



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

Statement of Work

Progetto

JustInTime

Riferimento	
Versione	0.2
Data	7/10/2024
Destinatario	Studenti di Ingegneria del Software 2024/25
Presentato da	Dashchuk Yulia, Fernandez Ferdinando Gregorio, Ferraro Giulia, Genovese Vincenzo
Approvato da	



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

Revision History

Data	Versione	Descrizione	Autori
07/10/2024	0.1	Prima stesura	Team
12/10/2024	0.2	Aggiunta requisiti funzionali e non funzionali	Team



INDICE

1 Introduzione.....	4
1.2 Obiettivi del Sistema.....	4
1. Data di Inizio e di Fine.....	4
2. Deliverables.....	5
3. Vincoli/Constraints.....	5
Vincoli collaborativi e comunicativi.....	5
Vincoli tecnici.....	5
4. Criteri di Accettazione/Acceptance Criteria.....	6
5. Requisiti.....	7
6.1 Requisiti Funzionali.....	7
6.2 Requisiti non funzionali.....	10

1 Introduzione

Il progetto JustInTime si propone alla creazione di una piattaforma gratuita per l'accesso all'omonimo gioco. L'idea è di rendere disponibile tale attività ad una comunità più estesa e permettere di giocare in gruppo per provare il gioco prima di acquistarlo.

1.2 Obiettivi del Sistema

Il sistema dovrà supportare le seguenti funzionalità:

- **Gestione Utenti:**
Ogni utente potrà registrarsi, autenticarsi ed effettuare il logout nella piattaforma; potrà inoltre visualizzare e modificare il proprio profilo. Ci sarà anche accessibilità alle proprie statistiche.
- **Gestione Partite:**
 - **Gioco:**
Sarà possibile la creazione di stanze locali per permettere partite tra un minimo di due giocatori e un massimo di 4.
Sarà disponibile un supporto per le regole di gioco:
il supporto è visionabile con un apposito tasto la quale l'utente stesso può decidere se visionare le istruzioni o meno.
 - **Interfaccia:**
Essa punta ad essere di facile comprensione, interattiva e soprattutto a rendere competitiva ed accattivante la sessione di gioco.
 - **Punteggi:**
Si potrà accedere ad una classifica globale e una locale per incoraggiare la competizione tra i giocatori.
 - **Classifica:**
L'utente può aumentare il proprio punteggio giocando partite a 4 giocatori e visualizzare la propria posizione nelle classifiche per incentivare la competizione tra gli utenti.

1. Data di Inizio e di Fine

- Inizio: Ottobre 2024
- Fine: Gennaio 2025

2. Deliverables

- RAD
- SDD
- ODD
- Testing funzionale (TCS, TIR, TP, TSR)

3. Vincoli/Constraints

Vincoli collaborativi e comunicativi.

- Rispetto scadenze delle scadenze intermedie/di fine progetto definite nello statement of work
- Uso di sistemi di versioning - GitHub in particolare
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo

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
- 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 membro del team dovrà effettuare il testing di unità, tramite category partition, dei casi d'uso specificati.

4. Criteri di Accettazione/Acceptance Criteria

- Utilizzo appropriato di GitHub.
- Adeguato utilizzo del pull-based development.
- Documentazione adeguata.
- 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.

5. Requisiti

6.1 Requisiti Funzionali

6.1.1 Gestione dell'utenza

RF_GU_1: Autenticazione Utente

- Il sistema dovrà permettere agli utenti registrati di autenticarsi utilizzando credenziali valide (nome utente/email e password).
- Se l'autenticazione ha successo, l'utente sarà reindirizzato alla schermata principale del gioco. Se fallisce, il sistema mostrerà un messaggio di errore.

RF_GU_2: Registrazione Utente

- Il sistema dovrà fornire agli utenti non registrati la possibilità di creare un nuovo account. L'utente dovrà inserire informazioni quali nome utente, email e password, dati personali.

RF_GU_3: Logout

- Il sistema dovrà permettere agli utenti autenticati di effettuare il logout, terminando la sessione corrente.
- Note: Dopo il logout, l'utente sarà reindirizzato alla schermata home.

RF_GU_4: Consultazione Profilo

- Il sistema dovrà permettere agli utenti autenticati di visualizzare il proprio profilo. Il profilo mostrerà informazioni come nome utente, dati e statistiche di gioco (partite giocate, vittorie, punteggio attuale).
- Note: Solo gli utenti autenticati potranno accedere a questa funzionalità.

RF_GU_5: Modifica Profilo

- Il sistema dovrà permettere agli utenti autenticati di modificare le proprie informazioni.
- Note: La modifica sarà possibile solo per utenti autenticati.

RF_GU_6: Eliminazione Profilo Utente

- Il sistema dovrà permettere agli utenti autenticati di eliminare definitivamente il proprio profilo. L'eliminazione comporterà la cancellazione di tutte le informazioni relative all'utente, incluse statistiche e dati di gioco.
- Note: L'operazione di eliminazione dovrà essere confermata dall'utente.

6.1.2 Gestione delle Partite

RF_GP_1: Creazione della Partita

- Il sistema dovrà garantire ad ogni utente registrato la creazione di una partita.

RF_GP_2: Numero di giocatori

- Deve essere presente un supporto per partite da 2 a 4 giocatori.

RF_GP_3: Setup Iniziale

- Deve avvenire una distribuzione casuale di 5 carte per ogni giocatore all'inizio di ogni partita.
- Sul tavolo da gioco saranno presenti altri 2 mazzi, il mazzo di scarto e il mazzo di pesca, inizialmente di 60 carte; il mazzo di scarti avrà una carta numerata o un jolly all'inizio della partita, Il totale delle carte presenti nel mazzo di pesca è 60.
- Sarà presente anche un timer che terrà conto del tempo di ognuno dei giocatori durante il loro turno.

RF_GP_4: Ordine di gioco

- Ad inizio di ogni partita deve essere deciso l'ordine e i turni di ogni giocatore.

RF_GP_5: Giocare una carta

- Il sistema deve permettere a ogni giocatore di giocare una carta con valore +1 o -1 rispetto alla carta presente sulla cima del mazzo di scarto.

RP_GP_6: Gestione del tempo

- In base alle regole deve avvenire una riduzione o un aumento del tempo del turno di ogni giocatore.

RP_GP_7: Pesca

- Se un giocatore pesca una carta, il sistema deve ridurre il tempo disponibile per i suoi turni futuri di 1 secondo.

RP_GP_8: Effetto delle carte speciali

- La carta “Accelera” deve ridurre di 2 secondi il tempo del turno del giocatore successivo.
- La carta “Rallenta” deve aggiungere 2 secondi al tempo del turno del giocatore che ha ottenuto la carta.
- La carta “Jolly” può essere giocata in qualunque momento del proprio turno e vale il numero sulla carta in cima al mazzo di scarti aumentato di +1.

RP_GP_8: Autenticazione utenti aggiuntivi

- Dopo aver scelto il numero di giocatori della partita il sistema fornisce ai singoli giocatori un metodo di autenticazione

6.1.3 Gestione della Classifica

RF_GC_1: Visualizzazione classifica

- Il sistema deve permettere all’utente registrato di poter visualizzare una classifica globale, insieme al suo punteggio e alla sua posizione in classifica.

RF_GC_2: Aggiornamento classifica

- Al termine di ogni partita la classifica verrà aggiornata, insieme al proprio punteggio.

6.1.4 Gestione del Feedback

RF_GF_1: Visualizzazione schermata Feedback

- Il sistema deve permettere all’utente di visualizzare una pagina che abbia un form che permetta di salvare la sua opinione.

RF_GF_2: Invio Feedback

- Il sistema deve permettere all’utente di inviare il feedback.

RF_GF_3: Visualizzazione Feedback

- Il sistema deve permettere all’admin di visualizzare i feedback inviati.

6.1.5 Gestione dei Achievement:

RF_GA_1: Visualizzazione Schermata Achievement



- Il sistema deve permettere all'utente di visualizzare una pagina che mostri tutti gli Achievement ottenuti.

RF_GA: Assegnamento automatica

- Il sistema deve assegnare automaticamente un achievement ai giocatori che soddisfano i criteri di completamento.

6.2 Requisiti non funzionali

6.2.1 Usabilità

RNF_U_1: Semplice Utilizzo

- L'utente meno esperto può interagire facilmente con il sistema poiché di comoda comprensione.

RNF_U_2: Intuitivo

- L'interfaccia utente deve essere chiara e semplice, con funzioni ben visibili e comprensibili.

RNF_U_3: Feedback Esplicito

- Ogni azione deve fornire un feedback chiaro, permettendo all'utente di comprendere e valutare facilmente il cambiamento.

RNF_U_4: Immediatezza d'utilizzo

- Il sistema deve essere intuitivo e utilizzabile senza bisogno di documentazione.

6.2.2 Affidabilità

RNF_A_1: Affidabilità delle operazioni

- Il sistema deve garantire che tutte le operazioni avvengano con successo.

RNF_A_2: Gestione delle informazioni sensibili

- Il sistema deve garantire un alto livello di sicurezza per preservare le informazioni sensibili.

RNF_A_3: Fallimento di sistema

- Il sistema deve sapersi comportare in situazioni di fallimento notificando l'utente, tramite appositi messaggi, l'errore e aiutarlo a completare le

operazioni che stava svolgendo.

RNF_A_4: Riavvio di sistema

- Il sistema deve garantire la preservazione dello stesso in caso di riavvio improvviso.

RNF_A_5: Gestione permessi sistema

- Il sistema deve garantire la separazione netta delle operazioni sulla base degli utenti che possono effettuarle.

6.2.3 Prestazioni

RNF_P_1: Sistema responsive

- Il sistema dovrà essere dotato di una interfaccia grafica responsive per potersi adattare ad ogni tipo di schermo.

RNF_P_2: Navigazione Concorrente

- Il sistema dovrà essere correttamente funzionante anche con più di un utente connesso in contemporanea

RNF_P_3: Quantità dei dati

- Il sistema dovrà garantire un'alta capacità di memorizzazione per poter conservare tutte le informazioni necessarie al suo funzionamento.

RNF_P_4: Disponibilità

- Il sistema dovrà essere disponibile 24 ore su 24.

RNF_P_5: Tempi di riposta

- Il sistema dovrà fornire la risposta alle operazioni degli utenti in un tempo di massimo 5 secondi.

6.2.4 Supportabilità

RNF_S_1: Manutenibilità

- Il sistema deve essere sviluppato rispettando gli standard per garantire una buona manutenibilità.

RNF_S_2: Estensibilità

- Il sistema deve essere sviluppato seguendo gli standard che assicurano una buona estensibilità delle funzionalità.



6.2.5 Implementazione

RNF_IM_1: Piattaforma Web

- Il sistema dovrà essere sviluppato come una piattaforma web based, seguendo il modello architetturale Tree Tier.

6.2.6 Interfacce

RNF_IN_1: Persistenza dei dati

- La Persistenza e coerenza dei dati memorizzati dal sistema dovrà essere garantita tramite Spring 3.4.1

6.2.7 Operation

RNF_OP_1: Gestione Sistema

- La gestione del sistema, della manutenzione e risoluzione dei problemi è affidata ad un Amministratore.

6.2.8 Packaging

RNF_PA_1: Applicazione Web

- Il sistema dovrà essere disponibile tramite pagina web da pc.

6.2.9 Legali

RNF_LE_1: Rispetto delle Leggi

- L'applicazione garantirà il rispetto delle leggi in materia di privacy, specificatamente del regolamento UE 2016/679 in materia di protezione e trattamento dei dati personali.