

Requisiti

Versione	Data	Descrizione
1.0	26 Novembre 2025	Prima versione del documento Requisiti
1.1	1 Dicembre 2025	Modifica Classi nei requisiti

Qualità del software

Per garantire la qualità del software utilizzeremo la tassonomia di McCall che definisce tre categorie di fattori di qualità; noi ci concentreremo solo su alcuni:

1. Funzionamento del prodotto:
 - Correttezza: dovrà coprire tutti i requisiti elicitati
 - corretto utilizzo del Product Backlog con tutte le User Stories
 - i requisiti dovranno avere criteri di accettazione chiari
 - Affidabilità: dovrà funzionare senza crash
 - gestione delle eccezioni
 - utilizzo piattaforma di logging
 - Usabilità: dovrà essere semplice ed intuitiva da usare
 - GUI facile da interpretare e utilizzare
2. Revisione del prodotto:
 - Manutenibilità: dovrà essere facile da mantenere
 - documentazione del codice e naming curati
 - Testabilità: dovrà avere test sulle regole di conteggio punti
 - test JUnit per simulare le giocate
3. Transizione del prodotto:
 - Portabilità: dovrà funzionare su diversi tipi di client
 - JVM per astrarre l'architettura del calcolatore

Ingegneria dei requisiti

Elicitazione dei requisiti

L'elicitazione dei requisiti è avvenuta tramite:

- Analisi delle attività a partire dall'analisi delle regole di gioco
- Brainstorming tra i membri del team di sviluppo
- Descrizioni in linguaggio naturale

Ovviamente il nostro team simulerà i bisogni e le richieste del cliente e impersonificherà il Product Owner.

Specificazione dei requisiti

Per la specifica dei requisiti utilizzeremo lo standard **IEEE 830** che ne definisce la struttura:

1. Introduzione
 - 1.1. Obiettivo: la specifica dei requisiti ha il compito di descrivere in modo formale e verificabile i requisiti del sistema

- 1.2. Scopo: il software deve permettere di far giocare a Poker un gruppo di utenti da calcolatori diversi
- 1.3. Definizioni, acronimi e abbreviazioni
 - JVM: Java Virtual Machine, ambiente di esecuzione che permette di eseguire programmi scritti in linguaggio Java su qualsiasi piattaforma che la supporta, tramite astrazione dell'hardware sottostante
 - DB: database
 - IDE: ambiente di sviluppo del codice
 - GUI: Graphical User Interface, ambiente di controllo per il giocatore
- 1.4. Riferimenti: regolamento ufficiale di Poker
2. Descrizione generale
 - 2.1. Prospettiva del prodotto
 - Il sistema è un'applicazione client/server per giocare a Poker progettata per piattaforme computer che supportano la JVM
 - Il sistema si interfaccia con un DB per la gestione dei dati utente
 - Il sistema si dividerà in 3 moduli: client, logica di gioco, server
 - 2.2. Funzioni del prodotto
 - Gestione utenti
 - Creazione partite
 - Gestione del flusso di gioco e GUI
 - Gestione del flusso di dati
 - 2.3. Caratteristiche dell'utente
 - Admin: moderatore e gestore del sistema
 - Player: utente finale
 - 2.4. Vincoli
 - Il sistema utilizza un'architettura client/server
 - Il codice è scritto in linguaggio Java su IDE Eclipse
 - 2.5. Presupposti e dipendenze
 - Necessaria una connessione stabile ad internet
 - **Il DB viene gestito con una dipendenza Maven**
3. Requisiti specifici
 - 3.1. Requisiti dell'interfaccia esterna
 - 3.1.1. Interfaccia utente
 - Login e registrazione
 - Tavolo da gioco, carte e fiches
 - Pulsanti di azione
 - 3.1.2. Interfaccia hardware
 - Client: pc che supporti una JVM e periferiche di controllo
 - Server: pc che ne supporti l'esecuzione
 - 3.1.3. Interfaccia software
 - libreria per supportare l'architettura client/server
 - 3.1.4. **Interfacce di comunicazione**

3.2. Richieste funzionali

3.2.1. Classe Client

	ID	Requisito	Descrizione
<input checked="" type="checkbox"/>	1	Creazione Player	L'utente può creare un nuovo account oppure accedere ad uno già esistente dalla schermata di Login
<input checked="" type="checkbox"/>	2	Partecipa partita	L'utente può partecipare alla partita unendosi al tavolo solo a partita appena creata
<input type="checkbox"/>	3	Esci partita	L'utente può uscire dalla partita abbandonando il tavolo solo dopo la fine della mano attuale.
<input type="checkbox"/>	4	Flusso di gioco	Deve poter gestire il flusso di gioco comunicando con il server
<input checked="" type="checkbox"/>	4.1	Invito	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	4.2	Apertura	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	4.3	Accomodo	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	4.4	Puntata	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	4.5	Showdown	Deve poter inviare e ricevere dati sulla fase dal server

3.2.2. Classe Player

	ID	Requisito	Descrizione
<input type="checkbox"/>	1	Azioni di gioco	<p>Le azioni di gioco saranno implementate sulla GUI come bottoni e rappresentano le scelte che il giocatore può effettuare durante le diverse fasi di gioco, in diverse condizioni.</p> <p>Prima di distribuire le carte (invito):</p> <ul style="list-style-type: none"> • Partecipa <ul style="list-style-type: none"> ◦ Punta • Non partecipare <p>Appena sono state distribuite le carte (apertura):</p> <ul style="list-style-type: none"> • Apri • Passa <p>Appena uno dei giocatori apre diventa il primo del giro, quindi tutti:</p> <ul style="list-style-type: none"> • Vedi • Punta • Lascia <p>Quando tutti hanno deciso se lasciare o vedere (accomodo):</p> <ul style="list-style-type: none"> • Cambio • Servito <p>Quando tutti hanno gestito le proprie carte, sempre dal giocatore che aveva aperto:</p> <ul style="list-style-type: none"> • Vedi • Punta • Lascia

<input checked="" type="checkbox"/>	1.1	Partecipa invito	L'utente partecipa alla mano puntando il minimo del piatto
<input checked="" type="checkbox"/>	1.2	Punta	<ul style="list-style-type: none"> L'utente può decidere la quantità da puntare inserendo il valore numerico in un textfield: <ul style="list-style-type: none"> puntata di partecipazione: fissa alla minima del tavolo puntata nella fase di punata: si continua a puntare fino a quando tutti gli attivi vedono
<input checked="" type="checkbox"/>	1.3	Apri	<ul style="list-style-type: none"> Può aprire solo il giocatore che ha almeno una coppia di Jack (ma può anche passare) Quando apre punta
<input checked="" type="checkbox"/>	1.4	Passa	<ul style="list-style-type: none"> Se l'utente non può aprire perchè non ha almeno una coppia di Jack allora è costretto a passare Si può passare anche se si può aprire Se nessuno può aprire (tutti hanno passato), il piatto viene diviso equamente
<input checked="" type="checkbox"/>	1.5	Lascia	Permette al giocatore di lasciare il tavolo
<input checked="" type="checkbox"/>	1.6	Vedi	<ul style="list-style-type: none"> Nella fase di apertura solo se un giocatore ha aperto allora l'utente può vedere partecipando alla mano con una puntata che eguagli la puntata massima (il debito viene saldato) Nella fase post-accomodo l'utente può vedere partecipando alla mano con una puntata che eguagli la puntata massima (il debito viene saldato)
<input checked="" type="checkbox"/>	1.7	Cambio	Il giocatore cambia fino a 5 carte (solo nella fase di accomodo)
<input checked="" type="checkbox"/>	1.8	Servito	Il giocatore non cambia carte (solo nella fase di accomodo)
<input type="checkbox"/>	2	Pronto	Il client è pronto per giocare la prossima mano
<input checked="" type="checkbox"/>	3	Fiches	Il credito iniziale di ogni giocatore è definito dal tavolo ed è uguale per tutti
<input checked="" type="checkbox"/>	4	Mano	Ogni giocatore avrà la sua mano

3.2.3. Classe Server

	ID	Requisito	Descrizione
<input checked="" type="checkbox"/>	1	Turnazione	La turnazione è definita dall'ordine di arrivo al tavolo
<input checked="" type="checkbox"/>	2	Mazzo di carte	Deve poter gestire il mazzo di carte
<input checked="" type="checkbox"/>	2.1	Numero di carte	Il mazzo sarà composto da 32 carte (dal 7 all'asso)
<input checked="" type="checkbox"/>	2.2	Shuffle	Deve prevedere una funzione per mescolare il mazzo di carte

<input type="checkbox"/>	2.3	Distribuire	<ul style="list-style-type: none"> distribuire 5 carte ad ogni giocatore cambiare le carte durante la fase di accomodo
<input type="checkbox"/>	3	Showdown	Deve saper gestire la fase di showdown
<input type="checkbox"/>	3.1	Check vincitore	Deve poter determinare il vincitore della mano. In tutti i casi vince il punteggio più alto dovuto alle carte che fanno punteggio in mano
<input checked="" type="checkbox"/>	3.1.1	Carta alta	Il punto è dato dalla carta più alta delle cinque
<input checked="" type="checkbox"/>	3.1.2	Coppia	Formata da due carte dello stesso valore più tre carte diverse
<input checked="" type="checkbox"/>	3.1.3	Doppia coppia	Formata da due coppie di carte dello stesso valore
<input checked="" type="checkbox"/>	3.1.4	Tris	Formato da tre carte dello stesso valore più due carte diverse
<input checked="" type="checkbox"/>	3.1.5	Scala	Formata da una sequenza continua di cinque carte in cui non viene tenuto conto del seme ma solo della corretta successione, partendo da qualsiasi valore
<input checked="" type="checkbox"/>	3.1.6	Colore	Formato da cinque carte, non in sequenza completa, dello stesso seme . Vince la mano con il seme di rango più alto
<input checked="" type="checkbox"/>	3.1.7	Full	Formato da un tris più una coppia
<input checked="" type="checkbox"/>	3.1.8	Poker	Formato da quattro carte dello stesso valore
<input checked="" type="checkbox"/>	3.1.9	Scala colore	Formata da una sequenza continua di carte dello stesso seme
<input checked="" type="checkbox"/>	4	Numero giocatori	Deve gestire 4 giocatori al massimo ad un tavolo, non permettendo ad altri client di connettersi
<input type="checkbox"/>	5	Flusso di gioco	Deve poter gestire il flusso di gioco e la comunicazione con il client
<input checked="" type="checkbox"/>	5.1	Invito	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	5.2	Apertura	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	5.3	Accomodo	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	5.4	Puntata	Deve poter inviare e ricevere dati sulla fase dal server
<input type="checkbox"/>	5.5	Showdown	Deve poter inviare e ricevere dati sulla fase dal server

3.2.4. Classe Database

	ID	Requisito	Descrizione
--	----	-----------	-------------

<input type="checkbox"/>	1	Connessione	La classe dovrà potersi connettere al file db presente nella repository
<input type="checkbox"/>	2	Metodi query	Dovrà fornire tutti i metodi necessari per la registrazione e il login
<input type="checkbox"/>	2.1	Username inesistente	Gestione eccezione di “account inesistente” durante il login
<input type="checkbox"/>	2.2	Password errata	Gestione eccezione di “password errata” durante il login
<input type="checkbox"/>	2.3	Username già esistente	Gestione eccezione di “account esistente” se ci si vuole registrare con un username già utilizzato
<input type="checkbox"/>	3	Protezione	Fornire diversi metodi in base ai permessi dei diversi utenti (controllo su quali istanze eseguono determinati metodi)
<input type="checkbox"/>	5	Username e password	Dovrà poter memorizzare username e password per ogni giocatore

3.2.5. Classe Carte

	ID	Requisito	Descrizione
<input checked="" type="checkbox"/>	1	Seme	Ogni carta deve avere un seme tra cuori, quadri, fiori, picche
<input checked="" type="checkbox"/>	2	Valore	Ogni carta deve avere un valore assegnato: <ul style="list-style-type: none"> • i numeri dal 7 al 10 hanno lo stesso valore del proprio numero • Jack vale 11 • Donna vale 12 • Re vale 13 • Asso vale 14 (non essendoci i numeri dal 2 al 6 l'asso conta come scala solo dopo il Re)

3.2.6. Classe GUI

	ID	Requisito	Descrizione
<input type="checkbox"/>	1	Schermata iniziale	Deve permettere il login e la registrazione di un nuovo account: <ul style="list-style-type: none"> • text field “username” • text field “password” • bottone login (eccezione account inesistente/password errata) • bottone registrati (eccezione username già esistente)
<input type="checkbox"/>	2	Schermata Menù principale	Schermata che deve permettere di poter effettuare una delle seguenti scelte, passando alla schermata corrispondente
<input type="checkbox"/>	2.1	Schermata Regolamento	Bottone che porta ad una schermata che mostra il documento che contiene le regole del gioco
<input type="checkbox"/>	2.2	Partecipa a partita	Bottone che permette di partecipare ad una partita entrando nel server

<input type="checkbox"/>	3	Schermata del tavolo da gioco	La GUI deve poter gestire la visione delle carte
<input type="checkbox"/>	3.1	Carte scoperte	<ul style="list-style-type: none"> • deve mostrare il valore della carta ed il suo seme • ogni giocatore deve poter vedere solo le proprie carte
<input type="checkbox"/>	3.2	Carte coperte	Deve visualizzare il logo JustPoker™
<input type="checkbox"/>	3.3	Bottoni per le azioni di gioco	I diversi bottoni saranno visibili solo per alcuni stati del giocatore
<input type="checkbox"/>	3.3.1	Partecipa (Invito)	Sarà visibile solo nella fase dell'invito (la puntata è fissa alla minima del tavolo, scelta dal server)
<input type="checkbox"/>	3.3.3	Apri	<ul style="list-style-type: none"> • Bottone che si attiva solo se si ha una condizione coppia di jack o superiore • Sarà visibile solo durante la fase di apertura • Al suo interno avrà una casella di testo per decidere la somma da puntare per aprire che deve rispettare i vincoli di puntata
<input type="checkbox"/>	3.3.4	Passa	<ul style="list-style-type: none"> • Bottone con cui il giocatore decide di concedere il turno al giocatore successivo • Sarà visibile solo durante la fase di apertura
<input type="checkbox"/>	3.3.5	Cambio	Bottone visibile durante la fase di accomodo con cui il giocatore può scegliere di cambiare fino a 5 delle carte presenti nella sua mano
<input type="checkbox"/>	3.3.6	Servito	Bottone visibile durante la fase di accomodo con cui il giocatore sceglie di non cambiare le proprie carte
<input type="checkbox"/>	3.3.7	Punta	<ul style="list-style-type: none"> • Bottone per puntare il credito di giocata • Visibile solo durante la fase di puntate, dopo l'accomodo • Al suo interno avrà una casella di testo per decidere la somma da puntare • Questo è il bottone che permette anche di rilanciare se si sceglie una somma maggiore dell'ultima puntata
<input type="checkbox"/>	3.3.8	Lascia	Bottone sempre visibile che permette al giocatore di lasciare la mano (anche durante la fase di invito)
<input type="checkbox"/>	3.3.9	Vedi	<ul style="list-style-type: none"> • Bottone visibile solo dopo che almeno un giocatore precedente ha effettuato una puntata • Permette di mettere in palio la stessa quantità di credito
<input type="checkbox"/>	3.3.10	Lascia la partita	Bottone visibile solo a fine mano che permette di lasciare il tavolo
<input type="checkbox"/>	3.3.11	Pronto	Bottone visibile solo a fine mano che permette di continuare verso la prossima mano

3.2.7. Classe ...

3.3. Requisiti di prestazione

- Il sistema non deve presentare difetti di programmazione che portino al crash delle partite.
 - Sono tollerate perdite di connessione client/server
- 3.4. Vincoli di progettazione
- Architettura client/server
 - **specificare i problemi e i vincoli**
 - Gestione delle dipendenze tramite Maven
 - Versionamento e gestione della configurazione tramite GitHub
 - Utilizzo di Eclipse come IDE (linguaggio Java)
 - Non utilizzare servizi esterni, solo dipendenze Maven e plugin concordati
- 3.5. Attributi del sistema software
- Come descritti nel paragrafo (1)
- 3.6. Altri requisiti

ID	Requisito	Descrizione