

# **“Software Engineering”**

## **Course**

### **a.a. 2019-2020**

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

# **Progetto 1**

## **Gestionale offerte di lavoro**

<b>Date</b>	<13/02/2020>
<b>Deliverable</b>	Deliverable 2
<b>Team (Name)</b>	Software Solutions

<b>Team Members</b>		
<b>Name &amp; Surname</b>	<b>Matriculation Number</b>	<b>E-mail address</b>
Giorgio Morico	252629	<a href="mailto:giorgio.morico@student.univaq.it">giorgio.morico@student.univaq.it</a>
Giovanni Spaziani	254691	<a href="mailto:giovanni.spaziani@student.univaq.it">giovanni.spaziani@student.univaq.it</a>
Matteo Leoncini	254187	<a href="mailto:matteo.leoncini@student.univaq.it">matteo.leoncini@student.univaq.it</a>

1	
---	--

# Table of Contents of this deliverable

<b>LIST OF CHALLENGING/RISKY REQUIREMENTS OR TASKS .....</b>	<b>4</b>
<b>A. STATO DELL'ARTE .....</b>	<b>6</b>
<b>B. RAFFINAMENTO DEI REQUISITI.....</b>	<b>8</b>
B.1 REQUISITI FUNZIONALI .....	8
B.2 REQUISITI NON FUNZIONALI .....	13
B.3 SCENARI.....	14
B.4 EXCLUDED REQUIREMENTS .....	17
B.5 ASSUNZIONI .....	17
B.6 USE CASE DIAGRAMS .....	18
<b>C. ARCHITETTURA SOFTWARE .....</b>	<b>19</b>
C.1 THE STATIC VIEW OF THE SYSTEM: COMPONENT DIAGRAM .....	19
C.2 THE DYNAMIC VIEW OF THE SOFTWARE ARCHITECTURE: SEQUENCE DIAGRAM .....	23
<b>D. DATI E LORO MODELLAZIONE.....</b>	<b>28</b>
D.1 MODELLO E-R.....	28
D.2 MODELLO RELAZIONALE .....	28
<b>E. DESIGN DECISIONS .....</b>	<b>31</b>
<b>F. DESIGN DI BASSO LIVELLO .....</b>	<b>33</b>
F.1 CLASS DIAGRAM.....	33
<b>G. EXPLAIN HOW THE FRS AND THE NFRS ARE SATISFIED BY DESIGN .....</b>	<b>35</b>

**H. EFFORT RECORDING..... 36**

**H.1 PERT ..... 36**

**H.2 LOGGING ..... 37**

**APPENDIX. PROTOTYPE..... 39**

**List of Challenging/Risky Requirements or Tasks**

<b>Challenging Task</b>	<b>Date the task is identified</b>	<b>Date the challenge is resolved</b>	<b>Explanation on how the challenge has been managed</b>
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.
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.
Gestione utenti nel prototipo	13/01/20	14/01/20	Nel prototipo si differenzia il tipo di utente tramite un flag, il quale indica se si tratta di un richiedente o di un offerente.
Modifica requisiti di un'offerta e relativa modifica del punteggio delle candidature	12/01/20	17/01/20	Nel caso in cui un offerente modifichi i requisiti di una determinata offerta, per tutte le candidature effettuate per quell'offerta, prima della modifica, viene ricalcolato il punteggio

Eliminazione candidature effettuate per un'offerta che viene eliminata	12/01/20	17/01/20	Quando un'offerta viene eliminata, di conseguenza vengono eliminate anche le candidature effettuate per quell'offerta
--	----------	----------	---

## A. Stato dell'Arte

I sistemi più conosciuti sul web che permettono agli utenti di trovare lavoro sono:

- **Jobsora** (<https://it.jobsora.com/>);
- **Jooble** (<https://it.jooble.org/>);
- **Monster** (<https://www.monster.it/>);
- **Indeed** (<https://it.indeed.com/>);
- **Infojobs** (<https://www.infojobs.it/>);
- **Trovolavoro** (<https://lavoro.corriere.it/>);
- **Cerco lavoro** (<https://www.cercolavoro.com/>);
- **Adzuna** (<https://www.adzuna.it/>);
- **LinkedIn** (<https://it.linkedin.com/>);

Entrando nel dettaglio di ciascuno e scoprendo come sono composti, si può notare che quasi tutti i siti (eccetto due) sono principalmente simili, eccetto alcune sottigliezze. Si parla di: '**Jobsora**', '**Jooble**', '**Monster**', '**Indeed**', '**Infojobs**', '**Trovolavoro**' e '**Cerco lavoro**'. Questi ultimi permettono agli utenti di poter cercare lavoro inserendo professione, parole chiavi, preferenze, l'azienda per la quale si vorrebbe lavorare, ed infine la Regione e la provincia dove l'utente desidererebbe lavorare. Il penultimo sito elencato, ovvero '**Adzuna**' è forse il più completo tra i siti precedentemente elencati. Esso, tramite una ricerca di lavoro avanzata, 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.

Per ultimo, ma non per ordine di importanza, c'è '**LinkedIn**'. Esso 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. Esso offre:

- **La possibilità di inserire le parole chiave nel sommario del profilo.**
- **La possibilità di creare gruppi.** Una volta iscritti, si può partecipare a delle discussioni. È un modo per accrescere la popolarità e poter scambiare idee ed opinioni con altri professionisti.
- **Linkedin ci dà la possibilità di personalizzare l'url.** Quindi inseriamo nell'url del profilo il nome e cognome in questo modo saremo maggiormente visibili e avremo maggiori probabilità di essere contattati.

Come la maggior parte dei sistemi citati, permetteremo all'utente di ricercare le offerte di lavoro inserendo parole chiavi, categoria e luogo; inoltre, è possibile cercare gli utenti (sia aziende che persone). Una volta cercate le offerte di lavoro è possibile filtrare la visualizzazione in base a stipendio e tipo di contratto (full-time, part-time). Una funzionalità interessante che vogliamo inserire nel nostro sistema è quella che consente agli utenti di creare ed iscriversi a dei gruppi, che verranno utilizzati per interagire con utenti appartenenti alla propria area di competenza. I servizi analizzati che non implementeremo sono:

- la possibilità di escludere delle parole per effettuare una ricerca più mirata perché crediamo che le modalità che implementeremo sono sufficienti per trovare ciò che l'utente cerca.
- La possibilità di personalizzare l'url perché non riteniamo sia un servizio fondamentale per l'utente.

## B. Raffinamento dei Requisiti

### B.1 Requisiti funzionali

#### 1. Offerente:

- 1.1. Deve potersi registrare tramite un form nel quale inserisce: username, password, dati personali (nome, cognome, recapito telefonico e indirizzo), ragione sociale (nel caso in cui l'offerente è un'azienda) e la sede di lavoro (*Priorità: alta, Complessità: bassa*)
- 1.2. Deve poter effettuare il login tramite un apposito form nel quale inserisce username e password specificati durante la registrazione (*Priorità: alta, Complessità: bassa*)
- 1.3. Deve poter effettuare il logout (*Priorità: alta, Complessità: bassa*)
- 1.4. Deve poter modificare le informazioni del proprio profilo (*Priorità: alta, Complessità: media*)
- 1.5. Deve poter inserire le offerte di lavoro utilizzando un form nel quale specifica: titolo offerta, categoria di impiego, descrizione, competenze necessarie alla candidatura, stipendio e tipo di contratto (es: part-time, full-time) (*Priorità: alta, Complessità: bassa*)
- 1.6. Selezionando una specifica offerta da esso inserita, l'offerente deve poter visualizzare la lista di coloro che si sono candidati a quell'offerta (*Priorità: alta, Complessità: bassa*)
- 1.7. Deve poter contattare il/i candidato/i scelto/i per invitarlo al colloquio. L'offerente, dalla lista dei candidati per una determinata offerta, utilizza un bottone "Invita" per invitare il/i candidato/i di interesse; a questo punto comparirà un form nel quale l'offerente stabilisce orario e luogo del colloquio. Una volta confermato l'inserimento, verrà inviata una mail ed una notifica al/ai richiedente/i scelto/i. (*Priorità: alta, Complessità: media*)
- 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 utente all'interno del sistema tramite una barra di ricerca nella quale specifica nome e cognome della persona (*Priorità: alta, Complessità: media*)
- 1.11. Deve poter cercare le offerte di lavoro presenti nel sistema tramite una barra di ricerca inserendo: parole chiave, tipo di contratto e luogo (*Priorità: alta, Complessità: bassa*)
- 1.12. Deve poter utilizzare una chat per comunicare con gli altri utenti del sistema (*Priorità: media, Complessità: alta*)
- 1.13. Deve poter creare un gruppo tramite un form specificando: nome, categoria e descrizione. (*Priorità: media, Complessità: media*)
- 1.14. Deve potersi iscrivere ad un gruppo, tramite un apposito bottone "Iscriviti al gruppo" (*Priorità: media, Complessità: bassa*)
- 1.15. Deve poter invitare nuovi utenti ad unirsi ad un gruppo al quale esso è iscritto (*Priorità: media, Complessità: media*)

## 2. Richiedente:

- 2.1. Deve potersi registrare tramite un form nel quale inserisce: username, password, dati personali (nome, cognome, recapito telefonico e indirizzo), età, titolo di studio, preferenze lavorative ed altri campi [Vedi Note]. Se è uno studente, deve inserire il corso di laurea che sta seguendo. (*Priorità: alta, Complessità: bassa*)
- 2.2. Deve poter effettuare il login tramite un apposito form nel quale inserisce username e password specificati durante la registrazione (*Priorità: alta, Complessità: bassa*)
- 2.3. Deve poter effettuare il logout (*Priorità: alta, Complessità: bassa*)
- 2.4. Deve poter modificare le informazioni del proprio profilo (*Priorità: alta, Complessità: media*)
- 2.5. Deve poter caricare il curriculum vitae sul proprio profilo [Vedi Note] (*Priorità: media, Complessità: bassa*)
- 2.6. Deve poter cercare le offerte di lavoro presenti nel sistema tramite una barra di ricerca inserendo: parole chiave, tipo di contratto e luogo (*Priorità: alta, Complessità: bassa*)
- 2.7. Deve poter cercare un utente all'interno del sistema tramite una barra di ricerca nella quale specifica nome e cognome della persona (*Priorità: alta, Complessità: bassa*)

- 2.8. Deve poter visualizzare le offerte di lavoro ricercate oltre a quelle presentate dal sistema facendo il matching con le proprie preferenze. (*Priorità: alta, Complessità: bassa*)
- 2.9. Deve potersi candidare ad una o più offerte di lavoro. Per ogni offerta di lavoro visualizzata, è presente un bottone “Candidati” tramite il quale il richiedente invia una candidatura alla quale verrà assegnato un punteggio (*Priorità: alta, Complessità: media*)
- 2.10. Deve poter visualizzare la lista delle candidature effettuate (*Priorità: bassa, Complessità: bassa*)
- 2.11. Deve poter eliminare una o più candidature effettuate (*Priorità: media, Complessità: media*)
- 2.12. Deve poter accettare gli inviti ricevuti per un colloquio comunicando l’esito tramite la dashboard utilizzando due bottoni: “Accetta” e “Rifiuta”. In base alla scelta fatta, il candidato utilizza il bottone appropriato. (*Priorità: alta, Complessità: media*)
- 2.13. Deve poter utilizzare una chat per comunicare con gli altri utenti del sistema (*Priorità: media, Complessità: alta*)
- 2.14. Deve poter creare un gruppo tramite un form specificando: nome, categoria e descrizione. (*Priorità: media, Complessità: media*)
- 2.15. Deve potersi iscrivere ad un gruppo, tramite un apposito bottone “Iscriviti al gruppo” (*Priorità: media, Complessità: bassa*)
- 2.16. Deve poter invitare nuovi utenti ad unirsi ad un gruppo al quale esso è iscritto (*Priorità: media, Complessità: media*)
3. **Utente non registrato:**
  - 3.1. Deve poter cercare un utente all’interno del sistema tramite una barra di ricerca nella quale specifica nome, cognome (*Priorità: alta, Complessità: bassa*)
  - 3.2. Deve poter cercare le offerte di lavoro presenti nel sistema tramite una barra di ricerca inserendo: parole chiave, tipo di contratto e luogo (*Priorità: alta, Complessità: bassa*)
4. Il sistema deve categorizzare le offerte (*Priorità: alta, Complessità: media*)
5. Il sistema deve presentare al richiedente le offerte di lavoro più pertinenti alle informazioni relative alle preferenze lavorative presenti nel proprio profilo (*Priorità: media, Complessità: alta*)

6. Il sistema deve generare un punteggio in base ad ogni candidatura effettuata dai richiedenti. Per la generazione del punteggio, vengono prese le informazioni presenti sul profilo del richiedente che vengono confrontate con quelle relative all'offerta; il punteggio viene generato tenendo conto di alcuni criteri [Vedi Note] (*Priorità: alta, Complessità: alta*)
7. Il sistema deve presentare agli offerenti i candidati migliori per le proprio offerte inserite. Quando un offerente apre un'offerta da lui inserita, visualizza la lista di coloro che si sono candidati a quell'offerta in ordine di punteggio (*Priorità: media, Complessità: bassa*)
8. Deve inviare notifiche all'utente [Vedi Note] (*Priorità: alta, Complessità: media*)
9. Il sistema deve fornire agli utenti una chat per permettergli di comunicare (*Priorità: media, Complessità: alta*)

**NOTE:**

*Requisito 2.1:* quando un richiedente si registra, 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 2.5:* il richiedente può caricare il suo curriculum in formato pdf che verrà visualizzato sul proprio profilo.

*Requisito 6:* 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;

Il punteggio sarà espresso tramite una valutazione da 1 a 5.

*Requisito 8:* 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. Per effettuare il login, è necessario inserire username e password stabiliti durante la registrazione
5. 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**)
6. Il sistema deve avere un’interfaccia semplice da utilizzare e con un buon aspetto grafico per rendere l’esperienza dell’utente migliore possibile (**Usability**)
7. Il sistema deve imporre una password sicura, che sia alfanumerica e con un minimo di caratteri prefissati per garantire un buon livello di sicurezza (**Security**)
8. 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**)
9. Il sistema deve prevedere un backup dei dati per prevenire perdite in caso di guasti e per mantenere l’operatività del sistema (**Backup**)
10. 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**)
11. Il sistema deve garantire che le offerte pubblicate rispettino le normative (es. senza contenuti offensivi o linguaggio scurrile)
12. Il sistema deve permettere l’integrazione con altri sistemi sia dal punto di vista dell’implementazione sia dal punto di vista dell’accesso. Inoltre, il sistema deve essere capace di cooperare e di scambiare informazioni o servizi con altri sistemi o prodotti in maniera più o meno completa e priva di errori, con affidabilità e con ottimizzazione delle risorse. Il sistema deve poter interagire con sistemi universitari/scolastici per ottenere informazioni relative ai corsi di studio. (**Interoperability**)
13. Il sistema deve poter gestire un quantitativo di dati in continuo aumento (**Scalability**)
14. Il sistema deve garantire il mantenimento, la consistenza e la precisione dei dati per l’intero ciclo di vita del sistema (**Data integrity**)
15. Il sistema deve mantenere e rispettare le funzioni, la performance ed i requisiti stabiliti per l’intero ciclo di vita (**Configuration management**)

### *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 tramite la quale può selezionare un'offerta specifica per vederne le informazioni o per candidarsi (nel caso del richiedente). L'utente non ha necessariamente bisogno di effettuare il log-in per poter fare delle ricerche nel sistema.

#### **Notifica invito al colloquio**

Quando un richiedente riceve un invito ad un colloquio, il sistema deve garantire che esso lo riceva. Deve comunicare 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.

#### **Tentativo di inserimento di offerte/candidature da parte di un utente non registrato**

Può succedere che un utente non registrato provi ad inserire un'offerta o una candidatura; in questo caso, dato che si tratta di operazioni vietate, non appena l'utente prova ad effettuare una delle due operazioni sopracitate gli compare un alert il quale lo avvisa che non è autorizzato a svolgere quella determinata operazione e viene invitato ad effettuare il login o la registrazione: in questo caso la seconda (si fornisce anche la possibilità di fare il login perché potrebbe esserci, al posto di un utente non registrato, un utente registrato ma che non ha effettuato il login). Ad esempio, un utente non loggato può vedere tutte le offerte, però non appena clicca sul bottone "Candidati" viene avvisato tramite un alert che per poter compilare il form, bisogna effettuare il login. Stessa cosa vale per l'inserimento dell'offerta, ovvero appena un utente non loggato clicca sul bottone "Inserisci offerta" viene mostrato l'alert. (La dashboard è la stessa sia per gli utenti loggati che non, la differenza sta che gli utenti non loggati vengono invitati ad effettuare il login se necessario).

**Cambiamento del punteggio dopo modifica ad un'offerta**

Se un offerente decide di modificare dei requisiti relativi ad una sua offerta, il punteggio delle candidature effettuate su quell'offerta può cambiare. Quindi, nel momento in cui l'offerente conferma le modifiche effettuate tramite un bottone "Conferma modifiche", i punteggi delle candidature effettuate precedentemente verranno ricalcolati. Ad esempio, se per una determinata offerta sono state sottomesse 10 candidature, un'eventuale modifica sulle informazioni dell'offerta comporterà un ricalcolo del punteggio delle 10 candidature sottomesse in precedenza.

**Tentativo di inserimento di un curriculum (in formato pdf) di dimensioni elevate**

Nel caso in cui un richiedente prova ad inserire un file pdf rappresentante il curriculum la cui dimensione va oltre il limite stabilito, l'inserimento fallirà e il richiedente dovrà inserire un file di dimensioni adeguate.

**Eliminazione candidature relative ad un'offerta che viene eliminata**

Nel caso in cui un offerente decide di eliminare una determinata offerta di lavoro, tutte le candidature effettuate per quell'offerta vengono di conseguenza eliminate dal sistema. L'eliminazione dell'offerta, e quindi anche della candidatura, viene comunicata al richiedente tramite una notifica.

**Creazione di un nuovo gruppo**

Un utente (offerente o richiedente) può decidere di creare un gruppo; per crearlo deve specificare un nome, la categoria di appartenenza e una descrizione utile per far capire ai possibili nuovi membri di cosa si parla all'interno di esso. Una volta creato, il gruppo verrà utilizzato per creare discussioni tra i membri iscritti e ogni membro può invitare altri utenti ad unirsi al gruppo in questione.

**Inserimento di una nuova offerta**

Un offerente può decidere di inserire una nuova offerta; per farlo, dopo aver cliccato sull'apposito bottone, deve compilare un form nel quale inserirà il titolo dell'offerta, la categoria di impiego, una descrizione testuale utile per illustrare al meglio l'offerta lavorativa, le competenze che sono necessarie per ottenere il lavoro, lo stipendio offerto ed il tipo di contratto che verrà stipulato con colui che verrà assunto. Una volta inserita l'offerta, essa sarà visibile a tutti gli utenti del sistema.

**Inserimento di una nuova candidatura**

Un richiedente può decidere di candidarsi ad un'offerta che gli interessa; una volta individuata l'offerta, tramite il click sull'apposito bottone, compila il form necessario per la candidatura, dove inserisce il numero di anni di esperienza nell'ambito stabilito nell'offerta, il titolo di studio più recente, le lingue straniere conosciute (se richieste) e l'età. Una volta inserita, la candidatura sarà visibile all'offerente che ha inserito quella determinata offerta, insieme al relativo punteggio.



### *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.

E-Learning: non è prevista la presenza di corsi E-Learning perché non è ritenuto un servizio fondamentale in questo sistema.

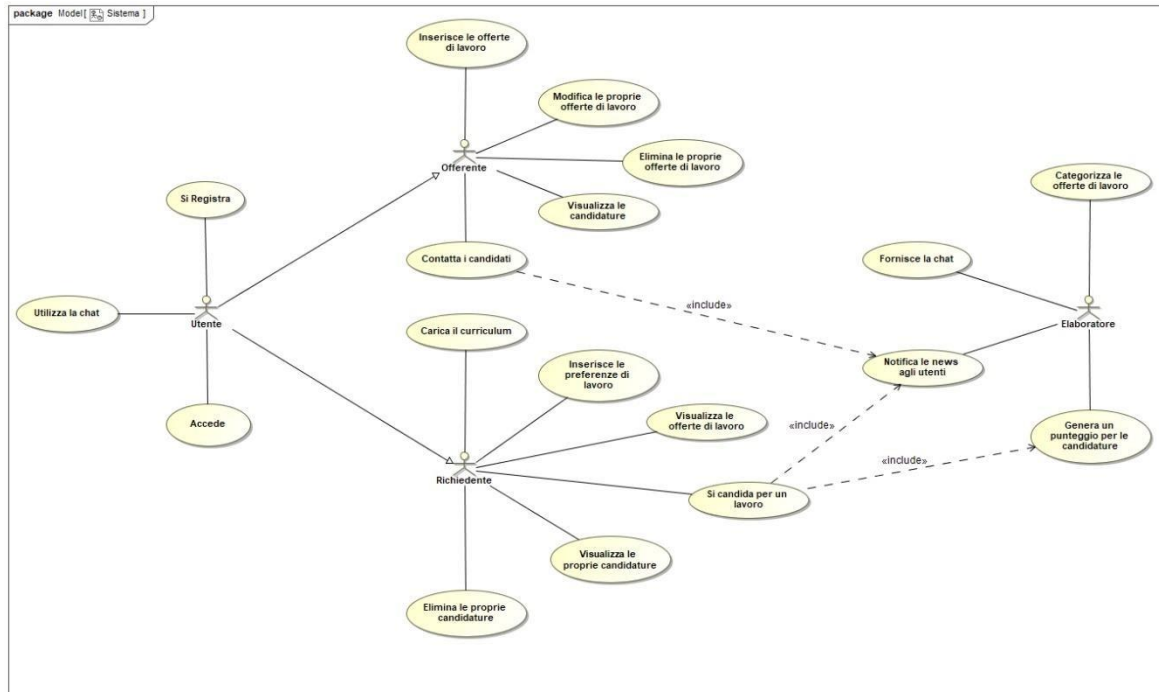
Sezione News: nel sistema non sarà presente una sezione news dedicata alle ultime notizie sul mondo del lavoro perché secondo noi non è un servizio importante per questo sistema.

### *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 dell'elaborazione dei dati: categorizza le offerte, calcola il punteggio per le candidature, manda le notifiche agli utenti e fornisce una chat per permettere agli utenti registrati di comunicare.

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.

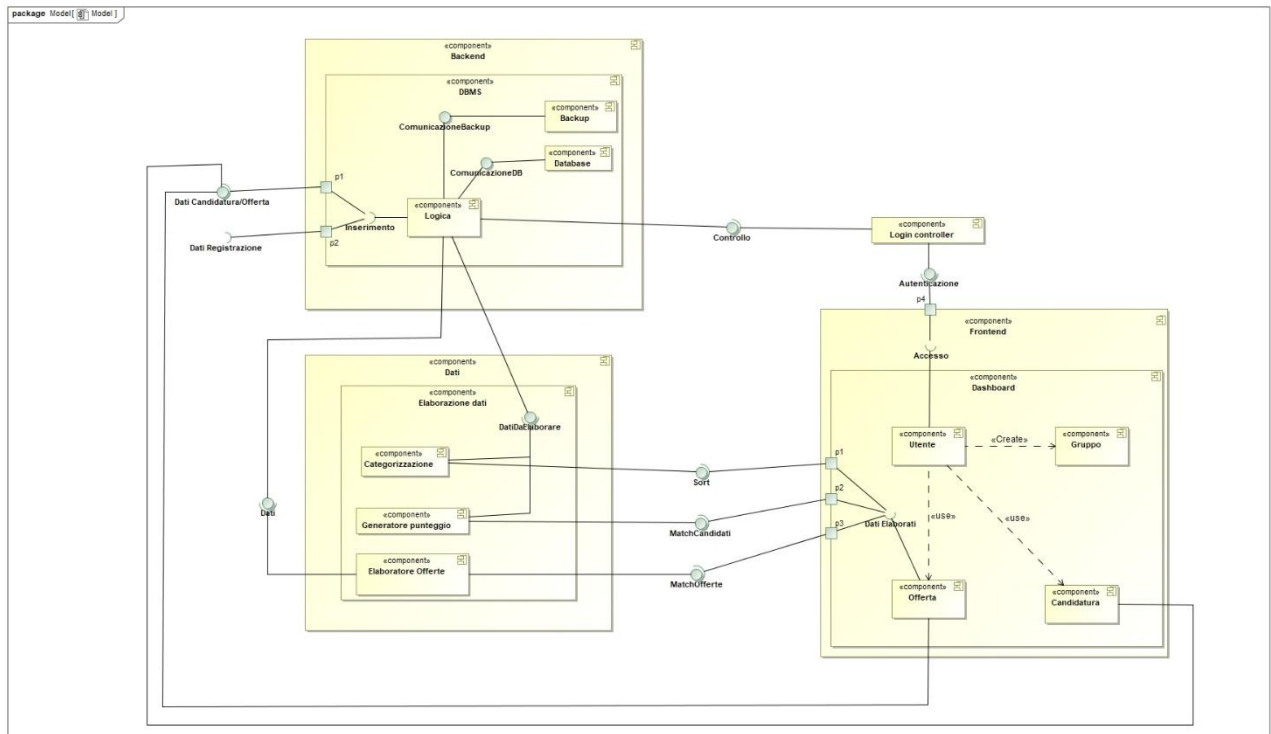
Gli use cases connessi attraverso l'«include» allo use case “notifica le news agli utenti” indicano le azioni che comportano l'invio di una notifica:

- 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:

- **Backend:** è il componente che rappresenta la parte di sistema che memorizza e organizza i dati e si assicura anche che tutto sul lato client del sistema funzioni correttamente. È la parte del sistema che non è possibile visualizzare e con la quale non si può interagire. È composto dal seguente componente:

**DBMS:** 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 dal component "Logica"; i due component comunicano tramite l'interfaccia "ComunicazioneDB".

Backup: componente utilizzato per effettuare un backup dei dati, che riceve dal component "Logica"; i due component comunicano tramite l'interfaccia "ComunicazioneBackup".

Logica (Logic Controller): componente che gestisce la logica di comunicazione con i component database e backup. Quando devono essere inseriti dei dati nel sistema, questo componente verifica la disponibilità di database e backup per effettuare l'invio di dati. Questo componente comunica con Database e Backup tramite due interfacce: ComunicazioneDB e ComunicazioneBackup. Attraverso la prima, il componente fa un check sullo stato del Database e, se esso è operativo, invia i dati; stesso procedimento, tramite la seconda, per quanto riguarda l'interazione con il Backup. Nel caso in cui sia Backup che Database non sono operativi, l'inserimento dei dati non andrà a buon fine e verrà inviato un messaggio di errore all'utente. Riceve in input i dati relativi a candidature, offerte e registrazioni tramite l'interfaccia "Inserimento".

- **Dati**: è il componente che rappresenta tutte le manipolazioni che vengono fatto sui dati presenti nel sistema. È composto dal seguente componente:

**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 effettuarne 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 dall’interfaccia “DatiDaElaborare”, mentre il componente “Elaborazione offerte” riceve le informazioni dall’interfaccia “Dati” (preferenze utente e offerte). Il component che fornisce i dati alle due interfacce sopracitate è “Logica”, il quale interroga i database per ottenere i dati e inviarli.

- **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.
- **Frontend:** è il componente che rappresenta la parte del sistema con la quale l’utente interagisce; include tutti i servizi messi a disposizione dell’utente. È composto dal seguente componente:

**Dashboard:** questo componente rappresenta la parte del sistema dedicata alla visualizzazione dei dati e alla fornitura dei servizi. È composto da i seguenti componenti:

Utente: è il componente che rappresenta la sessione dell’utente una volta effettuato l’accesso al sistema. Qui l’utente può usufruire dei servizi dedicati al proprio profilo e può accedere agli altri componenti della dashboard e ai relativi servizi. All’interno di questo componente sono inclusi anche i servizi relativi alle preferenze lavorative inserite dall’utente.

Gruppo: è il componente che rappresenta i servizi messi a disposizione dell’utente per quanto riguarda la creazione e la gestione di un gruppo.

Offerta: è il componente che rappresenta i servizi messi a disposizione dell’utente per inserire, modificare e visualizzare le offerte presenti nel sistema. All’interno di questo componente sono inclusi anche i servizi relativi ai requisiti inseriti nelle offerte.

Candidatura: è il componente che rappresenta i servizi messi a disposizione dell’utente per quanto riguarda l’inserimento e la visualizzazione delle candidature presenti nel sistema.

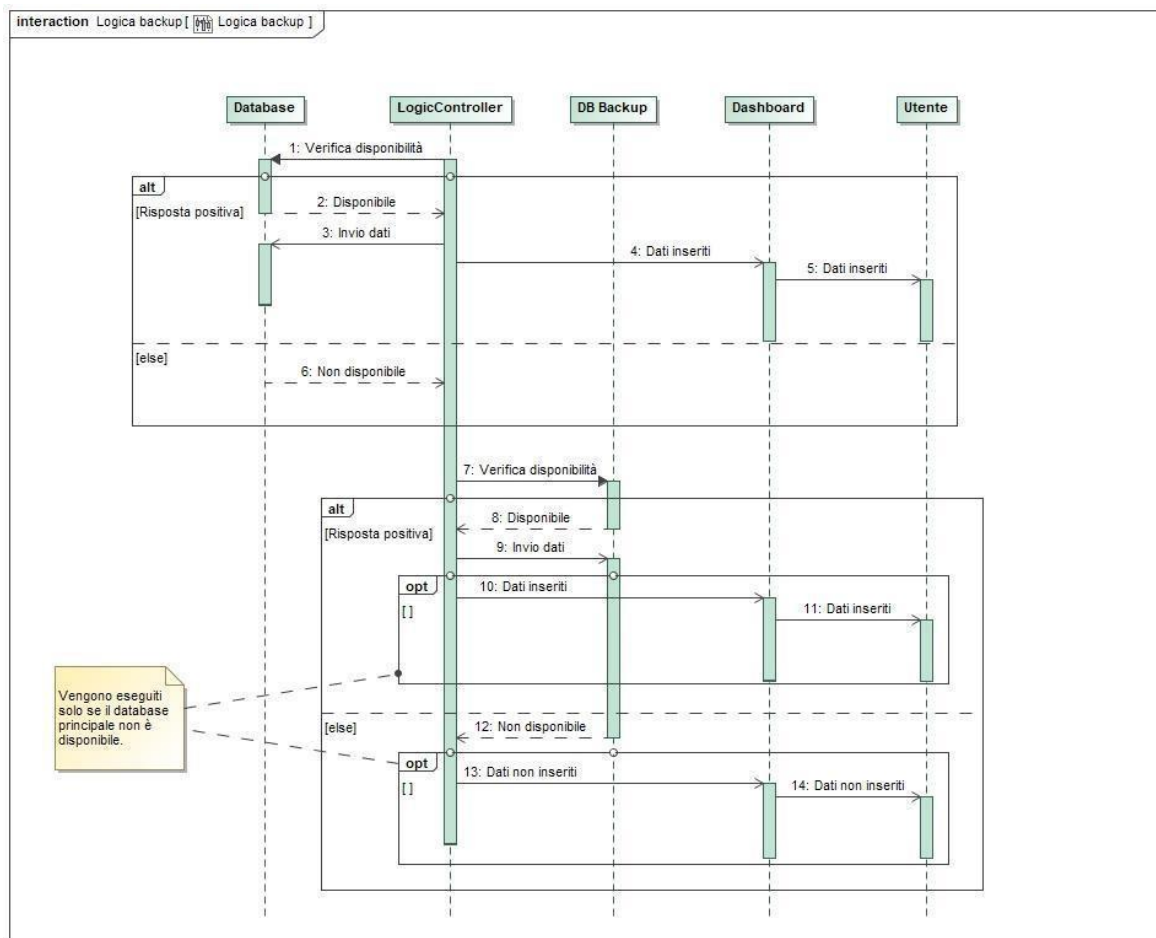
La dashboard utilizza tre interfacce, una per ogni servizio offerto, tramite le quali riceve i dati necessari per lo svolgimento del relativo servizio. Le interfacce in questione sono:

- *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 proprio 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: logica di backup, visualizzazione dati, inserimento ed eliminazione offerta, candidatura ed invito al colloquio.

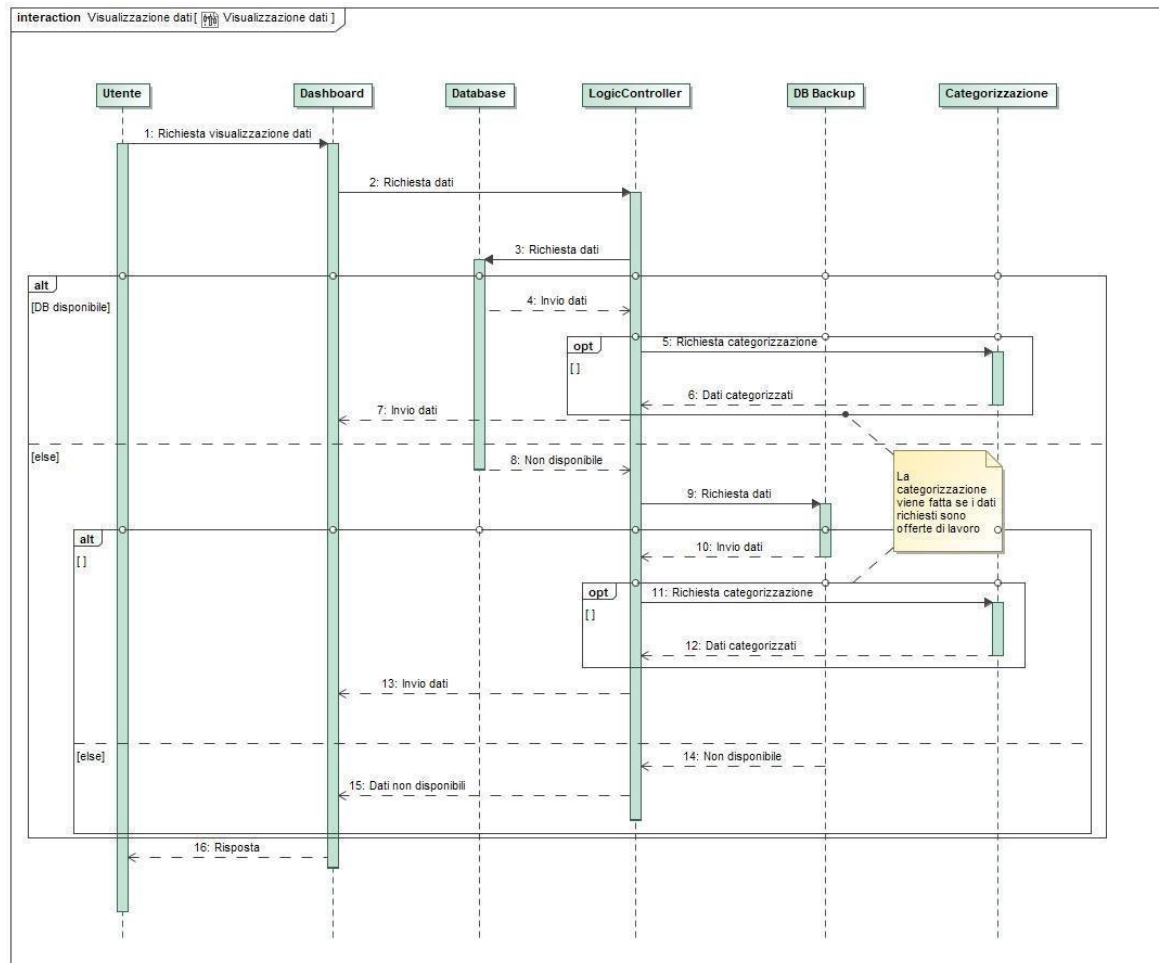
### Logica backup



Questo sequence diagram rappresenta il processo di logica backup in cui il logic controller verifica la disponibilità del database. Se la risposta è positiva, quindi se è disponibile, il logic controller invia i dati al database e viene notificato all'utente l'avvenuto inserimento. In seguito, a prescindere della disponibilità o non disponibilità del database principale, il logic controller verifica la disponibilità del database di backup. Se il DB backup è disponibile, Il logic controller invia i dati al

DB di backup e viene notificato all'utente l'avvenuto inserimento solo se il database principale non era disponibile. Se invece il DB di backup non è disponibile, viene eseguito il secondo opt, dove il logic controller comunica all'utente tramite la dashboard che i dati non sono stati inseriti.

### Visualizzazione Dati

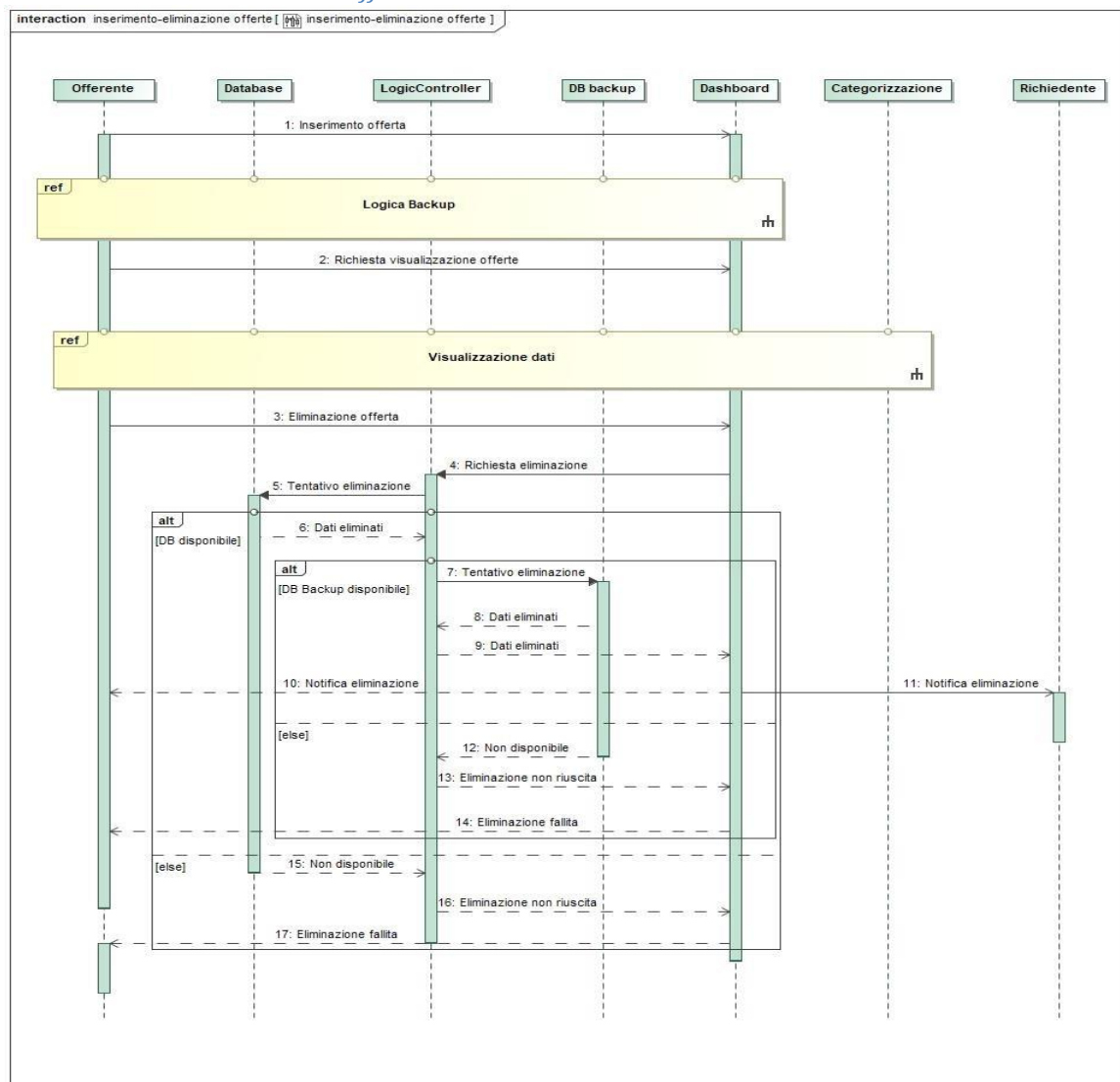


Questo sequence diagram rappresenta il processo di visualizzazione dati dal momento in cui un utente li richiede al momento in cui gli vengono mostrati. L'utente richiede la visualizzazione di dati attraverso la dashboard, la quale inoltra la richiesta al logic controller che richiede i dati prima al database principale e, se questo non fosse disponibile, li richiede al database di backup. Se i dati di cui si parla sono offerte di lavoro, il logic controller ne richiede la categorizzazione prima di presentarli all'utente. In caso di successo nel reperire i dati da uno dei database, questi



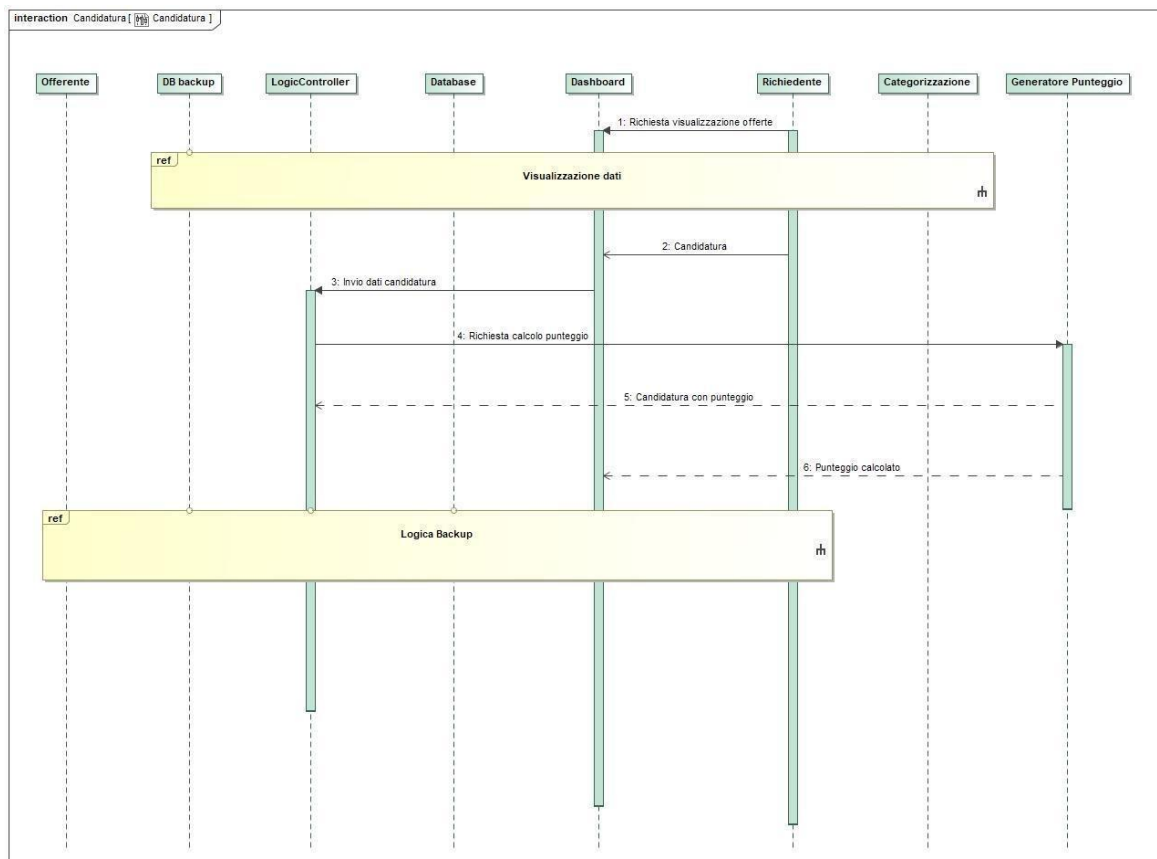
vengono mostrati all'utente attraverso la dashboard; nel caso in cui nessuno dei due database può rendere disponibili i dati, una notifica di errore viene mandata all'utente. Questo sequence diagram è stato reso generico perché viene utilizzato sia per la visualizzazione delle offerte (nei sequence "Inserimento ed eliminazione offerta" dall'offerente e nel sequence "Candidatura" dal richiedente) sia per la visualizzazione delle candidature (nel sequence "Invito al colloquio" dall'offerente). Questa operazione si riferisce a quando l'utente, per fare la richiesta di visualizzazione delle offerte e delle candidature, richiede i dati. Una volta richiesti i dati, questi gli vengono mostrati sullo schermo.

### *Inserimento ed eliminazione offerta*



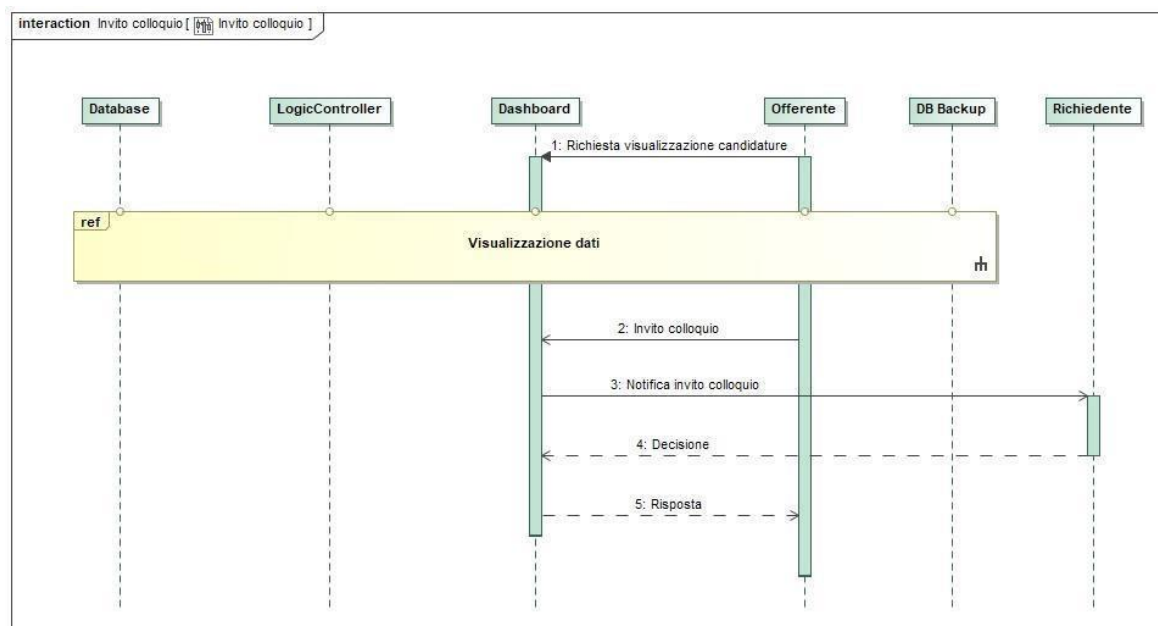
Questo sequence rappresenta i processi di inserimento ed eliminazione delle offerte di lavoro. L'offerente, tramite la dashboard, inserisce tutte le informazioni necessarie per creare un'offerta; fatto ciò, viene richiamato il sequence diagram "Logica Backup" il quale invia i dati al database principale e a quello di backup, se sono operativi. Quando un offerente invia una richiesta di visualizzazione delle offerte presenti nel sistema, viene chiamato in causa il sequence diagram "Visualizzazione Dati" che verifica la disponibilità dei dati e, in questo caso, li fornisce categorizzati. Nel caso in cui un offerente vuole eliminare un'offerta, invia tramite la dashboard una richiesta di eliminazione al Logic Controller che provvede ad eliminare i dati dal database principale e da quello di backup. Una volta eliminati i dati, viene inviato l'esito alla dashboard che comunica l'eliminazione, tramite una notifica, all'offerente che ha eliminato l'offerta e ai richiedenti che si erano candidati per quell'offerta. Se il database principale è disponibile, ma il database di backup no, l'eliminazione non avviene.

### Candidatura



Questo sequence diagram rappresenta la candidatura di un Richiedente ad un'offerta. Per prima cosa il richiedente invia una richiesta di visualizzazione delle offerte alla dashboard, viene richiamato il sequence diagram della Visualizzazione dei dati che mostra le offerte categorizzate all'utente. Dopo il reference, il richiedente invia la candidatura alla dashboard, che a sua volta invia i dati della candidatura al logic controller. Quest'ultimo richiede il calcolo del punteggio al generatore del punteggio, che a sua volta fa tornare indietro la candidatura con il punteggio ottenuto. Il punteggio calcolato, potrà essere mostrato all'offerente attraverso la dashboard. Infine, è presente un secondo reference, che richiama la Logica di backup, che inserisce i dati della candidatura con il punteggio nei database.

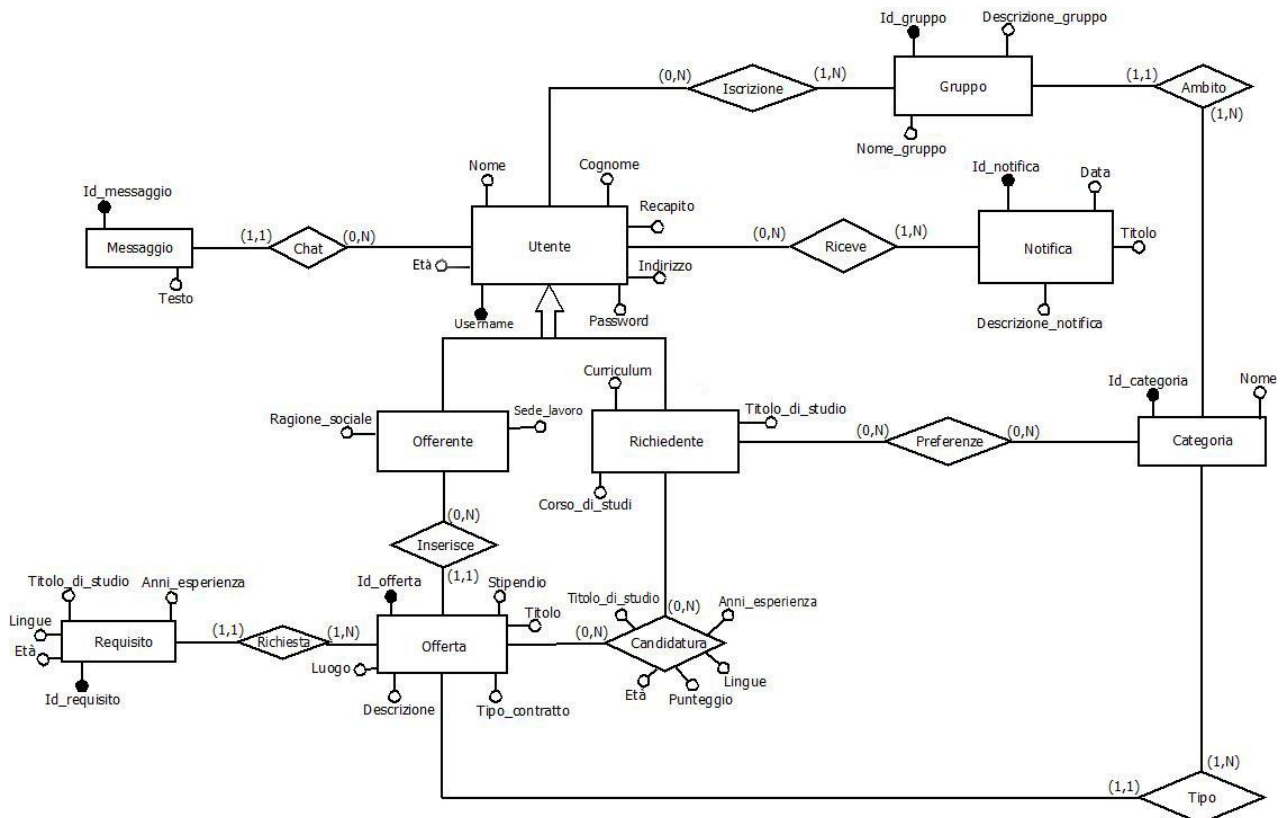
### *Invito al colloquio*



Questo sequence diagram rappresenta l'invito al colloquio inviato ad un richiedente da parte di un offerente. Dopo che l'offerente ha richiesto la visualizzazione delle offerte da lui inserite, viene eseguito il sequence diagram "Visualizzazione Dati" che mostra all'offerente le offerte che ha inserito da quando si trova nel sistema. A questo punto, dopo aver scelto un'offerta specifica, invita al colloquio il/i candidato/i che desidera; la comunicazione avviene tramite l'invio di una mail e di una notifica che viene inviata sulla dashboard del/i candidato/i scelto/i. Infine, il richiedente comunica l'esito tramite la dashboard utilizzando due bottoni: "Accetta" e "Rifiuta". In base alla scelta fatta, il candidato utilizza il bottone appropriato.

## D. Dati e loro modellazione

### D.1 Modello E-R



### D.2 Modello Relazionale

Utente (**Username**, Password, Nome, Cognome, Recapito, Indirizzo, Età, Sede\_lavoro, Ragione\_Sociale, Corso\_di\_studi, Curriculum, Titolo\_di\_studio , flag)

Offerta (**Id\_offerta**, Username, Descrizione, Stipendio, Luogo, Titolo, Tipo\_contratto)

Candidatura (**Username**, **Id\_offerta**, Punteggio, Età, Titolo\_di\_studio, Anni\_esperienza, Lingue)

Riceve (**Username**, **Id\_notifica**)

Notifica (**Id\_notifica**, Username, Data, Descrizione\_notifica, Titolo)

Gruppo (**Id\_gruppo**, Nome\_gruppo, Descrizione\_gruppo, Categoria, isprivate)

Iscrizione (**Id\_gruppo**, **Username**)

Categoria (**Id\_categoria**, Nome)

Messaggio (**Id\_messaggio**, Username, testo)

Requisito (**Id\_requisito**, *Id\_offerta*, Anni\_esperienza, Titolo\_di\_studio, Lingue, Età)  
Preferenze (*Username*, **Id\_categoria**)

## 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 dodici entità:

- Offerente;
- Richiedente;
- Offerta;
- Candidatura;
- Riceve;
- Notifica;
- Gruppo;
- Iscrizione;
- Categoria;
- Messaggio;
- Requisito;
- Preferenze;

Nella traduzione dal modello E-R al modello relazionale, la generalizzazione di utente viene rappresentata effettuando un collasso verso l'alto: le entità *offerente* e *richiedente* vengono eliminate e l'entità *utente* eredita gli attributi delle due entità eliminate (*Sede\_lavoro*, *Ragione\_Sociale*, *Corso\_di\_studi*, *Curriculum*, *Titolo\_di\_studio*). La differenziazione tra offerente e richiedente verrà effettuata tramite un valore booleano chiamato "flag".

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*.
- Un utente può iscriversi a più *gruppi*; ad un gruppo possono iscriversi più *utenti*.
- Un gruppo appartiene ad una categoria; ad una categoria possono appartenere più gruppi.
- Ogni *utente* potrebbe mandare da uno a più *messaggi* tramite una chat; Un *messaggio* viene mandato tramite una chat da un *utente*.
- Un'*offerta* richiede da uno a più *requisiti*; almeno un *requisito* è richiesto da un'*offerta*. *Requisito* ha gli stessi attributi di *candidatura* per poter calcolare il punteggio da assegnare alla *candidatura*.
- Ogni *offerta* ha un tipo di *categoria*; una *categoria* può avere diversi tipi di *offerte*.
- Ogni *richiedente* può preferire da 0 a più *categorie*; ogni *categoria* potrebbe essere preferita da 0 a 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.

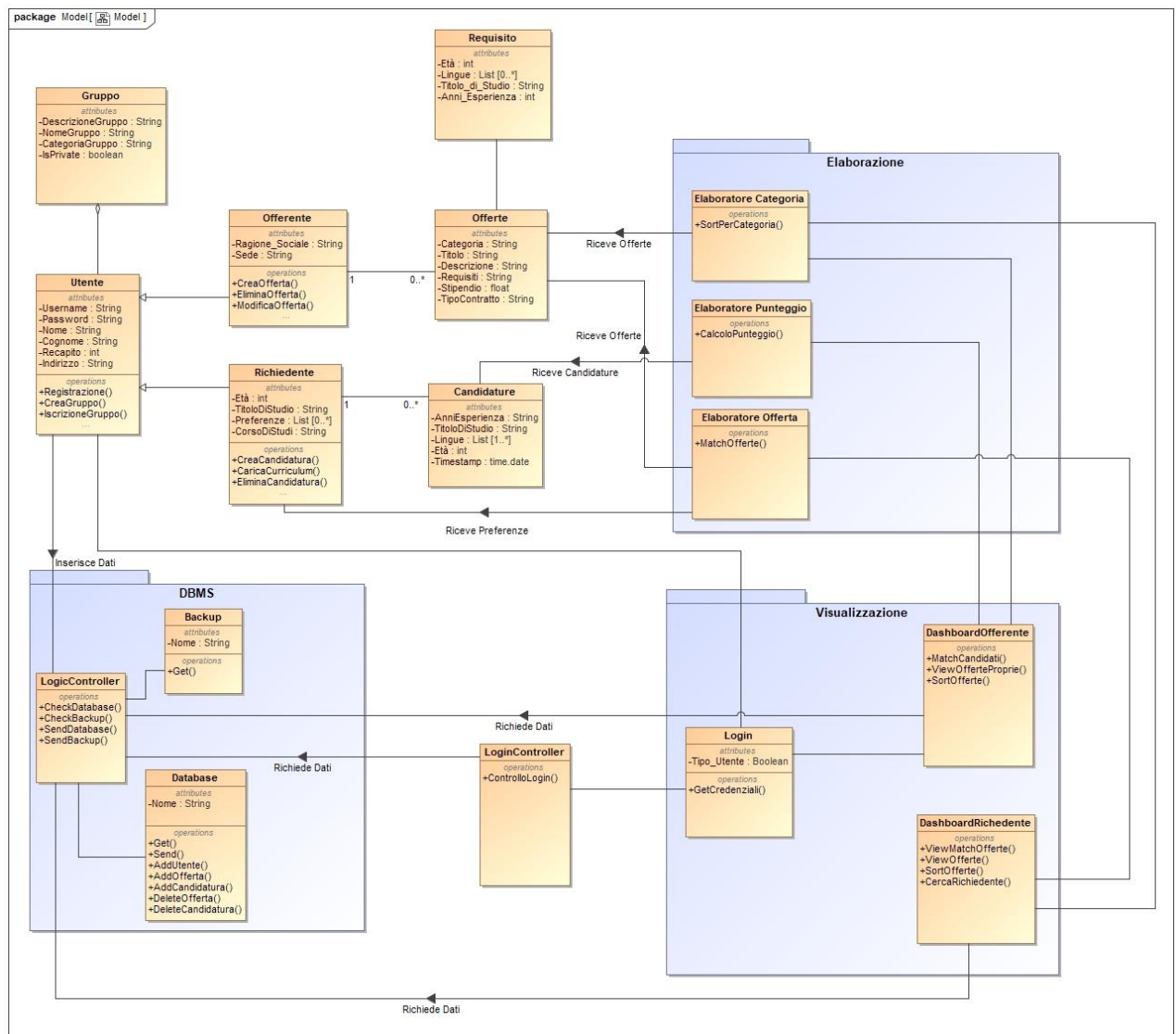
## Framework Laravel

Abbiamo scelto di utilizzare il framework Laravel perché garantisce buone performance, stabilità, sicurezza e affidabilità. Questo framework permette di effettuare coding elegante e pulito e, soprattutto, permette la lavorazione sul progetto in gruppo più semplice. È inoltre ottimo e veloce per scrivere API e il sistema di Routing è abbastanza potente per permettere di creare servizi RESTful con cui interfacciare verso l'esterno i nostri dati. Il livello di supporto (documentazione e supporto della community) è tra i più elevati e inoltre viene offerto un servizio chiamato Laracasts, il quale consiste in un catalogo di video tutorial fatti molto bene per aiutare neofiti ed esperti.



## 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). La classe “Requisito” rappresenta i requisiti inseriti dall’offerente in una determinata offerta di lavoro. La classe “Gruppo” rappresenta un insieme di utenti che creano un gruppo all’interno del sistema ed è collegata alla classe “Utente” tramite una relazione di aggregazione. I tre package con i quali queste classi interagiscono sono:

- *DBMS*: 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 gli stessi dati che vengono inviati al database e “LogicController” che si occupa di controllare lo stato del database principale e di quello di backup per poi procedere all’invio dei dati. Nel caso in cui sia “Backup” che “Database” non sono operativi, l’inserimento dei dati non andrà a buon fine e verrà inviato un messaggio di errore all’utente.
- *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 (principale o di backup) tramite la classe “LogicController” per verificare la presenza e la correttezza dei dati inseriti. L’interazione tra le classi “Login” e “LogicController” viene gestita dalla classe “LogicController”.

## 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.

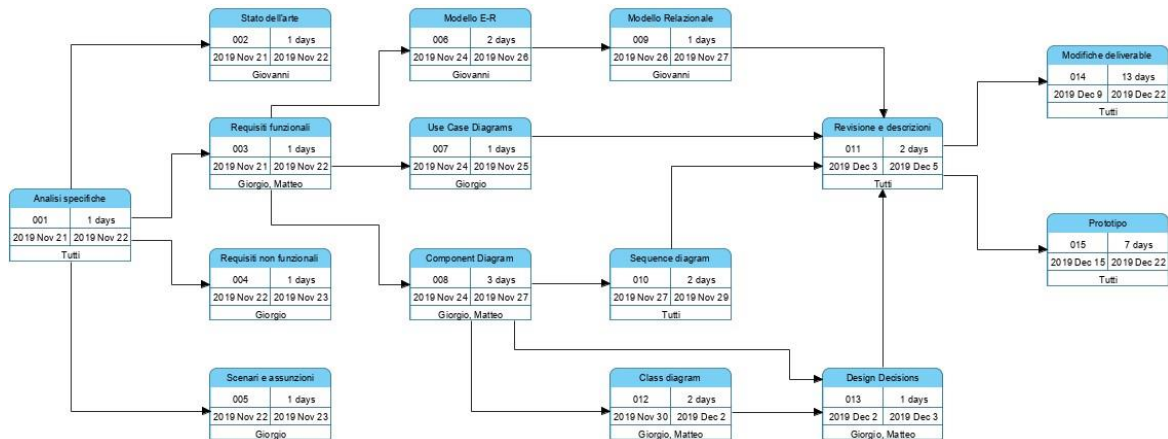
Ogni utente accede alla dashboard dedicata tramite la quale utilizza i servizi a lui offerti; la dashboard include una sezione di notifiche che ha il compito di informare l'utente su determinati avvenimenti.

Per questioni di sicurezza, ogni dato inserito nel database viene inviato anche all'interno di un database di backup; la comunicazione con le due parti viene gestita da un controller che, prima dell'invio, controlla lo stato dei due database.

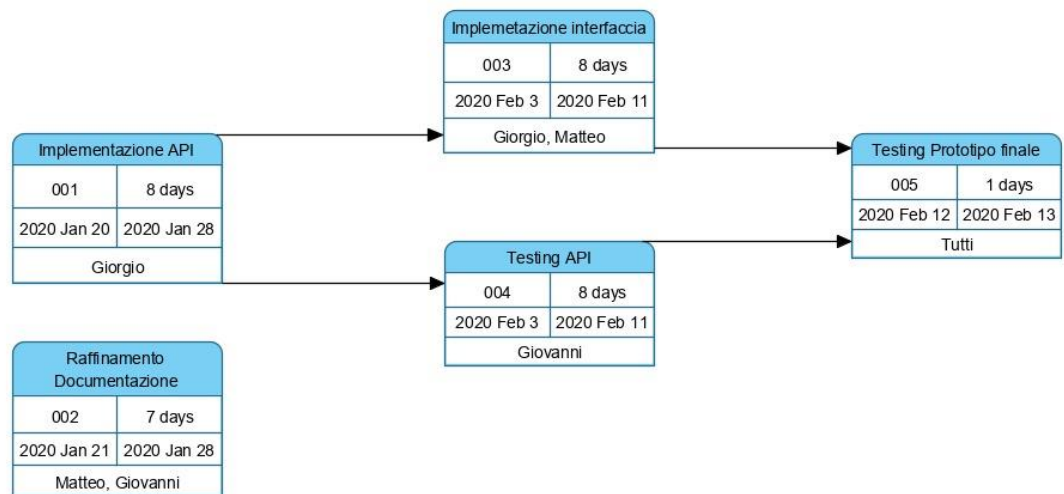
Per testare le performance del sistema, utilizzando le classi factory e i dei seed forniti dal framework Laravel, abbiamo popolato il database con 5000 utenti, 5000 offerte e 5000 candidature; abbiamo visto che le performance sono rimaste inalterate, rispetto ad una situazione iniziale nella quale erano presenti 10 elementi per quanto riguarda utenti, offerte e candidature. Pertanto, il sistema può gestire una grande quantità di dati in continuo aumento.

# H. Effort Recording

## H.1 PERT



PERT dopo il Milestone del Deliverable 2:



## H.2 Logging

## Personal Journal

Team (number and name): Software Solutions
Student name: Giovanni Spaziani, Giorgio Morico, Matteo Leoncini
Student number: 3
Email: giovanni.spaziani@student.univaq.it, giorgio.morico@student.univaq.it, matteo.leoncini@student.univaq.it

When (Month/Day)	Time spent	Partners (please report how many people have been working)	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 1 h	1	Stato dell'arte	Learning	A. Stato dell'Arte
21	11 1:30 h	2	Requisiti funzionali	Doing	B. Raffinamento dei requisiti
22	11 1 h	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 1 h	1	Correzione Component Diagram	Doing	C. Architettura Software
26	11 30 min.	3	Learning Sequence Diagram	Learning	C. Architettura Software
26	11 1 h	2	Sequence Diagram	Doing	C. Architettura Software
27	11 30 min.	2	Learning Class Diagram	Learning	F. Design di basso livello
27	11 1 h	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 1 h	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 1 h	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 1 h	3	Revisione finale	Doing	
12	12 30 min.	1	Nuovo stato dell'arte	Doing	A. Stato dell'Arte
12	12 1:30 h	2	Modifiche Sequence Diagrams	Doing	C. Architettura Software
13	12 20 min.	2	Completamento stato dell' arte	Doing	A. Stato dell'Arte
13	12 1:15 h	1	Modifiche Sequence Diagrams	Doing	C. Architettura Software
13	12 15 min.	1	Modifica Component	Doing	C. Architettura Software
13	12 15 min.	1	Modifica Class	Doing	F. Design di basso livello
13	12 30 min.	1	Modifiche requisiti	Doing	B. Raffinamento dei requisiti
14	12 40 min.	1	Modifiche requisiti	Doing	B. Raffinamento dei requisiti
14	12 30 min.	1	Modifiche generali deliverable	Doing	
15	12 40 min.	1	Modifiche requisiti	Doing	B. Raffinamento dei requisiti
15	12 20 min.	1	Modifiche generali deliverable	Doing	
15	12 1:20 h	1	Modifiche Sequence	Doing	C. Architettura Software
15	12 50 min.	1	Modifiche Modello E-R	Doing	D. Dati e loro modellazione
16	12 1 h	3	Modifiche requisiti	Doing	B. Raffinamento dei requisiti
17	12 1 h.	3	Learning Laravel e Postman	Learning	
18	12 2 h.	3	Learning Laravel e Postman	Learning	
20	12 3 h.	3	Correzione deliverable	Doing	
20	12 3 h.	3	Prototipo	Doing	
21	12 2 h.	1	Prototipo	Doing	
22	12 1 h.	1	Modifiche generali deliverable	Doing	
22	12 1:30 h	1	Prototipo	Doing	
14	1 2 h	1	Modifiche generali deliverable	Doing	
14	1 2 h	1	Prototipo	Doing	
17	1 3h	2	Modifiche generali deliverable	Doing	
17	1 3h	1	Prototipo	Doing	
18	1 20 min.	2	Test Prototipo	Doing	
18	1 1h	1	Modifiche generali deliverable	Doing	
20	1 3h	3	Prototipo	Doing	
22	1 3h	3	Prototipo	Doing	
23	1 4h	3	Prototipo	Doing	
27	1 4h	3	Prototipo	Doing	
3	2 3h	3	Prototipo	Doing	
4	2 4h	3	Prototipo	Doing	
5	2 4h	3	Prototipo	Doing	
6	2 4h	3	Prototipo	Doing	
7	2 3h	3	Prototipo	Doing	
8	2 4h	3	Prototipo	Doing	
10	2 5h	3	Prototipo	Doing	
11	2 4h	3	Prototipo	Doing	
12	2 2h	3	Prototipo	Doing	
13	2 1:30 h	1	Modifiche generali deliverable	Doing	
13	2 1:30 h	1	Prototipo	Doing	

*Ore totali:* 222h 30’

*Ore doing:* 212h 50’

*Ore learning:* 9h 40’

# Appendix. Prototype

---

## Documentazione Postman

<https://documenter.getpostman.com/view/9826509/SzKN128n>

Sono stati rappresentati i seguenti endpoint:

- Offerta: possono essere pubblicate, modificate ed eliminate dagli offerenti (un'offerta può essere modificata o eliminata solo da chi l'aveva pubblicata), possono essere visualizzate da tutti gli utenti.
- Utente: gli utenti si possono registrare, e di conseguenza possono loggare; possono modificare le informazioni inserite in fase di registrazione e hanno la possibilità di eliminare il proprio account.
- Candidatura: può essere inviata da un richiedente che può anche visualizzarle. Un utente può eliminare una sua candidatura, ma non può modificarla una volta inviata.
- Requisito: sono i requisiti richiesti ai potenziali candidati da colui che pubblica una determinata offerta. Possono essere successivamente modificati o eliminati e possono essere visualizzati da tutti gli utenti.
- Gruppo: possono essere creati o visualizzati per procedere all'iscrizione dagli utenti registrati (offerente o richiedente). Solo chi ha creato il gruppo può modificarlo o eliminarlo. Si può impostare l'accessibilità del gruppo, ovvero se renderlo pubblico o privato.
- Preferenza: possono essere aggiunte da un richiedente in fase di registrazione, che può successivamente eliminarle ed aggiungerne di nuove. Possono essere visualizzate solo dall'utente stesso.
- Notifica: sono inviate dal sistema, possono visualizzarle solo gli utenti a cui vengono inviate.

Nella documentazione Postman vengono mostrati degli esempi di risposta in caso di inserimento, visualizzazione, eliminazione e modifica dei dati attraverso le API. Inoltre, vengono mostrati alcuni esempi di errori (401, 403 e 404).