



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025



Gruppo: SWEg Labs

Email: gruppo.sweg@gmail.com

Verbale Riunione

11 Dicembre 2024

Indice

1	Informazioni generali	1
2	Ordine del giorno	2
3	Diario della riunione	3
4	Decisioni	4
5	Todo	5

1 Informazioni generali

- **Tipo di riunione:** interna
- **Luogo:** meeting Discord
- **Data:** 11/12/2024
- **Ora inizio:** 17:30
- **Ora fine:** 19:00
- **Responsabile:** Davide Verzotto
- **Scriba:** Federica Bolognini
- **Partecipanti:**
 - Federica Bolognini
 - Michael Fantinato
 - Giacomo Loat
 - Filippo Righetto
 - Riccardo Stefani
 - Davide Verzotto

2 Ordine del giorno

- Discussione sulla politica di comportamento del verificatore.
- *Casi d'uso*_G relazionati all'utente.
- Gli *attori*_G verranno descritti nel documento *Analisi dei Requisiti*_G.
- Aggiornamento *Database vettoriale*_G.
- Studio dei 5 temi tecnologici con conseguente *retrospettiva*_G.
- Avvio del *Proof of Concept*_G.
- Problema con la comprensione dei dati dal *bot*_G.
- Discussione sulla scelta dell'*architettura*_G.
- Avanzamento minimo garantito per il periodo natalizio.

3 Diario della riunione

- Durante la riunione si è discusso della politica di comportamento per il verificatore, delineando due possibili modalità di intervento:
 - Se il verificatore individua una modifica di piccola entità, può procedere a farla subito, informando l'autore del lavoro che la modifica è stata già effettuata.
 - Se il verificatore rileva una modifica più rilevante e impegnativa, può rinviare la correzione all'autore originale, comunicandogli cosa deve modificare.
- Si è deciso di procedere con la creazione dei *casi d'uso* relazionati all'utente, in conformità con le istruzioni apprese durante le lezioni in aula.
- Abbiamo stabilito di poter scrivere gli *attori* nel documento *Analisi dei Requisiti*.
- Durante la riunione si è discusso dell'aggiornamento del database vettoriale, dopo aver escluso la possibilità di un database remoto sono state considerate due possibili alternative:
 - La prima alternativa prevede l'uso di pacchetti gratuiti che creano un tunnel tra un server locale e rendono il servizio disponibile online. In questo modo, sarebbe possibile implementare una *GitHub Action_G*. Tuttavia, per ottenere un *URL_G* univoco è necessario un pagamento.
 - La seconda alternativa consiste nell'hostare un servizio di coda di richieste. In questo scenario, la *GitHub Action* si collegherebbe a un servizio di terze parti che memorizza le richieste *API_G* ricevute. Il container *Docker_G* si connette a questo servizio, verifica se sono state fatte modifiche nel branch principale (main) e avvia l'aggiornamento automatico. Questa soluzione potrebbe avere dei costi variabili, in base ai diversi piani offerti dal servizio.

In conclusione, è stato deciso di procedere con la soluzione del database locale, con la possibilità di riconsiderare l'opzione tra qualche mese.

- Durante la riunione si è discusso dei 5 temi tecnologici, assegnati come task nella precedente riunione. Successivamente, è stata fatta una retrospettiva sullo studio individuale e asincrono di questi temi, con il seguente riscontro:
 - Lo studio non è stato valutabile in modo completo, poiché il team è stato fortemente impegnato con l'*Analisi dei Requisiti*. Abbiamo deciso che lo studio proseguirà.
- Si è parlato di iniziare il *Proof of Concept*. Per avviare il progetto, si può creare la *repository_G*.
- Si è discusso di un potenziale problema nella consegna dei dati al *bot*, in particolare per far sì che il *bot* li comprenda correttamente. Un esempio è stato l'invio di una *issue_G* di *emphJira_G*, dove il *bot* non ha compreso che l'assegnatario era sotto la voce "assignee", nonostante avesse accesso ai dati inviati. Si è ipotizzato che potrebbe essere una limitazione di GPT-4o-mini, mentre una versione più avanzata come GPT-4o potrebbe risolvere il problema. Tuttavia, sarebbe utile fare un test per confermare questa ipotesi. Inoltre, si è proposta l'idea di provare con un header nel *emhprompt_G*, specificando di considerare solo i dati sotto determinate voci.
- Durante la riunione si è discusso della scelta dell'*architettura* per il progetto. Si è considerato che, per avere un' "architettura accogliente" come suggerito nel diario di bordo, potrebbe avere senso adottare il pattern *MVC_G*, poiché supporta aggiornamenti in tempo reale. Tuttavia, si è notato che *Angular_G* utilizza il pattern *MVVM_G*, che offre anch'esso la possibilità di gestire aggiornamenti in tempo reale. Al momento, il team ha deciso di non concentrarsi sul design e di procedere senza un pattern architetturale specifico. La questione sarà affrontata in seguito durante il *PB_G*.
- Abbiamo affrontato il tema di un avanzamento minimo garantito per il progetto durante le vacanze natalizie. Il team ha deciso di definire l'avanzamento minimo in seguito, in base a come evolverà il *PoC*.

4 Decisioni

Durante la riunione sono state prese le seguenti decisioni:

Codice	Descrizione
VI 18.1	Politica di comportamento per il verificatore: modifiche piccole fatte subito, modifiche grosse rimandate all'autore.
VI 18.2	I casi d'uso <i>back-end</i> \mathcal{G} devono essere scritti in relazione all'utente.
VI 18.3	Gli attori saranno descritti nel documento <i>Analisi dei Requisiti</i> .
VI 18.4	Richiedere un incontro online con il professore Cardin.
VI 18.5	Suddivisione dei compiti: Federica, Filippo e Davide per traduzione dei <i>casi d'uso</i> in <i>requisiti</i> mentre Riccardo, Giacomo e Michael per amministrazione e programmazione.
VI 18.6	Scelta del database locale per l'aggiornamento automatico.
VI 18.7	Continuare lo studio dei 5 temi tecnologici.
VI 18.8	Pianificazione delle strategie di programmazione.
VI 18.9	Inizio del <i>Proof of Concept</i> .
VI 18.10	Esplorazione di <i>Angular</i> in parallelo con il <i>PoC</i> .
VI 18.11	Non decidere subito l'architettura, iniziare senza un <i>pattern architetturale</i> \mathcal{G} .
VI 18.12	Bisogna stabilire un avanzamento minimo per il periodo natalizio.
VI 18.13	Bisogna stabilire un avanzamento minimo per il periodo natalizio.

5 Todo

Durante la riunione sono emersi i seguenti task da svolgere:

Codice	Dalla decisione	Assegnatario	Task Todo
BUD-85	VI 18.3	Filippo Righetto	Scrivere gli attori nel documento <i>Analisi dei Requisiti</i> .
BUD-86	VI 18.4	Tutto il gruppo	Richiedere un incontro online con il professore Cardin.
BUD-87	VI 18.5	Federica Bolognini, Filippo Righetto e Davide Verzotto	Tradurre i <i>casi d'uso</i> in <i>requisiti</i> .
BUD-88	VI 18.5	Riccardo Stefani	Sistema il suo piccolo lavoro svolto, inserendo le docstring.
BUD-89	VI 18.5	Riccardo Stefani	Testare la comprensione del <i>bot</i> e aggiungere un header nel <i>prompt</i> .
BUD-90	VI 18.5	Giacomo Loat	Aggiornamento automatico partendo dall'app.
BUD-91	VI 18.8	Michael Fantinato, Riccardo Stefani e Giacomo Loat	Pianificare delle strategie di programmazione.
BUD-92	VI 18.9	Riccardo Stefani	Creazione della repository per il <i>PoC</i> .
BUD-93	VI 18.10	Tutto il gruppo	Studio di <i>Angular</i> in parallelo con il <i>PoC</i> .
BUD-94	VI 18.12	Tutto il gruppo	Stabilire un avanzamento minimo per il periodo natalizio.
BUD-95	VI 18.13	Federica Bolognini	Redigere verbale interno del 11-12-24