

“Software Engineering” Course a.a. 2019-2020

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

Progetto 1. Gestionale offerte di lavoro

Date	<05/12/2019>
Deliverable	Deliverable 1 – Milestone
Team (Name)	Software Solutions

Team Members		
Name & Surname	Matriculation Number	E-mail address
Giorgio Morico	252629	giorgio.morico@student.univaq.it
Giovanni Spaziani	254691	giovanni.spaziani@student.univaq.it
Matteo Leoncini	254187	matteo.leoncini@student.univaq.it

Table of Contents of this deliverable

LIST OF CHALLENGING/RISKY REQUIREMENTS OR TASKS	3
A. STATO DELL'ARTE.....	4
B. RAFFINAMENTO DEI REQUISITI.....	6
B.1 REQUISITI FUNZIONALI	6
B.2 REQUISITI NON FUNZIONALI	8
B.3 SCENARI.....	8
B.4 EXCLUDED REQUIREMENTS.....	9
B.5 ASSUNZIONI.....	9
B.6 USE CASE DIAGRAMS	10
C. ARCHITETTURA SOFTWARE	11
C.1 THE STATIC VIEW OF THE SYSTEM: COMPONENT DIAGRAM	11
C.2 THE DYNAMIC VIEW OF THE SOFTWARE ARCHITECTURE: SEQUENCE DIAGRAM	13
D. DATI E LORO MODELLAZIONE.....	17
D.1 MODELLO E-R	17
D.2 MODELLO RELAZIONALE	17
E. DESIGN DECISIONS	19
F. DESIGN DI BASSO LIVELLO	20
F.1 CLASS DIAGRAM	20
G. EXPLAIN HOW THE FRS AND THE NFRS ARE SATISFIED BY DESIGN	22
H. EFFORT RECORDING.....	22
H.1 PERT	23
H.2 LOGGING	23
APPENDIX. PROTOTYPE.....	25

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
Suddivisione dashboard	27/11/2019	27/11/2019	Abbiamo deciso di creare due dashboard differenti: una per l'offerente, una per il richiedente; questo per differenziare al meglio i servizi dedicati.
Inserimento backup	24/11/2019	24/11/2019	Abbiamo deciso di utilizzare un database di backup per evitare eventuali perdite di dati.
Generazione punteggio	21/11/2019	23/11/2019	Per generare un punteggio è necessario compilare un form in fase di candidatura perché sarebbe troppo complesso generarlo da un curriculum vitae caricato come file.
Match offerte	21/11/2019	22/11/2019	Il richiedente in fase di registrazione può inserire delle preferenze di lavoro che verranno elaborate per presentargli le offerte più compatibili con il proprio profilo.
Rifiuto candidature	01/12/2019	01/12/2019	Un offerente non deve rifiutare le candidature, ma semplicemente invitare al colloquio i candidati scelti. Questo per semplificare il suo lavoro poiché potrebbero esserci centinaia di candidature.

A. Stato dell'Arte

I sistemi presi in considerazione sono:

- **Jobsora** (<https://it.jobsora.com/>): permette di cercare offerte di lavoro di tendenza, ma anche di scegliere la regione/città in cui si vuole cercare il lavoro
- **Jooble** (<https://it.jooble.org/>): permette di trovare offerte di lavoro provenienti da tutto il web. Fa visualizzare le offerte di lavoro popolari in Italia, e di analizzare tutte le richieste. Anche qui è possibile ricercare il lavoro per città. Raccoglie tutte le offerte da tutto il web, ed inoltre ha la propria Content Quality Team che controlla tutte le offerte di lavoro per proteggere i candidati dalle truffe.
- **Monster** (<https://www.monster.it/>): permette di cercare offerte di lavoro scrivendo alcune parole chiavi, e selezionando il Paese nella quale si desidera lavorare. È disponibile anche un'app per questo sito.
- **Indeed** (<https://it.indeed.com/>): permette di cercare offerte scegliendo la professione che si desidera fare, una parola chiave, oppure un'azienda. Si può scegliere dove si intende lavorare; vastità delle opportunità in qualsiasi settore;
- **Infojobs** (<https://www.infojobs.it/>): permette di cercare offerte in base alla posizione, all'azienda, oppure tramite parola chiave. Ma, a differenza degli altri siti citati in precedenza, Infojobs ti elenca le aziende che assumono, ti fa scoprire consigli e novità del mondo del lavoro, tramite dei corsi di formazione, ti permette di acquisire nuove competenze. Tramite i filtri, puoi scegliere un lavoro a tempo indeterminato, part-time, stage/internship, oppure un lavoro da vari stipendi mensili ed annuali. Ti permette di trovare lavoro vicino casa, e ti elenca le categorie professionali con più opportunità. Ci sono oltre 90'000 aziende registrate. È disponibile un'app per questo sito.
- **Trovalavoro** (<https://lavoro.corriere.it/>): permette di cercare lavoro (anche stagionale) tramite una parola chiave o un ruolo, e tramite Regione o Provincia. Elenca le aziende che assumono; e per ogni azienda, ti elenca i tipi di lavoro che si possono trovare.
- **Cerco lavoro** (<https://www.cercolavoro.com/>): permette di cercare lavoro, sfogliando tra i vari settori, inserendo la città di preferenza. Elenca le offerte di lavoro (anche divise per province italiane) classificate per Funzione e Mansione. Ci sono più di 50'000 aziende registrate e più di 55'000 offerte di lavoro presenti.
- **Adzuna** (<https://www.adzuna.it/>): probabilmente è il sito più esaustivo tra quelli sopra elencati. Tramite una ricerca di lavoro avanzata, ti permette di effettuare appunto delle ricerche scegliendo tra i seguenti punti:
 - Si possono immettere delle parole di riferimento per effettuare la ricerca.
 - Si possono escludere delle parole a scelta per poter effettuare una ricerca più mirata.
 - Si può cercare il nome di un'azienda, se essa è presente.
 - Si può selezionare lo stipendio che un'utente vorrebbe guadagnare.
 - Si può selezionare il tipo di contratto che si ha intenzione di firmare.
 - Si può selezionare un lavoro che richiede un impegno a tempo pieno oppure part time.
 - È possibile scegliere la località.

- È possibile selezionare il tempo di inserimento dell'annuncio.
- **LinkedIn** (<https://it.linkedin.com/>): LinkedIn è un social definibile come Facebook ma in ambito professionale. lo scopo di LinkedIn è quello di consentire di creare agli utenti registrati una rete di conoscenze in ambito lavorativo. Ecco quindi che la sua settorialità lo rendono un social molto appetibile non solo agli imprenditori ma a chiunque cerchi un lavoro o i professionisti che vogliano far conoscere la propria attività. Per utilizzare al meglio il proprio profilo LinkedIn, bisogna:
 - **Inserire le parole chiave nel sommario del profilo.**
 - **Una volta iscritti ad un Gruppo, cercare di essere partecipi alle discussioni.** È una cosa che richiede un certo impegno, ma partecipare alle discussioni è un modo per accrescere la nostra popolarità e poter scambiare idee ed opinioni con altri professionisti.
 - **Usare i gruppi al solo scopo di fare pubblicità a se stessi.** Nei gruppi è vietato fare pubblicità esplicita, ma possiamo ugualmente fare pubblicità indirettamente. Come? Magari rispondendo a domande, in questo modo mettiamo in mostra le nostre conoscenze facendoci pubblicità implicitamente.
 - **LinkedIn ci dà la possibilità di personalizzare l'url.** Quindi inseriamo nell'url del nostro profilo il nostro nome e cognome in questo modo saremo maggiormente visibili e avremo maggiori probabilità di essere contattati.

B. Raffinamento dei Requisiti

B.1 Requisiti funzionali

1. **Offerente:**

- 1.1. Deve potersi registrare (priorità: alta, complessità: bassa)
 - 1.1.1. Deve inserire i dati personali (nome, cognome, ecc...)
 - 1.1.2. Deve specificare un recapito telefonico
 - 1.1.3. Deve specificare la sede del lavoro
- 1.2. Deve poter effettuare il login (priorità: alta, complessità: bassa)
- 1.3. Deve poter effettuare il logout (priorità: alta, complessità: bassa)
- 1.4. Deve poter modificare le informazioni del profilo (priorità: alta, complessità: media)
- 1.5. Deve poter inserire le offerte di lavoro (priorità: alta, complessità: bassa)
 - 1.5.1. Deve poter scegliere la categoria di impiego
 - 1.5.2. Deve specificare le competenze necessarie alla candidatura
 - 1.5.3. Deve specificare il tipo di contratto (full-time o part-time)
- 1.6. Deve poter visualizzare la lista delle candidature per una determinata offerta di lavoro (priorità: alta, complessità: bassa)
- 1.7. Deve poter contattare il/i candidato/i scelto/i (priorità: alta, complessità: media)
 - 1.7.1. Deve poter invitare il candidato ad un colloquio
- 1.8. Deve poter eliminare le offerte di lavoro inserite (priorità: media, complessità: bassa)
- 1.9. Deve poter modificare le offerte di lavoro inserite (priorità: alta, complessità: bassa)
- 1.10. Deve poter cercare un richiedente all'interno del sistema (priorità: alta, complessità: media)

2. **Richiedente:**

- 2.1. Deve potersi registrare (priorità: alta, complessità: bassa)
 - 2.1.1. Deve inserire i dati personali (Nome, cognome, età, indirizzo, telefono ecc...)
 - 2.1.2. Deve inserire il titolo/i di studio
 - 2.1.3. Se studente, deve inserire il corso di laurea che sta seguendo
- 2.2. Deve poter effettuare il login (priorità: alta, complessità: bassa)
- 2.3. Deve poter effettuare il logout (priorità: alta, complessità: bassa)
- 2.4. Deve poter modificare il profilo (priorità: alta, complessità: media)
- 2.5. Deve specificare le preferenze di lavoro (priorità: bassa, complessità: bassa)
- 2.6. Deve poter caricare il suo curriculum vitae (priorità: media, complessità: bassa)

- 2.7. Deve poter visualizzare le offerte di lavoro (priorità: alta, complessità: bassa)
 - 2.7.1. Deve poter scegliere la categoria da visualizzare
- 2.8. Deve potersi candidare ad una o più offerte di lavoro (priorità: alta, complessità: media)
- 2.9. Deve poter visualizzare la lista delle candidature effettuate (priorità: bassa, complessità: bassa)
- 2.10. Deve poter eliminare una o più candidature effettuate (priorità: media, complessità: media)
- 2.11. Deve poter accettare gli inviti ricevuti per un colloquio (priorità: alta, complessità: media)
- 3. Il sistema deve categorizzare le offerte (priorità: alta, complessità: media)
- 4. Il sistema deve presentare al richiedente le offerte di lavoro più pertinenti alle informazioni presenti nel proprio profilo (priorità: media, complessità: alta)
- 5. Il sistema deve generare un punteggio in base ad ogni candidatura effettuata dai richiedenti (priorità: alta, complessità: alta)
- 6. Il sistema deve presentare agli offerenti i candidati migliori per le proprio offerte inserite (priorità: media, complessità: bassa)
- 7. Deve inviare notifiche all'utente (priorità: alta, complessità: media)

NOTE:

Requisito 2.6: il richiedente può caricare il suo curriculum in formato pdf che verrà visualizzato sul proprio profilo.

Requisito 2.8: quando un richiedente si candida ad un'offerta di lavoro, dovrà compilare dei campi che lo guideranno alla costruzione di una specie di curriculum vitae. Questo è necessario per il calcolo del punteggio, perché non sarebbe possibile calcolarlo se il curriculum viene caricato in formato pdf

Requisito 5: il punteggio viene calcolato in base ai seguenti criteri in ordine di importanza:

- 1. Esperienze di lavoro nell'ambito (anni);
- 2. Titolo di studio;
- 3. Conoscenza lingue (se richiesta);
- 4. Età;

Requisito 7: le notifiche vengono inviate:

- 1. All'offerente dopo la ricezione di una candidatura;
- 2. Al richiedente dopo l'invito ad un colloquio;
- 3. Al richiedente quando un'offerta alla quale si era candidato è stata eliminata;
- 4. All'offerente quando il candidato scelto accetta l'invito ad un colloquio;

B.2 Requisiti non funzionali

1. È necessario avere un account personale per operare sul sistema (es: effettuare modifiche, inserimenti e candidarsi ad offerte di lavoro)
2. Un’offerta può essere modificata o eliminata soltanto da colui che l’ha pubblicata
3. È necessaria una mail valida per confermare la registrazione
4. Il sistema deve rispettare il Regolamento Europeo in Materia di Protezione dei Dati Personali (UE) 2016/679 (GDPR) relativo alla protezione delle persone fisiche per quanto riguarda il trattamento dei dati personali (**privacy**)
5. Il sistema deve avere un’interfaccia semplice da utilizzare e con un buon aspetto grafico. (**usability**)
6. Il sistema deve imporre una password sicura, che sia alfanumerica e con un minimo di caratteri prefissati. (**security**)
7. Il sistema deve garantire l’operatività 24 ore su 24, può prevedere un sistema con funzionalità ridotte che permetta l’usabilità in caso di guasti o manutenzione. (**fault tolerance**)
8. Il sistema deve prevedere un backup dei dati per prevenire perdite in caso di guasti. (**backup**)
9. Il sistema deve avere tempi di risposta brevi, sia in fase di login, sia nella pubblicazione delle offerte di lavoro, sia nella comunicazione tra utenti. (**response time**)
10. Il sistema deve garantire che le offerte pubblicate rispettino le normative (es. senza contenuti offensivi o linguaggio scurrile).

B.3 Scenari

Ricerca filtrata

Un utente può fare una ricerca delle offerte di lavoro, opzionalmente può filtrare la ricerca in base a dei parametri (es. categoria, luogo). Verrà mostrata una lista delle offerte di lavoro che soddisfano la ricerca. L’utente non ha bisogno di effettuare il log-in per poter fare delle ricerche nel sistema.

Storico accessi

Quando un utente effettua il login, nel database viene registrato l’accesso (specificando l’utente e il timestamp). Può effettuare l’accesso sia un utente richiedente che un utente offerente.

Notifica invito al colloquio

Quando un richiedente riceve un’offerta di lavoro, il sistema deve garantire che esso riceva l’invito al colloquio. Deve notificare l’invito al richiedente via mail e tramite la sezione notifiche nel sistema.

Mancata conferma di partecipazione al colloquio

Quando un richiedente riceve un invito ad un colloquio, esso deve confermare la volontà di presentarsi ad esso. Una ulteriore notifica verrà inviata dal sistema 24h dopo la prima. Nel caso in cui l'utente non confermi la presenza, l'offerente sarà libero di annullare l'invito ed eventualmente contattare altri candidati.

B.4 Excluded Requirements

Gestione colloqui sul sistema: non appena un richiedente conferma la partecipazione ad un colloquio, eventuali altri contatti tra le parti avvengono al di fuori del sistema. Questo perché non è previsto un mezzo di comunicazione interno al sistema che consente l'interazione tra due utenti (es. chat), oltre al meccanismo di notifiche.

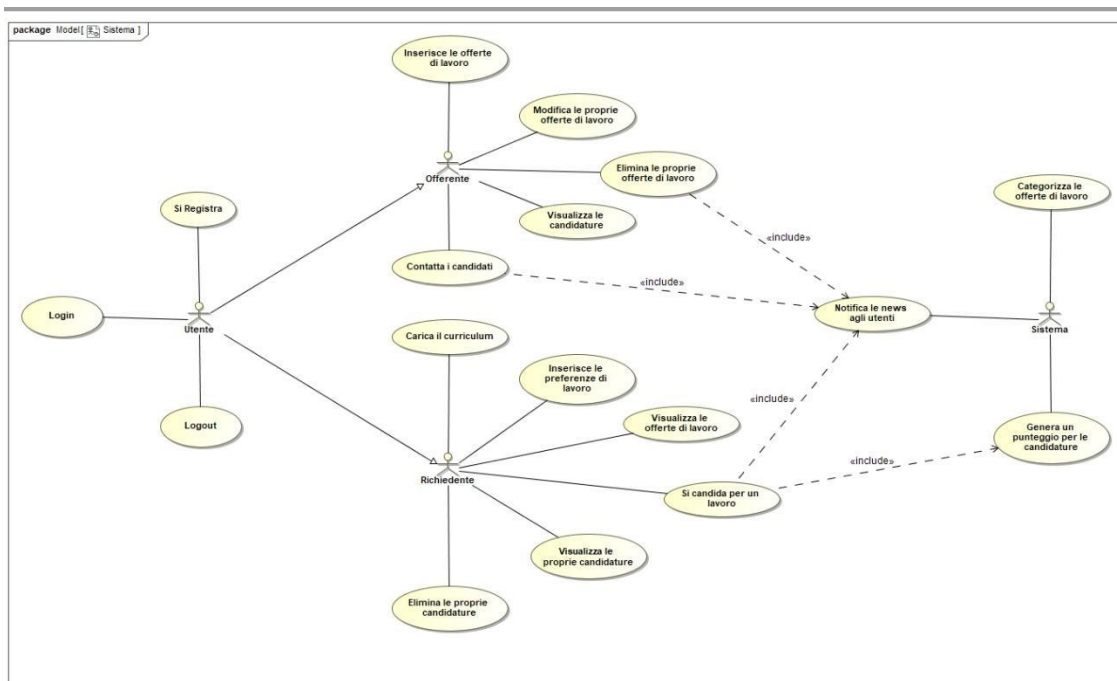
Assistenza: Per la prima implementazione del sistema non è prevista una chat di assistenza per gli utenti. Per rispettare i tempi previsti di consegna e, non essendo una funzionalità richiesta, pur essendo utile per gli utenti, verrà implementata in versioni future.

B.5 Assunzioni

Assumiamo che non sia necessario rifiutare una candidatura, l'utente offerente dovrà solo accettare ed invitare al colloquio i candidati scelti. Abbiamo fatto questa scelta perché un'azienda può ricevere centinaia di candidature. In questo modo, grazie anche alla generazione del punteggio per i candidati, il lavoro dell'offerente è molto più rapido e semplice.

Assumiamo che un offerente, nel caso sia un'azienda, avrà tanti account quante sono le sedi, anche se questo può portare a contattare lo stesso contatto più volte nel caso in cui esso rappresenti più sedi (meglio contattare più volte che perdere il contatto con qualche sede).

B.6 Use Case Diagrams



Il Sistema prevede quattro attori:

- *Utente*: colui che si registra al sistema attraverso una UI. In fase di registrazione deve scegliere il tipo di account che vuole creare: *offerente* (azienda, docente) o *richiedente* (studente, laureato in cerca di lavoro).
- *Offerente*: è un account che può appartenere ad un docente o ad un'azienda con il quale puoi pubblicare offerte di lavoro ed invitare i candidati ad un colloquio.
- *Richiedente*: è un account appartenente ad uno studente o ad un neo-laureato in cerca di lavoro che può ricercare l'offerta di lavoro che lo soddisfa e candidarsi.
- *Elaboratore*: è automatizzato e si occupa della gestione e l'elaborazione dei dati tra cui la categorizzazione delle offerte e il calcolo del punteggio per le candidature.

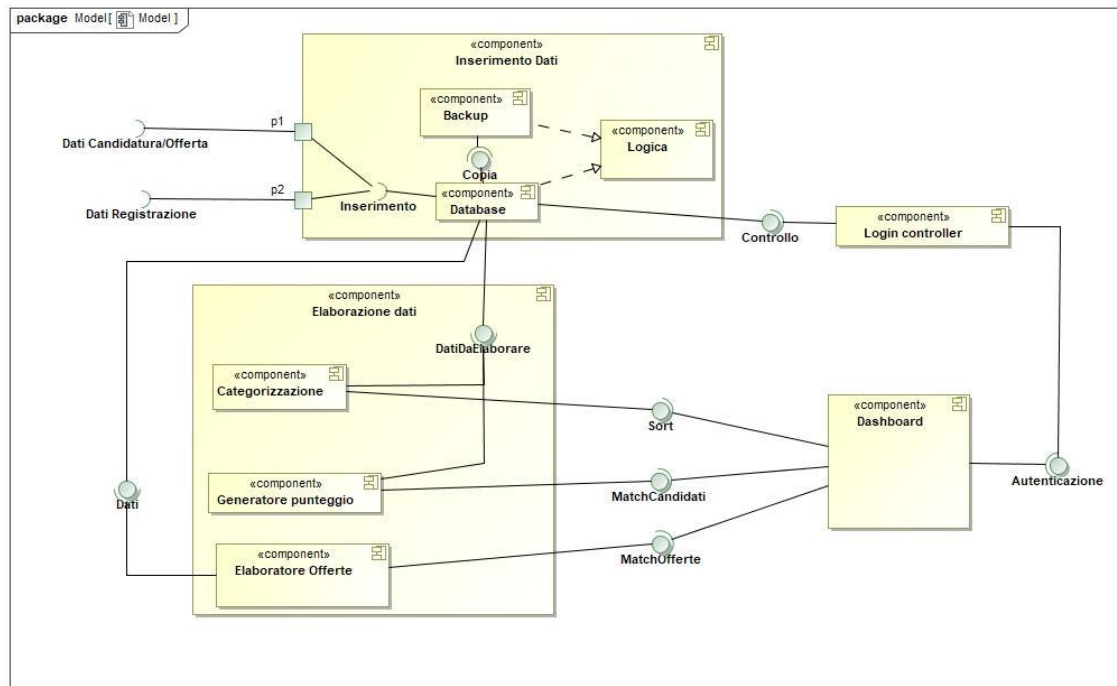
L'«include» dallo use case “si candida per un lavoro” a “genera un punteggio per le candidature” rappresenta la generazione automatica del punteggio relativo alla candidatura, nel momento in cui un richiedente si candida.

Gli use cases connessi attraverso l'«include» allo use case “notifica le news agli utenti” indicano le azioni che comportano l'invio di una notifica:

- Quando un offerente elimina una propria offerta di lavoro alla quale si era candidato un richiedente (che non è stato scelto), questo gli viene notificato.
- Quando un richiedente si candida per un lavoro, viene inviata una notifica all'offerente che ha inserito l'offerta.
- Quando un offerente sceglie di contattare un candidato e di invitarlo ad un colloquio, questo viene notificato al candidato scelto.

C. Architettura Software

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



La nostra architettura è composta dai seguenti componenti:

- **Inserimento Dati:** questo componente fa riferimento alla parte di immagazzinamento dei dati; al suo interno ha tre componenti:

Database: componente che riceve i dati inseriti dall'utente relativi alla registrazione e all'inserimento di candidature e offerte tramite l'interfaccia "Inserimento".

Backup: componente utilizzato per effettuare un backup dei dati, che riceve dal database tramite l'interfaccia "Copia".

Logica: componente che gestisce la logica di comunicazione tra database e backup.

- **Elaborazione Dati:** questo componente rappresenta la parte computazionale del sistema; è formato a sua volta da tre componenti:

Categorizzazione: componente che utilizza le informazioni relative alle offerte di lavoro per effettuare una categorizzazione.

Generatore Punteggio: componente che genera un punteggio per ogni candidatura in base ai dati inseriti dal richiedente, facendo riferimento all'offerta alla quale si è candidato. Il punteggio viene generato basandosi sulle seguenti informazioni: Esperienze di lavoro nell'ambito (anni), titolo di studio, conoscenza lingue (se richiesta) ed età.

Elaboratore Offerte: componente che elabora i dati relativi alle offerte inserite e alle preferenze stabilite dal richiedente fornendo una lista di offerte che fa match al meglio con le preferenze dell'utente.

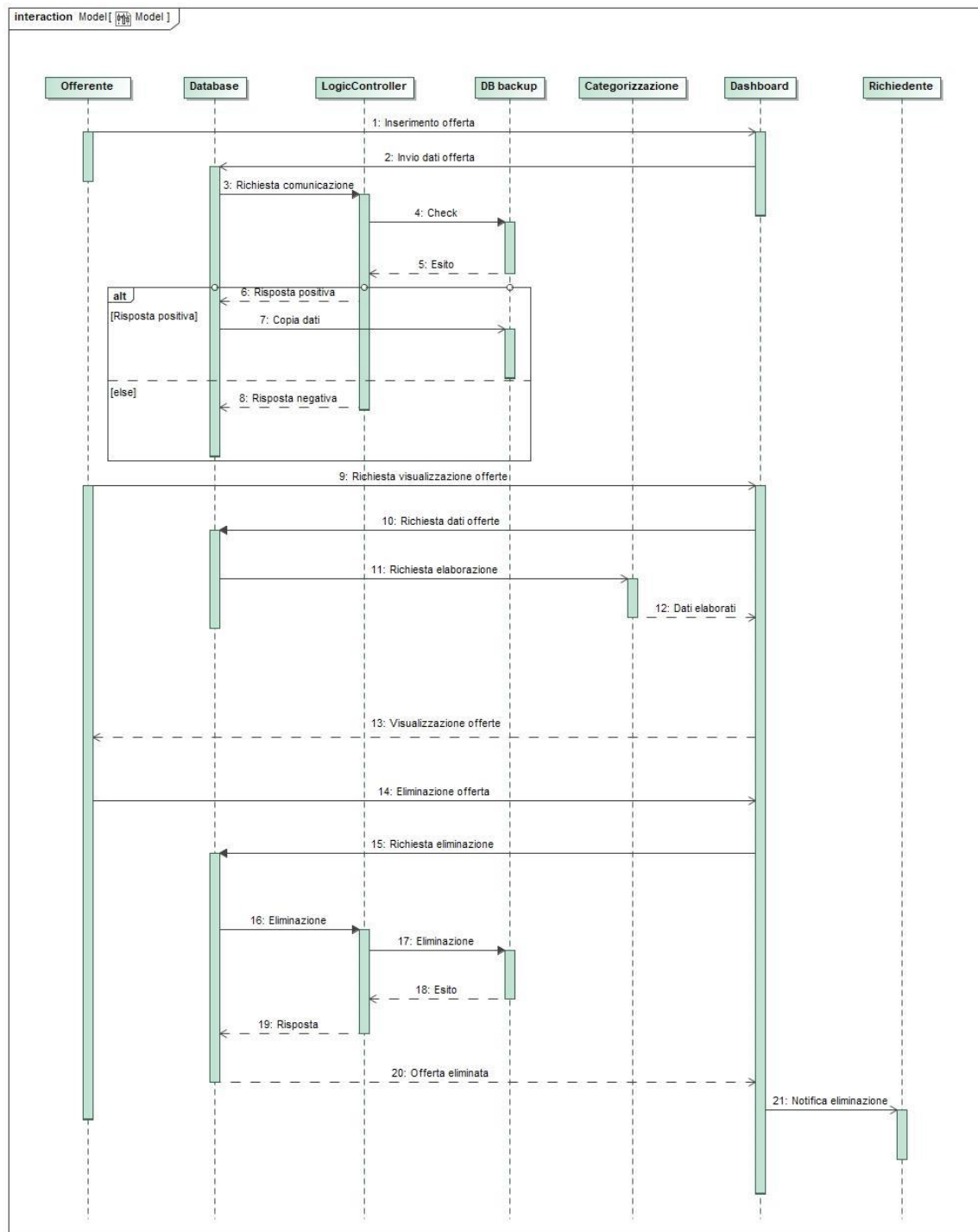
I componenti "Categorizzazione" e "Generatore punteggio" ricevono dati dal database tramite l'interfaccia "DatiDaElaborare", mentre il componente "Elaborazione offerte" riceve le informazioni dall'interfaccia "Dati" (preferenze utente e offerte).

- **Login Controller**: componente che effettua un controllo nella fase di login: quando un utente vuole accedere al sistema, questo componente verifica la presenza delle credenziali di accesso nel database tramite l'interfaccia "Controllo". L'esito del controllo viene mostrato tramite l'interfaccia "Autenticazione": se l'esito è positivo, l'utente può accedere alla dashboard, altrimenti verrà mostrato un messaggio di errore.
- **Dashboard**: questo componente rappresenta la parte del sistema dedicata alla visualizzazione dei dati e alla fornitura dei servizi. La dashboard utilizza tre interfacce, una per ogni servizio offerto, tramite le quali riceve i dati necessari per lo svolgimento del relativo servizio. Le interfacce in questione sono:
 - *Sort*: utilizzata per fornire all'utente il risultato della categorizzazione dei lavori; riceve i dati dal component "Categorizzazione".
 - *MatchCandidati*: ha il compito di mostrare all'offerente i candidati che fanno match con le proprie offerte di lavoro; riceve i dati dal component "Generatore punteggio".

- *MatchOfferte*: utilizzata per mostrare al richiedente le offerte che fanno match con le preferenze inserite nel proprio profilo; riceve i dati dal component “Elaboratore offerte”.

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

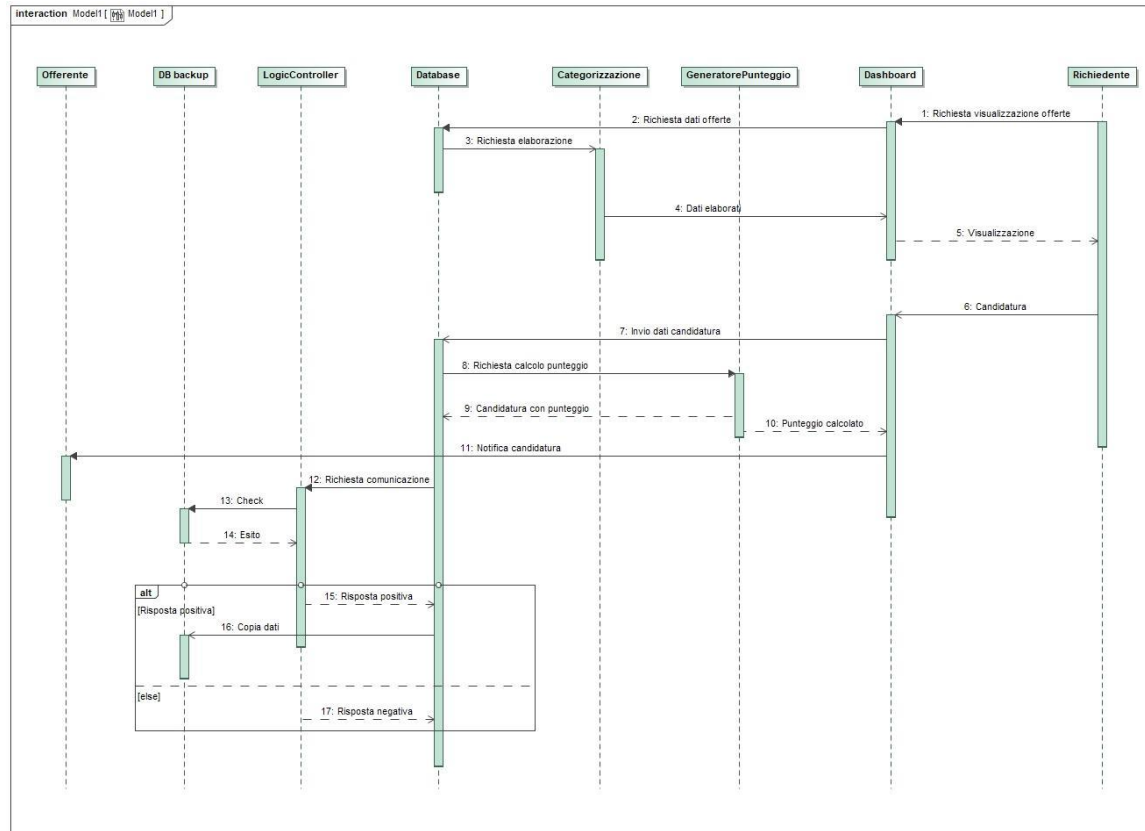
Abbiamo deciso di rappresentare con i sequence diagrams i processi più importanti del sistema: inserimento ed eliminazione offerta, candidatura ed invito al colloquio.

Inserimento ed eliminazione offerta

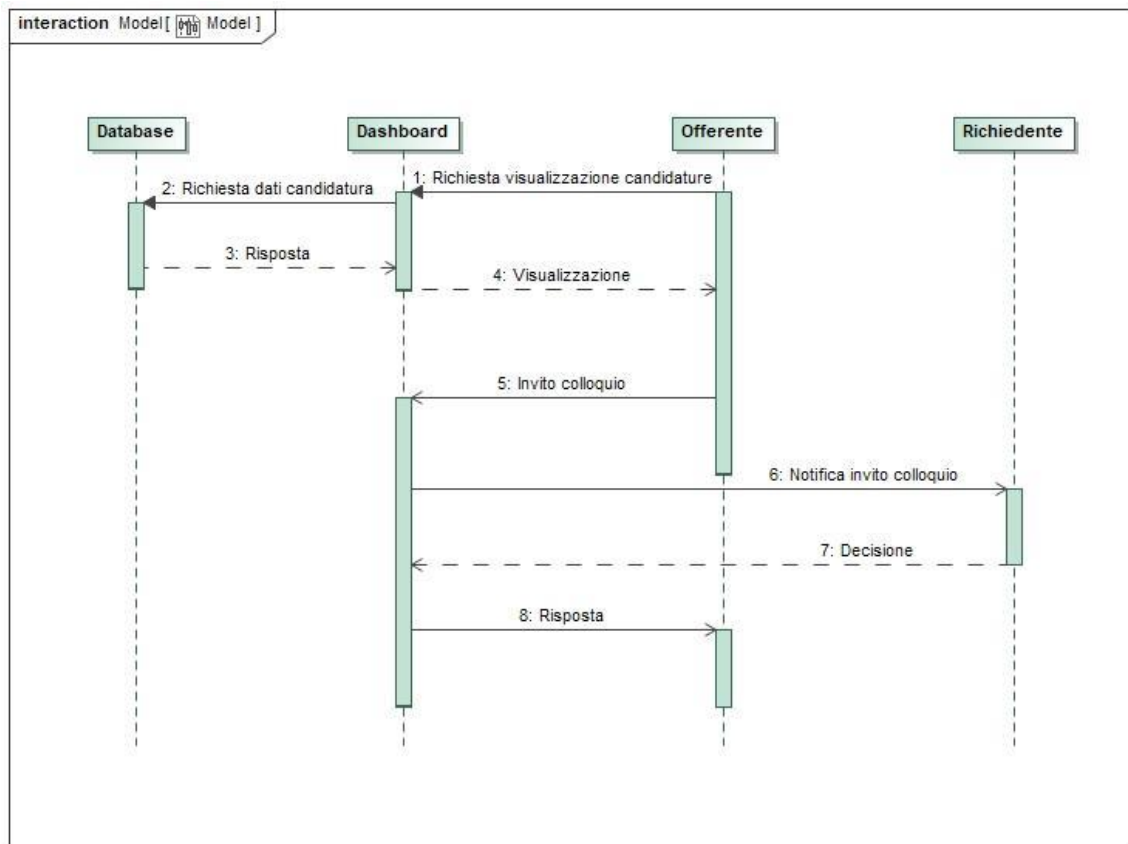
Quando un offerente inserisce un'offerta viene salvata nel database e, comunicando con il logic controller, viene copiata nel database di backup (a meno di eventuali errori).

Nel caso in cui l'offerente volesse eliminarla, può visualizzare le proprie offerte attraverso la dashboard e richiederne l'eliminazione; questo porta alla cancellazione dell'offerta dai database e la notificazione dell'eliminazione a tutti i candidati che non sono stati scelti.

Candidatura



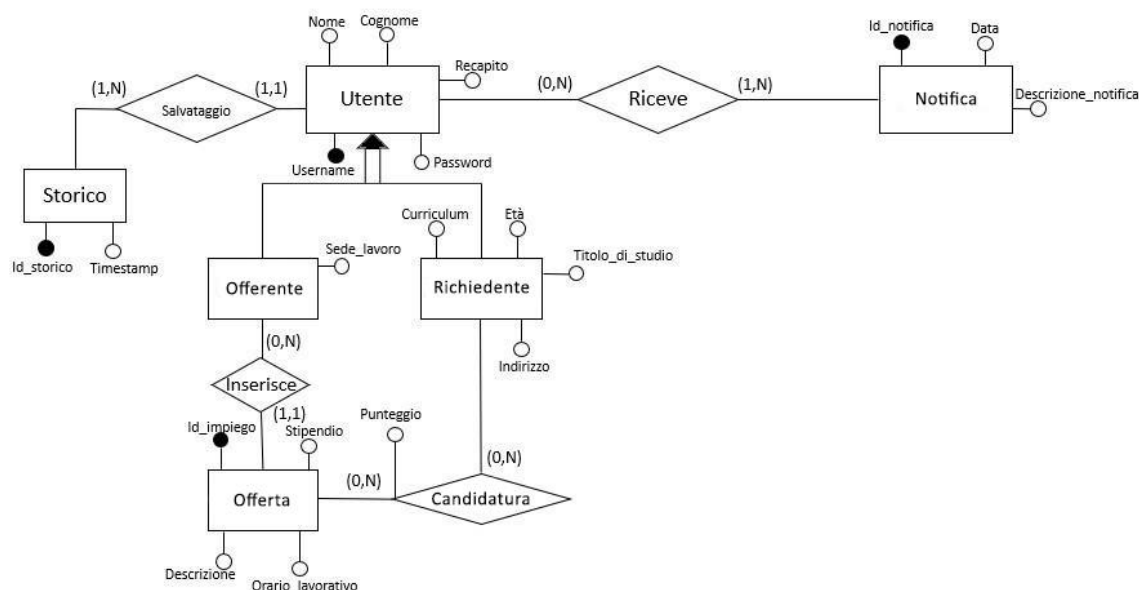
Quando un richiedente vuole candidarsi ad un'offerta di lavoro può sceglierne una o più tra quelle visualizzate ed inviare la candidatura che, dopo essergli stato assegnato un punteggio, viene inserita nel database e copiata nel DB di backup (passando attraverso il logic controller). Una notifica di candidatura viene inviata all'offerente.

Invito al colloquio

Un'offerente può visualizzare le candidature alle proprie offerte attraverso la dashboard, scelto uno o più candidati, può invitarli ad un colloquio; una notifica di invito viene inviata al richiedente che dovrà confermare o meno la partecipazione.

D. Dati e loro modellazione

D.1 Modello E-R



D.2 Modello Relazionale

Offerente (**Username**, Password, Nome, Cognome, Recapito, Sede_lavoro, *id_storico*)

Richiedente (**Username**, Password, Nome, Cognome, Recapito, Curriculum, Età, Titolo_di_studio, Indirizzo, *id_storico*)

Offerta (**Id_impiego**, Competenze, Descrizione, Orario_lavorativo, Stipendio)

Candidatura (**Username**, **Id_impiego**, Punteggio)

Notifica (**Id_notifica**, Username, Data, Descrizione_notifica)

Storico (**Id_storico**, Timestamp)

LEGENDA

In **grassetto** sono indicate le chiavi primarie;

In *corsivo* sono indicate le chiavi ereditate;

In **grassetto-corsivo** sono indicate le chiavi ereditate che sono anche chiavi primarie.

Descrizione

Per la costruzione del modello ER sono state individuate sei entità:

- Offerente;
- Richiedente;
- Offerta;
- Candidatura;
- Notifica;
- Storico

Nella traduzione dal modello E-R al modello relazionale, la generalizzazione di utente viene rappresentata effettuando un collasso verso il basso: l'entità *utente* viene eliminata e le entità *offerente* e *richiedente* ereditano gli attributi di *utente* (Username, Password, Nome, Cognome, Recapito, Indirizzo).

Di seguito vengono descritte le associazioni tra le entità:

- Ogni *utente* (che può essere quindi o un offerente o un richiedente) può ricevere più *notifiche* sul suo account, a sua volta una *notifica* può essere ricevuta da più utenti.
- Ogni *offerente* può pubblicare più offerte (non è obbligato a pubblicarne), al contrario ogni *offerta* è pubblicata da un *offerente*.
- Ogni *richiedente* può candidarsi a più offerte; ad ogni *offerta* si candidano più *richiedenti*.
- Gli accessi di un *utente* vengono salvati in uno *storico*; nello *storico* sono salvati gli accessi di più *utenti*.

Il Database prevede l'acquisizione di dati dall'esterno:

- La lista di scuole e università;
- Titoli di studio;
- Nazioni e città;

E. Design Decisions

Modalità di inserimento candidatura e curriculum

Abbiamo stabilito che un richiedente può caricare sul suo profilo il curriculum vitae in formato pdf ma dato che il curriculum è un file di tipo testuale, per motivi di complessità, in fase di candidatura bisogna compilare un form i cui campi verranno utilizzati per generare il punteggio.

Backup

Abbiamo deciso che è necessario un database di backup per prevenire eventuali perdite di dati e garantire all'utente l'integrità dei dati stessi.

Sicurezza del sistema

Per garantire la sicurezza degli utenti, si è deciso di imporre dei vincoli sulla complessità delle password: lettere maiuscole, lettere minuscole, lunghezza minima, numeri ed elementi di punteggiatura. Per proteggere i dati da modifiche indesiderate, sono stati definiti dei permessi di accesso per far sì che ogni utente acceda soltanto ai dati che può visualizzare.

Dashboard differenziate

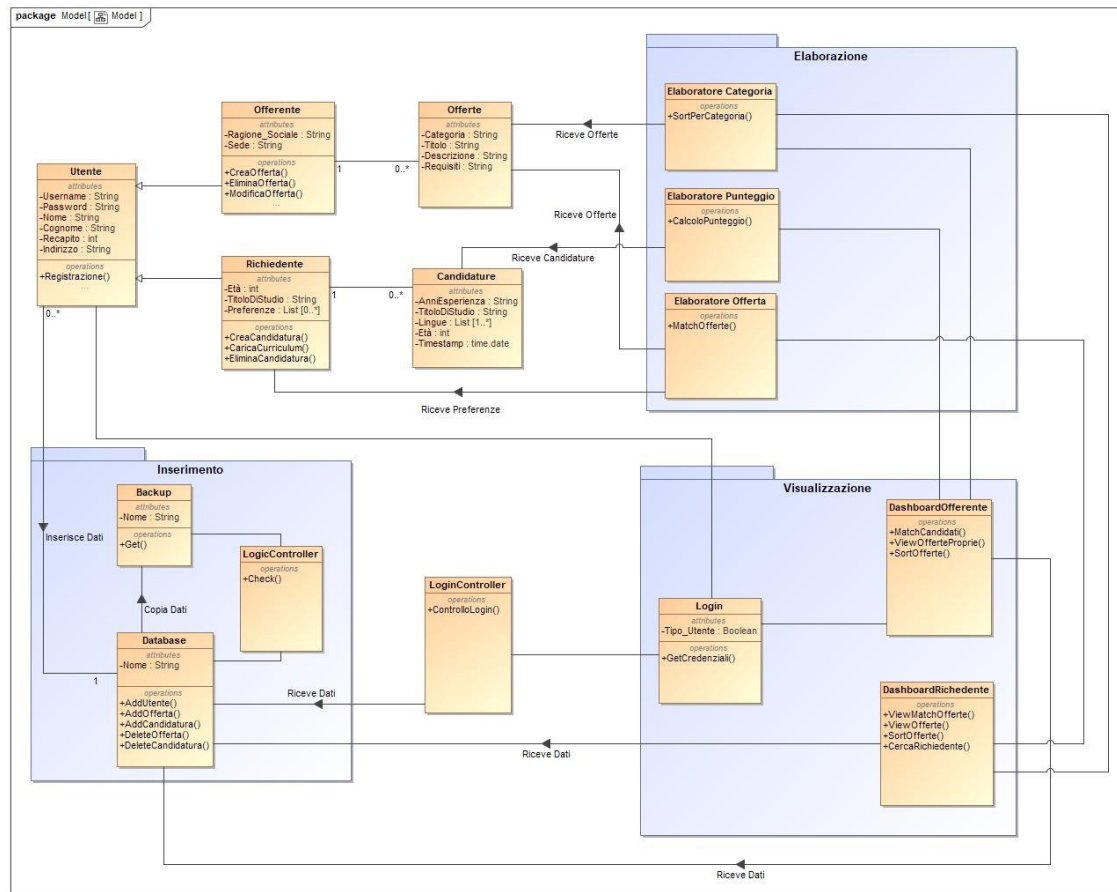
Abbiamo deciso di implementare due tipi di dashboard: una per il richiedente e una per l'offerente. Questa scelta è stata fatta per dividere i servizi in base al tipo di utente al quale sono destinati. Le due dashboard sono graficamente simili sia per facilità di implementazione sia per non avere troppe differenze tra richiedente e offerente.

Notifiche

Abbiamo inserito una sezione di notifiche per facilitare la comunicazione evitando così il solo utilizzo delle email. Le notifiche verranno utilizzate per comunicazioni riguardanti la ricezione di una candidatura, l'invito ad un colloquio, eliminazione di un'offerta alla quale un utente si era candidato e quando il candidato scelto dall'offerente accetta l'invito ad un colloquio. L'invio e la ricezione di notifiche vengono gestiti in modo automatico dal sistema.

F. Design di Basso Livello

F.1 Class Diagram



Questo class diagram è composto da tre package che contengono le classi con le quali interagiscono gli utenti e i dati inseriti da essi. La classe utente rappresenta le informazioni principali di chi utilizza il sistema e può essere di due tipi: richiedente e offerente. Per entrambe le tipologie di utente c'è una classe dedicata, alla quale è collegata una classe relativa a ciò che quell'utente inserisce ("Offerte" per l'offerente e "Candidature" per il richiedente). I tre package con i quali queste classi interagiscono sono:

- **Inserimento**: contiene le classi che rappresentano la parte di immagazzinamento dei dati, ovvero "Database", che riceve i dati inseriti dall'utente, "Backup", che rappresenta un database di backup nel quale vengono copiati i dati presenti nel database e "LogicController" che gestisce la comunicazione tra Database e Backup.

- *Elaborazione*: rappresenta la parte computazionale del sistema; è composto dalle classi che, tramite i metodi stabiliti, elaborano i dati ricevuti in input dalle classi relative all'utente, per fornire poi in output nelle rispettive dashboard i risultati delle elaborazioni.
- *Visualizzazione*: package che mette insieme le classi dedicate alla parte di visualizzazione dei dati; sono presenti due classi relative alla dashboard, differenziate in base al tipo di utente che vi accede (offerente o richiedente). La terza classe presente nel package, "Login", fa riferimento all'interfaccia di login tramite la quale l'utente accede al sistema; questa classe interagisce con il database per verificare la presenza e la correttezza dei dati inseriti. L'interazione tra le classi "Login" e "Database" viene gestita dalla classe "LoginController".

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

Un utente può essere di due tipi: offerente e richiedente. Nella fase di registrazione l'utente ha la possibilità di scegliere se registrarsi come offerente o richiedente; inserite tutte le informazioni relative alla registrazione, i dati vengono inseriti nel database tramite l'apposito collegamento. In fase di registrazione, il richiedente può specificare delle preferenze di lavoro che verranno elaborate, insieme alle offerte, per fornirgli una vista dei migliori impieghi compatibili con il suo profilo.

Il login viene gestito da un login controller il quale, una volta inseriti i dati di autenticazione, controlla se essi sono presenti nel database. Una volta effettuato il login, l'utente può inserire offerte/candidature che verranno inserite nel database specificandone l'autore; nell'inserimento di un'offerta è necessario specificare la categoria perché il sistema prevede un elaboratore che, ricevendo in input le offerte, le presenterà all'utente categorizzate.

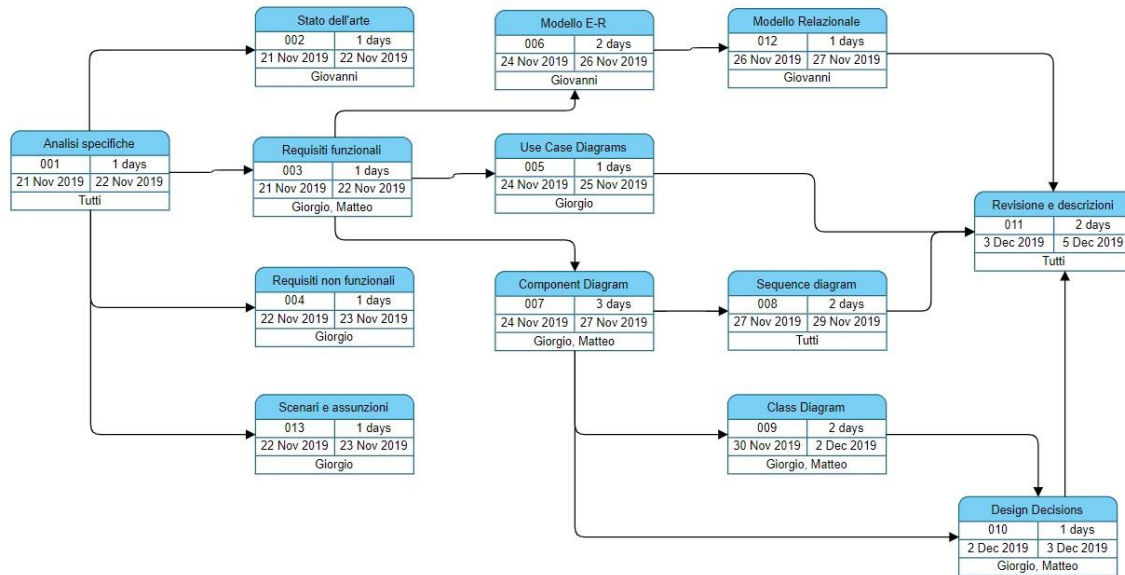
Ad ogni candidatura viene assegnato un punteggio generato da uno specifico elaboratore il quale confronta i dati della candidatura con quelli della rispettiva offerta; questo punteggio aiuta l'offerente nella scelta del candidato ideale.

Ogni utente accede alla dashboard dedicata tramite la quale utilizza i servizi a lui offerti; la dashboard include una sezione di notifiche che ha il compito di informare l'utente su determinati avvenimenti.

Per questioni di sicurezza, ogni dato inserito nel database viene copiato all'interno di un database di backup; la comunicazione tra le due parti viene gestita da un controller.

H. Effort Recording

H.1 PERT



H.2 Logging

Team (number and name): Software Solutions					
Student name: Giovanni Spaziani, Giorgio Morico, Matteo Leoncini					
Student number: 3					
Email: giovanni.spaziani@student.univag.it, giorgio.morico@student.univag, matteo.leoncini@student.univag					
When (Month/Da- y)	Time spent	Partners <i>Intervener request how many people how long working</i>	Brief Description of the performed task	Category	Sub-Category
21	11 30 min.	3	Analisi specifica e comprensione	Learning	
21	11 30 min.	3	Pert	Doing	
21	11 1h	1	Stato dell'arte	Learning	A. Stato dell'Arte
21	11 1:30 h	2	Requisiti funzionali	Doing	B. Raffinamento dei requisiti
22	11 1h	1	Requisiti non funzionali	Doing	B. Raffinamento dei requisiti
24	11 30 min.	1	Learning Use Case	Learning	B. Raffinamento dei requisiti
24	11 40 min.	1	Use Case Diagram	Learning	B. Raffinamento dei requisiti
24	11 30 min.	1	Modello E-R	Doing	D. Dati e loro modellazione
24	11 30 min.	1	Learning Component	Learning	C. Architettura Software
24	11 1:30 h	1	Component Diagram	Doing	C. Architettura Software
25	11 1:15 h	3	Revisioni	Doing	
25	11 40 min.	2	Component Diagram	Doing	C. Architettura Software
25	11 15 min.	1	Digitalizzazione Modello E-R	Doing	D. Dati e loro modellazione
25	11 15 min.	1	Modello relazionale	Doing	D. Dati e loro modellazione
25	11 1h	1	Correzione Component Diagram	Doing	C. Architettura Software
26	11 30 min.	3	Learning Sequence Diagram	Learning	C. Architettura Software
26	11 1h	2	Sequence Diagram	Doing	C. Architettura Software
27	11 30 min.	2	Learning Class Diagram	Learning	F. Design di basso livello
27	11 1h	2	Class Diagram	Doing	F. Design di basso livello
1	12 40 min.	1	Descrizione Modello E-R	Doing	D. Dati e loro modellazione
1	12 25 min.	1	Correzione Component Diagram	Doing	C. Architettura Software
1	12 25 min.	1	Descrizione Component Diagram	Doing	C. Architettura Software
1	12 15 min.	1	Aggiunta descrizione requisiti	Doing	B. Raffinamento dei requisiti
2	12 1:30 h	1	Sequence Diagram	Doing	C. Architettura Software
2	12 1h	2	Class Diagram	Doing	F. Design di basso livello
3	12 30 min.	3	Revisione lavori	Doing	
4	12 50 min.	2	Design decisions	Doing	E. Design decisions
4	12 50 min.	2	Come i requisiti sono soddisfatti dal design	Doing	G. Explain how the FRs and the NFRs are satisfied by design
5	12 1h	2	Correzione Sequence Diagram	Doing	C. Architettura Software
5	12 30 min.	1	Descrizione Class Diagram	Doing	F. Design di basso livello
5	12 15 min.	1	Descrizione Sequence Diagram	Doing	C. Architettura Software
5	12 1h	3	Revisione finale	Doing	

Ore totali: 40h 30'

Ore doing: 33h 50'

Ore learning: 6h 40'

Appendix. Prototype

<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>
