

## Gestione orari

Il progetto originario doveva sviluppare un sistema distribuito per la gestione dei corsi universitari, degli orari, e delle aule con tre classi principali di utenti: segretari, docenti e studenti.

I segretari avrebbero avuto sostanzialmente la funzione di admin dove avrebbero potuto gestire tutto a piacimento, pur non usufruendo attivamente del sistema stesso. I docenti avrebbero potuto invece sia usufruire del sistema per visualizzare gli orari sia prenotare le varie risorse. Infine gli studenti avrebbero potuto solamente consultare il sistema per visualizzare avvisi e orari senza la possibilità di modificare nulla.

Ovviamente il sistema doveva controllare che non ci fossero inconsistenze negli orari o nelle risorse, mantenere le informazioni di base sui corsi attivi e pregressi, i programmi e i docenti di ogni corso, mantenere il calendario degli esami per tutti e corsi (con relative iscrizioni) e doveva essere accessibile da qualsiasi Pc connesso a Internet.

Purtroppo a causa dell'abbandono dei colleghi con cui stavo realizzando il progetto e delle mie poche conoscenze in Java Front-End Back-End il risultato finale non rispecchia assolutamente le specifiche richieste e non ritengo sia minimamente soddisfacente

## Analisi dei requisiti

Basandosi sulle specifiche richieste sono stati inizialmente individuati nove casi d'uso, tuttavia nel progetto risulta essere sviluppato solo l'UC003.

## UC001 Inserimento risorsa.

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Segretario.

**Parti interessate e interessi:** Segretario, vuole inserire una nuova risorsa nell'elenco delle risorse disponibili a sistema.

**Precondizioni:** L'utente deve aver fatto log in come Segretario, la risorsa da inserire non deve essere già presente nel sistema.

**Garanzia di successo:** La risorsa risulta essere elencata tra quelle disponibili a sistema

**Scenario principale di successo:**

Il segretario seleziona la creazione di una nuova risorsa.

Inserisce il nome della risorsa.

Seleziona l'opzione per salvare la nuova risorsa.

Il sistema salva la risorsa nell'elenco di quelle disponibili.

Il sistema attende l'inserimento di un'altra risorsa.

**Estensioni:**

3a. Il segretario seleziona la modifica della disponibilità. Vedi UC002.

5a. Il segretario seleziona l'opzione per tornare al menù principale.

**Requisiti speciali:**

**Elenco delle varianti tecnologiche e dei dati:** nessuna

**Frequenza di ripetizione:** bassa

## **UC002 Modifica disponibilità.**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Segretario/Docente.

**Parti interessate e interessi:** Segretario/Docente, vogliono modificare la disponibilità di una risorsa.

**Precondizioni:** L'utente deve aver fatto log in come Segretario o come Docente, la risorsa da modificare deve essere già presente nel sistema.

**Garanzia di successo:** La disponibilità delle risorsa viene modificata in accordo con la scelta dell'utente, se possibile.

**Scenario principale di successo:**

L'utente seleziona l'opzione per la modifica della disponibilità di una data risorsa

Inserisce la disponibilità desiderata per uno specifico giorno a una specifica ora.

Il sistema rileva che la risorsa è libera e gli assegna quindi la nuova disponibilità.

Il sistema salva la risorsa modificata nell'elenco delle risorse disponibili.

Riparte al punto 2.

**Estensioni:**

\*Un Docente può modificare la disponibilità di una risorsa solo se risulta libera e non si tratta di un altro Docente.

2a) L'utente seleziona l'opzione per tornare al menù principale.

3a) Il sistema rileva che la risorsa è occupata

Il sistema chiede conferma dell'operazione di sovrascrittura.

Il segretario conferma.

2.1) Il segretario non conferma

2.2) Il sistema riparte al punto 2.

Il sistema sovrascrive la disponibilità e riparte dal punto 4.

**Requisiti speciali:**

**Elenco delle varianti tecnologiche e dei dati:** nessuno

**Frequenza di ripetizione:** elevata

## **UC003 Creazione Orario Didattico**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Segretario

**Parti interessate e interessi:** Segretario, vuole creare l'Orario Didattico di uno specifico insegnamento

**Precondizioni:** L'utente deve aver effettuato il log in come segretario.

**Garanzia di successo:** Il sistema ha creato l'orario didattico per uno specifico insegnamento

**Scenario principale di successo:**

Il Segretario seleziona l'opzione di creazione di un nuovo orario didattico.

Il Segretario seleziona uno specifico Corso di Studi.

Il Segretario seleziona uno specifico Insegnamento.

Il Segretario specifica i giorni della settimana, gli orari e le aule in cui si svolgerà l'insegnamento

Salva l'orario didattico.

Il sistema rileva la possibilità di creare l'orario didattico con le specifiche impostate (ovvero che non ci siano sovrapposizioni di lezioni o che una risorsa si trovi in due posti contemporaneamente).

Il sistema salva l'orario didattico creato e torna al punto 3.

**Estensioni:**

4a) Il Segretario seleziona l'opzione per tornare al menù principale.

5a) Il sistema rileva l'impossibilità di creare l'orario didattico con le specifiche impostate (ovvero vi sono sovrapposizioni di lezioni o una risorsa si trova in due posti contemporaneamente).

Il sistema segnala all'utente quali specifiche non sono applicabili.

Il sistema riparte dal punto 3.

7a) Il Segretario seleziona l'opzione per tornare al menù principale.

**Requisiti speciali:****Elenco delle varianti tecnologiche e dei dati:** nessuna**Frequenza di ripetizione:** media**UC004 Creazione Calendario Didattico****Portata:** Software per la Gestione Orari**Livello:** Obiettivo utente.**Attore Primario:** Segretario**Parti interessate e interessi:** Segretario, vuole creare il Calendario Didattico Universitario.**Precondizioni:** L'utente deve aver effettuato il log in come Segretario.**Garanzia di successo:** Il sistema ha creato il Calendario Didattico di uno specifico corso di studi**Scenario principale di successo:**

Il Segretario seleziona l'opzione di creazione di un nuovo Calendario Didattico.

Il Segretario seleziona le settimane (e i giorni) disponibili per l'anno in corso e quelli in cui l'università è chiusa.

Salva il Calendario.

Il sistema rileva la possibilità di salvare il calendario con le specifiche impostate (ovvero che l'Orario Didattico non si sovrapponga a uno precedentemente salvato nel Calendario).

Il sistema salva il calendario creato e torna al menù principale.

**Estensioni:**

4a) Il Segretario seleziona l'opzione per tornare al menù principale.

5a) Il sistema rileva l'impossibilità di salvare il calendario con le specifiche impostate.

Il sistema segnala all'utente quali specifiche non sono applicabili.

Il sistema riparte dal punto 4.

**Requisiti speciali:****Elenco delle varianti tecnologiche e dei dati:** nessuna**Frequenza di ripetizione:** media**UC005 Pubblicazione News/Avviso****Portata:** Software per la Gestione Orari**Livello:** Obiettivo utente.**Attore Primario:** Segretario, Docente.**Parti interessate e interessi:**

Segretario/Docente: desiderano pubblicare un messaggio in modo che sia consultabile da tutti gli utenti del sistema

**Precondizioni:** L'utente deve aver effettuato il log-in come Segretario o Docente.**Garanzia di successo:** L'informazione pubblicata deve essere accessibile e consultabile da tutti gli utenti.**Scenario principale di successo:**

L'utente seleziona la creazione di un nuovo messaggio da condividere.

Il sistema visualizza un editor di testo in cui l'utente può scrivere liberamente.

L'utente scrive il messaggio che intende condividere.

Quando soddisfatto del testo seleziona l'opzione per pubblicare il messaggio.

Il sistema salva il messaggio e lo rende pubblicamente consultabile nell'apposita sezione

**Estensioni:**

4a) L'utente seleziona l'opzione per annullare l'operazione

Il sistema torna al menù principale

**Requisiti speciali:****Elenco delle varianti tecnologiche e dei dati:** nessuna**Frequenza di ripetizione:** elevata

## **UC006 Visualizza orario**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Segretario, Docente, Studente.

**Parti interessate e interessi:**

Segretario/Docente/Studente: desiderano visualizzare l'orario didattico di uno specifico anno di un corso di studi o di uno specifico insegnamento

**Precondizioni:** L'utente deve aver effettuato il log-in.

**Garanzia di successo:** L'orario di tutti i corsi di studio e di tutti gli insegnamenti deve essere accessibile e consultabile da tutti gli utenti.

**Scenario principale di successo:**

L'utente seleziona l'opzione per visualizzare gli orari didattici.

L'utente seleziona l'opzione per visualizzare l'orario di uno specifico corso di studi.

Il sistema mostra a schermo l'orario selezionato.

L'utente seleziona l'opzione per tornare al menù principale.

Il sistema torna al menù principale.

**Estensioni:**

2a) L'utente seleziona l'opzione per visualizzare l'orario di uno specifico insegnamento.

Il sistema riparte dal punto 3.

4a) L'utente seleziona l'opzione per stampare l'orario visualizzato

Il sistema crea un file binario in cui inserisce le informazioni dell'orario

**Requisiti speciali:**

**Elenco delle varianti tecnologiche e dei dati:** nessuna

**Frequenza di ripetizione:** media

## **UC007 Creazione esame**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Docente.

**Parti interessate e interessi:**

Docente, desidera creare uno specifico esame.

**Precondizioni:** L'utente deve aver effettuato il log-in come Docente.

**Garanzia di successo:** L'esame risulta nell'elenco degli esami sostenibili e non è programmato nella stessa aula, stesso giorno e stessa ora di un altro esame o di una lezione.

**Scenario principale di successo:**

Il Docente seleziona l'opzione per creare l'esame.

Il Docente specifica Data, Ora e Aula dove verrà sostenuto l'esame.

Il sistema rileva che è possibile creare un nuovo esame con le condizioni specificate.

Il sistema modifica la disponibilità dell'aula per la data e l'ora specificati.

Il sistema aggiunge l'esame all'elenco degli esami sostenibili

Il sistema segnala l'avvenuta creazione con successo.

Il sistema torna al menù principale.

**Estensioni:**

3a) Il sistema rileva che non è possibile creare un nuovo esame con le condizioni specificate.

Il sistema segnala il problema all'utente invitandolo a cambiare i parametri.

Il sistema riparte al punto 2

**Requisiti speciali:**

**Elenco delle varianti tecnologiche e dei dati:** nessuna

**Frequenza di ripetizione:** media

## **UC008 Iscrizione esame**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Studente.

**Parti interessate e interessi:**

Studente, desidera iscriversi a uno specifico esame in modo da poterlo sostenere.

**Precondizioni:** L'utente deve aver effettuato il log-in come Studente, l'esame in oggetto deve esistere.

**Garanzia di successo:** Lo studente risulta nell'elenco degli iscritti all'esame

**Scenario principale di successo:**

Lo studente seleziona l'opzione per iscriversi all'esame.

Il sistema chiede conferma dell'operazione.

Lo studente conferma l'operazione.

Il sistema aggiunge lo studente all'elenco degli studenti che devono sostenere l'esame.

Il sistema segnala l'avvenuta iscrizione con successo.

Il sistema torna al menù principale.

**Estensioni:**

3a) Lo studente annulla l'operazione

Il sistema passa al punto 6.

**Requisiti speciali:**

**Elenco delle varianti tecnologiche e dei dati:** nessuna

**Frequenza di ripetizione:** media

## **UC009 Visualizza News/Avviso**

**Portata:** Software per la Gestione Orari

**Livello:** Obiettivo utente.

**Attore Primario:** Segretario, Docente, Studente.

**Parti interessate e interessi:**

Segretario/Docente/Studente: desiderano visualizzare una news o un avviso.

**Precondizioni:** L'utente deve aver effettuato il log-in.

**Garanzia di successo:** Gli utenti riescono a visualizzare le news e gli avvisi.

**Scenario principale di successo:**

L'utente seleziona l'opzione per visualizzare l'elenco delle news/avvisi.

Il sistema mostra a schermo l'elenco delle news/avvisi.

L'utente seleziona la news/avviso desiderati.

Il sistema mostra a schermo il testo completo della selezione.

L'utente seleziona l'opzione per tornare all'elenco delle news/avvisi

Il sistema torna al punto 2.

**Estensioni:**

5a) L'utente seleziona l'opzione per tornare al menù principale.

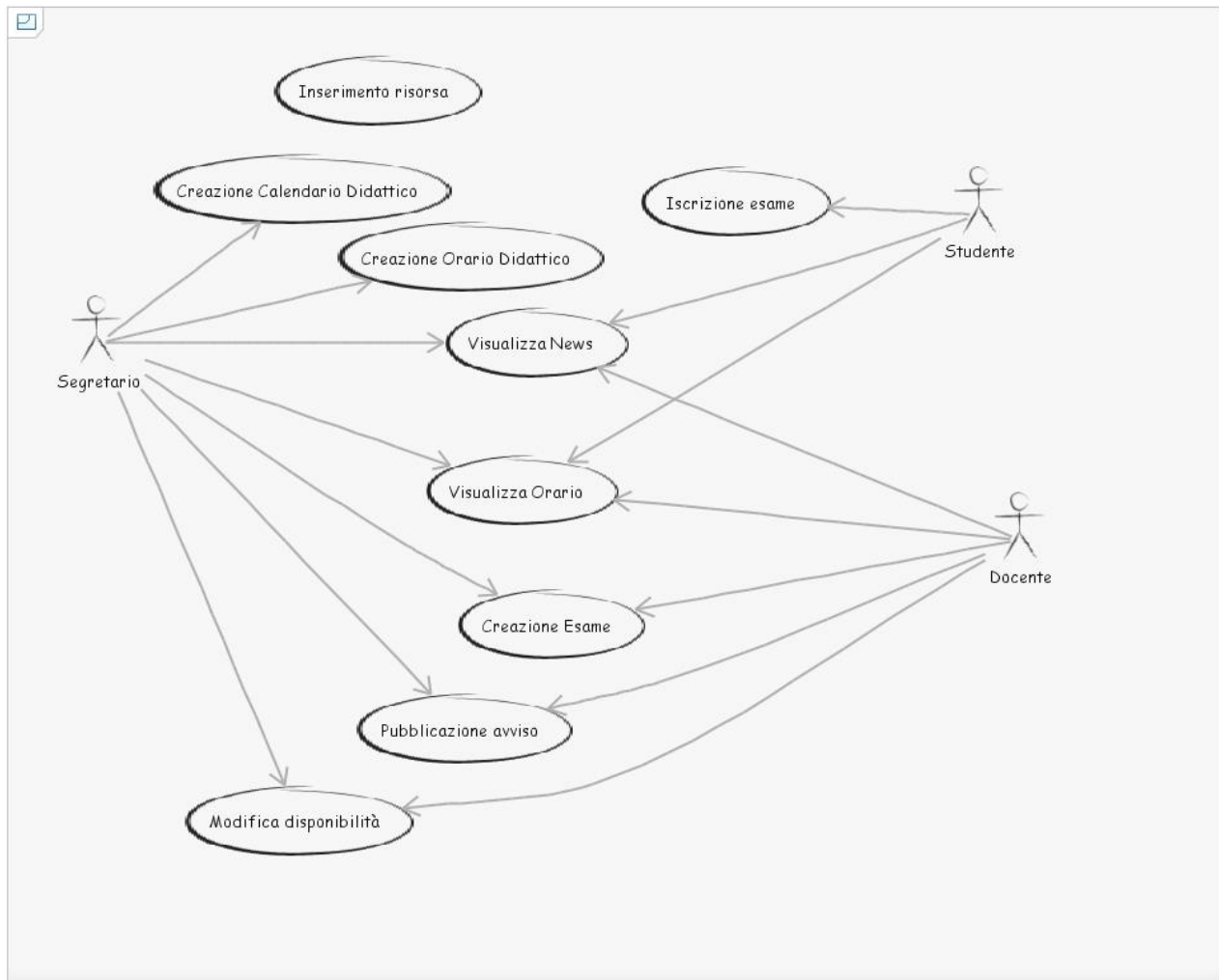
Il sistema torna al menù principale.

**Requisiti speciali:**

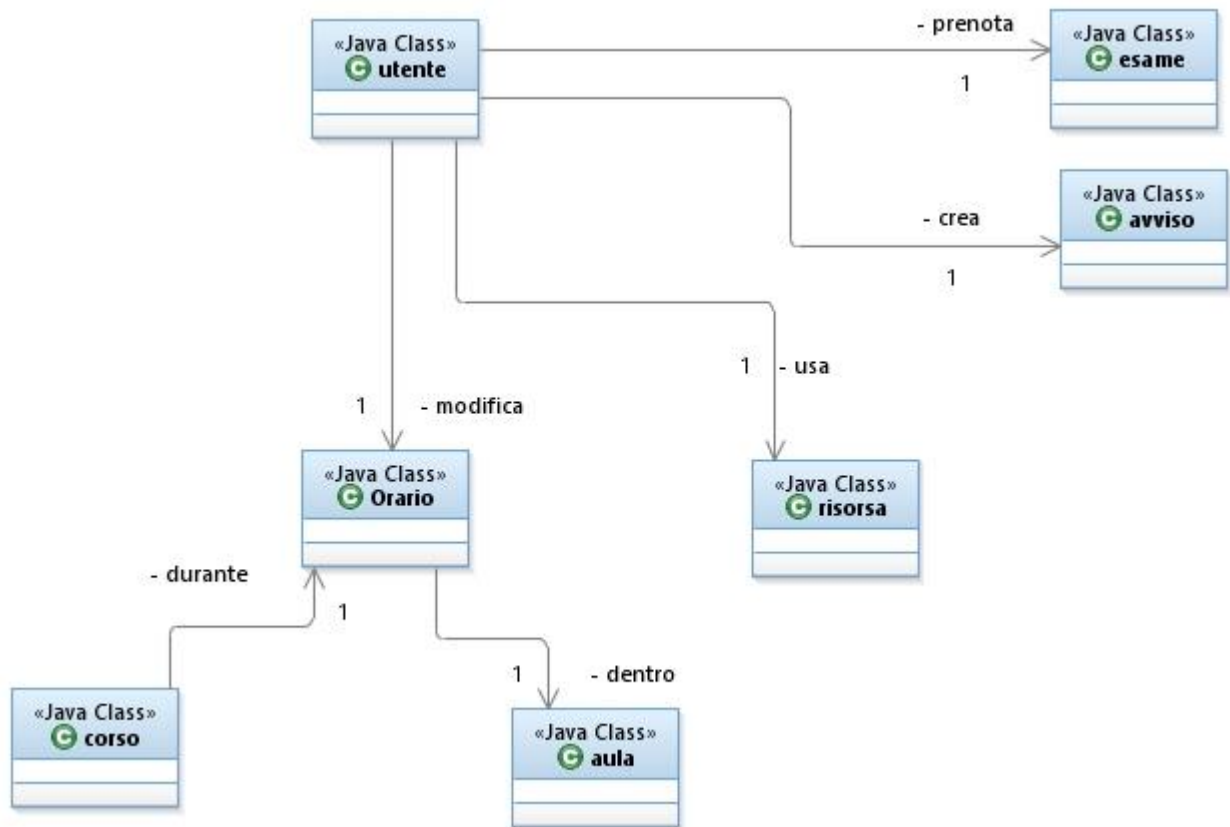
**Elenco delle varianti tecnologiche e dei dati:** nessuna

**Frequenza di ripetizione:** media

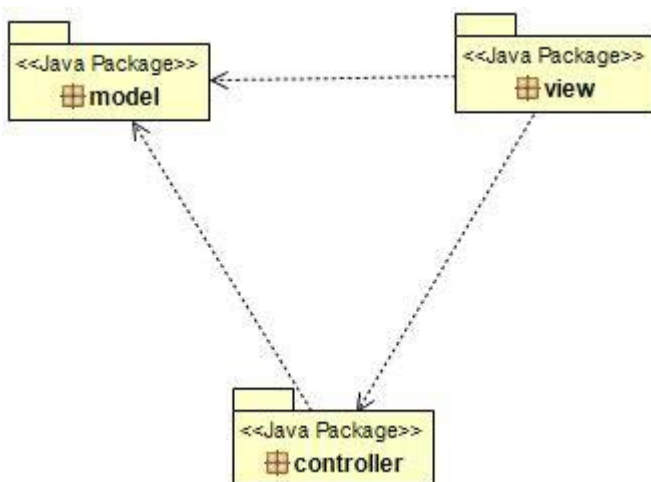
Diagramma dei casi d'uso



## Diagramma di dominio



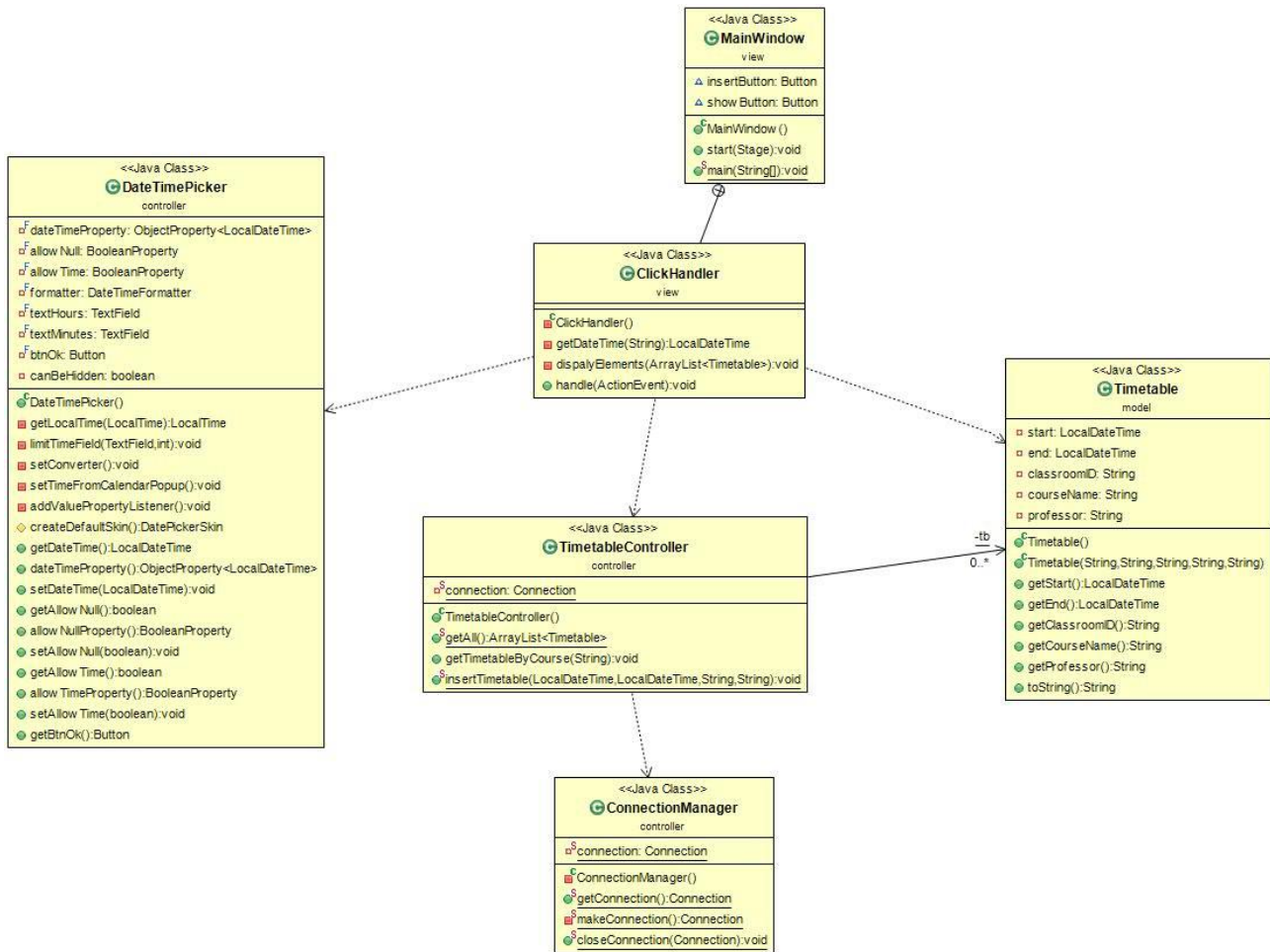
## Diagramma dei Package



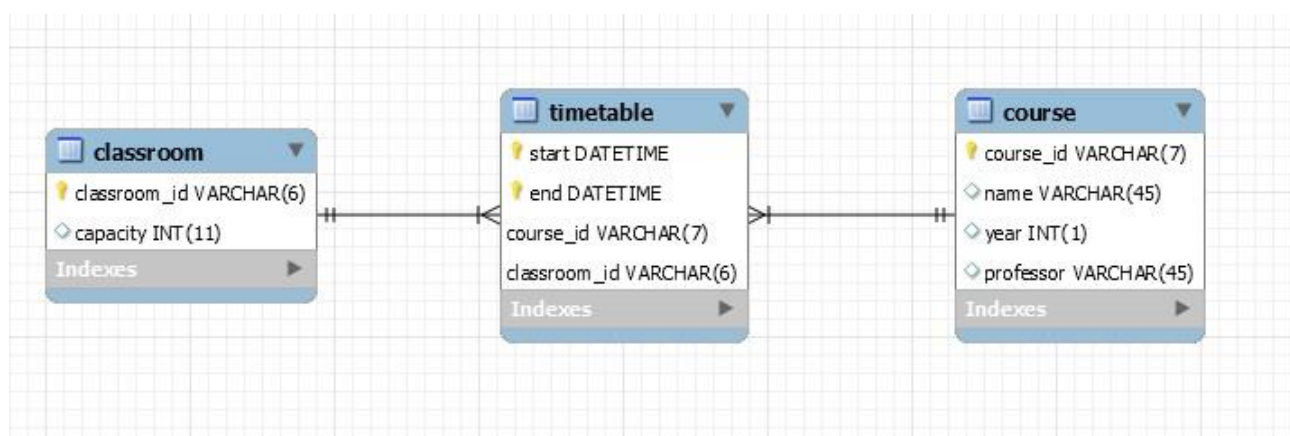
## Diagramma delle classi

Si è scelto di sviluppare il progetto senza l'utilizzo di alcun framework di persistenza per il database, ma utilizzando la *Java DataBase Connectivity*. L'unica parte effettivamente sviluppata, rispetto a tutti i casi d'uso, è quella riguardante l'inserimento dell'orario delle lezioni, tuttavia il metodo per l'inserimento non funziona (vedi sezione "Istruzioni programma").

Di seguito il diagramma delle classi.



## Struttura del Database



Course\_ID e Classroom\_ID in Timetable sono chiavi esterne che fanno riferimento alle tabelle delle classi (Classroom) e dei corsi (Courses), non si possono quindi inserire orari di lezioni di corsi e in aule non già presenti nel database. Ovviamente nel progetto originale sarebbero dovuti esistere dei metodi per permettere ai Segretari di poter aggiungere e modificare elementi di queste tabelle.

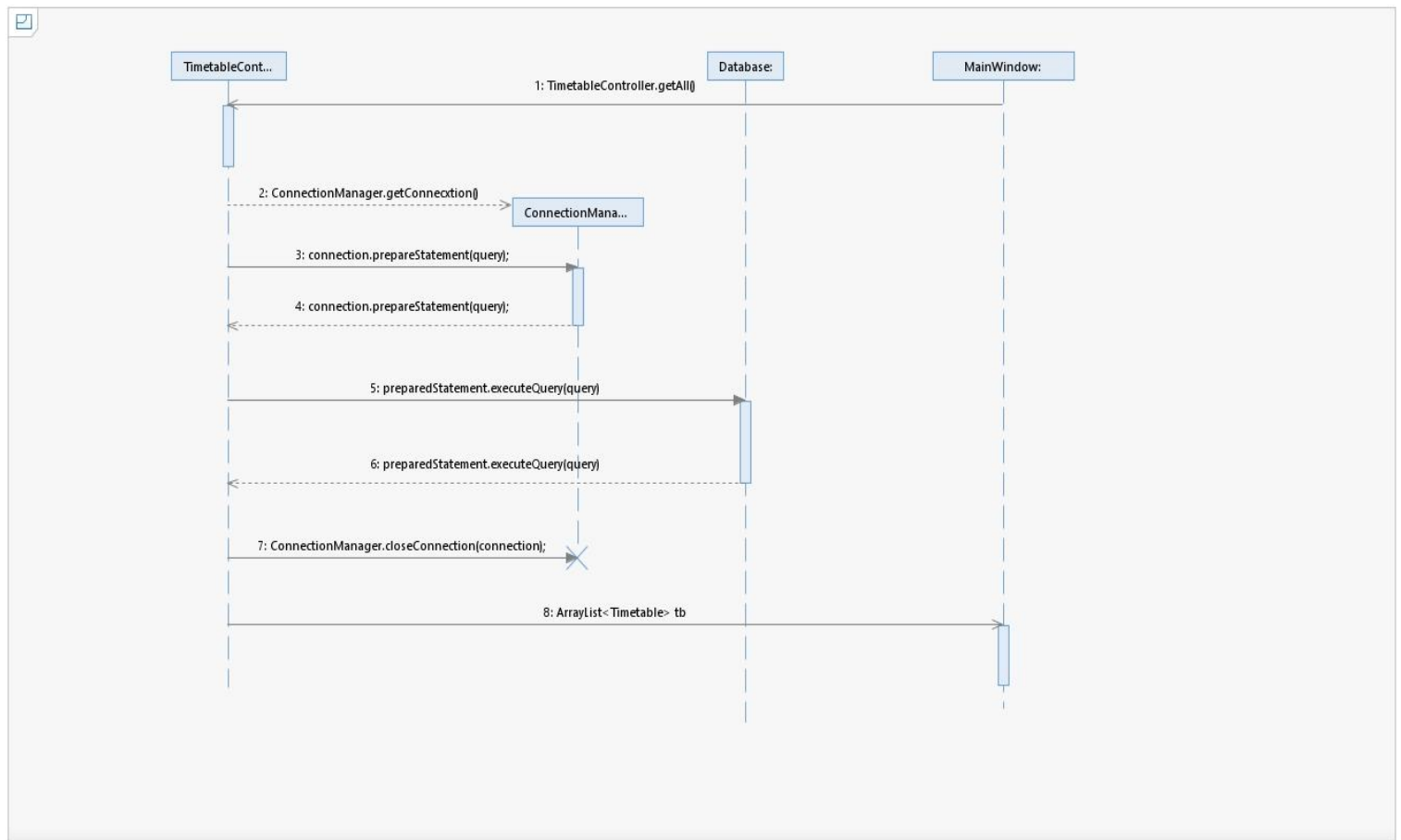


## Diagramma di sequenza

Si è scelto di implementare il diagramma di sequenza per illustrare il meccanismo con cui la classe `MainWindow` recupera le entry del database.

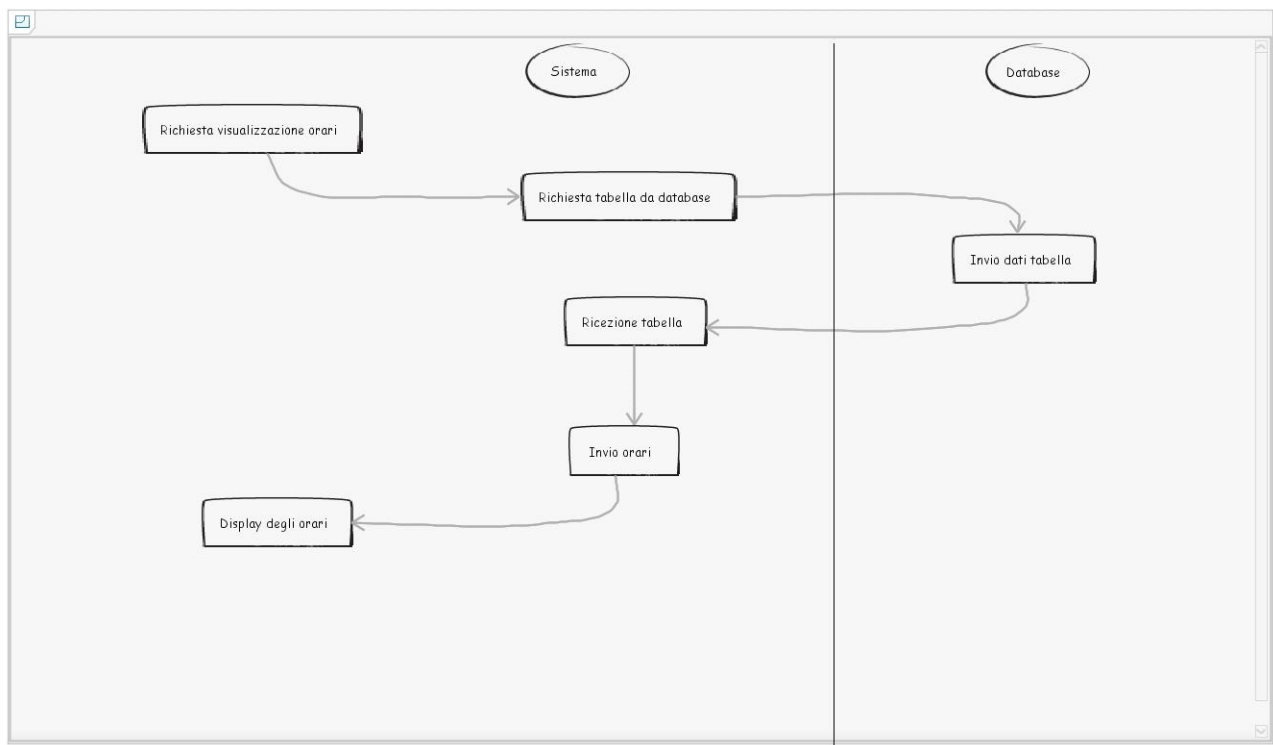
Per prima cosa `MainWindow` chiama il metodo `TimetableController.getAll()` che restituisce un `ArrayList` di `Timetable`. Quest'ultima è una classe che non fa altro se non tenere in memoria i dati di ogni riga della tabella `Timetable` del database.

Per prima cosa `TimetableController` crea una nuova connessione attraverso `ConnectionManager.getConnection()`, dopo di che crea un `PreparedStatement` (`connection.prepareStatement(query)`) con il quale interrogherà il database attraverso l'operazione `preparedStatement.executeQuery(query)`, salvando il risultato in un `ResultSet`. Dato che in seguito alla chiusura della connessione il `ResultSet` non sarebbe più utilizzabile si è scelto di creare un `ArrayList` che contiene oggetti `Timetable` ognuno dei quali corrisponde a una riga della tabella del database. In fine l'`ArrayList` viene restituito alla classe `MainWindow` che potrà utilizzarlo. Questo modello cerca di rispettare il pattern Architetturale Model-view-controller (MVC)



## Diagramma delle Attività

Diagramma delle attività per il recupero degli orari da database.



### Istruzioni per l'avvio del programma

Per testare il programma bisogna per prima cosa creare il database e l'utente che può accedervi.

All'interno del progetto vi è una cartella nominata UtilityFiles, vi si trovano diversi files, i due che servono al momento sono DB\_orari1.sql e create\_user.sql. Aprendoli con MySQL ed eseguendoli il primo creerà il database, mentre il secondo creerà un utente con i diritti per accedervi.

Il secondo passaggio necessario per l'esecuzione del programma è modificare la riga 40 della classe ConnectionManager. Qui si trova il path al file database.config.txt che contiene le informazioni che permettono a java di connettersi al nostro database. Ovviamente il progetto originale prevedeva di mostrare una finestra a schermo che richiedesse all'utente di inserire questo dato, ma nell'implementazione pratica è sorto un problema che non sono stato in grado di risolvere: quando si richiede all'utente di inserire un input da una finestra di JavaFX il sistema non attende l'inserimento del dato, ma mostra la finestra e salva il valore null nella variabile.

Lo stesso problema si presenta tentando di inserire una nuova entry nel database: nonostante il metodo insertTimetable della classe TimetableController funzioni, nel momento in cui compare a schermo il widget per la selezione della data e dell'ora le variabili che dovrebbero contenerle vengono istanziate a null e il programma va in exception.

Una volta impostato il path al file database.config.txt (che si trova anch'esso nella cartella UtilityFiles del progetto) sarà possibile avviare il programma che mostrerà a schermo una piccola finestra con due bottoni: "Inserisci Orario" e "Mostra Orario". Il primo apre altre due finestre (sovrapposte all'inizio) dove si dovrebbe poter inserire data e ora di inizio della lezione e una per la data e ora della fine. Non ho implementato quelle per l'inserimento dei valori per il corso e l'aula perché sarebbero state sostanzialmente identiche e avrebbero avuto lo stesso problema delle prime due.

Il tasto "Mostra Orari" mostra invece gli orari delle lezioni con relativi nomi dei corsi, classi in cui si svolgono e nome del professore che tiene la lezione. Non è quindi la tabella Timetable (che sarebbe poco utile a uno studente), ma il risultato di una join con le tabelle Course e Classroom per avere le informazioni più rilevanti.