



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C. Gravino

System Design Document Turing Careers

| | |
|---------------|---|
| Riferimento | |
| Versione | 1.3 |
| Data | 15/02/24 |
| Destinatario | Prof. C. Gravino |
| Presentato da | Antonio Pagnotta, Antonino Lorenzo, Claudio Gaudino, Jacopo Passariello, |
| Approvato da | |



Laurea Triennale in informatica-Università di Salerno
Corso di Ingegneria del Software- Prof. C. Gravino

Pagina intenzionalmente lasciata in bianco.



Revision History

| Data | Versione | Descrizione | Autori |
|------------|----------|--|--|
| 23/11/23 | 0.1 | Prima stesura | Antonio Pagnotta |
| 25/11/23 | 0.1.1 | Aggiunta Design Goals | Antonio Pagnotta |
| 26/11/23 | 0.1.2 | Revisione Design Goals e Aggiunta Trade-offs | Antonio Pagnotta, Jacopo Passariello, Antonino Lorenzo |
| 10/12/23 | 0.1.3 | Scomposizione in Sottosistemi Map Hardware Software | Antonio Pagnotta, Jacopo Passariello, Antonino Lorenzo |
| 12/12/23 | 0.1.4 | Prima Revisione | Team |
| 13/12/23 | 0.1.5 | Controllo degli Accessi | Team |
| 13/12/23 | 0.1.6 | Gestione dei Dati Persistenti | Antonio Pagnotta, Antonino Lorenzo |
| 17/12/2023 | 0.1.7 | Boundary Conditions | Team |
| 18/12/2023 | 1.0 | Seconda Revisione | Team |
| 01/02/2024 | 1.1 | Revisione Architettura | Antonio Pagnotta, Antonino Lorenzo |
| 13/02/2024 | 1.2 | Modifica Architettura in accordo ai suggerimenti del tutor | Antonio Pagnotta, Jacopo Passariello |
| 15/02/2024 | 1.3 | Revisione Finale del Documento | Antonio Pagnotta, Jacopo Passariello |



Sommario

1. Introduzione
 - 1.1. Scopo del Sistema
 - 1.2. Design Goals & trade-offs
 - 1.3. Definizioni, acronimi e abbreviazione
 - 1.4. Riferimenti
 - 1.5. Panoramica
2. Architettura dei Sistemi Simili
3. Architettura del Sistema Proposto
 - 3.1. Panoramica
 - 3.2. Decomposizione in sottosistemi
 - 3.2.1. Diagramma di Mapping Architetturale
 - 3.3. Mapping hardware/software
 - 3.4. Class Diagram Ristrutturato
 - 3.5. Gestione dati persistenti
 - 3.6. Controllo degli accessi e sicurezza
 - 3.7. Condizione limite
 - 3.7.1. Start-up
 - 3.7.2. Terminazione
 - 3.7.3. Fallimento
4. Servizi dei sottosistemi
5. Glossario



1. Introduzione

1.1 Scopo del sistema

Attualmente il mercato dello sviluppo software è tra quelli più in crescita e di conseguenza lo è anche la richiesta di professionisti. Nonostante ciò, il processo di assunzione sia per le aziende che per gli sviluppatori presenta molteplici sfide. In particolare, per gli sviluppatori, sia freelance che non, trovare un lavoro adatto alle proprie necessità e competenze è sempre più difficile in quanto richiede una lunga ricerca che può protrarsi per mesi, richiedendo il destreggiarsi tra piattaforme aziendali eterogenee. D'altro canto, anche per le aziende il processo di recruitment può essere dispendioso, sia in termini di tempo che di manodopera: richiede infatti molte interviste a molti individui. A tal proposito il sistema software proposto, Turing Careers, ha lo scopo di fornire una piattaforma digitale che provveda a semplificare il processo di ricerca di lavoro per gli sviluppatori, assieme al processo di ricerca di nuovi candidati per i datori di lavoro.

1.2 Design Goals & Trade-offs

| Rank | ID | Descrizione | Categoria | RNF Origine |
|------|--------|--|-----------------|----------------|
| 1 | DG_U_1 | Il sistema dovrebbe essere facilmente accessibile a tutti gli utenti. | Usabilità | RNF_U_1 |
| 1 | DG_U_2 | L'interfaccia deve presentarsi con chiarezza, offrendo diverse scorciatoie per raggiungere ogni sezione del sistema in modo inequivocabile e semplice. | Usabilità | RNF_U_2 |
| 1 | DG_U_3 | Ogni utente deve poter comprendere ogni funzionalità, in modo da poter usare ognuna di essa, in non più di 5 minuti. | Usabilità | RNF_U_3 |
| 3 | DG_U_4 | Il sito deve garantire una corretta visualizzazione su schermi di formati diversi. | Usabilità | RNF_U_4 |
| 3 | DG_P_1 | Quando viene effettuata la ricerca il sistema deve essere capace di produrre un risultato entro 2 secondi. | Prestazioni | RNF_P_1 |
| 2 | DG_P_2 | Il sistema deve completare il caricamento delle pagine entro un intervallo massimo di 2 secondo FCP. | Prestazioni | RNF_P_2 |
| 1 | DG_S_1 | Il sistema deve garantire la sicurezza dei dati degli utenti secondo del GDPR. | Sicurezza | RNF_L_1 |
| 1 | DG_I_1 | Il sistema deve essere accessibile attraverso i principali (chromium based, firefox, safari) browser. | Implementazione | RNF_I_1 |
| 2 | DG_M_1 | Il sistema deve essere facilmente manutenibile grazie alla modularità del codice e alla scomposizione in sottosistemi. | Manutenzione | Scelta di Mgmt |



| Trade-Off | Descrizione |
|--|---|
| Sicurezza vs Performance | In quanto il sistema potrebbe gestire informazioni degli utenti private, una priorità più alta è data alla sicurezza dei dati rispetto alle performance complessive. |
| Manutenibilità vs Complessità | Si mira ad avere un sistema altamente manutenibile al fine di abbattere i costi di gestione post-deployment a scapito della complessità generale del sistema. |
| Accessibilità vs Complessità | Affinché il sistema sia adottato da un grande numero di utenti, si preferisce implementare poche funzionalità semplici rispetto a molte funzionalità di elevata complessità. |
| Costi vs Affidabilità | Non essendo un sistema critico, al fine di contenere i costi di gestione la disponibilità del sistema sarà ristretta nell'arco delle 24 ore al fine di effettuare manutenzione. |
| Centralizzazione vs Distribuzione | La distribuzione del sistema su più nodi offre maggiore ridondanza e migliori performance, ma comporta una complessità più elevata. Per mantenere la gestione semplice, si opta per il deployment su una singola macchina, in modo da mantenere comunque dei moduli separati che possono essere eseguiti su nodi diversi. |



| | |
|---|---|
| Stabilità delle tecnologie esistenti vs Tecnologie Emergenti | Per assicurare stabilità, si preferisce utilizzare tecnologie consolidate anziché emergenti, pur sacrificando alcune performance per semplificare lo sviluppo e ridurre lo sforzo del team. |
|---|---|

1.3 Definizioni, acronimi e abbreviazione

| Acronimo | Definizione |
|----------|-------------------------------|
| TC | Turing Careers |
| GO | Gestione Offerta |
| GU | Gestione Utente |
| RS | Ricerca |
| AU | Autenticazione |
| FCP | First Contentful Paint |
| SOW | Statement of Work |
| RAD | Requirement Analysis Document |

1.4 Riferimenti

Bernd Bruegge, Allen H. Dutoit - Object-Oriented Software Engineering.

Dispense del Prof. C. Gravino.

Turing Careers RAD.

Turing Careers SOW.



1.5 Panoramica

Il seguente documento è diviso in quattro sezioni:

- **Introduzione:** sezione nel quale si trova lo scopo del sistema, i design goal, ordinati per priorità, trade-offs, informazioni sui termini usati e riferimenti ad altri documenti.
- **Architettura di Sistemi Simili:** sezione dedicata ad una breve overview di sistemi che offrono servizi simili a quelli del sistema proposto.
- **Architettura del Sistema Proposto:** sezione dedicata all'analisi accurata dell'architettura del sistema proposto.

La sezione è ulteriormente divisa in:

- Scomposizione in Sottosistemi.
- Mapping hardware/software.
- Gestione dei dati persistenti.
- Controllo degli accessi e sicurezza.
- Flusso di controllo globale.
- Condizioni di Boundary.
- **Servizi dei Sottosistemi:** sezione dedicata al mappare i servizi offerti dal sistema ai relativi sottosistemi in cui questi verranno implementati.
- **Glossario:** sezione dedicata alla costruzione di un dizionario dei termini usati (con relative descrizioni) nel documento allo scopo di garantire massima chiarezza delle informazioni presentate.



2. Architettura del Sistema Corrente

Sistemi simili che offrono le medesime funzionalità di Turing Careers sono:

- Indeed.
- LinkedIn.

Questi due sistemi offrono la possibilità di pubblicare e candidarsi ad offerte di lavoro oltre che a poter contattare potenziali candidati/datori di lavoro. LinkedIn fornisce questi servizi secondariamente tramite il suo sottosistema LinkedIn Talent Solutions, mentre Indeed fa di questi servizi la sua componente core. A differenza di questi, Turing Careers si propone come portale esclusivamente per lavori nel settore IT, non implementa funzionalità di Social Media e mira ad avere un sistema di ricerca e raccomandazione di offerte di lavoro e sviluppatori efficace.



3. Architettura del Sistema Proposto

3.1 Panoramica

L'architettura scelta per l'applicazione web è il modello **Three-Tier**. Questa selezione deriva dall'obiettivo di separare chiaramente le diverse funzionalità dell'applicazione in tre strati distinti: la presentazione, l'elaborazione e i dati.

I benefici dell'architettura sono:

- Separazione delle Responsabilità: La suddivisione chiara dei compiti tra i vari strati consente una gestione più efficiente delle funzionalità specifiche.
- Facilità di Manutenzione: La separazione dei livelli facilita la manutenzione e l'aggiornamento di singoli componenti senza influire sugli altri strati.
- Scalabilità e Riutilizzo del Codice: La struttura a tre livelli agevola la scalabilità orizzontale e verticale, oltre a favorire il riutilizzo del codice in parti specifiche dell'applicazione.

È stato inoltre deciso di realizzare una **Architettura Chiusa**, optando quindi per un sistema meno performante ma più sicuro e mantenibile e composto da sottosistemi meno accoppiati.

I Framework usati per implementare il sistema saranno:

- **JakartaEE** per la gestione del back-end del sistema Core, essendo una delle tecnologie più usate e stabili nel contesto Web.
- **Apache Tomcat** come Web Container del sistema core, essendo una tecnologia stabile e facile da usare.
- **MySQL** come RDBMS.
- **Fast API** e **Uvicorn** per esporre il sistema di Ricerca e Raccomandazione con bassi costi di sviluppo.
- **scikit-learn** per il sottosistema di Ricerca e Raccomandazione dato che fornisce gli algoritmi di Machine Learning necessari.
- **SQLAlchemy** per interazione con il database del sottosistema di ricerca, essendo lo standard Python per ORM.



Il sistema dovrà inoltre interfacciarsi con la seguente componente esterna:

- **TalkJS** per la gestione della messaggistica tra utenti, per garantire le funzionalità della messaggistica senza dover complicare il processo di sviluppo.

3.2. Decomposizione in sottosistemi

Gestione Utente (GU): il sottosistema si occupa della modifica, eliminazione del profilo utente, visualizzazione dei profili, salvataggio di profili sviluppatori da parte di datori di lavoro e creazione di conversazioni tra utenti tramite servizio esterno (TalkJS).

Autenticazione (AU): il sottosistema si occupa di registrazione e autenticazione degli utenti.

Gestione Offerta (GO): il sottosistema si occupa delle funzionalità riguardanti la creazione, modifica, archiviazione e terminazione delle offerte, inoltre per gli sviluppatori fornisce la visualizzazione delle offerte, oltre che l'invio di candidature da parte di sviluppatori.

Ricerca (RS): si occupa delle funzionalità riguardanti la ricerca e la raccomandazione di offerte e di sviluppatori.

TalkJS: Componente Off-The-Shelf dedicata all'implementazione di messaggistica real time tra utenti del sito. Scelta per la sua facilità di implementazione e per la sua sicurezza.

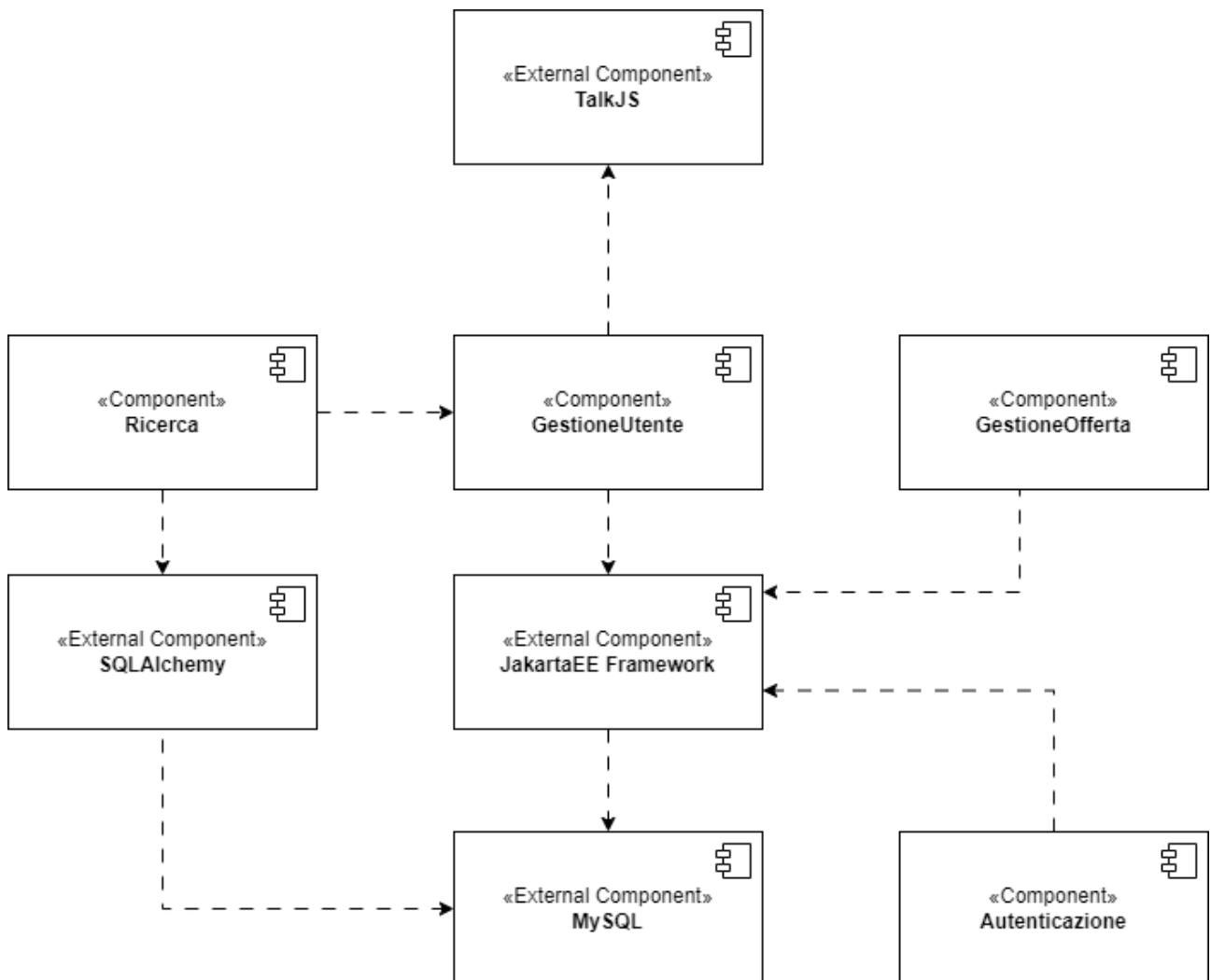
JakartaEE: framework Enterprise per Java utilizzato per gestire l'accesso al Database e Aspect-Oriented Programming.

SQLAlchemy: framework per accesso al database MySQL.

MySQL: RDBMS per la gestione di dati persistenti.

Component Diagram:

I sottosistemi appena descritti sono mostrati nel seguente Component Diagram in modo da visualizzarne le dipendenze:



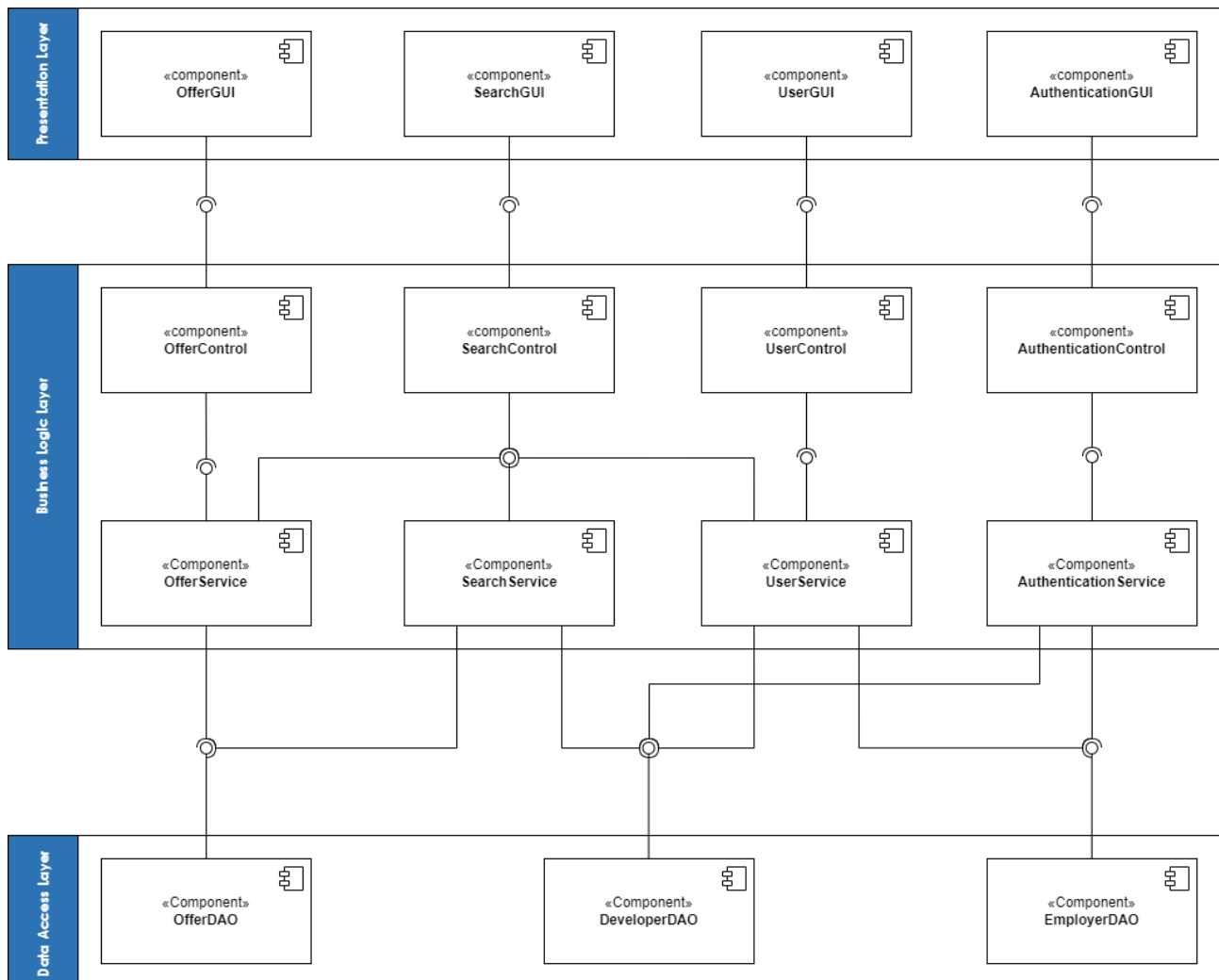


Dipendenze nel Component Diagram:

- **Autenticazione** dipende da JakartaEE Framework per la gestione dei dati persistenti.
- **Gestione Utente** dipende da JakartaEE Framework per la gestione dei dati persistenti, da TalkJS per realizzare il servizio di conversazione tra utenti.
- **Gestione Offerta** dipende da JakartaEE Framework per la gestione dei dati persistenti.
- **Ricerca** dipende dalla Gestione Utente per decidere quale servizio di ricerca esporre all'utente (Ricerca sviluppatori per datori di lavoro e viceversa). Dipende inoltre da SQLAlchemy per il recupero di informazioni persistenti dal Database MySQL.
- **JakartaEE Framework e SQLAlchemy** dipendono da MySQL in quanto è usato come RDBMS.

Architecture Diagram:

Segue il diagramma architetturale del sistema.



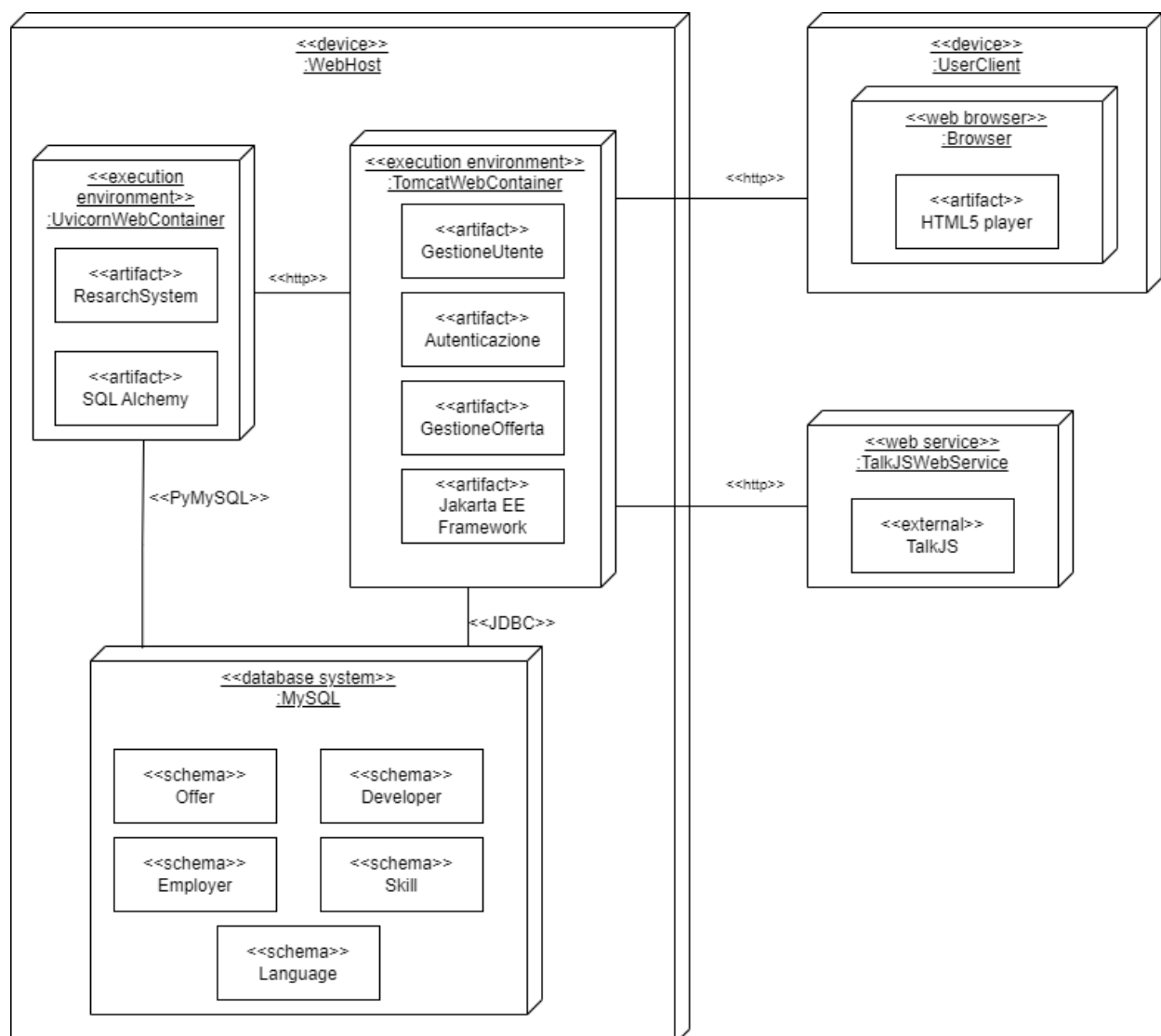
- Nel **Presentation Layer** si trovano le componenti che realizzano le GUI del sistema.
- Nel **Business Logic Layer** si trovano le componenti che controllano il flusso di esecuzione delle richieste fatte dagli utenti a livello di presentazione e le componenti che gestiscono la logica di business.
- Nel **Data Access Layer** si trovano le componenti DAO che realizzano le operazioni **CRUD** e le funzioni di **retrieve** delle entità.

Scelte e Considerazioni:

- E' stato scelto di mantenere un'alta **coesione** delle componenti a discapito di **accoppiamento più alto** tra di esse. Un esempio sono le componenti **User Service e Authentication Service**.

3.3 Mapping hardware/software

Deployment Diagram:





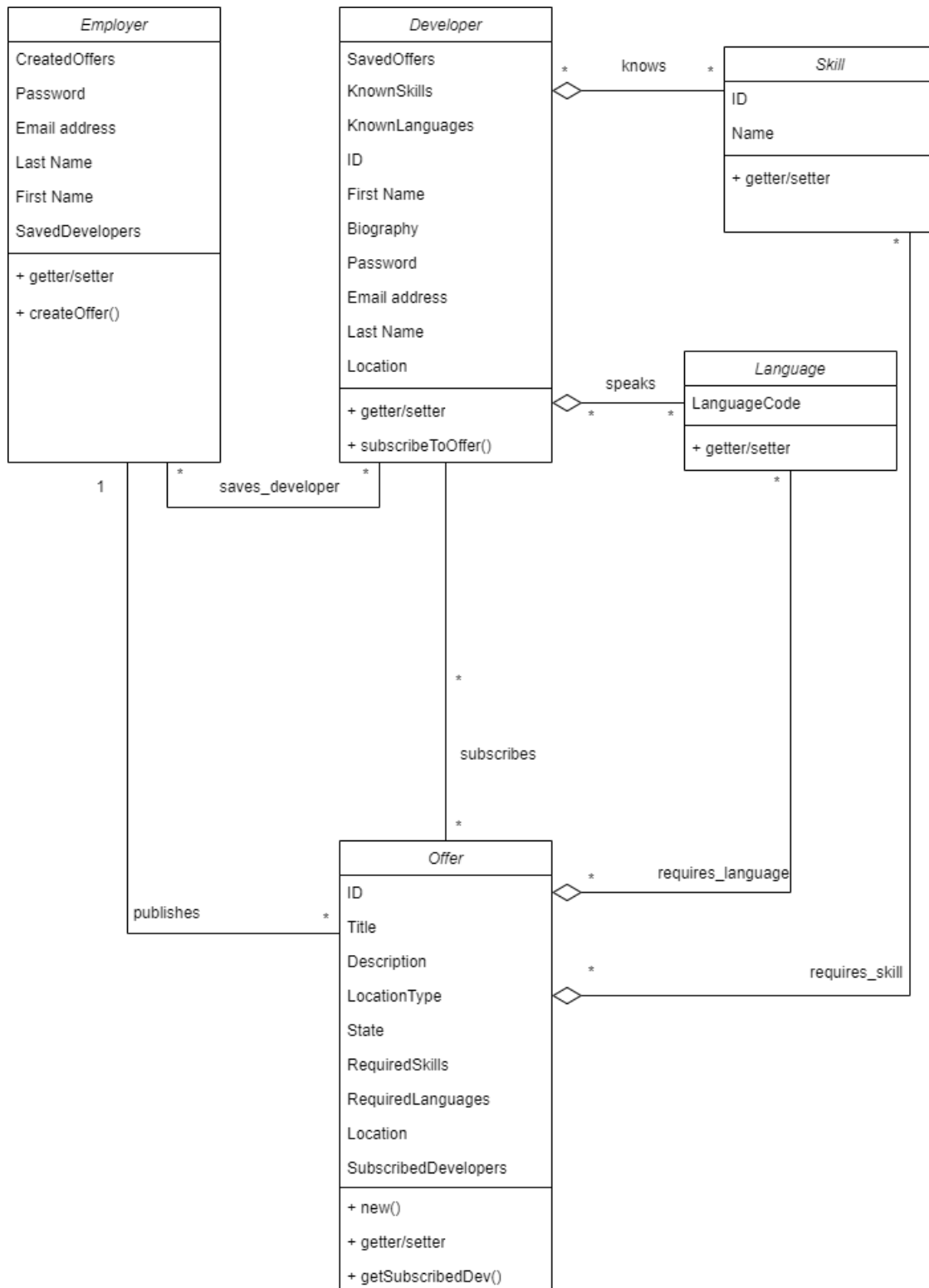
Il sistema viene dispiegato su di un'unica macchina con sistema operativo *Windows 10* dove si trovano:

- il Web Container principale (**Apache Tomcat**), dedicato al sistema Core che si occupa di fornire le risorse allo UserClient.
- il Web Container secondario (**Uvicorn**) dove si trova il sistema di Raccomandazione, riceve le richieste relative alla ricerca dal sistema Core.
- il Database **MySQL**, che conterrà le informazioni persistenti del sistema.

Il sistema si interfacerà con due componenti esterne che sono:

- **TalkJS** per fornire un servizio di messaggistica istantanea tra utenti.

3.4 Class Diagram Ristrutturato





Il class diagram è stato ristrutturato a seguito della scomposizione in sottosistemi:

Si è preferito non specificare ancora il tipo degli attributi delle entità per avere più libertà nelle fasi successive dello sviluppo, quando in base a scelte di implementazione saranno determinati i tipi più adatti. Seguono le modifiche alle entità e le relazioni nel database.

- **Language** è stata resa un'entità indipendente che contiene un codice univoco che identifica una lingua.
 - Essendo un requisito sia per Developer che per Offer è in relazione di aggregazione con entrambe le entità, la relazione è di tipo molti a molti.
- **Skill** è diventata una entità che contiene ID e Nome.
 - Analogamente a Language, è in relazione di aggregazione con Offer e Developer, anche questa relazione molti a molti.
- La gerarchia **User** è stata risolta spostando gli attributi dell'entità padre nelle entità figlie **Developer** e **Employer**.

3.5 Gestione dei Dati Persistenti

In quanto i dati persistenti nel database e le loro relazioni risultano immutate rispetto quelle del class diagram ristrutturato al punto precedente, si è scelto di non riportare un diagramma Entità-Relazione in quanto elemento ridondante.

Per via della natura strutturata dei dati è stato scelto **MySQL** come RDBMS del sistema. Questa scelta è stata anche dettata dalla familiarità del team di sviluppo con la tecnologia, in linea con il Trade-Off "Stabilità delle Tecnologie Esistenti vs. Tecnologie Emergenti".



3.6 Controllo degli accessi e sicurezza

Di seguito viene mostrata la tabella degli accessi (matrice di accesso) che modella i diritti di accesso su ogni classe. Le colonne rappresentano gli attori, le righe le classi. Ogni entry (attore, classe) contiene le operazioni consentite da quell'attore sulle istanze di quella classe.

| Oggetti | Attori | Sviluppatori | Datori di Lavoro |
|-----------------------|--------|---------------------------------------|---|
| Offerta | | InviaCandidatura VisualizzaOfferta | CreaOfferta ModificaOfferta SospendiOfferta AttivaOfferta ChiudiOfferta |
| ProfiloSviluppatore | | ModificaProfilo EliminaAccount | VisualizzaProfilo SalvaProfilo ContattaUtente |
| ProfiloDatoreDiLavoro | | VisualizzaProfilo ContattaUtente | ModificaProfilo EliminaAccount |
| Autenticazione | | SignIn SignOut SignUp | SignIn SignOut SignUp |
| Ricerca | | CercaOfferta | CercaSviluppatore |



3.7 Controllo flusso globale sistema

Il sistema è di tipo **event driven**, ciò significa che il flusso delle operazioni del sistema ha origine dall'interazione dell'utente con le GUI. La richiesta viene quindi inoltrata allo strato successivo, quello di controllo, che si occupa di processare la richiesta, e fornire i servizi richiesti. Per fare ciò, lo strato di business si interfaccia con lo strato dei dati per richiedere o immagazzinare informazioni persistenti.

3.8 Condizioni limite

| | | | |
|------------------------------|---|----------|-----------------|
| UCBC_CR | Avvio Sistema Core | Data | 12/11/23 |
| | | Versione | 1.1 |
| | | Autore | Claudio Gaudino |
| Descrizione | Use Case che descrive l'avvio del sistema core. | | |
| Attore Principale | Admin | | |
| Entry Condition | L'Admin ha installato il file WAR del Sistema Core L'Admin ha installato Tomcat e Il database MySQL è disponibile. | | |
| Exit Condition (on success) | Il sistema ha reso disponibile la visione di una lista, che può anche essere vuota, di sviluppatori. | | |
| Exit Condition (on failure) | Il sistema core è avviato con successo. | | |
| User Priority | Alta | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |



| | |
|-----|---|
| 1.1 | <p>L'Admin copia il file WAR nella directory dedicata al deploy nel percorso di installazione di Tomcat, poi apre un terminale nel percorso di installazione e esegue:</p> <pre>- create-domain --adminport [PORT] --user admin [domain-name]</pre> <pre>- asadmin start-domain [domain-name]</pre> |
| 1.2 | <p>Il Web Server Tomcat rende disponibile l'applicazione web.</p> |

| | | | |
|-----------------------------|---|----------|------------------|
| UCBC_RS_1 | Startup Sistema Ricerca | Data | 12/12/23 |
| | | Versione | 1.1 |
| | | Autore | Antonino Lorenzo |
| Descrizione | Use Case che descrive l'avvio del sottosistema di ricerca | | |
| Attore Principale | Admin | | |
| Entry Condition | <ul style="list-style-type: none"> - L'Admin ha installato il package contenente il sorgente per il Sistema di Ricerca - Python >= 3.12 è installato - Le librerie elencate in requirements.txt sono installate - Il database MySQL è disponibile - È disponibile un terminale bash | | |
| Exit Condition (on success) | La API di Ricerca è eseguita in background ed è disponibile per le richieste. | | |
| Exit Condition (on failure) | L'avvio della API è fallito. | | |



| | |
|--|--|
| Rilevanza/User Priority | Alta |
| Frequenza Stimata | 10/anno |
| Flusso di Eventi Principale: | |
| 1.1 | <p>L'admin esegue il seguente comando per eseguire lo script che si occupa dell'avvio dell'API di Ricerca:</p> <pre>nohup python <u>launcher.py</u> -db-address [ip:port] -db-user [Nome Utente] -db-psw [Password] > log.txt & echo \$! > pid.txt</pre> |
| 1.2 | <p>Lo script si connette al database MySQL all'indirizzo db-address con le credenziali db-user e db-psw, tramite SQLAlchemy; poi avvia un WebServer tramite Uvicorn sull'indirizzo locale alla porta port per rendere le funzionalità del sistema disponibili.</p> |
| Flusso di Eventi di Errore 1: Credenziali Errate | |
| 2.1 | <p>L'admin esegue il seguente comando per eseguire lo script che si occupa dell'avvio dell'API di Ricerca:</p> <pre>nohup python <u>launcher.py</u> -api-port [port] -db-address [ip:port] -db-user [Nome Utente] -db-psw [Password] > log.txt & echo \$! > pid.txt</pre> |
| 2.2 | <p>Lo script tenta di connettersi al database MySQL all'indirizzo db-address con le credenziali db-user e db-psw, ma le credenziali sono errate.</p> |
| Flusso di Eventi di Errore 2: Indirizzo Errato | |
| 3.1 | <p>L'admin esegue il seguente comando per eseguire lo script che si occupa dell'avvio dell'API di Ricerca:</p> |



| | |
|-----|--|
| | <pre>nohup python <u>launcher.py</u> -api-port [port] -db-address [ip:port] -db-user [Nome Utente] -db-psw [Password] > log.txt & echo \$! > pid.txt</pre> |
| 3.2 | Lo script tenta di connettersi al database MySQL all'indirizzo db-address con le credenziali db-user e db-psw , ma l'indirizzo non è corretto |

| | | | |
|------------------------------|--|----------|------------------|
| UCBC_DB_1 | Avvia Database | Data | 15/12/23 |
| | | Versione | 1.0 |
| | | Autore | Antonio Pagnotta |
| Descrizione | Use Case che descrive l'avvio del database. | | |
| Attore Principale | Admin | | |
| Entry Condition | L'Admin ha un file database.sql che contiene schemi delle tabelle e queries. L'Admin ha installato MySQL. | | |
| Exit Condition (on success) | Il database è stato creato, se non esiste, e avviato. | | |
| Exit Condition (on failure) | Non è stato possibile avviare il database. | | |
| Rilevanza/User Priority | Alta | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | L'Admin avvia MySQL. | | |
| 1.2 | L'Admin inserisce il seguente comando <i>mysql -u[username] -p[password] < [percorso/database.sql]</i> | | |
| 1.3 | Il sistema MySQL crea il database se non esiste e lo | | |



| | |
|---|---|
| | rende disponibile all'indirizzo locale su una porta libera. |
| Flusso di Eventi di Errore 1: Credenziali Errate | |
| 1.1 | L'Admin avvia MySQL. |
| 1.2 | L'Admin inserisce il seguente comando: <code>mysql -u[username] -p[password] < [percorso database.sql]</code> |
| 1.3 | Il sistema MySQL non permette nessuna operazione dato che le credenziali sono sbagliate. |

| | | | |
|------------------------------|---|----------|------------------|
| UCBC_DB_2 | Chiusura Database | Data | 15/12/23 |
| | | Versione | 1.0 |
| | | Autore | Antonio Pagnotta |
| Descrizione | Use Case che descrive la chiusura del database. | | |
| Attore Principale | Admin | | |
| Entry Condition | Il servizio MySQL è in esecuzione. | | |
| Exit Condition (on success) | Il database è stato chiuso | | |
| Exit Condition (on failure) | Non è stato possibile chiudere il database. | | |
| Rilevanza/User Priority | Alta | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | L'Admin inserisce il seguente comando <i>mysqladmin -u[username] -p[password] shutdown</i> | | |
| 1.2 | Il sistema MySQL chiude il servizio del database. | | |



| | | | |
|------------------------------|--|----------|------------------|
| UCBC_RS_2 | Shutdown Sistema Ricerca | Data | 15/12/23 |
| | | Versione | 1.0 |
| | | Autore | Antonino Lorenzo |
| Descrizione | Use Case che descrive lo spegnimento del sottosistema di raccomandazione | | |
| Attore Principale | Admin | | |
| Entry Condition | Il Sistema di Ricerca è stato spento con successo. | | |
| Exit Condition (on success) | La API di Ricerca non è più disponibile | | |
| Exit Condition (on failure) | Lo spegnimento della API è fallito. | | |
| Rilevanza/User Priority | Alta | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | L'admin esegue il seguente comando dal percorso in cui è stato avviato il Sistema di Ricerca: <i>kill \$(cat pid.txt)</i> | | |
| 1.2 | Il processo che esegue il Sistema di Ricerca viene terminato ed internamente viene gestita la deallocazione delle risorse. | | |



| | | | |
|------------------------------|--|----------|-----------------|
| UCBC_MSG_1 | Fallimento TalkJS | Data | 18/12/23 |
| | | Versione | 2.0 |
| | | Autore | Claudio Gaudino |
| Descrizione | Use Case che descrive come si comporta il sistema nel caso in cui il servizio fornito da TalkJS non è disponibile. | | |
| Attore Principale | Sistema | | |
| Entry Condition | Il sistema richiede le funzionalità di messaggistica a TalkJS. | | |
| Exit Condition (on success) | Il sistema gestisce l'errore ed informa l'utente. | | |
| Exit Condition (on failure) | Il sistema non gestisce l'errore e l'utente viene informato da un errore 500. | | |
| Rilevanza/User Priority | Alta | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | Il sistema istanzia la comunicazione con TalkJS per fornire le funzionalità di messaggistica ad un utente. | | |
| 1.2 | TalkJS non è disponibile ed il sistema viene notificato dell'errore. | | |
| 1.3 | Il sistema mostra all'utente un messaggio di errore che riporta la temporanea indisponibilità delle funzioni di messaggistica. | | |



| | | | |
|--|--|----------|------------------|
| UCBC_DB_3 | Fallimento Database | Data | 18/12/23 |
| | | Versione | 2.0 |
| | | Autore | Antonio Pagnotta |
| Descrizione | Use case che descrive un errore di comunicazione col database. | | |
| Attore Principale | Sistema | | |
| Entry Condition | Il sistema ha tentato di effettuare una operazione sul database. | | |
| Exit Condition (on success) | Il database è di nuovo disponibile. | | |
| Exit Condition (on failure) | Il database non è disponibile ed è necessario un intervento manuale. | | |
| Rilevanza/User Priority | Bassa | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | Il DBMS subisce un fallimento critico | | |
| 1.2 | Il sistema core rileva il fallimento e prova a riavviare il DBMS | | |
| 1.3 | Il DBMS è riavviato e disponibile | | |
| Flusso degli Eventi di Errore: UnableToRestart | | | |
| Descrizione | Il DBMS fallisce e il sistema core non riesce a riavviarlo, notifica quindi l'amministratore del problema. | | |
| 2.1 | Il DBMS subisce un fallimento critico. | | |



| | |
|-----|--|
| 2.2 | Il sistema core rileva il fallimento e prova a riavviare il DBMS |
| 2.3 | Il DBMS non è riavviato |
| 2.4 | Il sistema core notifica l'amministratore del problema e segnala il disservizio agli utenti. |

| | | | |
|------------------------------|--|----------|--------------------|
| UCBC_RS_3 | Fallimento Sistema di Ricerca | Data | 18/12/23 |
| | | Versione | 2.0 |
| | | Autore | Jacopo Passariello |
| Descrizione | Use Case che descrive un errore di comunicazione col Sistema di Ricerca. | | |
| Attore Principale | Sistema Core, Sistema di Ricerca, Admin | | |
| Entry Condition | Il sistema ha effettuato una richiesta al servizio di ricerca. | | |
| Exit Condition (on success) | Il sistema ha riavviato il sistema di ricerca con successo. | | |
| Exit Condition (on failure) | Il sistema ha tentato il riavvio del sistema di ricerca ma ha fallito, gli utenti che si connettono sono notificati del disservizio. | | |
| Rilevanza/User Priority | Bassa | | |
| Frequenza Stimata | 10/anno | | |
| Flusso di Eventi Principale: | | | |
| 1.1 | Il sistema di raccomandazione subisce un fallimento critico. | | |



| | |
|---|--|
| 1.2 | Il sistema core rileva il fallimento e prova a riavviare il sistema di ricerca. |
| 1.3 | Il sistema di raccomandazione è riavviato e disponibile. |
| Flusso degli Eventi di Errore: UnableToRestart | |
| Descrizione | Il sistema di raccomandazione fallisce e il sistema core non riesce a riavviarlo, notifica quindi l'amministratore del problema. |
| 2.1 | Il sistema di raccomandazione subisce un fallimento critico. |
| 2.2 | Il sistema core rileva il fallimento e prova a riavviare il sistema di ricerca. |
| 2.3 | Il sistema di raccomandazione non è riavviato. |
| 2.4 | Il sistema core notifica l'amministratore del problema e segnala il disservizio agli utenti. |



4. Servizi dei Sottosistemi

Sottosistema: Autenticazione

| Servizio | Descrizione | Interfaccia |
|----------------------|---|-----------------------|
| SignIn | Servizio adibito al login (connessione) dell'utente all'interno del sistema | AuthenticationControl |
| SignOut | Servizio adibito al logout (disconnessione) dell'utente all'interno del sistema | AuthenticationControl |
| SignUp | Servizio adibito alla registrazione dell'utente all'interno del sistema. | AuthenticationControl |
| CreazioneProfiloChat | Servizio adibito alla creazione di un profilo TalkJS per l'utente al fine di accedere al servizio esterno di Chat | AuthenticationControl |

Sottosistema: Ricerca

| Servizio | Descrizione | Interfaccia |
|---------------------|--|----------------|
| SearchOffer | Servizio fornito agli sviluppatori per effettuare la ricerca e raccomandazione di Offerte. | SearchControl |
| SearchDeveloper | Servizio fornito ai datori di lavoro per effettuare la ricerca di Profili di Sviluppatori. | SearchControl |
| Recommend Developer | Servizio fornito ai datori di lavoro per effettuare la raccomandazione di Profili di Sviluppatori in base alle Offerte pubblicate. | SearchControl. |



Sottosistema: Gestione Offerta

| Servizio | Descrizione | Interfaccia |
|-------------------|--|--------------|
| CreaOfferta | Servizio usato dai datori di lavoro per la creazione e compilazione iniziale dell'offerta. | OfferControl |
| ModificaOfferta | Servizio usato dai datori di lavoro per la modifica di un'offerta. | OfferControl |
| ChiudiOfferta | Servizio usato dai datori di lavoro per chiudere definitivamente un'offerta. | OfferControl |
| VisualizzaOfferta | Servizio fornito agli utenti per visualizzare una offerta. | OfferControl |
| InviaCandidatura | Servizio fornito agli sviluppatori per inviare una candidatura ad un'offerta di lavoro. | OfferControl |

Sottosistema: Gestione Utente

| Servizio | Descrizione | Interfaccia |
|-------------------|--|-------------|
| VisualizzaProfilo | Servizio fornito agli utenti per visualizzare il profilo di un altro utente. | UserControl |
| ModificaProfilo | Servizio fornito agli utenti per modificare le informazioni relative al proprio profilo | UserControl |
| EliminaAccount | Servizio fornito agli utenti per eliminare il proprio account | UserControl |
| SalvaProfiloDev | Servizio fornito ai datori di lavoro al fine di salvare il profilo di uno sviluppatore in una lista. | UserControl |
| ContattaUtente | Servizio fornito agli utenti per creare una nuova conversazione con un altro utente tramite TalkJS. | UserControl |



5. Glossario

| Nome | Descrizione |
|------|---|
| GDPR | General Protection Data Regulation, regolamento dell'Unione europea in materia di trattamento dei dati personali e di privacy. |
| FCP | First Contentful Paint, misura il tempo trascorso tra la prima visita dell'utente alla pagina e il rendering di qualsiasi parte dei contenuti della pagina sullo schermo. |