

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

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

Progetto 1-Team Fresh

Date	<05/12/2019>
Deliverable	D1
Team (Name)	Team Fresh

Team Members		
Name & Surname	Matriculation Number	E-mail address
Piero Ferri	#245484	
Davide Fasciano	#245433	

Table of Contents of this deliverable

Articolo I. Progetto 1. Gestionale offerte di lavoro (#12 #13)

Il software consiste in un gestionale che permette agli offerenti (ditte, enti, docenti) di descrivere gli impieghi offerti e ai richiedenti (ad esempio, studenti, ma non solo) di cercare tra le offerte e candidarsi.

Il software dovrà essere in grado, tra le altre cose, di:

- elaborare le offerte per effettuare una categorizzazione delle stesse;
- elaborare i dati utente con le rispettive preferenze di lavoro;
- generare un punteggio in base ad ogni candidatura;
- presentare in primo piano all'utente le offerte di lavoro che fanno match, al meglio, con il proprio profilo.
- presentare in primo piano agli offerenti i candidati che fanno match, al meglio, con le proprie offerte;
- registrazione ditte e registrazione richiedenti

Il software dovrà inoltre:

- garantire una valutazione equa delle candidature secondo dei criteri stabiliti;
- essere in grado di notificare gli utenti sulle nuove offerte e sullo stato delle candidature in maniera opportuna e senza essere invasivo.

Ove possibile, il software dovrà minimizzare il data-entry, integrandosi con applicativi esistenti. L'applicativo deve essere indipendente da una specifica Università.

L'applicativo dovrà rispettare le norme relative alla sicurezza e privacy nella conservazione dei dati utenti, dovrà essere in grado di garantire che l'utente abbia ricevuto l'invito al colloquio, dovrà permettere l'integrazione con altri sistemi.

List of Challenging/Risky Requirements or Task

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Matching tra offerte e candidati	28/11	01/12	La scelta della struttura del matching è stata sempre al centro dei primi brainstorming riguardanti la strutturazione dell'intero sistema. Abbiamo superato questa difficoltà utilizzando matching factors per stabilire le correlazioni tra i profili degli offerenti e dei candidati
Categorizzazione	22/11	27/11	Trovare la giusta categorizzazione è cruciale per avere un sistema più snello, intuitivo, ma comunque espressivo. Abbiamo costruito una categorizzazione partendo dai matching factors ed estrapolando delle macro categorie che vanno ad identificare dei settori lavorativi.
Candidatura ottima	28/11	30/11	Affinchè si possa avere una valutazione equa dei candidati abbiamo riflettuto su cosa fosse un candidato ideale. Abbiamo risolto il problema identificando una candidatura ottima come una candidatura con punteggio relativo dei matching factors più alto.

A. Stato dell'Arte

Durante i primi giorni dalla consegna delle specifiche del progetto (dal 22/11 al 25/11) abbiamo ragionato sullo sviluppo dello stesso, cercando idee ed intuizioni riguardanti il suo funzionamento e su come effettivamente concretizzarle.

Essendo la prima esperienza nello sviluppo di una piattaforma di “cerco/trovo lavoro” abbiamo affiancato al processo creativo un’attenta ricerca in rete per avere un’idea più completa e delineata di ciò che saremo andati a sviluppare. Abbiamo scelto, tra le varie opzioni online, Linkedin come esempio principe, in quanto permette di creare profili dettagliati attraverso l’elenco delle varie competenze che possono essere aggiunte.

Questo aspetto è stato molto utile poiché ci ha permesso di capire meglio come ideare i matching factors che in seguito definiscono la candidatura ottima di un utente; è stato un buono spunto per la categorizzazione delle offerte in base a competenze e macro settori lavorativi.

Infine ci ha aiutato nel comprendere la user experience di una piattaforma di questo genere: essa, senza dubbio, deve essere intuitiva, snella ma comunque espressiva.

Per comprendere appieno le funzionalità di Linkedin abbiamo creato un’account, compilato il profilo e svolto alcuni test nel sistema, cercando offerte tramite categorie e in base a diversi ambiti lavorativi.

Inoltre abbiamo intervistato alcuni nostri colleghi provenienti da vari corsi di laurea per capire cosa si aspetta un utente che accede ad una piattaforma di questo genere per la prima volta e così avere un quadro più oggettivo e di ampio respiro che si scosti dalla nostra propensione tecnica.

B. Raffinamento dei Requisiti

A partire dai servizi minimali richiesti, raffinate la descrizione dei servizi offerti dal vostro applicativo. Descrivete anche I requisiti non funzionali.

A.1 Servizi (con prioritizzazione)

- **Log-in** : L'utente accede al sistema immettendo le proprie credenziali (username e password) precedentemente immessi durante la registrazione.
- **Gestione Profilo** : L'utente loggato, attraverso la propria dashboard, entra nel suo profilo e nella sezione Gestione Profilo gestisce i propri dati immettendo o modificando le proprie competenze ed esperienze.
- **Ricerca Offerte** : La Ricerca Offerte è una funzione del sistema che permette agli utenti di ricercare lavoro in base ai macro ambiti lavorativi o in base a ruoli già definiti.
- **Candidatura a Offerte** : La candidatura a Offerte permette ad un utente di candidarsi ad un'offerta presente nel sistema.
- **Notifica offerte(perfect-match)** : La Notifica Offerte è una funzione che permette all'utente di visualizzare nella propria dashboard le offerte che si avvicinano il più possibile alle competenze dello stesso utente.
- **Controllo Status Candidature** : Il Controllo Status Candidature permette all'Utente Candidato di monitorare l'avanzamento della propria candidatura rispetto ad una determinata offerta.
- **Elimina candidature** : Tale funzione serve ad eliminare una candidature in caso di errori o impossibilità.

- **Pubblicazione offerta** : La Pubblicazione Offerta permette alle aziende presenti nel sistema attraverso un account Offerente di pubblicare offerte di lavoro comprensive delle competenze richieste.
- **Gestione offerte pubblicate** : Questa funzione permette ai possessori di un account Offerente di gestire le proprie offerte di lavoro pubblicate e modificare in caso fosse necessario.
- **Analisi candidature** : Questa funzione permette all'Utente Offerente di monitorare le eventuali candidature per una data offerta di lavoro pubblicata precedentemente.
- **Richiesta di colloquio** : Una volta analizzate le possibili candidature, l'offerente può scegliere le candidature più consone per ricoprire il ruolo che cercano e richiedere il colloquio ai candidati scelti.
- **Chiusura offerte** : Scelte le candidature ed inviate le notifica di colloquio ai candidati scelti, l'offerente può chiudere le offerte da lui pubblicate eliminandole dal sistema.

A.2 Requisiti non Funzionali

- Il database non verrà gestito o mantenuto da noi; le specifiche dell'hardware verranno indicate dal nostro team, mentre i costi dello stesso e della manutenzione saranno a carico dello stakeholder
- Dependability - fault tolerance: Il sistema deve continuare ad offrire i servizi anche se si è persa la funzionalità di una componente.
- Dependability - security: Il sistema deve essere in grado di proteggersi da attacchi esterni, accidentali o intenzionali. La security diventa un requisito essenziale, visto che il nostro sistema dovrà essere connesso alla rete per permettere l'accesso ai dati di offerte e candidature nonché dei dati personali di offerenti e candidati

- Scalability: Il sistema garantisce un'architettura scalabile per supportare future espansioni (es. web, mobile).
- Usability: Il sistema deve garantire un'esperienza user-friendly e risultare il più intuitiva possibile
- Performance: il sistema deve risultare efficiente e deve lavorare con tempi di esecuzione accettabili

A.3 Scenari d'uso dettagliati

Scenario 1: Iscrizione e gestione profilo Candidato

L'utente si registra inserendo nome utente, password e i dati del suo profilo. Una volta registrato l'utente accede al sistema tramite le credenziali e premendo il pulsante "log-in".

Dopo il log-in, viene caricata la dashboard utente del tipo selezionato all'atto dell'iscrizione (sono possibili 2 tipi di account : Candidato e Offerente).

Dalla Dashboard utente di tipo Candidato si possono effettuare diverse operazioni :

la gestione del profilo, la ricerca delle offerte di lavoro , la gestione delle candidature e controllare le notifiche.

Scenario 2 : Iscrizione e gestione profilo Offerente

L'utente di tipo Offerente segue una procedura simile d'iscrizione dell'utente Candidato, ma orientata all'offerta di lavoro che pubblicherà nel sistema.

Dalla Dashboard di tipo Offerente si potranno svolgere diverse funzioni tra le quali:

la pubblicazione delle offerte, la gestione delle offerte pubblicate e l'analisi delle candidature con eventuale richiesta di lavoro.

Scenario 3 : Pubblicazione di un'offerta

L'utente di tipo Offerente, dalla propria Dashboard, clicca sul pulsante "Pubblica Offerta". Una volta cliccato si apre la sezione della pubblicazione delle offerte, in cui l'Offerente compila un modulo dove specifica una breve descrizione dell'offerta, compila i campi delle competenze richieste e specifica l'ambito lavorativo (Macro-Categoria).

Una volta pubblicata l'offerta attraverso il pulsante "Pubblica", essa sarà visibile nella Dashboard "Le Mie Offerte".

Scenario 4 : Ricerca dell'Offerta

L'utente Candidato ha due modi per accedere alle Offerte ovvero : dalla propria Dashboard nell'area notifiche (queste saranno le offerte presenti nel sistema che fanno match con il "Punteggio" massimo, calcolato sui matching factors) e nell'area Ricerca Offerte.

Nell'Area Ricerca Offerte il Candidato potrà cercare tra le Offerte di lavoro tramite l'ambito lavorativo a lui congeniale.

Scenario 5 : Richiesta di Colloquio Offerente

L'Offerente, tramite la Dashboard "Le mie Offerte", analizzerà i profili dei candidati e sceglierà, in base alle proprie esigenze, i candidati da invitare a colloquio attraverso l'apposito pulsante "Richiedi Colloquio".

Tale azione produrrà una Notifica di "Richiesta di Colloquio", la quale sarà presente nella Dashboard del candidato scelto dall'Offerente.

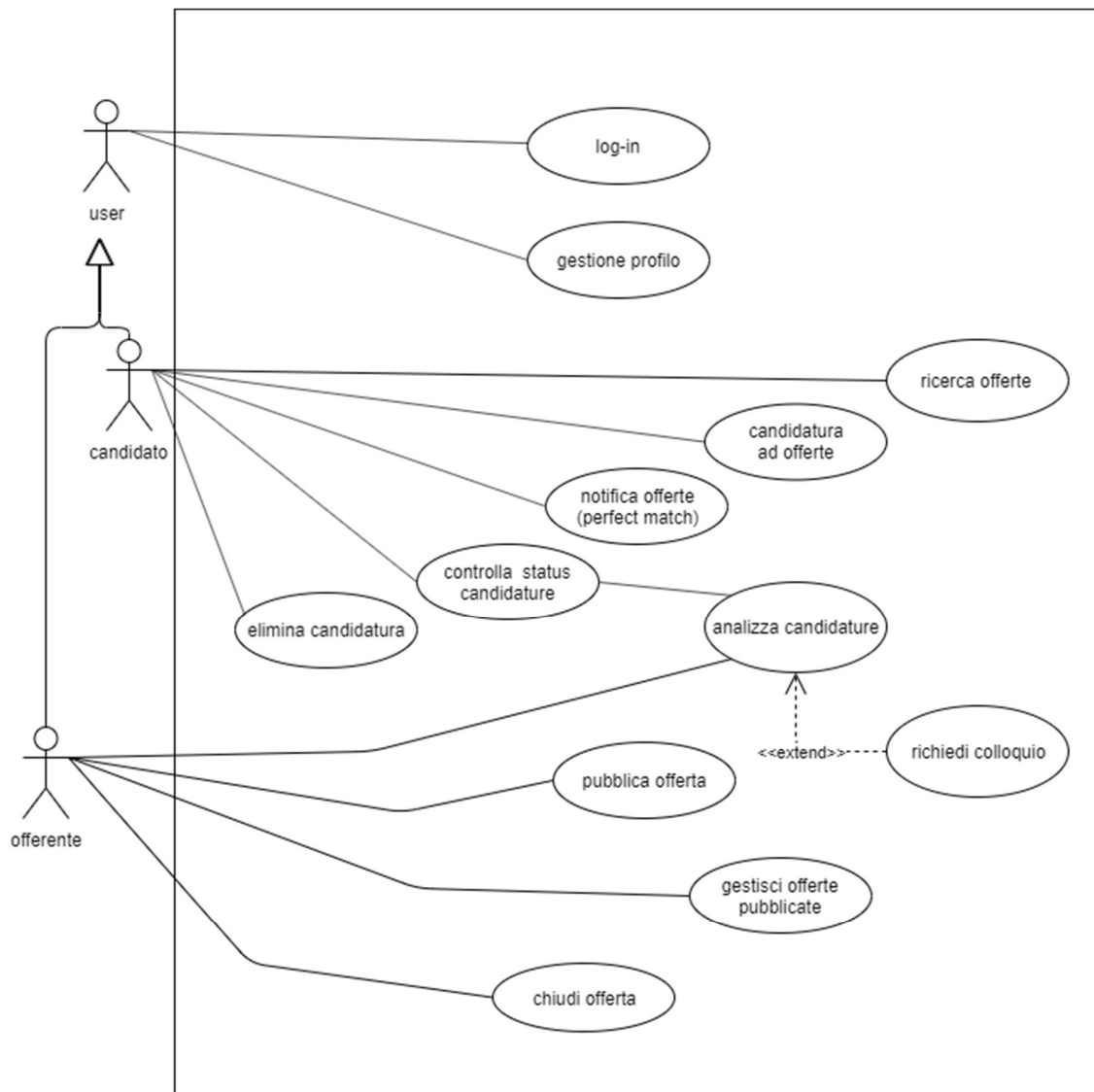
A.4 Excluded Requirements

-Abbiamo escluso il servizio di invio del curriculum e il servizio di corrispondenza elettronica in-app in favore di un "Profilo/curriculum" sul nostro sistema e un servizio di notifiche per rendere più efficace il matching, più snello il sistema in generale e meno invasivo. La corrispondenza tra Offerenti e Candidati e l'eventuale invio del curriculum non verrà coperto dal sistema evitando ulteriori potenziali minacce a livello privacy.

A.5 Assunzioni

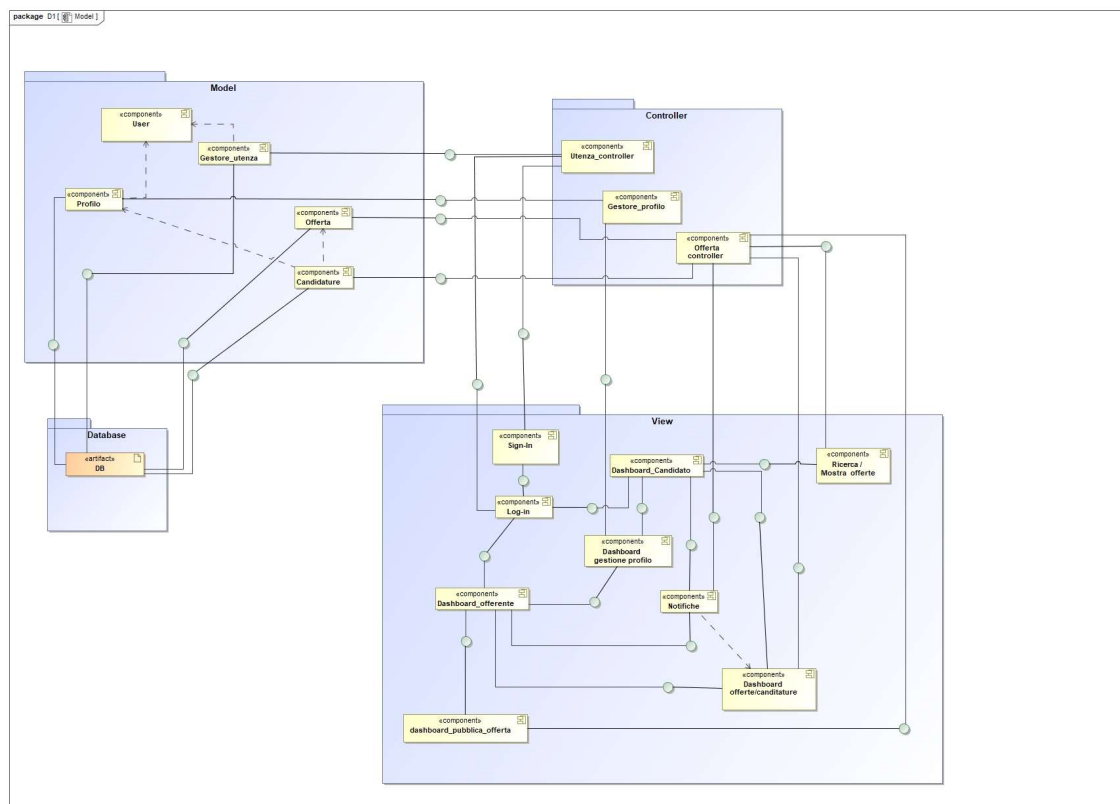
- Potrà utilizzare il sistema solo un utente registrato
- Assumiamo che l'utente accetti le condizioni di conservazione dei dati personali a noi necessarie per il funzionamento del sistema

A.6 Use Case Diagrams



C. Architettura Software

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



Il team ha deciso di utilizzare MVC Pattern (Model-View-Controller), ovvero un Pattern Architeturale molto diffuso nello sviluppo software, in particolare nell'ambito della programmazione orientata ad oggetti. La suddivisione in strati facilita la scalabilità e la manutenzione del software.

Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- Model fornisce i metodi per accedere ai dati utili all'applicazione;
- View visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- Controller riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti

D. Dati e loro modellazione

Definite le sorgenti di dati a voi necessarie per realizzare I servizi di cui sopra.

Modellat tali dati tramite un ER o similari. Specificate se e quali di tali dati sono gia' forniti da applicativi esistenti.

E. Design Decisions

<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

Functional requirements:

- Decomposizione funzionale: il sistema viene scomposto in funzionalità, ordinato per operazione e decomposto in altri piccoli moduli;
- Decomposizione object-oriented: il sistema viene scomposto in classi, ogni classe è un'entità importante nel dominio dell'applicazione, le classi possono essere scomposte in piccole classi.

Scelta:

Il team ha deciso di optare per la decomposizione funzionale perchè ritenuto più consono per svolgere tale progetto.

Non-functional requirements: il team ha deciso di dar più importanza ai seguenti requisiti non funzionali essendo essenziali per il corretto funzionamento del sistema:

- Dependability;
- Scalability;
- Usability;
- Performance.

Programming languages Decision: il team ha deciso per lo sviluppo del software di utilizzare il linguaggio java e di creare in particolare una desktop app. Tale decisione è stata concordata per conciliare sia i tempi e opportunità di progettazione e sviluppo sia la comodità, la sicurezza e le possibilità future di una desktop app.

In un prossimo futuro, grazie alla progettazione del team, questo sistema software potrebbe essere esteso anche su altre piattaforme (ad esempio web e mobile) grazie alla scalabilità del nostro sistema.

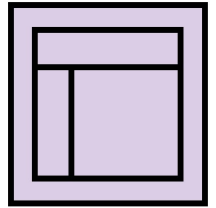
Mvc pattern decision: il team ha deciso di impiegare il “pattern mvc” per rappresentare l’architettura del software e il component diagram ad esso collegato. Tale decisione e’ stata concordata per organizzare il sistema in modo ottimale e logico, dividendolo quindi in tre parti. Siamo così in grado di facilitare la scalabilità e la manutenzione dell’applicazione.

F. Design di Basso Livello

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

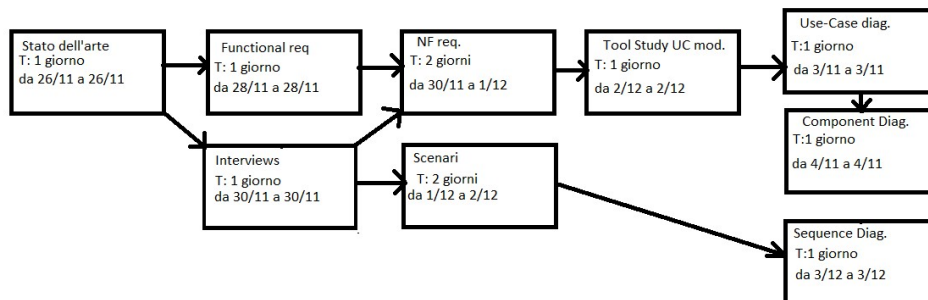
<Report in this section how the architectural and low level design you produced satisfies the FRs and the NFRs>

G. Effort Recording



PERT

Make a PERT documenting the tasks and timing you expect to spend on the deliverable. Try to be as precise as possible. Check, after the deliverable deadline, if and how you satisfied (or not) the deadlines.



Logging

As you are working on the assignment, record what you are doing and how long you spent. As a rule of thumb, you should add a log entry every time you switch tasks. For example, if you do something for two hours straight, that can be one log entry. However, if you do two or three things in half an hour, you must have a log entry for each of them. You do not need to include time for logging, but should include the time spent answering the other parts of this question.

For this purpose, please use the **LogTemplate.xls** file.

Categorization

When logging the time spent on the project, please create different sub- categories. Specifically, it is important to clearly distinguish between two main categories: the time spent for **“learning”** (the modeling languages, the tools, etc.) from the time needed for **“doing”** (creating the models, taking the decisions, ...). Learning tasks are in fact costs to be paid only once, while doing costs are those that will be repeated through the project. For each category, please define sub-categories. Examples follow. You may add other sub-categories you find useful.

Learning

- Requirements Engineering

Doing:

- Requirements discovery

- **Non functional Requirements**
- **Use Case Diagrams**
- **Tool study**
- **Requirements Modeling (UC diagrams)**

Summary Statistics

Based on the attributes defined above, calculate the summary statistics of the time spent for “learning”, the time spent for “doing”, and the total time.

Note: this Deliverable report shall document only the Summary Statistics for the different deliverables (D1, D2, and Final). Detailed information shall be reported in the Excel file.

When (Month/Day)		Time spent	Partners <i>(please report how many people have been working)</i>	Brief Description of the performed task	Category	Sub-Category
11	22	2	Group	Brainstorming	Learning	
11	26	2	Group	challenging task	Learning	
11	27	2	Group	stato dell'arte e req.eng.	Learning/Doing	req. Engineering
11	28	3	Group	challenging task, servizi	Learning/Doing	req. Discovery
11	30	4	Group	NR, interviews	Learning/Doing	req. Discovery
12	2	4	Group	scenari e use-case	Doing	
12	3	3	Group	component e revisione	Doing	
12	5	3	Group	generale	Doing	

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>