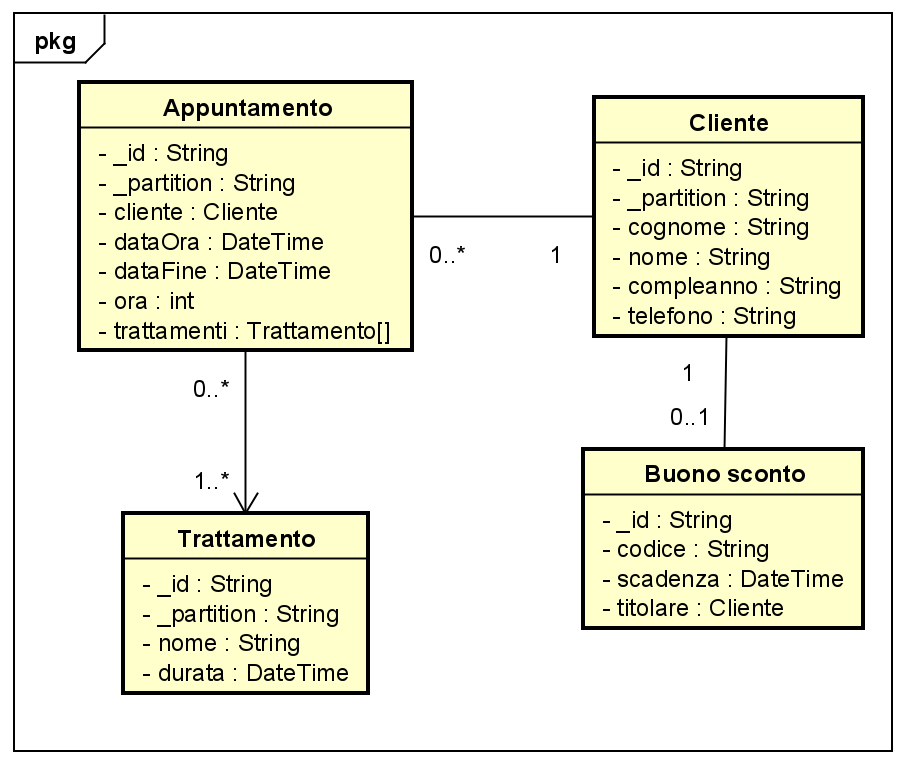
**ITERAZIONE 2**

**INTRODUZIONE**

Durante la preparazione per l’implementazione dei casi d’uso descritti nel paragrafo successivo si è resa necessaria una variazione all’interno del data model, causata dalla necessità di Realm di avere un campo \_partition in ogni documento nel database per poterlo recuperare dall’app.

Qui di seguito è riportato il data model che caratterizzerà l’elaborato da qui in avanti.

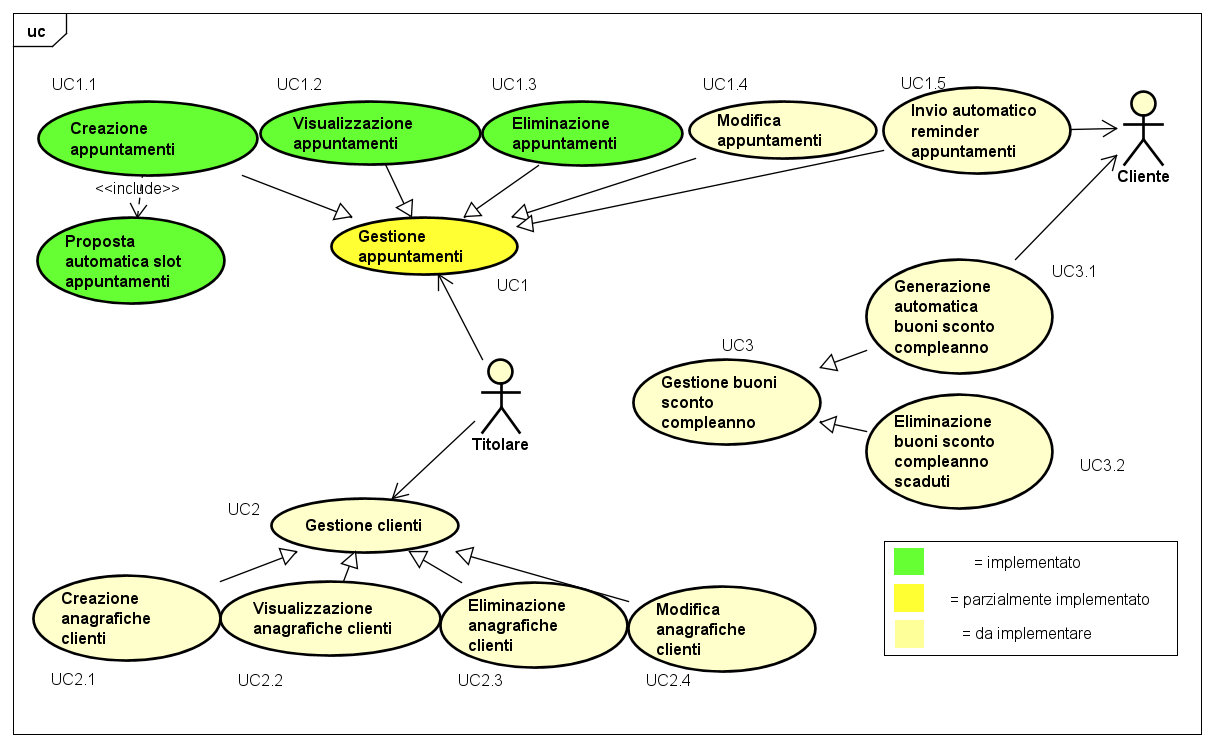


**CASI D’USO SELEZIONATI**

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

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

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



UC1.1: Creazione appuntamenti *Descrizione*: il titolare deve poter essere in grado di fissare appuntamenti sul “calendario virtuale” fornito dall’app, dettagliando per ogni appuntamento cliente interessato (che deve essere inseribile a sistema sul momento, nel caso in cui non fosse già inserito), trattamenti previsti e data e ora nelle quali l’appuntamento è fissato, specificando queste ultime manualmente oppure scegliendole tra le proposte fornite dal sistema.

*Attori coinvolti*: titolare, algoritmo di proposta automatica slot appuntamenti.   
  
*Trigger:* richiesta di creazione appuntamento previa specifica dei dati ad esso relativi?.  
 *Postcondizione:* appuntamento avente i dati specificati presente sull’apposito database. *Procedimento:* 1. il titolare apre l’interfaccia di creazione appuntamenti;  
2. il titolare seleziona cliente interessato (eventualmente inserendone l’anagrafica a sistema, nel caso in cui questo non fosse già registrato), i trattamenti previsti e la data e l’ora in cui l’appuntamento è previsto, selezionandole manualmente o scegliendo una delle proposte provenienti dall’algoritmo di proposta;  
 3. il titolare richiede la creazione dell’appuntamento appena dettagliato;  
 4. il titolare viene riportato alla vista del calendario appuntamenti;

UC1.2: Visualizzazione appuntamenti *Descrizione*: il titolare deve poter essere in grado di visualizzare in un formato agile tutti gli appuntamenti relativi a un giorno da lui indicato.

*Attori coinvolti*: titolare.  
  
*Trigger:* selezione data.  
  
*Postcondizione:* stampa a video appuntamenti previsti per la data selezionata in un formato facilmente interpretabile.  
  
*Procedimento:* 1. il titolare seleziona una data in un’interfaccia stile calendario;  
 2. l’interfaccia viene riempita con gli appuntamenti previsti per la data selezionata, in un formato facilmente interpretabile;

UC1.3: Eliminazione appuntamenti *Descrizione*: il titolare deve poter essere in grado di eliminare appuntamenti fissati in precedenza.

*Attori coinvolti*: titolare.  
  
*Trigger:* richiesta di eliminazione di uno o più appuntamenti precedentemente selezionati.  
  
*Postcondizione:* appuntamenti di cui è stata richiesta l’eliminazione non più presenti nel database.  
  
*Procedimento:* 1. *Procedimento UC1.2* + checkbox di selezione per ogni appuntamento;  
 2. il titolare seleziona gli appuntamenti che intende eliminare;  
 3. il titolare richiede l’eliminazione degli appuntamenti selezionati;  
 4. il titolare viene riportato alla vista del calendario appuntamenti;

**PSEUDOCODICE**UC1.1: Creazione appuntamenti

Immagine che contiene testo

Descrizione generata automaticamente

La complessità dell’algoritmo è Ω(nSlot), caso in cui tutte le proposte vengono trovate al primo tentativo e O(nSlot \* nAppuntamentiFuturi), caso in cui viene raggiunta una prossimaProposta successiva alla fine dell’ultimo appuntamento futuro solo nel corso dell’ultima iterazione del while.

**quando aggiungi il flow chart formatta sotto di modo che sia tutto ordinato – sopra l’è tot apost dio carlo**

**dai che am ciapa u bel vintot**

**RISULTATI implementazione**



**TESTING**

**TING**