Laurea Triennale in informatica-Università di Salerno

Corso di *Ingegneria Del Software*- Prof. C. Gravino

Statement of Work

# Progetto

## Ticket4U

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 0.1 |
| Data | 10/10/2023 |
| Destinatario | Studenti di Ingegneria del Software 2023/24 |
| Presentato da | Nicolò Sorà, Luca Greco, Simona Vigorito, Tresy  Sorrentino |
| Approvato da |  |

Laurea Triennale in informatica-Università di Salerno

Corso di *Ingegneria Del Software*- Prof. C. Gravino

Statement of Work (SOW) del Progetto

Ticket4U

### 1. Piano Strategico/Strategic Plan

Il Team di Ticket4U intende strutturare una piattaforma semplice e intuitiva per i propri clienti adibita ad organizzare la vendita a distanza di biglietti per eventi musicali, artistici, teatrali e culturali.

### 2. Obiettivi di Business/Business Needs

Il Team di Ticket4U intende semplificare e velocizzare l’acquisto di biglietti e di rendere immediata la possibilità di verificarne la disponibilità o in alternativa di prenotarli. In tal modo si potranno massimizzare le entrate invogliando i clienti a comprare tramite una piattaforma più organizzata e intuitiva.

### 3. Ambito del Prodotto/Product Scope

L’obiettivo del progetto è fornire uno strumento di vendita di biglietti online, assicurando agli utenti di poter accedere a tutte le informazioni di sorta sugli eventi e di acquistare titoli attraverso:

* Strumento di ricerca eventi per nome, zona, intervallo di tempo
* Catalogazione degli eventi per tipologia, zona e intervallo di costo
* Sistema di registrazione
* Calendario degli eventi personalizzato in base ai titoli acquistati

### 4. Data di Inizio e di Fine

Inizio: Ottobre 2023

Fine: Gennaio-Febbraio 2024. E’ possibile concordare la data di consegna che potrà essere una delle seguenti:

* I: circa metà Gennaio 2024
* II: fine Gennaio 2024
* III: prima decade di Febbraio 2024

### 5. 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.

### 6. 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.

### 7. 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**.

### 8. 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.).