**Capitolo 4**

**Utilizzo di ChatGPT**

* 1. **Introduzione**

Questo capitolo esplora come ChatGPT è stato utilizzato nelle diverse fasi dello sviluppo dell'applicazione EcoSwap, evidenziando il suo ruolo e la sua efficacia nel supportare lo sviluppo. L'obiettivo è valutare quindi l'impatto di ChatGPT durante l'analisi dei requisiti, l'implementazione delle funzionalità e il testing del software. L'analisi si concentrerà su come ChatGPT ha contribuito a migliorare la comprensione dei requisiti, a risolvere problemi di programmazione come la correzione di errori, e a verificare la correttezza delle funzionalità implementate. Attraverso esempi pratici e valutazioni critiche, verranno esaminati i vantaggi tangibili e i limiti riscontrati nell'uso di questo strumento. La panoramica fornita metterà in luce le potenzialità di ChatGPT nel migliorare il processo di sviluppo di applicazioni mobili. Alla fine del capitolo, una sezione conclusiva presenterà delle statistiche che riassumono per ogni fase dello sviluppo quanto ChatGPT è stato utilizzato e quanto è stato utile.

* 1. **Analisi dei requisiti**
     1. **Ruolo di ChatGPT nell’Analisi dei requisiti**

 Spiegazione di come ChatGPT è stato utilizzato per identificare e definire i requisiti dell'applicazione.

 Discussione delle sfide affrontate e come ChatGPT ha contribuito a risolverle.

* + 1. **Vantaggi e limiti**

 Valutazione dei benefici ottenuti dall'utilizzo di ChatGPT in questa fase.

 Discussione sui limiti riscontrati e sulle aree di miglioramento.

* 1. **Implementazione**
     1. **Ruolo di ChatGPT nell’Implementazione**

Durante la fase di implementazione delle funzionalità dell'applicazione, ChatGPT ha svolto un ruolo significativo come strumento di supporto per lo sviluppo. In particolare, ChatGPT è stato utilizzato per implementare specifiche funzionalità e ottimizzare il codice esistente. Piu specificatamente, ChatGPT è stato utilizzato per i seguenti aiuti:

* **Consigli su sintassi flutter**

Come già detto, abbiamo voluto testare anche quanto un LLM può aiutare nell’apprendimento di nuovi linguaggi. Uno dei primi utilizzi di ChatGPT è stato per porre domande su come scrivere del codice in Flutter. Ad esempio, abbiamo chiesto come funziona la creazione di pagine in Flutter, la creazione di elementi, come scrivere uno specifico algoritmo in Flutter. Inoltre, conoscendo meglio il linguaggio Java per lo sviluppo Android, abbiamo chiesto quali sono i corrispondenti di Flutter di alcuni elementi di Android, come i bottoni o una bottom/top bar, e come implementare alcune meccaniche presenti in Android, come i ViewModel o il meccanismo di Observer, in Flutter.

* **Suggerimenti per le funzionalità**

Un altro ambito in cui ChatGPT dimostra particolare efficacia è nel suggerimento di funzionalità per l’applicazione, fornendo numerosi spunti creativi per la creazione di pagine. Nel nostro progetto, l'abbiamo impiegato principalmente per lo sviluppo delle pagine principali, consultandolo su come integrare al meglio un requisito specifico in una pagina, o su come implementare determinate funzionalità e quali approcci adottare per ottenere i risultati desiderati. In particolare, abbiamo chiesto consigli su quali altre funzionalità potesse avere una pagina, come ad esempio l'aggiunta di un meccanismo per segnalare se l'ultimo messaggio in una chat è stato letto o ancora no.

* **Implementazione delle funzionalità**

ChatGPT ha fornito un grande aiuto nell’implementazione delle funzionalità, assistendo nella scrittura del codice per particolari requisiti complessi da implementare. Ad esempio, ha supportato l’integrazione dell’ImagePicker per la selezione delle immagini, la realizzazione di un meccanismo di caricamento progressivo degli elementi nella homepage, e l'implementazione di un sistema di ricerca degli item in base al loro nome, permettendo di cercare una stringa presente nel nome o nella descrizione. Inoltre, è stato molto utile nel calcolo della distanza tra utenti e oggetti. In quest'ultimo caso, dopo la richiesta, ChatGPT ha generato in linguaggio Dart la funzione di Haversine per calcolare la distanza tra due punti dati la loro latitudine e longitudine su un asse terrestre: Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 4.1: Codice rappresentante la formula di Haversine

* **Design di pagine**

ChatGPT è stato poi utilizzato per velocizzare il design delle pagine, permettendo tramite una semplice descrizione di costruire lo scheletro di una pagina, inserendo ogni widget necessario. Tuttavia, nonostante l’aiuto nell’inserimento degli elementi, che consente di risparmiare molto tempo, GPT e gli LLM mancano di sensibilità nel design effettivo. Infatti, dopo aver costruito la pagina con il codice generato, anche se non vi sono errori di formattazione o di elementi sovrapposti, c'è una totale mancanza di gusto e di design, generando una pagina che, dal punto di vista della UI e UX, risulta esteticamente poco piacevole. Questo comporta, nonostante l’aiuto nel trovare e inserire i widget necessari, una certa dose di lavoro per riuscire a dare una estetica piacevole alla pagina. Tuttavia, l’aiuto fornito nella costruzione della pagina contribuisce a risparmiare molto tempo, specialmente se non si conoscono bene i widget del linguaggio Flutter, rendendolo quindi molto utile per questo tipo di compiti, nonostante il punto di vista creativo.

* **Generazione di codice**

Un'altra funzione molto utilizzata di ChatGPT, simile alla precedente nel design delle pagine, è la generazione di codice data una domanda specifica. Per risparmiare tempo, abbiamo spesso sfruttato ChatGPT per generare codice da copiare, nonostante questa sia una pessima pratica. Tuttavia, è stato fatto solo per quei codici che erano stati scritti molteplici volte e che, quindi, risultavano tediosi e ripetitivi. Per trasformare questa pratica da cattiva a ottima, è essenziale comprendere a fondo il codice prima di copiarlo, in modo da evitare errori e conoscere ciò che si sta scrivendo. Detto questo, posso affermare che la generazione di codice da parte di ChatGPT, come ad esempio cicli per il riempimento di liste o creazioni di *GridView* o *ListView* in Flutter, aiuta a risparmiare molto tempo, essendo questi codici scritti un gran numero di volte.

* **Ottimizzazione di codice**

Un utilizzo in cui ChatGPT eccelle particolarmente è l’ottimizzazione del codice. Grazie a questa capacità, è possibile fornire a ChatGPT un codice che verrà ottimizzato rimuovendo bad practice o errori di sintassi, consentendo così di risparmiare tempo in futuro nella correzione di errori. Di seguito, viene riportato un esempio di una funzione ottimizzata da ChatGPT, il cui compito è semplicemente caricare un oggetto di noleggio in Firebase Database e la sua immagine su Firebase Storage:

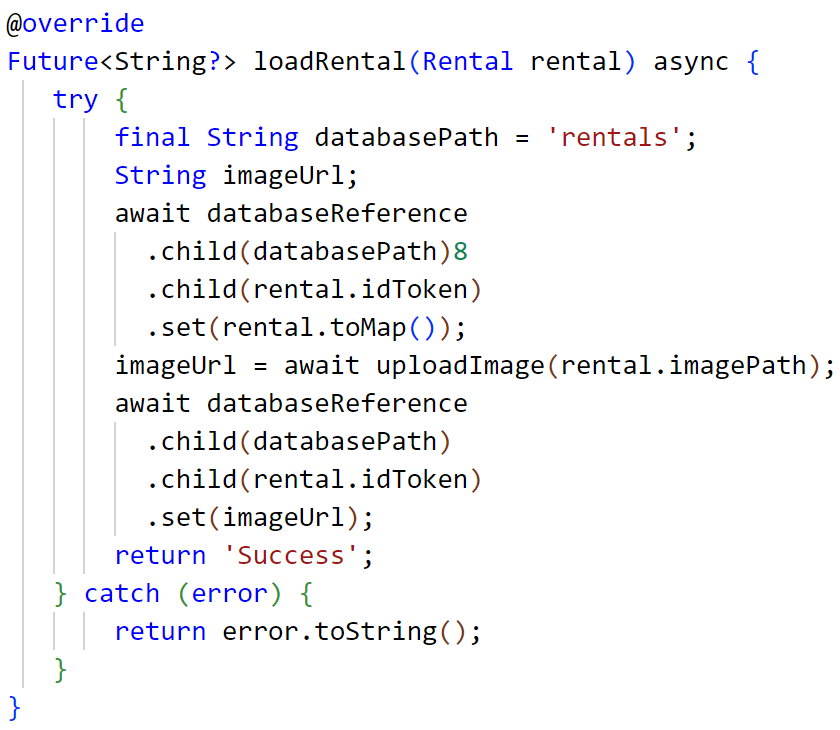


Figura 4.2: Funzione di caricamento del noleggio prima dell'ottimizzazione

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Figura 4.3: Funzione di caricamento del noleggio dopo l'ottimizzazione

Da come si può notare dalla funzione prima e dopo l’elaborazione, ChatGPT ha contribuito significativamente all’ottimizzazione del codice. Ha rimosso il codice ridondante, spostandolo in altre funzioni e ha aggiunto un miglior controllo sugli errori, rendendo il codice più pulito ed efficiente.

* **Correzione di errori**

Infine, un utilizzo in cui ChatGPT è stato impiegato in maniera estensiva è la correzione di errori. Fornendo il codice e la stampa dell’errore in runtime, ChatGPT offre una lista di possibili cause per il malfunzionamento del codice. In particolare, avendo accesso al codice, ChatGPT correggeva i punti che avrebbero potuto generare quegli errori. Tuttavia, quando non si tratta di errori di sintassi o di errori semplici da risolvere, affidarsi a ChatGPT in maniera estensiva potrebbe risultare controproducente. Dopo diverse correzioni sullo stesso codice, nel caso non riesca a risolvere l’errore, ChatGPT potrebbe iniziare a mostrare allucinazioni[[1]](#footnote-1), indicando errori che non esistono e modificando il codice in modo errato. Questo rende il codice sempre più problematico e lontano dalla soluzione desiderata. Nonostante ciò, quando ci si trova di fronte a problemi difficili da risolvere, utilizzare ChatGPT come supporto piuttosto che come unica risorsa può decisamente aiutare a identificare il problema, facilitando così la risoluzione di problemi ostici.

* + 1. **Vantaggi e limiti**

L'utilizzo di ChatGPT nello sviluppo dell'applicazione ha mostrato diversi vantaggi significativi. Uno dei principali benefici è stata la capacità di ottimizzare il codice esistente, rendendolo più efficiente e meno soggetto a errori grazie alla rimozione di pratiche di programmazione non ottimali e all'introduzione di migliori controlli sugli errori. Questo ha comportato un risparmio di tempo e un miglioramento della qualità del codice. ChatGPT si è rivelato utile anche nella generazione di codice ripetitivo e nella creazione di strutture base delle pagine, permettendo una maggior concentrazione sulle funzionalità specifiche e meno sulle operazioni di base. Inoltre, la capacità di ChatGPT di fornire suggerimenti di design e funzionalità stimolando la creatività, ha contribuito a sviluppare un'interfaccia utente più completa e funzionale. Tuttavia, l'uso di ChatGPT ha anche evidenziato alcuni limiti. La generazione di codice, seppur utile, può diventare una cattiva pratica se non viene accompagnata da una piena comprensione del codice prodotto, portando a possibili problemi di manutenzione futura. Inoltre, la correzione degli errori, se affidata completamente a ChatGPT, può diventare problematica. In casi complessi, il modello può produrre allucinazioni, complicando ulteriormente la risoluzione dei problemi. Questo indica che, sebbene ChatGPT possa essere un prezioso strumento di supporto, non può sostituire completamente l'intervento umano, soprattutto nelle fasi critiche di debug e ottimizzazione del codice.

* 1. **Testing**
     1. **Ruolo di ChatGPT nel Testing**

 Spiegazione di come ChatGPT è stato utilizzato per supportare la fase di testing dell'applicazione.

 Discussione dei problemi affrontati nel testing e come ChatGPT ha contribuito (o non contribuito) a risolverli.

* + 1. **Vantaggi e limiti**

 Valutazione dei benefici ottenuti dall'utilizzo di ChatGPT in questa fase.

 Discussione sui limiti riscontrati e sulle aree di miglioramento.

* 1. **Conclusione e statistiche**
     1. **Sintesi dell’utilizzo di ChatGPT**
     2. **Statistiche sull’utilizzo di ChatGPT**
     3. **Valutazione Globale dell’Utilità**

1. "Allucinazione" negli LLM si riferisce alla produzione di informazioni errate o fuorvianti, dove il modello genera risposte che sembrano plausibili ma non sono basate su dati reali o verificati. [↑](#footnote-ref-1)