Idea generale

Il nostro progetto fa riferimento alla progettazione di un software per antifurti. Il progetto si occupa della realizzazione di un sistema in cui sono presenti delle aree vigilate prestabilite ed un utente che senza essere registrato può inserire uno o più dispositivi all’interno di una di esse. Per rendere più realistico il sistema abbiamo scelto di autenticare l’utente per l’attivazione e il disarmo dell’antifurto. Autenticazione che avviene grazie ad una registrazione iniziale dell’utente al primo utilizzo, registrazione possibile grazie all’implementazione di un database MYSql.

Iterazione 1

Nella prima iterazione abbiamo realizzato il primo caso d’uso che consiste nell’inserimento di un codice identificativo, alcuni dati generali al primo utilizzo e inserimento dei dispositivi in ogni area. Siamo partiti dal realizzare il diagramma UML di tale caso d’uso. Nel modello di analisi abbiamo realizzato il modello di dominio e il diagramma di sequenza del sistema.

Nel modello di dominio la classe Amministratore è collegata tramite l’associazione Usa a VirHome la quale Gestisce un insieme di AreeVigilate. Ogni AreaVigilata contiene ed è controllata da uno o più Dispositivi i quali possono essere Telecamere o Sensori. Sensore varia da telecamera in quanto ha in più l’attributo sensibilità che è impostato inizialmente ad un valore di default pari a 5.

Il diagramma di sequenza del sistema (SSD) comprende l’amministratore e il sistema (VirHome). Per realizzare il primo caso d’uso sono necessari tre metodi del sistema che verranno utilizzati dall’amministratore quali:

* addUserToDatabase(amministratore)
* InserisciDispositivo(codiceArea, codiceDispositivo)
* confermaInserimento()

Nel modello di progetto abbiamo realizzato il diagramma delle classi, il caso d’uso d’avviamento, i diagrammi di sequenza per ogni metodo implementato nell’ SSD inoltre abbiamo utilizzato il pattern DAO in quanto è presente un database con una tabella user in grado di memorizzare nome, codice, domanda di sicurezza e telefono dell’amministratore registrato.

Nel diagramma delle classi, le classi implementate sono: VirHome, Amministratore, AmministratoreDbDAO che implementa l’interfaccia AmministratoreDAO, AreaVigilata e Dispositivo che a sua volta viene esteso da Telecamera e Sensore. L’idea di base è che VirHome gestisce le aree vigilate dove ogni area al suo interno gestisce un elenco di dispositivi al quale solo l’area può avere accesso. L’inserimento di un database ci ha portato all’utilizzo del pattern DAO, il quale permette di scollegare le classi del dominio dalle classi che si occupano di interfacciarsi al database.

Le relazioni per ogni classe sono le seguenti:

* VirHome usa AmministartoreDbDAO
* AmministratoreDbDAO crea Amministratore
* Virhome possiede un elencoAree di tipo AreaVigilata
* VirHome possiede un’areaCorrente di tipo AreaVigilata
* AreaVigilata possiede un elencoDispositivi di tipo Dispositivo
* AreaVigilata possiede un dispositivoCorrente di tipo Dispositivo

Il caso d’uso d’avviamento si occupa di spiegare il comportamento di VirHome all’avvio. Virhome crea un elencoAree di tipo AreaVigilata, crea le aree e le aggiunte a tale elenco. Mentre ogni area creerà un elencoDispositivi di tipo Dispositivi. Infine VirHome crea un istanza di AmministratoreDbDAO.

Nel diagramma di sequenza di addUserToDatabase esso viene chiamato in VirHome che a sua volta viene chiamato in AmministratoreDbDAO dove viene creata la connessione al database. Viene successivamente inserito l’utente all’interno di una tabella chiamata User, l’inserimento ritorna un valore booleano. Il secondo diagramma di sequenza prevede l’inserimento di un dispositivo (telecamera o sensore a scelta) in una specifica area vigilata. Viene cercata l’area tra l’elenco delle aree il quale compito sarà quello di creare un nuovo oggetto di tipo dispositivo. Il terzo diagramma di sequenza contiene la conferma inserimento il quale compito è quello di inserire il dispositivo corrente all’interno della mappa elencoDispositivi posseduta dall’areaCorrente.

Per l’implementazione abbiamo utilizzato i pattern GRASP creator, low copling, information expert, controller (in modo tale che ogni chiamata dalla Ui venga gestita da VirHome), high coesion. Abbiamo anche utilizzato il pattern GoF Singleton applicato per impedire la creazione di più istanze di VirHome.