



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

Statement of Work

Progetto

Foundly

Riferimento

Versione 0.1

Data 05/10/2025

Destinatario Studenti di Ingegneria del Software 2025/26

Presentato da

Approvato da



Revision History

Data	Versione	Descrizione	Autori
05/10/2025	0.1	Prima stesura	S. Lepore S. D. Passarelli N. Nappi



Statement of Work (SOW) del Progetto

Foundly

1. Scopo del Sistema

Lo scopo del sistema **Foundly** è progettare e realizzare una **webapp** per la gestione e il recupero di oggetti e animali smarriti, favorendo la collaborazione tra utenti e la creazione di una community basata sulla fiducia e sulla responsabilità condivisa.

Deve supportare:

- la **segnalazione di ritrovamenti** di oggetti o animali da parte degli utenti, con la possibilità di allegare foto, descrizioni e luogo del ritrovamento;
- la **ricerca da parte di chi ha perso un oggetto o un animale** tra i post pubblicati, con la possibilità di inoltrare un reclamo (secure claim) per richiederne la restituzione in modo sicuro e verificato;
- la **gestione dei profili utente e di una scoreboard** che premia chi ha aiutato a ritrovare più oggetti o animali, valorizzando l'impegno e la collaborazione all'interno della community;
- **permettere ai punti vendita di registrarsi come drop-point**, ovvero luoghi sicuri dove depositare o ritirare gli oggetti smarriti, facilitando così lo scambio anche senza incontro diretto tra gli utenti. In questo modo, i negozi partecipanti aumentano la propria visibilità e contribuiscono attivamente alla community.

In questo modo, Foundly si propone come una piattaforma digitale di supporto alle attività di recupero di oggetti e animali smarriti, favorendo comportamenti collaborativi e valorizzando il senso di fiducia e solidarietà tra le persone.

2. Data di Inizio e di Fine

Inizio: Ottobre 2025

Fine: fine Gennaio 2026



3. Deliverables

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

4. Vincoli/Constraints (adattare considerando quanto indicato nella lezione di introduzione al corso)

Vincoli collaborativi e comunicativi.

- Rispetto delle scadenze intermedie/di fine progetto **definite nello statement of work**
- Uso di sistemi di versioning - GitHub in particolare
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo
- Utilizzo di tool per la suddivisione dei task e attività (Trello)
- Utilizzo di tool di comunicazione tracciabile (Discord)

Vincoli tecnici

Analisi e specifica dei requisiti

- Specifica di **minimo 2** e **massimo 4** scenari per ogni membro del team;
- Specifica di **minimo 2** e **massimo 4** requisiti funzionali e non funzionali per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Esattamente** un diagramma a scelta tra statechart e activity diagram ogni due membri del team - ulteriori diagrammi **non** verranno valutati;
- Specifica di un class diagram per team - eventuali object diagram **non** verranno valutati.

System Design

- Specifica di **minimo 2** e **massimo 4** design goal per ogni membro del team.
- Definizione di **un diagramma** di decomposizione dei sottosistemi per team, con annessa descrizione e motivazione all'uso.



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

Object Design

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

Testing

- Ogni studente dovrà effettuare il testing di 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.

5. Criteri di Accettazione/Acceptance Criteria -

Criteri che, se non rispettati, portano al fallimento del progetto (adattare considerando quanto indicato nella lezione di introduzione al corso)

-
- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
 - Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
 - Adeguato utilizzo di Discord, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
 - Adeguato utilizzo di Trello, 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**.



6. Criteri di premialità (adattare considerando quanto indicato nella lezione di introduzione al corso)

- 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.).