

Progetto: sistema per test linguistici

Si vuole sviluppare un sistema per la creazione e la somministrazione di test clinici per individuare disturbi linguistici quali la dislessia e la discalculia. Il sistema sarà usato sia dagli esperti nei disturbi linguistici, i quali lo utilizzeranno per creare i test clinici, sia da medici di base e docenti delle scuole primarie e secondarie, i quali lo utilizzeranno per somministrare i test ai soggetti (bambini e adulti) che si sospetta possano avere un disturbo linguistico dell'apprendimento.

Per progettare un nuovo test un esperto nei disturbi linguistici prima crea un insieme di test pilota, contenenti un certo numero di domande. I test pilota vengono poi somministrati ad un insieme di soggetti privi di disturbi linguistici, vengono raccolte le risposte e, sulla base di queste, viene valutata ciascuna domanda presente all'interno dei test pilota. In base alla valutazione le domande potrebbero venir modificate dagli esperti, ad esempio per renderle più chiare. Quindi le domande modificate vengono aggregate in un nuovo insieme di test pilota, che vengono di nuovo somministrati a soggetti privi di disturbi linguistici, e così via fino a quando non si ritiene che le domande non debbano essere ulteriormente modificate. A questo punto le domande vengono aggregate nei test definitivi. Ogni test clinico viene registrato nel sistema con un nome ed un insieme di parole chiave (keywords) che descrivono il test stesso, ad esempio che patologia tale test valuta. Deve essere possibile ricercare un insieme di test a partire da un insieme di parole chiave. I test clinici vengono memorizzati su un server centralizzato, al quale gli esperti nei disturbi linguistici accedono attraverso un'interfaccia dedicata.

La somministrazione di un test clinico deve avvenire allo stesso modo per i test pilota e per i test definitivi. Chi somministra il test (esperto dei disturbi linguistici, medico di base, docente delle scuole primarie) preparerà un tablet sul quale sarà installato il software che permette di erogare il test: inserirà i dati anagrafici del soggetto al quale il test sarà somministrato, selezionerà il test, che il software sul tablet scaricherà automaticamente dal server centrale, e consegnerà il tablet al soggetto. Questi avrà a disposizione un pulsante di avvio, premuto il quale il test inizia. Un test è costituito da una sequenza di domande, e il soggetto risponderà alle domande nella sequenza proposta. Il software deve registrare sia la risposta data a ciascuna domanda, sia il tempo che ha impiegato per rispondere, a partire dall'istante in cui la domanda è stata visualizzata per la prima volta. Il soggetto può saltare una domanda, e successivamente tornare indietro alla domanda saltata ed inserire una risposta; inoltre può tornare indietro ad una domanda alla quale ha dato risposta allo scopo di modificare la risposta data. Quando il soggetto cambia la domanda correntemente visualizzata e ne visualizza una diversa, il timer della domanda precedentemente visualizzata viene arrestato, mentre quello della nuova domanda visualizzata inizia o riprende a scorrere. Il soggetto può terminare il test in qualsiasi istante premendo un opportuno pulsante di terminazione, o mettere il test in pausa premendo un pulsante di pausa, nel qual caso non viene visualizzata nessuna domanda fino a quando il soggetto non preme il pulsante per terminare la pausa. Ogni test ha un tempo massimo di completamento (timeout). Raggiunto il timeout il software termina automaticamente il test.

Una volta terminato il test, il tablet si collega al server centrale e invia tutti i dati del test somministrato, ossia i dati anagrafici del soggetto, l'identificatore del test eseguito, le risposte e i tempi di risposta per ciascuna domanda. Gli esperti dei disturbi linguistici, tramite la loro interfaccia dedicata, possono visualizzare una somministrazione del test, ossia le domande con le relative risposte e i tempi di risposta, e produrre una valutazione clinica del soggetto. La valutazione clinica è semplicemente un testo nel quale l'esperto fornisce una descrizione dei risultati del test e una eventuale diagnosi di disturbo linguistico. Chi ha somministrato il test può, attraverso un'interfaccia dedicata, visualizzare l'elenco dei test che ha somministrato, e per

ciascun test somministrato visualizzare le risposte e i relativi tempi di risposta, e leggere la valutazione clinica dei test se questa è pronta.

Cosa dovete fare:

1. Produrre i casi d'uso in formato dettagliato per l'applicazione descritta.
2. Produrre un diagramma di attività per la progettazione di un nuovo test clinico.
3. Produrre gli SSD per tutti i casi d'uso.

Analisi di dominio

Considerate la seguente analisi di dominio relativa alla struttura dei test linguistici.

Un test è costituito da una sequenza di sezioni; a sua volta una sezione è costituita da una sequenza di domande. Le sequenze sono ordinate, e l'utente in fase di progettazione del test vedrà le sezioni e le domande in esse contenute secondo l'ordine stabilito. L'utente in fase di progettazione può cambiare l'ordine delle sezioni e l'ordine delle domande in esse, può inoltre spostare una domanda da una sezione all'altra. Una sezione può essere marcata come sezione randomica, nel qual caso in fase di somministrazione del test (ma non in fase di progettazione) le domande della sezione randomica verranno presentate al soggetto in un ordine casuale, possibilmente diverso dall'ordine di progettazione.

Ogni domanda viene rappresentata a schermo da una slide. Le slides hanno tutte la stessa dimensione. All'interno di una slide possono esservi i seguenti elementi grafici:

- Labels: il loro scopo è visualizzare sulla slide un testo fisso.
- Riquadri immagini: il loro scopo è visualizzare sulla slide un'immagine in formato PNG.
- Caselle di testo: il loro scopo è consentire l'input da parte dell'utente di un testo.
- Check boxes: il loro scopo è consentire l'input da parte dell'utente di un'informazione di tipo booleano (si/no); una check box contiene una label, che viene visualizzata insieme alla check box.
- Combo boxes: il loro scopo è consentire l'input da parte dell'utente di un'informazione di scelta di un valore all'interno di un'enumerazione finita di valori; una combo box contiene tante labels quanti sono gli elementi dell'enumerazione, labels che vengono visualizzate insieme alle scelte nella combo box.
- Tabelle: il loro scopo è organizzare altri elementi grafici in una griglia. Una tabella ha un numero fisso di righe e colonne; ciascuna riga e colonna ha una sua altezza. Ogni casella della tabella contiene un altro elemento grafico (label, casella di testo...).

Ogni elemento grafico è delimitato da un rettangolo le cui dimensioni (larghezza e altezza) sono espresse in punti percentuali con tre cifre decimali dopo la virgola rispetto alla dimensione della slide. Ad esempio, un riquadro immagine potrebbe avere larghezza pari al 31,522% della larghezza della slide, ed altezza pari al 100,000% dell'altezza della slide. L'utente deve poter essere in grado di cambiare le dimensioni di un elemento grafico. Per alcuni elementi grafici le dimensioni del rettangolo che li contiene cambiano se l'utente cambia il loro contenuto: ad esempio, la dimensione di un riquadro immagine viene stabilita in base all'immagine che viene messa al suo interno, anche se poi l'utente può cambiare tale dimensione e l'immagine visualizzata viene scalata di conseguenza. Per alcuni elementi grafici (check box e combo box) le dimensioni sono calcolate in automatico a partire dalle dimensioni degli altri elementi grafici che questi contengono al loro interno, e l'utente può agire sulle dimensioni dei contenitori solo modificando le dimensioni di questi ultimi.

L'utente deve poter essere in grado di collocare istanze dei vari elementi grafici su di una slide, mettendoli nella posizione che preferisce. La posizione di un elemento grafico su una slide deve essere descritta in termini della distanza orizzontale e verticale dell'angolo in alto a sinistra del rettangolo che delimita l'elemento grafico rispetto all'angolo in alto a sinistra della slide. Anche in questo caso le distanze devono essere espresse in punti percentuali con tre cifre decimali dopo la virgola rispetto alla larghezza ed alla altezza della slide. Ad esempio, la posizione di un riquadro immagine potrebbe essere il 15,250% della larghezza della slide in distanza orizzontale (ossia partendo dal lato sinistro della slide e andando verso destra), e al 35,951% dell'altezza della slide in distanza verticale (ossia, partendo dal lato superiore della slide e andando verso il basso).

Infine considerate che:

- Per le labels è importante qual è il font del testo, e qual è la dimensione del font (espressa da un singolo numero); quando un utente cambia il font, o la dimensione del font, o il testo da visualizzare il sistema ridimensiona la label, aumentandone l'altezza (non la larghezza!) nel caso in cui la label non sia abbastanza grande da visualizzare tutto il testo con il font e dimensione del font correnti. Similmente quando l'utente cambia manualmente la dimensione di una label, il sistema verifica che la nuova dimensione sia abbastanza grande perché venga visualizzato tutto il testo della label con il font, le dimensioni del font correnti: in caso contrario il sistema rifiuta il ridimensionamento manuale della label. La formula usata in entrambi i casi è la seguente:

$$\text{larghezza_label} * \text{altezza_label} \geq \text{dim_font} * \text{num_lettere_nel_testo} * \text{fattore_font}$$

dove il termine `fattore_font` è un fattore numerico che dipende solo dal tipo di font.

- Le dimensioni di una check box sono sempre calcolate in automatico a partire dalle dimensioni della label contenuta secondo queste formule:

$$\begin{aligned}\text{altezza_check_box} &= \max(\text{altezza_label}, 0,2\%) \\ \text{larghezza_check_box} &= \text{larghezza_label} + 0,2\%\end{aligned}$$

Anche in questo caso il ridimensionamento viene impedito se la nuova dimensione è insufficiente a contenere il testo della label. Se invece la nuova dimensione è sufficiente, anche la label viene ridimensionata corrispondentemente in modo da rispettare queste relazioni.

- Le dimensioni di una combo box sono sempre calcolate in automatico a partire dalle dimensioni delle labels contenute secondo queste formule:

$$\begin{aligned}\text{altezza_combo_box} &= \max(\sum_i \text{altezza_label_i}, 0,2\%) \\ \text{larghezza_combo_box} &= \max_i \text{larghezza_label_i} + 0,2\%\end{aligned}$$

Anche in questo caso il ridimensionamento viene impedito se la nuova dimensione è insufficiente a contenere tutti i testi di tutte le labels. Se invece la nuova dimensione è sufficiente, tutte le labels vengono ridimensionate in modo da rispettare queste relazioni. Il ridimensionamento deve essere effettuato in maniera da mantenere le proporzioni tra le dimensioni delle labels.

- Le dimensioni di una tabella sono sempre calcolate in automatico a partire dalle dimensioni di righe e colonne della tabella, secondo la semplice formula

$$\begin{aligned}\text{altezza_tabella} &= \sum_i \text{altezza_riga_i} \\ \text{larghezza_tabella} &= \sum_i \text{larghezza_colonna_i}\end{aligned}$$

Inoltre l'altezza di una riga deve essere sempre maggiore o uguale alla massima tra le altezze degli elementi grafici nelle caselle che fanno parte di quella riga. Similmente per le larghezze delle colonne.

Il ridimensionamento della tabella, di una riga, o di una colonna, come al solito, deve essere impedito se i precedenti vincoli non vengono rispettati. Notate che gli elementi

grafici nelle caselle di una tabella **non** vengono ridimensionati se la tabella viene ridimensionata. Il ridimensionamento di una tabella avviene mantenendo le proporzioni tra le dimensioni delle righe e le dimensioni delle colonne.

L'utente in fase di editing di un questionario deve poter essere in grado di fare queste operazioni:

- Dividere una sezione in due, unire due sezioni successive (all'inizio nel questionario c'è una sola sezione), spostare una sezione (causa lo spostamento di tutte le slides in esse, preservando il loro ordine all'interno della sezione), cambiare il tipo di una sezione da non randomica a randomica e viceversa;
- Aggiungere, eliminare, duplicare, spostare una slide (anche tra test diversi);
- Aggiungere, eliminare, spostare un elemento grafico in una slide, per tutti i tipi di elementi grafici;
- Ridimensionare un elemento grafico in una slide, tenendo conto dei vincoli sopra espressi, e solo per gli elementi grafici in cui le dimensioni non sono automaticamente calcolate;
- Cambiare testo, font e dimensione font di una label;
- Cambiare l'immagine visualizzata in un riquadro immagine;
- Aggiungere o eliminare righe o colonne da una tabella;
- Cambiare dimensioni a righe o colonne di una tabella;
- Inserire o cancellare un elemento grafico in una casella di una tabella.

Cosa dovete fare:

1. Produrre il modello di dominio.
2. Modificare il caso d'uso che contiene lo scenario in cui un esperto crea un test clinico aggiungendo allo scenario:
 - a. L'inserimento di almeno una sezione;
 - b. Lo spostamento di almeno una slide dalla sua posizione ad un'altra;
 - c. L'inserimento in una slide di almeno tre elementi grafici di tipo diverso;
 - d. Lo spostamento di almeno un elemento grafico;
 - e. Il ridimensionamento di almeno un elemento grafico.
3. Ogni membro del gruppo deve selezionare un elemento grafico diverso e definire per esso quattro operazioni di sistema con i loro contratti; tra le operazioni devono obbligatoriamente esserci l'operazione di aggiunta dell'elemento grafico ad una slide e l'operazione di ridimensionamento dell'elemento grafico.

Architettura del sistema

Si vuole progettare l'architettura del sistema. L'architettura logica 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 l'erogazione di un questionario. 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.

Progetto di dettaglio

In questa ultima fase dovete effettuare il progetto di dettaglio del software. Considerate due operazioni di sistema, tra quelle che avete definito nelle fasi precedenti, per ogni membro del gruppo. Le operazioni di sistema devono far parte del caso d'uso (o dei casi d'uso) che descrivono la creazione di un test, e devono comprendere tanto un certo numero di operazioni di editing del test quanto un certo numero di operazioni di somministrazione di un test.

Nella documentazione della progettazione di dettaglio dovete produrre:

- Uno o più diagrammi delle classi che contengano le classi del software (classi Java);
- Un diagramma di sequenza (o di comunicazione) per ogni operazione di sistema, che parta da una invocazione dell'operazione di sistema ed indichi la successione di chiamate di metodo che viene effettuata per implementare tale operazione;
- L'indicazione dei pattern GRASP utilizzati per assegnare responsabilità alle classi, e degli eventuali pattern GoF applicati.

Riguardo allo strato Domain, raccogliete le operazioni di sistema utilizzando il pattern GRASP use case controller. Riguardo allo strato UI, considerate una semplice interfaccia utente testuale composta da una sola classe `LinguisticTestsTUI`. Riguardo allo strato dei servizi tecnologici, non è necessario che progettiate niente: assumete (temporaneamente) che l'applicazione sia tutta installata ed eseguita su una sola macchina, e che i dati creati non vengano salvati alla fine dell'esecuzione dell'applicazione.

Nel vostro progetto ponete particolare cura di queste cose (ogni deviazione sarà considerata un errore):

1. Il progetto di dettaglio deve essere coerente con l'architettura logica. Le classi devono fare parte di una partizione (può fare eccezione solo la classe `LinguisticTestsTUI`) in uno strato. Se non esiste nell'architettura logica una dipendenza tra una partizione ed un'altra, non devono esserci dipendenze tra le classi della prima partizione e le classi della seconda.
2. Rispettate i ruoli delle partizioni e degli strati: ad esempio, effettuare operazioni di input/output è permesso solo dalle classi nello strato UI (nel nostro caso, dalla sola classe `LinguisticTestsTUI`), e similmente implementare una operazione di sistema è permesso solo dalle classi nello strato Domain.
3. Nella indicazione di quali pattern GRASP o GoF usate specificate anche quali classi e quali operazioni sono coinvolte nel pattern.

Parte facoltativa

implementate in Java le operazioni di sistema progettate nella parte di progetto di dettaglio. La classe `LinguisticTestsTUI` deve proporre un semplice menu testuale che permetta di creare un test, di fare le (poche) operazioni di editing che avete progettato, di marcare un test come definitivo, e di somministrarlo memorizzando le risposte. Ponete estrema attenzione a rendere il codice Java coerente con l'architettura logica e con il vostro progetto di dettaglio: ad esempio, gli attributi e i metodi nelle classi Java devono essere identici in nome e signature ad attributi e operazioni della classe UML corrispondente che avete messo nel progetto di dettaglio, e nel codice di un metodo le chiamate ad altri metodi devono rispettare i diagrammi di sequenza o comunicazione prodotti nel progetto di dettaglio. Ogni deviazione sarà considerata un errore.