**ITERAZIONE 2**

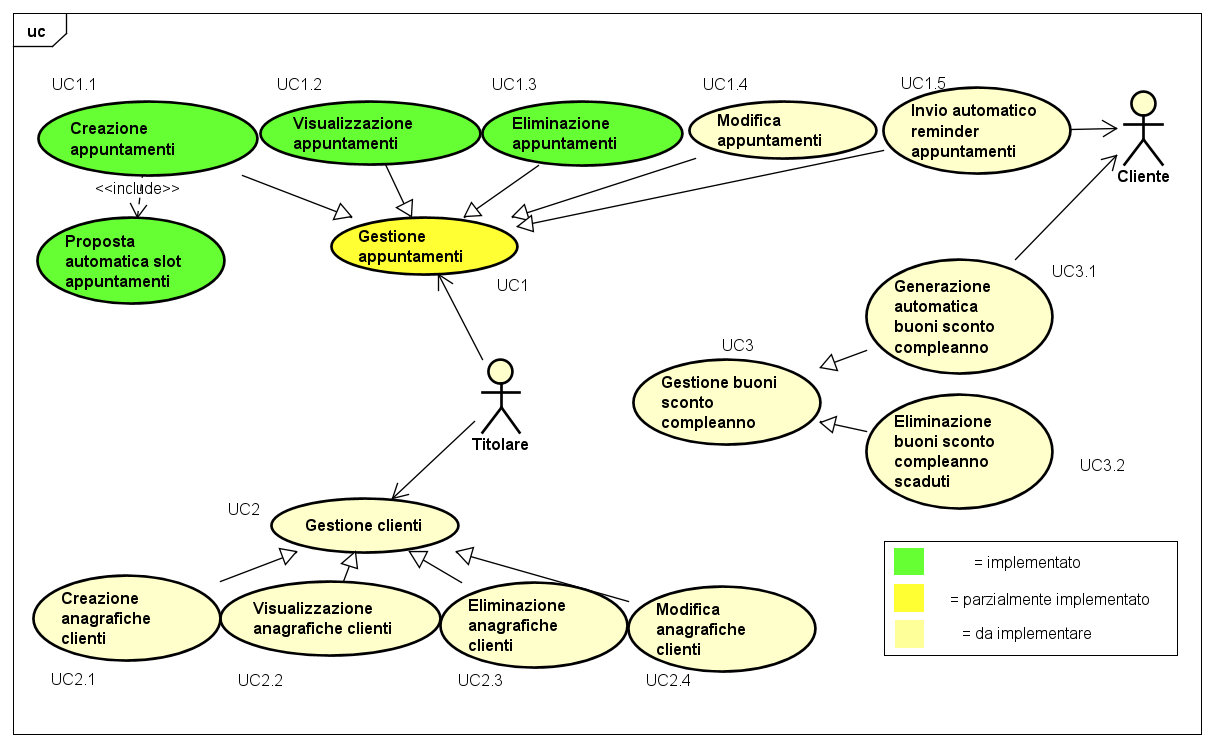
**INTRODUZIONE**

Nella seconda iterazione si è deciso di implementare i seguenti casi d’uso:

* UC1: Gestione appuntamenti
* UC1.1: Creazione appuntamenti;
* UC1.2: Visualizzazione appuntamenti;
* UC1.3: Eliminazione appuntamenti;

Dal diagramma sottostante è possibile analizzare l’evolversi dello stato del progetto rispetto all’iterazione precedente, in particolare si può notare i nuovi casi d'uso implementati di

colore verde, mentre la restante parte non evidenziata rispecchia l’architettura da implementare nella prossima iterazione.



**UC1.1: CREAZIONE APPUNTAMENTI**

*Descrizione*: il titolare deve poter essere in grado di creare nuovi appuntamenti fissati con i clienti: questo comprende avere la possibilità di inserire i dati relativi non solo al cliente ma anche all’appuntamento stesso.

È fondamentale conoscere l’orario di inizio e la tipologia dei trattamenti svolti, al fine di stimare la durata di ogni appuntamento per verificare la disponibilità di eventuali slot temporali in cui inserire il nuovo cliente.

*Attori coinvolti*: titolare, trigger temporale.

**UC1.2: visualizzazione APPUNTAMENTI**

*Descrizione*: il titolare deve poter essere in grado di visualizzare tutti gli appuntamenti: questo comprende avere la possibilità di consultare giorno per giorno il calendario per verificare il numero di clienti previsti per quel determinato giorno, con lo scopo di gestire al meglio le risorse **(sia materiali sia umane)** del saloon, nonché facilitare la verifica di eventuale disponibilità in caso arrivi un nuovo cliente.

*Attori coinvolti*: titolare, trigger temporale.

**UC1.3: eliminazione APPUNTAMENTI**

*Descrizione*: il titolare deve poter essere in grado di eliminare appuntamenti fissati in precedenza con i clienti: questo comprende avere la possibilità di cancellare da calendario un determinato appuntamento e liberare quindi uno slot temporale che potrebbe essere assegnato ad un nuovo cliente.

*Attori coinvolti*: titolare, trigger temporale.

**PSEUDOCODICE GESTIONE APPUNTAMENTI**

Per la gestione degli appuntamenti, invece della durata, si potrebbe passare l'elenco dei trattamenti richiesti per l'appuntamento da schedulare e aggiungere un ciclo for per il calcolo della durata in questa funzione:

propostaSlotAppuntamento(Appuntamento[] appuntamenti, int durata) DateTime

ordinaCrescente(appuntamenti, dataOra) // ordino appuntamenti per prossimità

proposta adesso // se c'è spazio, faccio venire il cliente ora

for Appuntamento a in appuntamenti

// se quello che diventerà l'appuntamento successivo a questo inizia dopo che questo finisce e questo inizia prima della chiusura del salone, termino la ricerca: ho il primo slot libero più recente.

if a.dataOra >= proposta + durata && proposta <= ORARIO\_CHIUSURA

break // altrimenti riparto a cercare dal momento in cui il prossimo appuntamento termina

proposta a.dataOra

for Trattamento t in a.trattamenti

proposta proposta + t.durata

return proposta

**TESTING**

Per l’iterazione 2 è stato effettuato il test delle funzioni descritte nei nuovi casi d’uso implementati.

I risultati sono riportati di seguito:

**SPAZIO PER IMMAGINI TESTING WHATSONSALON**