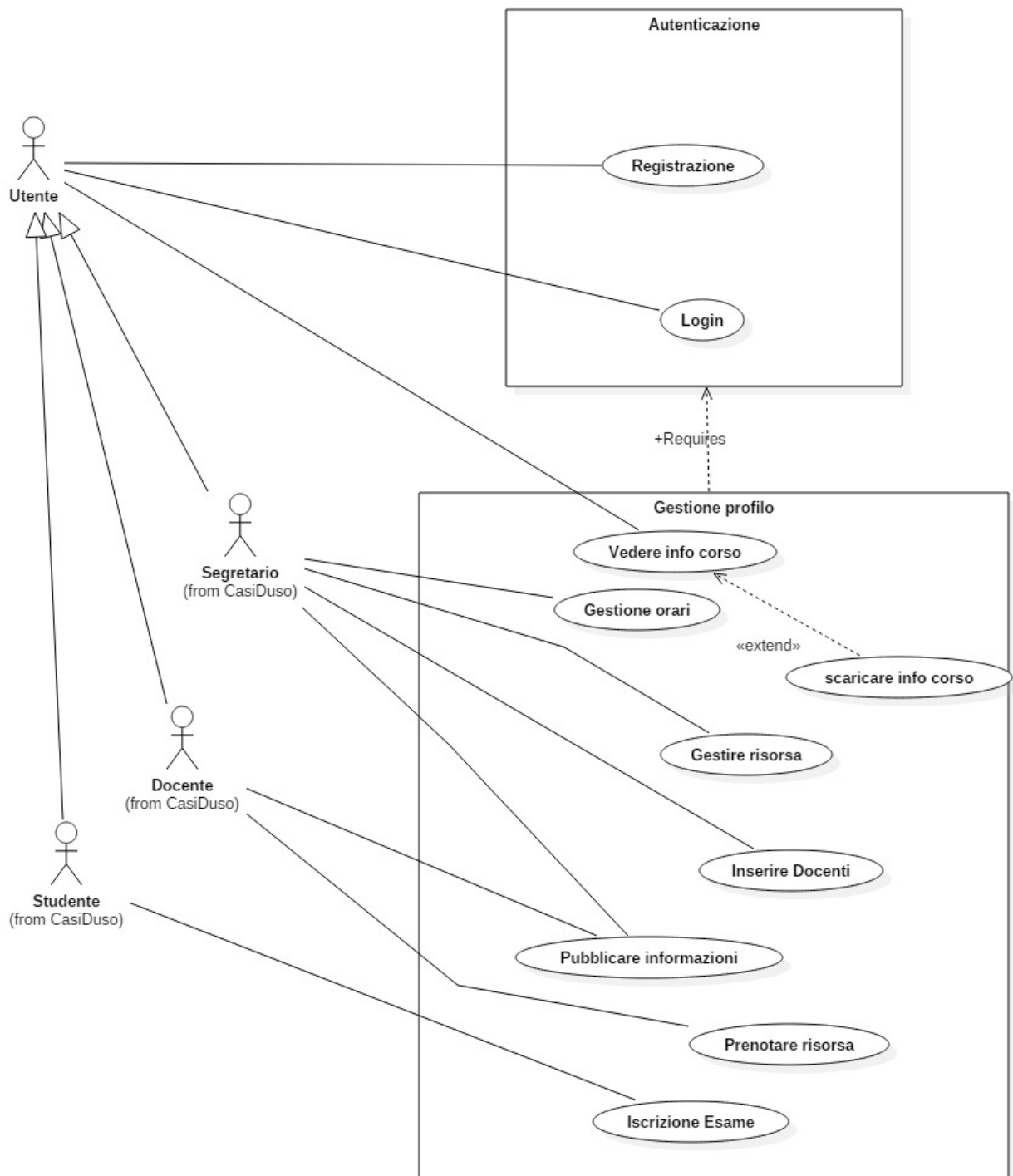
Assumiamo in questo caso che:

- il sistema dovrà essere accessibile esclusivamente da studenti, segretari o docenti.
- Gli studente scelgono i corsi a cui vogliono iscriversi durante la registrazione.
- Ogni professore puoi insegnare più corsi quindi scelgono i loro corsi durante la registrazione.
- Gli studenti e i docenti vedranno quindi nell'area personale solo le informazioni legate ai loro corsi

## **2 – Analisi e Progettazione**

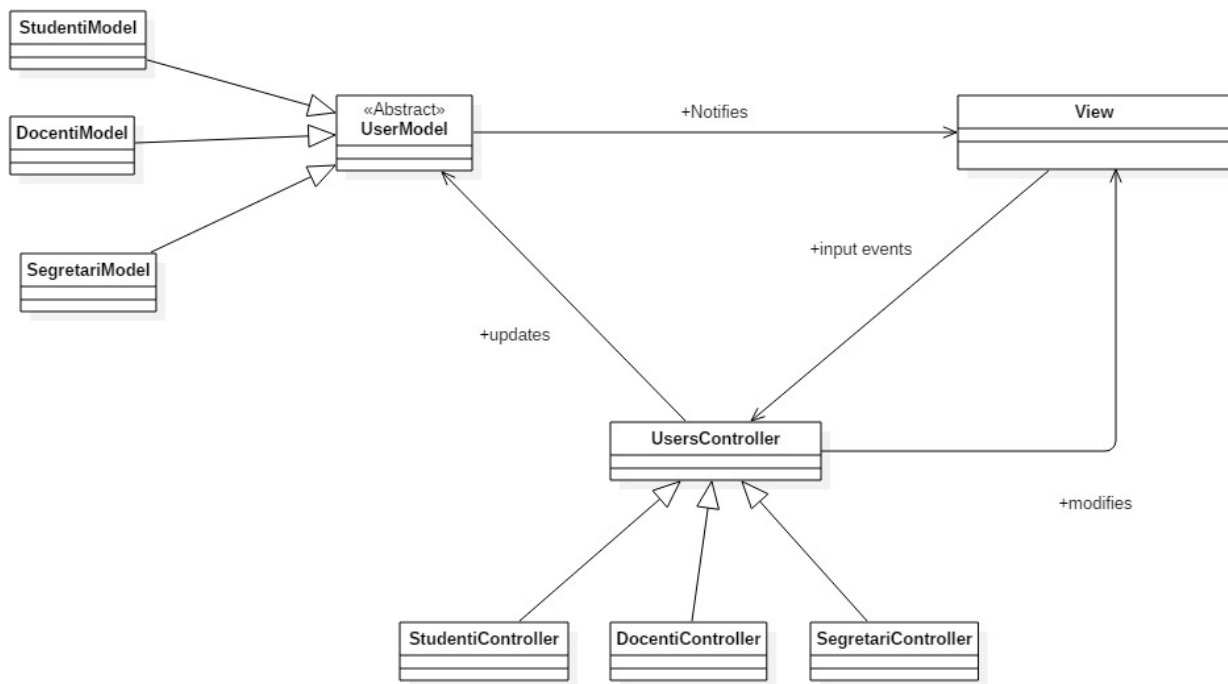
### **Diagramma dei casi d'uso**

- Per accedere al sistema i nuovi utenti devono registrarsi. Normalmente vale solo per gli studenti visto che in un sistema del genere non si può autorizzare qualsiasi persona ad iscriversi come docente o segretario però in questo caso per motivi di test sarà possibile.
- Gli studenti possono vedere il calendario dei corsi e scaricarlo. Possono vedere gli annunci pubblicati da docenti e segretari e iscriversi agli esami.
- I docenti e i segretari possono pubblicare annunci e i docenti possono pubblicare avvisi riguardanti i singoli corsi.
- I docenti possono prenotare le risorse disponibili.
- I segretari possono gestire/inserire nuove risorse.
- i segretari possono inserire nuovi professori assegnando a loro i corsi.



## Diagramma delle classi

Il seguente diagramma basato sul pattern MVC illustra la struttura e le relazioni tra le varie classi.



Il Controller si occupa di controllare ed eseguire tutte le operazioni richieste dal client usando o meno il Model corrispondente. Se ci sono modifiche la View viene aggiornata.

Nel diagramma completo ci sono gli attributi e i metodi delle varie classi del sistema.

## Diagramma delle classi (completo)

Ora vediamo tutti i dettagli sulla struttura delle classi.

### UsersController

Si occupa generalmente di gestire tutte le richieste in comune a tutti gli utenti, individuare il tipo di utente e i permessi di accesso.

### StudentiController, DocentiController, SegretariController

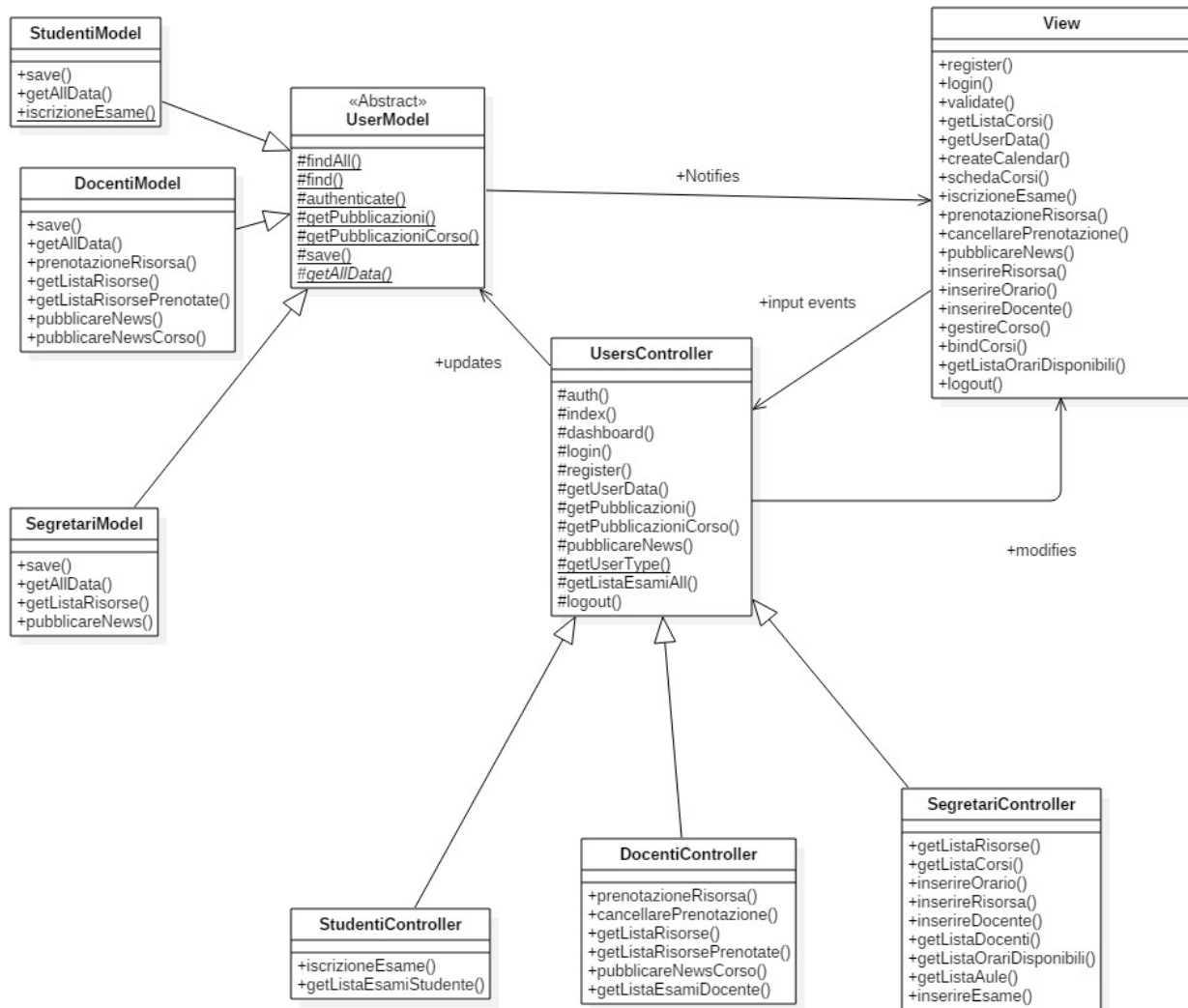
Estendono la UsersController e definiscono le particolarità e permessi propri al tipo di utente corrispondente.

### UserModel

Classe astratta usata dai vari controllers per la gestione, il recupero e gli updates dei dati del database.

## StudenteModel, DocenteModel, SegretarioModel

Gestiscono i dati e ridefiniscono i metodi in base al tipo di utente corrispondente alle chiamate.



### 3 – Ambiente di sviluppo e strumenti usati

Per avere un ambiente di sviluppo unico sia per l'UI che per il backend ho scelto di usare il Typescript per il backend, Vue.js per il frontend e Node.js per il server così da avere tutto l'ambiente in Javascript e rendere il progetto più semplice da configurare, modificare e testare.

**Node.js** è un runtime Javascript costruito sul motore Javascript V8 di Chrome che usa un modello I/O non bloccante ad eventi che lo rende un framework leggero ed efficiente con un grande ecosistema di librerie open source.

**TypeScript** è un linguaggio di programmazione(Super-set del Javascript) libero e open-source sviluppato da Microsoft che si basa su ECMAScript 6 .

**Vue.js** è un framework Javascript scalabile open-source per realizzare interfacce grafiche.

Per la gestione dei dati useremo un database leggero di tipo **SQLite** in quanto non richiede particolari installazioni e configurazioni per essere operativo ed è facilmente portatile.

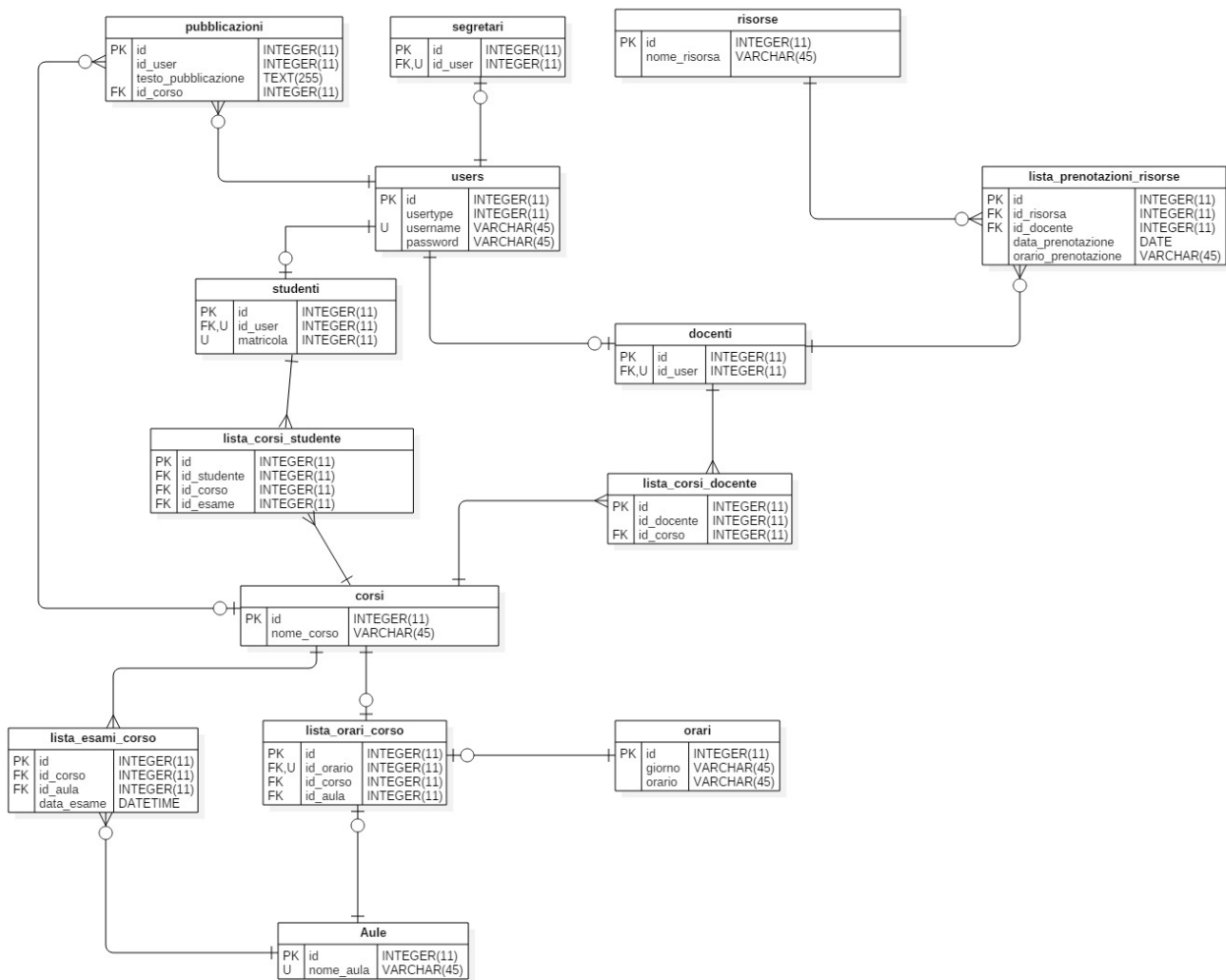
Tutti gli strumenti utilizzati sono installati come dipendenze del progetto tramite il gestionario di packages NPM di node. C'è un file package.json nella root del progetto che contiene le informazioni sulle dipendenze.

```
"dependencies": {  
  "body-parser": "^1.18.2",  
  "ejs": "^2.5.7",  
  "express": "^4.16.2",  
  "express-session": "^1.15.6",  
  "knex": "^0.14.3",  
  "nodemon": "^1.14.12",  
  "typescript": "^2.7.2"  
},  
"devDependencies": {  
  "@types/express": "^4.11.1",  
  "@types/express-session": "^1.15.8",  
  "@types/knex": "^0.14.7",  
  "@types/node": "^9.4.6",  
  "@types/sqlite3": "^3.1.1",  
  "npm": "^5.6.0"  
}
```

## 4 – Implementazione Soluzione

### Data Model ( Diagramma ER )

I seguente diagramma mostra le relazioni tra le diverse tabelle del database.



Alcune tabelle del database hanno dei dati predefinite che non dovranno essere inserite a mano per poter facilitare i test. Sono:

- La lista delle Aule.
- La lista degli orari ( sotto forma di intervalli da 2 ore).
- La lista dei corsi disponibili.

## Accesso al sistema

Il sistema chiede di autenticarsi o di registrarsi prima di poter accedere all'area personale.  
Le seguenti immagini mostrano i diversi stati possibili durante il l'accesso.

### Login

**Login**

Username

Password

Login

**Login**

Username

Password

Username o password sbagliata!

Login

### Registrazione

**Registrazione**

☒ Studente ☐ Docente ☐ Segretario

Username

Password

Registrati

☐ Analisi Matematica ☐ Fisica  
☐ Programmazione 1 ☐ Inglese  
☐ Programmazione 2 ☐ Algebra Lineare  
☐ Ingegneria del software ☐ Sicurezza ed affidabilità

**Registrazione**

☒ Studente ☐ Docente ☐ Segretario

Username

Password

Dati non validi!

Registrati

☐ Analisi Matematica ☐ Fisica  
☐ Programmazione 1 ☐ Inglese  
☐ Programmazione 2 ☐ Algebra Lineare  
☐ Ingegneria del software ☐ Sicurezza ed affidabilità

**Registrazione**

☒ Studente ☐ Docente ☐ Segretario

studente1

...

Utente già presente!

Registrati

☐ Analisi Matematica ☐ Fisica  
☐ Programmazione 1 ☐ Inglese  
☐ Programmazione 2 ☐ Algebra Lineare  
☐ Ingegneria del software ☐ Sicurezza ed affidabilità

**Registrazione**

☒ Studente ☐ Docente ☐ Segretario

student2

...

Registrazione effettuata!

Registrati

☐ Analisi Matematica ☐ Fisica  
☐ Programmazione 1 ☐ Inglese  
☐ Programmazione 2 ☐ Algebra Lineare  
☐ Ingegneria del software ☐ Sicurezza ed affidabilità

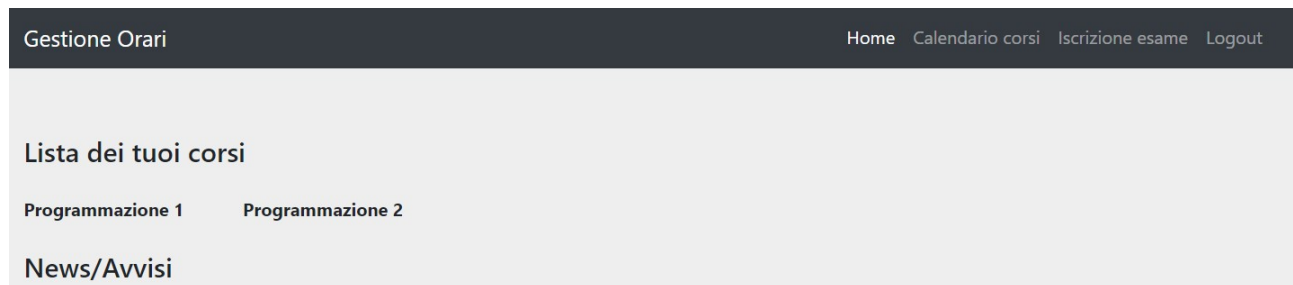


## Area personale

Una volta la registrazione e login effettuati si entra nell'area personale dell'utente che ha un aspetto diverso in base al tipo di utente.

Gli studenti hanno accesso a :

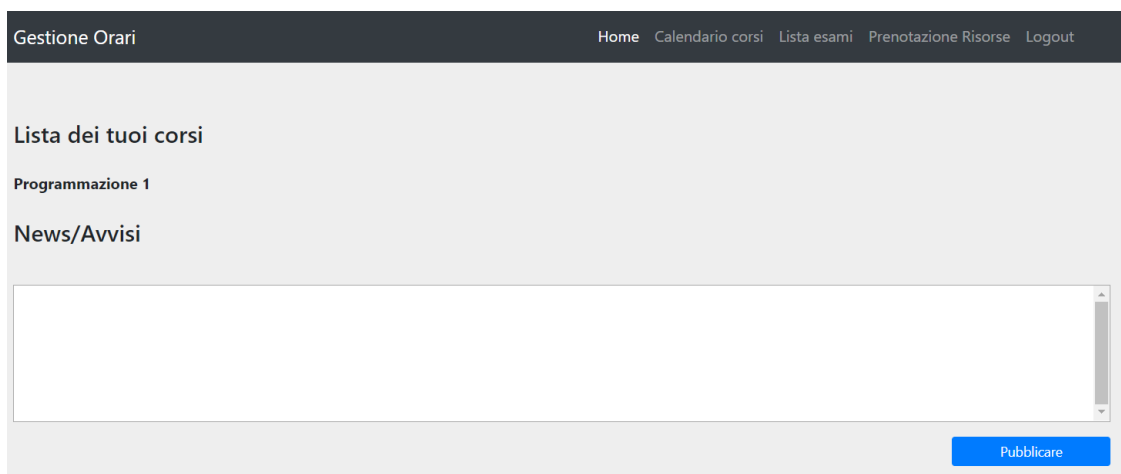
- la home page che contiene la lista dei loro corsi e tutti gli annunci pubblicati.(il click su un corso particolare porta su una pagina dove ci sono soltanto i news di quel corso)



- il calendario dei corsi (legato agli orari dei corsi scelti durante l'iscrizione).
- la pagina di iscrizione agli esami dove possono iscriversi oppure disiscriversi agli esami.

i docenti invece hanno accesso a :

- la home page che contiene la lista dei loro corsi, gli annunci pubblicati e la possibilità di pubblicare nuovi annunci.(il click su un corso particolare porta su una pagina dove ci sono soltanto i news di quel corso e possono pubblicare annunci legati esclusivamente a quel corso).
- il calendario dei corsi (legato agli orari dei corsi scelti durante l'iscrizione o durante l'inserimento dalla segreteria).
- La pagina per la prenotazione delle risorse.
- La pagina della lista degli esami legati ai corsi del docente.



i segretari hanno accesso a :

- la home page che contiene soltanto gli annunci pubblicati e la possibilità di pubblicare nuovi annunci.
- La gestione delle risorse ( principalmente l'inserimento ).
- La gestione dei docenti ( inserimento docenti con corsi ).
- La gestione degli orari ( per ogni corso presente nel database ).
- La pagina della lista di tutti gli esami (possono inserire nuovi esami)

The screenshot shows the 'Gestione Orari' interface. At the top is a dark navigation bar with links: Home, Lista esami, Gestione Risorse, Gestione Docenti, Gestione Orari, and Logout. Below the navigation bar, the main content area has a header 'News/Avvisi'. Under this header is a large, empty rectangular text box with a vertical scrollbar on the right. At the bottom right of this text box is a blue button labeled 'Pubblica'.

## Studio dei casi d'uso più interessanti

### Diagramma degli stati ( Caso d'uso Gestione orari )

La gestione degli orari si effettua in questo modo :

- il segretario sceglie il corso da gestire nella lista dei corsi disponibili.
- Appare una form per la selezione dell'aula e dell'orario
- La lista degli orari disponibili viene generata **soltanto dopo la scelta dell'aula** così da gestire la consistenza dei dati e evitare che la medesima aula venga accidentalmente occupata contemporaneamente da due corsi.
- Dopo la conferma viene inserito l'orario se tutti i dati sono validi(selezionati) oppure appare un messaggio di errore.

The screenshot shows the 'Gestione Orari' interface. At the top is a dark navigation bar with links: Home, Lista esami, Gestione Risorse, Gestione Docenti, Gestione Orari, and Logout. Below the navigation bar, the main content area has a header 'Scegliere il corso da gestire'. Under this header is a grid of course names: Analisi Matematica, Fisica, Programmazione 1, Inglese, Programmazione 2, and Algebra Lineare. Below the grid, there is a section titled 'Aggiungere un orario per il corso di Programmazione 1'. This section contains two dropdown menus: 'Aula:' with a 'Selezionare' button and 'Orario:' with a 'Selezionare' button. To the right of these dropdowns is a blue button labeled 'Aggiungere orario'.

Gestione Orari

HomeLista esamiGestione RisorseGestione DocentiGestione OrariLogout

Scegliere il corso da gestire

Analisi Matematica  
Ingegneria del  
software

Fisica  
Sicurezza ed  
affidabilità

Programmazione 1

Inglese

Programmazione 2

Algebra Lineare

Aggiungere un orario per il corso di Programmazione 1

Aula: 

Selezionare

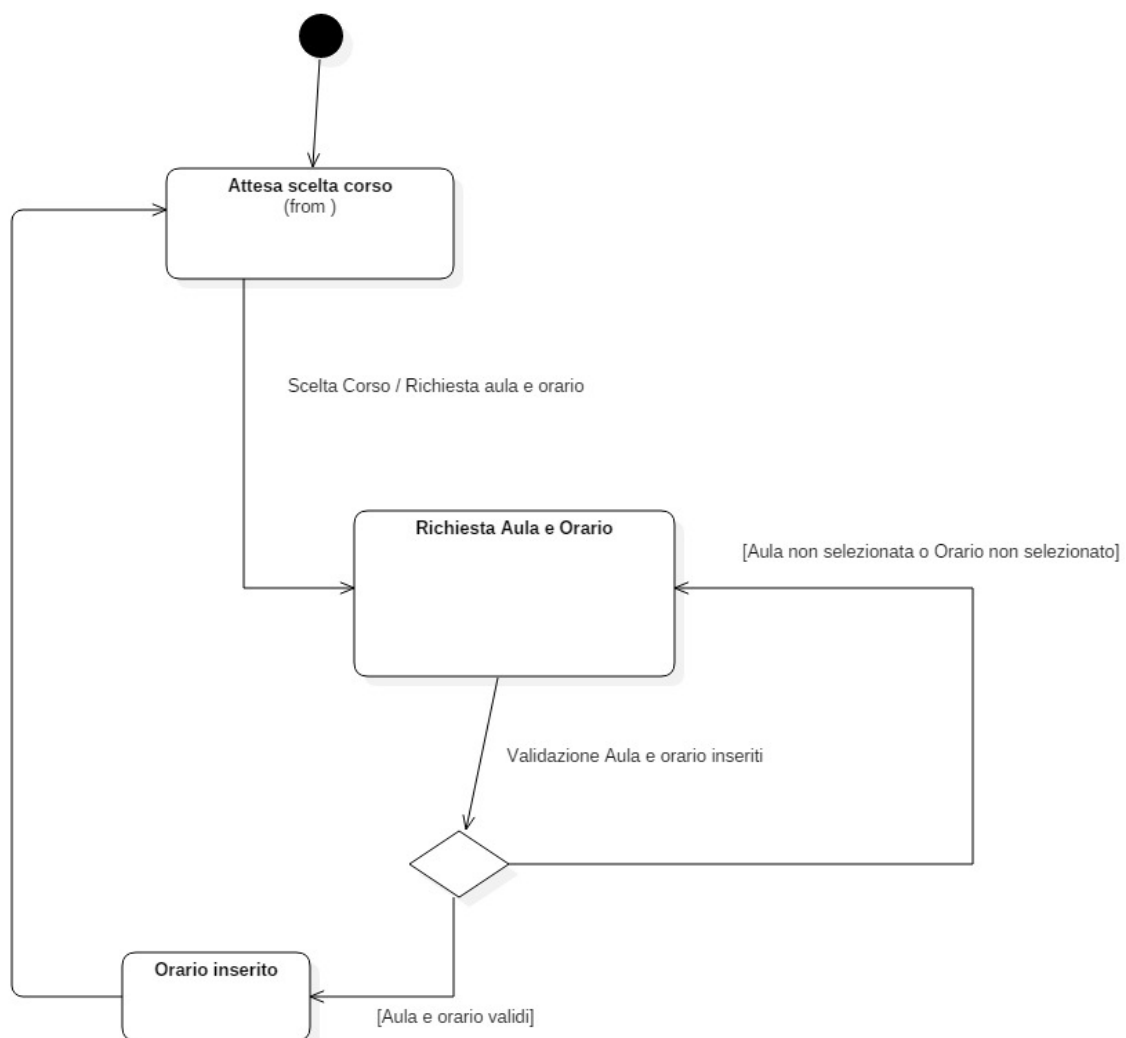
Orario: 

Selezionare

Aggiungere orario

Selezionare Dati validi!

Il seguente diagramma degli stati illustra il comportamento del sistema per l'inserimento degli orari.



Gestione Orari

Home

Calendario corsi

Prenotazione Risorse

Logout

Lista Risorse

video proiettore 1

Data:

Orario:

Prenotare

Scegliere una data posteriore a quella di oggi e un orario!

Una risorsa non può essere prenotata due volte per la stessa data e lo stesso orario.

**Gestione Orari** Home Calendario corsi Prenotazione Risorse Logout

### Lista Risorse

video proiettore 1 Data: 08/03/2018 Orario: 8:30 - 10:30 Prenotare

Risorsa già prenotata per quella data!

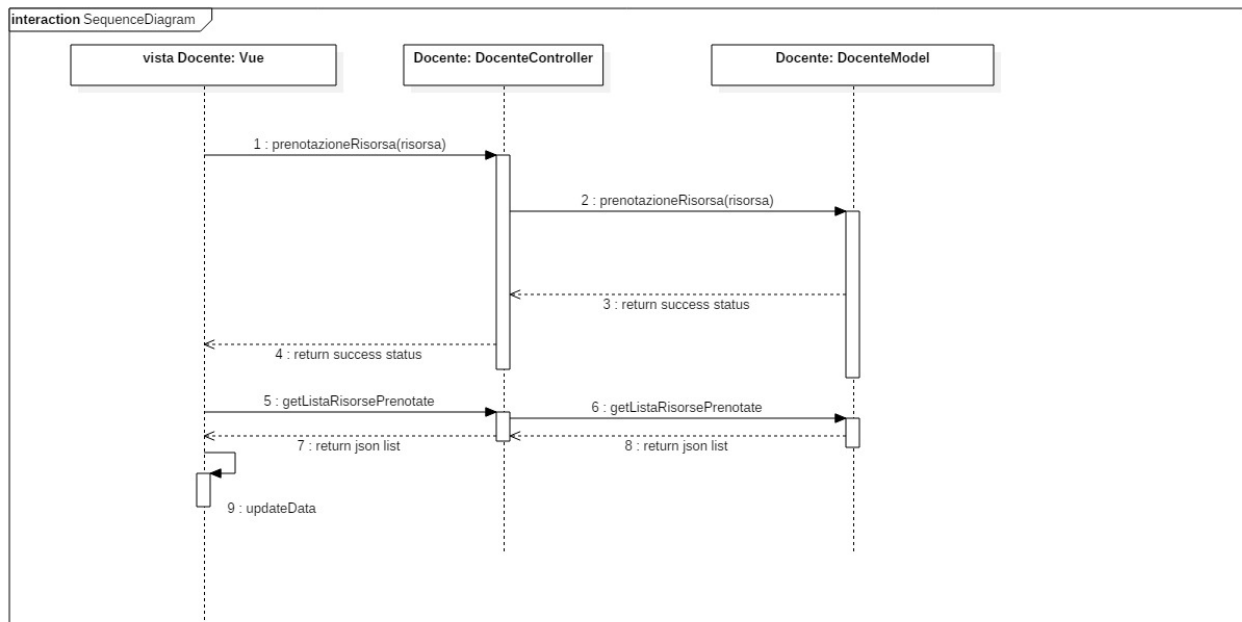
### Lista risorse prenotate

video proiettore 1	2018-03-08	8:30 - 10:30	Cancellare prenotazione
video proiettore 1	2018-03-07	8:30 - 10:30	Cancellare prenotazione
video proiettore 1	2018-03-01	8:30 - 10:30	Cancellare prenotazione

C'è anche la possibilità di cancellare le prenotazioni.

Il diagramma di sequenza qui sotto mostra i vari steps durante la prenotazione di una risorsa da un docente.

- La View fa una chiamata AJAX
- Il Controller chiama il metodo corrispondente che usa il Model per inserire i dati nel database.
- Il risultato della query viene restituito al Controller poi alla View.
- La View fa un'altra chiamata per aggiornarsi dopo l'operazione.



## Conclusione

La gestione degli orari è un sistema molto ampio ed è molto più complesso della soluzione proposta. In realtà l'idea è di avere una base (che potrà essere estesa abbastanza facilmente) con tutte le funzionalità indispensabili per fornire un sistema che simula gli aspetti più importanti del sistema completo.

Alcune funzionalità interessanti che mancano in questa soluzione:

- l'inserimento delle Aule e dei Corsi dai segretari.
- la possibilità di aggiungere anche l'orario per completare la data nella lista degli esami