

Architettura del sistema ed estensione dell'analisi di dominio

Si vuole progettare l'architettura logica del sistema, che deve essere strutturata a strati e partizioni. Riguardo agli strati, l'architettura logica deve avere tre strati:

1. uno strato UI che deve realizzare esclusivamente l'interfaccia utente del sottosistema;
2. uno strato Domain che deve realizzare esclusivamente la logica applicativa; e
3. uno strato Services che deve realizzare esclusivamente i servizi tecnici di persistenza e comunicazione distribuita.

L'architettura deve essere a strati stretta, pertanto le uniche dipendenze tra strati che possono essere presenti sono le seguenti: lo strato UI deve dipendere solo dal livello Domain, e il livello Domain deve dipendere solo dal livello Services. Il livello Services infine dipende dalle librerie Java per la persistenza dei dati (JDBC) e per la comunicazione distribuita (RMI). Non sono ammesse altre dipendenze verticali. Sono ammesse dipendenze orizzontali tra i package all'interno di uno stesso strato. Non sono in alcun caso ammesse dipendenze circolari tra package dell'architettura logica.

Per la divisione in partizioni del livello Domain si tenga conto del fatto che il sistema deve essere costituito da quattro sottosistemi:

1. Un server centralizzato, sul quale vengono memorizzati i test clinici (pilota e definitivi);
2. Un client dedicato agli esperti di disturbi del linguaggio, che collegandosi al server deve permettere ad un esperto di accedere ai test clinici pilota e definitivi da lui definiti e di effettuarne l'editing, e di accedere alle risposte ai test da lui definiti e di aggiungervi la valutazione clinica;
3. Un client dedicato ai somministratori, che permette di registrare i dati anagrafici di un soggetto, scegliere uno tra i test clinici definitivi memorizzati sul server, somministrare il test al soggetto, e che alla fine invia le risposte al test del soggetto al server, dove queste vengono memorizzate;
4. Un altro client dedicato ai somministratori, che permette ad un somministratore di vedere l'elenco dei test da lui somministrati, con le informazioni del soggetto che ha risposto, e se per tali test somministrati sono pronte le valutazioni cliniche, permette al somministratore di leggerle.

Il server (1) sarà deployato su un computer con una connessione di rete permanente; i client (2) e (4) verranno deployati sui laptop computer degli attori, e il client (3) sarà deployato su tablet. Chiarite per ogni partizione in quale sottosistema (o quali sottosistemi) debbano andare. Inoltre dalla fase di analisi si è determinato che il caso d'uso di creazione di un test da parte dell'esperto di disturbi cognitivi (ideazione, creazione dei test pilota, loro erogazione, selezione di un test pilota come test definitivo e sua pubblicazione) è il caso d'uso di maggiore importanza, e che pertanto occorre focalizzare lo sforzo di progettazione e implementazione su tale caso d'uso. A tale scopo tenete conto che un soggetto risponde ad un test definendo un valore opportuno per gli elementi grafici di input (casella di testo, check box, combo box) che sono presenti nelle slides del test. Notare che un elemento grafico su una slide potrebbe, durante ed alla fine del test, non aver associato alcuna risposta da parte del soggetto.

Cosa dovete fare:

1. Progettate l'architettura logica; dovete produrre un diagramma dei package in cui evidenziate i package degli strati verticali, delle partizioni orizzontali, e le dipendenze strato-strato e partizione-partizione.
2. Estendere casi d'uso, SSD, modello di dominio, e contratti delle operazioni di sistema per supportare completamente il progetto di dettaglio. In particolare, dovete produrre casi d'uso, SSD, classi di dominio, ed operazioni di sistema con i relativi contratti che descrivano la somministrazione di un test. Riguardo alle operazioni di sistema, è sufficiente che descrivate con i contratti la semantica delle operazioni che permettono di dare una risposta per un solo tipo degli elementi grafici (check box, combo box, casella di testo) a vostra scelta.
3. Estendere, se non già fatto, il modello di dominio e i contratti delle operazioni di sistema per includere il fatto che una operazione possa essere permessa o negata in funzione dell'identità del chiamante. A tale scopo definite una nuova classe di dominio, di nome Identity (o Identita), che viene restituita dall'invocazione di una operazione di sistema `login(username: String, password: String): Identity`, e che è parametro di input per ogni operazione di sistema, `op(id: Identity, ...)`, in maniera che il Sistema sappia sempre chi ha invocato l'operazione e possa controllarne i permessi. Estendere il modello di dominio perché rappresenti username e password e i permessi associati ad ogni Identity, e l'Identity dell'autore di un test, o del somministratore di un test. Creare, se non già fatto, un caso d'uso per il login, nel quale un attore effettua il login, invocando l'operazione di sistema corrispondente, e riceve in cambio (se il login ha successo) un'Identity.