

/* Suddivisione dei ruoli */

Ruben

// Tester

Francesco

// Software Architect e assistente Programmatore



Ivan

/* Programmatore e
assistente
Software Architect
*/

Alessio

// SCRUM Master

Mirco

// Presentatore Grafico del progetto

/* Linguaggio di programmazione e Brainstorming */

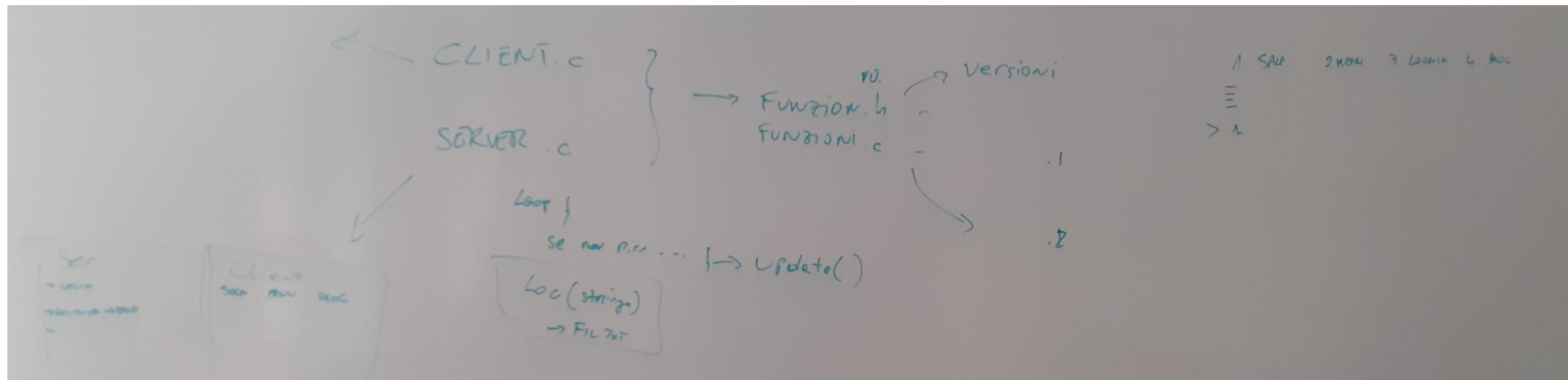
Come linguaggio di programmazione, è stato scelto **C** in quanto conosciuto da tutti i membri del gruppo.

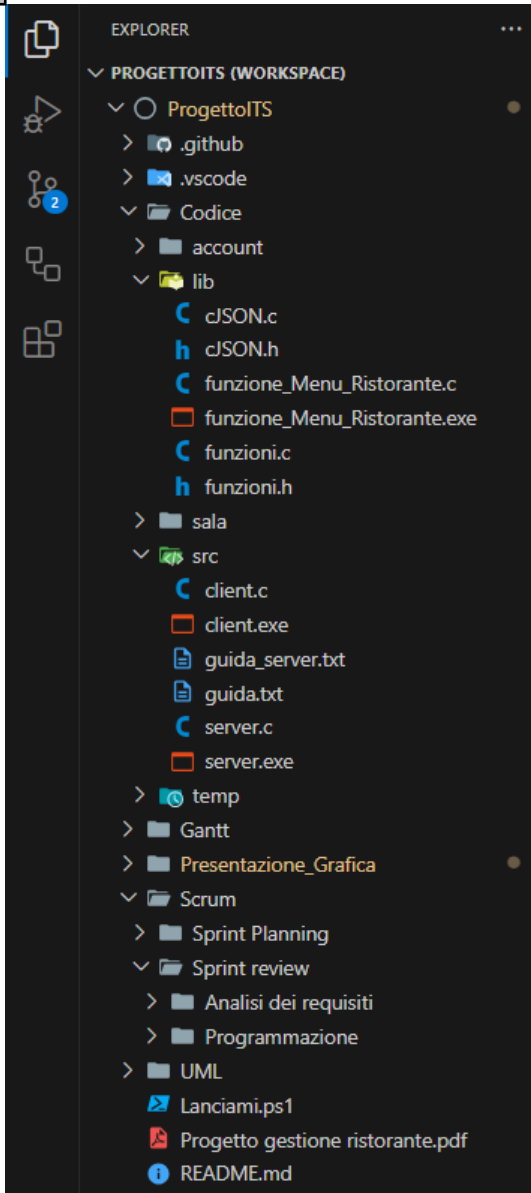
L'architetto del software ha suggerito un'**architettura Client / Server**, che interagiscono tra di loro tramite file temporanei.



```
#include <stdio.h>

void main()
{
    printf("La Gang del Bosco");
}
```





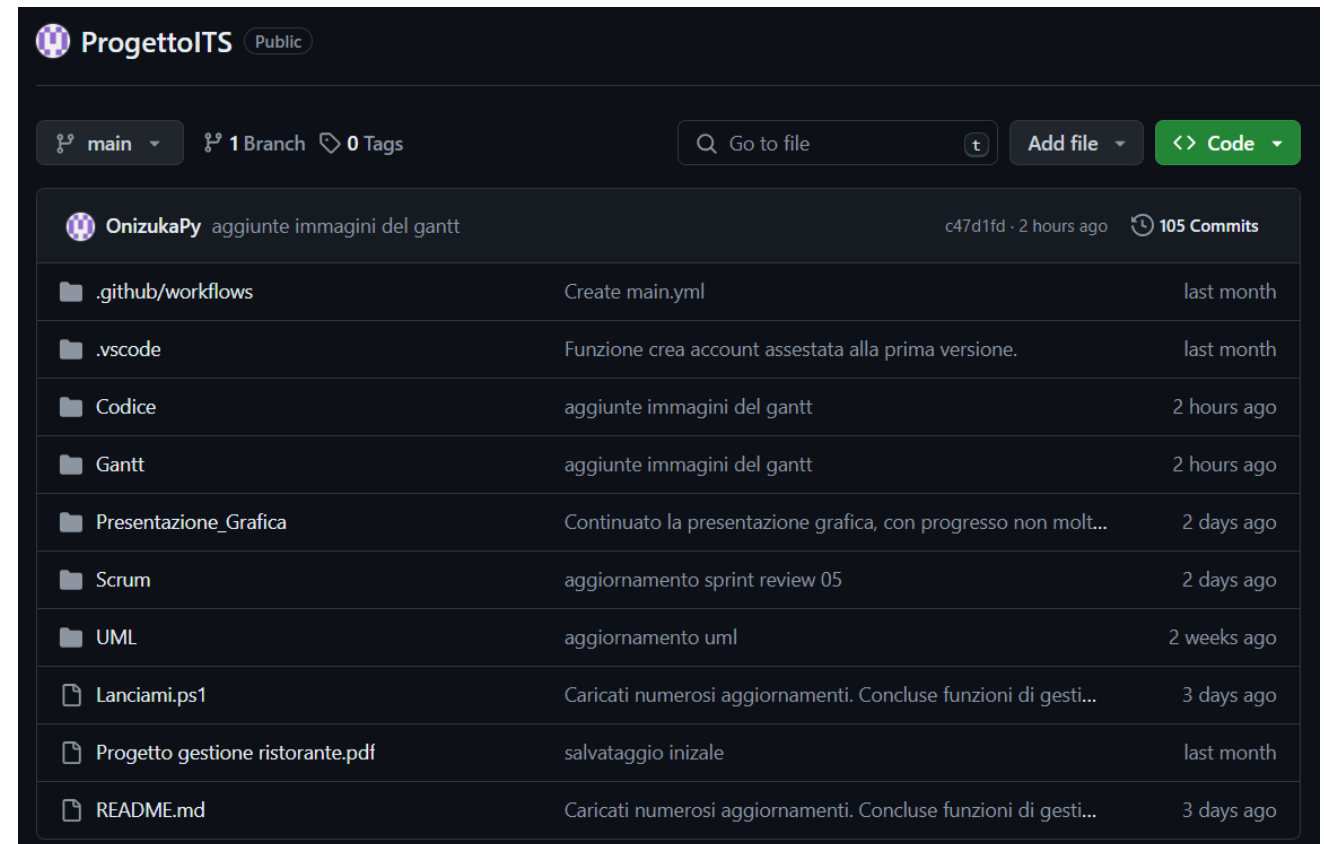
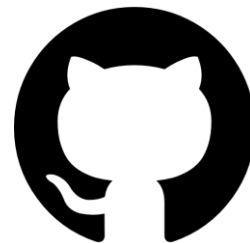
/* Ambiente di lavoro */

Per poter lavorare, il gruppo ha effettuato la sincronizzazione degli strumenti, installando **Visual Studio Code** come ambiente di sviluppo e collegandovi **GitHub** per committare gli aggiornamenti.

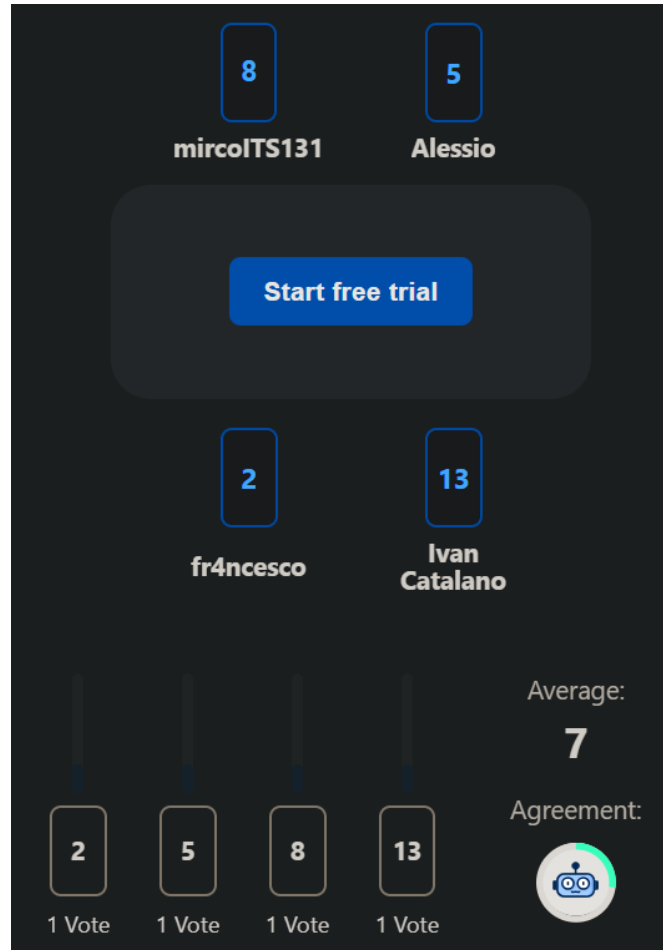


Visual
Studio
Code

GitHub



/* Analisi dei Requisiti */



// SCRUM: Planning Poker

Valore attribuito alla realizzazione di una funzione di difficoltà media = 1

Backlog

Calcolo delle stime delle singole task

Realizzazione dei diagrammi UML

Gestione utenti

Gestione dei Menù

Disposizione sala

Prenotazione dei tavoli

Gestione Ordini

Feedback clienti

Punteggio (ore di lavoro)

1

3,7

4,3

4

3,3

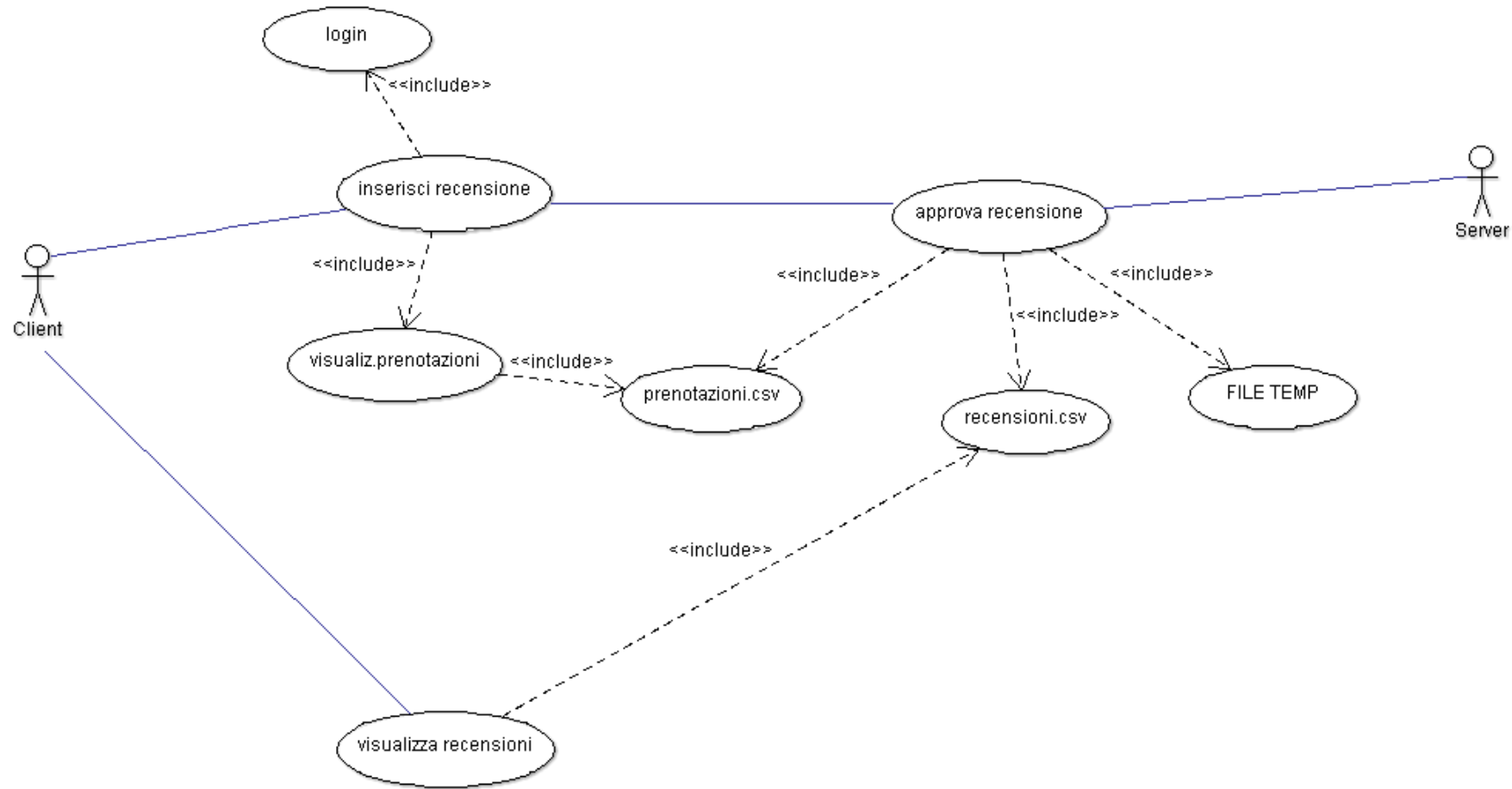
4,3

15,3

7

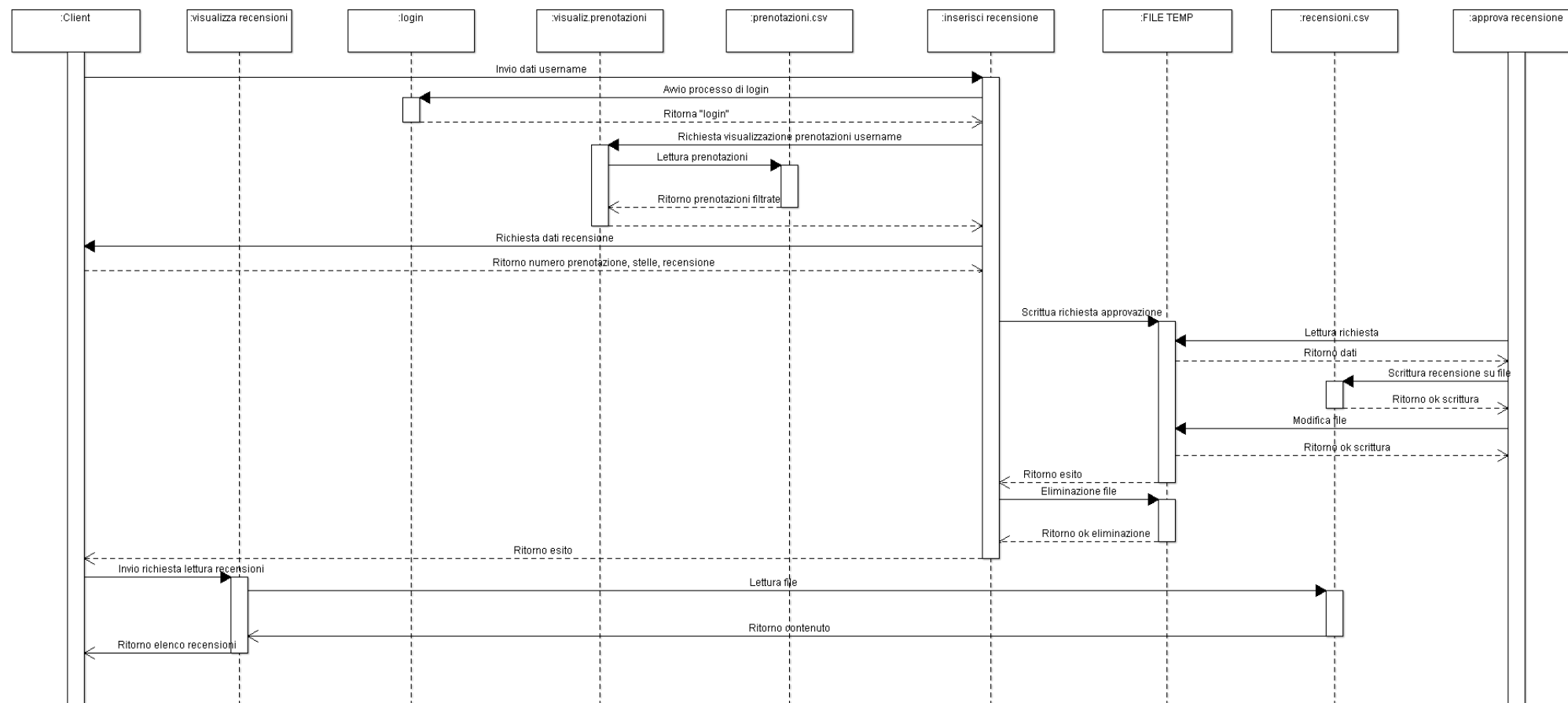
/* Analisi dei Requisiti - Esempi di diagrammi UML */

// Diagramma di Casi d'uso delle recensioni

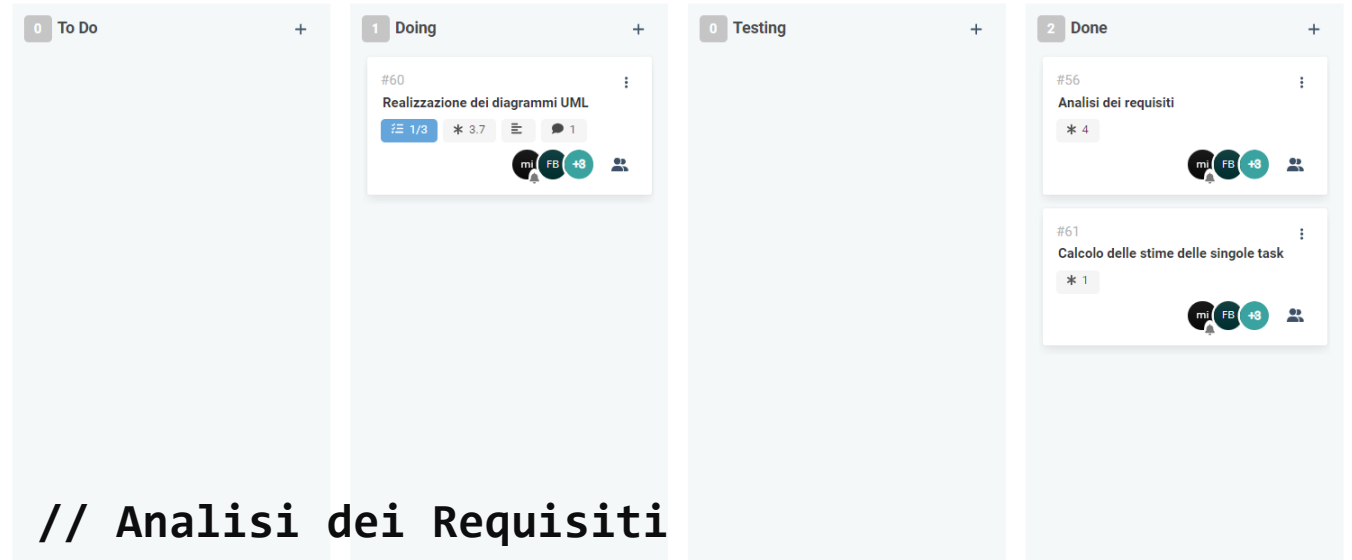
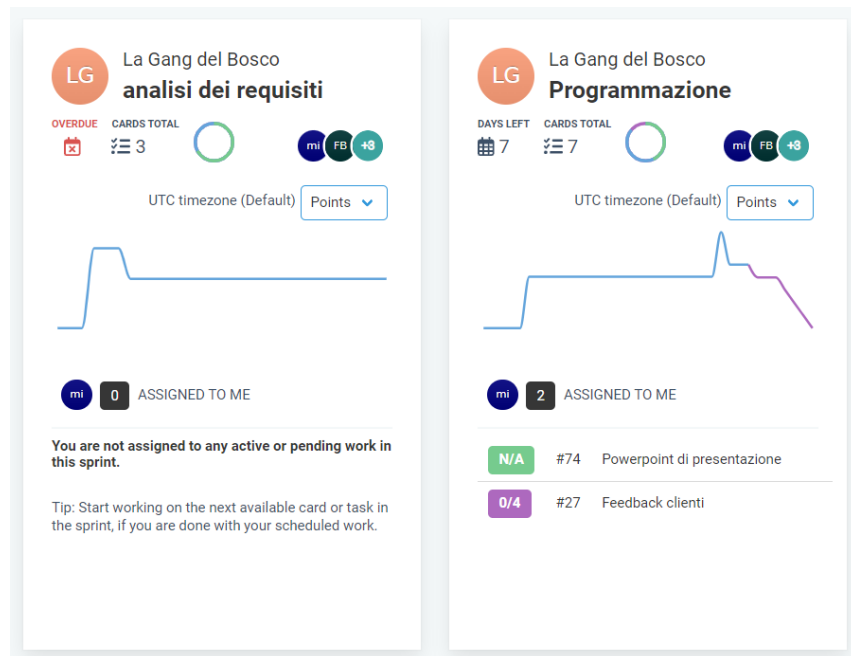


/* Analisi dei Requisiti - Esempi di diagrammi UML */

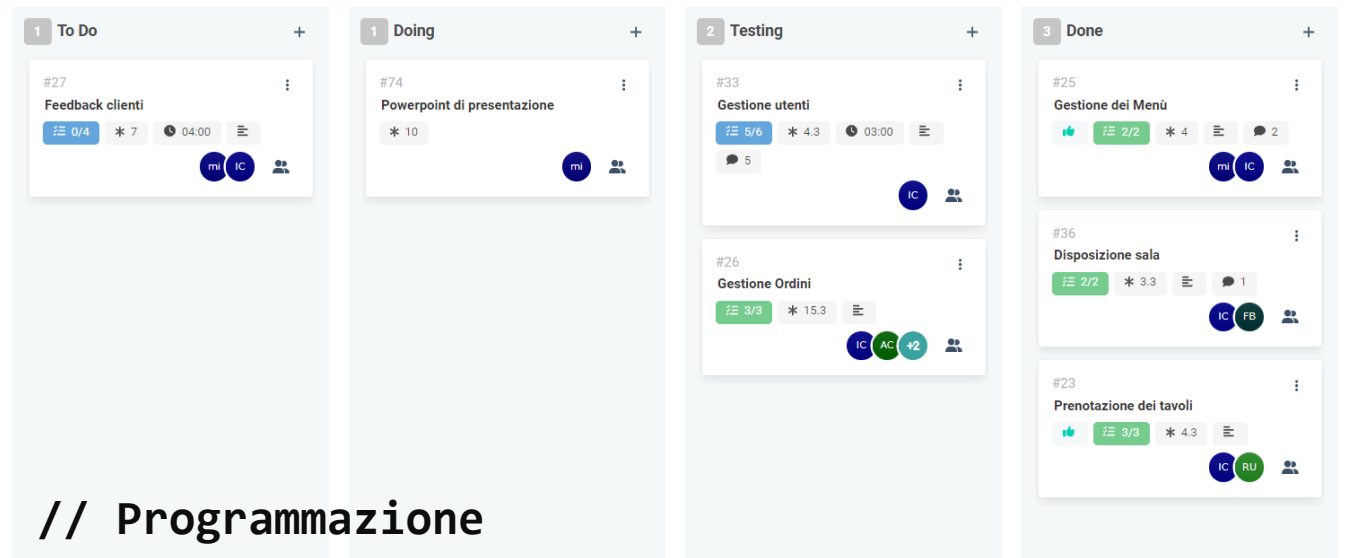
// Diagramma di Sequenza delle recensioni



/* Sprint Review in FlyingDonut */



// Analisi dei Requisiti



// Programmazione

/* Programmazione */

// Porzione di codice riguardante la funzione di visualizzazione delle prenotazioni.

```
937 // funzione per visualizzare le prenotazioni di un utente
938 void visualizza_prenotazioni(char* username, char *path){
939
940     // leggiamo il file delle prenotazioni.csv nella cartella path_sala
941     // facciamo un ciclo per leggere le righe del file
942     FILE *file = fopen(path, "r");
943     if(file == NULL){
944         printf("Errore nell'apertura del file\n");
945         return;
946     } else {
947         bool trovato = false;
948         // leggiamo il file e carichiamo ogni riga in una struttura Prenotazione.
949         char riga[MAX_LUNG_PORTATA];
950         // stampiamo le colonne della prenotazione
951         printf("          \tNum\tUser\tData\t\tTavolo\tPosti\n");
952         while(fgets(riga, MAX_PORTATE, file) != NULL){
953             // carichiamo la riga come tipo Prenotazione
954             Prenotazione pren = carica_prenotazioni(riga);
955             // se l'username è uguale a quello passato eliminiamo la prenotazione
956             if (strcmp(pren.username, username) == 0){
957                 printf("Prenotazione \t%d\t%s\t%s\t%d\t%d\n", pren.codice, pren.username, pren.data, pren.tavolo, pren.posti);
958                 trovato = true;
959             }
960         }
961         if (!trovato){
962             printf("Nessuna prenotazione trovata\n");
963         }
964         fclose(file);
965     }
966 }
```

/* Programmazione */

// Porzione di terminale riguardante il login e la visualizzazione della sala.
// Sono da fare ulteriori migliorie ai messaggi di errore.

```
[ Client@LaGangDelBosco.it ]: > ./client.exe -v icatalano
Verifica se l'account e' loggato
Inserisci la password: ****
In attesa di risposta
In attesa di risposta
2024-05-23 12:47:25: Account connesso
=====
Account loggato
-----
Dati dell'account:
Nome: Ivan
Cognome: Catalano
Username: icatalano
Email: icatalano@mail.it
Password: ****
Account approvato
-----
Prenotazioni effettuate:

Errore nell'apertura del file
-----
Ordini effettuati:

Errore nell'apertura del file
-----
=====
```

```
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala
Questo comando non ti e' permesso
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala 01/01/2025
In data 01/01/2025 non ci sono prenotazioni.
[ Client@LaGangDelBosco.it ]: > ./client.exe -sala template
Formato data non valido. Inserire la data nel formato gg/mm/aaaa
[ Client@LaGangDelBosco.it ]: > ./client.exe -p_tavolo icatalano 23/05/2024
Controllo se l'utente e' loggato
L'account esiste. Procediamo con la verifica del Login
Account gia' loggato
Non ci sono prenotazioni in data 23/05/2024, la sala e' libera
La disposizione della sala e' la seguente:
La sala non e' piena
Il numero di tavoli e': 4
Il tavolo 1 e' libero, e i posti disponibili sono: 4
Il tavolo 2 e' libero, e i posti disponibili sono: 4
Il tavolo 3 e' libero, e i posti disponibili sono: 4
Il tavolo 4 e' libero, e i posti disponibili sono: 4
Il numero di posti liberi totali e': 16
Tavolo 1:
+++++
+ @ @ @ @ +
+++++

Tavolo 2:
+++++
+ @ @ @ @ +
+++++
```

/* Programmazione */

// Esempio di baco riscontrato durante la fase di programmazione.

```
PS C:\Users\lenov\Documents\ProgettoGruppo801\ProgettoITS-1\Codice\src> ./client.exe -c
Inserisci il tuo nome: Tom
Inserisci il tuo cognome: Tom
Inserisci il tuo username: Tom
Inserisci la tua password: ***
Inserisci la tua email: Tom
Account in attesa di approvazione
Account in attesa di approvazione
2024-05-21 15:02:33: Account approvato
PS C:\Users\lenov\Documents\ProgettoGruppo801\ProgettoITS-1\Codice\src> ./client.exe -v Cate
Verifica se l'account e' loggato
Inserisci la password:
In attesa di risposta
In attesa di risposta
In attesa di risposta
2024-05-21 15:02:50: Account rifiutato
Ripeti l'operazione di login
=====
Dati dell'account:
Nome: Cate
Cognome: Cate
Username: Cate
Email: Cate
Password: ****
Account approvato
=====
Prenotazioni effettuate:


|              | Num | User | Data       | Tavolo | Posti |
|--------------|-----|------|------------|--------|-------|
| Prenotazione | 2   | Cate | 23_05_2024 | 2      | 4     |


=====
Ordini effettuati:


|                              | Cod | Pren | Data | User | Conto | Pagato |
|------------------------------|-----|------|------|------|-------|--------|
| Nessuna prenotazione trovata |     |      |      |      |       |        |


=====
PS C:\Users\lenov\Documents\ProgettoGruppo801\ProgettoITS-1\Codice\src> ./client.exe -e tavolo 2
Numero di prenotazioni: 3
Prenotazione 2 in eliminazione
In attesa di risposta
In attesa di risposta
In attesa di risposta
Prenotazione eliminata
PS C:\Users\lenov\Documents\ProgettoGruppo801\ProgettoITS-1\Codice\src> ./client.exe -e Cate
L'account esiste
Verifica se l'account e' loggato
Inserisci la password: ****
In attesa di risposta
```

Se si era loggati con un utente e si voleva visualizzare il profilo di un altro utente, nell'inserimento della password era solo necessario premere INVIO, senza dover digitare alcun carattere, ed il programma permetteva comunque di accedere al profilo dell'altro utente.

Un altro baco simile: se si era loggati con un utente e si voleva prenotare il tavolo ad un'altra persona, innanzitutto quella persona deve avere un account, poi se si premeva INVIO senza loggare al suo account quando si inseriva la password, era comunque possibile prenotare un tavolo a nome di quell'altra persona.

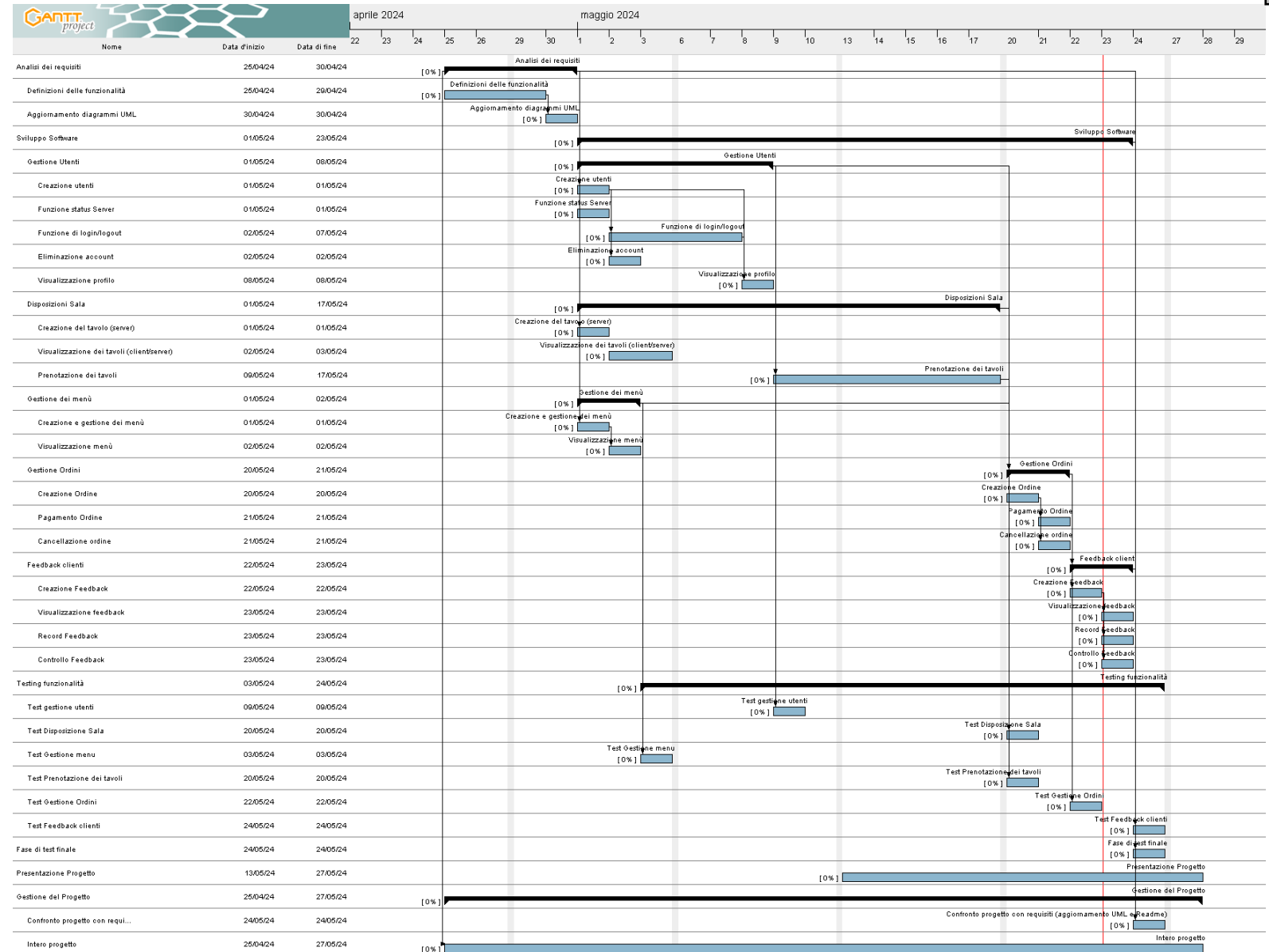
In sostanza, in entrambi i casi era possibile passare alla procedura successiva semplicemente premendo INVIO, senza dover inserire alcun carattere.

Diagrammi GANTT

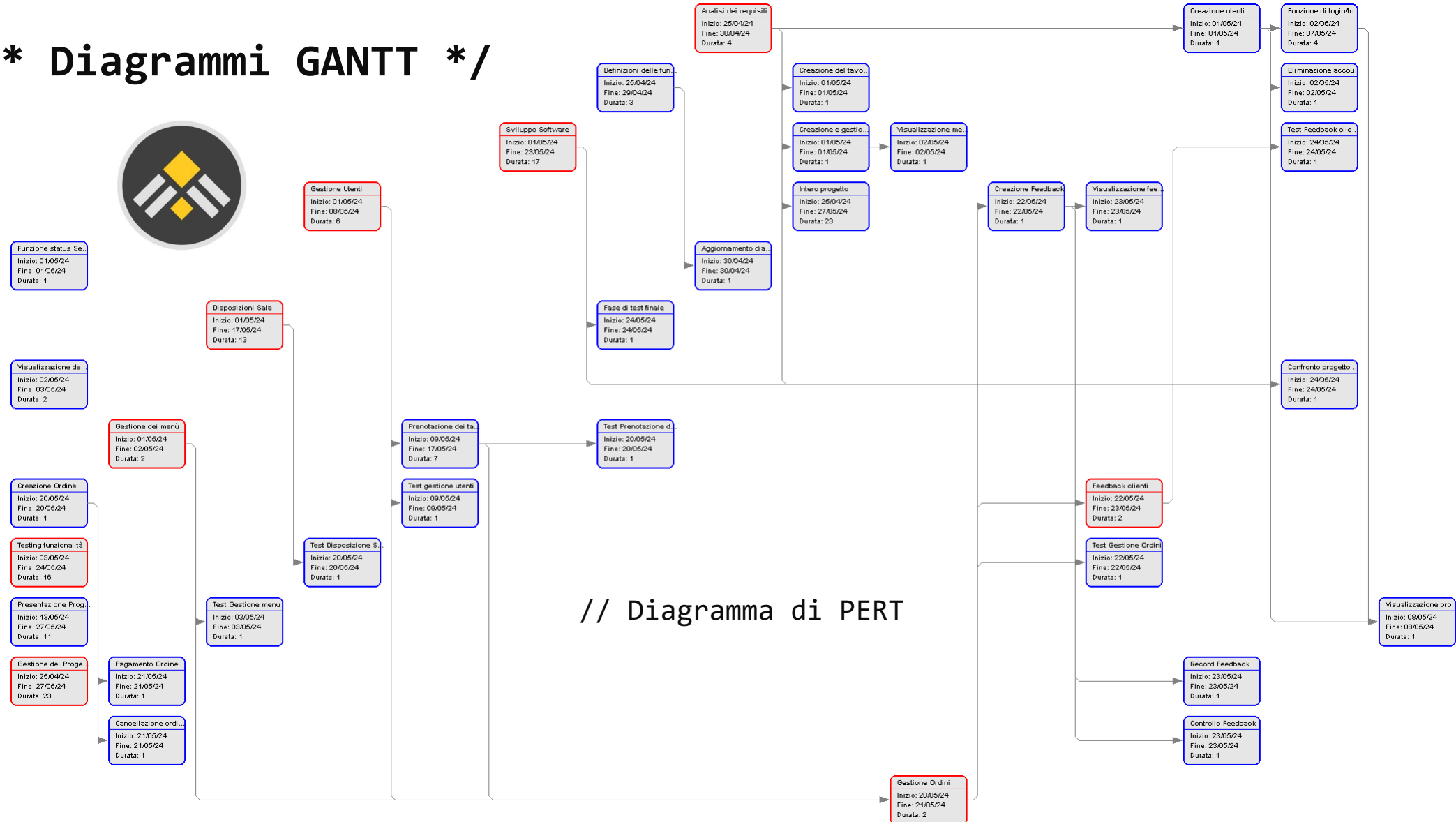
/*
*/



/*
Organizzazione
principale del lavoro
*/



/* Diagrammi GANTT */

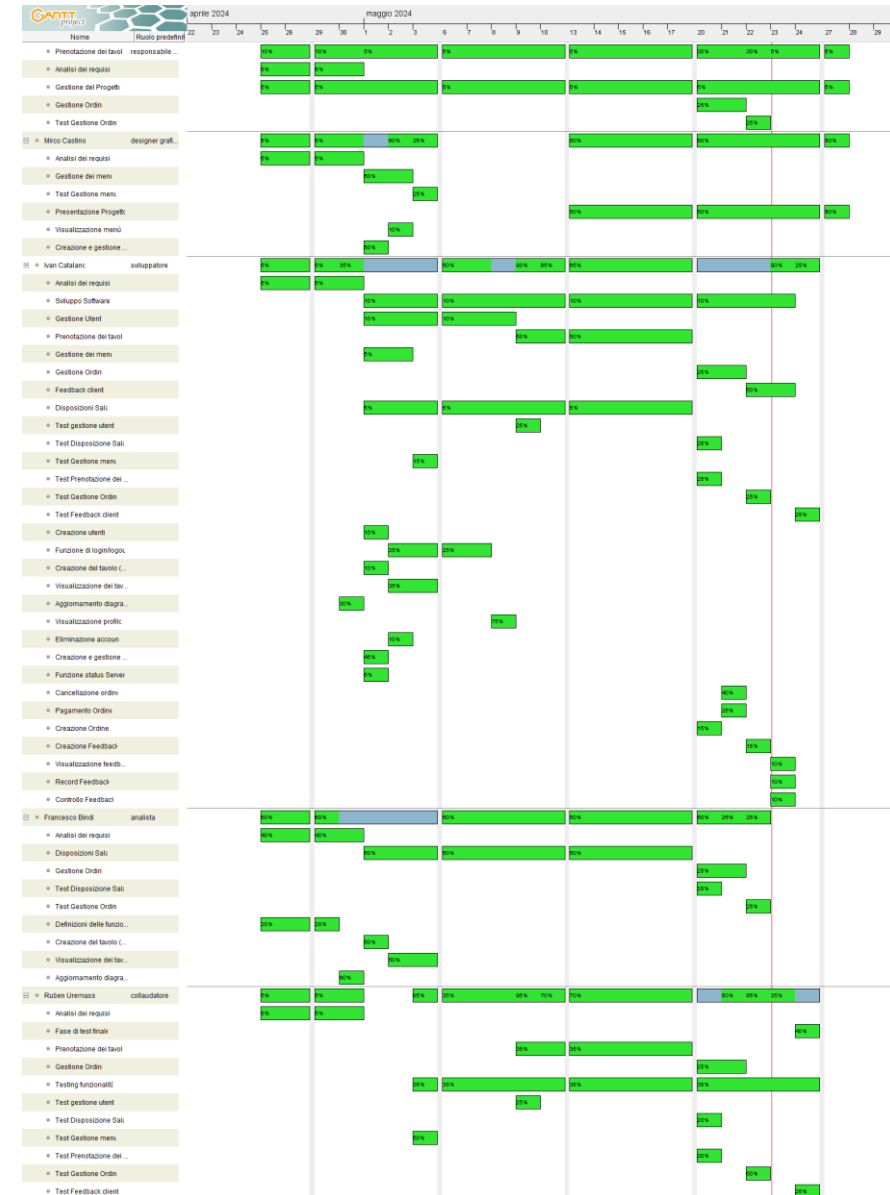


// Diagramma di PERT

/* Diagrammi GANTT */



// Diagramma delle risorse



/* README.md */

Uso del client

Per comprendere le operazioni eseguibili con il client, si può leggere la guida digitando `./client.exe --help`. I comandi a disposizione sono del tipo `./client.exe [PARAM] [OPTION/S]`.

Gestione dell'account

Per la gestione degli utenti, i *PARAM* e le *OPTIONS* sono:

Creazione account: `-c`, senza options.

Consente la creazione di un account nuovo. Verranno richiesti i dati dell'utente e la password verrà cifrata e resa non visibile durante la digitazione. Lo username è soggetto a controllo di esistenza e la password deve essere almeno di 8 caratteri.

Visualizzazione account: `-v`, seguito da "username".

Consente ad un account loggato, di visualizzare i dati dell'account, comprese le prenotazioni e le ordinazioni effettuate.

Modificare account: `-m`, seguito da "username".

Consente ad un account loggato, di modificare i campi di nome, cognome, email e password. Lo username non è modificabile.

Eliminazione account: `-e`, seguito da "username".

Consente ad un account loggato di eliminare l'account, comprese le prenotazioni effettuate. Gli ordini e le recensioni non vengono rimosse, in quanto storico del ristorante. Anche le ricevute emesse non vengono cancellate.

Login e Logout: `-login`, `-logout`, seguiti da "username".

Consentono il login e il logout dell'utente. Verrà richiesta la digitazione della password per entrambe le operazioni.



/* Black Box */