



**UNIVERSITÀ DEGLI STUDI DI SALERNO**

ANNO ACCADEMICO 2016/2017



# **System Design Document**

Gestione dei Dati Persistenti

*Versione 1.0*

**TOP MANAGER:**

Prof. Andrea De Lucia

**PROJECT MANAGER:**

Antonio Luca D'Avanzo

Fabiano Pecorelli



## Top Manager:

| Nome                  |
|-----------------------|
| Prof. De Lucia Andrea |

## Project Manager:

| Nome                  | Matricola   |
|-----------------------|-------------|
| Antonio Luca D'Avanzo | 051210 2502 |
| Fabiano Pecorelli     | 052250 0421 |

## Partecipanti:

| NOME                    | MATRICOLA   |
|-------------------------|-------------|
| Severino Ammirati       | 051210 2898 |
| Andrea Buonaguro        | 051210 2490 |
| Angelo Caputo           | 051210 2204 |
| Ferdinando D'Avino      | 051210 2360 |
| Paolo Di Filippo        | 051210 3120 |
| Alfredo Fiorillo        | 051210 1930 |
| Dario Galiani           | 051210 2276 |
| Giovanni Leo            | 051210 3062 |
| Fabricio Nicolas Madaio | 051210 2840 |
| Vincenzo Noviello       | 051210 3198 |

|                  |             |
|------------------|-------------|
| Andrea Sarto     | 051210 2912 |
| Lino Sarto       | 051210 2348 |
| Giorgio Vitiello | 051210 2318 |

## Revision History:

| Data       | Version<br>e | Descrizione                        | Autore          |
|------------|--------------|------------------------------------|-----------------|
| 19/11/2016 | 1.0          | Stesura del System Design Document | Membri del team |

## Indice

|  |          |
|--|----------|
| <b>3.4 Gestione dei dati persistenti</b> | <b>5</b> |
| 3.4.1 Strategia di implementazione       | 5        |
| 3.4.2 Class Diagram                      | 6        |
| 3.4.3 Descrizione Entity                 | 8        |

## 3.4 Gestione dei dati persistenti

### 3.4.1 Strategia di implementazione

Per la memorizzazione dei dati si è scelto di utilizzare un Database relazionale che tramite il concetto di tabelle e relazioni permette di gestire in modo ottimale l'archiviazione delle informazioni di varie entità del sistema e soprattutto permette una facile interrogazione per poter ottenere le informazioni.

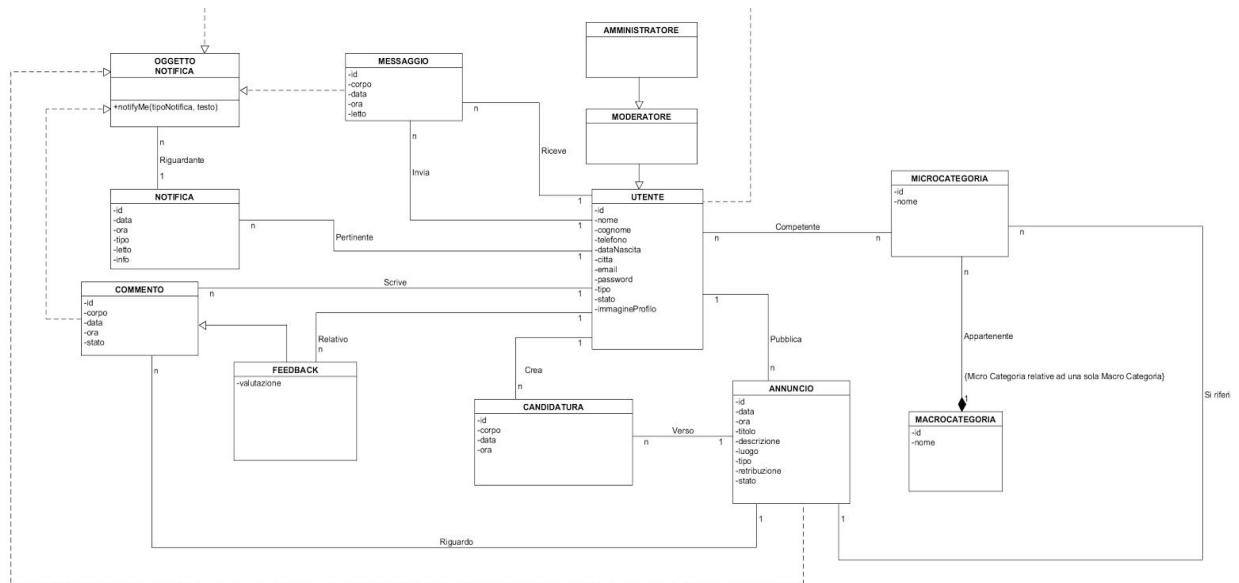
Le tecnologie utilizzate sono, per quanto riguarda il linguaggio di interrogazione, MySQL nella versione 5.6, mentre è stato scelto il Database Management System PhpMyAdmin alla versione 4.7.1.

Si è scelto di utilizzare un DBMS per rendere la base di dati gestibile anche da un amministratore non avvezzo ai sistemi informatici complessi grazie ad un'interfaccia utente molto semplice da utilizzare, rispetto alla visione del codice sorgente di ogni tabella.

Un DBMS inoltre garantisce:

- Sicurezza: grazie ad un sistema di autorizzazioni che per ogni utente stabilisce permessi di lettura, scrittura e modifica.
- Integrità: ovvero la gestione di più utenti contemporaneamente, impedendo, ad esempio, la modifica contemporanea di un dato.
- Supporto alle interrogazioni: permette di usare linguaggi di interrogazione mediante query per poter prelevare, aggiornare o inserire i dati.

### 3.4.2 Class Diagram



**Utente:** questa classe contiene tutte le informazioni relative ad un utente registrato. L'enumerazione chiarisce la tipologia dell'utente che può variare in Moderatore e Amministratore.

**Annuncio:** questa classe contiene le informazioni necessarie per la memorizzazione di un annuncio. La creazione di quest'ultimo è stata effettuata da un utente registrato. Attraverso l'associazione "Si riferisce", l'annuncio viene collegato ad una Microcategoria, in questo modo può essere ricercato da un Utente quando quest'ultimo cerca annunci relativi a quella Microcategoria.

**Commento:** questa classe contiene le informazioni relative a un commento lasciato ad un annuncio, da parte di un utente, allo scopo di fornire informazioni aggiuntive all'annuncio, pubbliche a tutti gli interessati a quest'ultimo, possibilmente richiesti dagli interessati stessi o inserite arbitrariamente dal creatore dell'annuncio.

**Messaggio:** la classe Messaggio mantiene le informazioni relative ad un Messaggio inviato da un Utente. Questo viene collegato ad Oggetto Notifica in quanto viene visualizzato dall'Utente destinatario, appunto, come una notifica.

**Notifica:** la classe Notifica mantiene le informazioni relative alle notifiche dell'utente. Ogni notifica quindi ha un riferimento con la classe Utente per identificare chi è il proprietario della notifica.

**Microcategoria:** oggetto che rappresenta una Microcategoria, questa risulta essere un insieme di sottoclassi di una Macrocategoria. Ogni Microcategoria ha quindi il riferimento alla Macrocategoria di appartenenza.

**Macrocategoria:** oggetto che rappresenta una Macrocategoria, ovvero una classe principale di interessi che fanno riferimento ad un annuncio oppure ad una competenza base di un utente.

**Candidatura:** la classe candidatura indica il legame che si crea tra un utente che si candida all'annuncio e l'annuncio stesso. Oltre ai riferimenti dei due elementi chiave (utente, annuncio) la

candidatura è composta anche da due enum che indicano lo stato della collaborazione tra il creatore dell'annuncio e l'utente candidato allo stesso. Più precisamente queste l'enum "richiesta inviata" può assumere 3 valori

- **non\_valutata:** la non valutazione, non aver ancora preso alcuna decisione.
- **inviata:** l'accettazione della stessa, che indica l'offerta del lavoro del datore al candidato.
- **non\_inviata:** il rifiuto della candidatura, quindi la volontà di non offrire il lavoro al candidato.

e da parte del candidato

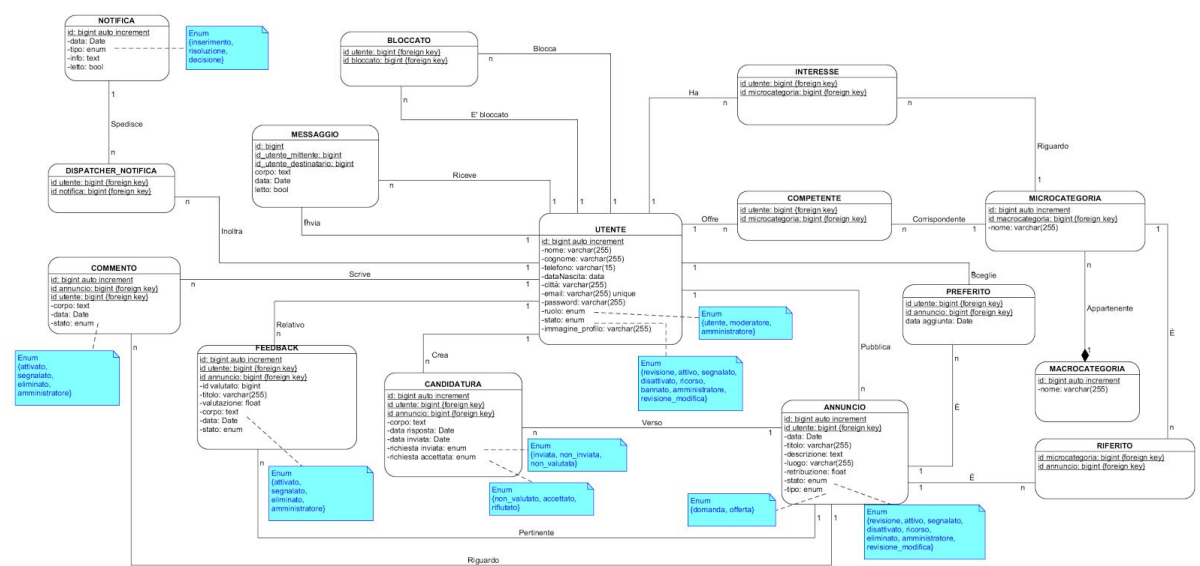
- **non\_valutata:** la non valutazione della proposta, non aver ancora preso alcuna decisione.
- **accettata:** l'accettazione della stessa, che indica la volontà di accettare l'offerta e iniziare una collaborazione.
- **rifiutata:** il rifiuto dell'offerta, quindi non accettare l'offerta di lavoro.

Tutto ciò è stato mappato per venire incontro alla necessità del rendere i feedback il più possibile veritieri, dando la possibilità di lasciare questi ultimi solo quando la è effettivamente avvenuta una collaborazione.

**Feedback:** La classe feedback indica una valutazione relativa ad un utente, data di chi usufruisce di un servizio, in base alla qualità dello stesso o da chi fornisce il servizio, in base alla qualità dell'esperienza lavorativa con il datore. La valutazione consiste in una breve descrizione della collaborazione, un voto da 1 a 5, con intervallo di 0.5, che sintetizza la qualità dell'esperienza lavorativa, il riferimento all'utente creatore del feedback e ovviamente quello relativo all'utente valutato.



3.4.3 Descrizione Entity



| UTENTE |         |
|--------|---------|
| Campo  | Vincolo |

|                  |   |
|------------------|---|
| id               | bigint auto_increment, primary key  |
| nome             | lunghezza massima: 255 caratteri, not null  |
| cognome          | lunghezza massima: 255 caratteri, not null  |
| telefono         | lunghezza massima: 15 caratteri   |
| dataNascita      | Date, not null  |
| città            | lunghezza massima: 255 caratteri, not null  |
| email            | lunghezza massima: 255 caratteri, not null  |
| password         | lunghezza massima: 255 caratteri, not null  |
| stato            | Enum: 'revisione', 'attivo', 'segnalato', 'disattivato', 'bannato', 'ricorso', 'amministratore', 'revisione_modifica' |
| partiva_iva      | lunghezza massima: 255 caratteri  |
| ruolo            | Enum: 'utente', 'moderatore', 'amministratore'  |
| immagine_profilo | lunghezza massima: 255 caratteri, not null  |

| FEEDBACK    |  |
|-------------|--|
| Campo       | Vincolo  |
| id          | bigint auto_increment, primary key                           |
| id_utente   | bigint, foreign key{UTENTE},primary key                      |
| Id_annuncio | bigint, foreign key{ANNUNCIO},primary key                    |
| id_valutato | bigint   |
| titolo      | varchar(255), not null                                       |
| valutazione | lunghezza massima: 255 caratteri, not null                   |
| corpo       | text: lunghezza massima(2 <sup>16</sup> -1), not null        |
| data        | Datetime, not null   |
| stato       | Enum: 'attivato', 'segnalato', 'amministratore', 'eliminato' |

| ANNUNCIO     |   |
|--------------|---|
| Campo        | Vincolo   |
| id           | bigint auto_increment, primary key  |
| id_utente    | bigint, foreign key{UTENTE},primary key   |
| data         | Date, not null  |
| titolo       | lunghezza massima: 255 caratteri, not null  |
| descrizione  | text: lunghezza massima(2^16 -1), not null  |
| luogo        | lunghezza massima: 255 caratteri, not null  |
| stato        | Enum: 'revisione', 'attivo', 'segnalato', 'disattivato', 'ricorso', 'eliminato', 'amministratore', 'revisione_modifica' |
| retribuzione | float   |
| tipologia    | Enum: 'domanda', 'offerta'  |

| MICROCATEGORIA    |  |
|-------------------|--|
| Campo             | Vincolo  |
| id                | bigint auto_increment, primary key               |
| nome              | lunghezza massima: 255, not null                 |
| id_macrocategoria | bigint, foreign key{MACROCATEGORIA}, primary key |

| MACROCATEGORIA |                                    |
|----------------|------------------------------------|
| Campo          | Vincolo                            |
| id             | bigint auto_increment, primary key |
| nome           | lunghezza massima: 255, not null   |

| MESSAGGIO              |  |
|------------------------|--|
| Campo                  | Vincolo                                    |
| id                     | bigint auto_increment, primary key         |
| data                   | Date, not null                             |
| corpo                  | text: lunghezza massima (2^16 -1) not null |
| letto                  | Boolean, not null                          |
| id_utente_mittente     | bigint, foreign key{UTENTE},primary key    |
| id_utente_destinatario | bigint, foreign key{UTENTE},primary key    |

| COMMENTO    |  |
|-------------|--|
| Campo       | Vincolo  |
| id          | bigint auto_increment, primary key                           |
| id_annuncio | bigint, foreign key{ANNUNCIO},primary key                    |
| id_utente   | bigint, foreign key{UTENTE},primary key                      |
| corpo       | text: lunghezza massima (2^16 -1), not null                  |
| data        | Date, not null   |
| stato       | Enum: 'attivato', 'segnalato', 'amministratore', 'eliminato' |

| NOTIFICA |  |
|----------|--|
| Campo    | Vincolo  |
| id       | bigint auto_increment, primary key                       |
| letto    | Boolean, not null  |
| data     | Date, not null   |
| tipo     | Enum:<br>'inserimento',<br>'risoluzione',<br>'decisione' |
| info     | text: lunghezza massima (2^16 -1), not null              |

| DISPATCHER_NOTIFICA |  |
|---------------------|--|
| Campo               | Vincolo                                    |
| id_utente           | bigint, foreign key{UTENTE}, primary key   |
| id_notifica         | bigint, foreign key{NOTIFICA}, primary key |

| CANDIDATURA         |  |
|---------------------|--|
| Campo               | Vincolo  |
| id                  | bigint auto_increment, primary key             |
| id_utente           | bigint, foreign key{UTENTE}, primary key       |
| id_annuncio         | bigint, foreign key{ANNUNCIO}, primary key     |
| corpo               | text, not null                                 |
| data_risposta       | Datetime                                       |
| data_inviata        | Datetime                                       |
| richiesta_inviata   | Enum: 'inviata', 'non_inviata', 'non_valutata' |
| richiesta_accettata | Enum: 'non_valutato', 'accettato', 'rifiutato' |

| INTERESSE         |  |
|-------------------|--|
| Campo             | Vincolo  |
| id_utente         | bigint, foreign key{UTENTE}, primary key         |
| id_microcategoria | bigint, foreign key{MICROCATEGORIA}, primary key |

| COMPETENTE        |  |
|-------------------|--|
| Campo             | Vincolo  |
| id_utente         | bigint, foreign key{UTENTE}, primary key         |
| id_microcategoria | bigint, foreign key{MICROCATEGORIA}, primary key |

| PREFERITO |
|-----------|
|-----------|

| Campo         | Vincolo                                   |
|---------------|---|
| id_utente     | bigint, foreign key{UTENTE},primary key   |
| id_annuncio   | bigint, foreign key{ANNUNCIO},primary key |
| data_aggiunta | Datetime, not null                        |

| RIFERITO          |   |
|-------------------|---|
| Campo             | Vincolo   |
| id_microcategoria | bigint, foreign key{MICROCATEGORIA},primary key |
| id_annuncio       | bigint, foreign key{ANNUNCIO},primary key       |

| BLOCCATO           |   |
|--------------------|---|
| Campo              | Vincolo                                 |
| id_utente          | bigint, foreign key{UTENTE},primary key |
| id_utente_bloccato | bigint, foreign key{UTENTE},primary key |