



MedQueue

Test Case Integration Document

MedQueue

Sommario

1. [Introduzione](#)
2. [Testing di integrazione](#)
 - 2.1 [Approccio di Integration Testing](#)
 - 2.2 [Componenti da testare per la WebApp](#)
 - 2.3 [Componenti da testare per l'App Desktop](#)
3. [Pass/Fail Criteri](#)

1. Introduzione

Per rilevare eventuali errori ci serviamo del testing, una delle fasi più importanti. Eseguendo il testing ci imponiamo l'obiettivo di testare il maggior numero di funzioni con un'alta frequenza, provando a trovare delle anomalie presenti nel codice.

2. Testing di integrazione

2.1 Approccio di Integration Testing

Utilizziamo come strategia di testing di integrazione "Bottom-up" la quale prevede che si parta dai layer del livello più basso della gerarchia, dopo averli testati singolarmente e vengono testati congiuntamente ai layer di livello superiore (cioè il layer della logica); successivamente si testano i due strati sottostanti con lo strato di presentazione. Procediamo con il testing eseguendo le query, per il layer dei dati, mentre per la logica testeremo i metodi implementati.

2.2 Componenti da testare per la Web App

La scelta della strategia di testing ci suggerisce l'ordine con cui testare le componenti.

Per quanto riguarda il DataAccess:

- AmbulatorioDao
- OperazioneDao
- PrenotazioneDao
- StrutturaDao
- UtenteDao

Per quanto riguarda la logica di business:

- AmbulatorioController
- LoginController
- OperazioneController
- PrenotazioneController
- StrutturaController
- UtenteController
- ViewCoda
- DriverManagerConnectionPool

Per quanto riguarda il Presentation layer:

- Convalida
- Home
- HomeUtente
- Login
- Prenotazione
- Registration
- VisualizzaCoda
- VisualizzaCodaUtente
- VisualizzaPrenotazioni

2.3 Componenti da testare per l'App Desktop

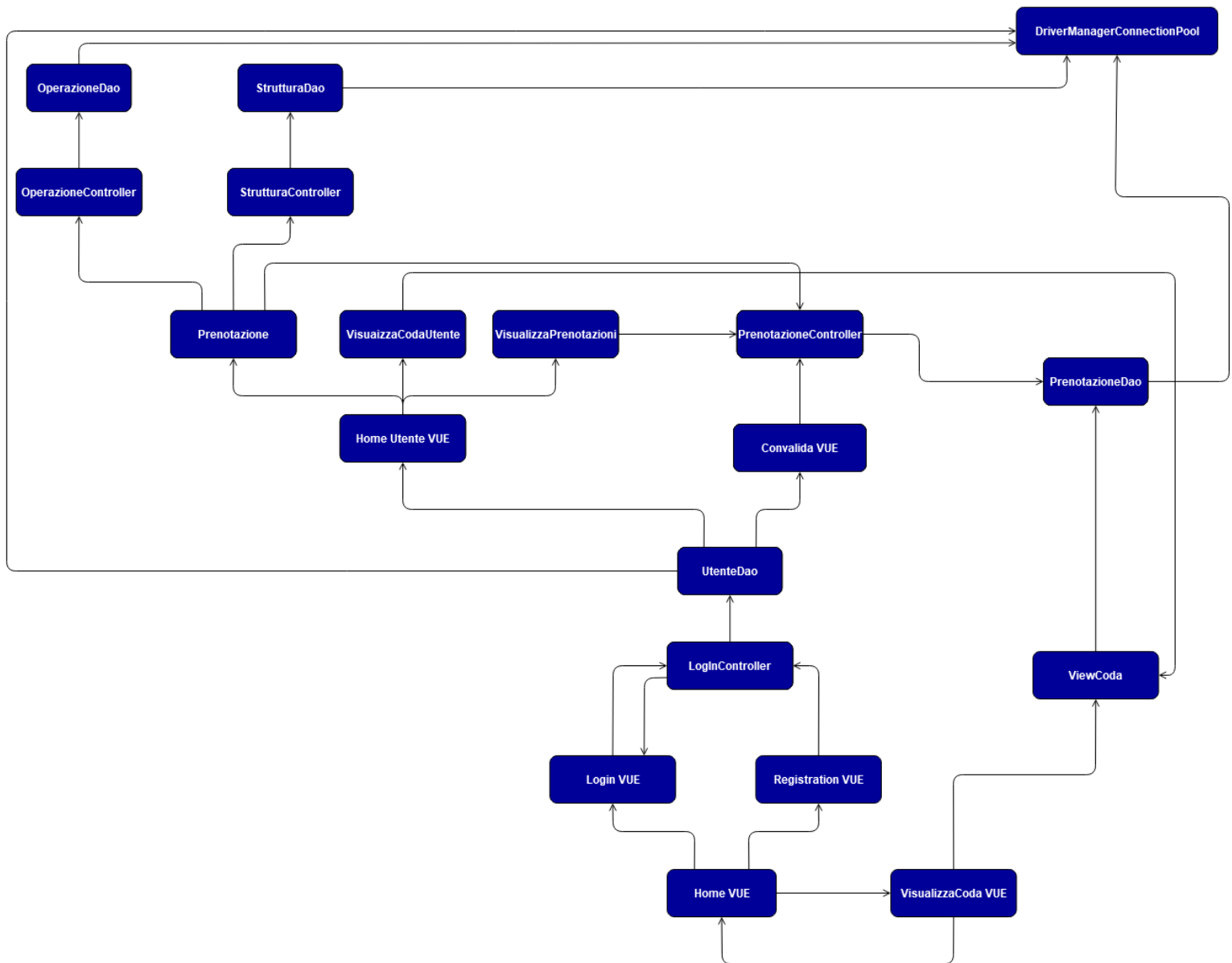
Per quanto riguarda l'app desktop il invece si divide in:

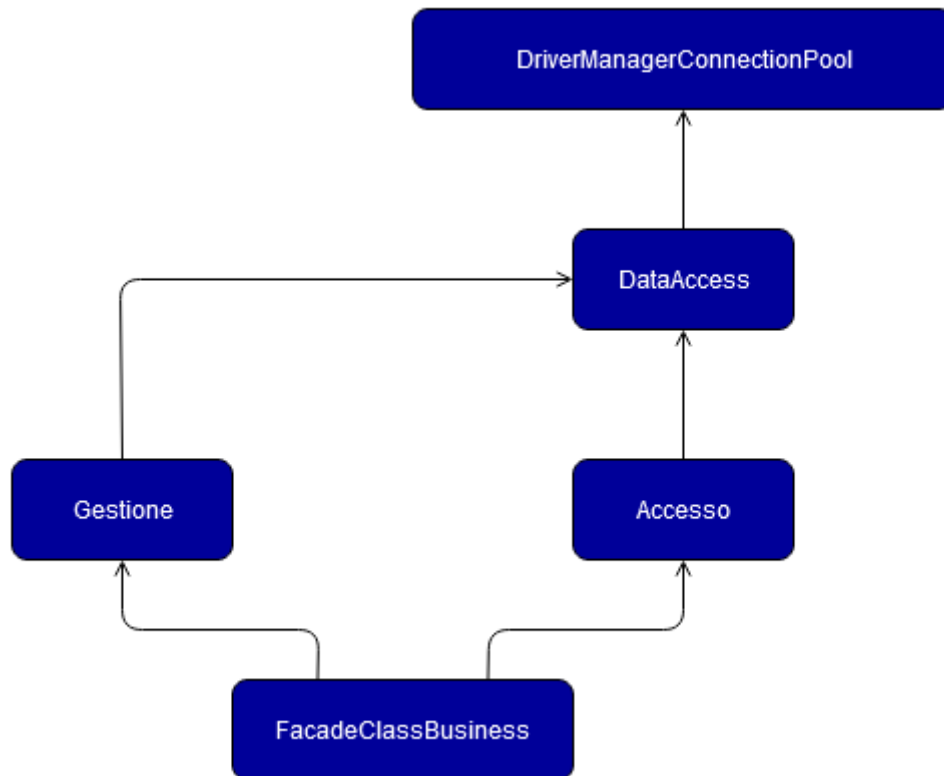
Persistence:

- DataAccess
- DriverManagerConnectionPool

Business:

- Accesso
- FacadeClassBusiness
- Gestione





3. Pass/fail criteri

Se durante il testing osserviamo che l'output che riceviamo è diverso da quello atteso allora siamo riusciti ad ottenere il risultato sperato.

Parleremo di SUCCESSO se verranno individuate delle failure, successivamente saranno analizzate e risolte con delle correzioni (se legati a dei fault). Si itera la fase di testing per verificare che le modifiche apportate abbiano avuto impatto su altri componenti del sistema.

Con FALLIMENTO intendiamo quando il test non riesce ad individuare un errore.