

Leonardo Minotti

Relatore: Daniela Micucci

Correlatore: Maria Teresa Rossi

Sviluppo di applicazioni mobile con supporto di ChatGPT: GreenWay per la sensibilizzazione sulla Co2



Applicazioni LLM:

1. Elaborazione del linguaggio naturale
2. Traduzione automatica
3. Generazione di testo automatico
4. Riconoscimento e generazione di immagini
5. Assistenza virtuale e chatbot

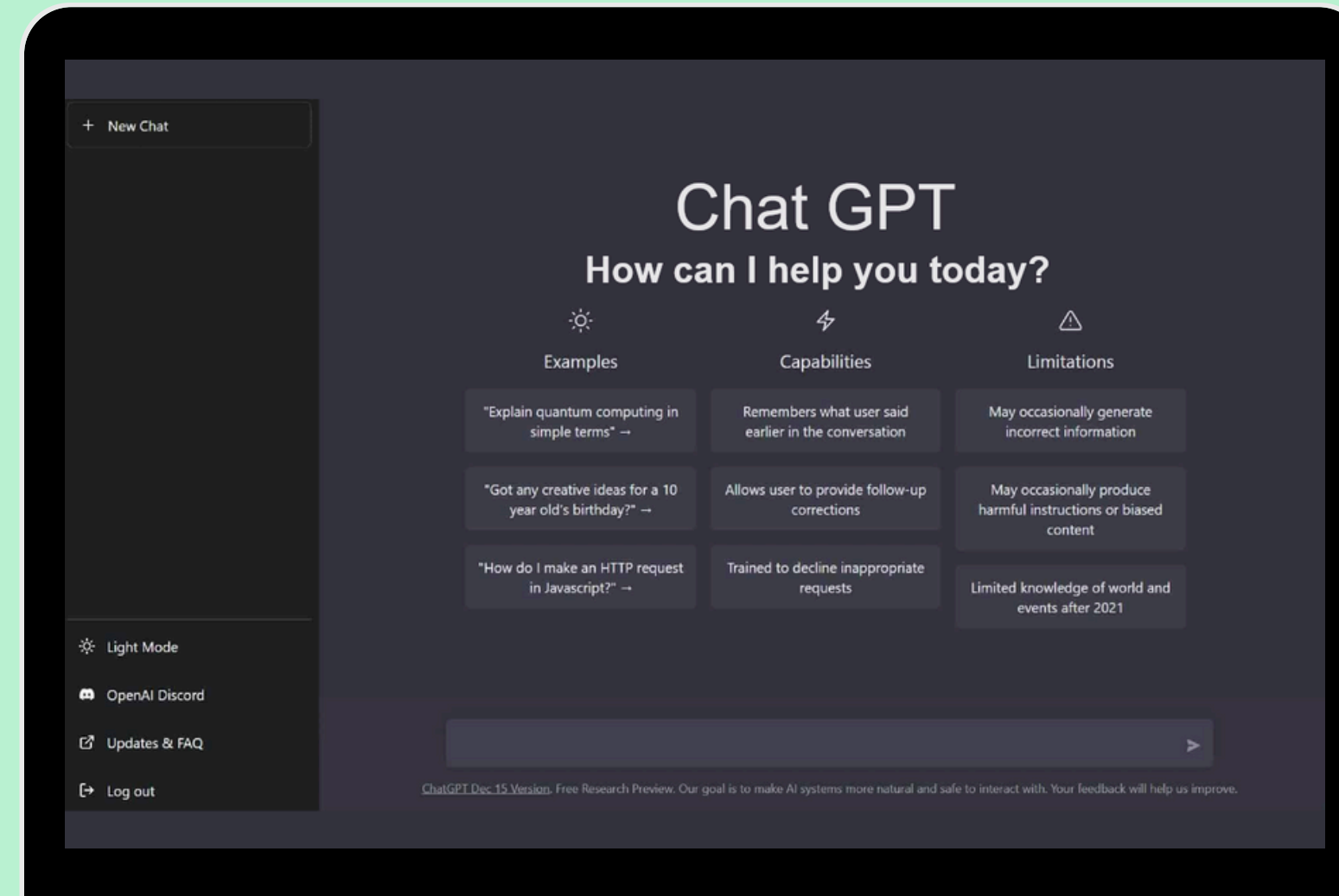
I LARGE LANGUAGE MODEL

Sviluppo software

Obbiettivo della sperimentazione:
verificare quanto uno strumento
come ChatGPT supporti nelle varie
fasi di sviluppo il
progettista/programmatore

Fasi di sviluppo:

1. Documentazione
2. Progettazione
3. Implementazione
4. Testing



METODO DI VERIFICA:

Sviluppo di Greenway:

scelta di sviluppare un'applicazione nell'ambito della sostenibilità.

Diario interazioni:

annotazione delle interazioni con ChatGPT durante le fasi di:

- specifica dei requisiti
- progettazione architettuale
- sviluppo
- testing



REQUISITI:

1

Funzionali:

- Visualizzazione qualità aria.
- Calcolo CO2 percorsi.
- Sistema di amicizie e sfide

2

Non funzionali:

- Efficienza
- Sicurezza
- Usabilità

3

Stakeholder:

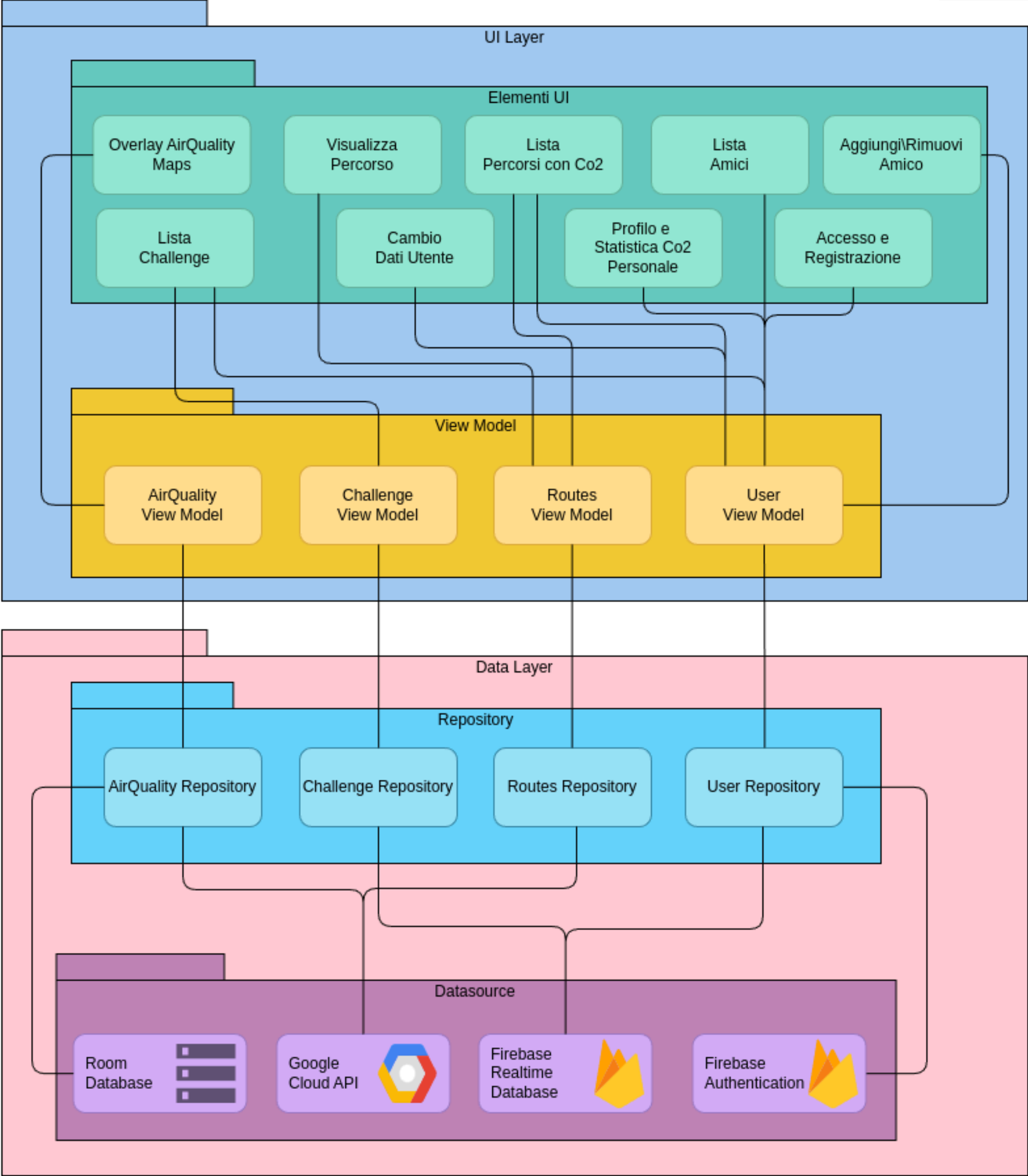
- Utenti
- Sviluppatori

4

Strumenti utilizzati:

- SDK
- API
- UI Design

ARCHITETTURA:



TESTING

Test locali:

- Framework Robolectric
- Libreria JUnit
- Libreria Mockito

Test strumentati:

- Monkey
- Firebase TestLab
- Espresso

Test di performance:

- Perfetto
- Android Profiler

Test qualità del software:

- SonarQube

```
@Test
public void testCo2Converter() {

    // Test 1: Verifica la conversione con un valore intero
    double input1 = 1000;
    String expectedOutput1 = "1.000kg";
    String actualOutput1 = instance.co2Converter(input1);
    assertEquals(expectedOutput1, actualOutput1);

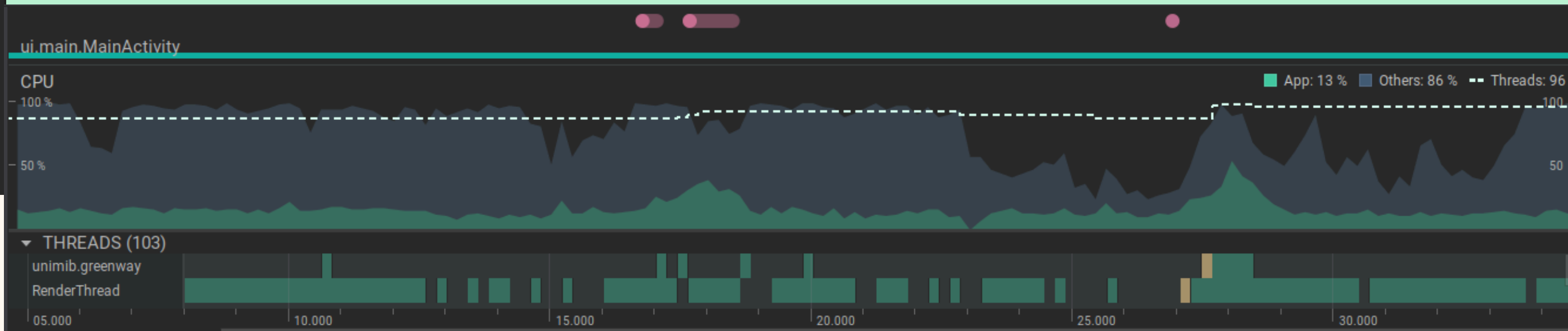
    // Test 2: Verifica la conversione con un valore decimale
    double input2 = 2500.5;
    String expectedOutput2 = "2.501kg";
    String actualOutput2 = instance.co2Converter(input2);
    assertEquals(expectedOutput2, actualOutput2);

    // Test 3: Verifica la conversione con un valore molto piccolo
    double input3 = 0.5;
    String expectedOutput3 = "0.001kg";
    String actualOutput3 = instance.co2Converter(input3);
    assertEquals(expectedOutput3, actualOutput3);

    // Test 4: Verifica la conversione con un valore molto grande
    double input4 = 1000000;
    String expectedOutput4 = "1000.000kg";
    String actualOutput4 = instance.co2Converter(input4);
    assertEquals(expectedOutput4, actualOutput4);

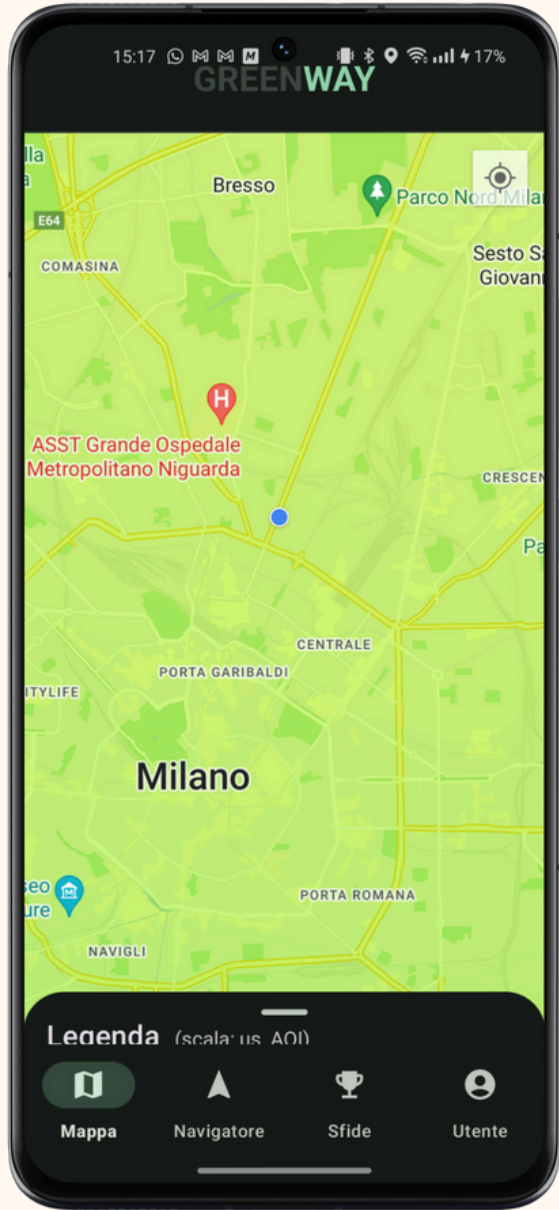
    // Test 5: Verifica la conversione con zero
    double input5 = 0;
    String expectedOutput5 = "0.000kg";
    String actualOutput5 = instance.co2Converter(input5);
    assertEquals(expectedOutput5, actualOutput5);

    // Test 6: Verifica la conversione con un valore negativo
    double input6 = -1000;
    String expectedOutput6 = "-1.000kg";
    String actualOutput6 = instance.co2Converter(input6);
    assertEquals(expectedOutput6, actualOutput6);
}
```

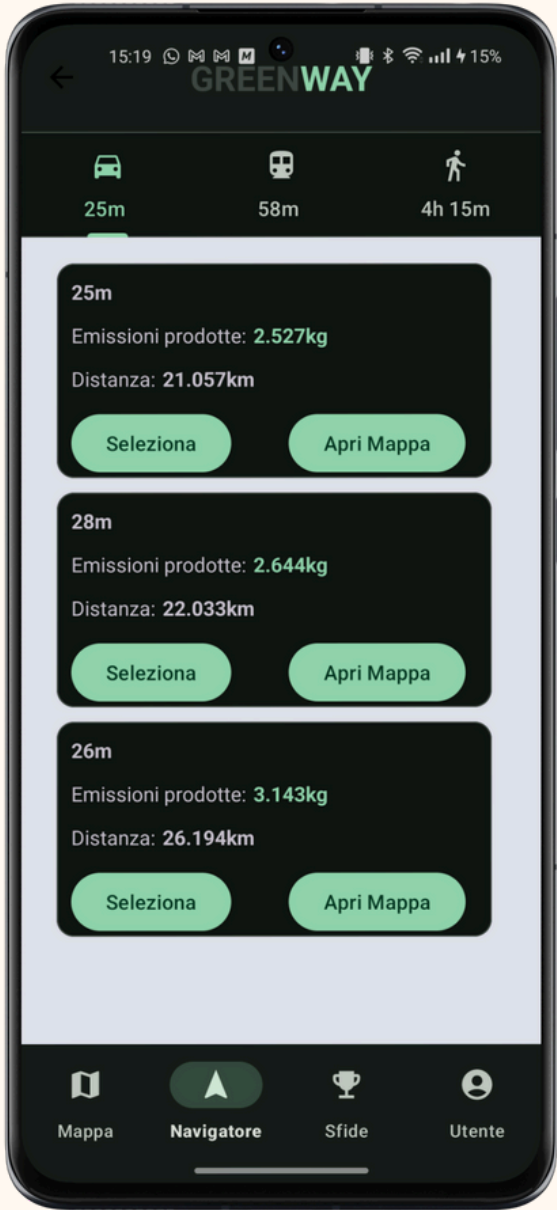


SCHERMATE GREENWAY

MAPPA



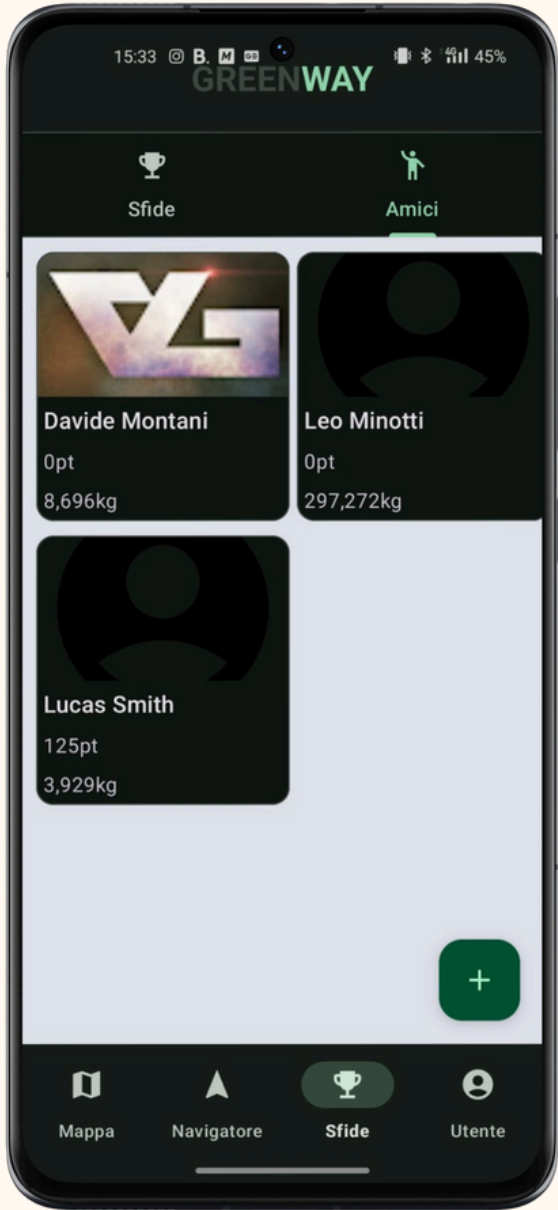
NAVIGATORE



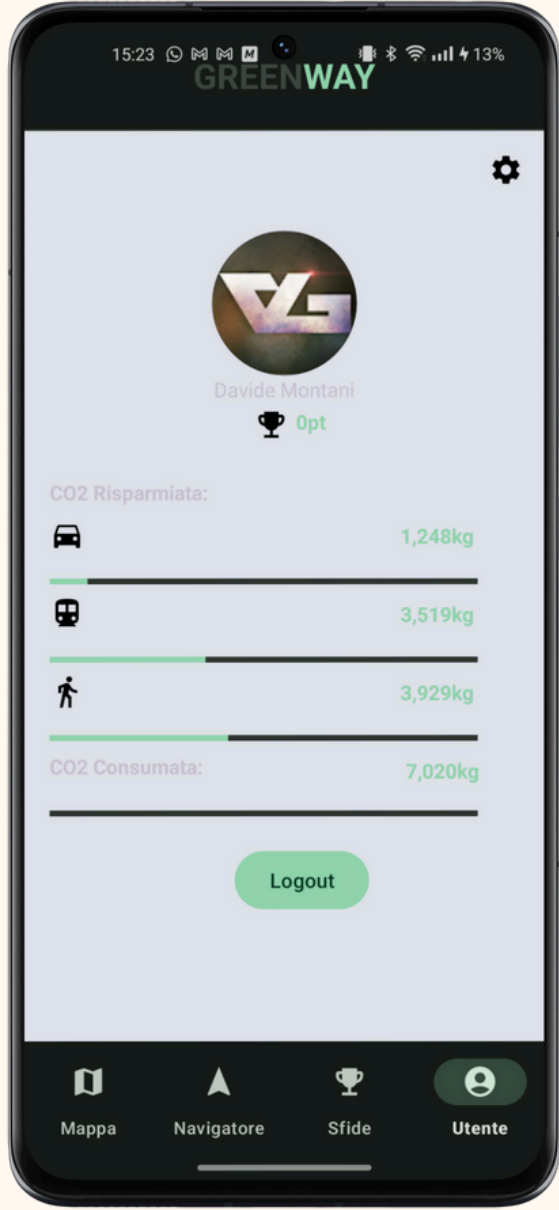
SFIDE



AMICI



PROFILO



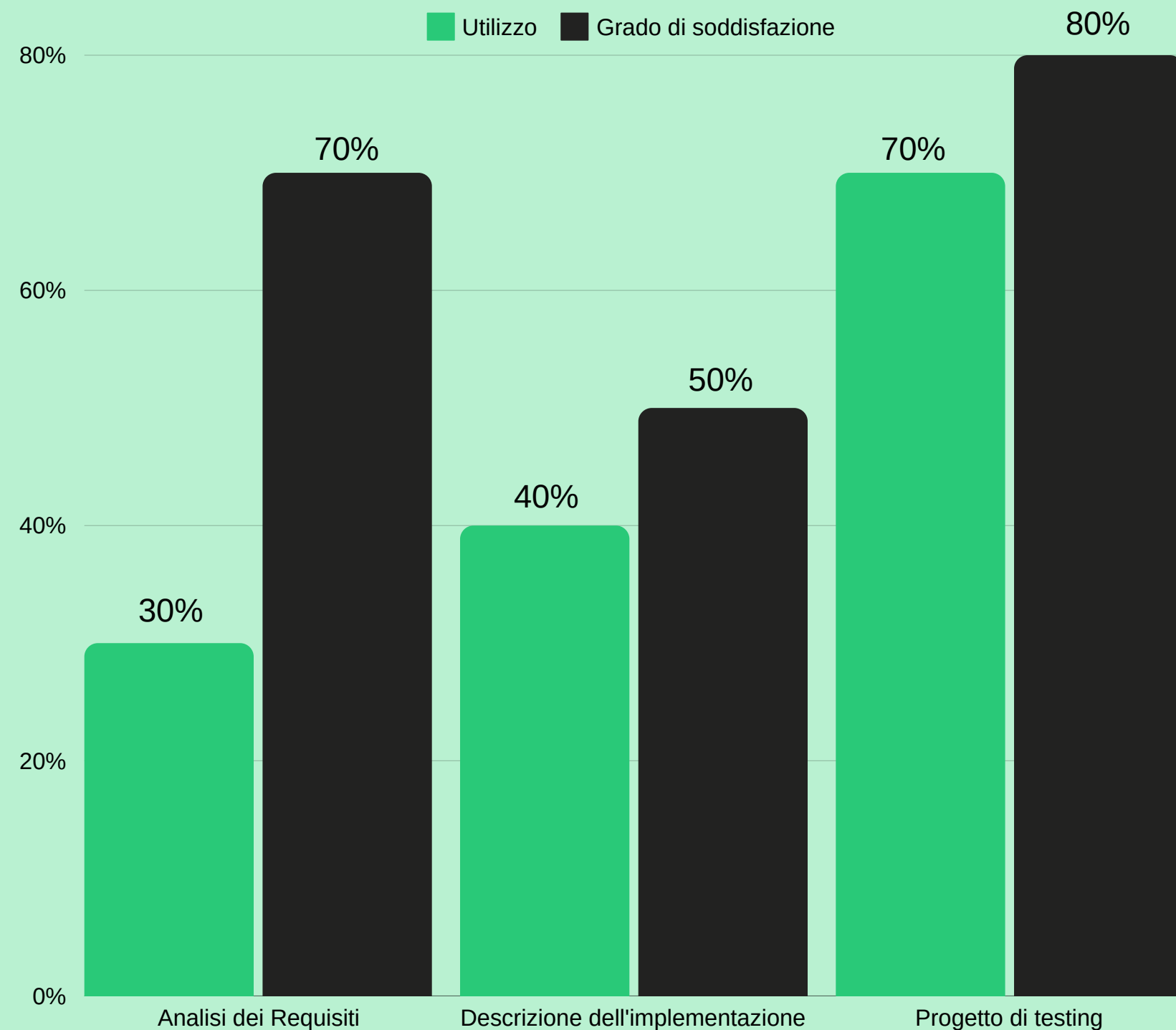
ANALISI CRITICA E CONSIDERAZIONI FINALI

Punti di forza:

- Un'alta utilità nella fase di testing.
- Risposte generalmente accurate e utili per l'analisi dei requisiti.

Criticità:

- Necessità di migliorare l'accuratezza nella fase di implementazione.
- Essenziale verificare e correggere le risposte generate da ChatGPT.



**GRAZIE PER
L'ATTENZIONE**