



Laurea Magistrale in informatica-Università di Salerno  
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci



# SDD: System Design Document

IFY Internship For You

Riferimento	
Versione	1.0
Data	08/01/2020
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Roberto Calabrese, Giusy Castaldo, Geremia Cavezza, Benedetta Coccoaro, Simone Civale, Carmine Ferrara, Giacomo Izzo, Alessia Natale
Approvato da	Anna Belardo, Rosanna Coccoaro



## Revision History

Data	Versione	Descrizione	Autori
28/11/2019	0.1	Aggiunta diagramma di deployment	Coccaro Benedetta, Cavezza Geremia
28/11/2019	0.1	Aggiunta Mapping hardware/software	Civale Simone
28/11/2019	0.1	Aggiunta Controllo degli accessi e sicurezza e condizioni limite	Calabrese Roberto
29/11/2019	0.1	Aggiunta Individuazione sottosistemi e relativi servizi	Ferrara Carmine, Castaldo Giusy
29/11/2019	0.1	Aggiunta Gestione dati persistenti	Natale Alessia, Civale Simone
29/11/2019	0.1	Aggiunta Architetture simili	Natale Alessia
29/11/2019	0.1	Aggiunta Design Goals e Trade-offs	Izzo Giacomo
04/12/2019	0.2	Modifica gestione dati persistenti e individuazione dei sottosistemi	Natale Alessia, Civale Simone, Ferrara Carmine, Castaldo Giusy
05/12/2019	0.2	Modifica condizione limite	Calabrese Roberto
11/12/2019	0.3	Modifica Architettura del sistema e sottosistemi individuati	Castaldo Giusy, Ferrara Carmine
08/01/2020	1.0	Modifiche e correzioni design goals e trade-off, modifiche formattazione e revisione generale	Coccaro Benedetta, Izzo Giacomo, Giusy Castaldo



## Sommario

Revision History .....	2
1. Introduzione .....	4
1.1 Obiettivo del sistema .....	4
1.2 Design Goals e Trade-off .....	4
1.2.1 Design Goals .....	4
1.2.2 Design Trade-off .....	7
1.3 Definizioni, Acronimi e Abbreviazioni .....	7
1.4 Riferimenti .....	9
1.5 Panoramica .....	9
2. Architettura dei sistemi simili .....	9
3. Architettura del sistema proposto .....	10
3.1 Panoramica .....	10
3.2 Decomposizione in sottosistemi .....	10
3.2.1 Architettura del sistema e sottosistemi individuati .....	10
3.2.2 Deployment Diagram .....	13
3.3 Mapping hardware/software .....	13
3.4 Gestione Dati Persistenti .....	14
3.5 Controllo degli accessi e sicurezza .....	18
3.6 Controllo del flusso globale del sistema .....	20
3.7 Condizioni limite .....	20
4. Servizi dei sottosistemi .....	24



## 1. Introduzione

### 1.1 Obiettivo del sistema

Il sistema proposto ha come scopo principale la gestione digitale dei processi di convenzionamento aziendale e di avvio di tirocini esterni per studenti iscritti ai corsi di laurea triennali e magistrali del Dipartimento di Informatica dell'Università degli Studi di Salerno.

Il nostro team si pone come obiettivo principale la realizzazione di un sistema che permetta di velocizzare il processo burocratico attuale che causa, spesso, tempi di attesa troppo lunghi e una vasta dispersione di documenti da presentare dando, in questo modo, una visuale chiara e concisa delle informazioni necessarie ad ogni tipologia di utenza coinvolta nel dominio di progetto.

Il sistema progettato sarà un'applicazione Web che dovrà consentire l'accesso a 3 tipologie di utenza (studente del Dipartimento di Informatica, Delegato Aziendale, Responsabile Ufficio Tirocini), ciascuna con funzionalità a disposizione previste nella fase di analisi e raccolta dei requisiti.

Dallo studio fatto in fase di analisi, possiamo dividere il sistema in 5 categorie:

- Gestione dell'utenza;
- Gestione delle convenzioni;
- Gestione delle iscrizioni degli studenti;
- Gestione progetti formativi;
- Gestione delle domande di tirocinio.

### 1.2 Design Goals e Trade-off

#### 1.2.1 Design Goals

Priorità	ID Design Goal	Descrizione design goal	Categoria	Origine
Alta	DG_1	Affidabilità: La visualizzazione delle informazioni sullo stato delle richieste di iscrizione, convenzionamento e delle domande di tirocinio deve essere attendibile; ossia deve rispecchiare la situazione reale all'interno del Dipartimento di Informatica in base agli ultimi dati aggiornati.	Affidabilità	RNF_A5
Alta	DG_2	Robustezza: Il sistema controllerà la validità degli input immessi dall'utente e segnalerà eventuali input non validi con opportuni messaggi	Affidabilità	RNF_A3



		d'errore. I controlli verranno effettuati sia nel layer di presentazione(View) sia nel layer di controllo(Control).		
Alta	DG_3	<p>Usabilità:</p> <p>Un' interfaccia grafica intuitiva, che si adatta alle aspettative dell'utente garantirà un alto grado di usabilità; l'interfaccia sarà corredata inoltre da piccoli testi e immagini contestuali, che guideranno l'utente nelle scelte da fare in ogni punto della navigazione.</p>	Utente finale	RNF_U1 RNF_U2 RNF_U4
Alta	DG_4	<p>Estensibilità:</p> <p>Il sistema è progettato per garantire l'aggiunta di nuove funzionalità in maniera agevole. A tale scopo le componenti saranno progettate in maniera modulare e indipendente, l'architettura del sistema segue il modello a layers chiusi MVC (Model, View, Controller) e l'implementazione delle componenti farà uso di design patterns appropriati alla realizzazione del criterio.</p>	Manutenzione	RNF_S3
Alta	DG_5	<p>Tempo di risposta:</p> <p>Per la visualizzazione delle domande di tirocinio, delle richieste di convenzionamento e delle richieste di iscrizione il tempo di risposta deve essere di 1 secondo.</p> <p>Per l'invio delle richieste di convenzionamento e delle richieste di iscrizione il tempo di risposta deve essere di 3 secondi.</p> <p>Per l'accettazione o il rifiuto di richieste in attesa il tempo di risposta deve essere di 2 secondi.</p>	Prestazione	RNF_P2
Alta	DG_6	<p>Tracciabilità:</p> <p>Il sistema utilizza un meccanismo di tracciabilità che garantisce l'identificazione delle dipendenze di ogni tipologia di componente dai relativi requisiti, utilizzando una notazione convenzionale per l'identificazione delle</p>	Manutenzione	Documenti di management



		componenti e delle matrici di tracciabilità per evidenziarne le dipendenze.		
Media	DG_7	Disponibilità: Il sistema deve essere disponibile in maniera continuativa, senza nessuna interruzione dei servizi, per tutte le tipologie di utenza.	Affidabilità	Dominio applicativo
Media	DG_8	Modificabilità: Il sistema sarà progettato tenendo a mente la necessità di cambiamenti futuri delle funzionalità. Le componenti saranno progettate in maniera modulare, leggibile ed utilizzando un'architettura a layers indipendenti (MVC), debolmente accoppiati.	Manutenzione	Dominio applicativo
Media	DG_9	Utilità: Il sistema fornirà un supporto consistente al lavoro dell'utente offrendo i vantaggi di un sistema digitale per la raccolta, gestione e visualizzazione delle informazioni e rappresentando un mezzo di connessione tra le parti notevolmente più veloce ed efficiente.	Utente finale	Dominio applicativo
Media	DG_10	Leggibilità: In concordanza con gli obiettivi di estendibilità e modificabilità il codice del sistema sarà prodotto per avere un alto grado di leggibilità, seguendo linee guida che prevedono l'utilizzo di convenzioni per i nomi di metodi, classi e interfacce, la produzione di un'adeguata documentazione per il codice e l'utilizzo di strumenti automatici per il controllo della leggibilità.	Manutenzione	RNF_S2
Bassa	DG_11	Memoria: La dimensione complessiva del sistema dipende dalla memoria utilizzata per il mantenimento del database.	Prestazione	Dominio applicativo



Bassa	DG_12	Portabilità: Il sistema sarà portabile perché utilizza protocolli, tecnologie e sistemi esterni indipendenti dal tipo di piattaforma in cui è installato. Il sistema è indipendente da servizi specifici del sistema operativo.	Manutenzione	RNF_I1 RNF-I2
Bassa	DG_13	Costi di sviluppo I costi di sviluppo previsti ammontano a 400 ore per la progettazione e lo sviluppo (50 ore per ogni team member).	Costo	Documenti di management

### 1.2.2 Design Trade-off

#### Memoria - Tempo di Risposta

Una frequenza elevata di richieste al sistema potrebbe produrre una diminuzione nei tempi di risposta, compromettendo la qualità del servizio. Una possibile soluzione al problema prevede l'allocazione di più memoria centrale, per minimizzare i tempi di accesso. Il team di analisi e sviluppo ha scelto di privilegiare il tempo di risposta, per garantire sempre un'alta qualità del servizio.

#### Costo - Robustezza

Il controllo e la gestione di input invalidi da parte dell'utente (Robustezza) è considerata una caratteristica ad elevata priorità, perché affine all'obiettivo generale del sistema di costruire un'alternativa digitale al metodo cartaceo, che sia affidabile ed efficiente. Di conseguenza nell'analisi del trade-off tra criterio di Robustezza e Costo di sviluppo è stata data precedenza al primo.

#### Tempo di Consegna - Funzionalità

La necessità da parte del cliente di disporre di un sistema funzionante in tempi brevi e un'analisi delle priorità da parte del management ha orientato la pianificazione delle attività verso una riduzione dei tempi di progettazione a scapito dello sviluppo di funzionalità più complesse. Nel trade-off tra Funzionalità e Tempo di sviluppo è stata data precedenza alla necessità di contenere i tempi di sviluppo, per cui verranno implementate soltanto funzionalità a priorità media e alta.

#### Costi di sviluppo - Tracciabilità

La tracciabilità delle dipendenze tra requisiti e componenti è stata valutata con priorità alta dal team di analisi, perché consente di prevenire la propagazione di errori nel modello di sviluppo in fasi avanzate ed è al contempo una condizione necessaria per perseguire i criteri di manutenibilità a cui è stata associata priorità alta.

Di conseguenza nel trade-off tra costi di sviluppo e tracciabilità si è dato precedenza a quest'ultima.



#### Costo - Manutenzione

Nel trade-off tra costo e manutenzione è stata data precedenza a questo secondo criterio. Il team di management ha scelto di includere nei costi un meccanismo di tracciabilità delle componenti, il perseguimento di linee guida e buone pratiche di codifica e l'utilizzo di un approccio modulare con lo scopo di sviluppare un sistema che sia facilmente manutenibile.

### 1.3 Definizioni, Acronimi e Abbreviazioni

#### Definizioni

- Design Goal: Obiettivi di design progettati per il sistema proposto;
- Design Trade-off: Scelte e compromessi tra design goals dissonanti;
- Software web-based: Prodotto software progettato e sviluppato con metodologie e tecnologie per un'architettura web;
- Pattern MVC: Modello architetturale del sistema a tre livelli;
- Model: Layer del pattern architetturale per la gestione e memorizzazione dei dati persistenti;
- Controller: Layer del pattern architetturale per la gestione e il controller della logica di business;
- View: Layer del pattern architetturale per la gestione e controllo d'interfaccia tra le risorse del sistema e l'utente finale;
- Deployment Diagram: Diagramma UML di specifica per le relazioni tra le componenti realizzate (con relative tecnologie implementative) e le risorse Hardware e Software necessarie al corretto funzionamento del sistema;
- Varchar, Integer & Date: Tipologie di dati necessarie per la memorizzazione dei dati persistenti nella base dati;
- Access Control List: Metodologia di controllo degli accessi ai sottosistemi individuati in fase di design, dove per ogni sottosistema sono specificate coppie (Attore, Operazione) indicanti ognuna un'operazione disponibile ad un determinato attore del sistema;
- Tipologia event-driven: Tipologia di controllo del flusso del sistema basata sull'uso di eventi;
- Diagramma dei servizi: Diagramma di specifica dei servizi offerti dai sottosistemi progettati.

#### Acronimi

- RAD Requirement Analysis Document;
- DG Design Goal;
- MVC: Model View Controller;
- GC Gestione Convenzioni;
- GPF Gestione Progetti Formativi;
- GDT Gestione Domande Tirocinio;
- GT Gestione Tirocini;
- GU Gestione Utenza;
- US Use Case;
- SU Start Up;
- SD Shut Down.





## 1.4 Riferimenti

- IFY\_RAD\_v.1.1
- Slide di System Design, presenti sulla piattaforma del corso
- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition (Bernd Bruegge & Allen H. Dutoit).

## 1.5 Panoramica

- Capitolo 1: Contiene l'introduzione con l'obiettivo del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione dell'intera documentazione.
- Capitolo 2: Descrive le funzionalità offerte dal sistema corrente.
- Capitolo 3: Presenta l'architettura del sistema proposto, in particolare la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.
- Capitolo 4: Presenta i servizi dei sottosistemi.

## 2. Architettura dei sistemi simili

Al momento non esiste un sistema software per la gestione dei tirocini esterni relativo al Dipartimento di Informatica dell'Università degli Studi di Salerno. L'invio di una domanda di tirocinio risulta essere molto lenta e poco agevole poiché attualmente viene eseguita a mano con la compilazione di numerosi fogli cartacei. Grazie alla realizzazione del software "IFY Internship For You" web-based tale procedura diventa più veloce in quanto eseguita in maniera automatica.

Anche se non esiste per il Dipartimento di Informatica dell'Università degli Studi di Salerno un sistema software che gestisce tale funzionalità, esistono dei sistemi simili nel campo della gestione dei tirocini. Tra i sistemi che più si avvicinano al software "IFY Internship For You" ci sono:

- "Gestione tirocini" di Sistema Piemonte, un sistema che identifica in modo univoco chi accede e, in base al riconoscimento, mette a disposizione i servizi utili alla gestione di tirocini quali: acquisizione delle comunicazioni di assunzione, cessazione e relative rettifiche e annullamenti, compilazione del Progetto Formativo e vari altri servizi per i tirocinanti;
- "JobSoul", un sistema che si occupa della ricerca di tirocini disponibili con possibilità di candidarsi da parte degli studenti, mentre le aziende hanno un'area dedicata dove possono registrarsi e pubblicare le loro offerte di lavoro;
- "E-R Lavoro per te", un portale di servizi della Regione Emilia-Romagna, realizzato in collaborazione con le Province e i Centri per l'Impiego per rendere più semplice l'incontro fra chi offre e chi cerca lavoro e i servizi sono rivolti sia ai cittadini che alle imprese. Tale sistema ha a disposizione anche una piattaforma mobile per gestire le proprie funzionalità.

Ciascuno dei sistemi citati è accessibile tramite sito web e quindi utilizza un web browser; inoltre adopera un'architettura basata a tre livelli, come IFY, in quanto è necessaria la registrazione e/o il login con il conseguente accesso ad un model come un database che gestisce i suddetti dati dell'utente.

## 3. Architettura del sistema proposto

### 3.1 Panoramica

Il sistema da noi proposto è una nuova applicazione Web per la gestione di servizi nell'ambito dei tirocini esterni.

Al fine di garantire una buona coesione e un basso accoppiamento tra le funzionalità del sistema si sceglie un'architettura semplice che tenga traccia, in modo chiaro, del continuo cambiamento delle informazioni persistenti.

A tale scopo, si opta per uno stile architetturale di tipo repository, in particolare per la sua specializzazione: il pattern MVC; la scelta ricade su questa particolare tipologia di architettura in quanto mette a disposizione l'ottima logica di separazione tra i dati di business, la presentazione dei dati, delle funzionalità e la memorizzazione dei dati persistenti.

Questo stile architetturale permette di implementare le funzionalità previste nella fase di analisi; in particolare si prevede che l'utente potrà effettuare una richiesta di iscrizione alla piattaforma come studente oppure di convenzionamento come azienda, consultare le varie aziende convenzionate con i relativi progetti formativi, effettuare il login e il logout.

Inoltre, prevediamo diverse funzionalità mirate a seconda della tipologia di utente (studente, delegato aziendale, responsabile ufficio tirocini) che ha effettuato l'accesso alla piattaforma, nel dettaglio:

- lo studente, partendo dalla visualizzazione dei progetti formativi di un'azienda, potrà inoltrare una domanda di tirocinio e seguirne lo stato di avanzamento dalla sua dashboard;
- Il delegato aziendale potrà inserire e gestire progetti formativi inerenti alle attività di tirocinio che l'azienda intende fornire. Inoltre, potrà accettare o rifiutare le domande di tirocinio pervenute dagli studenti e successivamente visualizzarne l'evoluzione;
- Il responsabile dell'ufficio tirocini potrà visualizzare le domande di tirocinio pervenute dopo l'accettazione aziendale, gestirne la valutazione finale.

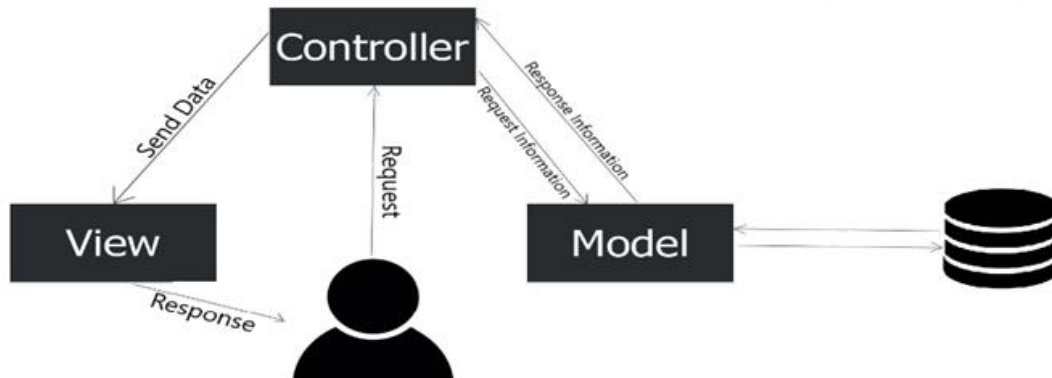
Ogni tipologia di utente registrato avrà, inoltre, una visualizzazione mirata al ruolo ricoperto nel sistema dei tirocini in corso.

### 3.2 Decomposizione in sottosistemi

#### 3.2.1 Architettura del sistema e sottosistemi individuati

Come specificato nella definizione teorica del pattern MVC, la decomposizione del nostro sistema è composta da tre layer aventi ciascuna aspetti e funzionalità differenti:

- View: layer di presentazione delle funzionalità del sistema e relativo interfacciamento con l'utenza;
- Controller: layer che si occupa della gestione della logica di business del sistema e relativi flussi di controllo dei dati;
- Model: layer che si occupa della gestione dei dati persistenti e relativo interscambio degli stessi tra i sottosistemi.



Per motivi di efficienza, abbiamo deciso di utilizzare un'architettura chiusa suddividendo l'intero sistema in sottosistemi.

Come analizzato precedentemente, si è deciso di utilizzare basso accoppiamento ed elevata coesione per la gestione di componenti singole in modo tale da garantire, in caso di modifiche, il minor numero di aggiornamenti tra i sottosistemi.

Da un'analisi degli oggetti Boundary individuati nella fase di analisi definiamo un sottosistema view:

- UserInterface (sottosistema per la gestione dell'interfacciamento con l'utente).

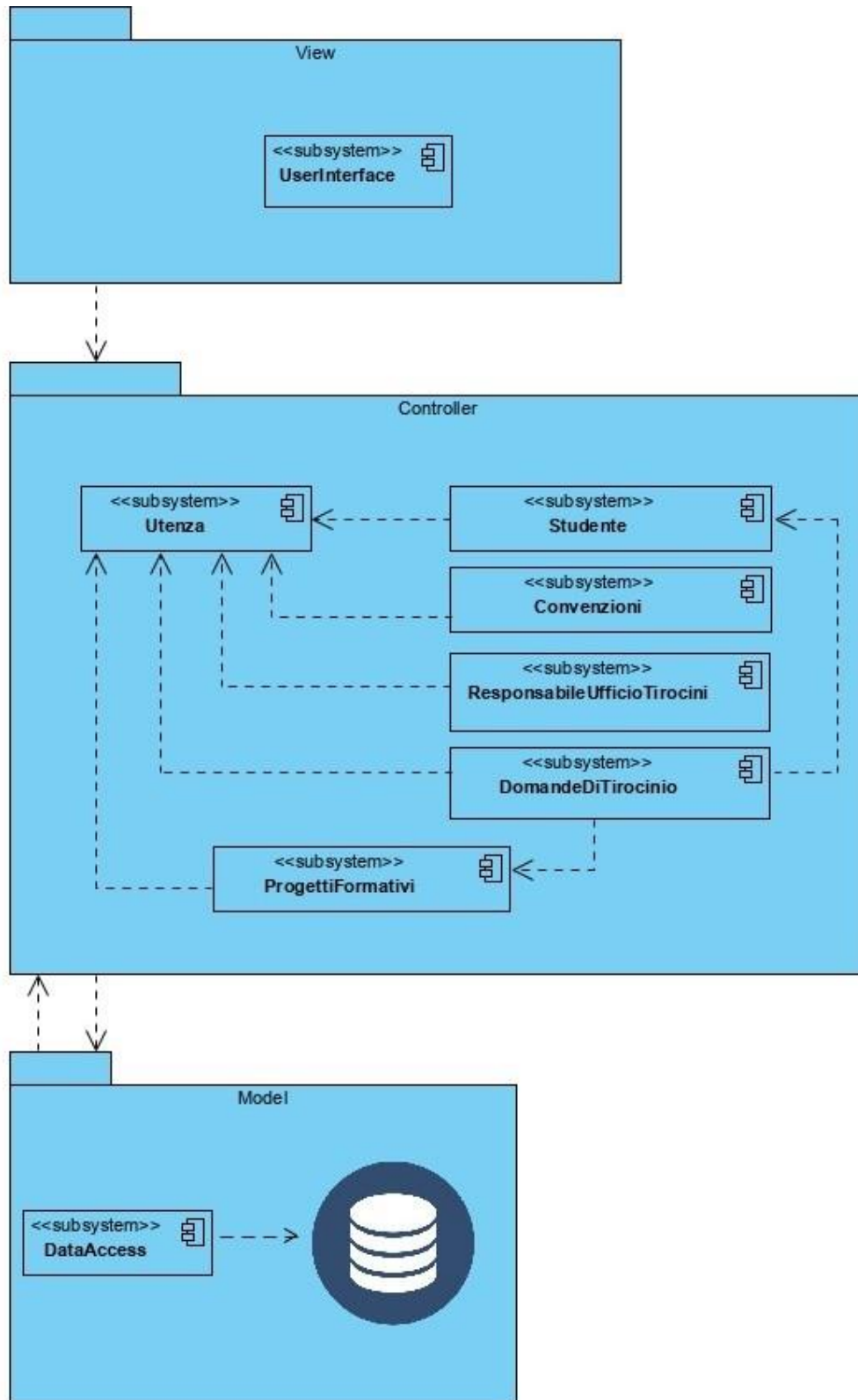
Da un'analisi degli oggetti Controller individuati nella fase di analisi definiamo sei sottosistemi:

- Utente (sottosistema per la gestione del login, logout degli utenti registrati; delle richieste di registrazione);
- Responsabile (sottosistema per la gestione dei dati del responsabile);
- Studente (sottosistema per la gestione delle richieste di registrazione);
- Convenzioni (sottosistema per la gestione delle convenzioni);
- DomandeDiTirocinio (sottosistema per la gestione delle domande di tirocinio);
- ProgettiFormativi (sottosistema per la gestione dei progetti formativi).

Da un'analisi degli oggetti Model individuati nella fase di analisi definiamo un sottosistema:

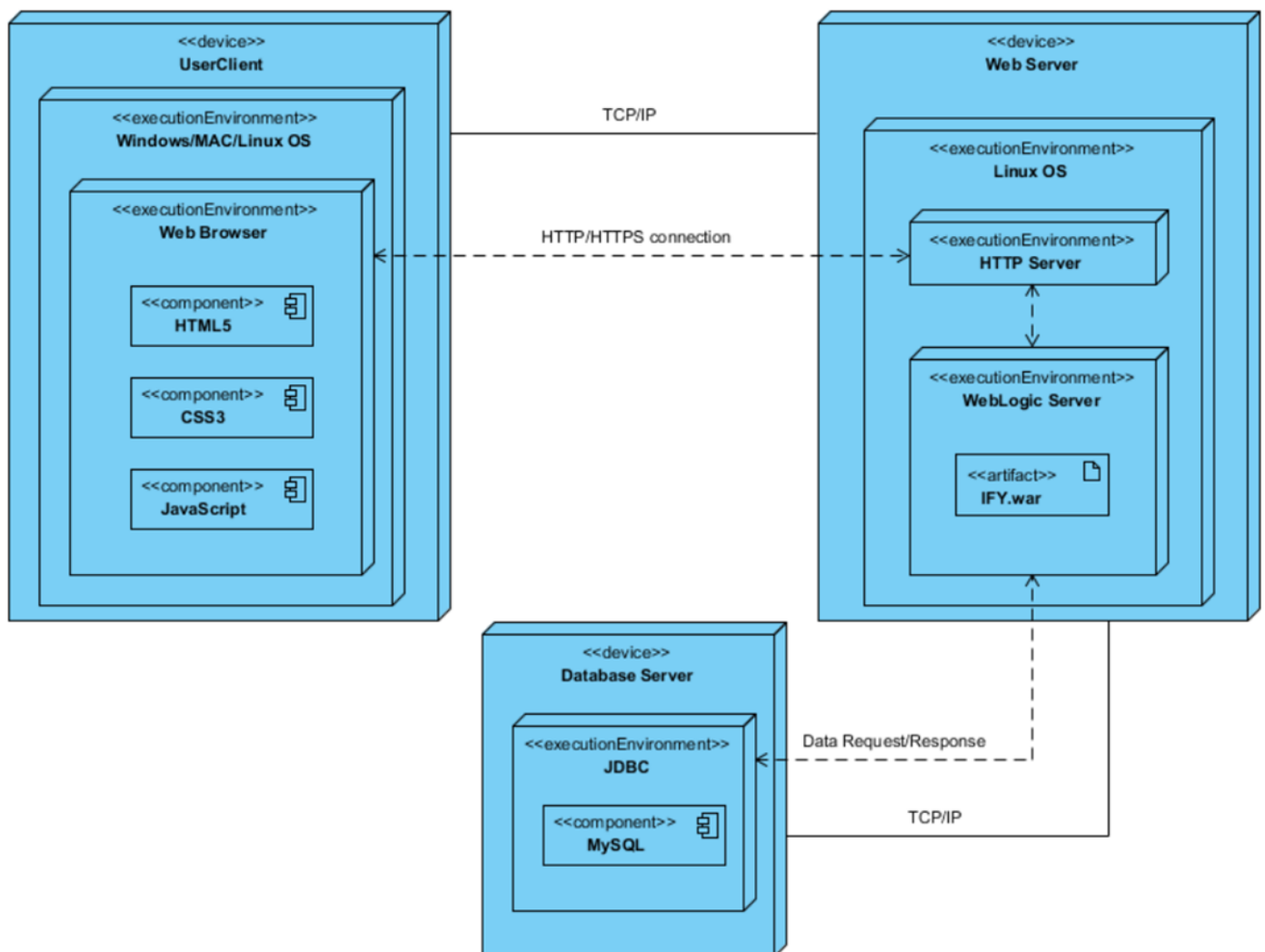
- DataAccess.

Dunque, il sistema si compone di 8 sottosistemi come mostrato in figura:



### 3.2.2 Deployment Diagram

Lo User Client richiede, tramite l'interfaccia, le funzionalità che il sistema mette a disposizione nel caso in cui si disponga di un Web Browser capace di eseguire codice JavaScript, HTML5 e CSS3. Lo User Client si connette, tramite il Web Browser, al Web Server che fornisce la view e la logica. Nella parte WebLogic vengono eseguite le funzioni apposite per il completamento delle richieste dello User Client fornendo anche la connessione al database Server.



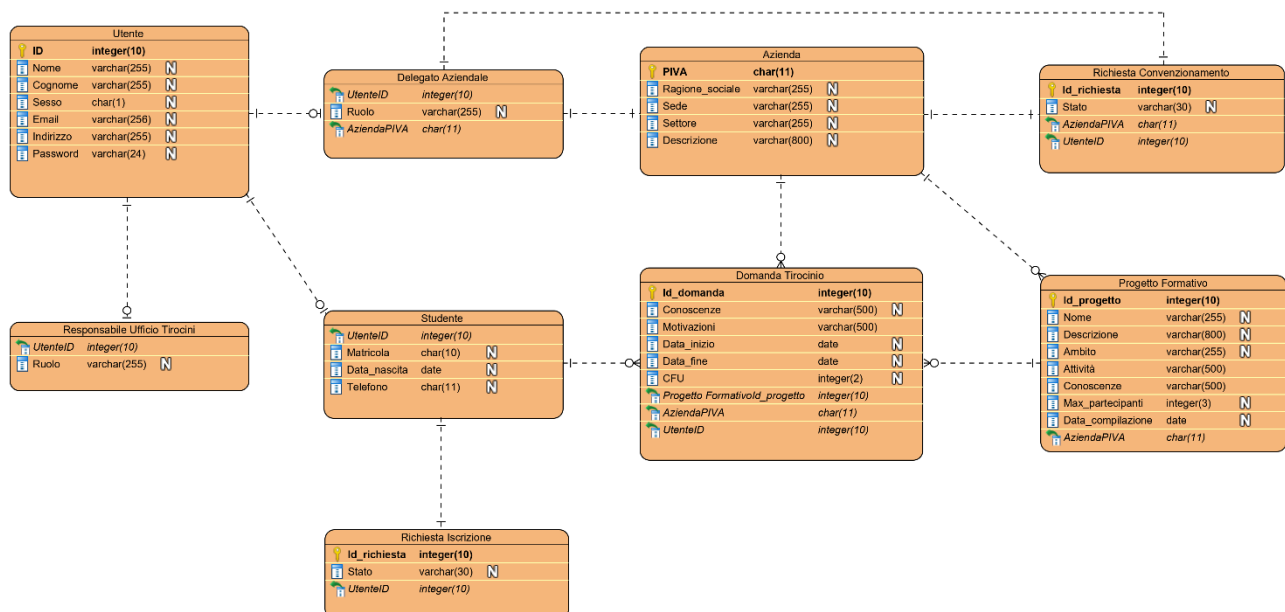
### 3.3 Mapping hardware/software

Il sistema utilizzerà una struttura hardware costituita da un Server che risponderà alle richieste inviate dai client. Il client è una qualsiasi macchina con una connessione Internet e un web server installato su di esso, in modo da potersi connettere al Server per interagire col sistema. Invece, il server è una qualsiasi macchina con una connessione Internet, dove è presente MySQL ed è capace di immagazzinare una certa

quantità di dati. Esso, inoltre, gestisce la logica e i dati persistenti contenuti nel database. La comunicazione tra il client e il server è basata sul protocollo HTTP a cui il client invierà delle richieste e il server provvederà alle risposte. La componente software necessaria è un DBMS per consentire la comunicazione con più client.

### 3.4 Gestione Dati Persistenti

Il sistema gestisce la memorizzazione di dati persistenti tramite l'utilizzo di un database relazionale per consentire un accesso veloce e sicuro ai dati e un ampio spazio di archiviazione. Grazie all'uso del DBMS i dati sono protetti quindi diversi utenti con differenti operazioni possono accedere a diverse sezioni del database. Viene anche garantito l'accesso concorrente ai dati affidabili salvandone una copia in modo da poter ripristinare lo stato del database in caso di danni software o hardware.



#### Utente

Nome	Tipo	Null	Key
ID	Integer(10)	Not null	Primary Key
Nome	Varchar(255)	Not null	
Cognome	Varchar(255)	Not null	
Sesso	Char(1)	Not null	



Email	Varchar(256)	Not null	
Indirizzo	Varchar(255)	Not null	
Password	Varchar(24)	Not null	

#### Studente

Nome	Tipo	Null	Key
Matricola	char(10)	Not null	
Data_nascita	Date	Not null	
Telefono	char(11)	Not null	
ID	Integer(10)	Not null	Foreign key

#### Delegato Aziendale

Nome	Tipo	Null	Key
Ruolo	Varchar(255)	Not null	
ID	Integer(10)	Not null	Foreign key
PIVA	char(11)	Not null	Foreign key

#### Responsabile Ufficio Tirocini

Nome	Tipo	Null	Key
Ruolo	Varchar(255)	Not null	
ID	Integer(10)	Not null	Foreign key



### Azienda

Nome	Tipo	Null	Key
PIVA	char(11)	Not null	Primary Key
Ragione_sociale	Varchar(255)	Not null	
Sede	Varchar(255)	Not null	
Settore	Varchar(255)	Not null	
Descrizione	Varchar(800)	Not null	

### Progetto Formativo

Nome	Tipo	Null	Key
Id_progetto	Integer(10)	Not null	Primary Key
Nome	Varchar(255)	Not null	
Descrizione	Varchar(800)	Not null	
Ambito	Varchar(255)	Not null	
Attività	Varchar(500)		
Conoscenze	Varchar(500)		
Max_partecipanti	Integer(3)	Not null	
Data_compilazione	Date	Not null	
PIVA	char(11)	Not null	Foreign Key





### Domanda Tirocinio

Nome	Tipo	Null	Key
Id_domanda	Integer(10)	Not null	Primary Key
Conoscenze	Varchar(500)	Not null	
Motivazioni	Varchar(500)		
Data_inizio	Date	Not null	
Data_fine	Date	Not null	
CFU	Integer(2)	Not null	
Id_progetto	Integer(10)	Not null	Foreign Key
ID	Integer(10)	Not null	Foreign Key
PIVA	char(11)	Not null	Foreign Key

### Richiesta Iscrizione

Nome	Tipo	Null	Key
Id_richiesta	Integer(10)	Not null	Primary Key
Stato	Varchar(30)	Not null	
ID	Integer(10)	Not null	Foreign Key

### Richiesta Convenzionamento

Nome	Tipo	Null	Key
Id_richiesta	Integer(10)	Not null	Primary Key
Stato	Varchar(30)	Not null	



PIVA	char(11)	Not null	Foreign Key
UtenteID (Delegato)	Integer(10)	Not Null	Foreign Key

### 3.5 Controllo degli accessi e sicurezza

In “IFY Internship For You” vi sono differenti attori ognuno avente l’autorizzazione di compiere differenti operazioni. Per riassumere al meglio il controllo degli accessi si è utilizzata il meccanismo dell’Access Control List, dove per ogni sottosistema è stata riportata una lista di coppie [Attore, Operazione] così da indicare le operazioni consentite ad ogni attore in uno specifico sottosistema.

Lista di controllo degli accessi per il sottosistema UtenzaController

Attore	Operazione
Utente Registrato	Login ()
Utente Registrato	Logout()

Lista di controllo degli accessi per il sottosistema StudenteController

Attore	Operazione
Ospite	InvioRichiestascrizione()
Responsabile Ufficio Tirocini	VisualizzaRichiestascrizione()
Responsabile Ufficio Tirocini	AccettaRichiestascrizione()
Responsabile Ufficio Tirocini	RifiutaRichiestascrizione()



Lista di controllo degli accessi per il sottosistema ConvenzioniController

Attore	Operazione
Ospite	InviaRichiestaConvenzionamento()
Utente	VisualizzaAziendeConvenzionate()
Responsabile Ufficio Tirocini	VisualizzaRichiesteConvenzionamento()
Responsabile Ufficio Tirocini	AccettaRichiestaConvenzionamento()
Responsabile Ufficio Tirocini	RifiutaRichiestaConvenzionamento()

Lista di controllo degli accessi per il sottosistema ProgettiFormativiController

Attore	Operazione
Utente	VisualizzaProgettiFormativiAttivi()
Delegato Aziendale	InserimentoProgettoFormativo()
Delegato Aziendale	ModificaProgettoFormativo()
Delegato Aziendale	ArchiviaProgettoFormativo()
Delegato Aziendale	RiattivaProgettoFormativo()

Lista di controllo degli accessi per il sottosistema DomandeTirocinioController

Attore	Operazione
Studente	InviaDomandaDiTirocinio()
Studente	VisualizzaStatoDomandeInviato()
Studente	VisualizzaTirociniInCorsoPersonalizzati()



Delegato Aziendale	VisualizzaDomandeTirociniInAttesa()
Delegato Aziendale	VisualizzaDomandeTirociniInAttesa()
Delegato Aziendale	AccettaDomandaTirocinio()
Delegato Aziendale	RifiutaDomandaTirocinio()
Delegato Aziendale	VisualizzaStatoDomandeAccettate()
Delegato Aziendale	VisualizzaTirociniInCorsoAzienda()
Responsabile Ufficio Tirocini	VisualizzaDomandeTirocinioAccettate()
Responsabile Ufficio Tirocini	ApprovaDomandaTirocinio()
Responsabile Ufficio Tirocini	RespingiDomandaTirocinio()
Responsabile Ufficio Tirocini	VisualizzaTirociniiInCorsoDipartimento()

### 3.6 Controllo del flusso globale del sistema

Il sistema IFY fornisce una funzionalità che richiede una continua interazione da parte dell'utente, per questo motivo abbiamo adottato un controllo del flusso globale del sistema di tipo event-driven, ovvero guidato dagli eventi.

### 3.7 Condizioni limite

#### *Start-Up*

Per il primo start-up del sistema “IFY Internship For You” è indispensabile l'avvio di un web server che doti il servizio di una base di dati MySQL per l'amministrazione dei dati persistenti.

Successivamente, attraverso l'interfaccia di pagina iniziale, sarà possibile accedere al sito tramite adeguate credenziali (e-mail e password). Una volta effettuata l'autenticazione, “IFY Internship For You” presenterà all'utente la propria home page, differente in base al tipo di utente, dalla quale sarà possibile servirsi di tutte le funzionalità di cui la piattaforma dispone.



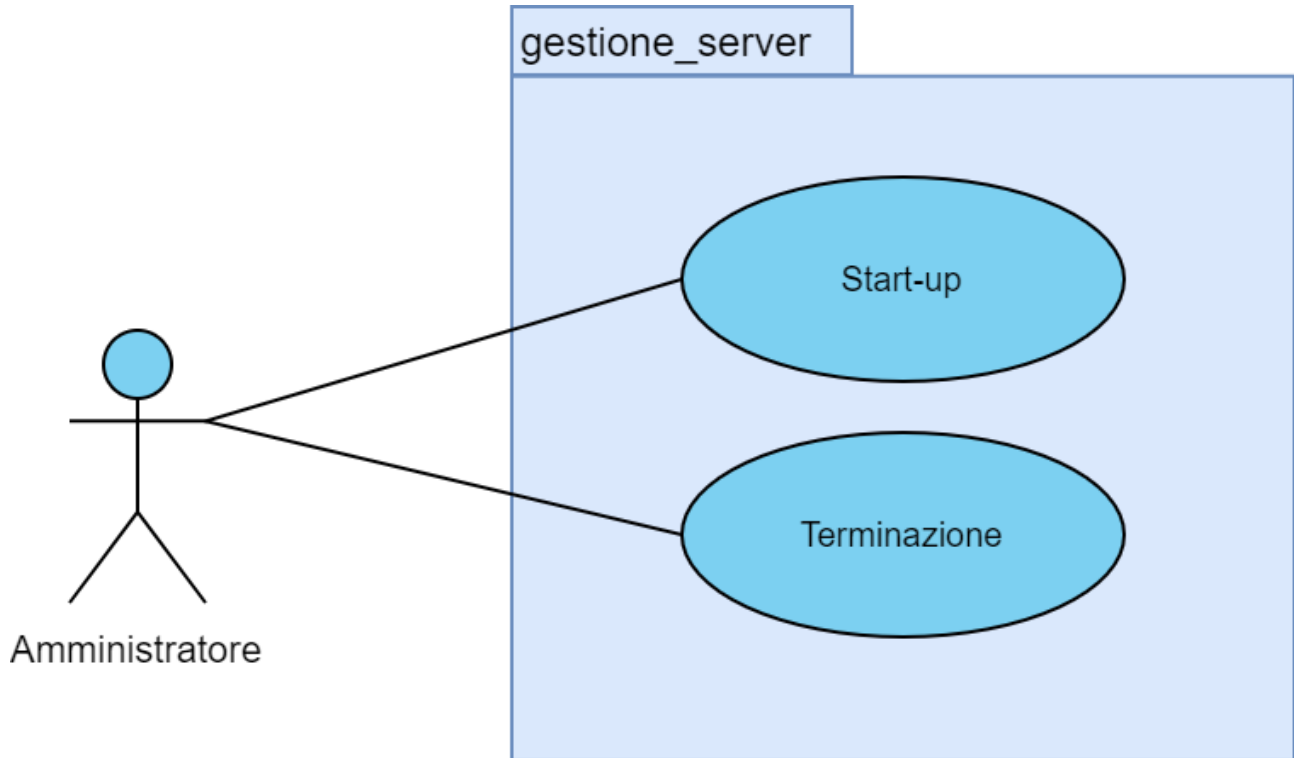
Identificativo UC_SU		Avvio del server	Data	28/11/2019
			Vers.	0.00.001
			Autore	RC
Descrizione		Lo UC fornisce la funzionalità di avvio del server da parte dell'amministratore.		
Attore Principale		Amministratore È interessato ad avviare il server.		
Attori secondari		NA		
Entry Condition		L'amministratore ha accesso al server.		
Exit condition On success		Il sistema è avviato correttamente.		
Exit condition On failure		Il sistema non è avviato.		
Rilevanza/User Priority		Alta		
Frequenza stimata		6 usi/anno		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Amminis tratore:	Stabilisce la connessione con il database.		
2	Amminis tratore:	Avvia uno script di configurazione del database.		
3	Sistema:	Inizializza il database.		
4	Amminis tratore:	Esegue il deploy dell'archivio "ify.war" del sistema sul server.		
5	Sistema:	Se in precedenza il server è stato spento normalmente, legge la lista di aziende, studenti e impiegati. Se in precedenza il server si è bloccato, notifica l'amministratore.		
Scenario/Flusso di eventi Alternativo: lo script inserito è incorretto				
1	Sistema:	Visualizza un messaggio di errore.		



### ***Terminazione***

Alla chiusura dell'applicazione, si ha la terminazione del sistema attraverso un logout. Per consentire la giusta terminazione del server, la gestione del sistema dovrà effettuare la procedura di terminazione, dopo la quale nessun client potrà collegarsi al sistema.

Identificativo UC_SD		Terminazione del server	Data	28/11/2019
			Vers.	0.00.001
			Autore	RC
Descrizione		Lo UC fornisce la funzionalità di terminazione del server da parte dell'amministratore.		
Attore Principale		<b>Amministratore</b> È interessato a terminare il server.		
Attori secondari		NA		
Entry Condition		Il server è avviato e l'amministratore ha effettuato l'accesso.		
Exit condition On success		Il sistema è terminato correttamente.		
Exit condition On failure		Il sistema non termina.		
Rilevanza/User Priority		Alta		
Frequenza stimata		6 usi/anno		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Amminis tratore:	Avvia uno script per terminare la connessione al database.		
2	Sistema:	Esegue la terminazione del server.		
Scenario/Flusso di eventi Alternativo: lo script inserito è incorretto				
1	Sistema:	Visualizza un messaggio di errore.		



### ***Fallimento***

- Nel caso di difetti dovuti al sovraccarico della base di dati con conseguente fallimento della stessa è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione della base di dati;
- Nel caso in cui avvenga un'interruzione imprevista dell'alimentazione non sono previste metodologie che ripristinino lo stato del Sistema prima della disattivazione non voluta;
- Un altro caso di fallimento potrebbe provenire dal software stesso che produce una chiusura inattesa causata da errori effettuati durante la fase di implementazione. Non essendo previste politiche correttive, l'unica operazione ammessa in questa specifica situazione è la chiusura del sistema e il suo successivo riavvio;
- Un altro caso di insuccesso potrebbe essere dovuto ad un difetto critico nell'hardware, contro il quale non è presente alcuna contromisura.

## 4. Servizi dei sottosistemi

