



Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof. C. Gravino

Statement of Work

Hitchpicks

Riferimento	
Versione	1.0
Data	05/10/2024
Destinatario	Studenti di Ingegneria del Software 2024/25
Presentato da	Daniele De Martino, Christian Esposito, Federico De Rosa, Francesco Durante, Orazio Torre
Approvato da	



Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof. C. Gravino

Revision History

Data	Versione	Descrizione	Autori
05/10/2024	0.1	Prima stesura	Team
24/10/2024	0.2	Revisione	DDM
16/11/2024	0.3	Rimozione degli scenari (non necessari)	DDM
04/12/2024	0.4	Aggiornamento deliverables dopo ulteriori istruzioni	Team
07/01/2024	1.0	Revisione	Team



Statement of Work (SOW) del Progetto

Hitchpicks

1. Piano Strategico/Strategic Plan

La piattaforma Hitchpicks è un sistema software **web** progettato per consentire agli utenti di **tracciare**, **ricercare** e **organizzare film** e **serie TV** in **liste** personalizzate, al fine di monitorare i contenuti di cui l'utente fruisce. Oltre alla gestione individuale dei contenuti, la piattaforma integra una componente **sociale** che permette agli utenti di visualizzare le liste create da altri, condividere opinioni attraverso **recensioni**, promuovendo così un'interazione attiva tra gli utenti.

2. Ambito del Prodotto/Product Scope

L'obiettivo della piattaforma Hitchpicks è offrire agli utenti un sistema intuitivo per tracciare, ricercare e organizzare film e serie TV in liste personalizzate allo scopo di organizzazione il catalogo dei contenuti fruiti dall'utente. Deve supportare:

- La **registrazione** di un nuovo utente, con creazione di 5 liste predefinite (Watching, Planning, Completed, Paused, Dropped);
- La **creazione** di **liste** personalizzate;
- L'aggiunta di **contenuti** ad una lista, con priorità e stato;
- La visualizzazione di liste **predefinite** e **personalizzate** di altri utenti;
- La visualizzazione di determinate **statistiche** per ciascuna lista;
- La **ricerca** di **contenuti**, con **filtri**;
- La **ricerca** di **utenti**, con **filtri**;
- La visualizzazione di informazioni sui contenuti (**dati informativi**, **valutazioni**, **piattaforme** su cui guardare, ecc.);



- L'inserimento di **note personali** (annotazioni, valutazioni, ecc.) relativi a ciascun contenuto;
- L'inserimento e la **moderazione** di **recensioni pubbliche** relative a ciascun contenuto;
- La **gestione** degli **utenti**;
- La **gestione** dei **contenuti**;
- Integrazione con **IMDB** per velocizzare l'inserimento dei contenuti.



3. Data di Inizio e di Fine

Inizio: Ottobre 2024

Fine: Dicembre 2024

4. Deliverables

RAD, SDD, Test Plan, Test Case Specification, Test Incident Report, Test Summary Report.

5. Vincoli/Constraints

Vincoli collaborativi e comunicativi

- Rispetto scadenze
- Utilizzo di Git quale sistema di versioning e GitHub come piattaforma, dove tutti i membri del team forniscono il loro contributo
- Utilizzo degli standard dei messaggi di commit adottati
- Utilizzo di Trello quale tool di task management
- Utilizzo di Discord quale piattaforma di comunicazione e messaggistica

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** 6 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** 5 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 sistema, tramite category partition, di **esattamente** un caso d'uso specificato.



6. 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 Discord, 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**.

7. Criteri di premialità

- Uso adeguato di sistemi di **build** (ad esempio **Maven** o **Gradle**);
- Uso adeguato di un processo di **continuous integration** tramite Travis;
- Uso adeguato del tool **CheckStyle** di controllo della qualità;
- Adozione di processi di **code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).