

# Statement of Work

## Progetto: Turing Careers

Gruppo NC-25

### Membri del Gruppo:

- Gaudino Claudio
- Lorenzo Antonino
- Pagnotta Antonio
- Passariello Jacopo

### Revision History:

Data	Versione	Descrizione	Autori
11/10/23	0.1	Prima Stesura	Claudio Gaudino, Antonino Lorenzo.
11/10/23	0.2	Correzione di Forma	Jacopo Passariello.
15/10/23	0.3	Rimozione placeholders	Jacopo Passariello.

## Problema

Attualmente il mercato dello sviluppo software è tra quelli più in crescita e di conseguenza anche la richiesta dei professionisti. Nonostante ciò, il processo di assunzione sia per le aziende che per gli sviluppatori presenta molteplici sfide.

In particolare:

- Per gli sviluppatori, sia freelancer 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 destreggiarsi tra piattaforme aziendali eterogenee.
- Per le aziende il processo di recruitment può essere dispendioso, sia in termini di tempo che manodopera: il processo di recruitment richiede spesso molte interviste a molti individui. Affidarsi ad una piattaforma online invece risulta in un compromesso tra sforzo e qualità.

## Obiettivo

L'obiettivo di *Turing Careers* è quello di realizzare una piattaforma web per facilitare il processo di assunzione di sviluppatori software da parte delle aziende e il processo di ricerca di lavoro da parte degli sviluppatori.

## Product Scope

Per permettere una comunicazione efficiente e bidirezionale, *Turing Careers* deve supportare:

- La possibilità per aziende di pubblicare offerte di lavoro per un certo target di sviluppatori.
- La possibilità per gli sviluppatori di realizzare un profilo.
- La ricerca di sviluppatori software da parte delle aziende e viceversa.
- Un sistema di raccomandazione che faciliti la ricerca.
- La possibilità per un'azienda di comunicare tramite messaggi con sviluppatori.

## Data di Inizio e Fine

Inizio: ottobre 2023

Fine: tra metà gennaio 2024 e inizio febbraio 2024, (corrispondente a tra il Pre Appello e il Primo Appello).

## Deliverables

- RAD, SDD, ODD, Matrice di Tracciabilità, Test Plan, Test Case Specification, Test Incident Report, Manuale d'Uso, Manuale d'Installazione e ogni altro documento richiesto per lo sviluppo del sistema.

## Vincoli/Constraints

### *Vincoli collaborativi e comunicativi*

- Rispetto delle scadenze intermedie e di fine progetto sopra citate.
- Budget orario non superiore a cinquanta ore per membro del gruppo, per un totale di duecento ore.
- Uso di sistemi di versioning dove tutti i membri del gruppo forniscano il loro contributo – GitHub in particolare.
- Utilizzo di tool per la suddivisione di task e attività – Trello o similare.
- Utilizzo di tool per la comunicazione tracciabile – Slack.

### Vincoli Tecnici:

#### *Analisi e specifica dei requisiti*

- Specifica di **minimo due** e **massimo quattro** scenari per ogni membro del gruppo.
- Specifica di **minimo due** e **massimo quattro** requisiti funzionali e non funzionali per ogni membro del gruppo.
- **Esattamente** un *Caso d'Uso* per ogni membro del gruppo – i casi d'uso aggiuntivi non saranno valutati.
- **Esattamente** un *Sequence Diagram* ogni due membri del gruppo – i *Sequence Diagram* aggiuntivi **non** saranno valutati.
- **Esattamente** un diagramma a scelta tra *State Chart* e *Activity Diagram* ogni due membri del gruppo – ulteriori diagrammi **non** verranno valutati.

## System Design

- Specifica di **minimo due** e **massimo quattro** design goal per ogni membro del gruppo.
- Definizione di un ***decomposition diagram*** dei sottosistemi per gruppo, con annessa descrizione e motivazione d'uso.

Definizione di un ***deployment diagram*** per gruppo, con annessa descrizione e motivazione all'uso.

## Object Design

- Uso di **minimo uno** e **massimo due** *design pattern* per gruppo (devono essere selezionati tra quelli presentati a lezione).
- Uso di **UML**.

## Testing

- Ogni studente dovrà effettuare il testing di un unità, tramite category partition, di **esattamente** un metodo di una classe sviluppata.
- Ogni studente dovrà effettuare il testing di sistema, tramite category partition, di **esattamente** una funzionalità del sistema sviluppato.

## Criteri di Accettazione

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Utilizzo adeguato del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Utilizzo adeguato di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Utilizzo adeguato di Slack, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli**.

- Appropriato test di sistema di una funzionalità del sistema sviluppato, che preveda il **rispetto dei vincoli**.

### **Criteri di Premialità**

- Uso adeguato di sistemi di **build**;
- Uso adeguato di un processo di **Continuous Integration** tramite Travis;
- Uso adeguato di tool di controllo della qualità (ad esempio, **CheckStyle**);
- Adozione di processi di **code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).