Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci

GUARDIAN FLOW
FEELING SAFE

Report di conformità
Guardian Flow

Riferimento	
Versione	0.1
Data	16/01/2024
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Raffaele Mezza, Martina Mingione
Approvato da	

Sommario

Revision History

Data	Versione	Descrizione	Autori
16/01/2024	0.1	Prima stesura	Raffaele Mezza, Martina Mingione

Report di conformità

Guardian Flow

1. Scopo del documento

Lo scopo di questo documento è fornire un resoconto dettagliato e conclusivo del progetto, basandosi sull'analisi e la verifica dell'adempimento dei vincoli collaborativi e tecnici, dei criteri di accettazione e di premialità stabiliti nella fase inziale del progetto. Inoltre, il documento contiene una sezione dedicata alle metriche del software, Linee di Codice (LOC), numero di design pattern implementati e altri indicatori rilevanti. L'obiettivo è fornire un resoconto completo dei vincoli prefissati e dei criteri in accordo con quanto dichiarato nello Statement of Work.

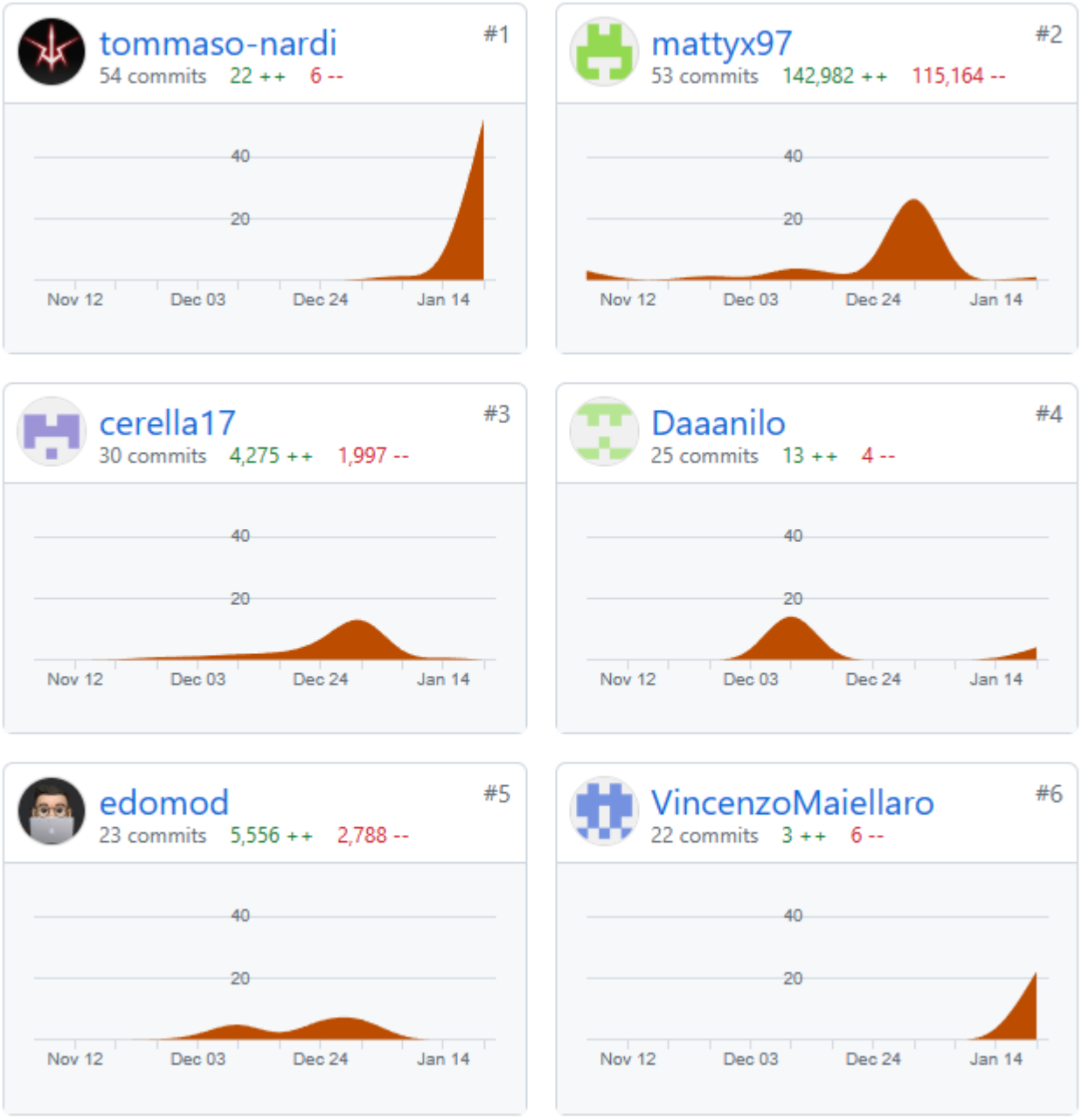
1. Metriche di fine progetto

Di seguito sono riportate le principali metriche del software per il progetto GuardianFlow, calcolate alla data dello status report finale.

Metriche del software:

Nome metrica	Valore di Accettazione	Valore Finale
Lines of Code (LOC)	N/A	3812
Design Pattern	N/A	1
Test Case	21	21
Test Suite	6	6
Branch Coverage dei casi di test	Maggiore del 75%	90%
Warning identificati di …	N/A	Identificati e risolti: 12
		Identificati e non risolti: 3
Commit GitHub	N/A	Totali: 231
		Edmondo De Simone: 23
		Giuseppe Cerella: 30
		Danilo Gisolfi: 25
		Mattia Guariglia: 58
		Vincenzo Maiellaro: 22
		Tommaso Nardi: 54

Segue una rappresentazione grafica delle commit effettuate su GitHub:



Per lo sviluppo del codice, è stato impiegata l'estensione Live Share di Visual Studio Code. Questo strumento si è rivelato particolarmente utile poiché ha permesso al team di lavorare simultaneamente sugli stessi file, evitando conflitti e agevolando la collaborazione. È importante sottolineare che questa scelta è stata motivata dal fatto che alcuni membri del team non avevano esperienza nell'approccio a questo tipo di sviluppo. Pertanto, l'utilizzo di Live Share è risultato fondamentale per facilitare l'integrazione di tutti i componenti del gruppo nello sviluppo, utilizzando tecniche precedentemente inedite per alcuni di loro. Questo spiega il motivo per cui le commit su GitHub sono state prevalentemente effettuate da coloro che hanno condiviso attivamente il codice durante le sessioni Live Share.

2. Vincoli

2.1 Vincoli collaborativi e comunicativi

Di seguito sono riportati i vincoli collaborativi di progetto.

Vincolo	Rispettato?	Come è stato rispettato?
Rispetto scadenze (di fine e quelle intermedie definite dal management)	Si	Le scadenze sono state rispettate seguendo il planning
Budget/Effort non superiore a 50	Si	Sono state rispettate attraverso una attenta pianificazione dei task
Utilizzo di tool di management Trello	Si	La programmazione dei task del progetto è avvenuta su Trello
Utilizzo di Slack per le comunicazioni ufficiali	In parte	Slack è stato utilizzato nella prima fase ma poco sfruttato nella seconda fase
Utilizzo di GitHub come sistema di versioning	Si	Il team ha utilizzato GitHub come unico strumento di versioning per la realizzazione dell'applicativo
Parte di progetto con approccio Agile (Scrum)	Si	Scrum è stato implementato nella seconda fase del progetto.

2.2 Vincoli tecnici

Di seguito sono riportati i vincoli tecnici di progetto.

Vincolo	Valore di Accettazione	Valore Finale	Verifica
Numero di scenari	minimo due e massimo quattro per ogni membro	12	Attraverso il documento RAD
Requisiti funzionali e non funzionali	minimo due e massimo quattro per ogni membro	24	Attraverso il documento RAD
Use case	uno per ogni membro	6	Attraverso il documento RAD
Sequence diagram	tre per team	3	Attraverso il documento RAD
Activity Diagram	tre per team	3	Attraverso il documento RAD
Class Diagram	uno per team	1	Attraverso il documento RAD
Design Goal	minimo 2 e massimo 4 per ogni membro	13	Attraverso il documento SDD
Diagramma di decomposizione dei sottosistemi	uno per team	1	Attraverso il documento SDD
Deployment diagram	uno per team	1	Attraverso il documento SDD
Design pattern	minimo uno e massimo due per team	1	Attraverso il documento ODD
Uso di UML	N/A	Si	Attraverso il documento RAD
Testing di unità	un metodo per ogni classe sviluppata		Attraverso il documento UTR
Testing di sistema	una funzionalità del sistema sviluppato	6	Attraverso il documento STER

3. Criteri di accettazione

Di seguito sono riportati i criteri di accettazione del progetto:

Criterio	Valore di Accettazione	Valore Finale	Verifica
Rispetto dei vincoli nello sviluppare le funzionalità del sistema	N/A	SI	Verificabile attraverso l'utilizzo del sistema.
Branch coverage dei casi di test	75%	90%	Verificabile attraverso la visione del documento UTR.
Buona Manutenibilità	N/A	SI	Verificabile attraverso la lettura del codice.

4. Criteri di premialità

Di seguito sono riportati i criteri di premialità del progetto:

Criterio	Valore di Accettazione	Valore Finale	Verifica
Utilizzo di un processo di Continuous Integration, tramite l'utilizzo di GitHub.	N/A	SI	Verificabile andando a controllare la repository pubblica su GitHub.
Adozione di processi di code review	N/A	No	-