**ITERAZIONE 3**

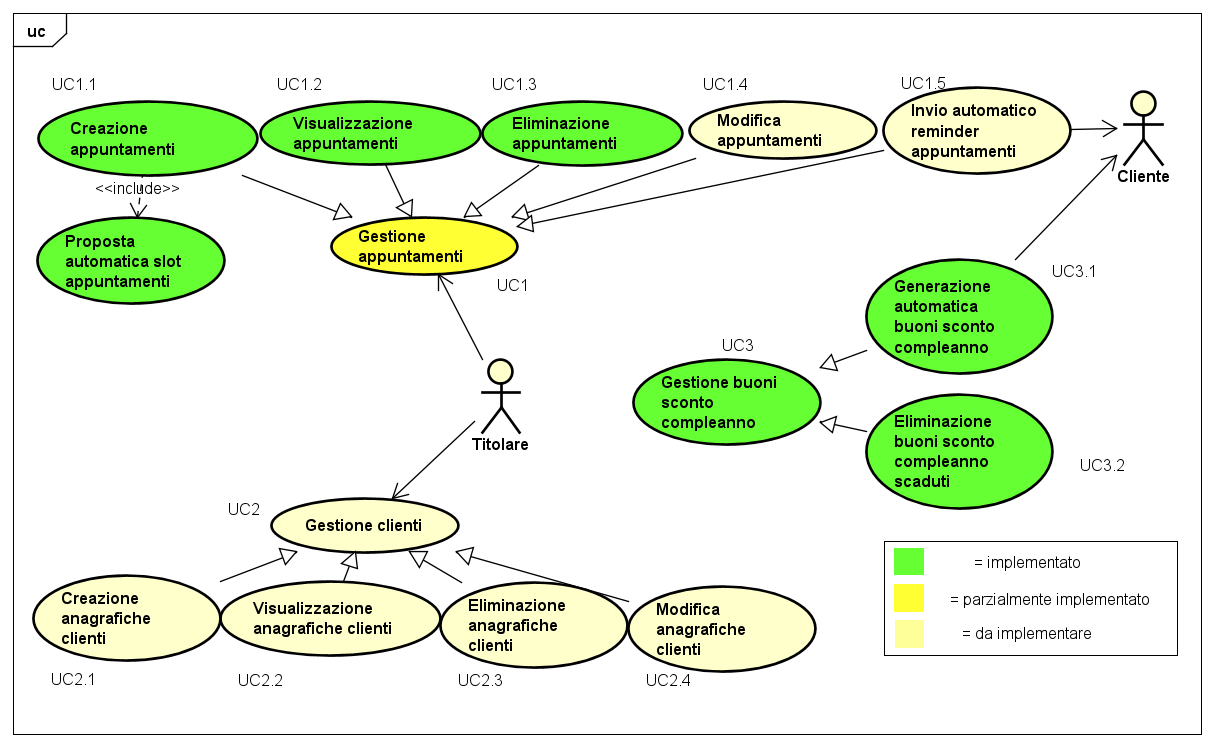
**4.1 CASI D’USO SELEZIONATI**

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

* UC3: Gestione buoni sconto compleanno
* UC3.1: Generazione automatica buoni sconto compleanno;
* UC3.2: Eliminazione automatica buoni sconto compleanno scaduti;

Oltre all’implementazione dei suddetti UC, sono state apportate migliorie di entità minima ma di enorme utilità all’interfaccia grafica di WhatsOnSalon.

Dal diagramma dei casi d’uso sottostante è possibile analizzare l’avanzamento del progetto facendo riferimento alla legenda in basso a destra.



UC3.1: Generazione automatica buoni sconto compleanno

*Descrizione*: in occasione dei compleanni dei clienti, il sistema deve generare automaticamente buoni sconto utilizzabili entro due mesi dalla data del compleanno e inviarli.

Oltre alla scadenza, deve essere associato ad ogni buono anche un codice univoco.

*Attori coinvolti*: trigger temporale.  
  
*Trigger:* arrivo della mezzanotte.  
  
*Postcondizione:* il database dei buoni sconto conterrà un nuovo buono sconto della valenza di due mesi per ogni cliente che compie gli anni nella giornata odierna.  
  
*Procedimento:* 1. allo scatto della mezzanotte, l’algoritmo di gestione buoni sconto viene lanciato;  
 2. l’algoritmo recupera gli \_id dei clienti che compiono gli anni quel giorno;  
 3. per ogni \_id recuperato, viene generato un buono sconto di codice univoco della valenza di due mesi (a partire dal giorno stesso);  
 4. per ogni buono sconto generato, viene inviato un messaggio di auguri e di notifica generazione buono al numero di telefono del cliente per cui è stato emesso;  
  
UC3.2: Eliminazione automatica buoni sconto compleanno scaduti *Descrizione*: il sistema deve eliminare automaticamente i buoni sconto scaduti non appena possibile.  
  
*Attori coinvolti*: trigger temporale.  
  
*Trigger:* arrivo della mezzanotte.  
  
*Postcondizione:* il database dei buoni sconto conterrà solo buoni non ancora scaduti.  
  
*Procedimento:* 1. allo scatto della mezzanotte, l’algoritmo di gestione buoni sconto viene lanciato;  
 2. l’algoritmo confronta la data odierna con la data di scadenza di ogni buono sconto nell’apposito database, andando a eliminare i buoni che risultano scaduti;  
  
Migliorie all’interfaccia grafica di WhatsOnSalon

* possibilità di ricerca cliente per nome e/o cognome durante la creazione di un appuntamento;
* predisposizione per l’aggiunta di nuovi clienti al database durante la creazione di un appuntamento – pulsante che sarà programmato per questo scopo nelle iterazioni successive;
* aggiunta numero di telefono del cliente tra le informazioni mostrate nel riepilogo degli appuntamenti in programma;
* piccole variazioni nella presentazione grafica dell’app;

**4.2 PSEUDOCODICE**

UC3: Gestione buoni sconto compleanno

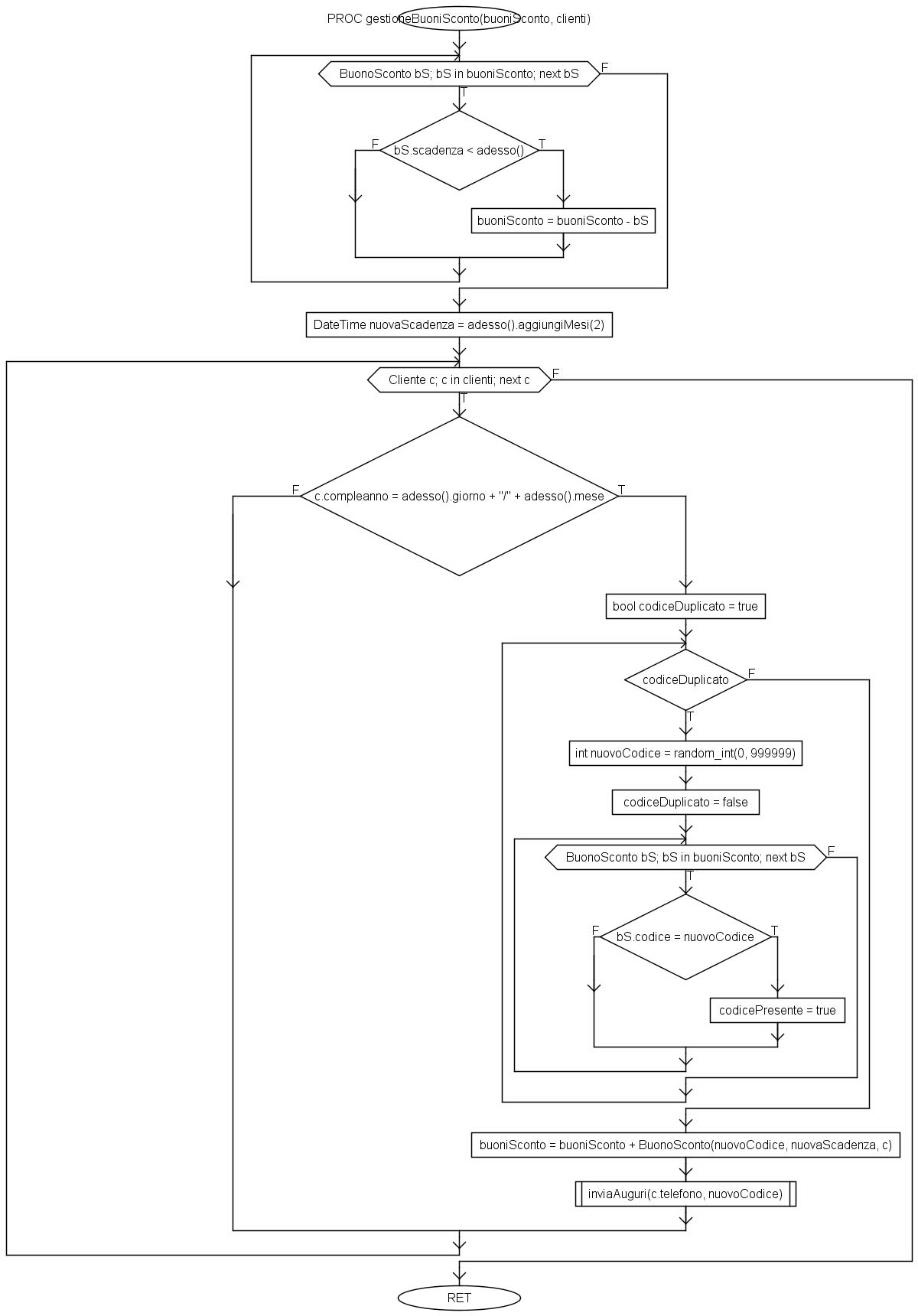
Immagine che contiene testo

Descrizione generata automaticamente

La complessità dell’algoritmo è Ω(nBuoniSconto), caso in cui non ci sono né buoni da cancellare né buoni da generare e si esegue solo il primo for, e O(nBuoniSconto + nClienti), caso in cui - indipendentemente dal numero di buoni sconto da eliminare – tutti i clienti compiono gli anni e bisogna generare un buono per ognuno, dando così luogo a nClienti iterazioni del secondo for.

Nell’implementazione vera e propria i for che scandagliano i database di buoni sconto e clienti sono sostituiti da API call tramite il Node.js MongoDB driver, che hanno complessità inferiore e abbassano dunque la complessità teorica dell’algoritmo.

Si veda di seguito il flow chart dell’algoritmo.



**4.3 RISULTATI IMPLEMENTAZIONE**

Tutto ciò che riguarda l’UC3 non ha alcun impatto sull’interfaccia grafica di WhatsOnSalon né introduce alcun nuovo ambiente accessibile da parte del titolare: tutto è gestito automaticamente e mostrare i risultati è più di interesse della fase di testing che della descrizione della user experience.

Per quanto riguarda le migliorie all’UI di WhatsOnSalon, che invece non necessitano di testing dato che non cambiano il comportamento dell’app, ecco i risultati (non molto diversi da quelli osservabili nell’iterazione 2).



**4.4 TESTING**

Per testare il corretto funzionamento di gestioneBuoniSconto, si è partiti dall’analisi statica della sua implementazione Node.js tramite l’apposito tool codelyzer, che non ha evidenziato warning o errori nel codice.

Immagine che contiene testo

Descrizione generata automaticamente

Con la sicurezza che il codice rispettasse buoni standard, si è potuto procedere con l’analisi dinamica dello stesso.   
Innanzitutto, è stato creato un ambiente di test, così strutturato (data test: 07/03/2022):

* database buoni sconto avente sia buoni sconto scaduti che buoni sconto validi, con copertura casi limite:
  + scadenza = data test (dovrebbe essere eliminato);
  + scadenza = data test + 1 giorno (non dovrebbe essere eliminato);

Immagine che contiene testo

Descrizione generata automaticamente

* database clienti avente sia clienti che compiono gli anni nella data di testing che clienti che nati in altre date (sempre coprendo i due casi limite sopra enunciati);

Immagine che contiene testo

Descrizione generata automaticamente

* trigger temporale che simuli lo scatto della mezzanotte;  
  Immagine che contiene testo

  Descrizione generata automaticamente

Idealmente, i risultati avrebbero dovuto essere i seguenti:

* database buoni sconto popolato dai buoni sconto validi tra quelli precedentemente esistenti e da quelli generati per ogni cliente festeggiato nella data di testing;
* database clienti invariato;
* trigger temporale che innesca esattamente un’esecuzione del codice;

Di fatto, così è stato:

* Immagine che contiene testo

  Descrizione generata automaticamente
* Immagine che contiene testo

  Descrizione generata automaticamente
* (tempo di esecuzione: 1364.23ms)  
  Immagine che contiene testo

  Descrizione generata automaticamente

Si può dunque concludere che lo sforzo iterativo è andato a buon fine e ci si può organizzare per intraprendere l’iterazione successiva.

PER BIBLIOGRAFIA

http://codelyzer.com/