



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

System Design Document JustInTime

Riferimento	
Versione	1.2
Data	18/11/2024
Destinatario	Studenti di Ingegneria del Software 2024/25
Presentato da	Dashchuk Yulia, Fernandez Ferdinando Gregorio, Ferraro Giulia, Genovese Vincenzo
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
18/11/2024	0.1	Prima stesura	Team
19/11/2024	0.2	Aggiunta delle sezioni 2 e 3	Vincenzo Genovese Yulia Dashchuk
21/11/2024	0.3	Scomposizione in sottosistemi	Team
22/11/2024	0.4	Controllo degli accessi e sicurezza	Yulia Dashchuk
22/11/2024	0.5	Aggiunta di condizioni limite	Vincenzo Genovese
25/11/2024	0.6	Aggiunta dei Design Goals	Giulia Ferraro
24/11/2024	0.7	Aggiunta della Sezione 1	Ferdinando Gregorio Fernandez
26/11/2024	0.8	Aggiunta di Condizioni limite spegnimento, accesso a dati persistenti. Aggiunto sottosistema gestione partita	Ferdinando Gregorio Fernandez
26/11/2024	0.9	Aggiunta di servizi dei sottosistemi registrazione e autenticazione	Yulia Dashchuk
28/11/2024	0.10	Revisione component diagram e deployment diagram	Vincenzo Genovese
15/12/2024	0.11	Aggiunta sottosistema gestione Achievement	Ferdinando Gregorio Fernandez
15/12/2024	0.12	Revisione Component Diagram	Genovese Vincenzo
15/12/2024	0.13	Aggiunta del diagramma Architetturale	Team
17/12/2024	0.14	Correzione formattazione	Yulia Dashchuk



		documento	
22/12/2024	0.15	Modifiche e correzione dei sottosistemi	Team
22/12/2024	0.16	Correzione diagramma architetturale e component diagram	Yulia Dashchuk
22/12/2024	0.17	Correzione mapping hardware/software	Vincenzo Genovese
22/12/2024	0.18	Completa ristrutturazione del diagramma architetturale e dei servizi dei sottosistemi.	Ferdinando Gregorio Fernandez Giulia Ferraro
22/12/2024	0.19	Correzione formattazione documento	Yulia Dashchuk
26/12/2024	1.0	Correzione diagramma architetturale	Ferdinando Gregorio Fernandez Giulia Ferraro
26/12/2024	1.1	Correzione diagramma sottosistema Gestione Carte	Giulia Ferraro
29/12/2024	1.2	Revisione finale	Team



INDICE

Revision History.....	2
1 Introduzione.....	5
1.1 Scopo del sistema.....	5
1.2 Obiettivi di Design (design goals).....	5
2 Architettura del sistema corrente.....	11
3 Architettura del sistema proposto.....	11
3.1 Panoramica sulla sezione.....	11
3.2 Decomposizione in sottosistemi.....	11
3.3 Mapping Hardware/Software.....	17
3.4 Gestione dei dati persistenti.....	18
3.5 Controllo degli accessi e sicurezza.....	19
3.6 Controllo globale del software.....	19
3.7 Condizioni limite.....	20
4 Servizi dei sottosistemi.....	24

1 Introduzione

1.1 Scopo del sistema

JustInTime è un gioco di carte online che punta a offrire un'esperienza di intrattenimento veloce e dinamica, dove ogni mossa deve essere fatta in un tempo limitato.

Il sistema è gestito da vari **Amministratori**. JustInTime funge anche da **piattaforma di prova** per un futuro gioco di carte commerciale, mentre la versione attuale rimarrà gratuita online.

Le principali funzioni del sistema includono:

- **Partite e Gameplay:** gestione dei turni, delle regole e del tempo disponibile per le mosse, con distribuzione automatica delle carte ai giocatori.
- **Classifiche e Punteggi:** aggiornamento in tempo reale delle classifiche e registrazione dei punteggi in base alle performance di gioco.
- **Feedback degli Utenti:** raccolta di opinioni e suggerimenti per migliorare il gioco attuale e il futuro prodotto commerciale.

JustInTime si propone di essere semplice da usare e accessibile, offrendo un ambiente di gioco piacevole per tutti, dai principianti ai giocatori più esperti.

1.2 Obiettivi di Design (design goals)

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goal.

Seguendo le linee guida del libro Bernd Bruegge – Object Oriented Software Engineering i design goal sono stati suddivisi nelle seguenti categorie:

- **Performance:** includono i requisiti di spazio e velocità imposti sul sistema.
- **Dependability:** determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza) e le loro conseguenze.

- **Maintenance:** determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
- **End User:** includono qualità che sono desiderabili dal punto di vista dell'utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ciascun design goal è descritto da:

- **Rank**, che ne specifica un valore di priorità compreso tra 1 e 16 (1 massima e 16 minima).
- **ID Design Goal**, un identificatore univoco e un nome esplicativo.
- **Descrizione**, una descrizione del design goal.
- **Categoria**, ovvero la categoria di appartenenza del design goal.
- **RNF di origine**, ovvero il requisito non funzionale che lo ha generato.

Design goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
3	DG_1 Tempi di risposta	Il sistema deve garantire un tempo di risposta non superiore a 5 secondi.	Performance	RNF_P_5
4	DG_2 Quantità di dati	Il sistema dovrà gestire tutti i dati riguardanti gli utenti i giocatori e le partite, servirà quindi un grande spazio per contenere questi dati.	Performance	RNF_P_3
9	DG_3 Navigazione concorrente	Il sistema dovrà essere correttamente funzionante anche con un elevato numero di utenti connessi in contemporanea.	Performance	RNF_P_2
5	DG_4 Affidabilità	Il sistema deve garantire che tutte le operazioni avvengano con successo.	Dependability	RNF_A_1

	delle operazioni			
14	DG_5 Fallimento di sistema	Il sistema deve sapersi comportare in situazioni di fallimento notificando l'utente, tramite appositi messaggi.	Dependability	RNF_A_3
7	DG_6 Gestione permessi	Il sistema deve garantire una divisione tra le varie categorie di utenti, al fine di limitare le funzionalità accessibili ad ognuno e mantenere così l'integrità del sistema.	Dependability	RNF_A_5
15	DG_7 Disponibilità del sistema	Il Sistema deve garantire la massima disponibilità.	Dependability	RNF_P_4
1	DG_8 Sicurezza dei dati	Il Sistema deve garantire la massima sicurezza dei dati conservati, utilizzando protocolli di comunicazione sicuri, conservando in maniera sicura i dati persistenti, e assicurando la visualizzazione dei dati solo agli utenti che hanno diritto ad accedervi.	Dependability	RNF_A_2
8	DG_10 Manutenibilità	Il sistema deve essere facilmente manutenibile ed estendibile.	Maintenance	RNF_S_1

8	DG_11 Estensibilità	Il sistema si presta facilmente all'aggiunta di nuove funzionalità poiché segue lo standard ISO/IEC 9126	Maintenance	RNF_S_2
7	DG_12 Facilità d'uso	Il sistema deve risultare facilmente comprensibile ed utilizzabile anche da un'utenza meno esperta seguendo lo standard ISO 9241-210	End user	RNF_U_1
10	DG_13 Interfaccia intuitiva	L'interfaccia utente della piattaforma deve permettere di eseguire azioni in modo chiaro e semplice, rendendo ben esplicita la funzionalità di ogni elemento visuale.	End User	RNF_U_2
12	DG_14 Feedback esplicito	Ogni azione all'interno della piattaforma in seguito ad un'interazione dell'utente deve comunicare un chiaro feedback allo stesso	End User	RNF_U_3

Trade-offs

Trade-off	Descrizione
-----------	-------------

Tempi di risposta vs sicurezza

Per garantire una sicurezza del sito si punta ad implementare sistemi che aumentino la stessa a discapito della velocità delle operazioni le quali potrebbero impiegare fino a 5 secondi.

1.3 Definizione, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.
- **Mapping hardware/software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **TIR:** Test Incident Report
- **TP:** Test Plan
- **TCS:** Test Case Specification
- **TSR:** Test Summary Report

1.4 Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

- Statement of Work
- Requirements Analysis Document
- Test Incident Report
- Test Plan
- Test Case Specification
- Test Summary Report

1.5 Organizzazione del documento

Il presente documento di System Design consta di quattro sezioni:

- **Introduzione:** Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.
- **Architettura software corrente:** Viene descritto lo stato attuale dell'architettura del software già presente.



- **Architettura software proposta:** Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.

2 Architettura del sistema corrente

Al momento, non esiste alcun software che condensi l'interesse delle funzionalità di JustInTime in un unico servizio. Il mercato delle possibili alternative a questo software è pertanto frammentato e esistono software che ne rappresentano solo parzialmente le funzionalità, come Skip-Bo (per le dinamiche basate sulle carte) e Scacchi Partita Lampo (il quale presenta un timer e quindi incentiva la competitività del gioco).

3 Architettura del sistema proposto

3.1 Panoramica sulla sezione

Il sistema proposto è basato sullo stile architetturale Three Tier implementato utilizzando Spring MVC. Il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come il nostro sistema, poiché la separazione della logica di presentazione da quella di elaborazione, migliora una serie di qualità, tra le quali:

- Leggibilità
- Manutenzione
- Riutilizzo

Nello sviluppo del sistema verranno usati HTML5 e CSS per la parte di front-end e la generazione delle view.

Per la logica applicativa e quindi il back-end sarà utilizzato Java SPRING.

Per la gestione del database sarà usato Spring.

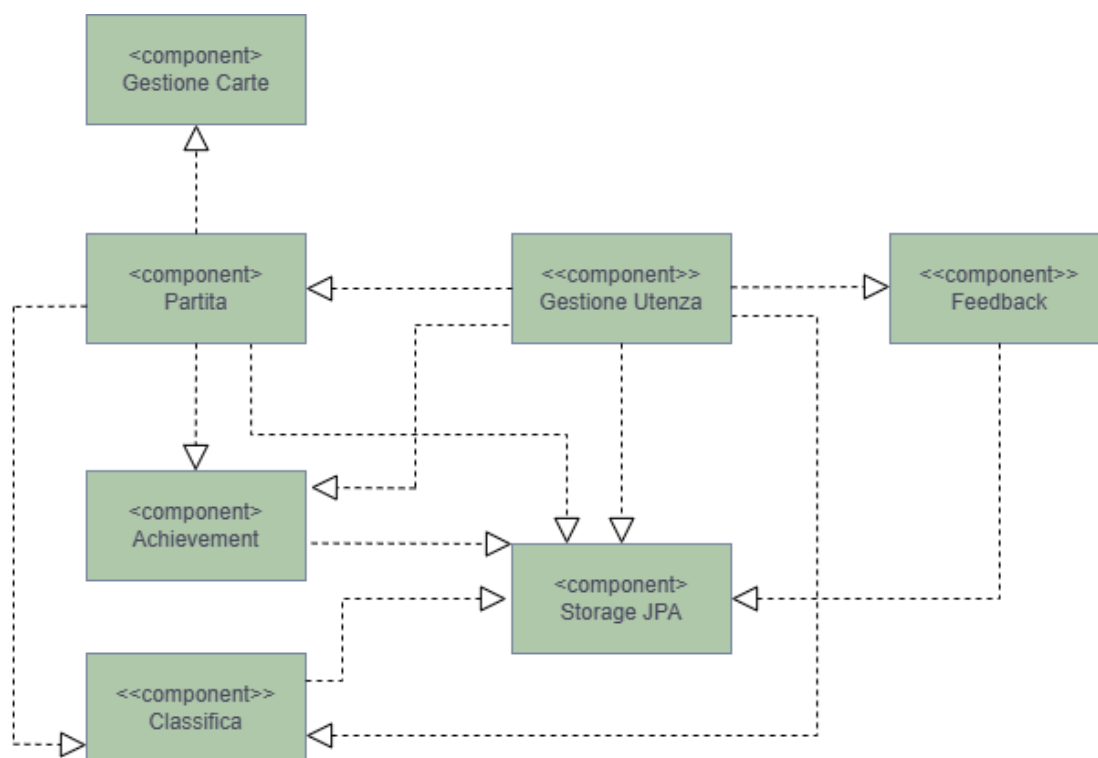
3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Registrazione:** si occupa di gestire la registrazione del giocatore
- **Autenticazione:** è responsabile delle funzionalità di Login, Logout, visualizzazione area utente e la modifica dati account.
- **Gestione Carte:**
Si occupa della gestione del mazzo di pesca e di scarto e di come interagiscono con le mani dei player.
- **Classifica:** è responsabile della visualizzazione della classifica.

- **Gestione partita:** si occupa delle funzioni riguardanti la creazione della partita, la visualizzazione dei risultati e il salvataggio delle informazioni.
- **Feedback:** si occupa di gestire il rilascio di un feedback
- **Achievement:** si occupa della gestione degli achievement.
- **Persistenza:** si occupa di gestire la persistenza dei dati con un database.
- **Storage JPA:** si interpone tra i vari sottosistemi e il sottosistema di Persistenza.

Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un component diagram UML.

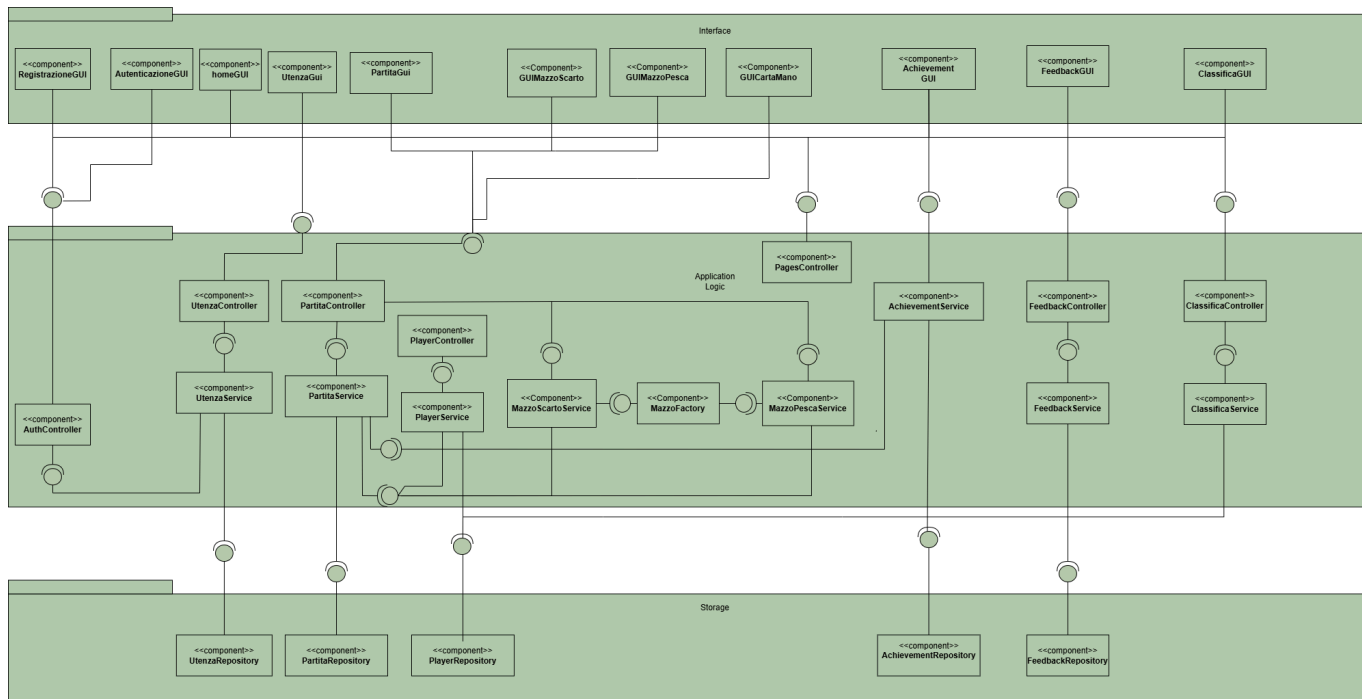


Alcuni sottosistemi saranno gestiti da componenti COTS (Commercial off the shelf):
storage JPA verrà gestito da Spring Data JPA

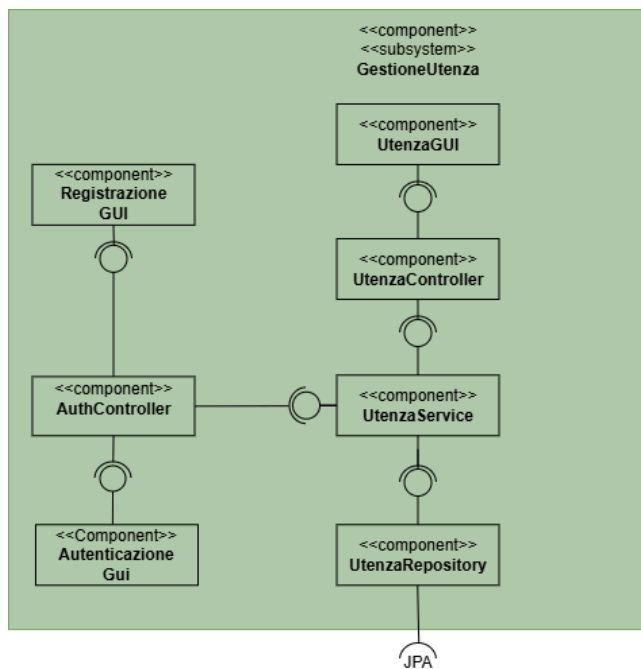
Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principali:

- GUI: Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare al cliente.
- Controller: si occupa della logica per il controllo del sistema.
- Service: si occupa della logica di business.
- DAO: Data Access Object, che si occupa di fornire accesso ai dati persistenti.

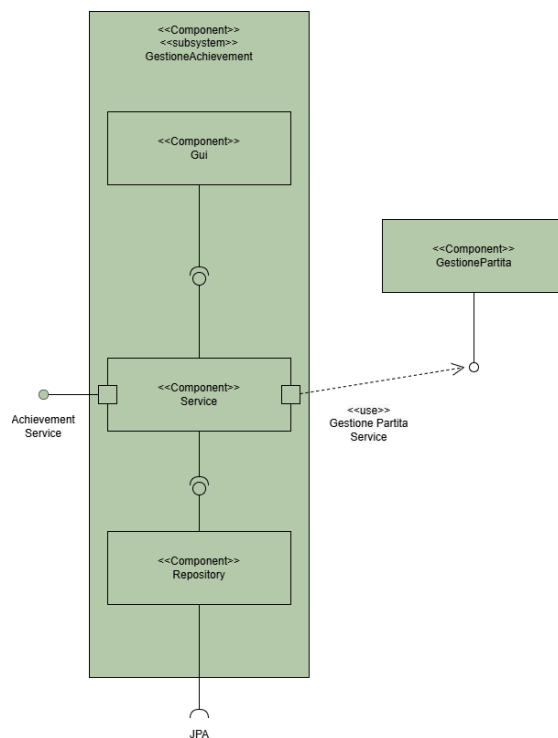
Diagramma architetturale



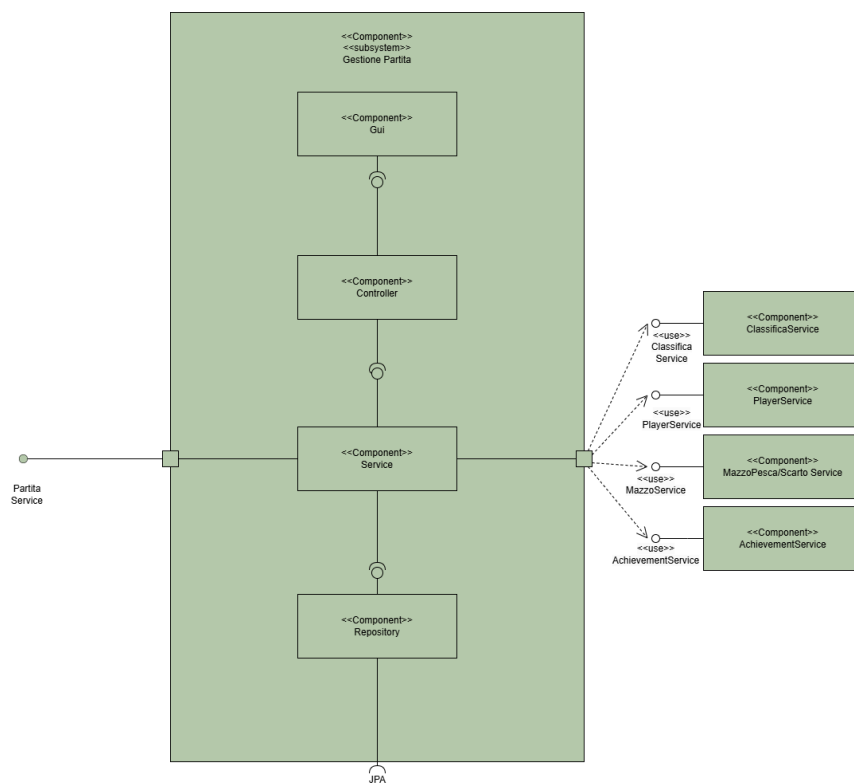
Sottosistema Gestione Utente



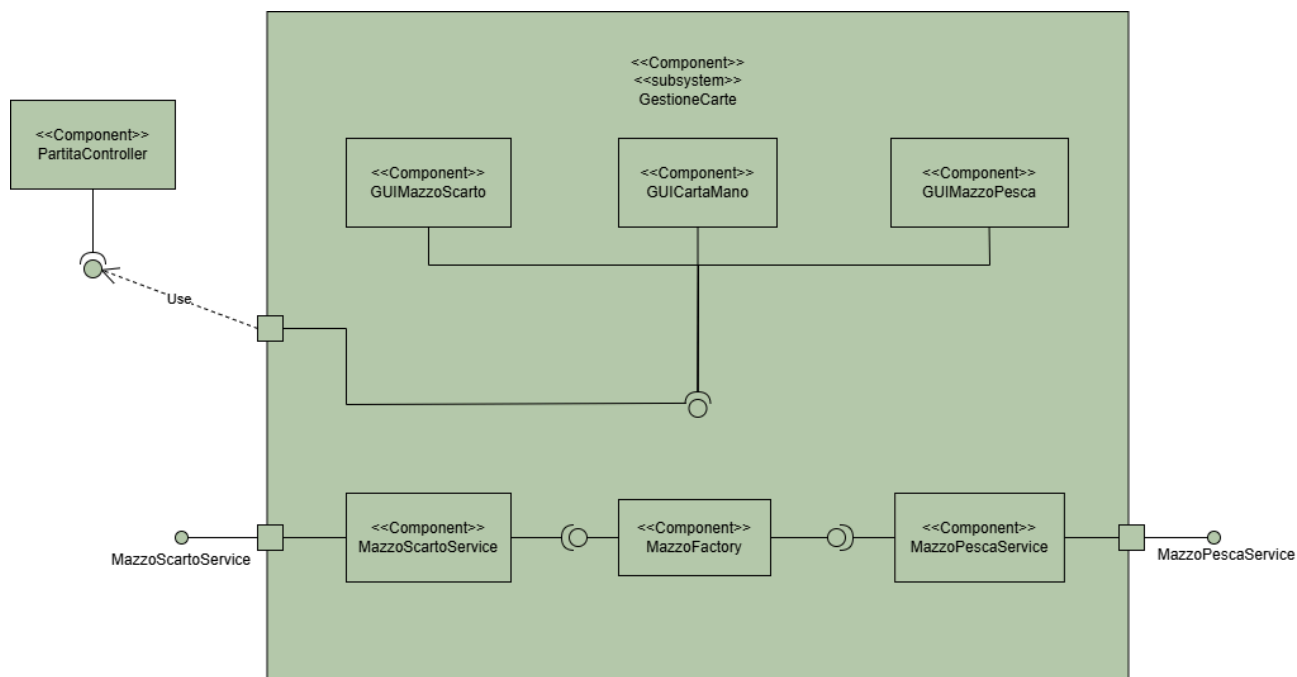
Sottosistema Achievement



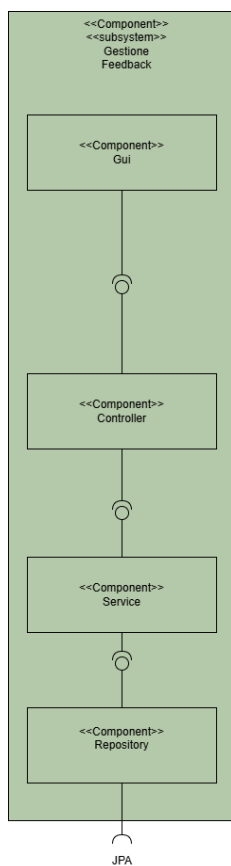
Sottosistema Gestione Partita



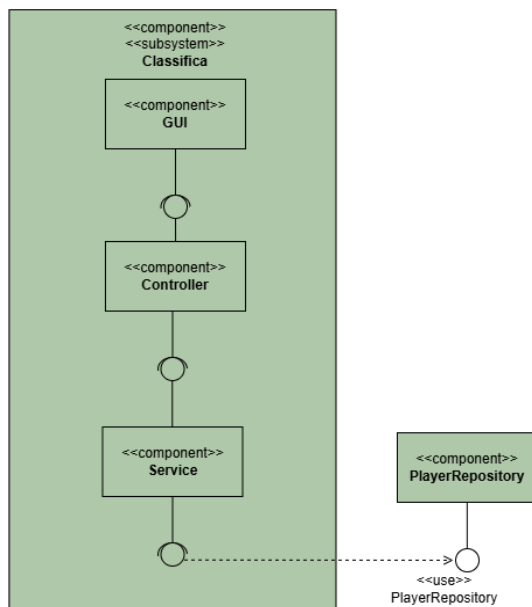
Sottosistema Gestione Carte



Sottosistema Feedback



Sottosistema Classifica

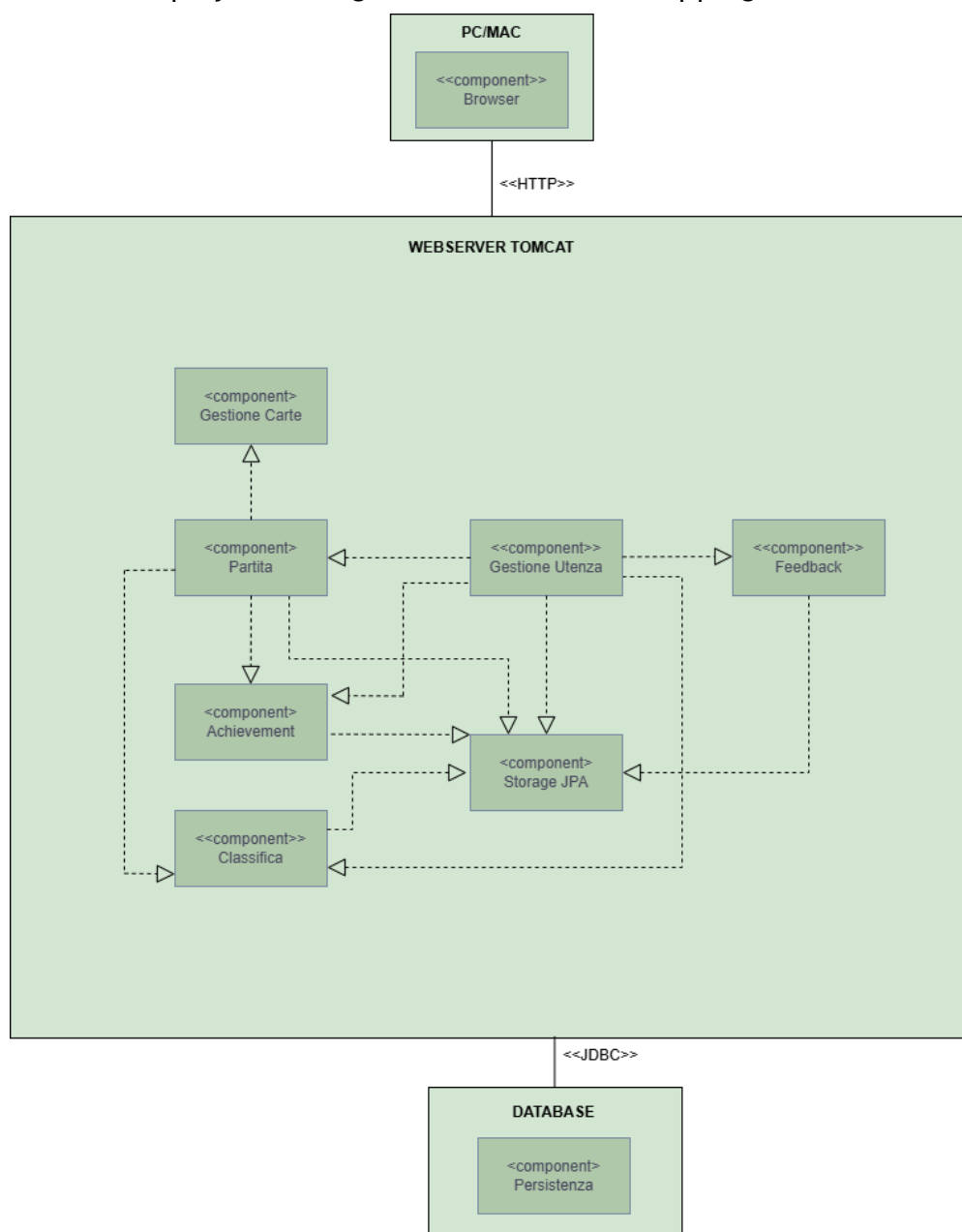


3.3 Mapping Hardware/Software

L' applicazione web che verrà sviluppata si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dai clienti da una qualsiasi macchina con un browser ed una connessione ad Internet.

Essendo che il nostro sistema è una web application e risiede su un web server, e che si basa su un'architettura non distribuita, risiede su un solo nodo.

Di seguito un UML deployment diagram che descrive il mapping hardware/software.



3.4 Gestione dei dati persistenti

Introduzione

Per la gestione del salvataggio dei dati persistenti del sistema si è deciso di utilizzare un database relazionale, al fine di gestire agevolmente l'accesso concorrente ai dati e contemporaneamente garantire la consistenza dei dati tramite l'utilizzo di un DBMS.

La scelta di utilizzo di un DBMS è stata presa al fine di mantenerci quanto più possibile coerenti con i design goals stabiliti, potendo contare su:

- **Imposizioni di vincoli di integrità sui dati**, poiché un DBMS permette di specificare diversi tipi di vincoli per mantenere l'integrità dei dati e controlla che tali vincoli siano soddisfatti quando la base di dati cambia
- **Privatezza dei dati**, garantita dal fatto che un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- **Affidabilità dei dati**, infatti un DBMS offre dei metodi per salvare copie dei dati e per ripristinare lo stato della base di dati in caso di guasti software e hardware.
- **Atomicità delle operazioni**, data dal fatto che un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base. L'atomicità delle transazioni permette di mantenere uno stato della base di dati consistente con la realtà modellata.

3.5 Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere ai vari servizi offerti dal sistema.

Attori			
Oggetti	Amministratore	Ospite	Utente
Registrazione	Registrazione		
Autenticazione	Login Logout VisualizzazioneUtenti		Login Logout VisualizzazioneAreaUtente ModificaDatiUtente CancellazioneAccount
Partita	InizioPartita		
Classifica	VisualizzazioneClassificaLocale VisualizzazioneClassificaGlobale		VisualizzazioneClassificaLocale VisualizzazioneClassificaGlobale
Feedback	VisualizzazioneFeedback		RilascioFeedback

3.6 Controllo globale del software

Il sistema JustInTime è un sistema interattivo per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente tramite l'uso di un'interfaccia grafica. Quindi quando un utente vuole accedere ed utilizzare una funzionalità del sistema può farlo tramite l'interfaccia grafica la quale selezionerà il controllo corrispondente.

L'azione scatenerà un evento il quale verrà gestito dal suo handler. L'handler a sua volta indirizzerà il controllo del flusso di eventi al sottosistema che si occupa della logica di controllo e gestore del controllo che poi si rivolge ai servizi per la logica applicativa. Per tali motivi il sistema utilizzerà un meccanismo di controllo del flusso di tipo event-driven essendo una web-application.

3.7 Condizioni limite

Nel presente paragrafo verranno presentate le boundary conditions inerenti all'avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

Avvio del sistema

Identificativo	UCBC_1 - Avvio del sistema	Data	26/11/2024
		Versione	1.0
		Autori	Giulia Ferraro
Descrizione	L'UC permette l'avvio del sistema.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'Amministratore accede al Server		
Exit condition On success	Il sistema viene avviato correttamente		
Exit condition On failure	Il sistema non viene avviato		
Flusso di eventi principale			
1	Amministratore	Esegue sulla macchina il comando che avvia il sistema.	
2	Sistema	Verifica la sanità dei dati persistenti e, se sani, rende disponibili i suoi servizi e rende le sue funzionalità disponibili agli utenti.	
I Flusso di eventi alternativo: I dati persistenti sono danneggiati			
2.a1	Sistema	Notifica l'Amministratore di problemi ai dati persistenti e non effettua l'avvio.	
2.a2	Amministratore	Corregge i dati persistenti .	
2.a3	Amministratore	Esegue il passaggio 1.	

Spegnimento del sistema

Identificativo	UCBC_2 - Spegnimento del sistema	Data	26/11/2024
		Versione	1.0
		Autori	Ferdinando Gregorio Fernandez
Descrizione	L'UC permette lo spegnimento del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore accede al server AND Il sistema è stato avviato in precedenza AND Il sistema è ancora acceso		
Exit condition On success	Il sistema viene spento correttamente		
Exit condition On failure	Il Sistema non viene spento		
Flusso di eventi principale			
1	Amministratore	Invia Un segnale di spegnimento del sistema	
2	Sistema	Controlla che non ci siano connessioni aperte da e verso l'esterno. Se non ci sono termina il sistema.	
I Flusso di eventi alternativo: Ci sono ancora connessioni aperte			
2.a1	Sistema	Notifica all'admin che sono ancora presenti delle connessioni aperte verso l'esterno.	
2.a2	Sistema	Attende un eventuale periodo di tempo finché non vengono piu' generate nuove connessioni dall'esterno se non per rispondere a richieste già in corso.	
2.a3	Sistema	Controlla non ci siano connessioni aperte e termina il sistema.	
2.a4	Sistema	Notifica lo spegnimento del sistema.	
II Flusso di eventi alternativo: Ci sono ancora connessioni aperte			
2.a3.a1	Sistema	Recide le connessioni verso l'esterno.	

2.a3.a2	Sistema	Notifica dell'avvenuto spegnimento e del numero di connessioni recise.
---------	---------	--

Fallimento del sistema

Identificativo	UCBC_3 - Fallimento del Sistema	Data	22/11/2024
		Versione	1.0
		Autori	Genovese Vincenzo
Descrizione	L'UC definisce il comportamento del Sistema in caso di fallimento.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il Sistema viene terminato inaspettatamente		
Exit condition On success	Il sistema viene riavviato correttamente		
Exit condition On failure	Il Sistema non viene riavviato		
Flusso di eventi principale			
1	Amministratore	Include UCBC_1	

Errore di Accesso ai Dati Persistenti

Identificativo	UCBC_4 - Errore di accesso ai dati persistenti	Data	26/11/2024
		Versione	1.0
		Autori	Ferdinando Gregorio Fernandez
Descrizione	L'UC definisce il comportamento del Sistema in caso di errore di accesso ai dati persistenti		
Attore Principale	Utente		
Attori secondari	NA		
Entry Condition	Il sistema tenta di accedere ai dati persistenti ma si verifica un errore.		
Exit condition On success	Il sistema accede ai dati persistenti.		

Exit condition On failure		Il sistema non accede ai dati persistenti.
Flusso di eventi principale		
1	Utente	L'utente esegue un'operazione che richiede l'accesso ai dati persistenti (Es: Visualizzazione classifica o achievements).
2	Sistema	Il Sistema rileva un errore durante l'accesso ai dati persistenti.
3	Sistema	Il Sistema notifica l'errore all'amministratore tramite un messaggio appropriato (es. "Impossibile accedere ai dati persistenti, dato non trovato).
4	Sistema	L'operazione termina senza successo.
I Flusso di eventi alternativo: Impossibile connettersi al database		
2.a1	Sistema	Il Sistema rileva un errore durante l'accesso ai dati. (Impossibile connettersi al database).
2.a2	Sistema	Il Sistema notifica l'errore all'amministratore l'impossibilità di connessione al database.
2.a3	Sistema	.L'operazione termina senza successo.

4 Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

Sottosistema Gestione Utenza

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare tutte le funzionalità che offre.	UtenzaService
Registrazione utente	Questa funzionalità permette di registrarsi sulla piattaforma come utente.	UtenzaService
Logout	Questa funzionalità permette di disconnettersi dal sistema.	UtenzaService
Visualizza area utente	Permette di visualizzare i dati relativi alla propria area utente.	UtenzaService
Modifica dati utente	Permette di modificare i dati relativi alla propria area utente.	UtenzaService
Cancellazione account	Permette di cancellare il proprio account sulla piattaforma.	UtenzaService

Sottosistema Achievement

Servizio	Descrizione	Interfaccia
Visualizza Achievement	Permette di visualizzare la lista degli achievements disponibili, quelli sbloccati dall'utente e quelli ancora bloccati.	AchievementService
Sblocca Achievement	Registra lo sblocco di un achievement per un utente al raggiungimento di determinati criteri.	AchievementService

Sottosistema Gestione Partita

Servizio	Descrizione	Interfaccia
Avvia Partita	Permette di creare e avviare una nuova partita configurando i parametri necessari (giocatori, regole, ecc.).	ConfigPartita Service
Gestione turno	Permette di gestire i turni dei giocatori, determinando chi può giocare e monitorando il tempo limite.	Partita Service
Gestione achievement	Riconosce e assegna obiettivi raggiunti dai giocatori durante una partita.	Partita Service
Abbandona partita	Permette ai giocatori di abbandonare la partita.	Partita Service

Sottosistema Classifica

Servizio	Descrizione	Interfaccia
Visualizza Classifica Locale	Permette di creare e visualizzare la classifica locale, ovvero in base alla sua nazione, e la posizione del giocatore	ClassificaService
Visualizza Classifica Globale	Permette di creare e visualizzare la classifica globale e la posizione del giocatore	ClassificaService

Sottosistema Feedback

Servizio	Descrizione	Interfaccia
Selezione Stelle	Permette di selezionare il quantitativo di stelle da attribuire al feedback	FeedbackService
Descrizione	Permette di scrivere un parere riguardo il gioco	FeedbackService

Sottosistema Gestione Carte

Servizio	Descrizione	Interfaccia
Gioca Carta	Permette di giocare la carta selezionata dal giocatore	Partita Service
Pesca Carta	Permette al giocatore di pescare una carta dal mazzo di pesca	Partita Service