**ANALISI STATICA**

Per l’analisi statica del codice è stato utilizzato un plug-in disponibile in Android Studio chiamato MetricsReloaded che fornisce le metriche del codice sorgente in esame.  
Il plugin fornisce delle misure di complessità del codice. In particolare:

* **CogC:** Cognivite Complexity  
  Misura la complessità in termini di comprensione del codice. Questo valore aumenta all’aumentare del numero di strutture controllate e strutture annidate.
* **ev(G)**: Essential Cyclomatic Complexity   
  Misura quanto sia mal strutturato il flusso di controllo di un metodo.
* **iv(G)**: Design Complexity  
  Calcola la complessità di progettazione di un metodo. La complessità della progettazione è correlata al modo in cui un flusso di controllo dei metodi è interconnesso con le chiamate ad altri metodi.
* **v(G)**: Cyclomatic Complexity  
  È una misura del numero di percorsi di esecuzione distinti attraverso ciascun metodo. Questo può anche essere considerato come il numero minimo di prove necessarie per esercitare completamente il flusso di controllo di un metodo. In pratica, questo è 1 + il numero di if, while, for, do, switch case, catch, espressioni condizionali, and e or nel metodo.

L’analisi con MetricsReloaded ha dato i seguenti risultati, ordinati in base al valore decrescente di v(G).

Immagine che contiene testo

Descrizione generata automaticamente

**com.example.whatsonsalon.SelezionaSlotAppuntamento.proponiSlot(syncConfiguration)**

Per questo metodo si hanno alti valori per Cognitive Complexity, Design Complexity e Cyclomatic Complexity, giustificati dalla presenza di strutture di controllo annidate e chiamate ad altri metodi.

Per interpretare al meglio il valore di Cyclomatic Complexity è necessario considerare il valore di Essential Cyclomatic Complexity. Il primo parametro può essere interpretato come difficoltà di testing del metodo in quanto rappresenta il numero di percorsi di esecuzione distinti, definiti dalla presenza di punti di decisione del (if, while, for, do, switch case, catch, espressioni condizionali, and e or). Per il metodo in esame ci sono 17 possibili percorsi da testare. In genere, 1-4 è una complessità bassa, 5-7 indica una complessità moderata, 8-10 è una complessità elevata e 11+ è una complessità molto elevata.   
L’elevata complessità ciclomatica viene però compensata dal valore unitario della complessità essenziale. Questo parametro, che indica quanto sia mal strutturato il flusso di controllo di un metodo, indica che è relativamente semplice scomporre il metodo in sotto-metodi più semplici da testare.

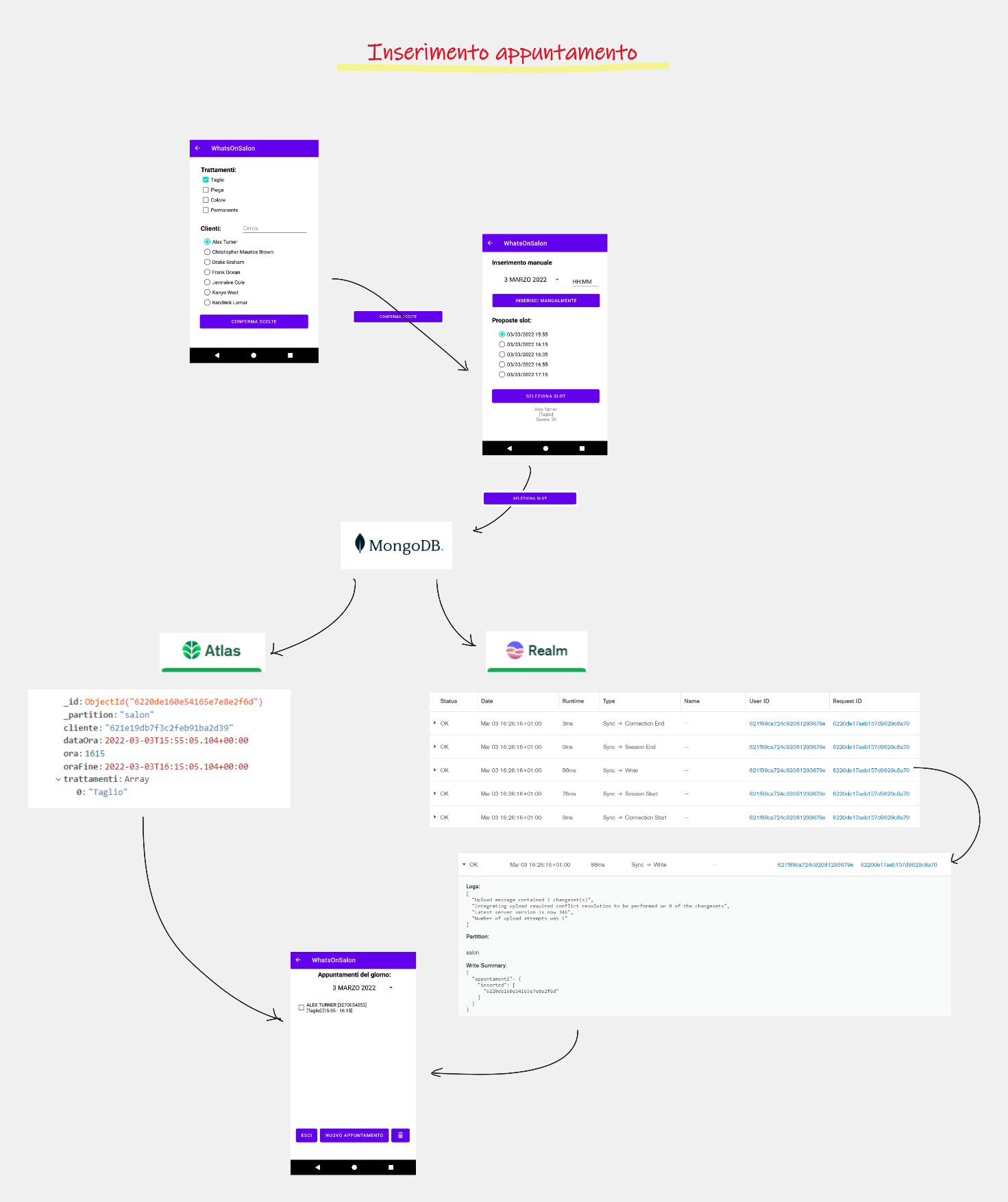
Le considerazioni fatte per questo metodo si riflettono anche per gli altri metodi con stessi risultati.

**ANALISI DINAMICA**

Per effettuare un’analisi dinamica del progetto si è preferito verificare il corretto funzionamento delle chiamate al database. Per facilitare questa fase ci si è affidati alla console di logs offerto dalla piattaforma Realm e alla verifica manuale degli inserimenti/cancellazioni dei documenti dal data lake Atlas.

A tal proposito si è simulato un inserimento di un appuntamento e alla sua successiva eliminazione dal database.

**Inserimento appuntamento**



**Cancellazione appuntamento**

