**System Design**

1. **Introduzione**
   1. **Obiettivi del sistema**

Il sistema che si intende realizzare è un sito di job-placement: “EasyJob.it”.   
Sulla nostra piattaforma si possono pubblicare annunci di lavoro, ognuno dei quali è caratterizzato da un titolo, una descrizione del profilo ideale per l’occupazione, dei tag che esprimono l’ambito lavorativo, i requisiti necessari ed il tipo di contratto. Gli annunci possono essere pubblicati dagli utenti registrati come azienda e possono essere invece ricercati (tramite tag e città) e visualizzati da qualsiasi utente. Le aziende sono caratterizzate dal nome, il logo, l’e-mail, l’username, la password, l’indirizzo, la data di fondazione, il numero di dipendenti attuali, la partita IVA ed una breve descrizione.  
A questi annunci si possono candidare gli utenti registrati come inoccupati. Gli inoccupati sono caratterizzati da nome, cognome, e-mail, username, password, data di nascita, città di residenza, indirizzo e il documento del loro curriculum.  
Ogni utente registrato ha a disposizione una pagina di profilo: quella dell’inoccupato gli consente di visualizzare l’elenco delle candidature effettuate, modificare e visualizzare il proprio curriculum e gli consente inoltre di visualizzare le notifiche; quella dell’azienda invece gli consente di visualizzare l’elenco dei candidati per ogni annuncio pubblicato, di visualizzare i loro curriculum e di inviare messaggi per proporre colloqui ai candidati.  
La gestione del sito è affidata all’amministratore e al moderatore degli annunci.  
L’amministratore ha il compito di rimuovere utenti che, al seguito di una o più segnalazioni, non rispettano il regolamento del sito, mentre il moderatore degli annunci rimuove gli annunci qualora non siano conformi al regolamento e segnala gli utenti all’amministratore.

1. **Design Goals**
   1. **Criteri di affidabilità**

* I dati personali (curriculum) degli inoccupati sono visibili alle aziende solamente dietro candidatura ad un annuncio.
* Eventuali inserimenti di input non validi sono opportunamente segnalati all’utente, il quale ha la possibilità di effettuare il re-inserimento.
  1. **Criteri di usabilità**
* L’interfaccia si adatta a qualsiasi dispositivo.
* Ogni sezione del sito è raggiungibile in non più di 5 click.

1. **Definizioni, acronimi ed abbreviazioni**
   * RAD: Requirements Analysis Document.
   * SDD: System Design Document.
   * ODD: Object Design Document.
   * DB: Database.
   * MVC: Model View Control.
   * Greenfield engineering: Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall’utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell’utente.
2. **Riferimenti**

* Bernd Bruegge & Allen H. Dutoit, Object Oriented Software Engineering: Using UML, Patterns and Java, (2nd Prentice Hall, 2003).
* www.linkedin.com

1. **Architettura del sistema e dei sottosistemi**
   1. **Overview**

Il sistema proposto è un’applicazione web, distribuita secondo il modello client server. L’obiettivo che si pone è quello di offrire un luogo di incontro efficiente e semplice per chi offre lavoro e chi lo domanda.

* 1. **Decomposizione in sottosistemi**

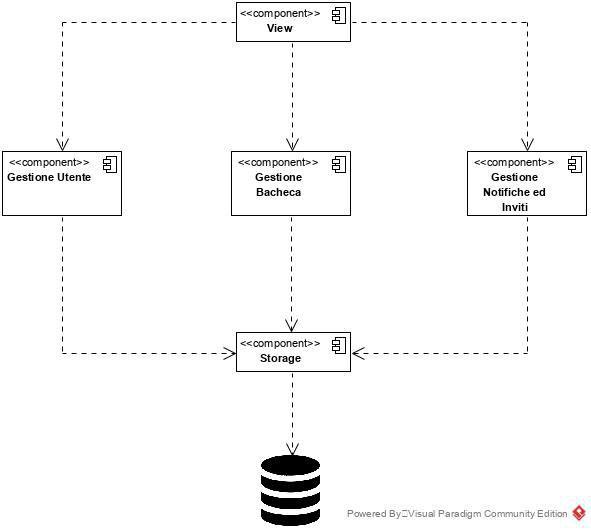
Il sistema è stato scomposto in tre sottosistemi al fine di raggiungere l’obiettivo di minimizzare l’accoppiamento e garantire una forte coesione.

I sottosistemi sono divisi per tipo di funzionalità (dove facciamo riferimento nel documento RAD, nella sezione 1 di “Requisiti”):

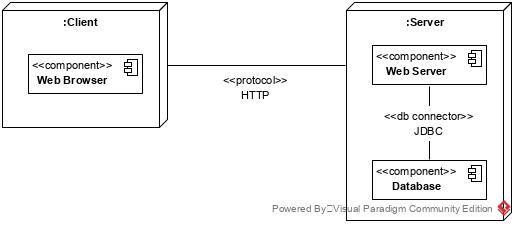
* Gestione Utente: Comprende tutte le funzionalità riguardanti gli utenti, e corrispondono ai requisiti funzionali: RF4, RF5, RF6, RF10, RF19
* Gestione Bacheca: Comprende le funzionalità sia per gli annunci che per le candidature. Le funzionalità corrispondo ai requisiti funzionali: RF1, RF2, RF3,

RF7, RF8, RF11, RF12, RF14, RF15, RF16, RF17.

* Gestione Notifiche ed Inviti: Comprende le funzionalità di segnalazioni, notifiche ed inviti a colloquio. Le funzionalità corrispondo ai requisiti: RF9, RF13, RF18, RF20.



* 1. **Diagramma di Deployment**
  2. **Mapping Hardware-Software**



Il sistema che si vuole realizzare sarà installato su una macchina che fungerà da server, sarà quindi provvista di un Web Server ed un DBMS, rispettivamente verrà usato Apache Tomcat e MySQL.

Il client sarà un qualsiasi personal computer provvisto di un browser ed una connessione ad internet.

1. **Gestione persistente dei dati**

Per la persistenza dei dati, si è pensato di utilizzare un database piuttosto che un file, la scelta è basta sul fatto si necessita di un’organizzazione logica dei dati, di sicurezza per l’utilizzo e gestione dei dati sensibili.

Inoltre la scelta è influenzata dal fatto che i dati sono accessibili in concorrenza da più utenti che li richiedono.

I dati che saranno resi persistenti sono (Riferimenti RAD, sezione: “Class Diagram”):

* Inoccupato, Azienda, Moderatore Annunci, Amministratore
* Annunci
* Segnalazioni
* Invito
* Candidature