**“Software Engineering”**

**Course**

**a.a. 2019-2020**

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

**Progetto 4**

**Calendario, personalizzato, delle lezioni**

|  |  |
| --- | --- |
| **Date** | 05/12/2019 |
| **Deliverable** | D1 |
| **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 ...........................................................................................................................................3

Raffinamento dei requisiti.........................................................................................................................4

Servizi (con prioritizzazione)..........................................................................................................................4

Requisiti non funzionali.................................................................................................................................5

Scenari d’uso dettagliato...............................................................................................................................5

Excluded Requirements.................................................................................................................................6

Assunzioni .....................................................................................................................................................6

Use Case Diagrams........................................................................................................................................6

Architettura Software................................................................................................................................7

Component Diagram.....................................................................................................................................7

Sequence Diagram........................................................................................................................................8

Dati e loro modellazione..........................................................................................................................10

Design Decision.......................................................................................................................................10

Design di Basso Livello.............................................................................................................................11

FRs and NFRs: are satisfied by design.......................................................................................................12

Effort Recording......................................................................................................................................12

Appendix. Prototype...............................................................................................................................12

***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** |
| Integrazione con sistemi già esistenti | 27/11/2019 |  |  |
| Indipendenza da una specifica università | 27/11/2019 |  |  |
| Limitazione del data entry | 27/11/2019 |  |  |
| Implementazione supporto vocale | 27/11/2019 |  |  |
| Gestione cambio aula/orario lezione | 27/11/2019 |  |  |
| Implementazione widget | 27/11/2019 |  |  |

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 Esse3: un sistema molto comune e utilizzato da molte università di prestigio. Lo scopo principale del sistema è quello di fornire allo stesso tempo sia le funzionalità rivolte all’amministrazione universitaria (segreteria studenti) per gestire tutto il percorso di attraversamento dello studente, sia i servizi web a valore aggiunto rivolti al singolo studente.

Ad oggi sono quasi il 90% delle Università italiane pubbliche ad utilizzare Esse3.

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.

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.
* **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.

#### A.2 Requisiti non Funzionali

* **Portabilità**  
  Il sistema dovrà essere disponibile per le piattaforme più diffuse, mobile e non.
* **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.
* **Log in**   
  L’utente accede al servizio usando le credenziali messe a disposizione dall’università d’appartenenza.
* **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, e esami che ancora dovrà superare. Scegliendo i corsi che vuole seguire il sistema crea un orario settimanale contenendo i corsi da questo selezionati.
* **Modifica dell’orario**  
  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 suoi già scelti, qui poi potrà deselezionare i corsi che non desidera più seguire e selezionare i nuovi corsi. Il sistema crea in automatico il nuovo orario e lo presenta nella homepage.

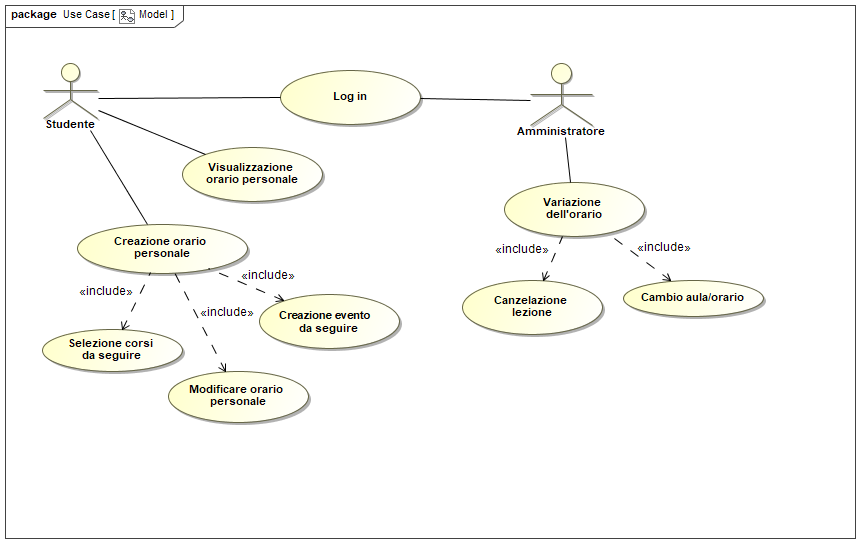
### A.4 Excluded Requirements

* 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, il professore dovrà avvisare la segreteria universitaria che provvederà alla modifica dell’orario sul sito dell’Università.
* L’Università in questione avrà un sistema di gestione occupazione aule e laboratori.

## 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.

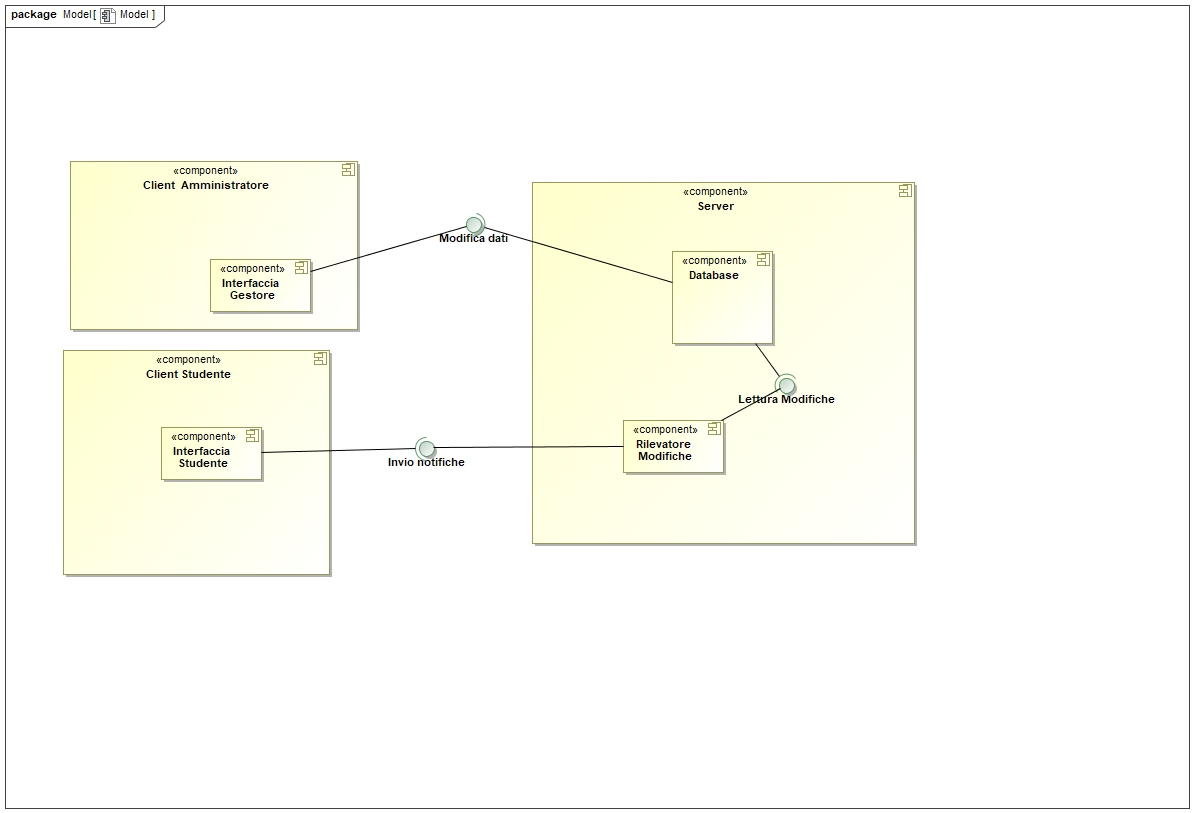
Lo studente avrà accesso all’orario personale e alla lista di corsi a lui disponibili, in base all’anno accademico e i corsi già superati. Questo creerà il suo orario personale usando le funzioni offerte dal sistema (configurazione iniziale, modifica e creazione evento indipendente) e potrà essere visualizzato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

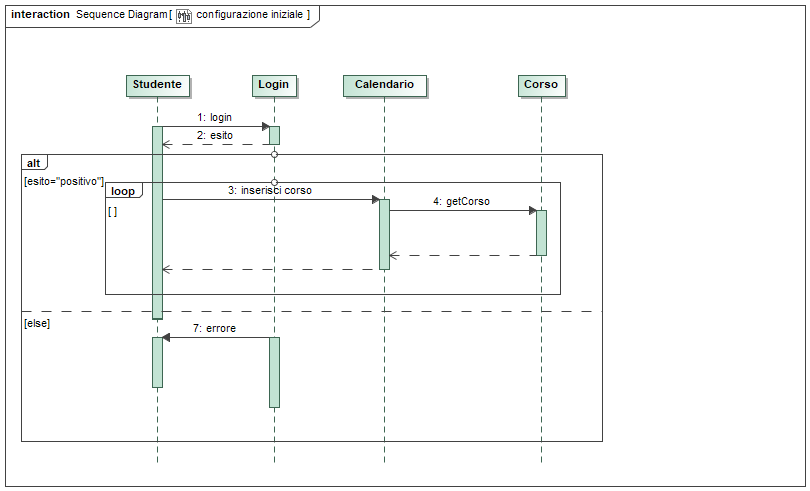
## C.1The static view of the system: Component Diagram

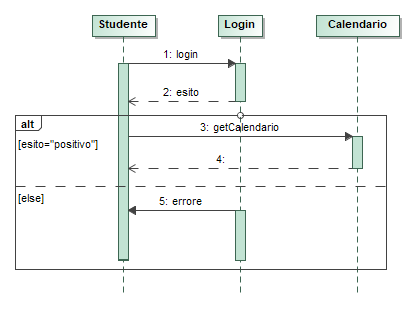


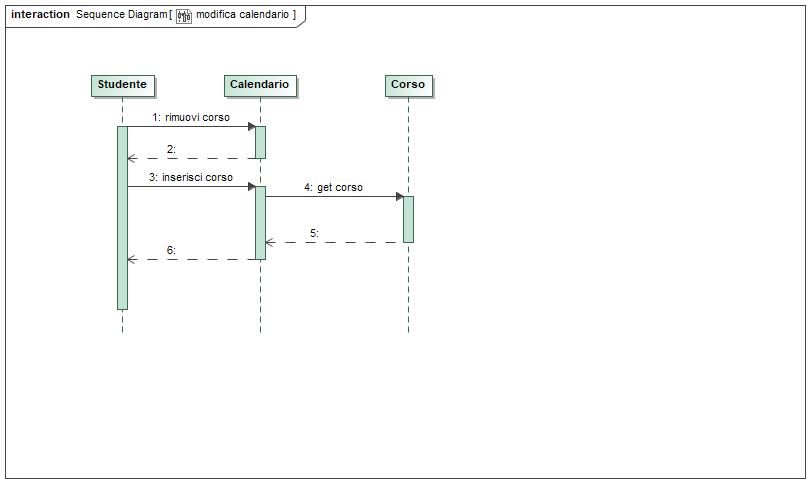
Nel nostro component diagram abbiamo individuato tre componenti fondamentali:

* Client Amministratore: rappresenta l’interfaccia a disposizione dell’amministratore con cui quest’ultimo può modificare i dati relativi a corsi, lezioni, aule e docenti contenuti nel database;
* Client Studente: rappresenta l’interfaccia attraverso il quale uno studente può creare il proprio calendario personalizzato, visualizzare la lista dei docenti associati ai corsi che si intende seguire e il widget per le materie per cui si stanno tenendo lezioni in real-time.
* Server: questo componente contiene a sua volta due 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 invierà informazioni su ogni eventuale modifica al database alle interfacce degli studenti interessati da tali cambi.

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







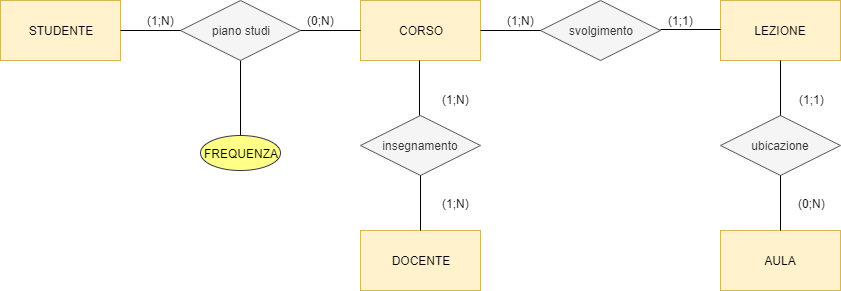
* Abbiamo presunto che il Login sia andato a buon fine.

1.Accesso e visualizzazione: La lifeline studente chiede l’autentificazione alla lifeline login. Se questo è andato a buon fine lo studente può procedere con la visualizzazione del calendario; se l’autentificazione non andrà a buon fine lo studente riceverà un messaggio di errore.

2.Configurazzione calendario iniziale: Dopo che lo studente esegue la prima autentificazione questo procederà con la creazione del calendario scegliendo i vari corsi da seguire.

3. Modifica calendario: Lo studente procede a modificare il suo calendario eventualmente eliminando i corsi presenti sul calendario e li sostituirà con la nuova scelta.

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, orario\_ricevimento, dipartimento)

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

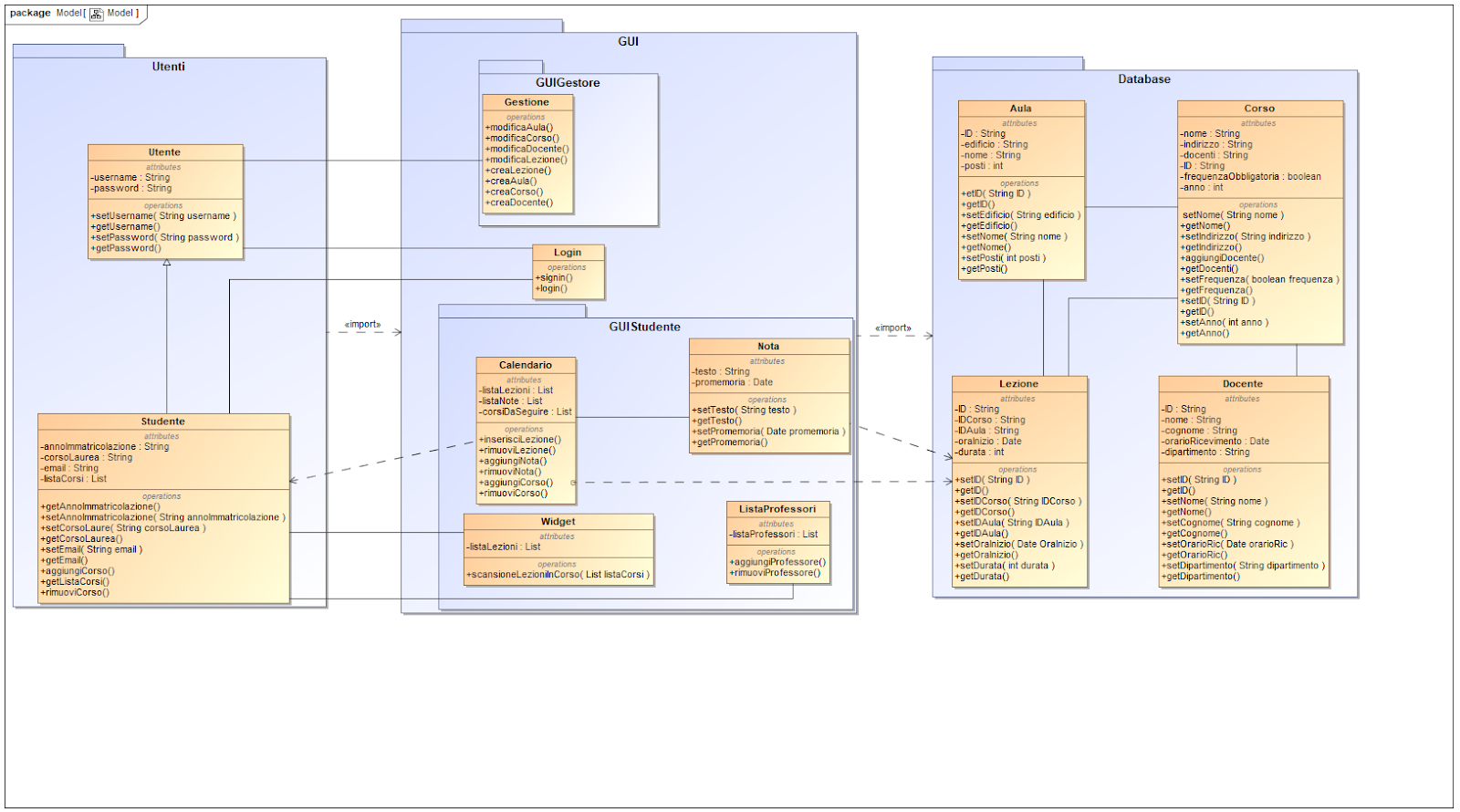
LEZIONE: (id\_lezione, id\_corso, id\_aula, ora\_inizio, durata)

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.
7. In un AULA possiamo trovare lezioni oppure nessuna.

E. Design Decisions

* **Scelta del linguaggio di programmazione**Il linguaggio principale scelto dal team è Java. Questa scelta deriva dal fatto che è uno dei linguaggi più conosciuti dai membri. Per quanto riguarda le interfacce si utilizzerà HTML e CSS perché entrambi di facile utilizzo. Inoltre, se non sarà possibile creare un collegamento con il sistema già esistente dell’Università, il team ha deciso di utilizzare MySQL poichè si connette in modo facile e veloce con Java.
* **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 su orari, aule e lezioni.  
  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.
* **Scelta colori**Per facilitare l’utilizzo del sistema, l’interfaccia sarà implementata usando una paletta di colori speciale per aiutare le persone con deuteranomalia (daltonismo).

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 e la lista dei corsi seguibili.
* GUI: contiene la classe Login, che fornisce i metodi per la registrazione e l’accesso al sistema, il package GUIGestore contenete la classe Gestione, che fornisce i metodi per modificare i dati nel database, e il package GUIStudente, che contiene le classi calendario che conterrà le materie che si intende seguire, Widget per la visualizzazione real-time delle lezioni in corso (del proprio corso di laurea), ListaProfessori per ottenere la lista dei professori associati alle materie che si intende seguire e Nota per gestire l’aggiunta di note e promemoria alle lezioni.
* Database: contiene tutte le classi per la gestione dei dati relativi a corsi, aule docenti e lezioni.

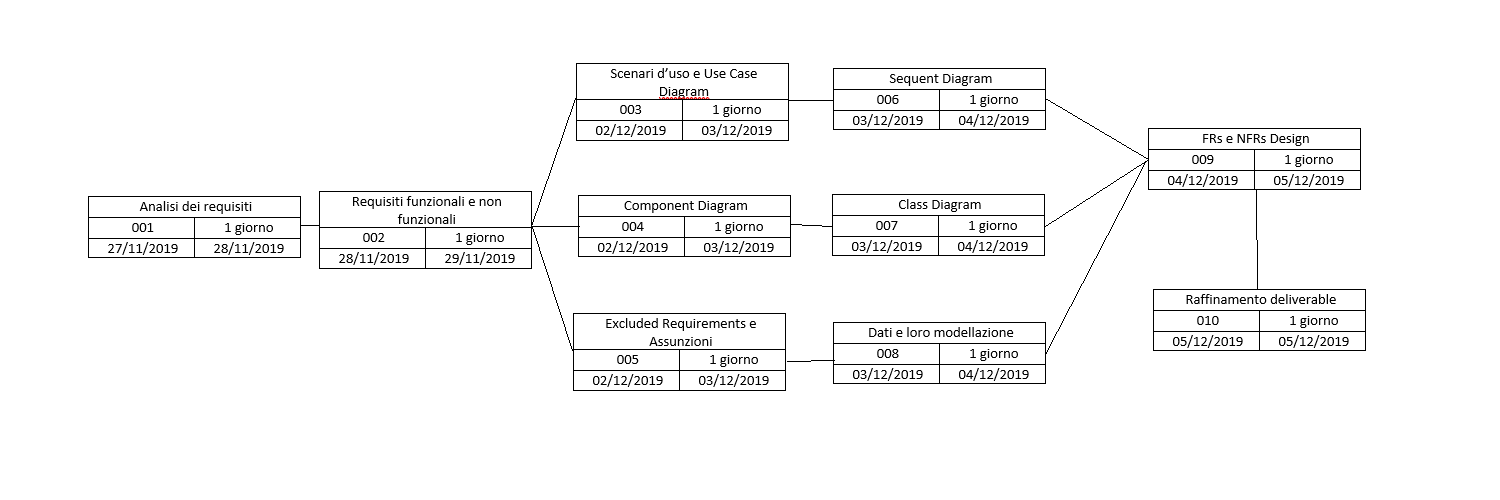
G. Explain how the FRs and the NFRs are satisfied by design

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**

|  |  |  |
| --- | --- | --- |
| DESCRIZIONE | ORE | MEMBRI |
| Ore totali | 18 | Tutti |
| Learning | 5 | Tutti |
| Doing | 13 | Tutti |

Appendix. Prototype