

PROGETTAZIONE

CARLOAN

Versione 1.0

Data di rilascio:

INGEGNERIA DEL SOFTWARE A.A. 2014-2015

Realizzato da

Alessio Lucarella 588360 Corso di Informatica

Valentino Arcuri 592144 Corso di Informatica

INDICE

INDICE 2

1. ARCHITETTURA 3

1.1 LIVELLI ARCHITETTURALI 3

1.2 DIAGRAMMA DELLE COMPONENTI 3

1.3 DIAGRAMMA DI CONFIGURAZIONE 3

2. PROGETTO DI DETTAGLIO 4

2.1 DIAGRAMMA DELLE CLASSI 4

2.2 SPECIFICHE DELLE CLASSI 4

2.3 DIAGRAMMA DEGLI STATI 4

2.4 DIAGRAMMI DI SEQUENZA 4

3. PROGETTO DEI DATI 5

3.1 DATABASE 5

3.1.1 *Diagramma delle Dipendenze dei Dati* 5

3.1.2 *Modello del Database* 5

3.1.3 *Dettaglio dei Dati* 5

3.2 FILE SYSTEM 5

3.2.1 *Grammatiche file XML* 5

3.2.2 *Altri file* 5

4. APPENDICE 6

4.1 PATTERN UTILIZZATI 6

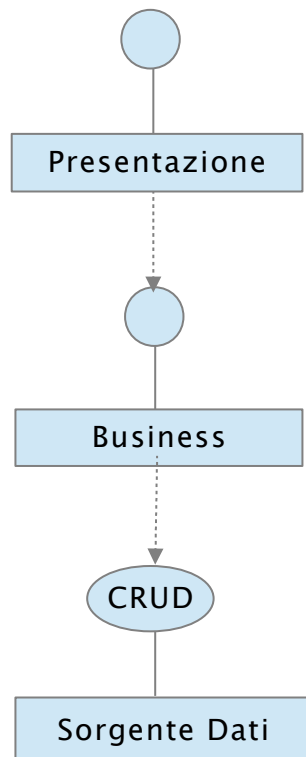
4.2 ALTRO 6



1. ARCHITETTURA

1.1 Livelli Architetturali

Si è scelto di realizzare un'architettura a tre livelli:



Livello di Presentazione:

Le componenti di questo livello si occupano della logica di presentazione e dell'intercettazione delle richieste da parte delle diverse view, invocando i relativi servizi di business, della gestione della navigazione tra le view del sistema ed infine della gestione della sessione corrente del sistema e dei dati relativi ad essa.

Livello di Business:



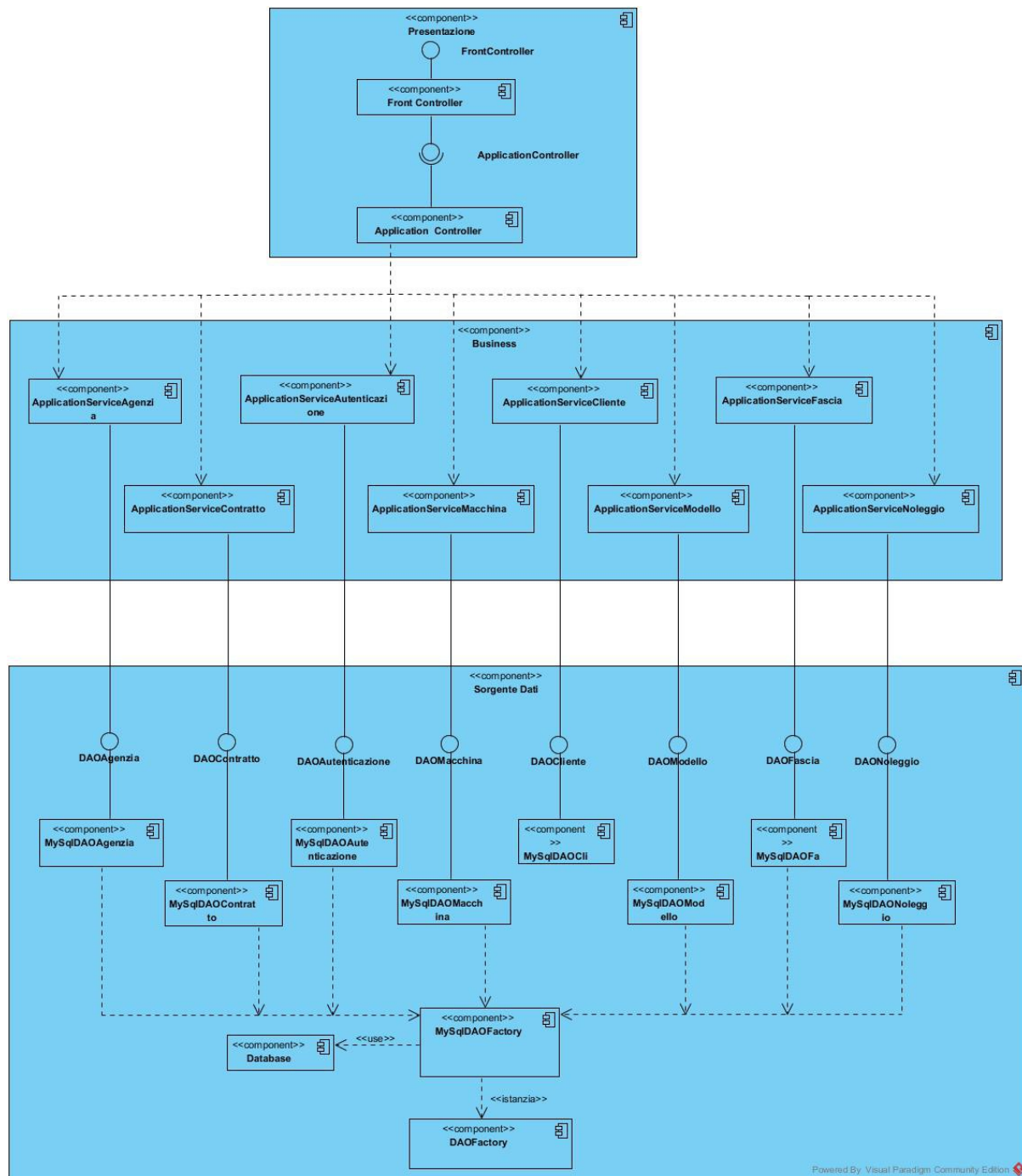
Le componenti di questo livello contengono la logica di business per l'esecuzione dei servizi del dominio e la gestione dell'accesso ai servizi del livello sottostante.

Livello Sorgente Dati:

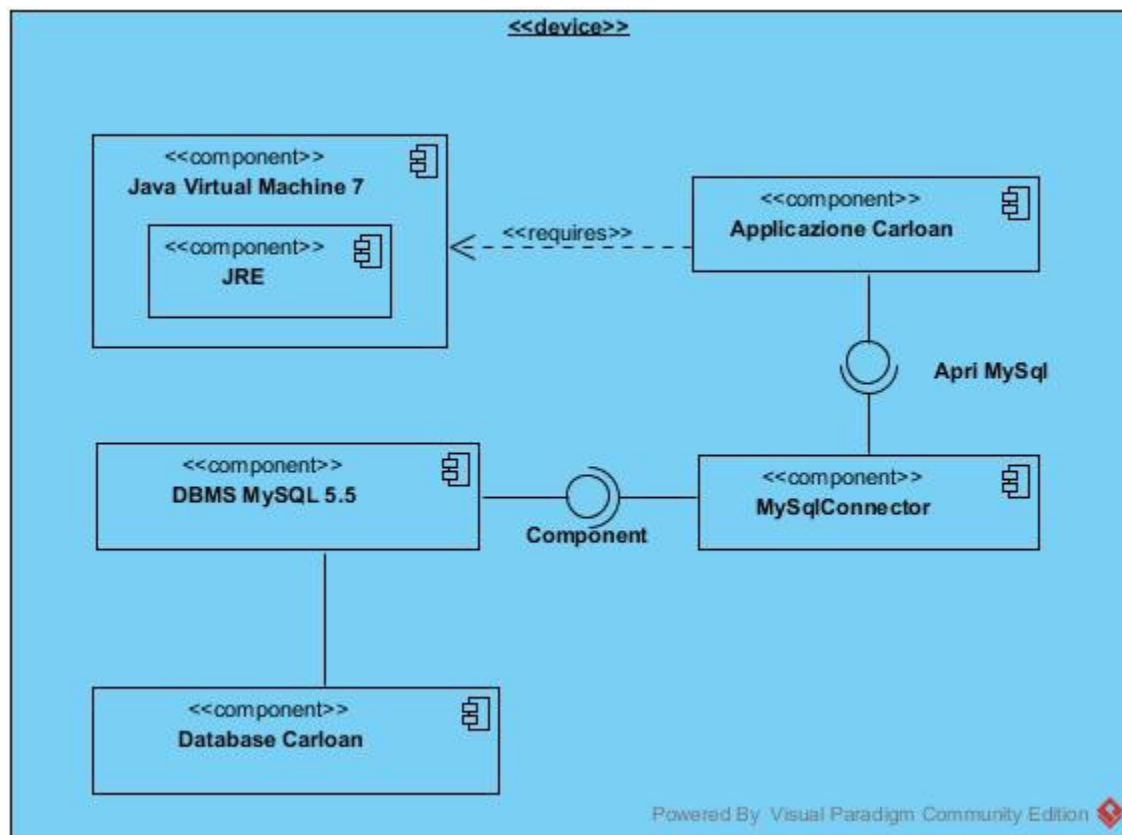
Le componenti di questo livello hanno accesso ai dati persistenti ed offrono le operazioni di creazione, lettura, aggiornamento e cancellazione (CRUD) su di essi.

1.2 Diagramma delle Componenti





1.3 Diagramma di Configurazione



2. PROGETTO DI DETTAGLIO

2.1 Diagramma delle Classi

Per ragioni di leggibilità del documento, si è deciso di riportare il Diagramma delle Classi completo del sistema, nel file pdf che si trova nella cartella documentazione.

2.2 Specifiche delle Classi

- **Main:** classe che conterrà il metodo per l'avvio dell'applicazione.
- **Login:** finestra che consente di inserire username e password per le operazioni di riconoscimento da parte del sistema.
- **Amministrazione:** finestra che presenta una serie di funzionalità per la gestione dei dati riguardanti l'agenzia, le macchine, le tariffe e delle fasce.
- **InserisciAgenzia:** finestra che consente la registrazione dell'agenzia.
- **InserimentoMacchina:** finestra che consente l'inserimento nel sistema di una nuova macchina.
- **InserimentoTariffa:** finestra che consente l'inserimento nel sistema di una nuova tariffa.
- **InserimentoNoleggio:** finestra che consente l'inserimento nel sistema di un nuovo tipo di noleggio.

2.3 Diagramma degli Stati



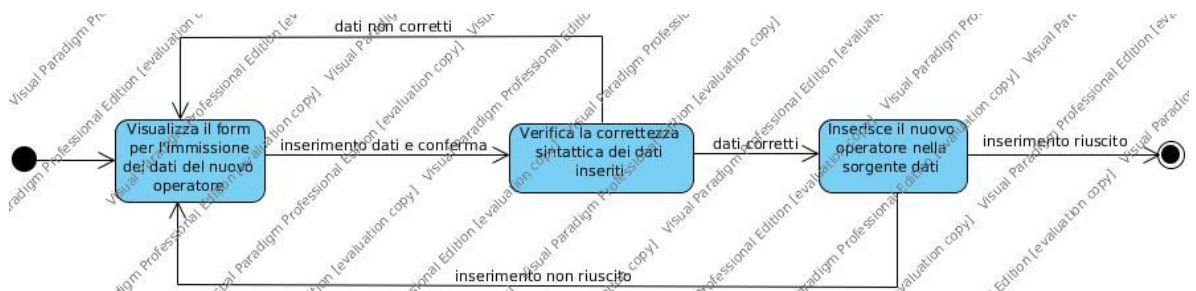
ELIMINA AGENZIA



INSERIMENTO MACCHINA



INSERIMENTO AGENZIA(OPERATORE)



2.4 Diagrammi di Sequenza

Per ragioni di leggibilità del documento, si è deciso di riportare il Diagramma delle Classi completo del sistema, nel file pdf che si trova nella cartella documentazione.



3. PROGETTO DEI DATI

3.1 Database

3.1.1 Diagramma delle Dipendenze dei Dati

3.1.2 Modello del Database

3.1.3 Dettaglio dei Dati

3.2 File System

3.2.1 Grammatiche file XML

3.2.2 Altri file



4. APPENDICE

4.1 Pattern utilizzati

Front Controller

Si è utilizzato il Front Controller per centralizzare la gestione delle richieste dei servizi da parte delle view e fornire un unico punto di accesso a tali servizi. L'utilizzo del Front Controller ha garantito dunque un aumento della manutenibilità del livello di Business, mantenendolo nettamente separato dal livello di presentazione ed ha permesso di evitare codice duplicato.

Application Controller

Si è utilizzato l'Application Controller per centralizzare l'azione e la gestione del dispatching delle View. L'application Controller contiene la logica decisionale per l'esecuzione dei servizi attraverso i vari Application Service. Funge da terminale per il Front Controller, che delega la decisione per l'esecuzione del servizio all'Application Controller, attraverso il quale instanziano gli Application Service appropriati. L'utilizzo dell'Application Controller permette un notevole incremento della manutenibilità del sistema: è possibile infatti apportare facilmente modifiche o espansioni ai servizi offerti, avendo la possibilità di sviluppare il sistema attraverso incrementi successivi e potendo testare separatamente le parti interessate. L'application Controller utilizza il view dispatcher per il dispatching delle view.

Application Service

Si sono utilizzati una serie Application Service che incapsulano al loro interno la logica inerente uno specifico oggetto di business, permettendo



la separazione degli interessi tra i vari oggetti del livello e la centralizzazione dei servizi offerti da uno specifico oggetto di business.

Data Access Object

Il Data Access Object permette la separazione tra il livello di Business e il livello di Sorgente Dati. Sono stati realizzati diversi DAO specifici per ogni oggetto di Business: in questo modo, si è garantita un'interfaccia di accesso ai dati unica per qualsiasi implementazione reale della sorgente dei dati.

4.2 Altro

