***Statement of Work Progetto*** *NetGun*

|  |  |
| --- | --- |
| Riferimento |  |
| destinatario | Università degli Studi Di Salerno |
| Versione | 0.1 |
| data | 11/10/2022 |
| presentato da | Carlo Colizzi, Giulio Incoronato, Antonio Mazzarella |
| approvato da |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autori |
| 11/10/2022 | 0.1 | Prima stesura | Carlo Colizzi, Giulio Incoronato, Antonio Mazzarella |

**Vincoli/Constraints**

***Vincoli collaborativi e comunicativi.***

* Rispetto scadenze delle scadenze intermedie/di fine progetto **definite nello statement of work**
* Budget/Effort non superiore a 50\*n ore dove n sono i membri del team
* 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 di per la suddivisione dei task e attività (Trello o similare)
* Utilizzo di tool di comunicazione tracciabile (Slack)

***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.

Criteri di Accettazione/Acceptance Criteria

* 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 Slack, 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**.

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.).