



TirocinioSmart

SDD System Design Document

TirocinioSmart

Data	Versione	Cambiamenti	Autori
26/11/2017	1.0	Versione iniziale	[tutti]
01/12/2017	1.0.1	Ristrutturazione diagramma dei servizi	AC
10/12/2017	1.1	Aggiornamento divisione in sottosistemi	AC
12/12/2017	1.2	Ristrutturazione divisione in sottosistemi	AC
14/01/2017	1.2.1	Aggiornamento servizi dei sottosistemi	AC
27/12/2017	1.3	Aggiornamento schema E-R	AC
03/02/2018	1.4	Revisione generale con aggiornamento dei servizi dei sottosistemi e della matrice d'accesso	AC

Sommario

1. Introduzione	4
1.1 Obiettivi del sistema.....	4
1.2 Design Goals & Trade-offs	4
Tempo di rilascio vs Funzionalità	5
Prestazioni vs Costi.....	5
Prestazioni vs Affidabilità	5
1.3 Definizioni, acronimi e abbreviazioni	5
1.4 Riferimenti	5
1.5 Panoramica	6
2. Architettura di Sistemi simili	6
3. Architettura del Sistema proposto	6
3.1 Panoramica	6
3.2 Decomposizione in sottosistemi.....	6
3.2.3 Diagramma di deployment.....	7
3.3 Mapping hardware/software	8
3.4 Gestione dati persistenti	8
3.5 Controllo degli accessi e sicurezza.....	9
3.6 Controllo flusso globale del sistema.....	9
3.7 Condizione limite.....	10
3.7.1 Start-up	10
3.7.2 Terminazione.....	11
3.7.3 Fallimento	12
4. Servizi dei Sottosistemi.....	12
Glossario	15

1. Introduzione

1.1 Obiettivi del sistema

Un'esperienza di tirocinio è buona se permette al tirocinante di maturare un bagaglio di esperienze competenze realmente utili al suo ingresso nel mercato del lavoro. Tutto questo è possibile anche attraverso il supporto di un sistema che agevoli quelle che sono le sue richieste e renda migliore tale esperienza. Nasce così TirocinioSmart!

Il sistema proposto parte dall'idea di poter fornire una piattaforma che sia in grado di assolvere a quelle che sono le esigenze di entità coinvolte nell'ambito della gestione dei tirocini. Si vuole quindi realizzare una piattaforma web che consentirà lo svolgimento delle attività da parte degli utenti tramite un semplice web browser, indipendentemente dal dispositivo utilizzato.

TirocinioSmart avrà la necessità di gestire dati persistenti: la scelta, per motivi di efficienza, ricade quindi su un database relazionale che permetterà di poter tener traccia delle informazioni fondamentali necessarie al sistema. Da tale database attingerà un'applicazione web deputata alla gestione delle interazioni con l'utente ed alla manipolazione dei suddetti dati.

La piattaforma garantirà il controllo degli accessi tramite la possibilità di autenticarsi in seguito inserimento di una username e di una password. La creazione di account per gli studenti è filtrata manualmente dall'ufficio tirocini a causa dell'impossibilità di comunicare direttamente con il sistema esse3 utilizzato dall'ateneo.

1.2 Design Goals & Trade-offs

Illustriamo nella seguente tabella gli obbiettivi di design per il sistema e le relative priorità (a numeri più bassi corrispondono priorità più elevate). Per ogni obbiettivo riportiamo anche l'origine, facendo riferimento, in particolare, all'identificativo del requisito non funzionale ad esso associato.

Priorità	ID	Ddescrizione	Categoria	Origine
1	DG_1	Leggibilità: Il codice prodotto dev'essere semplice da comprendere. Ogni metodo e campo non banale dev'essere documentato opportunamente al fine di aumentarne la comprensione	Manutenzione	RNF-S1
2	DG_2	Robustezza: Vogliamo proporre un sistema che abbia la capacità di sopravvivere ad input non validi immessi dall'utente. Pertanto, il sistema deve garantire il filtraggio dei dati inconsistenti o errati inseriti dall'utente, invitandolo a reinserirli.	Dependability	RNF-A3
2	DG_3	Affidabilità: Il sistema dev'essere in grado di riconoscere situazioni anomale e prevenire modifiche ai dati persistenti al fine di garantirne la consistenza	Dependability	RNF-A2
3	DG_4	Sicurezza: Siccome TirocinioSmart prevede l'immissione da parte degli utenti di dati sensibili, si rende necessario fornire uno strumento di autenticazione sicuro, composto dalla richiesta di username e password prima di ogni accesso ad informazioni riservate così da proteggere i dati da accessi non autorizzati. Le suddette password saranno crittografate	Dependability	RNF-A1

2	DG_5	Costi di sviluppo: Lo sviluppo del prodotto richiederà costi ridotti sia in termini di risorse umane (per cui è fissato un tetto di 75 ore-lavoro), sia in termini economici (per cui si punta a ricorrere a soluzioni off-the-shelf open source)	Costo	Top management
3	DG_6	Usabilità: Il sistema deve essere facile da apprendere ed intuitivo da utilizzare senza necessariamente consultare la documentazione. I contenuti dovranno essere fruibili attraverso dispositivi sia desktop che mobile ed accessibili attraverso un numero ridotto di interazioni	End User	RNF-U1 RNF-U2 RNF-U3
3	DG_7	Tempi di risposta: Il sistema deve elaborare le richieste e produrre output in meno di 2 secondi (al netto di ritardi dovuti alla trasmissione su rete)	Performance	RNF-P1
3	DG_8	Throughput: Il sistema deve permettere l'interazione contemporanea di almeno 100 utenti diversi	Performance	RNF-P2
1	DG_9	Modificabilità: Le funzionalità del sistema devono essere facilmente modificabili	Manutenibilità	RNF-S2
1	DG_10	Estensibilità: Il sistema deve agevolare l'introduzione di nuove funzionalità	Manutenibilità	RNF-S3

Riportiamo ora quelli che sono i compromessi considerati e la posizione del team di in relazione ad ognuno di essi.

Tempo di rilascio vs Funzionalità

Sebbene i tempi siano piuttosto proibitivi, preferiamo consegnare con leggero ritardo un prodotto che faccia ciò che promette piuttosto che un prodotto che non possa essere utilizzato a causa della mancanza di funzionalità.

Prestazioni vs Costi

Considerato il budget ridotto a disposizione, si preferisce rientrare nei costi dedicando un numero ridotto di ore-lavoro alla massimizzazione delle prestazioni.

Prestazioni vs Affidabilità

I dati gestiti dal sistema sono piuttosto sensibili, pertanto preferiamo garantire un maggior controllo di input e consistenza a scapito dei tempi di risposta.

1.3 Definizioni, acronimi e abbreviazioni

- Webapp: abbreviazione per “applicazione web”

1.4 Riferimenti

- Requisiti funzionali: Sezione 3.2 del RAD
- Requisiti non funzionali: Sezione 3.3 del RAD
- [JobSoul](#)
- [UniMoRe](#)
- [E-R Formazione Lavoro](#)
- [Apache Tomcat](#)
- [MySQL](#)
- [Layer versus Tier](#)

1.5 Panoramica

Nel documento verranno affrontati l'analisi delle architetture di sistemi simili, la decomposizione in sottosistemi del sistema proposto con la definizione della strategia di deploy e le condizioni limite. Verranno quindi definiti i servizi esposti da ciascun sottosistema.

2. Architettura di Sistemi simili

Nel nostro caso non è presente un'architettura software già esistente, pertanto abbiamo analizzato le architetture di sistemi simili.

Le piattaforme prese in considerazione sono JobSoul, la piattaforma tirocini di UniMoRe e E-R FormazioneLavoro, che offrono la possibilità di poter scegliere tra una lista di offerte di lavoro e di tirocini in tutta Italia. Tutte e 3 sono accessibili tramite sito web raggiungibile con qualsiasi web browser e nessuna delle 3 dispone di una companion app per dispositivi mobili o fissi.

Ciò che si è evinto dall'analisi è che alla base di tutte le piattaforme è presente un database per la memorizzazione dei dati persistenti interrogato da applicazioni web progettate con diverse tecnologie (rispettivamente PHP, ASP e JSP operanti su server Apache). Deduciamo che ognuna delle suddette piattaforme sia quindi sviluppata secondo un'architettura a 3 strati, ma non siamo in grado di stabilire se siano semplicemente 3 livelli oppure 3 differenti tier.

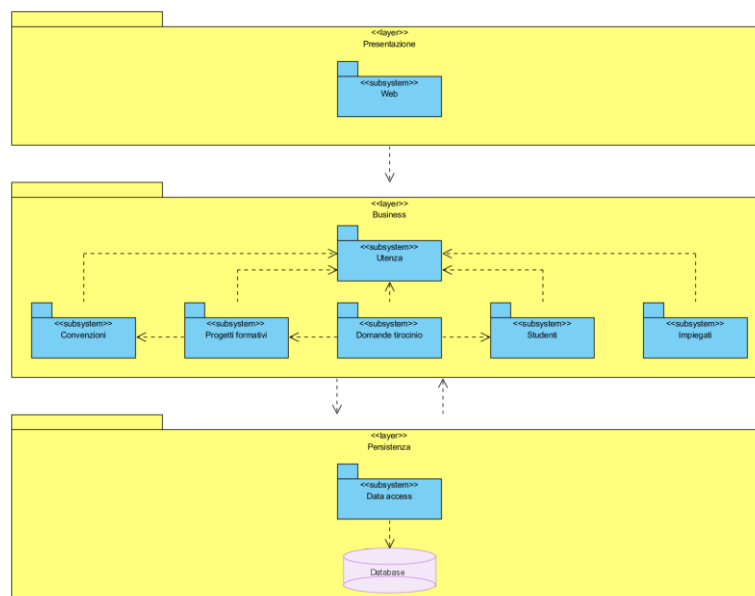
Per quanto riguarda il controllo degli accessi, quello che si evince è che tutti e 3 i sistemi presentano un'autenticazione con username e password in modo da rendere disponibili le funzionalità che offrono solamente ad utenti autorizzati, mentre non possiamo conoscere i criteri di sicurezza utilizzati per la gestione dei dati sensibili (cifratura delle credenziali o simili).

3. Architettura del Sistema proposto

3.1 Panoramica

TirocinioSmart è un'applicazione distribuita che fornisce un'interfaccia web. Essa, inoltre, ricorre all'utilizzo di un database relazionale per il salvataggio dei dati persistenti.

3.2 Decomposizione in sottosistemi



Il sistema è suddiviso in 3 livelli logici: presentazione, business e persistenza che si occupano rispettivamente di presentazione delle informazioni all'utente, definizione della logica applicativa e gestione dei dati persistenti.

Il livello di presentazione è composto da un solo sottosistema:

- **Web:** definisce l'interfaccia utente e si occupa del controllo del flusso di esecuzione

Il livello di business è composto da sei sottosistemi:

- **Utenza:** definisce l'utente generico del sistema ed offre tutti i servizi relativi all'autenticazione
- **Convenzioni:** modella tutto ciò che riguarda il processo di convenzionamento e le aziende convenzionate in termini di entità e operazioni
- **Studenti:** modella gli studenti ed il processo d'iscrizione alla piattaforma, definendo tutte le operazioni ad essi relative
- **Impiegati:** modella gli impiegati dell'ufficio tirocini
- **Progetti formativi:** modella i progetti formativi offerti dalle aziende e le relative operazioni
- **Domande tirocinio:** modella le domande di tirocinio e le operazioni ad esse associate

Il livello di persistenza è invece composto da un solo sottosistema:

- **Data access:** si occupa del reperimento e del salvataggio delle informazioni manipolate dalla webapp dal/sul database sottostante

Si noti come la divisione in sottosistemi sia stata realizzata tramite una strutturazione 3-layer al fine di disaccoppiare l'interfaccia dalla logica di business dell'applicazione: il livello di business non è infatti a conoscenza di come l'informazione sarà presentata all'utente e ciò permette, in futuro, di poter realizzare un client mobile oppure un'interfaccia desktop piuttosto che web.

Nella divisione in sottosistemi sono stati utilizzati concetti fondamentali del pattern MVC, con l'allocazione di View e Controller al livello di presentazione e la realizzazione del Model tramite i livelli di Business e Data Access.

È inoltre rilevante osservare come la gestione dei tirocini sia stata inglobata in quella delle domande di tirocinio, mentre la gestione dell'utenza è stata scorporata in gestione degli studenti, gestione degli impiegati ed in una gestione dell'utenza che offre esclusivamente servizi generici a tutti gli utenti

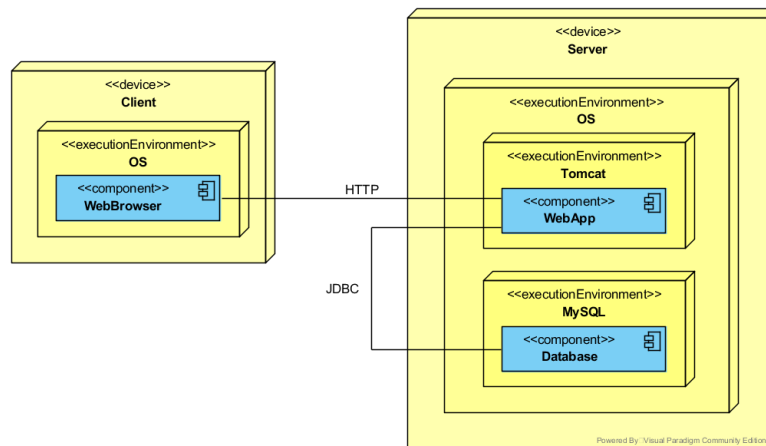
Una nota va dedicata al sottosistema che modella gli impiegati dell'ufficio tirocini: sebbene possa sembrare inutile, data la mancanza di servizi offerti ad altri sottosistemi, abbiamo ritenuto utile realizzarlo in prospettiva futura. Nella prima release, infatti, TirocinioSmart non permetterà la gestione di impiegati multipli dell'ufficio tirocini, ma tale miglioramento è previsto per le prossime versioni. Offrire quindi una strutturazione simile già dalla prima versione del sistema apre ad una maggiore manutenibilità.

3.2.3 Diagramma di deployment

TirocinioSmart consiste di un'applicazione distribuita installabile su un qualsiasi server in grado di eseguire Java e MySQL.

L'architettura scelta prevede l'interazione della webapp con un database: data la ridotta quantità di dati gestiti ed il numero piuttosto basso di utenti giornalieri, le suddette componenti sono installate sulla stessa macchina secondo un'architettura client-server.

Il sistema sarà accessibile tramite comuni browser web installati sui dispositivi a disposizione degli attori.



3.3 Mapping hardware/software

TirocinioSmart si compone di due componenti principali:

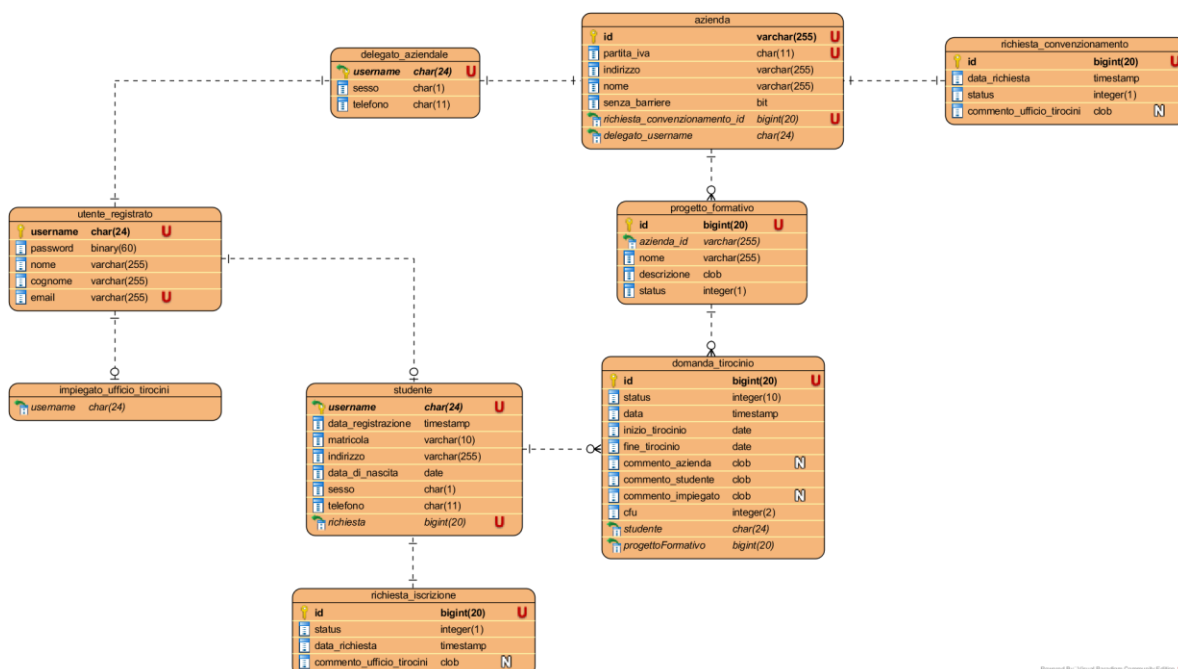
- Webapp, cui saranno allocati i layer di presentazione e business oltre al sottosistema di data access
- Database, realizzante il layer di persistenza

Il sistema necessita di una macchina in grado di supportare Apache Tomcat, al fine di garantire l'operabilità della WebApp, e MySQL, per garantire invece l'operabilità del database con cui la WebApp si interfaccia.

WebApp e DBMS saranno quindi installati sullo stesso nodo in modo da ridurre i possibili fallimenti o ritardi di propagazione delle informazioni dovuti a problemi di connettività.

3.4 Gestione dati persistenti

Per la gestione dei dati persistenti, TirocinioSmart si affida ad un database relazionale gestito tramite MySQL. La struttura dei dati memorizzati segue il seguente schema:



Le associazioni uno ad uno tra l'utente registrato e le sue specializzazioni (delegato_aziendale, studente e impiegato_ufficio_tirocini) hanno come regola di business che per ognuna delle entry di utente_registrato, esattamente una delle tre associazioni debba essere soddisfatta.

3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di username e password per gli utenti del sistema che hanno possibilità di creare o modificare gli oggetti che modellano entità di dominio, così da prevenire accessi non autorizzati ad informazioni sensibili. Sottolineiamo che il sistema non fornirà un metodo di recupero o modifica delle password, almeno nella sua prima versione.

Si ricorrerà all'utilizzo della sessione del server per tenere traccia dell'utente loggato. Per questioni di efficienza, la sessione sarà attiva per soli 30 minuti dopo l'ultima interazione dell'utente col sistema.

Per questioni legate al budget a disposizione del team, il salvataggio delle password sarà in chiaro su database: non ci sarà alcun tipo di cifratura, almeno nella prima versione del sistema.

Nelle prime versioni non sarà inoltre utilizzato SSL su connessione HTTP tra client e server ma non ne è escluso l'utilizzo in futuro.

Le operazioni che gli utenti dell'applicazione web possono effettuare sugli oggetti sono riportate nella tabella che segue:

Oggetto Attore	Utenza	Convenzioni	Progetti formativi	Domande tirocinio	Studenti	Impiegati
Utente		Consultazione aziende convenzionate	Consultazione progetti formativi			
Ospite		Richiesta convenzionamento			Richiesta iscrizione	
UtenteRegistrato	Autenticazione Logout					
Delegato Aziendale			Aggiunta e archiviazione progetto formativo			
Studente				Invio nuove domande e consultazione domande inviate		
Impiegato UfficioTirocini		Consultazione, approvazione e rifiuto richieste convenzionamento		Consultazione, approvazione e respinta domande tirocinio	Consultazione, approvazione e rifiuto richieste iscrizione	

Dalla tabella si evince come l'utente non abbia interazioni dirette con il sottosistema di accesso al database, cui invece accedono i singoli sottosistemi di business: per questo motivo si è deciso di non riportarlo nella matrice d'accesso.

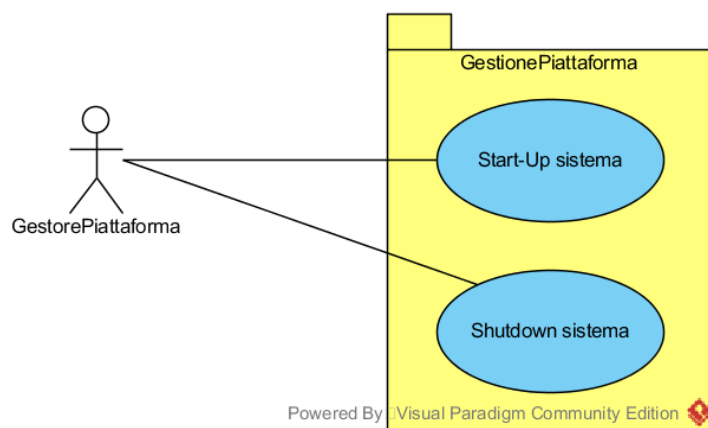
3.6 Controllo flusso globale del sistema

Il sistema adotta un controllo del flusso globale di tipo thread-driven, questo perché il web container (Tomcat) web permette l'interazione concorrente tra la WebApp e più client tramite l'intercettazione di eventi generati proprio da questi ultimi.

Questa soluzione permette al sistema di poter rispondere a più utenti contemporaneamente ma richiede che gli accessi in scrittura ai dati persistenti avvengano sequenzialmente, gestendo opportunamente le sezioni critiche.

In generale, ogni richiesta da parte di un utente verrà eseguita in un thread dedicato.

3.7 Condizione limite



3.7.1 Start-up

Identificativo UC_GPI	Start-up sistema	Data	24/11/2017
		Versione	1.1
		Autore	AC
Descrizione	Lo use case definisce la funzionalità di avvio del sistema per il gestore della piattaforma		
Attore Principale	Gestore piattaforma È interessato ad avviare il sistema		
Attori secondari	N/A		
Entry Condition	Il gestore ha accesso alla macchina su cui è installato il sistema		
Exit condition on success	Il sistema è avviato correttamente		
Exit condition on failure	Il sistema non è avviato		
Rilevanza/User Priority	Elevata		
Frequenza stimata	1 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Gestore sistema	Accende il server	
2	Gestore sistema	Lancia il servizio del DBMS tramite l'apposito comando: <ul style="list-style-type: none">Lanciando il servizio MySQL57 su server WindowsIl comando <code>sudo /etc/init.d/mysql start</code> su server Unix	
3	Gestore sistema	Lancia il web container tramite: <ul style="list-style-type: none">Il comando <code>java -jar TirocinioSmart.war</code>	
4	Sistema	Comunica al gestore che lo startup si è concluso con successo	

Scenario/Flusso di eventi di ERRORE: non è possibile avviare il sistema		
4.e1	Sistema	Mostra all'impiegato dell'ufficio tirocini un messaggio che ne specifica il motivo

3.7.2 Terminazione

Identificativo UC_GP2	Shutdown sistema	Data	24/11/2017
		Versione	1.0
		Autore	AC
Descrizione	Lo UC fornisce al gestore della piattaforma la possibilità di terminare il sistema		
Attore Principale	Gestore piattaforma È interessato a terminare il sistema		
Attori secondari	N/A		
Entry Condition	Il gestore ha accesso alla macchina su cui il sistema è in esecuzione		
Exit condition on success	Il sistema è terminato correttamente		
Exit condition on failure	Il sistema resta in esecuzione		
Rilevanza/User Priority	Elevata		
Frequenza stimata	1 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Gestore sistema	Termina il servizio del web container chiudendo la shell in cui questo è stato avviato	
2	Sistema	Comunica al gestore che il servizio è stato terminato correttamente	
3	Gestore sistema	Termina il servizio del DBMS tramite l'apposito comando: <ul style="list-style-type: none">• Terminando il servizio MySQL57 su server Windows• Il comando <code>sudo /etc/init.d/mysql stop</code> su server Unix	
4	Sistema	Comunica al gestore che il servizio è stato terminato correttamente	
Scenario/Flusso di eventi di ERRORE: non è possibile terminare il servizio			
2.e1	Sistema	Mostra al gestore del sistema un errore che riporta l'errore	

3.7.3 Fallimento

TirocinioSmart può incorrere in diversi casi di fallimento, riguardanti sia l'hardware che il software:

- Fallimenti Hardware
 - *Crash del disco su cui i dati persistenti sono salvati*: il sistema non prevede alcuna strategia di backup e ripristino dei dati
- Fallimenti nell'ambiente di esecuzione
 - *Interruzione della fornitura elettrica al server*: il sistema non prevede alcuna strategia che ne garantisca l'operabilità in questo tipo di condizione
- Fallimenti Software
 - *Impossibilità di stabilire una connessione col database*: il sistema mostra all'utente una schermata che riporta il rilevamento di un errore interno

4. Servizi dei Sottosistemi

Data Access	
Servizio	Descrizione
Caricamento richieste di convenzionamento	Il sottosistema permette di ottenere dal database tutte le richieste di convenzionamento
Caricamento richiesta di convenzionamento	Il sottosistema permette di ottenere dal database una specifica richiesta di convenzionamento
Caricamento richieste di convenzionamento in attesa	Il sottosistema permette di ottenere dal database tutte le richieste di convenzionamento che non sono ancora state gestite
Salvataggio richiesta di convenzionamento	Il sottosistema permette di scaricare su database le informazioni relative ad una richiesta di convenzionamento
Aggiornamento richiesta di convenzionamento	Il sottosistema permette di aggiornare su database una richiesta di convenzionamento
Caricamento aziende	Il sottosistema permette di ottenere dal database tutte le aziende presenti nel sistema
Caricamento azienda	Il sottosistema permette di ottenere dal database una specifica azienda presente nel sistema
Caricamento aziende senza barriere architettoniche	Il sottosistema permette di ottenere dal database tutte le aziende senza barriere architettoniche presenti nel sistema
Caricamento aziende convenzionate	Il sottosistema permette di ottenere dal database tutte le aziende convenzionate
Esistenza azienda	Il sottosistema permette di determinare se un'azienda è già presente nel database a partire dalla sua partita IVA
Salvataggio azienda	Il sottosistema permette di scaricare su database le informazioni relative ad un'azienda
Caricamento richieste d'iscrizione in attesa	Il sottosistema permette di ottenere dal database tutte le richieste d'iscrizione che non sono ancora state gestite
Caricamento richiesta d'iscrizione	Il sottosistema permette di ottenere dal database una specifica richiesta d'iscrizione
Salvataggio richiesta d'iscrizione	Il sottosistema permette di scaricare su database una richiesta d'iscrizione
Aggiornamento richiesta d'iscrizione	Il sottosistema permette di aggiornare su database una richiesta d'iscrizione
Esistenza utente	Il sottosistema permette di determinare se un utente è già presente nel database a partire dal suo username o email
Caricamento delegato	Il sottosistema permette di ottenere dal database un delegato aziendale a partire dai suoi username e password (o solo username)

Salvataggio delegato	Il sottosistema permette di scaricare su database le informazioni relative ad un delegato aziendale
Caricamento studente	Il sottosistema permette di ottenere dal database uno studente a partire dai suoi username e password (o solo username)
Esistenza studente	Il sottosistema permette di stabilire la presenza di uno studente su database o meno a partire dalla sua matricola
Salvataggio studente	Il sottosistema permette di scaricare su database le informazioni relative ad uno studente
Caricamento impiegato	Il sottosistema permette di ottenere dal database un impiegato dell'ufficio tirocini a partire dai suoi username e password (o solo username)
Caricamento progetti formativi azienda	Il sottosistema permette di ottenere dal database tutti i progetti formativi attivi di un'azienda
Caricamento progetto formativo	Il sottosistema permette di ottenere dal database un progetto formativo
Aggiornamento progetto formativo	Il sistema permette di aggiornare su database uno specifico progetto formativo
Caricamento domande tirocinio	Il sottosistema permette di ottenere dal database tutte le domande di tirocinio che si trovano in uno stato specificato
Caricamento domanda tirocinio	Il sottosistema permette di ottenere dal database una specifica domanda di tirocinio
Caricamento domande tirocinio studente	Il sottosistema permette di ottenere dal database tutte le domande di tirocinio associate ad uno studente
Caricamento domande tirocinio progetto formativo	Il sottosistema permette di ottenere dal database tutte le domande di tirocinio associate ad un progetto formativo
Salvataggio domanda tirocinio	Il sottosistema permette di scaricare su database le informazioni relative ad una domanda di tirocinio
Aggiornamento domanda tirocinio	Il sottosistema permette di aggiornare su database una specifica domanda di tirocinio

Utenza	
Servizio	Descrizione
Login	Il sottosistema permette di autenticare un utente nel sistema tramite una coppia di username e password
Logout	Il sistema permette di rimuovere l'utente che si era precedentemente autenticato nel sistema
Ottenimento utente autenticato	Il sottosistema permette di ottenere l'utente autenticato nel sistema

Convenzioni	
Servizio	Descrizione
Registrazione richiesta convenzionamento	Il sottosistema permette di registrare una richiesta di convenzionamento con tutte le informazioni di azienda e delegato associate
Approvazione richiesta convenzionamento	Il sottosistema permette di approvare una richiesta di convenzionamento esistente
Rifiuto richiesta convenzionamento	Il sottosistema permette di rifiutare una richiesta di convenzionamento esistente specificandone la motivazione
Elencazione richieste convenzionamento in attesa	Il sottosistema permette di elencare le richieste di convenzionamento non ancora gestite
Elencazione aziende convenzionate	Il sottosistema permette di elencare le aziende convenzionate
Ottenimento azienda	Il sottosistema permette di ottenere tutte le informazioni relative ad una specifica

	azienda
--	---------

Progetti formativi	
Servizio	Descrizione
Aggiunta progetto formativo	Il sottosistema permette di aggiungere un progetto formativo alla lista di quelli offerti da un'azienda
Archiviazione progetto formativo	Il sottosistema permette di archiviare un progetto formativo esistente
Elencazione progetti formativi	Il sottosistema permette di elencare i progetti formativi offerti da un'azienda
Ottenimento progetto formativo	Il sottosistema permette di ottenere tutte le informazioni relative ad un progetto formativo

Domande tirocinio	
Servizio	Descrizione
Registrazione domanda di tirocinio	Il sottosistema permette di registrare una nuova domanda di tirocinio
Accettazione domanda di tirocinio	Il sottosistema permette di accettare una domanda di tirocinio esistente
Rifiuto domanda di tirocinio	Il sottosistema permette di rifiutare una domanda di tirocinio esistente
Approvazione domanda di tirocinio	Il sottosistema permette di approvare una domanda di tirocinio esistente
Respinta domanda di tirocinio	Il sottosistema permette di respingere una domanda di tirocinio
Elencazione domande di tirocinio ricevute	Il sottosistema permette di elencare le domande di tirocinio ricevute da un'azienda o dall'ufficio tirocini
Elencazione domande di tirocinio inviate	Il sottosistema permette di elencare le domande di tirocinio inviate da uno studente o da un'azienda
Elencazione tirocini in corso	Il sottosistema permette di elencare i tirocini in corso di uno studente, presso un'azienda o in generale

Studenti	
Servizio	Descrizione
Registrazione richiesta d'iscrizione	Il sottosistema permette di registrare una nuova richiesta d'iscrizione con tutte le informazioni relative allo studente associato
Approvazione richiesta d'iscrizione	Il sottosistema permette di approvare una richiesta d'iscrizione esistente
Rifiuto richiesta d'iscrizione	Il sottosistema permette di rifiutare una richiesta d'iscrizione esistente
Elencazione richieste d'iscrizione in attesa	Il sottosistema permette di elencare tutte le richieste d'iscrizione non ancora gestite

Impiegati	
Servizio	Descrizione
Ottenimento impiegato	Il sottosistema permette di ottenere tutte le informazioni relative ad un impiegato

Glossario

Java: linguaggio di programmazione orientato agli oggetti

Applicazione web: programma accessibile tramite browser web ed in grado di elaborare richieste e risposte HTTP

esse3: piattaforma web utilizzata dall'Università degli Studi di Salerno per la gestione di insegnamenti e carriere degli studenti

Storage: memoria persistente

Throughput: misura della capacità del sistema di condurre task contemporanei

Web Server: macchina connessa alla rete dotata di un ambiente di esecuzione

Application Server: sistema software per la gestione delle richieste/risposte provenienti dai client

Tomcat: specifico application server

DBMS: sistema software per la gestione dei dati persistenti su database

MySQL: specifico DBMS