

## Verbale Riunione Esterna

Versione	Data	Sez.modificate	Autore	Verificatore	Validatore
1.0.0	2024-11-26				Alberto Maggion
0.1.0	2024-11-22	Analisi dei Requisiti individuati proposti - commenti azienda		Michele Nesler	
0.0.1	2024-11-22	Stesura iniziale	Annalisa Egidi		

### Abstract

La riunione si è tenuta in sede Zucchetti in data 19/11/2024 per discutere della Bozza di Requisiti proposta dal gruppo.

## Contents

<b>1 Presenze</b> .....	2
<b>2 Ordine del Giorno</b> .....	2
<b>3 Discussione</b> .....	2
3.1 Dubbi e domande sul capitolato .....	2
3.2 Analisi dei Requisiti individuati proposti .....	3
3.2.1 Requisiti utente .....	3
3.2.2 Requisiti di sistema .....	4
3.2.3 Requisiti opzionali .....	5
3.2.4 Consigli al gruppo .....	5
<b>4 Argomenti da discutere in futuro</b> .....	5

## 1 Presenze

Rif.	Nominativo	Presenze	Ruolo
S	Annalisa Egidi	P	
S	Alberto Maggion	P	
S	Filippo Bellon	P	
S	Michele Nesler	P	
S	Alessandro Rossi	P	
S	Luca Calzetta	P	
A	Gregorio Piccoli	P	
A	Marco	P	
A	Michele	P	
A	Alberto	P	

Presenze	
P	Presente
A	Assente

Riferimenti	
A	Azienda
S	Studente

## 2 Ordine del Giorno

1. Dubbi e domande sul capitolato
2. Analisi dei requisiti

## 3 Discussione

### 3.1 Dubbi e domande sul capitolato

**Domanda** La scelta tra set di domande è un requisito facoltativo?

**Risposta** Si

**Domanda** C'è una percentuale minima di test da superare?

**Risposta:** Dipende dai dati, in base a come viene valutata la risposta troveremo un criterio di valutazione empirico.

**Domanda** E' possibile dividere le domande per argomento?

**Risposta:** Si è un'ottima idea, c'è la possibilità di dedurre l'argomento della domanda dal contenuto del contenuto della risposta stessa. Consigliamo di estrarre il json o sql della risposta per renderlo utilizzabile al confronto.

**Domanda** La scelta delle domande da porre deve avvenire all'interno del set di domande o dal database generale?

**Risposta:** Se vi risulta più semplice dividere le domande in set allora si può lavorare a compartimenti stagni, questo inoltre diminuirebbe anche il tempo di analisi e risposta del programma stesso, sarebbe positivo.

**Domanda** Ci sono già dei set o pool di domande?

**Risposta:** Noi prepareremo dei set di domande che vi forniremo ma conviene che anche voi abbiate dei set per fare delle prove immediate.

## 3.2 Analisi dei Requisiti individuati proposti

### 3.2.1 Requisiti utente

#### Proposti dal gruppo

1. Visualizzazione e modifica di una lista di domande e risposte attese
  - Visualizzazione domande e risposte
    - Selezione domande da eliminare
      - Pulsante conferma o annullamento eliminazione
    - Aggiunta nuova domanda
      - Inserimento campi obbligatori: domanda e risposta attesa
      - Pulsante conferma o ulteriore aggiunta e annulla
    - Selezione domanda da modificare
      - Modifica domanda selezionata
      - Pulsante salvataggio o annullamento modifica
2. Selezione domande da porre
3. Strumento di esecuzione domande e registrazione risposte
  - Apertura automatica strumento valutazione all'arrivo delle risposte
4. Strumento di presentazione dei risultati del test
  - Presentazione risultati migliori e peggiori
  - Lista risposte ottenute

#### Commenti dell'azienda

1. Potrebbe esserci una checkbox che faccia pre-scegliere il metodo da utilizzare per l'analisi e/o confronto della risposta e vedendo la risposta (es. json) avere un hint della metodologia di analisi da utilizzare.
  - Utilizzare una chiave unica per il salvataggio di ciascuna domanda nel database.
  - Includere l' inserimento dell'autore nella creazione delle domande
  - Il caso d'uso sembra andare troppo nel dettaglio (es. bottoni), ma il caso d'uso generale deve essere CRUD+L (create, read, update, delete, list) quindi sarebbe preferibile dare un caso d'uso più generico in modo che non sia limitante, un caso d'uso troppo nel dettaglio poi non si presta all'implementazione - consigliamo di chiedere al prof. Cardin
2. Possiamo dire che le domande verranno fatte tutte, poi in tempo di implementazione opzionale si potrà fare una selezione delle domande in set
3. Viene eseguita la domanda, si ottiene la risposta, la risposta viene affiancata dalla trustness e lo score. La domanda, la risposta, la trustness e lo score di ogni risposta deve essere salvata in un archivio (che sia un foglio excel, oppure un csv, una stampa,...).  
Ci dovrà essere un archivio delle domande, un archivio dei casi di test e un archivio delle esecuzioni (delle run - "ho eseguito questo test su quel sistema") in modo che sia replicabile per vedere la differenza nel risultato al cambiamento del sistema - ma questo è opzionale. Di non opzionale c'è la run e la produzione del report.
4. Requisito obbligatorio: mostrare nella run le migliori e peggiori risposte. Serve la presentazione dei risultati per la singola run per vedere cosa è andato male nella risposta. Requisito opzionale: confrontare run differenti.

5. Per la rappresentazione dei risultati il grafico è immediato (a torta con gli score) ed essenziale, si ha la necessità anche una tabella avente all'interno in particolar modo le domande andate male in cui viene mostrata: la domanda, la risposta che mi aspettavo e la risposta ricevuta e l'algoritmo/metodo utilizzato per lo score.
  - per la rappresentazione dei risultati è possibile la creazione di un csv che viene poi utilizzato per la rappresentazione dei risultati

### 3.2.2 Requisiti di sistema

#### Proposti dal Gruppo

1. Adozione di pattern architetturali robusti e duttili
2. Sistema di test automatico
  - Adeguata copertura test automatici
3. Pubblicazione progetto su repository pubblico
  - Inclusione di Repo documentazione esterna
4. Implementazione sistema API per connessione LLM
5. Sistema di esecuzione LLM su cui eseguire i test
6. Algoritmo per il confronto e la valutazione delle risposte
7. Sistema di archiviazione e modifica domande
8. Interfaccia che espone gli strumenti disponibili all'utente

#### Commenti dell'azienda

1. MVC consigliabile per la presentazione e differente per la parte di confronto risposte
2. Da definire a fronte dello sviluppo del programma di valutazione
3. Come da specifica
4. ci saranno le API per connettersi con LLM per il confronto, e le API per connettersi al programma e fare le domande
5. Come da specifica
6. Come da specifica
7. tendenzialmente l'azienda usa PostgresQL
8. Come da specifica

### 3.2.3 Requisiti opzionali

#### Proposti dal Gruppo

1. Storico dei test effettuati
2. Confronto automatico tra diverse esecuzioni salvate dei test
  - Utilizzo di un indice sintetico e di un programma di confronto
  - Evidenziazione miglioramenti e peggioramenti
3. Implementazione di diversi set di domande per testare diversi aspetti delle risposte
4. Caricamento di un file contenente le domande
5. Possibilità dell'utente di scaricare le risposte avute dal LLM
6. Aggiunta etichette al set di domande per valutare ambiti differenti (matematico, logico, analisi del testo ....)

#### Commenti dell'azienda

1. nessun commento
2. nessun commento
3. nessun commento
4. nessun commento
5. Si potrebbe pensare di scrivere in un file di testo/json per poi fare un caricamento massivo
6. nessun commento

### 3.2.4 Consigli al gruppo

In questo progetto non è focale una bella interfaccia grafica ma piuttosto la parte di confronto risposte

## 4 Argomenti da discutere in futuro

Nessun argomento rimandato.

Approvazione azienda

Nome:

GRIGORIO PICCOLI

Firma:



**Zucchetti S.p.A.**

Via Solferino, 1 - 26900 LODI

Tel. 0371.5945700 - Fax 0371.5945753

Sede Op.: Via G. Cittadella, 7 - 35137 PADOVA

P. IVA e Cod. Fisc. 05006900962

THE UNIVERSITY OF CHICAGO  
LIBRARY  
540 EAST 57TH STREET  
CHICAGO, ILL. 60637  
TEL: 773-936-3700