

# Impiccato

**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

1	Introduzione.....	3
1.1	Informazioni sul progetto.....	3
1.2	Abstract .....	3
1.3	Scopo .....	3
2	Analisi.....	5
2.1	Analisi del dominio .....	5
2.2	Analisi e specifica dei requisiti .....	5
2.3	Use case .....	9
2.4	Pianificazione .....	10
2.5	Analisi dei mezzi.....	11
2.5.1	Software .....	11
2.5.2	Hardware.....	11
3	Progettazione .....	12
3.1	Design dell'architettura del sistema .....	12
3.1.1	Package Views .....	13
3.1.2	Package Game .....	14
3.2	Design delle interfacce .....	15
4	Implementazione .....	17
4.1	Package game .....	17
4.1.1	Client.....	17
4.1.2	ClientHandler .....	18
4.1.3	ServerConnection .....	18
4.1.4	DateServer .....	19
4.1.5	Player.....	19
4.1.6	Game .....	19
4.1.7	GameHandler.....	19
4.2	Package Views.....	20
5	Test.....	21
5.1	Protocollo di test.....	21
5.2	Risultati test.....	24
5.3	Mancanze/limitazioni conosciute.....	24
6	Consuntivo.....	25
7	Conclusioni .....	27
7.1	Sviluppi futuri.....	27
7.2	Considerazioni personali.....	27
8	Bibliografia.....	27
8.1	Sitografia .....	27
9	Allegati.....	27

## **1 Introduzione**

---

### **1.1 Informazioni sul progetto**

In questo progetto sono coinvolti: Gioele Cavallo come allievo, e Guido Montalbetti come docente responsabile.

Il progetto è svolto alla CPT di Trevano, nella sezione informatica del terzo anno nella materia di laboratorio dei progetti.

Il progetto ha inizio il 16.09.2021 con termine il 23.12.2021.

### **1.2 Abstract**

When on a rainy day you can't leave the house and you have to find a way to spend time with your friends, to fight the boredom there is nothing better than to play hanged man together. But in an alternative way, sitting comfortably in front of the computer and not writing on sheets of paper.

The success of this project would allow people to spend their free time playing and having fun with their friends without having to waste sheets of paper.

### **1.3 Scopo**

Lo scopo del progetto è quello di creare un applicativo che permetta di giocare all'impiccato da computer differenti. I giocatori inoltre potranno creare le proprie partite e potranno partecipare a partite già create con l'utilizzo di un token univoco, relativo alla partita la quale si vuole giocare.



## 2 Analisi

### 2.1 Analisi del dominio

In giornate di noia tra amici bisogna trovare un buon metodo per passare il tempo. Grazie al gioco con interfaccia dell'impiccato si potrà giocare con i propri amici nella rete locale creando delle partite personali o unirsi a partite già esistenti e partecipare ad intense sessioni di gioco.

### 2.2 Analisi e specifica dei requisiti

ID: REQ-01	
<b>Nome</b>	Possibilità di creare una partita
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Possibilità di creare una partita, alla creazione viene creato un token univoco per unirsi alla partita
Sotto requisiti	
<b>001</b>	Bisogna identificarsi tramite un username

ID: REQ-02	
<b>Nome</b>	Possibilità di unirsi ad una partita
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Possibilità di unirsi ad una partita tramite il token
Sotto requisiti	
<b>001</b>	Bisogna identificarsi tramite un username
<b>002</b>	Bisogna avere la connessione al server che ospita la partita
<b>003</b>	Bisogna fornire il token della partita

ID: REQ-03	
<b>Nome</b>	Controllo nome utente
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Il nome utente deve essere al massimo di 20 caratteri e privo di volgarità

ID: REQ-04	
<b>Nome</b>	Chat broadcast
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Quando un utente scrive una lettera deve essere visualizzata in questa chat
Sotto requisiti	
<b>001</b>	Bisogna essere dentro una partita

ID: REQ-05	
<b>Nome</b>	Invio lettere
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Per inviare le lettere bisogna premere invio
Sotto requisiti	
<b>001</b>	Bisogna essere dentro una partita

ID: REQ-06	
<b>Nome</b>	Lista giocatori
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Viene mostrata una lista dei giocatori presenti, se hanno indovinato la parola viene colorato il nome in verde
Sotto requisiti	
<b>001</b>	Bisogna essere dentro una partita

ID: REQ-07	
<b>Nome</b>	Minimo di parole
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	L'applicazione deve disporre di almeno 500 parole

ID: REQ-08	
<b>Nome</b>	Fallimento round
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Il round viene perso se si fanno più di 10 errori, e l'utente deve aspettare il round successivo per giocare
Sotto requisiti	
<b>001</b>	Bisogna essere dentro una partita

ID: REQ-09	
<b>Nome</b>	Fine partita
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	La partita finisce dopo 5 round e viene mostrata la classifica dei giocatori

ID: REQ-10	
<b>Nome</b>	Gestione caratteri
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	L'applicazione deve gestire i caratteri dalla tabella UTF-0 fino all' U+00FF

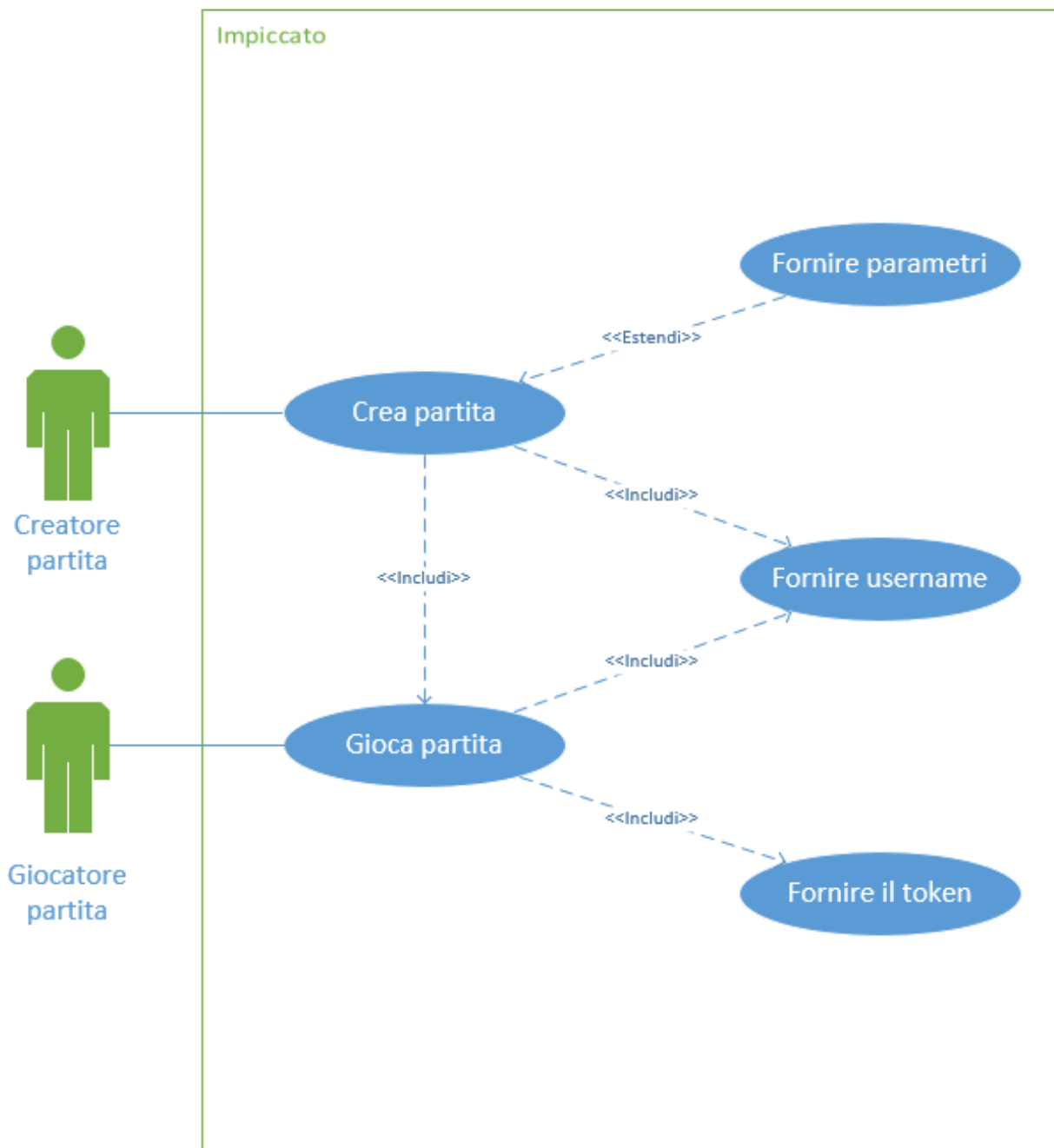
ID: REQ-11	
<b>Nome</b>	Parametrizzazione durata partita
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Alla creazione della partita si deve poter decidere il tempo massimo della durata di un round (minimo 30 sec.). L'applicazione setterà come tempo di default 30 secondi

ID: REQ-12	
<b>Nome</b>	Parametrizzazione round partita
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Alla creazione della partita si deve poter decidere il numero di round

ID: REQ-13	
<b>Nome</b>	Calcolo punteggio finale
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Deve essere calcolato il punteggio a fine partita
Sotto requisiti	
<b>001</b>	Bisogna essere dentro una partita



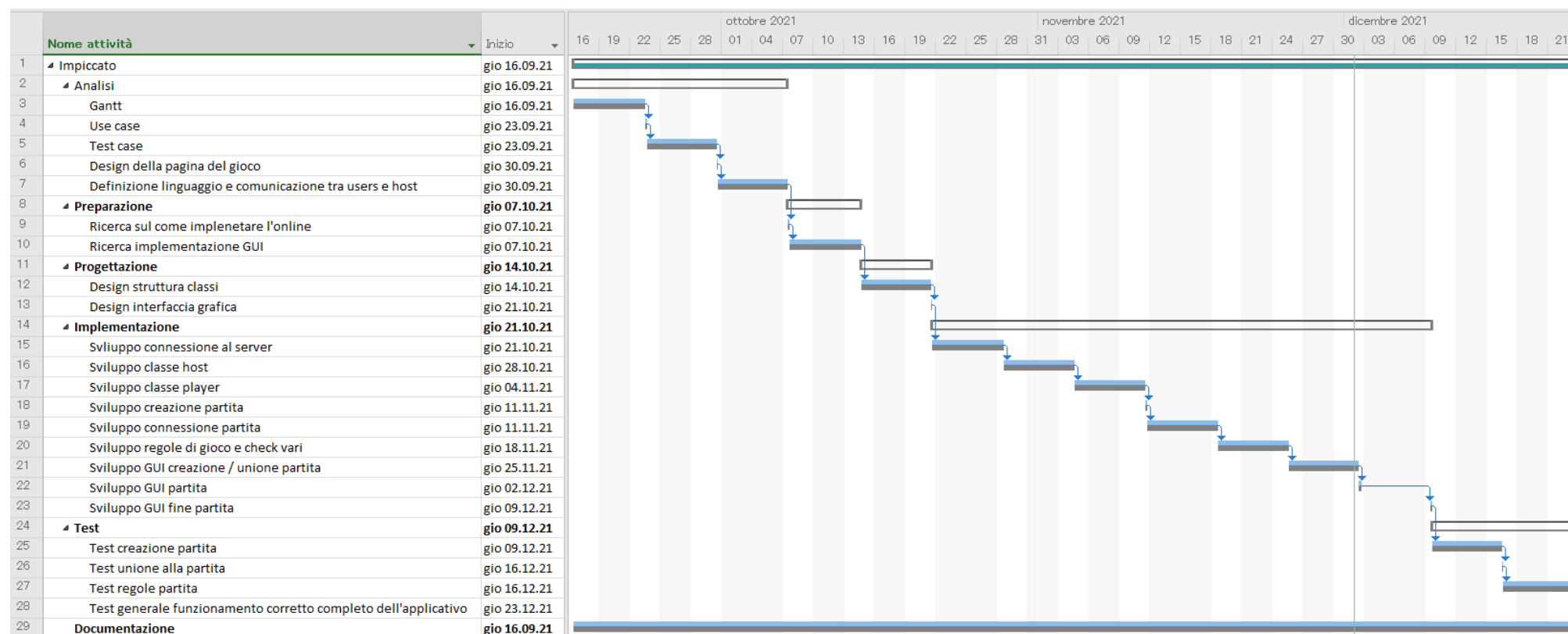
## 2.3 Use case



Ci sono due tipi di utenti:

Chi crea la partita e chi le gioca. La prima tipologia di utente è automaticamente appartenente alla seconda, perché alla creazione di una partita si viene assegnati a tale partita. Mentre la seconda tipologia di utente accede ad una partita con l'utilizzo di un token univoco.

## 2.4 Pianificazione



**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

## **2.5 Analisi dei mezzi**

### **2.5.1 Software**

- Apache NetBeans IDE 12.4
- Java 16.0.1
- javac 16.0.1
- OS: Windows 10 64 bit

### **2.5.2 Hardware**

Utilizzo un computer con:

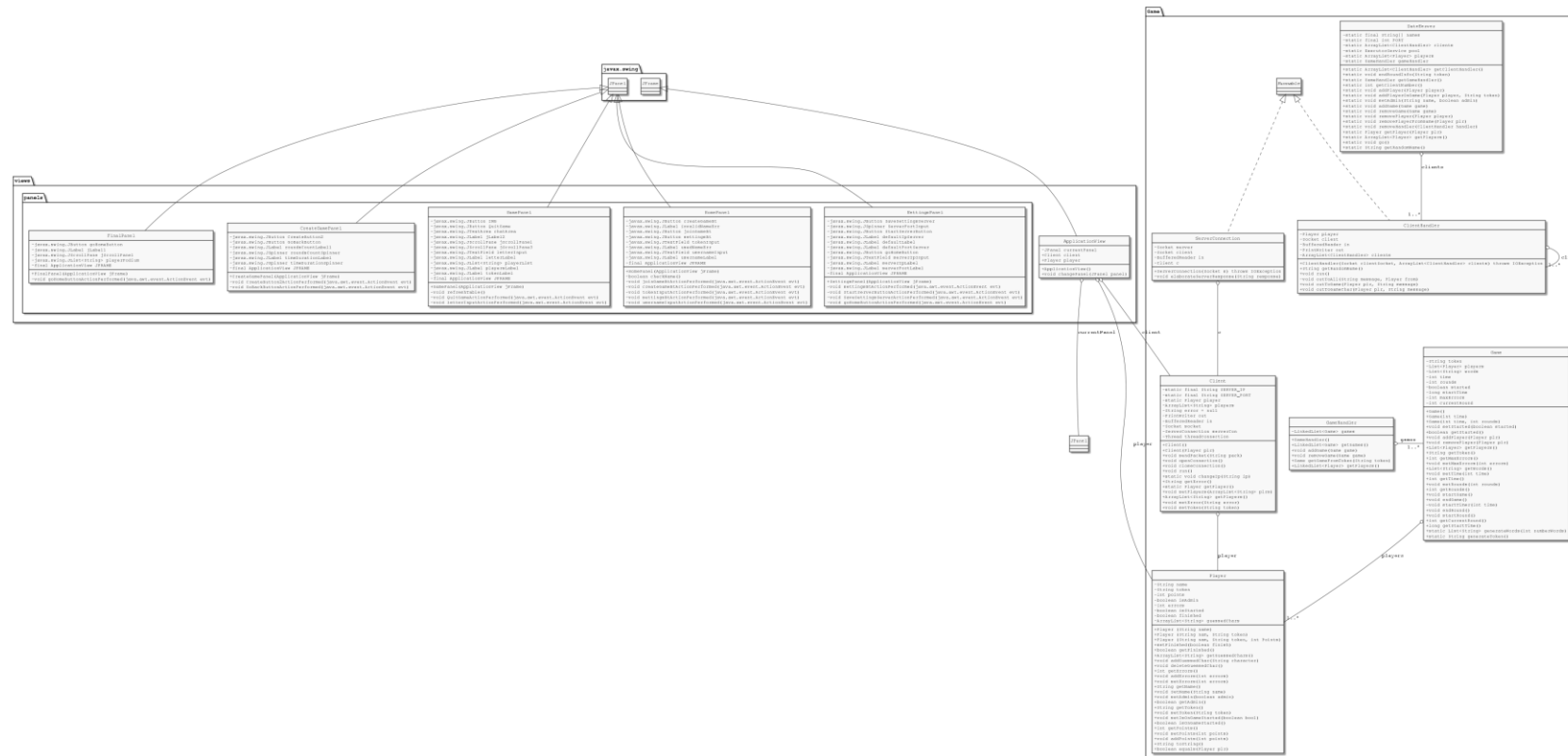
- CPU: Intel Xeon E3-1240 v5 3.5 GHz
- RAM 16 GB

**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

### 3 Progettazione

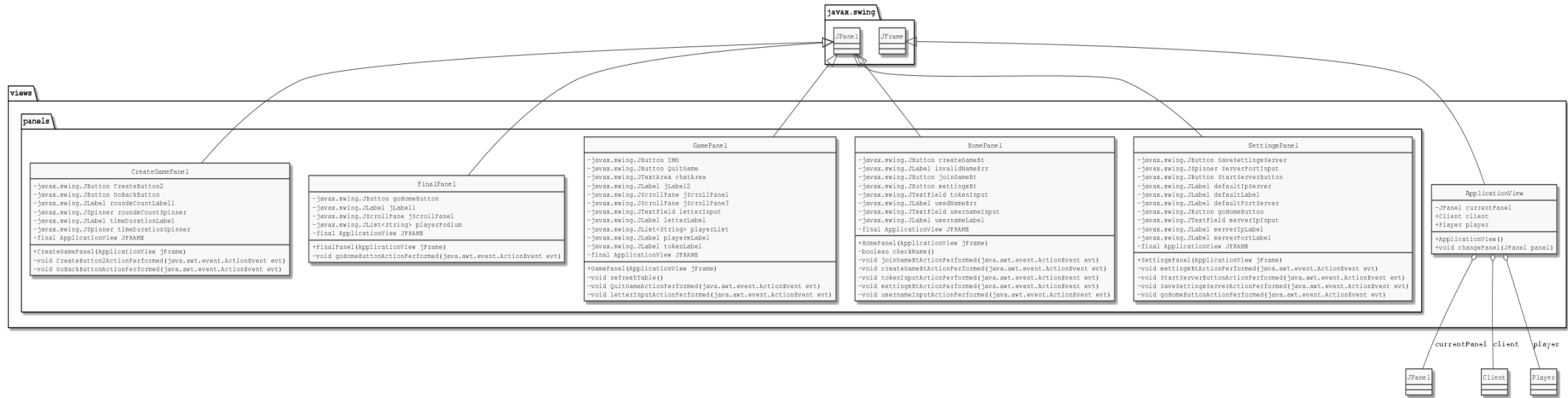
Questo capitolo descrive esaurientemente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all'esecutore di evitare fraintendimenti e imprecisioni nell'implementazione del prodotto.

#### 3.1 Design dell'architettura del sistema

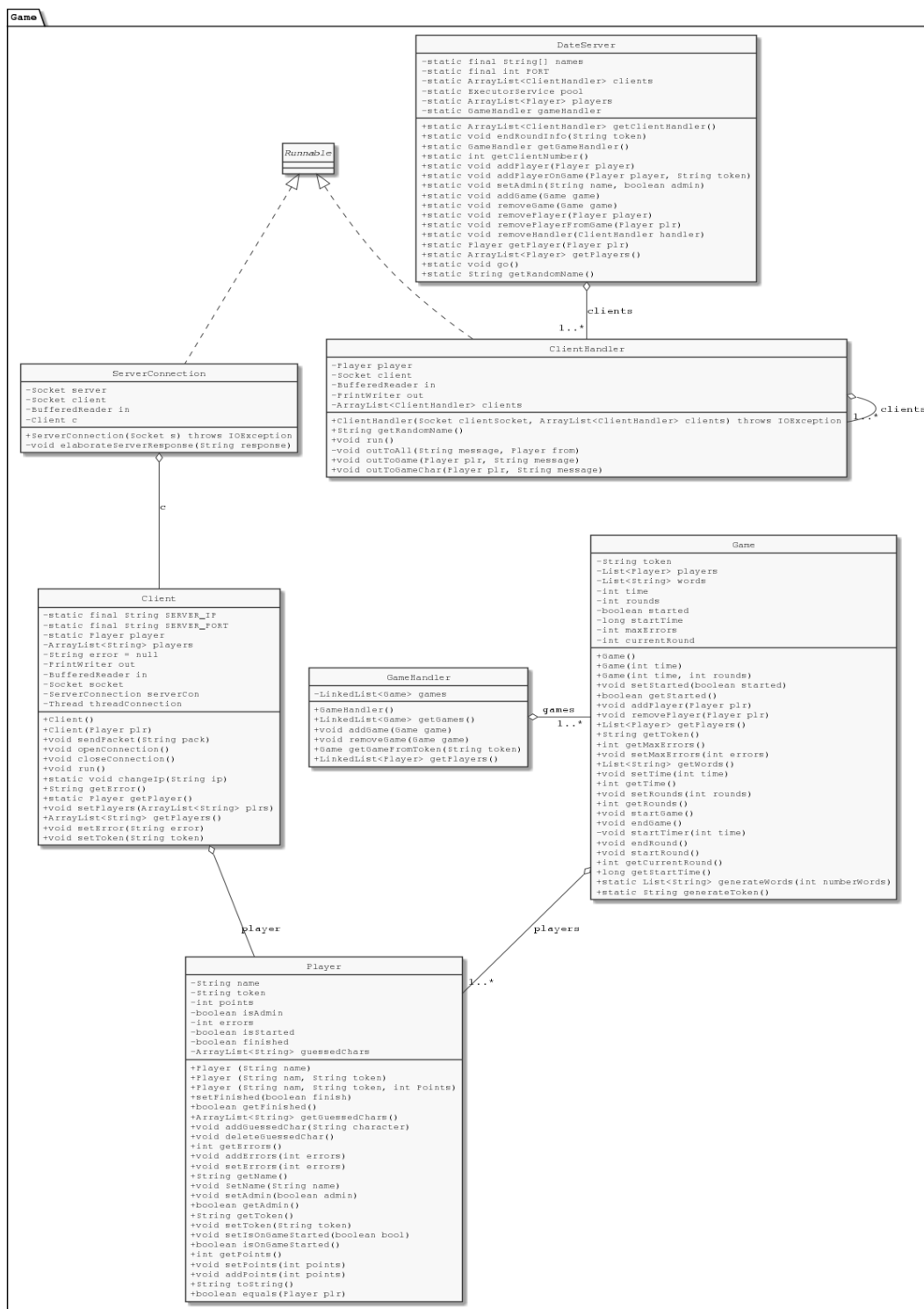


**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

### 3.1.1 Package Views



### 3.1.2 Package Game



<b>Titolo del progetto:</b>	Documentazione impiccato
<b>Alunno/a:</b>	Gioele Cavallo
<b>Classe:</b>	I3BC
<b>Anno scolastico:</b>	2021/2022
<b>Docente responsabile:</b>	Guido Montalbetti

### 3.2 Design delle interfacce

*Home Panel*

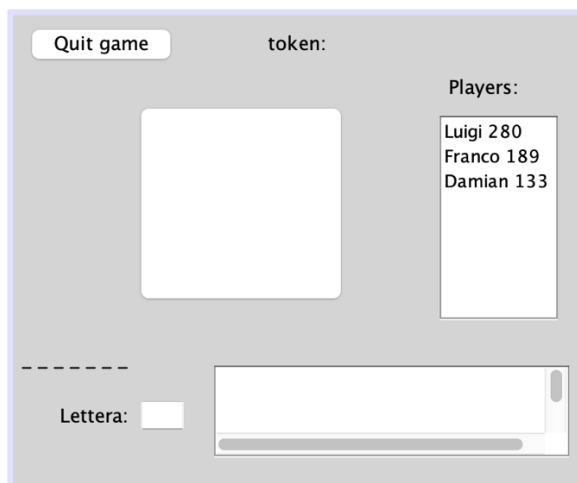
L'interfaccia Home permette di immettere l'username, e nel caso fosse già utilizzato oppure non è valido, vengono mostrati i rispettivi errori (mostrati in rosso nell'immagine soprastante).

Dopo aver immesso il proprio nome si potrà creare o giocare una partita.

Il bottone con la rotella permette di accedere alle impostazioni le quali permettono di connettersi ad un altro host che non sia quello di default (127.0.0.1:9090).

*Create Game Panel*

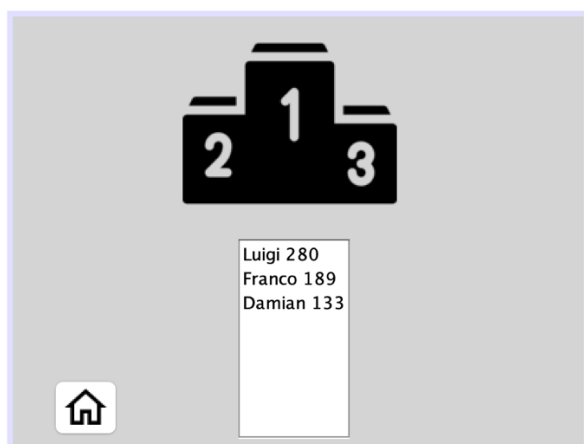
Questa interfaccia permette di creare una partita facendo inserire i numeri di rounds ed il tempo di durata per ogni round.



*Game Panel*

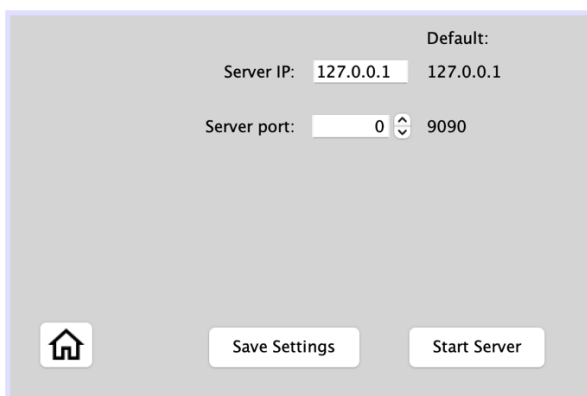
L'interfaccia della partita visualizza:

- Nel centro l'immagine dell'impiccato che cambia a dipendenza di quanti errori si fanno.
- A destra l'elenco dei giocatori nella partita affiancati dal loro punteggio.
- In basso a destra la chat nella quale appariranno le lettere dette dagli altri giocatori.
- In basso a sinistra il campo per inserire la lettera e la parola della partita censurata.
- In alto a sinistra il bottone per abbandonare la partita.



*Final Panel*

L'interfaccia della fine del gioco visualizza un immagine di un podio con sotto l'elenco, ordinato in base al punteggio, dei giocatori ed il loro punteggio.



*Settings Panel*

L'interfaccia per i settings permette di inserire l'IP e la porta del host al quale ci si vuole connettere e salvare le impostazioni.



## 4 Implementazione

### 4.1 Package game

#### 4.1.1 Client

La classe Client è quella che si occupa di mandare le richieste alla classe ClientHandler.

```
public void openConnection() throws IOException {
    // creazione del socket con il quale avverrà la connessione al server
    this.socket = new Socket(SERVER_IP, SERVER_PORT);

    // creazione della connessione al server tramite socket
    this.serverCon = new ServerConnection(this.socket, this);

    // dichiarazione del output al quale verranno mandati i pacchetti
    this.out = new PrintWriter(this.socket.getOutputStream(), true);

    // dichiarazione del input dal quale vengono presi i comandi del Client
    this.in = new BufferedReader(new InputStreamReader(this.socket.getInputStream()));

    // apertura della connessione e messa in funzione
    this.threadConnection = new Thread(this.serverCon);
    this.threadConnection.start();
}
```

Questo metodo è fondamentale per il programma, infatti apre la connessione al socket del server.

```
/**
 * @param pack : il messaggio da mandare al server
 */
public void sendPacket(String msg) {
    try {
        // viene aperta la connessione al server
        this.openConnection();

        // Viene creato il pacchetto che serve per l'identificazione del player
        String packet = "%" + player.getName() + "," + player.getToken() + "," + player.getPoints() + "%";

        // se il comando é 'quit' allora termina l'applicazione
        // chiudendo la connessione al socket.
        if (msg.toLowerCase().equals("quit")) {
            this.out.println(packet + "close");
            this.socket.close();
            System.exit(0);
            return;
        } else if (msg.toLowerCase().contains("get name")) {
            System.out.println(player.getName());
            return;
        }

        // viene mandato al printWriter il pacchetto creato
        this.out.println(packet + msg);
    } catch (IOException ioe) {
        System.out.println("Unable to send the packet");
    }
}
```

Il metodo sendPacket si occupa di mandare il messaggio passatogli al server.

<b>Titolo del progetto:</b>	Documentazione impiccato
<b>Alunno/a:</b>	Gioele Cavallo
<b>Classe:</b>	I3BC
<b>Anno scolastico:</b>	2021/2022
<b>Docente responsabile:</b>	Guido Montalbetti

#### 4.1.2 ClientHandler

Questa classe serve per interfacciarsi al server.

#### 4.1.3 ServerConnection

ServerConnection è la classe che si occupa di gestire le richieste inoltrate dal ClientHandler.

```
@Override
public void run() {
    try {
        while (true) {
            // aspetta che gli si venga mandato un messaggio
            String serverResponse = in.readLine();
            if (serverResponse == null) {
                break;
            }

            // stampa nella sua console cosa viene mandato per debugging
            System.out.println("[SERVER] " + serverResponse);

            // elabora il messaggio mandatogli
            elaborateServerResponse(serverResponse);
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    } finally {
        try {
            in.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Il metodo run in ServerConnection aspetta sempre un messaggio dal ClientHandler e ogni messaggio viene elaborato con il metodo elaborateServerResponse, che in base a cosa viene passatogli esegue determinate operazioni.

#### 4.1.4 DateServer

Questa classe è il server principale, è dove i client si collegano e dove vengono stocate le informazioni riguardanti: le partite, i giocatori, i round, ecc.

```
public static void go() throws IOException {
    System.out.println("Welcome to the server");

    // apre la porta alla quale si collegheranno i client
    ServerSocket listener = new ServerSocket(PORT);

    while (true) {
        try {
            // aspetta che si colleghi un client
            System.out.println("[SERVER] Waiting for client connection...");

            Socket client = listener.accept();
            System.out.println("[SERVER] Client connected.");

            // crea un ClientHandler per il client appena connesso
            ClientHandler clientThread = new ClientHandler(client, clients);
            clients.add(clientThread);
            try {
                pool.execute(clientThread);
            } catch (IllegalArgumentException iae) {
            }

            // debugging per sapere quante connessioni ci sono
            System.out.println("connections: " + clients.size());
        } catch (IllegalArgumentException iae) {
        }
    }
}
```

Il metodo go è il metodo principale della classe DateServer, perché permette l'esecuzione del server. Quindi crea il server e memorizza le connessioni.

#### 4.1.5 Player

La classe Player è quella utilizzata per creare dei giocatori, con i relativi attributi.

#### 4.1.6 Game

Questa classe riguarda le informazioni delle partite, memorizza i player nella partita ed il token, con tutte le impostazioni relative alla partita.

#### 4.1.7 GameHandler

Questa classe gestisce tutte le partite create nel gioco. Grazie a questa classe la gestione delle partite è semplificata.

## 4.2 Package Views

Questo package serve per gestire la grafica.

La classe **ApplicationView** Si occupa di contenere i diversi tipo di panels da mostrare al suo interno.

```
/**
 * @param panel : il nuovo pannello da mostrare
 */
public void changePanel(JPanel panel){
    if(panel instanceof CreateGamePanel){
        panel = new CreateGamePanel(this);
    }else if(panel instanceof HomePanel){
        panel = new HomePanel(this);
    }else if(panel instanceof FinalPanel){
        panel = new FinalPanel(this);
    }else if(panel instanceof GamePanel){
        panel = new GamePanel(this);
    }else if(panel instanceof SettingsPanel){
        panel = new SettingsPanel(this);
    }
    panel.setBounds(0, 0, 400, 300);
    this.getContentPane().remove(this.currentPanel);
    this.getContentPane().add(panel);
    this.currentPanel = panel;

    this.invalidate();
    this.validate();
    this.repaint();
}
```

Questo metodo viene adoperato dalla classe ApplicationView per cambiare il JPanel corrente.

## 5 Test

### 5.1 Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l'adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Possibilità di creare una partita
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Deve esserci la possibilità di creare una partita		
<b>Prerequisiti:</b>	Essere connessi al server ed avere un nome		
<b>Procedura:</b>	Una volta inserito il nome utente premere "Crea Partita"		
<b>Risultati attesi:</b>	Si apre la schermata del gioco e viene generato il token della partita		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Possibilità di unirsi ad una partita
<b>Riferimento:</b>	REQ-02		
<b>Descrizione:</b>	Deve esserci la possibilità di unirsi ad una partita		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere connessi al server</li> <li>Avere un nome</li> <li>Avere il token della partita</li> </ul>		
<b>Procedura:</b>	Una volta inserito il nome utente ed il token della partita e premere "Unisciti"		
<b>Risultati attesi:</b>	Si apre la schermata del gioco		

<b>Test Case:</b>	TC-003	<b>Nome:</b>	Controllo nome utente
<b>Riferimento:</b>	REQ-03		
<b>Descrizione:</b>	Il nome utente deve essere controllato dall'applicativo		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	Inserire il nome utente e premere "Crea Partita"		
<b>Risultati attesi:</b>	Se il nome è maggiore di 20 caratteri o contiene delle parole proibite appare un messaggio di errore. Altrimenti creerà una partita		

<b>Test Case:</b>	TC-004	<b>Nome:</b>	Chat broadcast
<b>Riferimento:</b>	REQ-04		
<b>Descrizione:</b>	Deve esserci la possibilità di scrivere in una chat condivisa da tutti i partecipanti della partita		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Entrare dentro una partita con 2 utenti differenti e scrivere qualcosa nella chat		
<b>Risultati attesi:</b>	L'altro utente deve poter vedere cosa è stato scritto nella chat dal primo utente		

<b>Test Case:</b>	TC-005	<b>Nome:</b>	Invio lettere
<b>Riferimento:</b>	REQ-05		
<b>Descrizione:</b>	Deve esserci la possibilità di inviare le lettere tramite il tasto “invio” della tastiera		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Partecipare ad una partita e scrivere una lettera, poi premere invio		
<b>Risultati attesi:</b>	si deve vedere se la lettera inviata è giusta oppure sbagliata. Se arriva questa conferma significa che la lettera è arrivata al server.		

<b>Test Case:</b>	TC-006	<b>Nome:</b>	Lista giocatori
<b>Riferimento:</b>	REQ-06		
<b>Descrizione:</b>	Deve esserci una lista di giocatori, se viene indovinata la parola il nome del giocatore viene colorato di verde		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Entrare dentro ad una partita ed indovinare la parola		
<b>Risultati attesi:</b>	Una volta indovinata il nome nella lista viene colorato di verde		

<b>Test Case:</b>	TC-007	<b>Nome:</b>	Minimo di parole
<b>Riferimento:</b>	REQ-07		
<b>Descrizione:</b>	Devono esserci 500 parole disponibili nel gioco		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	Creo un ciclo che percorre il file .txt sul quale sono salvate le parole e controllo che siano più di 500 utilizzando un contatore		
<b>Risultati attesi:</b>	Il contatore deve segnare almeno 500 parole		

<b>Test Case:</b>	TC-008	<b>Nome:</b>	Fallimento round
<b>Riferimento:</b>	REQ-08		
<b>Descrizione:</b>	Il round deve poter finire al superamento dei 10 errori. l'utente dovrà poi aspettare il prossimo round per giocare nuovamente		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Entrare dentro una partita e sbagliare più di 10 volte la lettera		
<b>Risultati attesi:</b>	Viene bloccata la possibilità di scrivere e si deve poter visualizzare un messaggio che informa il fallimento del round		

<b>Test Case:</b>	TC-009	<b>Nome:</b>	Fine partita
<b>Riferimento:</b>	REQ-09		
<b>Descrizione:</b>	La partita termina al concludersi dei 5 round		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Entrare dentro una partita e concludere i 5 round		
<b>Risultati attesi:</b>	Verrà visualizzato un messaggio che informa del concludersi della partita e viene visualizzata la classifica dei giocatori		

<b>Test Case:</b>	TC-010	<b>Nome:</b>	Gestione caratteri
<b>Riferimento:</b>	REQ-10		
<b>Descrizione:</b>	l'applicazione deve poter gestire i caratteri dalla tabella UTF-0 fino all' U+00F		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	Entrare dentro una partita e scrivere i caratteri speciali per verificarne la corretta visualizzazione		
<b>Risultati attesi:</b>	Bisogna poter visualizzare correttamente il carattere scritto dentro la chat		

<b>Test Case:</b>	TC-011	<b>Nome:</b>	Parametrizzazione durata partita
<b>Riferimento:</b>	REQ-11		
<b>Descrizione:</b>	Deve esserci la possibilità di decidere alla creazione della partita quanto far durare un round		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Poter creare una partita</li> </ul>		
<b>Procedura:</b>	Creare una partita e decidere un tempo. Una volta fatto partire il round aspettare il tempo deciso come durata dei round.		
<b>Risultati attesi:</b>	Una volta passato il tempo prestabilito si passerà ad un round successivo.		

<b>Test Case:</b>	TC-012	<b>Nome:</b>	Parametrizzazione round partita
<b>Riferimento:</b>	REQ-12		
<b>Descrizione:</b>	Deve esserci la possibilità di decidere alla creazione della partita di quanti round è composto il gioco		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Poter creare una partita</li> </ul>		
<b>Procedura:</b>	Creare una partita e decidere il numero di round. Ora giocare il numero di round e verificare se sono esattamente quelli impostati		
<b>Risultati attesi:</b>	Una volta passati il numero di round prestabiliti deve essere visualizzata la schermata della fine del gioco.		

<b>Test Case:</b>	TC-013	<b>Nome:</b>	Calcolo punteggio finale
<b>Riferimento:</b>	REQ-13		
<b>Descrizione:</b>	Il punteggio deve essere calcolato correttamente		
<b>Prerequisiti:</b>	<ul style="list-style-type: none"> <li>Essere dentro ad una partita</li> </ul>		
<b>Procedura:</b>	<p>Entrare dentro una partita e finire il gioco. Alla fine della partita verificare se il punteggio coincide con ciò che è stato fatto.</p> <p>Calcolo per 1 round: durataRound – secondiIndovinataParola</p> <p>Calcolo finale: Somma dei punti di ogni round</p>		
<b>Risultati attesi:</b>	Deve essere visualizzato il risultato esatto dei punti fatti a fine partita.		

## 5.2 Risultati test

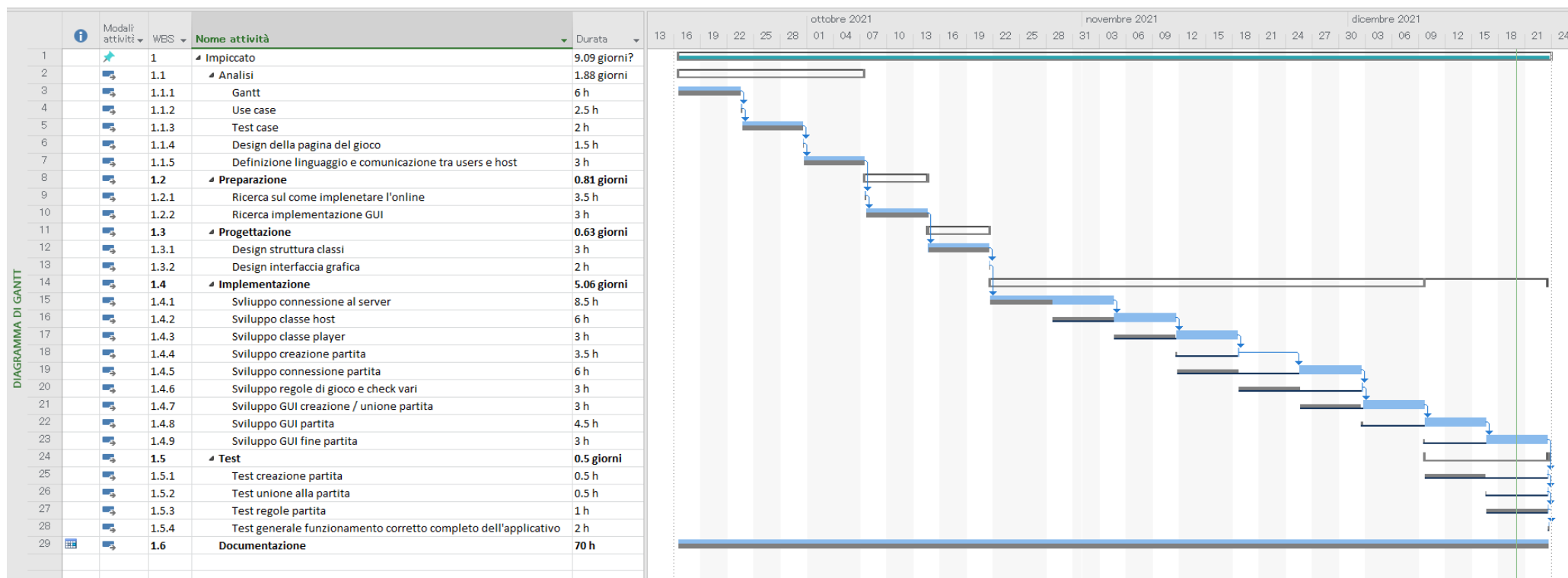
Test Case	Risultato	Commento	Data
TC-001	OK	Il player crea la partita e gli viene assegnato un token uguale a quello della partita	18.11.2021
TC-002	OK	Il server quando stampa i giocatori notifica che il giocatore è presente nella partita	18.11.2021
TC-003	OK	L'utente se il nome che inserisce non rispetta i criteri prestabiliti viene notificato	16.12.2021
TC-004	NO OK	Non è stata sviluppata	16.12.2021
TC-005	OK	Il server stampa la lettera che il player ha inviato nella console	25.11.2021
TC-006	NO OK	Non è stata sviluppata	16.12.2021
TC-007	OK	Il file dal quale prendo le parole è di 6000 righe	25.11.2021
TC-008	NO OK	Non potendo ancora startare la partita non posso verificare che il round fallisca	16.12.2021
TC-009	NO OK	Non potendo ancora startare la partita non posso verificare che essa finisca	16.12.2021
TC-010	OK	Il server scrive i caratteri corretti che gli vengono passati nella console	25.11.2021
TC-11	OK	Il server al cambiamento della durata dei round della partita, notifica nella console il cambiamento con i nuovi valori	11.12.2021
TC-012	OK	Il server al cambiamento dei round della partita, notifica nella console il cambiamento con i nuovi valori	11.12.2021
TC-013	NO OK	Non potendo finire una partita non posso verificare che calcoli correttamente il punteggio	16.12.2021

## 5.3 Mancanze/limitazioni conosciute

Il tempo quale è venuto a mancare a fatto in modo che lo sviluppo dell'applicativo non venisse terminato.



## 6 Consuntivo



**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

**Titolo del progetto:** Documentazione impiccato  
**Alunno/a:** Gioele Cavallo  
**Classe:** I3BC  
**Anno scolastico:** 2021/2022  
**Docente responsabile:** Guido Montalbetti

## 7 Conclusioni

---

L'applicativo, anche se non completamente funzionante, ha delle basi abbastanza solide per chi in un futuro volesse finirne l'implementazione. Questo progetto è stato svolto seguendo la linea guida dei requisiti emessi dal cliente così da poter avere un applicativo che simuli il gioco dell'impiccato online.

### 7.1 Sviluppi futuri

In futuro, oltre a rendere funzionante il prodotto, si potrebbero trasmettere gli eventi avvenuti tra il client ed il server tramite le interfacce di Java, le quali renderebbero lo scambio di informazioni molto più semplice. Un altro sviluppo potrebbe essere quello di renderlo multiplatforma, così da potersi collegare tramite browser al server e giocare anche con chi ha la versione eseguibile del gioco. Questo aprirebbe le porte ad un'implementazione di un database nel progetto, così da permettere un login agli utenti per fare in modo di salvare i propri progressi e avere una classifica mondiale.

### 7.2 Considerazioni personali

Nonostante il progetto non sia ultimato, sono molto soddisfatto di quello che ho imparato. Mi sono affacciato alle connessioni tramite socket in java, il che ha rafforzato le mie conoscenze specifiche del linguaggio di programmazione. Mi sono dovuto mettere in gioco per trovare soluzioni a problemi sempre più ricorrenti, questo mi ha permesso di sfruttare al massimo le mie conoscenze di programmazione. Anche se sono rammaricato del non completamento di questo progetto sono comunque soddisfatto delle conoscenze che ho acquisito e delle conoscenze rafforzate.

## 8 Bibliografia

---

### 8.1 Sitografia

- <https://www.youtube.com/watch?v=BWjGQllkgT4>, *Java socket programming Part 1*, 09-09-2021
- <https://www.baeldung.com/a-guide-to-java-sockets> *A Guide to Java Sockets*, 09-09-2021
- [https://github.com/napolux/paroleitaliane/blob/master/paroleitaliane/60000\\_parole\\_italiane.txt](https://github.com/napolux/paroleitaliane/blob/master/paroleitaliane/60000_parole_italiane.txt), *6000 parole italiane*, 16-11-2021

## 9 Allegati

---

Elenco degli allegati, esempio:

- Diari di lavoro
- Codici sorgente/documentazione
- Manuale d'uso per l'utente
- Codice sorgente presente su GitHub
  - <https://github.com/GioeleCavallo/impiccato>