|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Statement of Work  Progetto RojinaReview   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.2 | | Data | 12/10/2022 | | Destinatario | Carmine Gravino | | Presentato da | Carmine Iemmino, Andrea Vitolo, Carmine Pio Nardo | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 09/10/2022 | 0.1 | Prima stesura | Carmine Iemmino, Andrea Vitolo, Carmine Pio Nardo |
| 12/10/2022 | 0.2 | Seconda stesura | Carmine Iemmino, Andrea Vitolo, Carmine Pio Nardo |

Statement of Work (SOW) del Progetto  
Rojina Review

1. Piano Strategico/Strategic Plan

Il cliente intende fornire ai videogiocatori un luogo in cui poter leggere, commentare notizie e recensioni riguardanti i propri videogiochi preferiti e acquistare merchandising relativo ad essi; e inoltre di poter attribuire un proprio voto personale ad un videogioco.

1. Obiettivi di Business/Business Needs

Attualmente sono presenti numerosi sistemi software di questo tipo, ma si intende affiancare alla parte giornalistica, un’ulteriore parte rivolta al merchandising di videogiochi.

Per raggiungere gli obiettivi si intende creare un sistema software in cui ospitare le notizie e recensioni che possono aiutare l’utente videogiocatore nella decisione di quali videogiochi comprare, confrontandosi eventualmente con altri utenti nella sezione commenti e soddisfare il collezionismo attraverso l’acquisto di gadget e indumenti appartenenti al mondo videoludico.

1. Ambito del Prodotto/Product Scope

L’obiettivo del progetto è fornire una rivista di videogiochi. Deve supportare le seguenti macro-funzionalità:

* Consultazione di recensioni, notizie e prodotti presenti sulla piattaforma
* Poter filtrare recensioni, notizie e prodotti in base a categorie
* Esprimere la propria opinione tramite commenti
* Valutare un videogioco con un voto numerico
* Acquisto di più prodotti tramite un carrello
* Permettere ad un utente di registrarsi
* Consultazione di un’area utente con le proprie informazioni, i voti dati, i commenti effettuati e lo storico ordini
* Permettere ad un giornalista di inserire recensioni e notizie sul mondo videoludico
* Permettere ad un giornalista di aggiungere un gioco nel sistema
* Permettere ad un amministratore di gestire la parte e-commerce, con inserimento, modifica e cancellazione prodotti

Scenario 1:

1. Data di Inizio e di Fine

Inizio: Ottobre 2022

Fine: Gennaio-Febbraio 2023. E’ possibile concordare la data di consegna che potrà essere una delle seguenti:

* I: circa metà Gennaio 2023
* II: fine Gennaio 2023
* III: prima decade di Febbraio 2023

1. 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.

1. 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\*3 ore dove 3 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.

1. Criteri di Accettazione/Acceptance Criteria (Criteri che, se non rispettati, portano al fallimento del progetto)

* 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**.

1. 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.).