***Analisi progetto weekend 05/11/2017 Francesco Caniglia***

Per iniziare ho sviluppato il data modeling del contesto. I types e le relazioni individuate sono le seguenti:

***Types:***

* Hospital ;
* HospitalPerson (Types con attributi comuni a Patient e Doctor);
* Patient (estende HospitalPerson)
* Doctor (estende HospitalPerson)
* Ward
* PhysicalExamination

***Relations:***

* Hospital-Ward🡪one-to-many
* Ward-Doctor🡪one-to-many
* Ward-Patient🡪one-to-many
* Patient-PhysicalExamination🡪one-to-many
* Doctor-PhysicalExamination🡪one-to-one
* Doctor-Patient🡪one-to-many

Inoltre sono stati definiti 3 enumtypes:

* genderPerson: MALE/FEMALE🡪HospitalPerson
* examinationResult: POSITIVE/NEGATIVE🡪PhysicalExamination
* recoveryStatus: RECOVERED/QUIT🡪Patient

Ho successivamente proceduto con la realizzazione del Dao. Per rispondere alla richiesta dell’esercizio, in base alla mia struttura dati, ho effettuato due FlaxibleSearchQuery: una richiamando tutte le PhysicalExamination, l’altra richiamando le PhysicalExamination relative però, solo al paziente di competenza, passando come parametro la PK di tale paziente.

Successivamente ho realizzato anche il Service ed il Facade. Per quanti riguarda il Facade, realizzato in una seconda estensione hospitalfacade, ho sopportato il suo funzionamento con la scrittura di classi Data e Converter/Populator appropriate, usabili per una eventuale implementazione di un Controller.

Ho gestito semplicemente l’attributo dinamico , sfruttando il metodo *Days.daysBetween(enterDate, escDate).getDays(),* nel caso in cui sia presente una escDate!=null. Se il paziente invece ancora non è stato dimesso, quindi escDate==null, setto l’attributo days come i giorni intercorsi tra la data di ricovero all’attuale.

Come richiesto dall’esercizio, ho realizzato un CronJob che stampa attraverso un LOG settato a livello info, tutte le visite prenotate per il giorno successivo. Il Trigger del CronJob è settato alle 23.30 di ogni giorno.

Successivamente ho realizzato le classi Event e Listener. Parametro dell’event è la PK del paziente relativo al PhysicalExaminationModel processato dal ValidateInterceptor. Nel Listener ho implementato una scrittura su File configurando il path dello stesso all’interno del project-properties di competenza.

L’evento è generato da un ValidateInterceptor che verifica se il PhysicalExaminationModel sta venendo salvato sul DB con l’attributo result=POSITIVE. In questa parte però l’applicazione ha un problema: quando effettuo un salvato con attributo POSITIVE dall’hmc, viene lanciata l’eccezione ModelSavingException eseguendo la riga di codice *eventService.publishEvent(****new*** *PhysicalExaminationEvent(pem.getPk()));*

Invio il progetto con questo bug ancora non risolto, ma provvederò a porre rimedio.

***Francesco Caniglia***