



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

Riassunto Consegna Progetto WasteGone



Riferimento	C01_RCP
Versione	1.0.0
Data	16/12/2024
Destinatario	Studenti di Ingegneria del Software 2024/25
Presentato da	C01 - Alessia Gatto (A.G.), Elisa Picilli (E.P.), Francesco Laudano (F.L.), Giovanni Croce (G.C.), Marco Iannuzzi (M.I.), Michela Palmieri (M.P.), Simon Carbone (S.C.)
Approvato da	



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

Revision History

Data	Versione	Descrizione	Autori
16/12/2024	1.0.0	Prima Stesura e consegna	D.F.P e F.D.



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

Indice

Revision History	2
1. Main Informations	4
2. Project Description	5
3. Link	6
4. Constraints	7
5. Acceptance Criteria	9
6. Reward Criteria	11



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

Retrospective del Progetto WasteGone

1. Main Informations

Titolo del progetto: WasteGone

Team ID: C01

Nome del team: GIGA

Data d'inizio del progetto: 18/10/2024

Consegna prototipo del progetto: 13/12/2024

Data fine del progetto: 28/03/2025



2. Project Description

L'obiettivo del progetto è fornire una piattaforma IT di supporto alle attività di gestione dei rifiuti, assicurando che tutti gli stakeholder coinvolti possano agire in maniera agevole ed efficiente. Per i cittadini, verrà realizzato un modulo AI che consenta di riconoscere il tipo di rifiuto, con le informazioni principali su dove e quando conferirlo.

Deve supportare le seguenti caratteristiche di base, che descrivono il sistema attuale:

- Creazione di un profilo per i cittadini;
- Visualizzazione dell'informativa sulle categorie dei rifiuti;
- Visualizzazione degli orari per il conferimento dei rifiuti;
- Notifiche per ricordare ai cittadini di conferire i rifiuti casalinghi (alla porta);
- Visualizzazione sulla mappa dei vari punti di ritiro per il conferimento di rifiuti speciali/ingombranti;
- Segnalazione per presenza di discariche abusive o errato conferimento dei rifiuti.

Sono previste anche le seguenti caratteristiche innovative:

- Prenotazione di uno slot per il ritiro a casa di oggetti/indumenti che se in buono stato saranno destinati alle donazioni, altrimenti saranno destinati allo smaltimento;
- Sezione eventi, per pubblicizzare eventi sul territorio che sensibilizzano sulle tematiche dell'ambiente e del riutilizzo;
- Sezione per effettuare baratto di prodotti che non sono più utilizzati;
- Sezione per effettuare pagamenti delle tasse sui rifiuti;
- Realizzazione modulo AI per riconoscimento del tipo di rifiuto.



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

3. Link

Link al github: <https://github.com/Tensa53/WasteGone>



4. Constraints

Criterio	Rispettato	Note
Rispetto delle scadenze intermedie/di fine progetto definite dai project manager, per i progetti di tipo A, e definite nello statement of work, per i progetti di tipo B.	SI	//
Uso di sistemi di versioning - GitHub in particolare	SI	Link al repository nella sezione 3 di questo documento
Uso di tool per la gestione di task e attività - Trello o simili	SI	//
Uso di un tool di comunicazione tracciabile - Slack, MS Teams o Discord	SI	Abbiamo utilizzato Discord e Telegram
Specifiche di minimo 2 e massimo 4 scenari per ogni membro del team	SI	Sono stati prodotti 2 scenari per ogni Team Member
Esattamente uno use case per ogni membro del team - i casi d'uso aggiuntivi non saranno valutati	SI	//
Specifiche degli oggetti boundary, control e entity per gli use case specificati	SI	Derivati dagli artefatti prodotti nell'analisi dei requisiti
Esattamente un sequence	SI	Sono stati prodotti 4



diagram ogni due membri del team - i sequence diagram aggiuntivi non saranno valutati;		sequence diagrams
Almeno un activity diagram per sistema esistente o sistema proposto e almeno uno statechart diagram ogni due membri del team. La somma degli activity e statechart diagram deve essere uguale al numero di membri del team diviso due.	SI	Sono stati prodotti un activity diagram del sistema proposto e 3 statechart diagram
Specificazione dei design goal e analisi dei trade-off relativi ad almeno due coppie di design goal	SI	
Definizione dell'architettura del sistema	SI	
Individuazione di dover poter utilizzare due design pattern, indicandone l'obiettivo e come sarebbero implementati.	SI	
Ogni studente dovrà effettuare il testing di unità, tramite category partition, dei casi d'uso specificati	SI	



5. Acceptance Criteria

Criterio	Rispettato	Note
Utilizzo appropriato di GitHub, che preveda il rispetto delle linee guida definite nel contesto del primo lab. Chi non lo fa, non sarà valutato al progetto	SI	//
Adeguate utilizzo del pull-based development, che preveda il rispetto delle linee guida definite nel contesto del primo lab	SI	Sono stati sviluppati branch separati, per ogni sottosistema, che, dopo un'attenta revisione da parte dei PM, sono stati "mergiati" nel master.
Adeguate utilizzo di Slack o simili, che preveda il rispetto delle linee guida definite nel contesto del secondo lab	SI	Abbiamo utilizzato Discord e Telegram
Adeguate utilizzo di Trello, che preveda il rispetto delle linee guida definite nel contesto del secondo lab	SI	
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	SI	
Utilizzo di tool di controllo della qualità per l'implementazione (CheckStyle)	SI	Ogni Team Member ha implementato il plugin di CheckStyle sulla propria macchina e risolto la maggior parte dei warning



Laurea Triennale in Informatica - Università di Salerno

Corso di Ingegneria del Software - Proff. F. Ferrucci, F. Palomba

Appropriato test di unità di un metodo sviluppato, che preveda il rispetto dei vincoli	SI	
--	----	--



6. Reward Criteria

Criterio	Rispettato	Note
Uso adeguato di sistemi di build	SI	Abbiamo utilizzato Gradle, che è ciò su cui si basa una app Native Android
Uso adeguato di un processo di continuous integration tramite Travis	NO	
Adozione di processi di code review	SI	
Uso adeguato di tool avanzati di testing (e.g., Mockito, Cobertura, etc.)	SI	Abbiamo utilizzato Mockito, per effettuare alcuni test di unità