



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Schedule Management Plan Progetto WasteGone



Riferimento	C01_SMP
Versione	1.0
Data	17/12/2024
Destinatario	Docenti di Gestione dei Progetti Software 2024/25
Presentato da	C01 - D'Antuono Francesco Paolo, Fabiano Daniele
Approvato da	



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Revision History

Data	Versione	Descrizione	Autori
23/10/2024	0.1	Prima Stesura	D.F.P. e F.D.
13/12/2024	0.2	Aggiunta contenuti all'interno delle sezioni	D.F.P e F.D.
17/12/2024	1.0	Revisione finale e consegna	D.F.P. e F.D.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Indice

Revision History	2
1. Task List	4
2. Network Diagram	7
3. GANTT Diagram	8



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Time Management Plan del Progetto

WasteGone

1. Task List

WasteGone

13 dic 2024

GIGA

Responsabile di progetto

Date di inizio e fine progetto

18 ott 2024 - 14 dic 2024

Completamento

100%

Attività

49

Risorse

9

Risorse

Nome

Ruolo predefinito

Francesco Paolo D'Antuono

Project Manager

Daniele Fabiano

Project Manager

Michela Palmieri

Team Member

Alessia Gatto

Team Member

Francesco Laudano

Team Member

Giovanni Croce

Team Member

Elisa Picilli

Team Member

Marco Iannuzzi

Team Member

Simon Carbone

Team Member



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Attività

Nome	Data d'inizio	Data di fine
WasteGone Start	18/10/24	18/10/24
RAD	18/10/24	06/11/24
Start Activities	18/10/24	18/10/24
Raccolta dei Requisiti	18/10/24	25/10/24
Introduzione	18/10/24	22/10/24
Sistema Attuale	18/10/24	22/10/24
Sistema Proposto	18/10/24	22/10/24
<i>Fino alla sezione 3.3 "Requisiti non funzionali" del template</i>		
Scenari	23/10/24	24/10/24
<i>Rispettare i vincoli, 2 scenari a testa.</i>		
Casi D'Uso	25/10/24	25/10/24
<i>Rispettare i vincoli, 1 caso d'uso a testa.</i>		
Analisi dei Requisiti	18/10/24	06/11/24
Mock-up and Navigational Paths	25/10/24	06/11/24
Glossario	18/10/24	06/11/24
Object Model	28/10/24	04/11/24
Dynamic Model	05/11/24	06/11/24
<i>Rispettare i vincoli, 1 sequence diagram ogni due membri del team, 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.</i>		
SDD	08/11/24	14/11/24
Introduzione	08/11/24	12/11/24
<i>Rispettare i vincoli, Specifica dei design goal e analisi dei trade-off relativi ad almeno due coppie di design goal.</i>		
Architettura Proposta	08/11/24	14/11/24
Panoramica	08/11/24	12/11/24
Decomposizione in Sottosistemi	08/11/24	12/11/24
<i>Rispettare i vincoli, Definizione dell'architettura del sistema.</i>		
Nome	Data d'inizio	Data di fine
Hardware/Software Mapping	13/11/24	14/11/24
Gestione dei Dati Persistenti	08/11/24	14/11/24
ODD	15/11/24	21/11/24
Coding Conventions and Guidelines	15/11/24	21/11/24
Design Pattern	15/11/24	21/11/24
<i>Rispettare i vincoli, Individuazione di due design pattern, indicandone l'obiettivo e come sarebbero implementati.</i>		
Packages	15/11/24	21/11/24
Class Interfaces	15/11/24	21/11/24
<i>In questa parte bisogna solo definire quali saranno i tool che verranno utilizzati per la generazione delle interfacce (es., JavaDoc).</i>		
Implementazione	22/11/24	12/12/24
Configurazione Ambiente di Sviluppo e Repository	22/11/24	25/11/24
<i>Configurazione iniziale delle componenti utili per il progetto e la repository.</i>		
Suddivisione dei Packages	22/11/24	25/11/24
<i>Suddivisione iniziale dei pacchetti di sistema. Rivedere la suddivisione in sottosistemi nel System e Object Design.</i>		
Generazione JavaDoc	26/11/24	12/12/24
<i>Tutti i partecipanti al progetto dovranno implementare i commenti per la JavaDoc, questo task specifica solo la generazione di essa, con la relativa scrittura del riferimento nella sezione "Class Interfaces" dell'ODD. I responsabili di questo task dovranno assicurarsi che la javadoc sia esportabile e che tutto il codice sia stato commentato.</i>		
Gestione Profilo Utente	26/11/24	02/12/24
GPU Persistent Layer	26/11/24	02/12/24
<i>Devono essere implementati solamente Registrazione, Login e Notificazione Utente (per consentire la personalizzazione dell'orario). Per questo layer bisogna preparare il db Firestore e salvare le entità all'interno del sistema, che dovranno essere poi legate all'application, tramite la vostra implementazione dello storage.</i>		
GPU Business Layer	26/11/24	02/12/24
<i>Devono essere implementati solamente Registrazione, Login e Notificazione Utente (per consentire la personalizzazione dell'orario). Per questo layer si dovrà implementare tutta la logica di business e completare le transazioni che sono preparate nel persistent layer, in modo da legare storage e application.</i>		

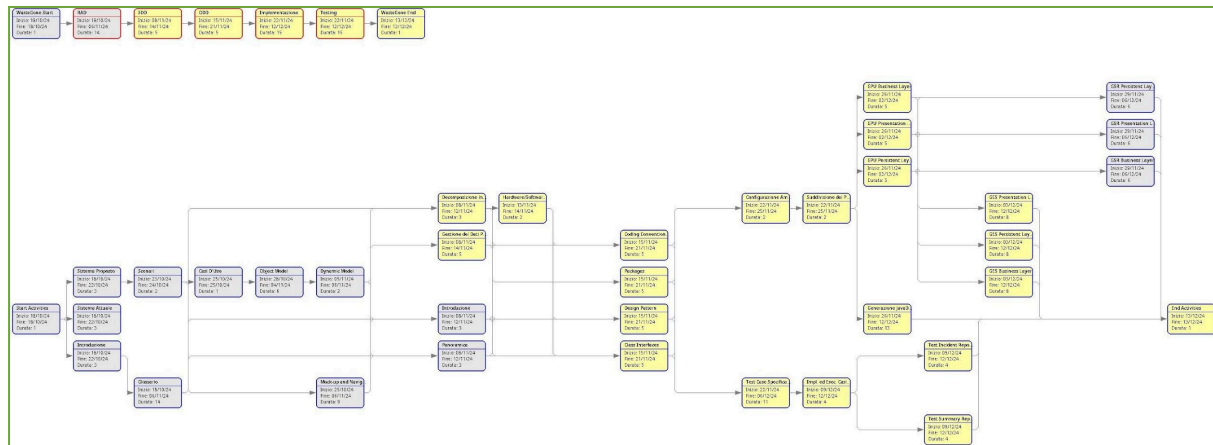


Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Nome	Data d'inizio	Data di fine
GPU Presentation Layer <i>Devono essere implementati solamente Registrazione, Login e Notificazione Utente (per consentire la personalizzazione dell'orario). Per questo layer si dovranno implementare le interfacce utente e preparare la classe Activity per essere completata da chi si occupa del business layer.</i>	26/11/24	02/12/24
Gestione Smaltimento Rifiuti	29/11/24	06/12/24
GSR Persistent Layer <i>Devono essere implementati solamente Visualizzazione Dettagli Rifiuti e Personalizzazione Notifiche Rifiuti. Per questo layer bisogna preparare il db Firestore e salvare le entità all'interno del sistema, che dovranno essere poi legate all'applicatory, tramite la vostra implementazione dello storage.</i>	29/11/24	06/12/24
GSR Business Layer <i>Devono essere implementati solamente Visualizzazione Dettagli Rifiuti e Personalizzazione Notifiche Rifiuti. Per questo layer si dovrà implementare tutta la logica di business e completare le transazioni che sono preparate nel persistent layer, in modo da legare storage e application.</i>	29/11/24	06/12/24
GSR Presentation Layer <i>Devono essere implementati solamente Visualizzazione Dettagli Rifiuti e Personalizzazione Notifiche Rifiuti. Per questo layer si dovranno implementare le interfacce utente e preparare la classe Activity per essere completata da chi si occupa del business layer.</i>	29/11/24	06/12/24
Gestione Eventi e Sensibilizzazione	03/12/24	12/12/24
GES Persistent Layer <i>Devono essere implementati solamente Sezione Eventi, Prenotazione Ritiro a Casa e Visualizzazione Tasse. Per questo layer bisogna preparare il db Firestore e salvare le entità all'interno del sistema, che dovranno essere poi legate all'applicatory, tramite la vostra implementazione dello storage.</i>	03/12/24	12/12/24
GES Business Layer <i>Devono essere implementati solamente Sezione Eventi, Prenotazione Ritiro a Casa e Visualizzazione Tasse. Per questo layer si dovrà implementare tutta la logica di business e completare le transazioni che sono preparate nel persistent layer, in modo da legare storage e application.</i>	03/12/24	12/12/24
GES Presentation Layer <i>Devono essere implementati solamente Sezione Eventi, Prenotazione Ritiro a Casa e Visualizzazione Tasse. Per questo layer si dovranno implementare le interfacce utente e preparare la classe Activity per essere completata da chi si occupa del business layer.</i>	03/12/24	12/12/24
Testing	22/11/24	12/12/24
Test Case Specification <i>Specifica dei casi di test per i casi d'uso che sono stati esplicitati all'interno del RAD. Ogni persona dovrà prendersi carico del proprio caso d'uso e fornire i casi di test.</i>	22/11/24	06/12/24
Nome	Data d'inizio	Data di fine
Impl. ed Esec. Casi di Test <i>Implementazione dei casi di test sviluppati ed esecuzione per rilevare failure.</i>	09/12/24	12/12/24
Test Incident Report <i>In questo documento dovrà essere esposta il risultato dell'esecuzione dei casi di test.</i>	09/12/24	12/12/24
Test Summary Report <i>In questo documento, si mettono sotto indagine le eventuali failure documentate nel TIR e si documenterà l'eventuale risoluzione del problema.</i>	09/12/24	12/12/24
End Activities	13/12/24	13/12/24
WasteGone End	13/12/24	13/12/24

2. Network Diagram



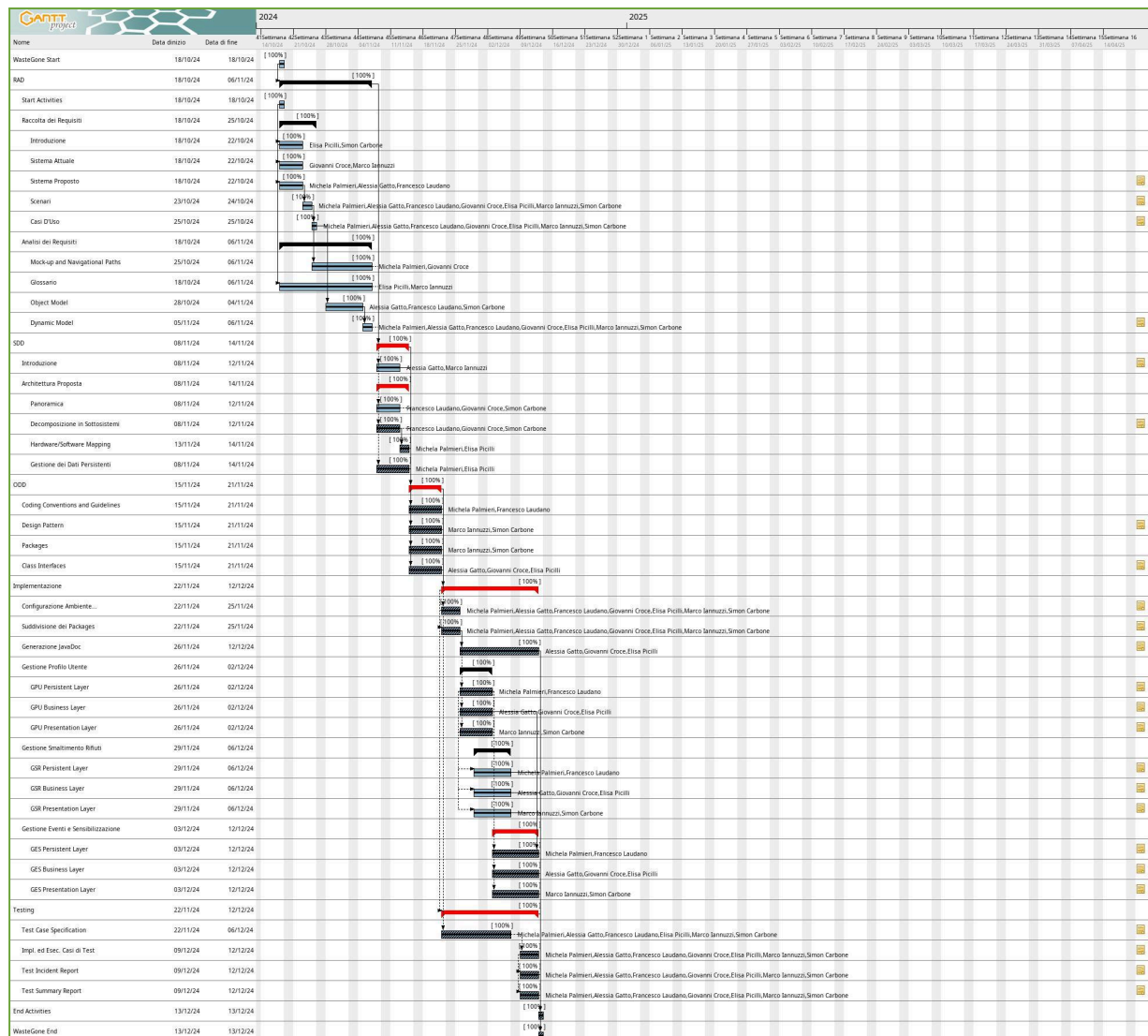
Il diagramma che viene mostrato sopra, rappresenta il cosiddetto *Activity-in-the-node diagram*, che consente di mostrare l'attività svolta in un certo periodo, con la sua effettiva durata. Questo diagramma è utile per calcolare il **critical path** (evidenziato in giallo) del nostro progetto. Bisogna monitorare costantemente le attività che fanno parte del critical path, allocando le risorse in maniera efficace, assicurando che ogni fase sia completata entro i tempi previsti.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

3. GANTT Diagram



Il diagramma che viene mostrato sopra, rappresenta la suddivisione temporale delle attività del progetto. Nel dettaglio, si può notare che tutte le attività sono state completate nei tempi schedulati e le barre in rosso rappresentano il nostro critical path. Inoltre, sono rappresentate anche le risorse allocate su ogni attività. In merito alla fase di testing, i PM si sono occupati della pianificazione delle attività, realizzando l'apposito documento di Test Plan, che è stato condiviso ai Team Member prima dell'inizio della stesura dei casi di test. Sulla base di queste motivazioni, nelle attività schedulate non è presente alcun task relativo alla stesura del documento di Test Plan.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Il GANTT verrà aggiornato successivamente, quando verranno avviate le ulteriori attività, per il completamento del progetto, dopo il rilascio del primo prototipo. Di solito, per le attività non è stata data più di una settimana di tempo, evitando di aggiungere tempo in più di quello che secondo noi era necessario (*fudge factors*), così da evitare le conseguenze della legge di Parkinson. Le stime dei tempi, sono state fatte in base a due componenti fondamentali:

- **Assunzioni:** Sono alla base di tutte le stime. Ciò vuol dire che le stime possono cambiare in maniera frequente, se le assunzioni fatte non sono solide. Nel nostro caso, le assunzioni fatte sono:
 - **Tempi ristretti:** In alcuni casi era complicato trovare punti in comune per i meeting e per lo svolgimento delle attività, poiché i Team Member, sono impegnati anche con altri progetti e attività varie. Ci sono stati casi specifici in cui, abbiamo fornito un po' più di tempo del solito, per dare la possibilità ai ragazzi di conciliare le diverse attività progettuali sulle quali sono coinvolti;
 - **Ritardo controllato sulle attività:** In alcuni casi, le attività sono state programmate per essere concluse il Venerdì, concedendo la possibilità ai Team Member di rivedere e sistemare gli artefatti da dover produrre, durante i giorni del weekend. In questo modo, se c'era ancora qualcosa da completare, sarebbe stato pronto per l'inizio della nuova settimana lavorativa. Ciò è dovuto dai tanti impegni settimanali in cui sono coinvolti i Team Member, essendo, prima di tutto, delle figure junior che sono in continua formazione.
- **Dipendenze tra attività:** In alcuni casi, le attività avevano una dipendenza non solo temporale, ma anche di realizzazione dell'artefatto (per esempio, non si può fare l'hardware/software mapping, senza aver prima decomposto il sistema in sottosistemi). Ciò ha portato a forzare la schedulazione di alcune attività, in maniera sequenziale.