Test Plan

Agency Formation

Riferimento	AF_TP_v1.3
Versione	1.3
Data	10/12/2021
Destinatario	Filomena Ferrucci (FF), Fabio Palomba (FP)
Presentato da	GC, LG, GL, MN, DP, ES, PS
Approvato da	AC, VP

Revision History

Data	Versione	Descrizione	Autori
10/12/2021	1.0	Prima stesura	GL, MN, PS
14/12/2021	1.1	Stesura Testing di integrazione: modifica descrizione e approccio (aggiunta testing unità)	PS, ES
18/12/2021	1.2	Aggiunta di nuovi Test Frame causati dalla modifica dello schema ER, aggiunta di nuovi attributi	PS
10/01/2022	1.3	Aggiunta nuovo Test Frame per la verifica dell'e-mail nel database	PS, LG

Tabella dei contenuti

1. Introduzione

Agency Formation è una piattaforma web nata per supportare le attività di reclutamento e di formazione dei dipendenti dell'azienda AF Consulting.

L'azienda mette a disposizione questa piattaforma per supportare attività come:

- Registrazione da parte di un candidato e login/logout da parte di un utente già registrato
- Caricamento del CV, attestati e certificazioni da parte dei candidati.
- Accettazione/Rifiuto del curriculum da parte degli HR con possibilità di fissare un colloquio.
- Creazione/Scioglimento dei Team da parte dei Team Manager con selezione ed inserimento dei dipendenti al loro interno e specifica delle competenze necessarie per lo svolgimento del progetto.
- Caricamento materiale di formazione per i dipendenti membri dei Team da parte degli HR.
- Modifica del profilo da parte dei dipendenti, cambiando alcuni dati personali e aggiungendo / rimuovendo (in caso di perdita) le skill.

Le attività di testing sono pianificate per le seguenti gestioni:

- Gestione Reclutamento
- Gestione Team
- Gestione Formazione
- Gestione Autenticazione

Lo scopo della piattaforma Agency Formation è quello di assicurare l'efficienza e agevolare l'interazione tra tutti gli utenti coinvolti.

2. Relazione con altri documenti

Per la corretta individuazione dei test case si fa riferimento ad altri documenti:

- **Relazioni con il Requirements Analysis Document (RAD)**

La relazione tra Test Plan e RAD riguarda in particolare: requisiti funzionali, requisiti non funzionali, gli Use Case del sistema. I test saranno eseguiti su ogni funzionalità e caso d'uso e terranno conto delle specifiche espresse nel RAD

- **Relazioni con il System Design Document (SDD)**

La relazione tra Test Plan e SDD riguarda in particolare la suddivisione del sistema in sottosistemi. I test case devono attenersi a questa suddivisione.

- **Relazioni con l'Object Design Document (ODD)**

La relazione tra Test Plan e ODD riguarda le dipendenze tra gli oggetti individuate in quest'ultimo. Le stesse dipendenze saranno oggetto di testing.

3. Panoramica del sistema

Il software proposto è una piattaforma web con lo stile architetturale Three Tier.
I linguaggi che verranno utilizzati per lo sviluppo del sistema sono: Java, HTML5, CSS3, JS, JSP mentre per la gestione del Database verrà usato MySQL.
Come specificato anche nel System Design Document, il server con il quale si interfaccerà il sistema proposto sarà Tomcat.

4. Feature da testare/da non testare

Nell'ambito del testing di sistema saranno verificate le funzionalità principali per ciascuna gestione. Di seguito sono elencate le funzionalità scelte:

- Gestione Reclutamento:
  - Caricamento Curriculum con attestati e certificazioni
  - Accettazione Candidatura
- Gestione Team:
  - Creazione team
  - Aggiunta dipendenti nel Team
- Gestione Formazione:
  - Specifica delle competenze
  - Caricamento materiale di formazione
  - Aggiunta skill
- Gestione Autenticazione:
  - Registrazione candidato
  - Login

5. Pass/Fail criteria

Un test avrà successo (pass) se l'esecuzione di un caso di test determina una failure, ovvero si ottiene un risultato diverso da quello atteso.
Al contrario, un test fallirà (fail) quando il risultato osservato è uguale a quello atteso.
Tutto il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

- Testare tutti i requisiti funzionali ad alta priorità;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 75%.

6. Approccio

Il testing dell'intero sistema si compone di tre fasi:

- Testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti, quest'ultimo servirà a verificare che il sistema soddisfi le richieste del committente;
- Testing di integrazione, per testare l'integrazione dei vari sottosistemi;
- Testing di unità, eseguito sulle singole componenti, in modo da testare nello specifico la correttezza di ciascuna unità, cercando di rovinare il corretto funzionamento delle singole unità di codice.

Verranno progettati nell'ordine appena definito, ma verranno eseguiti in ordine inverso. Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione; durante la fase implementativa avverrà la progettazione dei casi di test di unità. Durante lo sviluppo saranno eseguite periodiche attività di revisione sul codice prodotto, così da avere immediato riscontro degli errori di implementazione. Quando verrà introdotta una nuova funzionalità nel codice, verranno eseguiti tutti i test di unità legati alla funzionalità e i test di integrazione che la coinvolgono. Ogni volta che verranno effettuate delle modifiche su una componente, verranno rilanciati i test di unità e di integrazione delle componenti dipendenti da quella modifica.

Testing di sistema

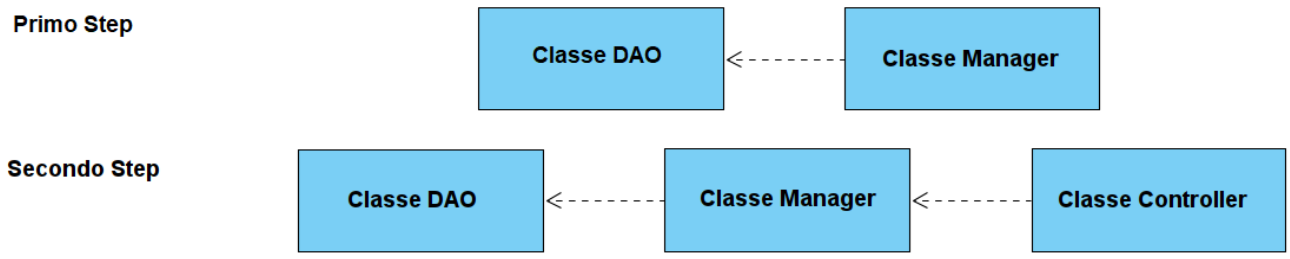
Lo scopo principale di questa fase è quello di dimostrare che l'intero sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non funzionali, descritti nel Requirement Analysis Document (RAD). Per questo tipo di testing sarà utilizzato il tool *Selenium*, che permette di registrare delle interazioni utente su un browser, e così di implementare ed eseguire i casi di test di sistema. Durante il testing di sistema, il deploy del server sarà in localhost.

Testing di integrazione

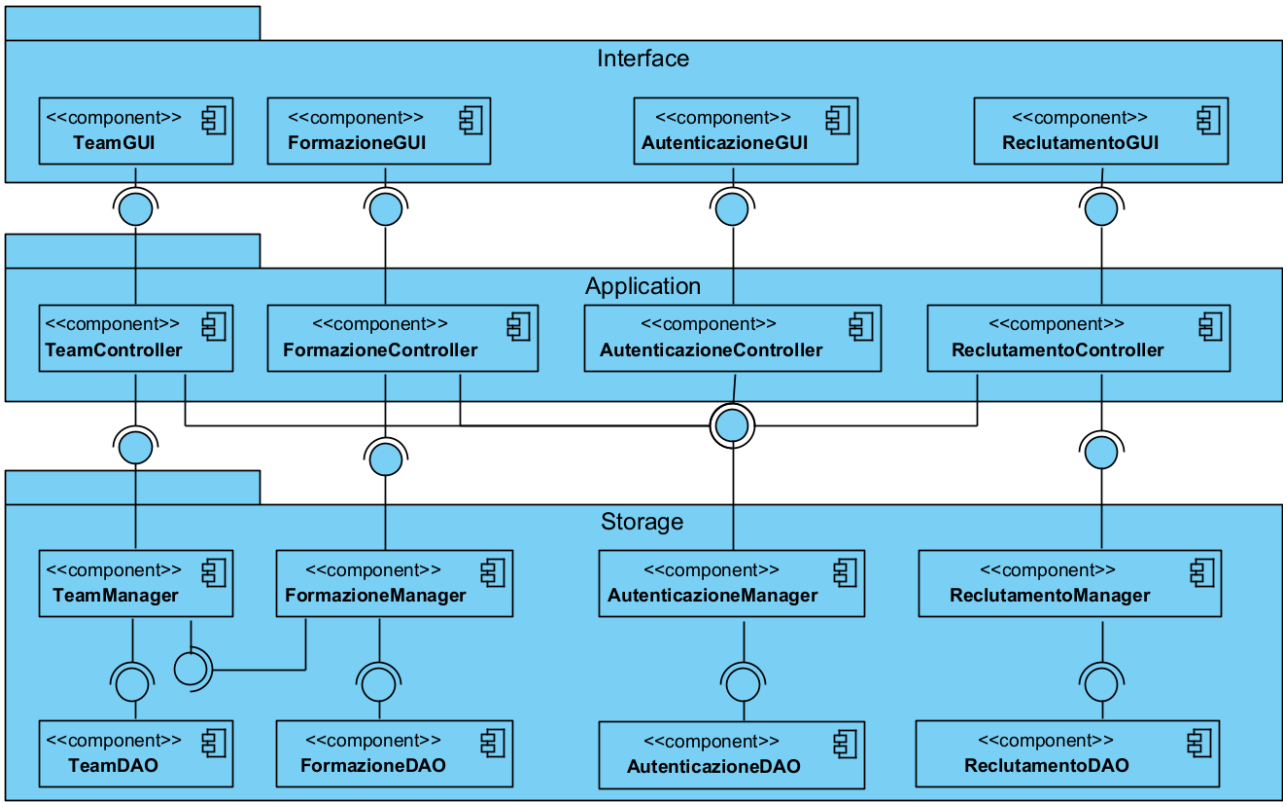
Per il testing di integrazione, verrà utilizzato un approccio bottom-up, metodo ritenuto più adatto per un software basato su Object Oriented. La definizione dei test case avverrà tramite il framework *JUnit*, mentre verrà usato *Mockito* per il mocking. L'automatizzazione del run dei test sarà gestita da *Gradle*, ed infine come tool di misurazione e report coverage sarà utilizzato JaCoCo. Il testing di integrazione sarà il medesimo per tutte le componenti da testare. Nello specifico si procederà prima con il test delle classi Manager e successivamente con il test delle classi Controller. Durante questa seconda esecuzione, la chiamata al Controller sarà simulata usando Mockito.

L'esecuzione dei casi di test di integrazione avverrà dopo aver eseguito tutti i casi di test unitari.

Di seguito viene presentato un esempio grafico di test di integrazione diviso nei due steps.



In generale, il test di integrazione sarà compreso nella stessa classe di test di quelli unitari. Per ciò che concerne le dipendenze tra i sottosistemi, di seguito si riporta il diagramma architetturale.



Testing di unità

Per il testing di unità la strategia prevista consiste nel testare ogni metodo delle classi del sistema. Da esse, sono escluse le interfacce e le classi entity, poiché quest'ultime presentano solo metodi getters e setters. I casi di test saranno definiti attraverso un approccio black-box e saranno documentati direttamente nel codice, attraverso l'uso del framework per il testing di classi Java *JUnit*. Verrà definita una classe di test per ciascuna classe sorgente. Se l'esecuzione di tutti i test unitari non porta alla branch coverage minima richiesta del 75%, verranno definiti altri casi di test attraverso un approccio white-box. Alcune tecnologie usate in tale fase saranno:

- *Mockito*: per la costruzione degli stub e l'isolamento della componente testata;
- *JaCoCo*: per il calcolo di metriche tra le quali la Branch Coverage;
- *Gradle*: per la build e l'esecuzione automatica dei tests.

7. Sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test.
Verranno inoltre specificate le attività di test che devono essere ripetute quando si riprende il test.

- **Criteri di sospensione**
  Il testing non verrà sospeso fino alla sua terminazione anche in caso di rilevazione di una failure.
- **Criteri di ripristino**
  Il testing verrà ripreso dopo aver risolto i fault individuati nella precedente esecuzione dei test.

8. Materiale di testing

L'hardware necessario per l'attività di test è un pc non necessariamente avente connessione ad Internet, in quanto il sistema non è stato ancora rilasciato

9. Test cases

In seguito, verranno elencati tutti i tipi di test case individuati suddivisi per gestione. Si assume che ci sia una dipendenza di tipo gerarchico tra i parametri dal primo all'ultimo.

9.1 Gestione Reclutamento

9.1.1 Caricamento Curriculum attestati e certificazioni

Parametro: File documento Formato: .+.pdf, .+.docx,. +.jpg, .+.png	
Selezionato [SD]	1. non selezionato [error] 2. selezionato [property SD_OK]
Dimensione [DD]	1. > 10MB [if SD_OK] [error] 2. > 0 KB and <= 10MB [if SD_OK] [property DD_OK]
Formato [FD]	1. non rispetta i vincoli [if SD_OK] [if DD_OK] [error] 2. rispetta i vincoli [if SD_OK] [if DD_OK] [property FD_OK]

Codice	Test Frame	Esito
TC_1.1_1	SD1	Errore: file non selezionato
TC_1.1_2	SD2, DD1	Errore: dimensione non valida
TC_1.1_3	SD2, DD2, FD1	Errore: formato non valido
TC_1.1_4	SD2, DD2, FD2	Corretto: file sottomesso

9.1.2 Accettazione candidatura

Parametro: DataOraColloquio	
DataOra [DO]	1. <= Data corrente e ora corrente [error] 2. <= Data corrente e (oraColloquio > 20:30 or oraColloquio < 8:00) [error] 3. > Data corrente e ora corrente and (oraColloquio <= 20:30 and oraColloquio >= 8:00) [property DO_OK]

Codice	Test Frame	Esito
TC_1.2_1	DO1	Errore: data e orario non disponibile
TC_1.2_2	DO2	Errore: data e orario non disponibile
TC_1.2_3	DO3	Corretto: candidatura accettata

9.2 Gestione Team

9.2.1 Creazione Team

Parametro: Nome Team **Formato: ^[A-Za-z ']+$**	
Lunghezza [LT]	1. < 1 or > 32 [error] 2. >= 1 and <= 32 [property LT_OK]
Formato [FT]	1. Non match [if LT_OK] [error] 2. Match [if LT_OK] [property FT_OK]
Parametro: Nome Progetto **Formato: ^[A-Za-z ']+$**	
Lunghezza [LP]	1. < 1 or > 32 [error] 2. >= 1 and <= 32 [property LP_OK]
Formato [FP]	1. Non match [if LP_OK] [error] 2. Match [if LP_OK] [property FP_OK]
Parametro: Descrizione **Formato: ^(.|\s)*[a-zA-Z]+(.|\s)*$**	
Lunghezza [LD]	1. < 1 or > 128 [error] 2. >= 1 and <= 128 [property LD_OK]
Formato [FD]	1. Non match [if LD_OK] [error] 2. Match [if LD_OK] [property FD_OK]
Parametro: Numero Membri	
Quantità [QM]	1. <= 0 [error] 2. > 0 [property QM_OK]

Codice	Test Frame	Esito
TC_2.1_1	LT1	Errore: lunghezza Team non valida
TC_2.1_2	LT2, FT1	Errore: formato non valido
TC_2.1_3	LT2, FT2, LP1	Errore: lunghezza non valida
TC_2.1_4	LT2, FT2, LP2, FP1	Errore: formato non valido
TC_2.1_5	LT2, FT2, LP2, FP2, LD1	Errore: lunghezza non valida
TC_2.1_6	LT2, FT2, LP2, FP2, LD2, FD1	Errore: formato non valido
TC_2.1_7	LT2, FT2, LP2, FP2, LD2, FD2, QM1	Errore: numero membri non valido
TC_2.1_8	LT2, FT2, LP2, FP2, LD2, FD2, QM2	Corretto: Team creato

9.2.2 Aggiunta dipendenti nel Team

Parametro: Membri Attuali	
Quantità [QM]	1. >= max [error] 2. >= 0 and < max [property QM_OK]

Codice	Test Frame	Esito
TC_2.2_1	QM1	Errore: numero membri esaurito
TC_2.2_2	QM2	Corretto: dipendente aggiunto

9.3 Gestione Formazione

9.3.1 Specifica delle competenze

Parametro: Competenza **Formato:** ^.*(?=.*[A-Za-z])(?=[.])+(?=.*[!#$%&? "]).*$	
Lunghezza [LC]	1. < 1 or > 512 [error] 2. > 0 and <= 512 [property LC_OK]
Formato [FN]	1. non rispetta i vincoli [if LC_OK] [error] 2. rispetta i vincoli [if LC_OK] [property FN_OK]

Codice	Test Frame	Esito
TC_3.1_1	LC1	Errore: lunghezza non valida
TC_3.1_2	LC2, FN1	Errore: formato non valido
TC_3.1_3	LC2, FN2	Corretto: competenza specificata

9.3.2 Caricamento materiale di formazione

Parametro: File documento **Formato: +.pdf, +.docx, +.jpg, +.png**	
Selezionato [SD]	1. non selezionato [error] 2. selezionato [property SD_OK]
Dimensione [DD]	1. > 10MB [if SD_OK] [error] 2. > 0 KB and <= 10MB [if SD_OK] [property DD_OK]
Formato [FD]	1. non rispetta i vincoli [if SD_OK] [if DD_OK] [error] 2. rispetta i vincoli [if SD_OK] [if DD_OK] [property FD_OK]

Codice	Test Frame	Esito
TC_3.2_1	SD1	Errore: file non selezionato
TC_3.2_2	SD2, DD1	Errore: dimensione non valida
TC_3.2_3	SD2, DD2, FD1	Errore: formato non valido
TC_3.2_4	SD2, DD2, FD2	Corretto: file sottomesso

9.3.3 Aggiunta skill

Parametro: Nome Skill **Formato:** ^.*(?=.*[A-Za-z])(?=[.])+(?=.*[!#$%&? "]).*$	
Lunghezza [LN]	1. < 1 or > 64 [error] 2. > 0 and <= 64 [property LN_OK]
Formato [FN]	1. non rispetta i vincoli [if LN_OK] [error] 2. rispetta i vincoli [if LN_OK] [property FN_OK]
Parametro: Livello Skill	
Livello [LV]	1. < 1 and > 5 [error] 2. >= 1and <= 5 [property LV_OK]
Parametro: Descrizione Skill **Formato:** ^.*(?=.*[A-Za-z])(?=[.])+(?=.*[!#$%&? "]).*$	
Lunghezza [LZ]	1. < 1 or > 512 [error] 2. > 0 and <= 512 [property LZ_OK]
Formato [FM]	1. non rispetta i vincoli [if LZ_OK] [error] 2. rispetta i vincoli [if LZ_OK] [property FZ_OK]

Codice	Test Frame	Esito
TC_3.3_1	LN1	Errore: lunghezza non valida
TC_3.3_2	LN2, FN1	Errore: formato non valido
TC_3.3_3	LN2, FN2, LV1	Errore: livello non valido
TC_3.3_4	LN2, FN2, LV2, LZ1	Errore: lunghezza non valida
TC_3.3_5	LN2, FN2, LV2, LZ2, FM1	Errore: formato non valido
TC_3.3_6	LN2, FN2, LV2, LZ2, FM2	Corretto: Skill sottomessa

9.4 Gestione Autenticazione

9.4.1 Registrazione candidato

Parametro: Nome **Formato: ^[A-Za-z ']+$**	
Lunghezza [LN]	1. < 1 or > 32 [error] 2. > 0 and <= 32 [property LN_OK]
Formato [FN]	1. non rispetta i vincoli [if LN_OK] [error] 2. rispetta i vincoli [if LN_OK] [property FN_OK]
Parametro: Cognome **Formato: ^[A-Za-z ']+$**	
Lunghezza [LC]	1. < 1 or > 32 [error] 2. > 0 and <= 32 [property LC_OK]
Formato [FC]	1. non rispetta i vincoli [if LC_OK] [error] 2. rispetta i vincoli [if LC_OK] [property FC_OK]
Parametro: Email **Formato: ^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$**	
Lunghezza [LE]	1. < 1 or > 32 [error] 2. > 0 and <= 32 [property LE_OK]
Formato [FE]	1. non rispetta i vincoli [if LE_OK] [error] 2. rispetta i vincoli [if LE_OK] [property FE_OK]
Esiste [EE]	1. e-mail esistente nel DB [if LE_OK] [if FE_OK] [error] 2. e-mail non esistente nel DB [if LE_OK] [if FE_OK] [property EE_OK]

Parametro: Password **Formato: /^[A-Za-z0-9.] {3,16} $/;**	
Lunghezza [LP]	1. < 1 or > 16 [error] 2. > 0 and <= 16 [property LP_OK]
Formato [FP]	1. non rispetta i vincoli [if LP_OK] [error] 2. rispetta i vincoli [if LP_OK] [property FP_OK]

Codice	Test Frame	Esito
TC_4.1_1	LN1	Errore: lunghezza non valida
TC_4.1_2	LN2, FN1	Errore: formato non valido
TC_4.1_3	LN2, FN2, LC1	Errore: lunghezza non valida
TC_4.1_4	LN2, FN2, LC2, FC1	Errore: formato non valido
TC_4.1_5	LN2, FN2, LC2, FC2, LE1	Errore: lunghezza non valida
TC_4.1_6	LN2, FN2, LC2, FC2, LE2, FE1	Errore: formato non valido
TC_4.1_7	LN2, FN2, LC2, FC2, LE2, FE2, EE1	Errore: e-mail già esistente
TC_4.1_8	LN2, FN2, LC2, FC2, LE2, FE2, EE2, LP1	Errore: password non valida
TC_4.1_9	LN2, FN2, LC2, FC2, LE2, FE2, EE2, LP2, FP1	Errore: formato non valido
TC_4.1_10	LN2, FN2, LC2, FC2, LE2, FE2, EE2, LP2, FP2	Corretto: candidato registrato

9.4.2 Login

Parametro: Email	
Lunghezza [LE]	1. =0 [error] 2. >0 [property LE_OK]
Match [ME]	1. e-mail non esistente [if LE_OK] [error] 2. e-mail esistente [if LE_OK] [property ME_OK]
Parametro: Password	
Lunghezza [LP]	1. =0 [error] 2. >0 [property LP_OK]
Match [MP]	1. password utente non corretta [if LP_OK] [error] 2. password utente corretta [if LP_OK] [property MP_OK]

Codice	Test Frame	Esito
TC_4.2_1	LE1	Errato: mail non inserita
TC_4.2_2	LE2, ME1	Errato: mail non esistente
TC_4.2_3	LE2, ME2, LP1	Errato: password non inserita
TC_4.2_4	LE2, ME2, LP2, MP1	Errato: password non corretta
TC_4.2_5	LE2, ME2, LP2, MP2	Corretto: login effettuato

10. Testing schedule

Per lo schedule del testing si rimanda ai documenti di progetto.