Updated Development Prompt – PrimaBruxismClinicalSuite

⚙️ STRUTTURA DEFINITIVA DEL SOFTWARE  
  
🧠 Nome progetto: PrimaBruxismClinicalSuite  
📁 Cartelle principali:  
\* src/model/ – Strutture dati (anagrafica, FTS, diagnosi, AFI, ITSL)  
\* src/logic/ – Calcoli, soglie, mappature, logiche ITSL e AFI  
\* src/ui/ – JavaFX GUI  
\* src/io/ – Import/Export CSV, salvataggio profili  
\* resources/ – File statici, icone, template CSV

\*patients/ - Cartelle pseudonime dei pazienti

✅ SESSIONI DEFINITIVE PER LO SVILUPPO  
  
✅ PROMPT DEFINITIVO – SESSIONE 1  
Titolo progetto: PrimaBruxismClinicalSuite  
Contesto:  
Stiamo sviluppando un software Java (da realizzare in Visual Studio Code) che implementa il modello decisionale clinico PRIMA per la diagnosi e trattamento del bruxismo. Il modello, descritto in dettaglio in un articolo medico già finalizzato, si basa su una struttura euristica multilivello che parte da dati anamnestici, auto-valutativi e strumentali, e guida il clinico, attraverso un calcolo strutturato (inclusivo di FTS, AFI Stratification e ITSL), a un piano terapeutico personalizzato.  
  
📘 Premessa per la programmazione  
Il software sarà incluso nei Supplementary Materials di un articolo scientifico multidisciplinare. In questa e nelle successive sessioni, caricheremo:  
\* L’intero articolo in formato IMRAD  
\* Tutte le tabelle (Tables 1–6)  
\* Tutte le Appendix (4–8)  
Questi file contengono:  
\* Le domande cliniche da somministrare al paziente  
\* Le soglie diagnostiche  
\* Le logiche decisionali e terapeutiche  
\* I mapping strumenti/terapie  
\* La stratificazione degli Aggravating Factors in sottogruppi clinici (IFA Stratification), con logica di impatto su diagnosi, prognosi e risposta terapeutica, come definito in Appendix 9  
\* La logica di sequencing intra-dominio (ITSL)  
Ogni parte del programma dovrà leggere e interpretare file .csv generati direttamente da queste tabelle, quindi non saranno scritte a mano ma caricate dinamicamente.  
  
🔁 IMPORTANTE  
In questa sessione caricherò il file PROJECT\_CONTEXT.md, che contiene:  
\* La struttura completa del progetto  
\* Le logiche cliniche  
\* Le architetture software  
\* Il piano di sviluppo completo  
Questo file va letto e usato come stato iniziale del progetto. Verrà aggiornato al termine della sessione.  
  
🔧 OBIETTIVO della SESSIONE 1: Setup completo del progetto Java  
1. Crea la struttura base del progetto PrimaBruxismClinicalSuite, con le seguenti cartelle:  
src/  
├── model/ # Classi cliniche (PatientProfile, FTS, AFI, ITSL, etc.)  
├── logic/ # Motori decisionali (calcolo FTS, AFI, diagnosi, terapia)  
├── ui/ # Interfaccia utente (JavaFX)  
├── util/ # Parser CSV, gestori file  
data/ # File CSV di input clinico  
resources/ # Icone, CSS, immagini GUI  
test/ # Unit test

├── patients/ # Cartelle pseudonime dei pazienti

│ ├── PRIMA-ID-000123/

│ │ ├── STAB\_input.csv

│ │ ├── FTS\_output.json

│ │ ├── TherapyLog.txt

│ │ └── session.log  
README.md  
PROJECT\_CONTEXT.md  
PrimaBruxismLauncher.java  
  
2. Crea le classi segnaposto:  
\* PatientProfile.java  
\* CSVDataSet.java  
\* FTSCalculator.java  
\* AFICalculator.java

\* IFAParser.java

\* IFAEvaluator.java

\* IFAReportGenerator.java  
\* ThresholdEvaluator.java  
\* DiagnosticEngine.java  
\* ITSLSequencer.java  
\* TherapyEngine.java  
\* MainApp.java (entry point, con interfaccia JavaFX)  
  
3. Crea il file README.md con:  
\* Descrizione generale del progetto  
\* Collegamento con il modello PRIMA  
\* Requisiti tecnici (Java 17+, JavaFX, OpenCSV o simili)  
\* Istruzioni di compilazione e avvio  
  
📎 Allegati che caricherò in questa sessione:  
\* PROJECT\_CONTEXT.md (obbligatorio)

\* Appendix 9 – IFA Stratification (CSV)

\* Esempio di output del calcolo IFA (CSV e JSON)  
\* Esempi di file CSV per simulare l’input clinico del paziente  
  
🎯 Procedi ora con:  
\* Lettura del file PROJECT\_CONTEXT.md  
\* Creazione della struttura di progetto come richiesto  
\* Output di conferma pronto per continuare con SESSIONE 2 (inserimento dati paziente)

✅ **SESSIONE 2 – Anagrafica paziente + Importazione CSV**

**Siamo nella SESSIONE 2 del progetto *PrimaBruxismClinicalSuite*, un software Java con GUI basato sul modello PRIMA per il bruxismo.** **Inizia caricando il file PROJECT\_CONTEXT.md aggiornato.** Questo file contiene tutte le specifiche cliniche, le strutture di cartelle e le logiche di calcolo già definite nella SESSIONE 1.

**Obiettivo della SESSIONE 2:**

1. Completare PatientProfile.java con:
   * Dati anagrafici: nome, cognome, codice fiscale, email, cellulare, data di nascita, patologia.
   * Possibilità di caricare i file .csv delle Appendix Table A1–A4.
2. Creare CSVImporter.java per leggere i file e trasformarli in oggetti CSVDataSet.
3. GUI JavaFX per inserimento dati e caricamento CSV.

✅ Al termine, fornirò la versione aggiornata del **PROJECT\_CONTEXT.md** da usare nella SESSIONE 3.

✅ **SESSIONE 3 – Calcolo FTS (Functional Tendency Scores)**

**Siamo nella SESSIONE 3. Inizia caricando il PROJECT\_CONTEXT.md aggiornato.** I file **.csv** delle Appendix A1–A4 sono stati già importati.

**Obiettivo della SESSIONE 3:**

1. Creare FTSCalculator.java per calcolare i 4 FTS:
   * Muscular, Articular, Occlusal, Psychological.
   * Calcolo percentuale sul massimo.
   * Classificazione: Unlikely (0–25%), Possible (26–65%), Probable (>65%).
2. Memorizzazione dei risultati nel profilo paziente.
3. Visualizzazione GUI dei punteggi FTS.

✅ Al termine, aggiornerò il **PROJECT\_CONTEXT.md** da usare nella SESSIONE 4.

✅ **SESSIONE 4 – Suggerimenti diagnostici**

**Siamo nella SESSIONE 4. Carica il PROJECT\_CONTEXT.md aggiornato.**

**Obiettivo della SESSIONE 4:**

1. Creare DiagnosticEngine.java:
   * In base ai FTS, suggerire strumenti diagnostici per ciascun dominio (secondo Table 2).
2. GUI per mostrare:
   * Suggerimenti generati
   * Possibilità per il medico di aggiungere test manuali
3. Classe DiagnosticResult.java per inserimento dei risultati strumentali.

✅ Al termine, aggiornerò il **PROJECT\_CONTEXT.md** per la SESSIONE 5.

✅ **SESSIONE 5 – Confronto con soglie + Strategia terapeutica**

**Siamo nella SESSIONE 5. Carica il PROJECT\_CONTEXT.md aggiornato.**

**Obiettivo della SESSIONE 5:**

1. ThresholdEvaluator.java:
   * Confronta i risultati strumentali con le soglie cliniche (Table 3).
   * Attiva terapia solo se:
     + Score ≥13/15
     + ≥60% da parametri “Core Diagnostic”
2. TherapyEngine.java:
   * Suggerisce terapie da Table 4 in base ai domini attivati.
3. GUI con pannello decisionale e possibilità di aggiustamento manuale.

✅ Al termine, aggiornerò **PROJECT\_CONTEXT.md** per la SESSIONE 6.

✅ **SESSIONE 6 – Follow-up e andamento clinico**

**Siamo nella SESSIONE 6. Carica il PROJECT\_CONTEXT.md aggiornato.**

**Obiettivo della SESSIONE 6:**

1. Classe FollowUpEntry.java:
   * Registra follow-up successivi per ciascun dominio.
2. Visualizzazione andamento clinico nel tempo tramite grafici JavaFX.
3. Possibilità di rivalutare FTS e strategie terapeutiche ad ogni follow-up.

✅ Al termine, aggiornerò **PROJECT\_CONTEXT.md** per la SESSIONE 7.

✅ **SESSIONE 7 – Esportazione e salvataggio profili**

**Siamo nella SESSIONE 7. Carica il PROJECT\_CONTEXT.md aggiornato.**

**Obiettivo della SESSIONE 7:**

1. Esportazione dati:
   * PDF, CSV, JSON del profilo paziente con tutti i dati (FTS, test, terapie, follow-up).
2. Salvataggio e caricamento locale (es. con SQLite o JSON).
3. Pulsanti GUI per esportazione/archiviazione.

✅ Al termine, aggiornerò **PROJECT\_CONTEXT.md** per la SESSIONE 8.

✅ **SESSIONE 8 – Finalizzazione progetto per Submission**

**Siamo nella SESSIONE 8 (finale). Carica il PROJECT\_CONTEXT.md aggiornato.**

**Obiettivo della SESSIONE 8:**

1. Refactoring e JavaDoc completo.
2. Preparazione README.md con:
   * Struttura progetto
   * Istruzioni d’uso
   * Requisiti tecnici
   * Collegamento al paper
3. Packaging .zip del software per pubblicazione nei Supplementary Materials:
   * Codice completo
   * File .csv di esempio
   * Manuale rapido in PDF
   * Licenza MIT o simile