**“Software Engineering”**

**Course**

**a.a. 2019-2020**

**Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)**

**Progetto 4**

**Calendario, personalizzato, delle lezioni**

|  |  |
| --- | --- |
| **Date** | 31/01/2020 |
| **Deliverable** | D2-consegna finale |
| **Team (Name)** | Team CAD |

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| **Name & Surname** | **Matriculation Number** | **E-mail address** |
| Chiara Michelucci | *252633* | *chiara.michelucci@student.univaq.it* |
| Dragos Stratulat | *248779* | *dragoscatalin.stratulat@student.univaq.it* |
| Alessandro Carestia | *244236* | *alessandro.carestia@student.univaq.it* |

Table of Contents of this deliverable

List of Challenging/Risky Requirements or Tasks3

Stato dell’arte ...........................................................................................................................................4

Raffinamento dei requisiti.........................................................................................................................9

Servizi (con prioritizzazione)..........................................................................................................................9

Requisiti non funzionali...............................................................................................................................10

Scenari d’uso dettagliato.............................................................................................................................11

Excluded Requirements...............................................................................................................................12

Assunzioni ...................................................................................................................................................12

Use Case Diagrams......................................................................................................................................12

Architettura Software..............................................................................................................................13

Component Diagram...................................................................................................................................13

Sequence Diagram ......................................................................................................................................14

Dati e loro modellazione..........................................................................................................................19

Design Decision.......................................................................................................................................20

Design di Basso Livello.............................................................................................................................21

FRs and NFRs: are satisfied by design.......................................................................................................22

Effort Recording ......................................................................................................................................22

Pert..............................................................................................................................................................22

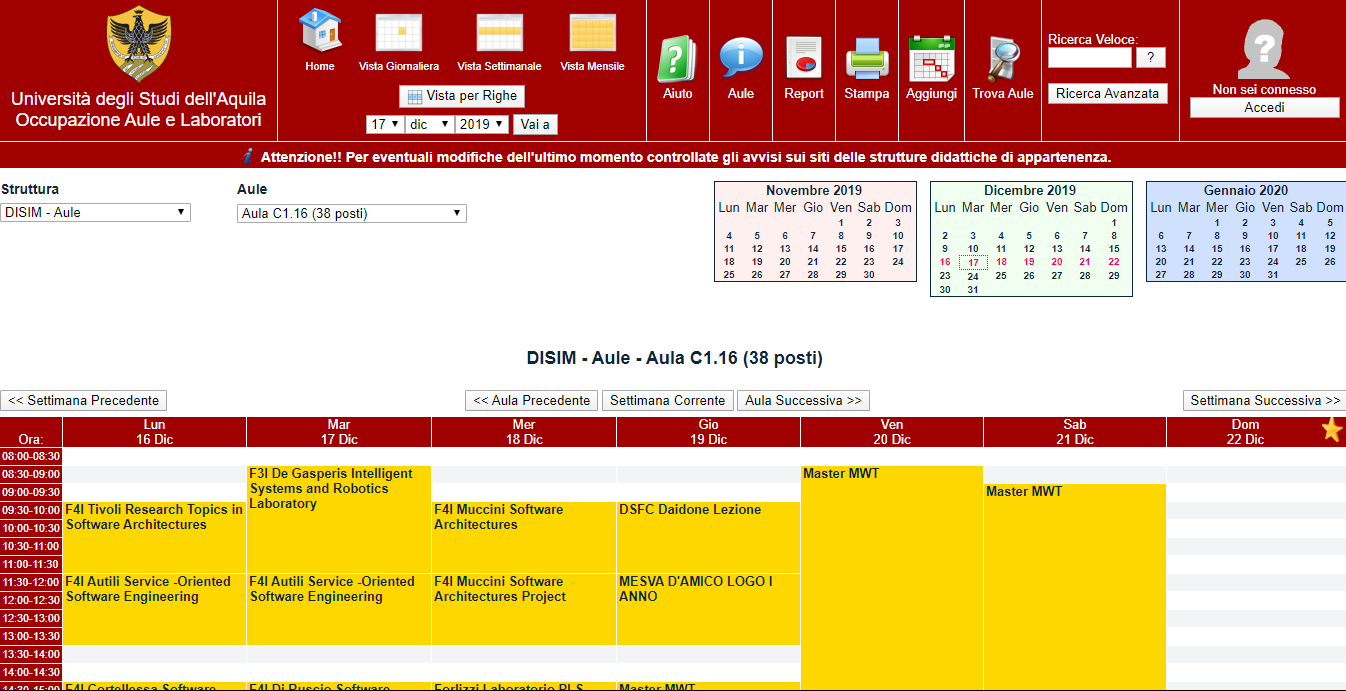
Logging........................................................................................................................................................23

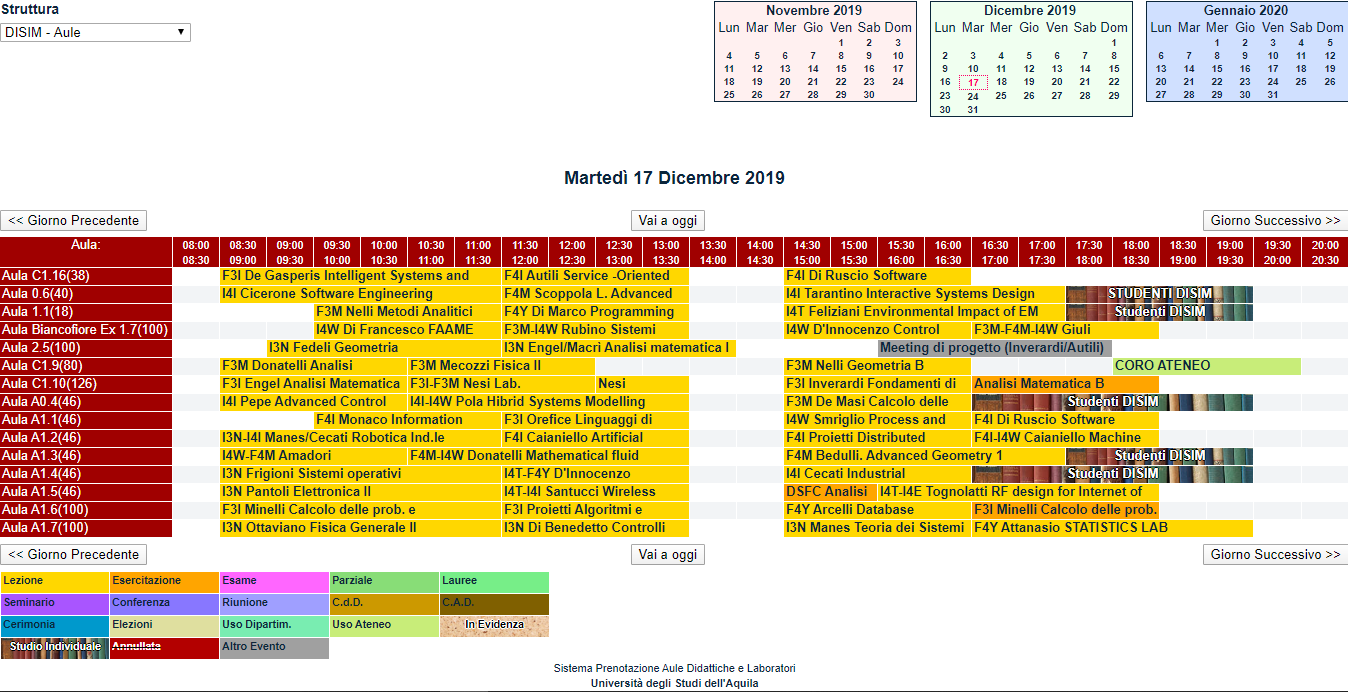
Appendix. Prototype ...............................................................................................................................24

|  |  |  |  |
| --- | --- | --- | --- |
| ***List of Challenging/Risky Requirements or Tasks*Challenging Task** | **Date the task is identified** | **Date the challenge is resolved** | **Explanation on how the challenge has been managed** |
| Limitazione del data entry | 27/11/2019 | 19/12/2019 | Gli amministratori avranno a disposizione funzionalità per facilitare l’inserimento multiplo di lezioni. Agli studenti verrà facilitato la selezione delle materie di interesse grazie ai dati inseriti durante la fase di registrazione. Le soluzioni adottate saranno discusse nel dettaglio nelle sezioni a seguire. |
| Implementazione supporto vocale | 27/11/2019 | 19/12/2019 | Il gruppo ha convenuto che un eventuale supporto vocale non porterebbe ad un significativo miglioramento della user experience, in quanto tale funzionalità trova maggior utilità in applicazioni come quelle telefoniche (ad esempio call center automatici), sistemi di dettatura e sistemi di controllo per la navigazione satellitare o del telefono in auto tramite comandi vocali. Pertanto non verrà implementato. |
| Gestione cambio aula/orario lezione e/o annullamento senza recupero | 27/11/2019 | 19/12/2019 | Gli amministratori avranno a disposizione più funzionalità specifiche per far fronte ai differenti casi di modifica delle lezioni. Il sistema provvederà a notificare (tramite e-mail) tali cambi agli studenti interessati. |
| Come tenere aggiornati i dati di una dashboard studente limitando le richieste client | 19/12/2019 |  |  |
| Tempo di vita dei dati relativi alle singole lezioni | 19/12/2019 | 19/12/2019 | Vista la diversità tra i calendari didattici di corsi diversi il team ha convenuto che non è possibile stabilire una durata standard, pertanto verrà data agli amministratori la possibilità di impostarlo manualmente per ogni corso. |
| Gestione diversificata per utenti (studenti) registrati e non | 19/12/2019 | 19/12/2019 | Gli utenti non registrati potranno scegliere solo il corso di laurea e il relativo anno di interesse, e generare un calendario in base ad esso. Gli utenti registrati, invece, avranno a disposizione diverse funzionalità grazie ai dati inseriti in fase di registrazione. |
|  |  |  |  |
|  |  |  |  |

1. Stato dell’Arte

Il team ha analizzato in primis il sistema utilizzato dall’Università de l’Aquila per la gestione dei corsi, orario lezioni e aule impegnate, ovvero Linfcop: un sistema molto comune e utilizzato da molti anni. Il sistema si appoggia al The Meeting Room Booking System, un software open source, gratuito per l’utilizzo, distribuzione e modifiche. Lo scopo principale del sistema è quello di fornire allo stesso tempo sia le funzionalità rivolte all’amministrazione universitaria, sia i servizi web a valore aggiunto rivolti al singolo studente.

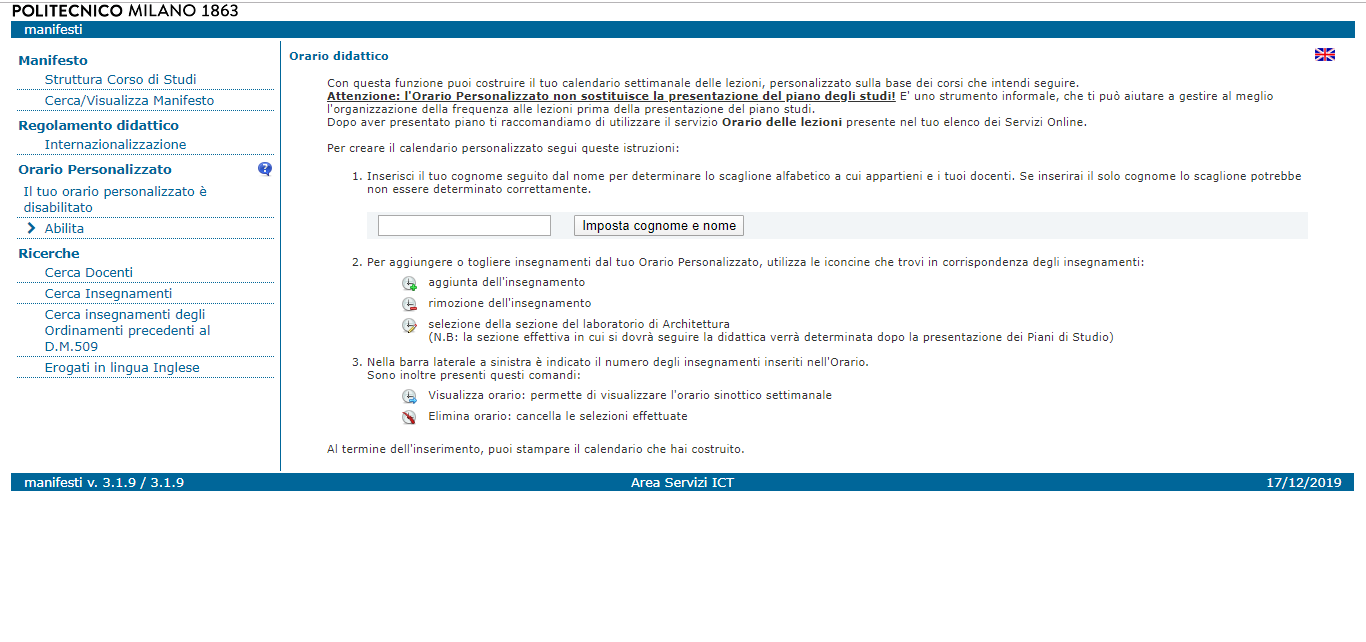




The Meeting Room Booking System attualmente è un sistema utilizzato da molte Università di prestigio.

Successivamente sono stati presi in considerazione i servizi di altre Università italiane, tra cui Milano e Bologna.

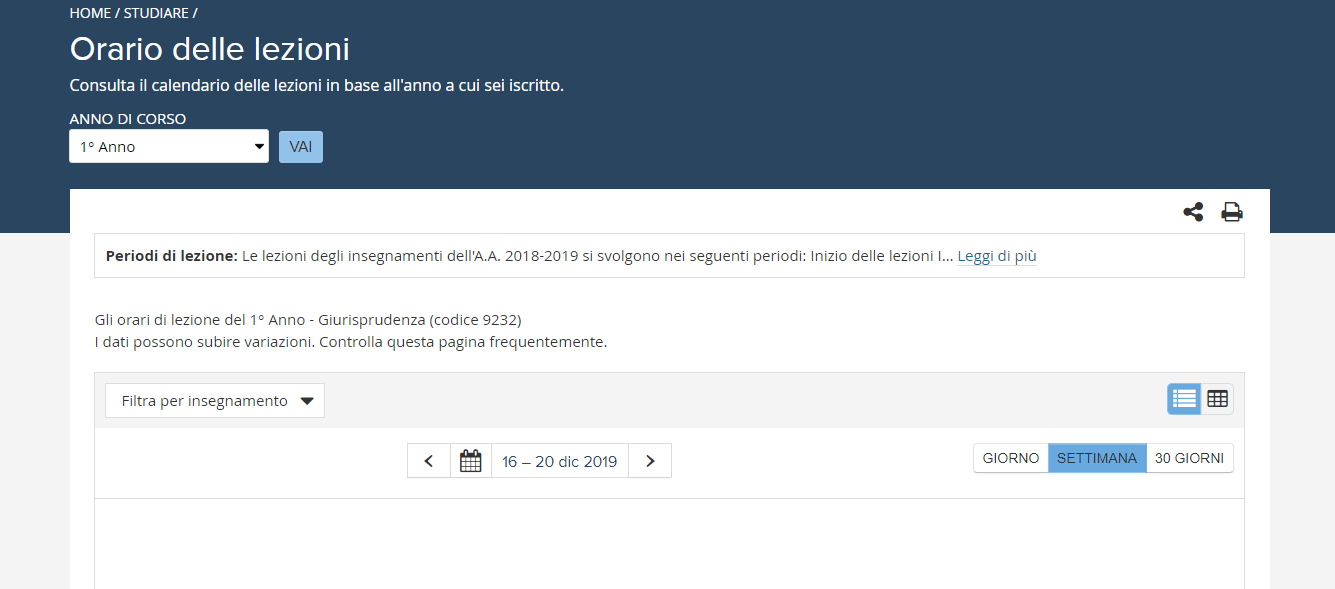
Il Politecnico di Milano da’ la possibilità di inserire il proprio nome e cognome per poi procedere alla creazione del calendario personalizzato delle lezioni.

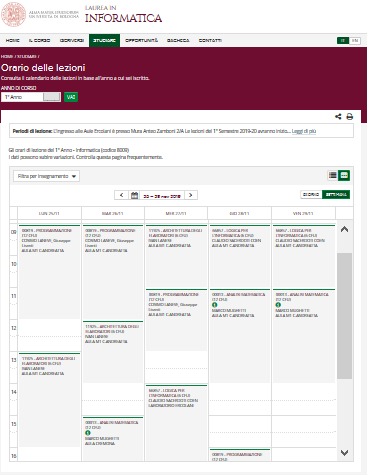




Questo servizio si avvicina di molto al genere che il team dovrà sviluppare.

L’Università di Bologna invece da la semplice possibilità di visualizzare l’orario delle lezioni in base al corso di laurea e all’anno.





Analizzando altri servizi il team ha notato che non tutte le Università mettono a disposizione la possibilità di creare, ad uno e più studenti, un calendario personalizzato delle lezioni.

Un ulteriore servizio scoperto dal team è EasyRoom, utilizzato dalle Università di Salerno, Parma, Genova, Milano, Modena e Reggio Emilia ed altre ancora. Questo sistema permette di inserire delle richieste per prenotare un’aula e visualizzare l’orario di lezione.



Cliccando sulla sede di interesse, è possibile accedere alla visualizzazione dell’agenda della giornata odierna delle aule della sede selezionata, all’interno della quale troverà gli impegni già confermati.

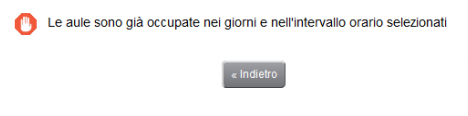
L’agenda visualizzata sarà composta da una griglia in cui sulle colonne troverà le Aule della sede selezionata mentre nelle righe gli intervalli temporali della giornata.

Le caselle colorate indicano gli impegni già confermati in aula; i colori delle caselle consentono di visualizzare a colpo d’occhio quali sono le tipologie di impegni che saranno svolti in aula.

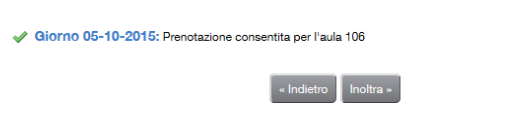
Gli spazi colorati di grigio chiaro rappresentano invece momenti della giornata in cui l’aula non risulta ancora occupata e quindi possono essere da Lei utilizzati per effettuare una richiesta di prenotazione.



Nel momento in cui si prenota un’aula, se quest’ultima è occupata si visualizza un messaggio di errore,



altrimenti si avrà un messaggio positivo che andrà inoltrato per avere ulteriore conferma.



Questo sarà gestito da noi per l’eventuale spostamento di orario/aula di lezioni.

Il servizio EasyRoom rientra molto nelle caratteristiche decise dal team e verrà preso in considerazione insieme a The Meeting Room.

1. Raffinamento dei Requisiti

#### A.1 Servizi (con prioritizzazione)

* **Accesso al sistema**   
  Il sistema permetterà agli studenti di effettuare il login utilizzando le stesse credenziali della piattaforma esse3web.
* **Visualizzazione corsi seguiti** (o seguibili)  
  il sistema genererà automaticamente una lista di corsi in base all’anno di corso e all’indirizzo di studi dello studente (informazioni inserite dallo studente in fase di registrazione). Materie a scelta esterne all’indirizzo di appartenenza dovranno essere inserite nella lista manualmente (per facilitare tale compito verranno messi a disposizione dei filtri).
* **Generazione calendario personalizzato**   
  Una volta selezionati i corsi che si vogliono realmente seguire, potrà essere generato il calendario personalizzato, che consisterà in una vista settimanale (sarà possibile cambiare la settimana visualizzata) del proprio programma, in cui sarà visibile anche l’aula in cui si tiene la lezione.
* **Generazione lista docenti**  
  Sempre facendo riferimento alla lista dei corsi selezionati per la creazione del calendario, ad ogni corso sarà allegato il nome del docente e tramite link ci si collegherà al portale universitario e verranno visualizzate le varie informazioni (dal catalogo corsi CINECA) con i relativi orari di ricevimento (ottenibili sito del dipartimento a cui appartiene).
* **Visualizzazione lezioni in corso** (della propria lista)   
  Nella homepage sarà presente un widget che mostrerà in real-time tutte le lezioni del proprio corso di laurea (indicando il nome della materia e dell’aula) che si stanno svolgendo.
* **Gestione cambio aula/orario lezioni nei calendari personalizzati**Il sistema sarà in grado di notificare eventuali cambi di aula/orario delle lezioni e di modificare opportunamente il calendario dello studente.
* **Note/promemoria lezioni**   
  Lo studente avrà la possibilità di aggiungere note testuali o promemoria (tipo sveglia) sulle lezioni presenti nel proprio calendario.
* **Gestione DataBase**Il sistema permetterà agli amministratori di gestire le attività CRUD su aule, corsi, lezioni e docenti.
* **Aggiunta eventi esterni**  
  Il sistema deve dare la possibilità allo studente di creare eventi personali, questi eventi non verranno proposti automaticamente dal sistema in comparazione con i corsi e serviranno a coprire qualsiasi altro evento scolastico che lo studente vorrà seguire o partecipare (quali seminari, esami, ecc...).

|  |  |  |
| --- | --- | --- |
| ALTA | MEDIA | BASSA |

Legenda:

* L’ORDINE E’ DECRESCENTE

|  |  |
| --- | --- |
| **IMPORTANZA** | **COMPLESSITA’** |
| *Accesso al sistema* | *Visualizzazione lezioni in corso* |
| *Generazione calendario personalizzato* | *Gestione cambio aula/orario* |
| *Gestione DataBase* | *Generazione calendario personalizzato* |
| *Gestione cambio aula/orario* | *Gestione DataBase* |
| *Visualizzazione lezioni in corso* | *Note/Promemoria lezioni* |
| *Aggiunta eventi esterni* | *Aggiunta eventi esterni* |
| *Note/Promemoria lezioni* | *Generazione lista docenti* |
| *Generazione lista docenti* | *Visualizzazione corsi seguiti* |
| *Visualizzazione corsi seguiti* | *Accesso al sistema* |

#### A.2 Requisiti non Funzionali

* **Portabilità**  
  Il sistema dovrà essere disponibile per dispositivi mobile e non e per i principali sistemi operativi attualmente in circolazione.
* **Efficienza**  
  Data la disomogeneità dei dispositivi per quanto riguarda le caratteristiche tecniche, il sistema dovrà limitare l’utilizzo di risorse per garantire un corretto funzionamento anche a dispositivi di fascia bassa.
* **Usabilità**  
  Come da specifica, il sistema sarà semplice da usare, permetterà all’utilizzatore di usufruire delle diverse funzionalità in pochi semplici passi limitando il data entry.
* **Supportabilità**   
  il sistema sarà facile da modificare in modo da renderlo indipendente da una specifica università.
* **Affidabilità**  
  Il sistema deve poter funzionare in qualsiasi momento (margine di errore molto basso) e deve fornire, agli utenti che lo utilizzano, dati veritieri e sempre aggiornati.
* **Sicurezza**  
  Il sistema non deve permettere la visualizzazione dei dati di terzi soggetti da parte dell’utente connesso al momento.

## A.3 Scenari d’uso dettagliati

* **Visualizzazione dell’orario**  
  Una volta completata la prima configurazione, il calendario settimanale presentato nella homepage verrà riempito con i corsi selezionati dallo studente. Questo potrà essere visualizzato solo accedendo al sistema. In caso di non autentificazione da parte dell’utente, verranno presentati 3 campi che devono essere completati con: corso di laurea, anno del percorso di laurea e semestre. Una volta compiuto questo passo gli verrà stampato l’orario settimanale generico con tutti i corsi del corso di laurea, dell’anno e semestre selezionati.
* **Log in**   
  Al momento dell’iscrizione in una università, ogni singolo studente riceverà un username per accedere ai vari servizi messi a disposizione dall’ateneo. Queste credenziali verranno riprese dal nostro servizio “calendario” per permettere agli utenti di accedervi con maggiore facilità.
* **Prima configurazione dell’orario**   
  Dopo la prima autentificazione, all’utente verrà presentato un calendario settimanale vuoto. Lo studente dovrà creare un orario delle lezioni personalizzato e, una volta scelta questa opzione, gli verrà presentata una lista di corsi che esso potrà seguire, in base al suo anno di immatricolazione, che comprenderà anche materie già frequentate ma di cui non si è ancora sostenuto (con successo) l’esame. Una volta scelti i corsi che lo studente vorrà seguire, il sistema creerà un orario settimanale contenente i corsi selezionati.
* **Modifica dell’orario personalizzato**  
  Quando lo studente desidera cambiare il suo orario potrà farlo accedendo di nuovo alla funzione di configurazione dove gli verrà presentata una lista con i corsi attualmente scelti, qui poi potrà rimuovere i corsi che non desidera più seguire e/o inserirne di nuovi. Il sistema quindi creerà in automatico il nuovo orario e lo presenta nella homepage.
* **Variazione data/orario/aula di una lezione**Le variazioni (cambio aula oppure cambio orario/giorno di lezione, o entrambi) potranno essere effettuate solo da parte di un amministratore autenticato. Una volta effettuato l’accesso al sistema, il principale compito di quest’ultimo consisterà nell’apportare la gestione del le opportune variazioni. nel modo seguente:  
   1) Selezione della lezione (specificando aula, giorno e ora) oggetto della modifica;  
  2) Scelta dell’azione da intraprendere: cancellazione senza recupero; cancellazione specificando una lezione di recupero; (comprende la modifica totale o parziale dei dati riguardanti lo svolgimento di una lezione). Se la variazione proviene da un docente, quest’ultimo dovrà comunicare ad un amministratore la variazione da effettuare; lo studente potrà visualizzarla subito.
* **Scenario Utente non loggato**Un utente non loggato potrà semplicemente visualizzare l’orario di lezione base dell’università, secondo il corso e l’anno da lui selezionato, la lista professori e i widget di quell’anno. Gli sarà impossibile creare un calendario personalizzato in quanto non c’è possibilità di mantenere tracciamento di dati.

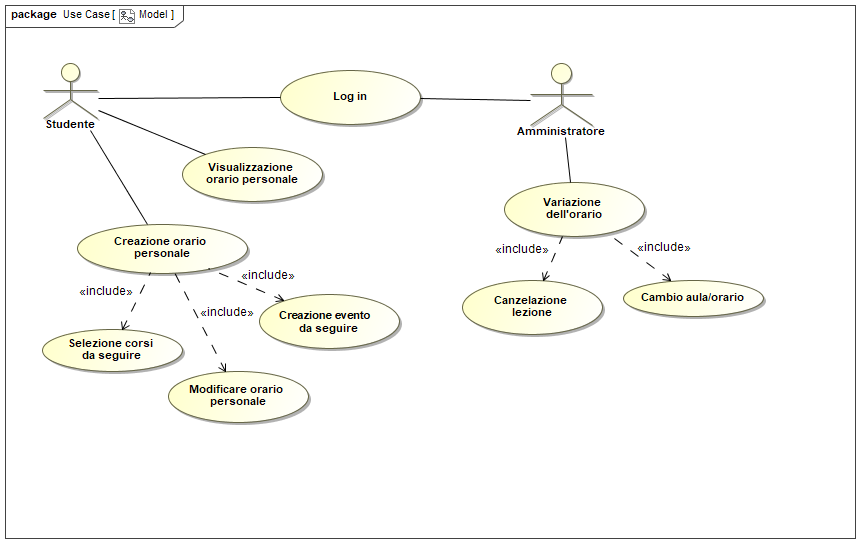
### A.4 Excluded Requirements

* Ufficio professori: Il sistema non si preoccuperà di fornire la locazione dell’ufficio del professore, ma solo un orario di ricevimento.
* Nel widget non verranno visualizzati gli orari di ricevimento dei singoli professori, ma solo le lezioni in corso.

## A.5 Assunzioni

* In caso di lezione annullata,spostamento aula o altre variazioni, il professore dovrà avvisare la segreteria universitaria che provvederà alla modifica dell’orario sul sito dell’Università. Tale modifica verrà poi apportata dall’amministratore del sistema e tramite notifica verrà comunicato ai vari studenti.
* Gli slot saranno di 30 minuti (a partire dalle 08:00, fino alle 20:00) visto che molti sistemi per la calendarizzazione degli eventi prevedono slot temporalmente più grandi .
* Si presuppone che l’Università svolga lezioni dal lunedì al sabato. (in caso le lezioni siano da lunedì a venerdì si avrà la sezione del sabato vuota)
* Sarà possibile avere lezioni annullate senza prevedere lezioni di recupero. In tal caso non si mostrerà nulla nella sezione di recupero.
* Una qualsiasi variazione di una lezione (che sia ora, giorno o aula) implica la “cancellazione” della lezione modificata e la creazione di una nuova con dati nuovi. (per cancellazione, dal punto di vista pratico, si intende che cambia il valore del campo ‘annullata’).

## A.6 Use Case Diagrams



Gli attori principali nello Use Case saranno gli studenti e gli amministratori. Entrambi dovranno accedere al sistema con le credenziali personali.

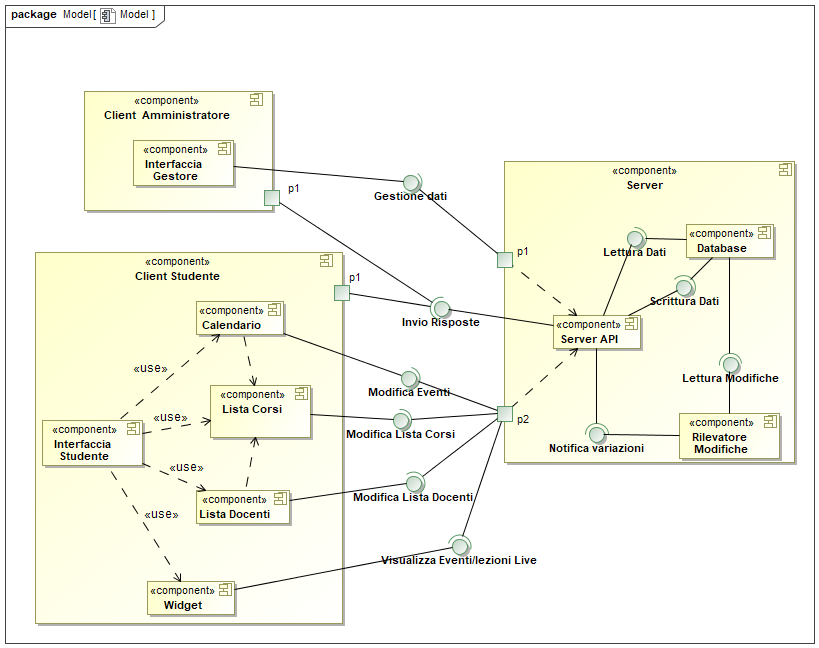
Lo studente avrà accesso all’orario personale e alla lista di corsi a lui disponibili, in base all’anno accademico e i corsi già superati. Questo creerà il suo orario personale usando le funzioni offerte dal sistema (configurazione iniziale, modifica e creazione evento indipendente) e potrà essere visualizzato in qualsiasi momento desidera, in quanto questo è autenticato nel sistema.

La figura dell’amministratore serve per portare qualsiasi tipo di variazione (cancellazione lezione, modifica orario di lezione oppure aula di lezione) a questo orario. Abbiamo presunto che l’amministratore abbia accesso anche al sistema dell’Università per gestire le aule e quindi sarà quest’ultimo ad apportare tali modifiche, dopo che il professore

comunicherà l’opportuna variazione da fare.

C. Architettura Software

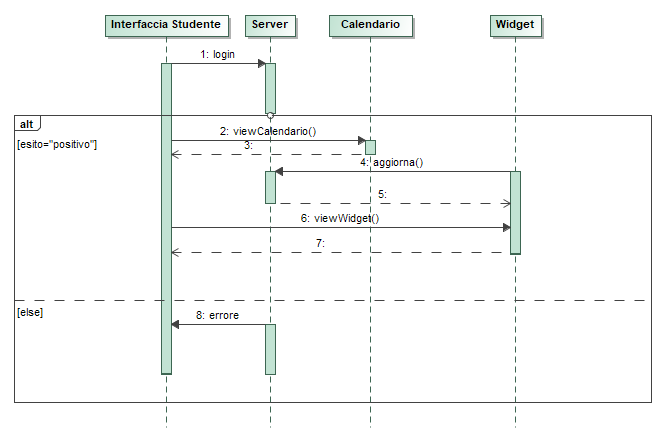
## C.1The static view of the system: Component Diagram



Nel nostro component diagram abbiamo individuato tre componenti fondamentali:

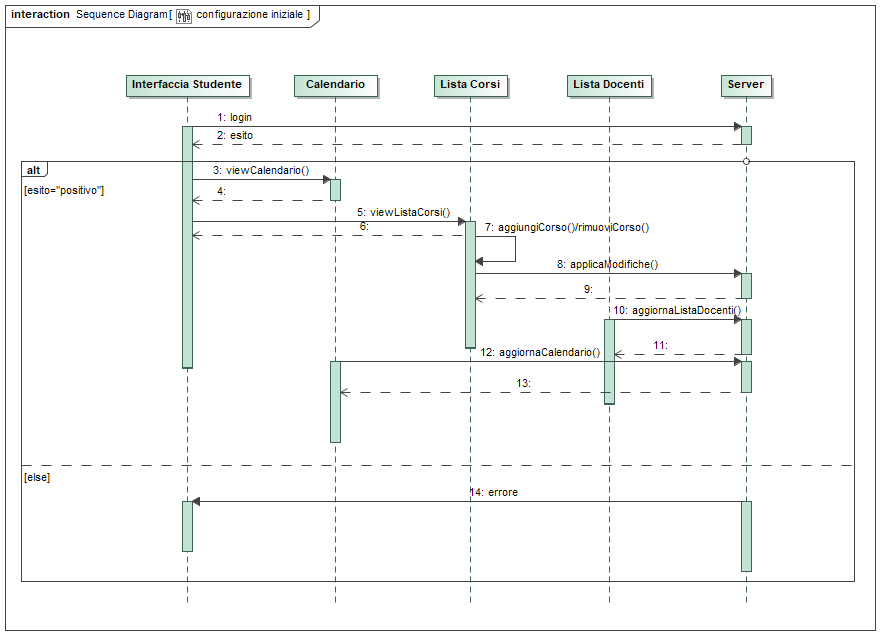
* **Client Amministratore**: contiene l’interfaccia gestore a disposizione dell’amministratore con cui quest’ultimo può modificare i dati relativi a corsi, lezioni, aule e docenti contenuti nel database;
* **Client Studente**: contiene l’interfaccia studente che si occupa della disposizione e visualizzazione del calendario, della lista studenti, della lista corsi e del widget nella schermata. Questi ultimi comunicano con il server API per modificare i dati da loro contenuti..
* **Server**: questo componente contiene a sua volta tre sottocomponenti: il primo è il *database* che conterrà tutti i dati relativi a corsi, lezioni, aule e insegnanti inseriti e/o modificati dagli amministratori; il secondo è il *rilevatore di modifiche*, che notificherà ogni eventuale modifica al database; Il terzo è il *server API* che ha il compito di ricevere ed elaborare le richieste client e comunica con il database e rilevatore modifiche per rispondere ad esse.

## C.2 The dynamic view of the software architecture: Sequence Diagram

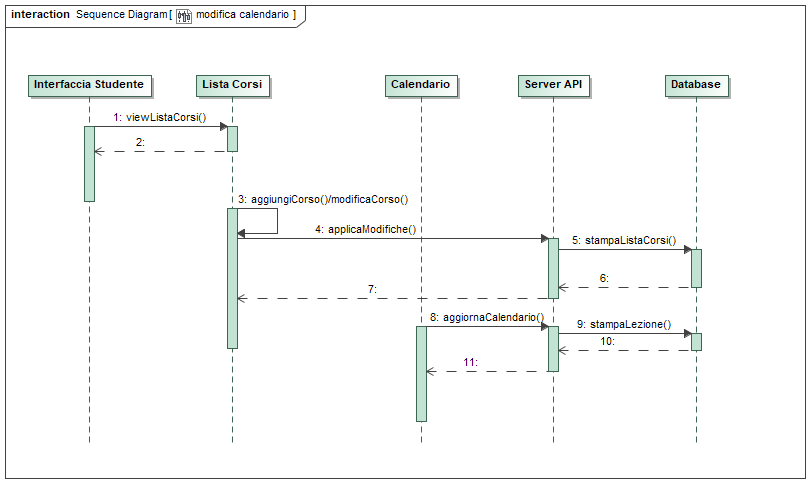


* **Abbiamo presunto che il Login sia andato a buon fine.**

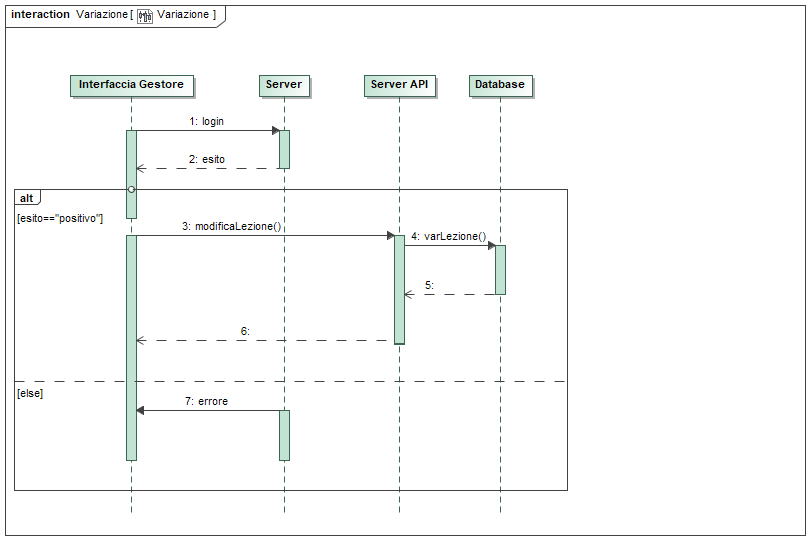
**Accesso** **e visualizzazione:** Una volta effettuato il login con successo, l’utente visualizzerà il proprio calendario personalizzato contenente i corsi nella propria lista di corsi di interesse; visualizzerà il widget contenente le informazioni real-time. In caso contrario si rimanda un errore.



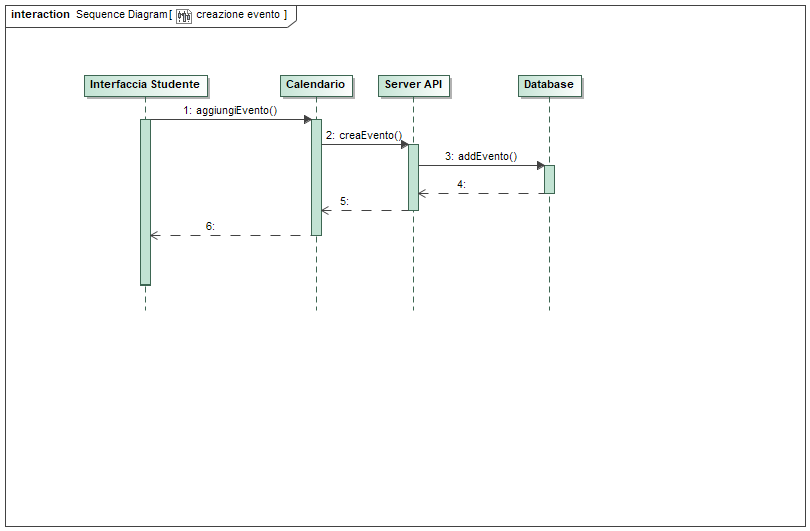
**Configurazione iniziale**: Dopo che lo studente esegue la prima autentificazione questo procederà con la creazione del calendario scegliendo i vari corsi da seguire; una volta scelti, l’interfaccia invia le modifiche apportate (alla propria lista) al server in modo che siano riportate nel database. Si aggiornerà la lista dei docenti e a sua volta il calendario. Tutto questo poi sarà visualizzato nell’interfaccia principale.



**Modifica calendario**: L’interfaccia utente mostrerà la lista dei corsi, successivamente si consente di modificare quest’ultima eliminando/aggiungendo corsi e notificherà tali cambiamenti al server API. Si andranno quindi a prelevare i dati dei corsi appartenenti alla lista aggiornata che andranno restituiti al client così che vengano utilizzati dal calendario per il suo aggiornamento.

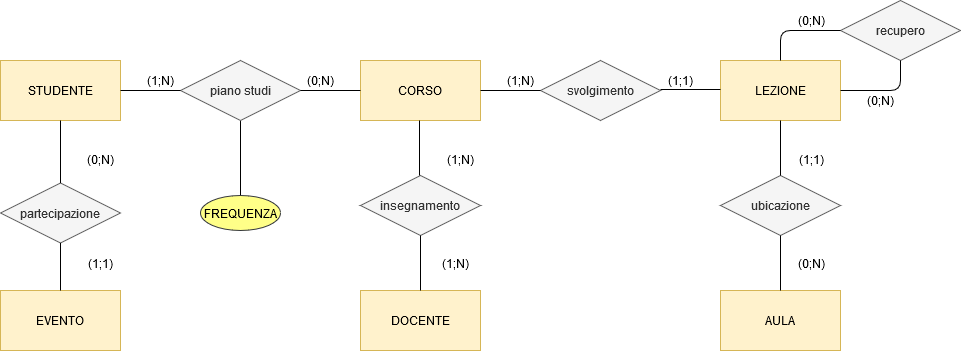


**Variazione**: Dopo che l’utente accede al sistema e dimostra di essere un amministratore, questo potrà procedere a registrare le opportune variazioni delle lezioni, inviando le modifiche al server API che si occuperà di comunicarle al database.



**Creazione Evento**: Lo studente, tramite l’interfaccia studente, aggiunge un evento nel calendario il quale comunica al server la creazione. Qui il server API aggiunge le informazioni relative all’evento nel database.

D. Dati e loro modellazione



STUDENTE: (username, password, e-mail, anno\_immatricolazione)

CORSO: (id\_corso, docente, nome, indirizzo, frequenza\_obbligatoria, anno)

DOCENTE: (id\_docente, nome, cognome, giorno\_ricevimento, orario\_ricevimento, dipartimento)

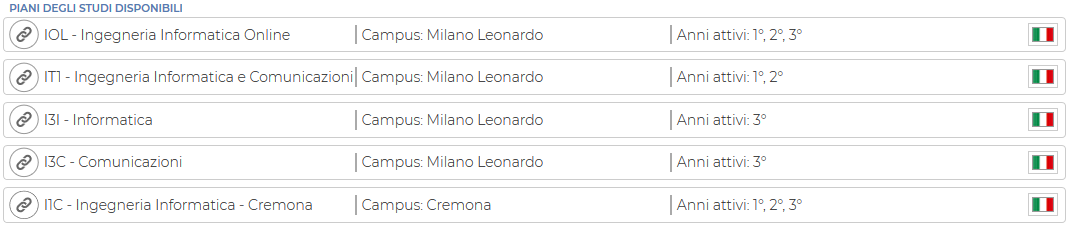
AULA: (id\_aula, edificio, nome, posti)

LEZIONE: (id\_lezione, id\_corso, id\_aula, giorno/ora\_inizio, giorno/ora\_fine, annullata, descrizione)

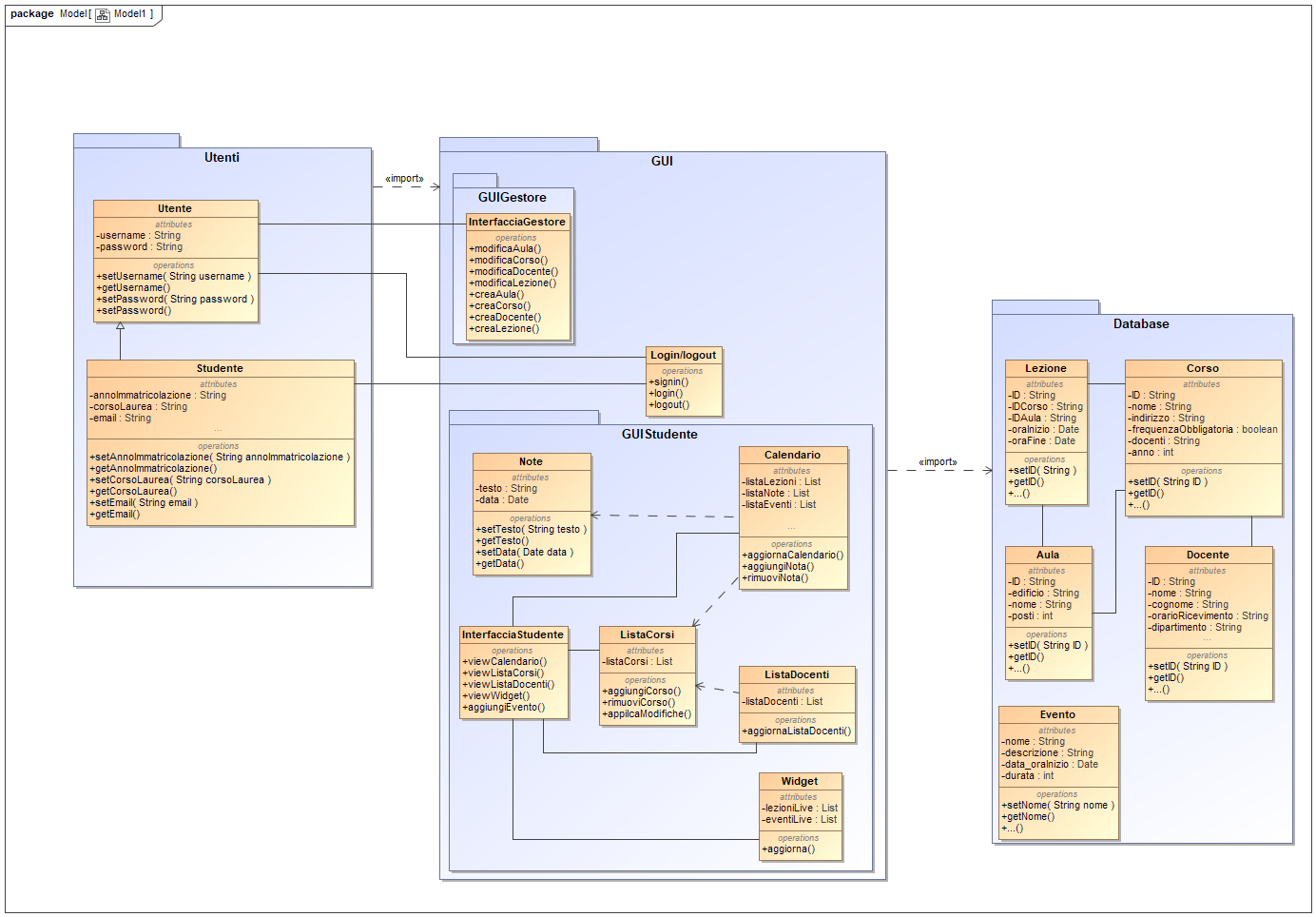
EVENTO: (id\_evento, nome, descrizione data, ora\_inizio, durata)

1. Uno STUDENTE nel suo piano di studi può avere 1 o più (N) corsi e da esso dipende la frequenza.
2. Un CORSO può trovarsi nel piano di studi di N studenti oppure può anche non esser presente in nessun piano di studi.
3. L’insegnamento di un corso può essere svolto da uno o più (N) docenti.
4. Un DOCENTE può insegnare in un corso ma può avere anche più corsi (N) a suo carico.
5. Lo svolgimento di un corso si esegue tramite una o più (N) lezioni.
6. Una LEZIONE può essere ubicata in una singola aula in un certo giorno ad una certa ora.
7. In un’ AULA possiamo trovare lezioni oppure nessuna.
8. Uno STUDENTE può partecipare a 0 o più (N) eventi.
9. Un EVENTO sarà unico per ogni studente, in quanto si parla di eventi personalmente creati da uno studente e inseriti nel suo calendario.
10. Una lezione annullata potrà avere zero o più (N) lezioni di RECUPERO.

E. Design Decisions

* **Accesso sicuro al sistema**Per accedere al sistema si è deciso che bisognerà inserire username e password utilizzate per accedere al servizio gestito dall'università. Un utente potrà essere Amministratore o Studente, l'amministratore avrà il compito di modificare le varie informazioni contenute nel database.  
  Questa pagina di login sarà composta da due campi di testo dove inserire le proprie credenziali e un pulsante per accedere.
* **Struttura Home Page**Dopo aver eseguito il primo accesso al sistema, si visualizzerà un calendario inizialmente vuoto (andrà modificato successivamente dall’utente), in alto a quest’ultimo si troverà il widget con le varie informazioni e lateralmente si avrà un menù con le varie voci riguardanti le azioni da poter eseguire.
* **Integrazione corsi**Ogni Università ha il proprio ordinamento interno quindi, per quanto riguarda i corsi di laurea, lasciamo a loro l’integrazione del sistema di questi ultimi e le varie materie. Il sistema deve offrire il supporto necessario alle Università per qualsiasi loro necessità. Questa decisione è stata presa analizzando il sistema del Politecnico di Milano in quanto permette di scegliere per il terzo anno corsi diversi, o che hanno sempre a che fare con quelli dei primi due anni. In allegato lo screen della sezione di scelta.  
  
* **Utente non loggato**Una volta aperta la home page del servizio, l’utente non loggato potrà solo selezionare l’anno e il corso di suo interesse e li visualizzerà l’orario, il widget e l’elenco professori appartenenti al corso.
* **Scelta del database e framework per l’implementazione**Il team ha deciso di utilizzare MySQL poiché è utilizzato nella maggior parte dei sistemi di calendarizzazione esistenti esaminati. Inoltre la scelta di tale tecnologie già conosciute dai membri del team ridurrà i tempi di learning e avrà un impatto positivo sui tempi di sviluppo. Il framework che verrà utilizzato per l’implementazione sarà Laravel, particolarmente adatto per la creazione di applicazioni.

F. Design di Basso Livello



Il class diagram è diviso in tre package principali:

* Utenti: contiene la classe Utente (amministratore), contenente gli attributi username e password necessari per l’autenticazione, e la classe Studente, a cui si aggiungono gli attributi che indicano l’anno di immatricolazione, l’email istituzionale, il corso di laurea a cui è iscritto.
* GUI: contiene la classe Login, che fornisce i metodi per la registrazione e l’accesso al sistema e il logout dal sistema; il package GUIGestore contenete la classe InterfacciaGestore, che fornisce i metodi per modificare i dati nel database; il package GUIStudente, contiene le classi: calendario che conterrà la lista delle lezioni, degli eventi e le note; Widget per la visualizzazione real-time delle lezioni ed eventi in corso (del proprio corso di laurea); ListaDocenti per ottenere la lista dei professori associati alle materie che si intende seguire; Note per gestire l’aggiunta di note con l’opportuna data;InterfacciaStudente che fornisce i metodi per la visualizzazione del calendario, corsi, docenti e widget; ListaCorsi che conterrà la lista dei corsi selezionati dall’utente.
* Database: contiene tutte le classi per la gestione dei dati relativi a corsi, aule docenti, eventi e lezioni.

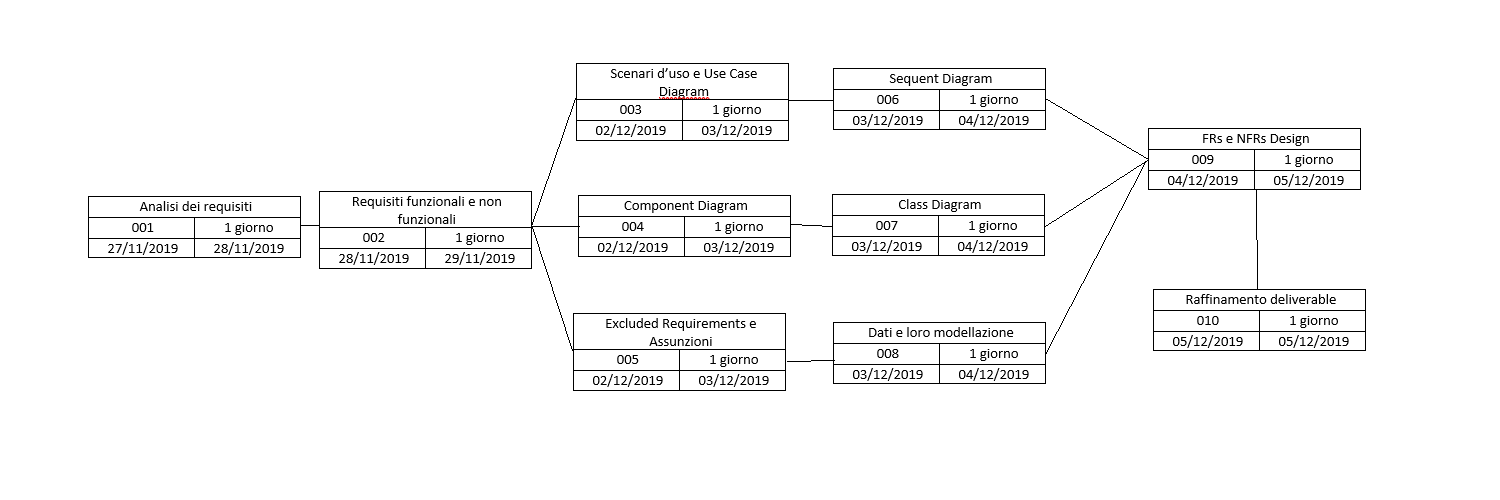
G. Explain how the FRs and the NFRs are satisfied by design

Ecco come alcuni requisiti, funzionali e non, identificati dal sistema vengono soddisfatti attraverso le decisioni di design prese.

* Il sistema consente una giusta navigazione tramite un apposito menu e vari pulsanti per le azioni.
* Per quanto riguarda la sicurezza, il sistema sarà dotato di una parte iniziale di login dove li verrà accertata l’identità e inoltre avverrà il controllo dei dati inseriti, in caso di risultato positivo si potrà accedere al sistema, in caso contrario si visualizzerà una schermata di errore.
* Tutto sarà gestito in modo semplice così da poter essere utilizzato da qualsiasi utente; la creazione del calendario avverrà automaticamente una volta selezionate le materie che l’utente intende realmemte seguire.
* Per avere dati sempre aggiornati, l’amministrazione si occuperà delle varie modifiche, aggiunte o rimozioni, delle varie informazioni interessate.

G. Effort Recording

**PERT**

****

**Logging**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | TimeSpent | Description | Partners |
| 27/11/2019 | 2 learning + 1 doing | Comprensione specifica + List of challenging | 3 |
| 02/12/2019 | 3 doing | Raffinamento requisiti | 3 |
| 04/12/2019 | 3 doing | Architettura software + Design Decision | 3 |
| 05/12/2019 | 3 learning + 6 doing | Completamento Deliverable1 | 3 |
| 17/12/2019 | 5 doing | Revisione documento prima consegna | 3 |
| 19/12/2019 | 4 learning + 2 doing | Definizione Web Services + PostMan | 3 |
| 20/12/2019 | 3 doing | PostMan + Completamento Deliverable 2a scadenza | 3 |
| 15/01/2020 | 4 learning | Laravel + specifiche varie | 3 |
| 16/01/2020 | 6 doing | Miglioramento API + Laravel + modifiche Deliverable | 3 |
| 17/01/2020 | 2 learning + 1 doing | Laravel + Completamento Deliverable 3a scadenza | 3 |
| 29/01/2020 | 3 doing | Vari miglioramenti | 3 |
| 30/01/2020 | 8 doing | DataBase + Interfacce + CollegamentoAPI | 3 |
| 31/01/2020 | 6 doing | Completamento Progetto | 3 |

Learning: 15 ore

Doing: 47 ore

Ore Totali: 62(x3) -> 186 ore

Appendix. Prototype

Link alla Collection su PostMan

<https://documenter.getpostman.com/view/9813985/SWLYAVnT>

* **Descrizione endpoint**

*Gli endpoint principali del nostro sistema sono:*

**/utenti/** : verrà utilizzato per l’utente generico non logato

**/utenti/orari/** : l’utente generico potrà selezionare un certo orario in base al corso di laurea, anno e semestre selezionati

**/utenti/login/** : sezione dove un utente si potrà autenticare

**/amministratori/** : sezione per gli amministratori accessibile solo dopo l’autenticazione. L’amministratore potrà selezionare una lezione e la variazione da apportare a quest’ultima

**/amministratori/lezioni** : sezione degli amministratori per la ricerca e la stampa delle lezioni

**/amministratori/lezioni/add** : sezione degli amministratori per la aggiunta delle lezioni

**/amministratori/lezioni/var** : sezione degli amministratori per la variazione delle lezioni

**/studenti/** : sezione per gli studenti accessibile solo dopo l’autenticazione

**/studenti/calendari/** : sezione dove uno studente può visualizzare l’orario personale

**/studenti/eventi/add** : sezione degli studenti per l’aggiunta di eventi

/**studenti/eventi/var** : sezione degli studenti per la variazione di eventi

**/studenti/eventi/delete** : sezione degli studenti per la cancellazione di eventi

**/studenti/corsi/** : sezione dove uno studente seleziona i corsi da seguire per la creazione dell’orario personale

**/studenti/docenti/** : sezione dove uno studente può visualizzare la lista di docenti dei corsi seguiti

**\*Gli endpoint della componente di routing Laravel sono contenute nella cartella IngSoft\_progetto4 presente nella Root di GitHub.  
Percorsi da seguire:**IngSoft\_progetto4/app/Http/Controllers/...

IngSoft\_progetto4/routes/web.php

**\*DataBase del progetto**

Il database è contenuto nella Root di GitHub, il file è db\_teamcad.php

Sono state prese alcune decisioni per la funzionalità:

-per rendere facile e funzionabile il widget abbiamo avuto la necessità di creare una tabella EVENTS che corrisponde all’unione di LEZIONE ed EVENTO (presenti nello schema er). È stato necessario far ciò per rendere più semplice la prelevazione dei dati e la loro visualizzazione nel widget.