**AGGIORNAMENTO: penso che dovremmo mandare in persistenza le modifiche appena le facciamo. Visto che alcune operazioni richiedono il database mentre altre no potremmo avere problemi di consistenza.**

**COSA MANCA AL CLIENTE?**

* RISORSE:

1. thread lanciato una sola volta all'accensione dell'applicazione che cambia lo stato dei farmaci se sono scaduti (da utilizzabile a scaduto). Serve per poter spostare i farmaci scaduti in riciclaggio
2. opzione per vedere la disponibilità in farmacia
3. opzione per cercare le farmacie vicine con maps
4. trasforma il thread del punto 1 in un semplice metodo (elisa)

* RICICLAGGIO:

1. bottone RICICLA 🡪 cambia lo stato del farmaco (il cui riferimento è presente nella sessione) da scaduto a smaltito e va subito a scrivere sul db questa modifica. Inoltre va a chiamare l’update delle viste perché questo farmaco non deve più comparire in “Ricicla i tuoi articoli” ma in “storico ricicli” (elisa)
2. bottone PRENOTA RITIRO 🡪 va implementato tutto il caso d’uso. Come fare?
3. Mettere un pulsante di REFRESH in storico ricicli (grafica in fxml). Per l’implementazione vedi il punto 4
4. Bottone di REFRESH 🡪 va a vedere se questo farmaco nel db ha stato “verificato”. (elisa)
5. Se il farmaco è stato verificato incrementa i punti dell’utente (i punti sono presenti nella sessione) e chiamare l’update delle cose cambiate (classe utente e classe farmaco).

* ACCOUNT:

1. il pulsante di logout porta le cose in persistenza

* EVENTI:

1. Se non hai eventi dà errore perché fa null.size() va cambiato
2. In “Tutti gli eventi attivi” dovremmo mostrare anche gli eventi in corso
3. Bottone di REFRESH 🡪 da mettere nella pagina “Tutti gli eventi attivi”. Va a chiedere al database quali sono gli eventi attivi ed è necessario perché l’utente non ha questi dati nella sessione ma gli vengono forniti da un altro utente.
4. Per tutti gli eventi a cui partecipo va implementato un modo per automatizzare il cambio di stato. Poiché gli eventi non cambiano più di una volta al giorno forse basta un metodo che va ad aggiornare lo stato quando si crea la sessione (come abbiamo fatto per aggiornare lo stato dei farmaci nelle risorse)
5. Bottone JOIN 🡪 aggiunge l’evento scelto alla sessione (e anche alla classe Cliente) e forse chiama l’update?

**COSA MANCA ALLA FARMACIA?**

* RISORSE:

1. Gestire la scadenza e la descrizione (attenzione: se si lascia la scadenza al FARMACOFARMACIA quando un utente va a cercare la disponibilità deve fare la somma delle disponibilità di tutti i medicinali con uno stesso nome)
2. Visualizza solo i farmaci non scaduti (però, visto che nel db ci sono, devi fare in modo che quando un utente va a cercare la disponibilità non conti i pezzi scaduti) (oppure fai in modo che i farmaci scaduti vengano eliminati dal database)
3. Aggiungi la possibilità di cambiare la quantità (come per il cliente)
4. Mostra i dati

* ACCOUNT:

1. il pulsante di logout deve portare le cose in persistenza
2. Mostra i dati

* RITIRO: tutto da implementare (sia pagina che implementazione)
* EVENTI:

1. Bottone CREATE 🡪 crea una nuova classe evento, mettila nella sessione e falla osservare dai controllori grafici.
2. Bottone ELIMINA 🡪 consenti di eliminare gli eventi solo se non sono ancora iniziati
3. Implementa il meccanismo di premiazione (pattern decorator)
4. …?

* GESTIONE:

1. ~~Bottone REFRESH 🡪 da mettere nella pagina ”Autenticazione Riciclaggi” . Va a chiedere al database se ci sono nuove richieste di verifica di riciclo.~~

**(penso che non serva… ogni volta che accediamo alla pagina dobbiamo andare nel db perché non sono dati presenti nella sessione)**

1. Bottone VERIFICA 🡪 prende il farmaco scelto e cambia lo stato da “smaltito” a “verificato”. Infine riporta le modifiche nel database in modo da comunicarlo all’utente. (elisa)
2. Nell’operazione di verifica va calcolato l’incremento di punti dell’utente e, se quest’ultimo raggiunge un nuovo livello, va modificata la sua classe nella sessione (con tutte le conseguenze per gli osservatori)
3. Bottone PREMIA 🡪 implementare il comportamento. Come? Manda un’email?