PROGETTAZIONE

CARLOAN

Versione 1.0

Data di rilascio:

17/11/2015

INGEGNERIA DEL SOFTWARE A.A. 2014-2015

Realizzato da

De Liddo Morena 622819 Informatica ITPS
m.deliddo@studenti.uniba.it

Marzulli Simone 617357 Informatica ITPS
s.marzulli3@studenti.uniba.it

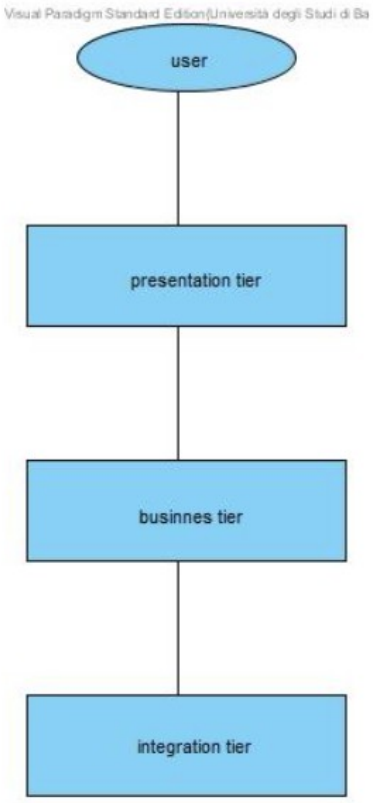
INDICE

1. ARCHITETTURA

1.1 Livelli Architetturali

Per la realizzazione del caso di studio sono stati utilizzati i tre livelli architetturali in figura.

L'utente visualizza e interagisce solamente con l'interfaccia grafica del sistema. Il livello dell'interfaccia grafica non utilizza direttamente i servizi del business tier, ma si avvale di un gestore di servizi che instrada le sue richieste ai livelli inferiori.



Presentation Tier: Le componenti di questo livello si occupano della logica di presentazione e dell'intercettazione delle richieste degli utenti, invocando i relativi servizi di business per gestirle.

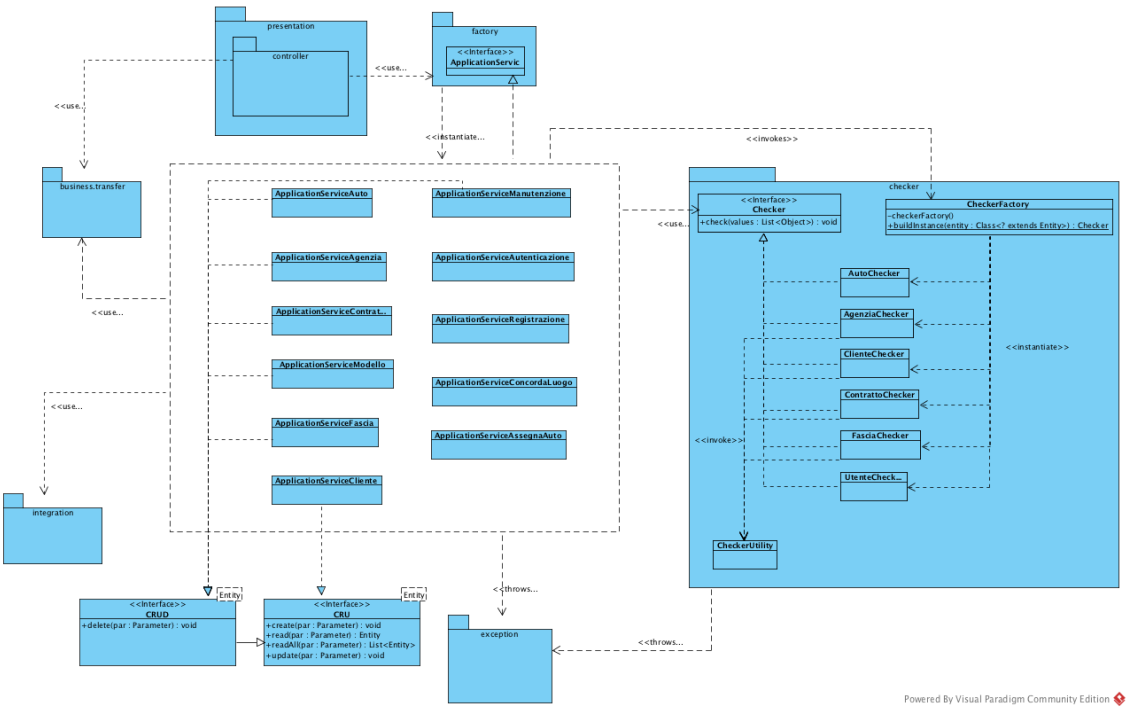
Business tier: Le componenti di questo livello contengono la logica di business per l'esecuzione dei servizi del dominio e la gestione dell'accesso dei servizi del livello sottostante.

Integration tier: le componenti di questo livello hanno accesso al database ed offrono le operazioni di creazione, lettura, aggiornamento e cancellazione su di esso.

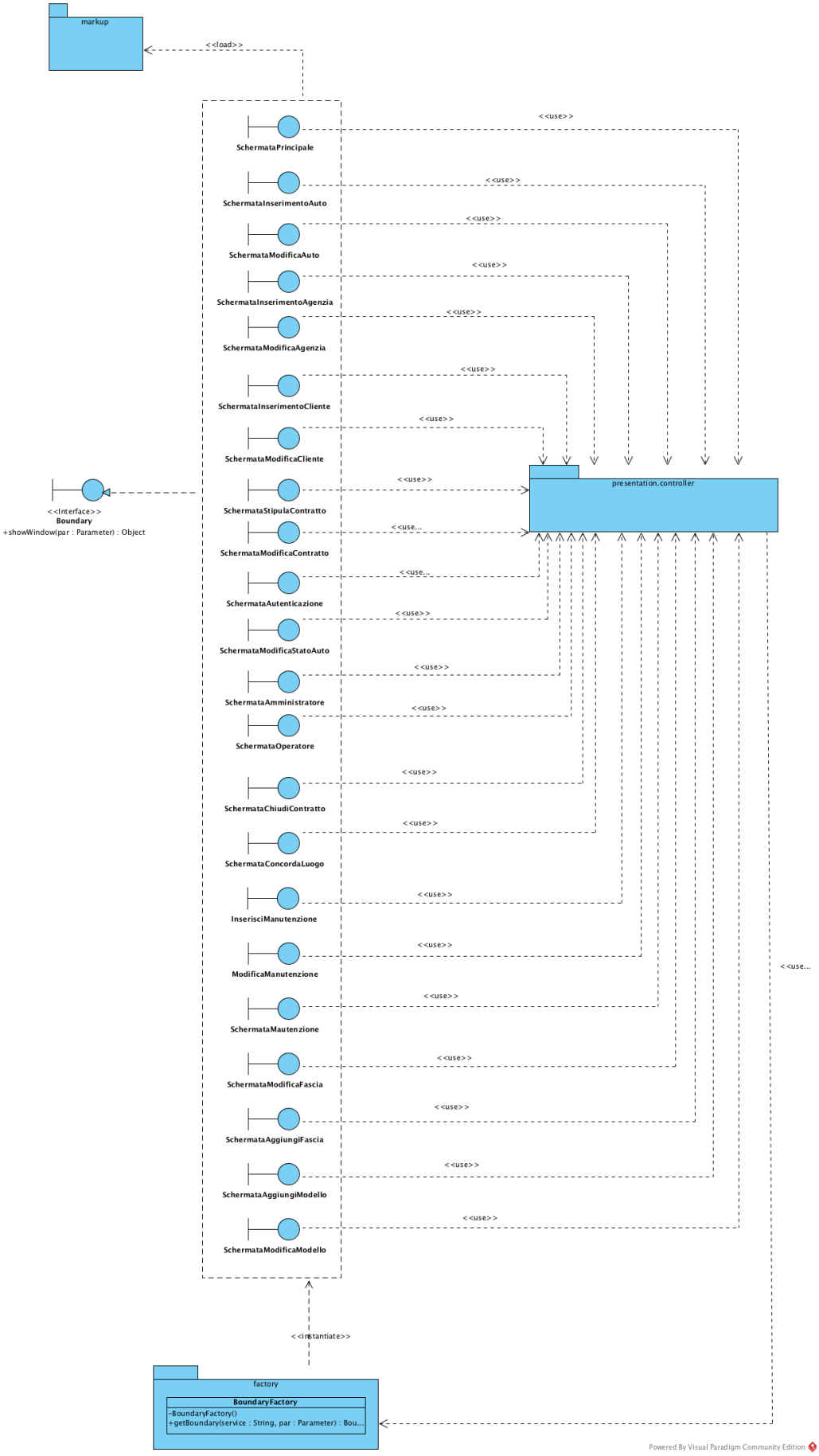
2. PROGETTO DI DETTAGLIO

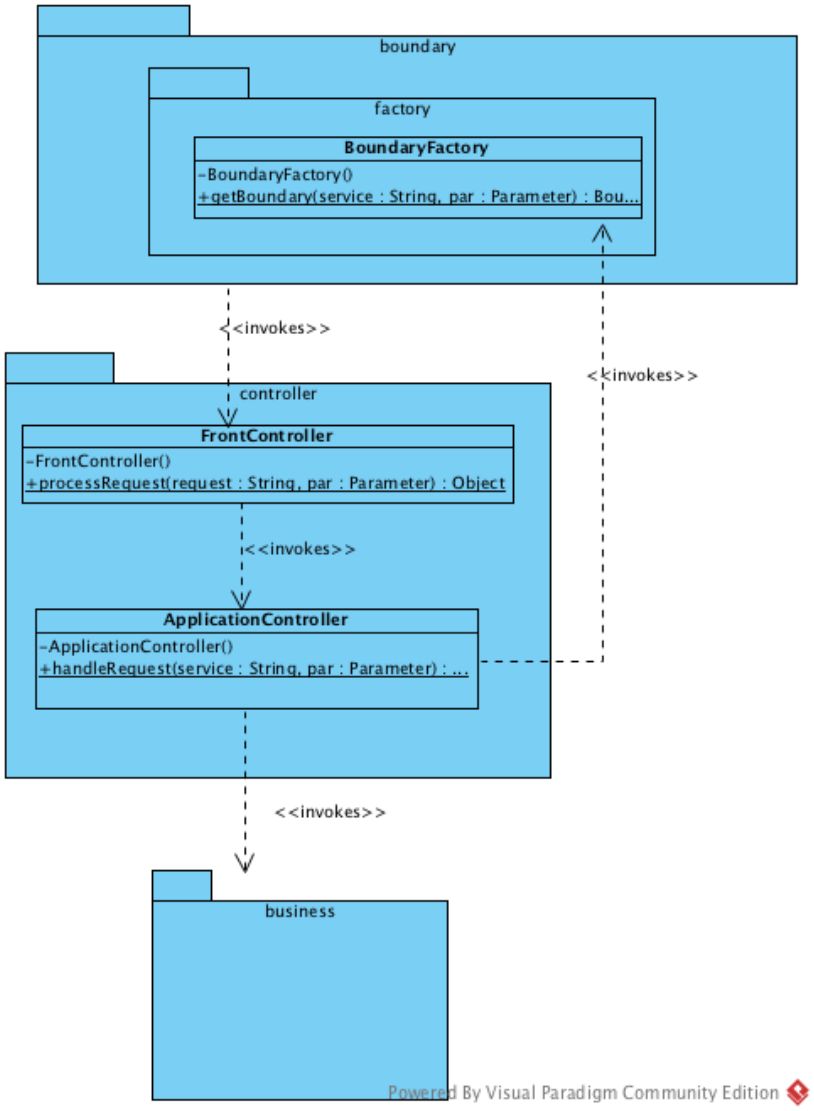
2.1 Diagramma delle Classi

PROJ Application Service

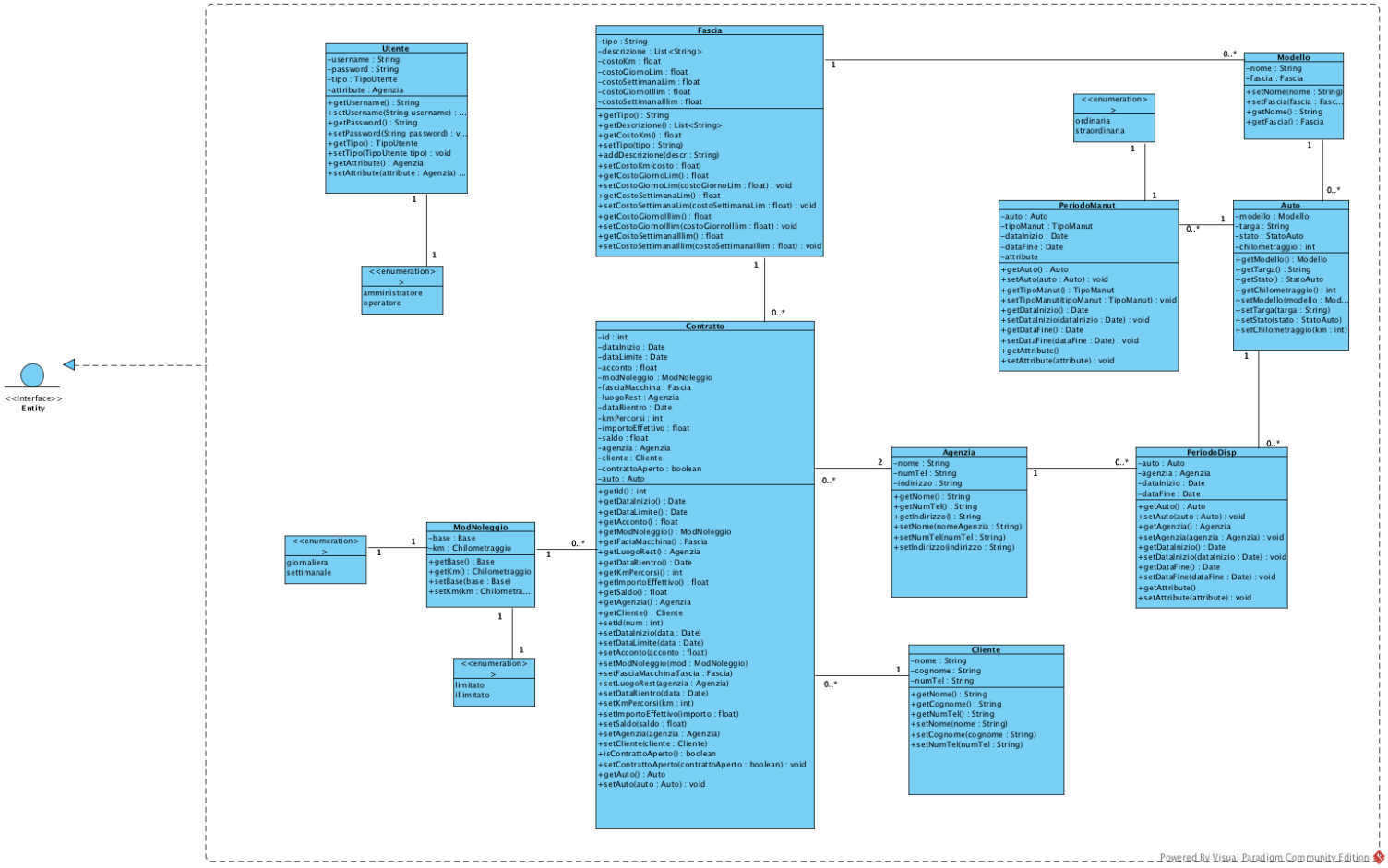


PROJ Boundary

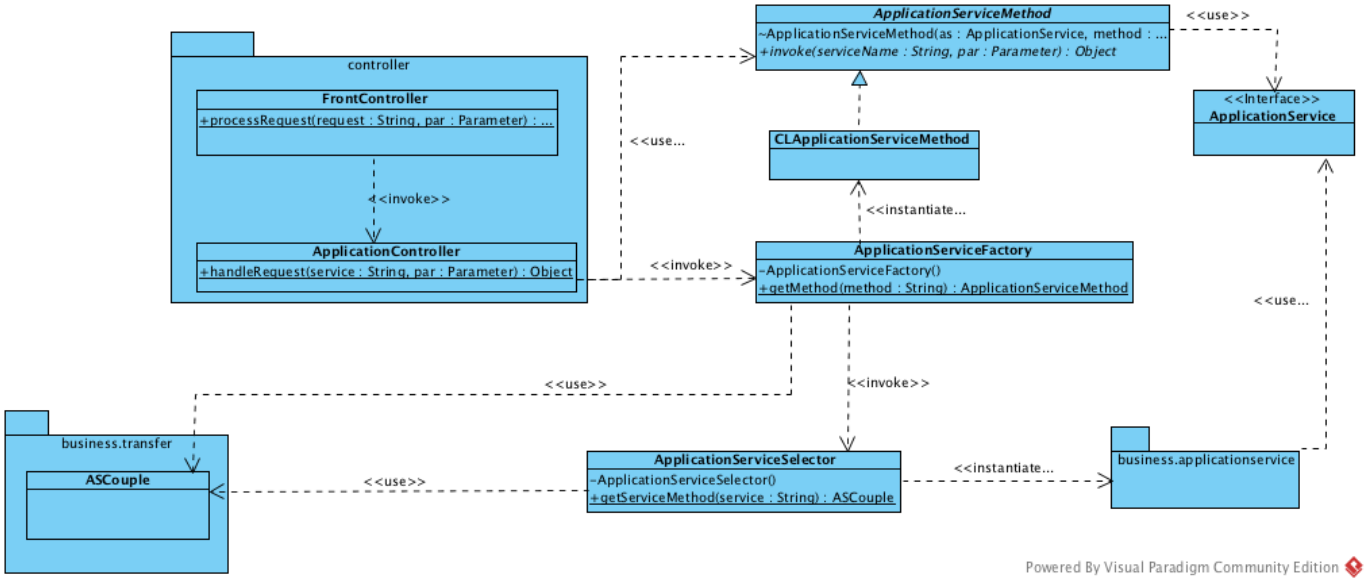



boundary
factory
BoundaryFactory
-BoundaryFactory()
+getBoundary(service : String, par : Parameter) : Bou...
<<invokes>>
<<invokes>>
controller
FrontController
-FrontController()
+processRequest(request : String, par : Parameter) : Object
<<invokes>>
ApplicationController
-ApplicationController()
+handleRequest(service : String, par : Parameter) : ...
<<invokes>>
business
Powered By Visual Paradigm Community Edition

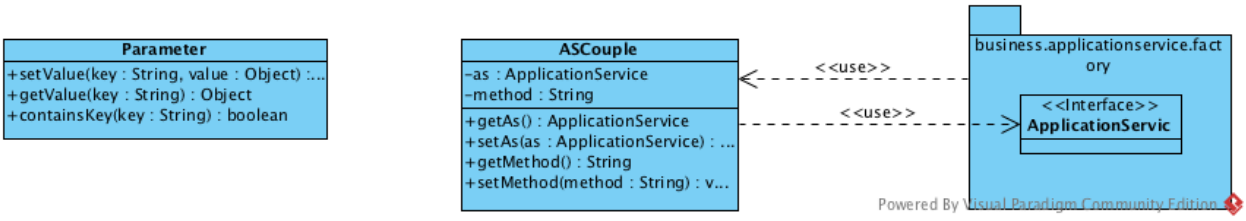
PROJ Entity

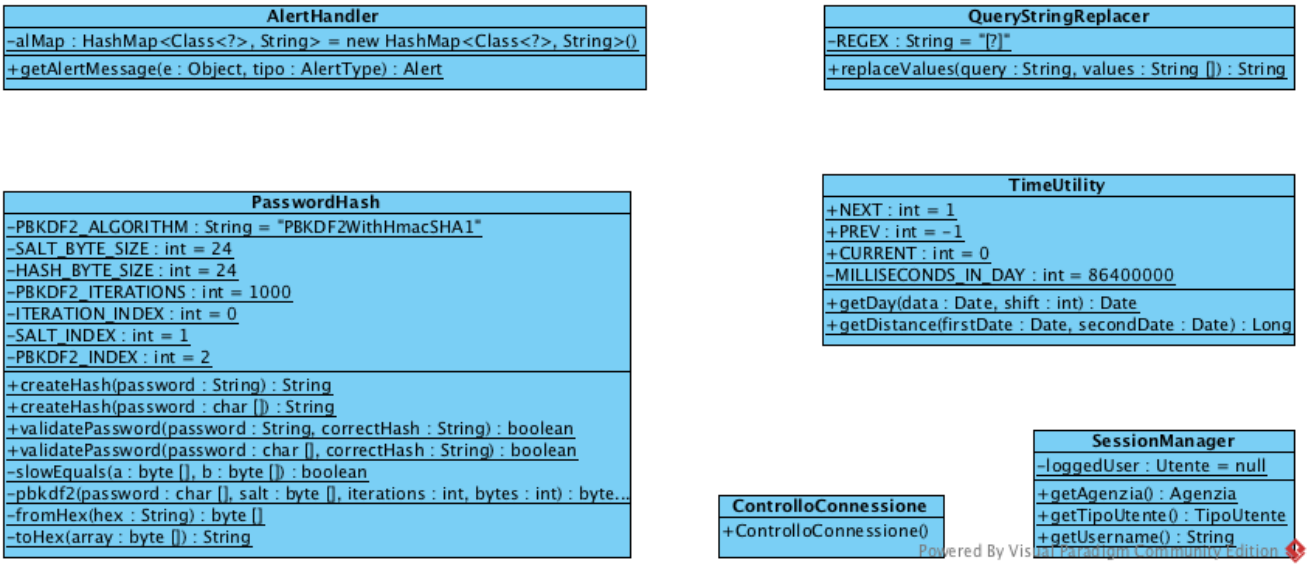

Utente
-username : String
-password : String
-tipo : TipoUtente
-attribute : Agenzia
+getUsername() : String
+setUsername(String username) : ...
+getPassword() : String
+setPassword(String password) : v...
+getTipo() : TipoUtente
+setTipo(TipoUtente tipo) : void
+getAttribute() : Agenzia
+setAttribute(attribute : Agenzia) ...
<<enumeration>>
amministratore
operatore
Fascia
-tipo : String
-descrizione : List<String>
-costoKm : float
-costoGiornoLim : float
-costoSettimanaLim : float
-costoGiornoIllim : float
-costoSettimanaIllim : float
+getTipo() : String
+getDescrizione() : List<String>
+getCostoKm() : float
+setTipo(tipo : String)
+addDescrizione(descr : String)
+setCostoKm(costo : float)
+getCostoGiornoLim() : float
+setCostoGiornoLim(costoGiornoLim : float) : void
+getCostoSettimanaLim() : float
+setCostoSettimanaLim(costoSettimanaLim : float) : void
+getCostoGiornoIllim() : float
+setCostoGiornoIllim(costoGiornoIllim : float) : void
+getCostoSettimanaIllim() : float
+setCostoSettimanaIllim(costoSettimanaIllim : float) : void
Modello
-nome : String
-fascia : Fascia
+setNome(nome : String)
+setFascia(fascia : Fasc...
+getNome() : String
+getFascia() : Fascia
<<enumeration>>
ordinaria
straordinaria
PeriodoManut
-auto : Auto
-tipoManut : TipoManut
-dataInizio : Date
-dataFine : Date
-attribute
+getAuto() : Auto
+setAuto(auto : Auto) : void
+getTipoManut() : TipoManut
+setTipoManut(tipoManut : TipoManut) : void
+getDataInizio() : Date
+setDataInizio(dataInizio : Date) : void
+getDataFine() : Date
+setDataFine(dataFine : Date) : void
+getAttribute()
+setAttribute(attribute) : void
Auto
-modello : Modello
-targa : String
-stato : StatoAuto
-chilometraggio : int
+getModello() : Modello
+getTarga() : String
+getStato() : StatoAuto
+getChilometraggio() : int
+setModello(modello : Mod...
+setTarga(targa : String)
+setStato(stato : StatoAuto)
+setChilometraggio(km : int)
Contratto
-id : int
-dataInizio : Date
-dataLimite : Date
-acconto : float
-modNoleggio : ModNoleggio
-fasciaMacchina : Fascia
-luogoRest : Agenzia
-dataRientro : Date
-kmPercorsi : int
-importoEffettivo : float
-saldo : float
-agenzia : Agenzia
-cliente : Cliente
-contrattoAperto : boolean
-auto : Auto
+getId() : int
+getDataInizio() : Date
+getDataLimite() : Date
+getAcconto() : float
+getModNoleggio() : ModNoleggio
+getFasciaMacchina() : Fascia
+getLuogoRest() : Agenzia
+getDataRientro() : Date
+getKmPercorsi() : int
+getImportoEffettivo() : float
+getSaldo() : float
+getAgenzia() : Agenzia
+getCliente() : Cliente
+setId(num : int)
+setDataInizio(data : Date)
+setDataLimite(data : Date)
+setAcconto(acconto : float)
+setModNoleggio(mod : ModNoleggio)
+setFasciaMacchina(fascia : Fascia)
+setLuogoRest(agenzia : Agenzia)
+setDataRientro(data : Date)
+setKmPercorsi(km : int)
+setImportoEffettivo(importo : float)
+setSaldo(saldo : float)
+setAgenzia(agenzia : Agenzia)
+setCliente(cliente : Cliente)
+isContrattoAperto() : boolean
+setContrattoAperto(contrattoAperto : boolean) : void
+getAuto() : Auto
+setAuto(auto : Auto) : void
Agenzia
-nome : String
-numTel : String
-indirizzo : String
+getNome() : String
+getNumTel() : String
+getIndirizzo() : String
+setNome(nomeAgenzia : String)
+setNumTel(numTel : String)
+setIndirizzo(indirizzo : String)
PeriodoDisp
-auto : Auto
-agenzia : Agenzia
-dataInizio : Date
-dataFine : Date
+getAuto() : Auto
+setAuto(auto : Auto) : void
+getAgenzia() : Agenzia
+setAgenzia(agenzia : Agenzia) : void
+getDataInizio() : Date
+setDataInizio(dataInizio : Date) : void
+getDataFine() : Date
+setDataFine(dataFine : Date) : void
+getAttribute()
+setAttribute(attribute) : void
<<enumeration>>
giornaliera
settimanale
ModNoleggio
-base : Base
-km : Chilometraggio
+getBase() : Base
+getKm() : Chilometraggio
+setBase(base : Base)
+setKm(km : Chilometra...
<<enumeration>>
limitato
illimitato
Cliente
-nome : String
-cognome : String
-numTel : String
+getNome() : String
+getCognome() : String
+getNumTel() : String
+setNome(nome : String)
+setCognome(cognome : String)
+setNumTel(numTel : String)
<<Interface>>
Entity
Powered By Visual Paradigm Community Edition

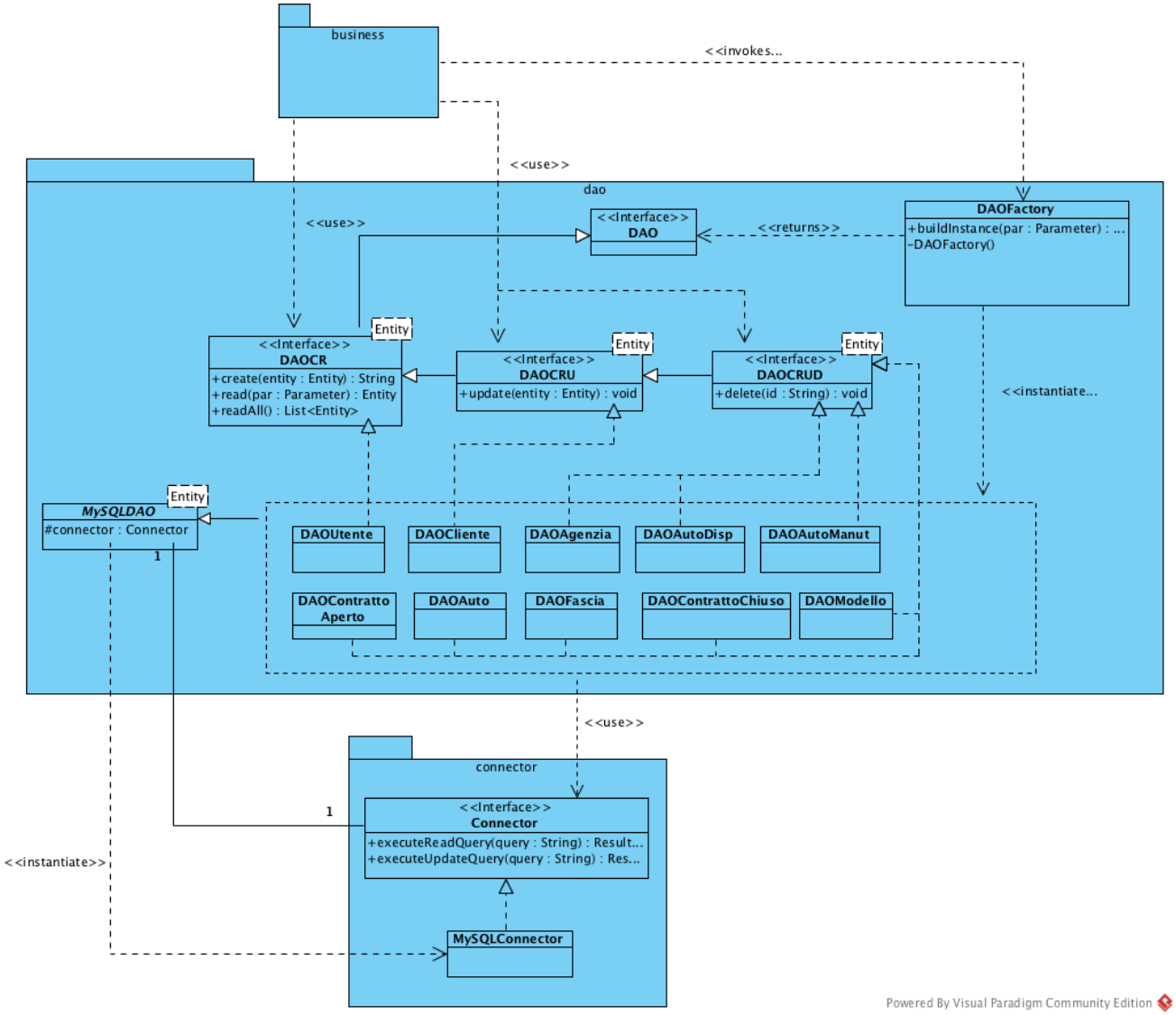
PROJ Factory



PROJ Transfer

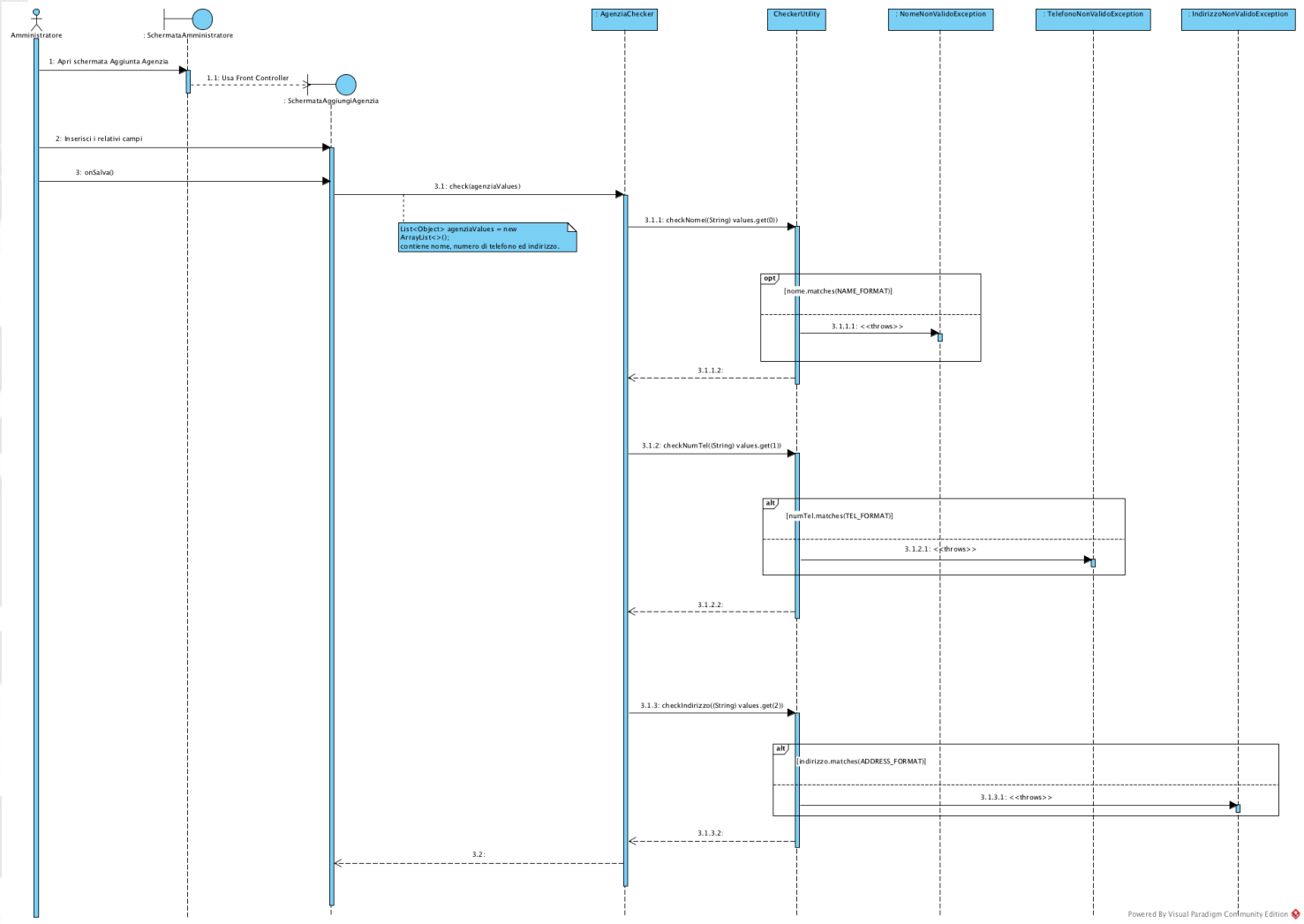



AlertHandler
-alMap : HashMap<Class<?>, String> = new HashMap<Class<?>, String>()
+getAlertMessage(e : Object, tipo : AlertType) : Alert
QueryStringReplacer
-REGEX : String = "[?]"
+replaceValues(query : String, values : String []) : String
PasswordHash
-PBKDF2_ALGORITHM : String = "PBKDF2WithHmacSHA1"
-SALT_BYTE_SIZE : int = 24
-HASH_BYTE_SIZE : int = 24
-PBKDF2_ITERATIONS : int = 1000
-ITERATION_INDEX : int = 0
-SALT_INDEX : int = 1
-PBKDF2_INDEX : int = 2
+createHash(password : String) : String
+createHash(password : char []) : String
+validatePassword(password : String, correctHash : String) : boolean
+validatePassword(password : char [], correctHash : String) : boolean
-slowEquals(a : byte [], b : byte []) : boolean
-pbkdf2(password : char [], salt : byte [], iterations : int, bytes : int) : byte...
-fromHex(hex : String) : byte []
-toHex(array : byte []) : String
TimeUtility
+NEXT : int = 1
+PREV : int = -1
+CURRENT : int = 0
-MILLISECONDS_IN_DAY : int = 86400000
+getDay(data : Date, shift : int) : Date
+getDistance(firstDate : Date, secondDate : Date) : Long
SessionManager
-loggedUser : Utente = null
+getAgenzia() : Agenzia
+getTipoUtente() : TipoUtente
+getUsername() : String
ControlloConnessione
+ControlloConnessione()
Powered By Visual Paradigm Community Edition

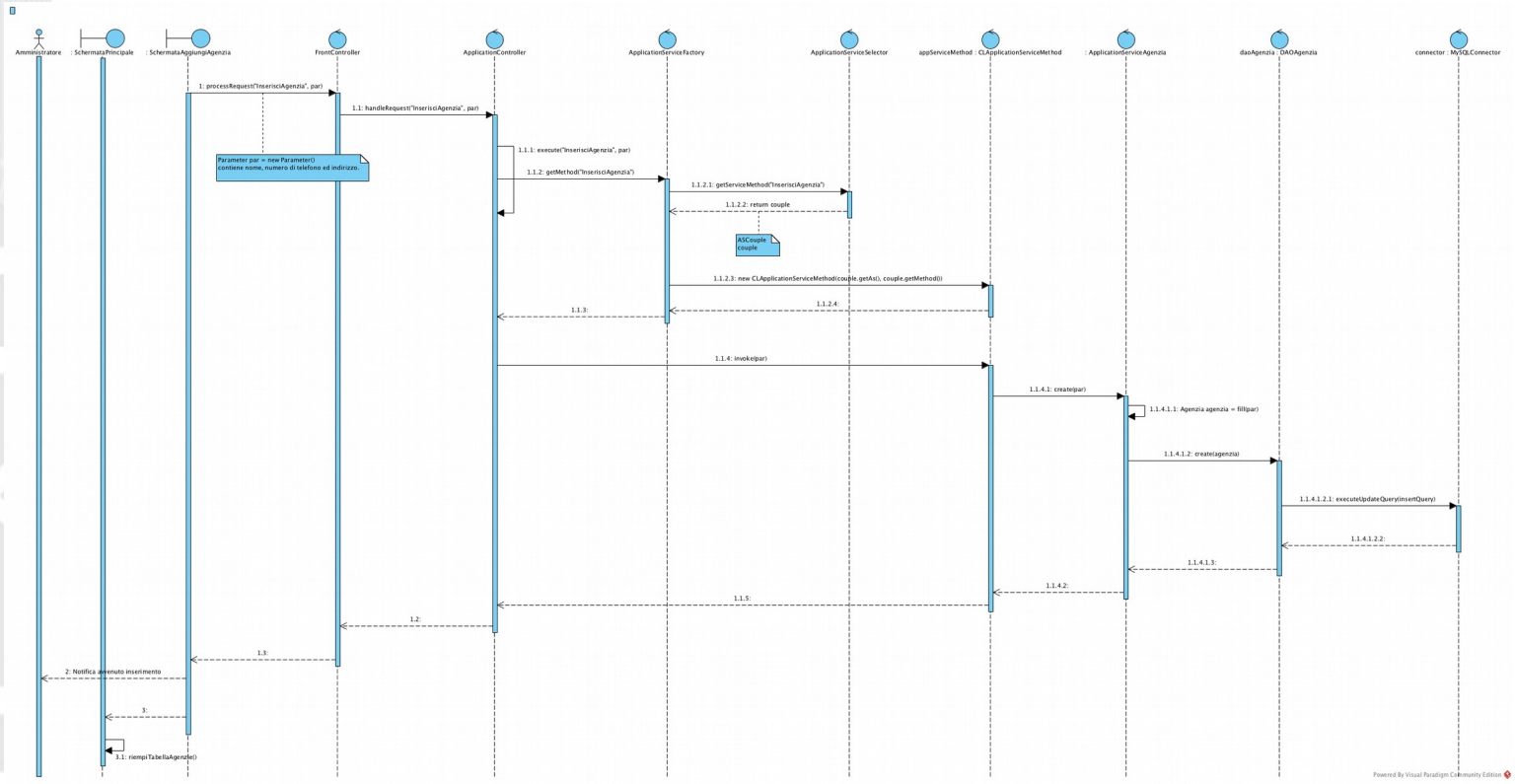

business
<<invokes...
<<use>>
dao
<<use>>
<<Interface>>
DAO
<<returns>>
DAOFactory
+buildInstance(par : Parameter) : ...
-DAOFactory()
Entity
<<Interface>>
DAOCR
+create(entity : Entity) : String
+read(par : Parameter) : Entity
+readAll() : List<Entity>
Entity
<<Interface>>
DAOCRU
+update(entity : Entity) : void
Entity
<<Interface>>
DAOCRUD
+delete(id : String) : void
<<instantiate...
Entity
MySQLDAO
#connector : Connector
1
DAOUtente
DAOCliente
DAOAgenzia
DAOAutoDisp
DAOAutoManut
DAOContratto Aperto
DAOAuto
DAOFascia
DAOContrattoChiuso
DAOModello
<<use>>
connector
1
<<Interface>>
Connector
+executeReadQuery(query : String) : Result...
+executeUpdateQuery(query : String) : Res...
<<instantiate>>
MySQLConnector
Powered By Visual Paradigm Community Edition

2.2 Diagrammi di Sequenza

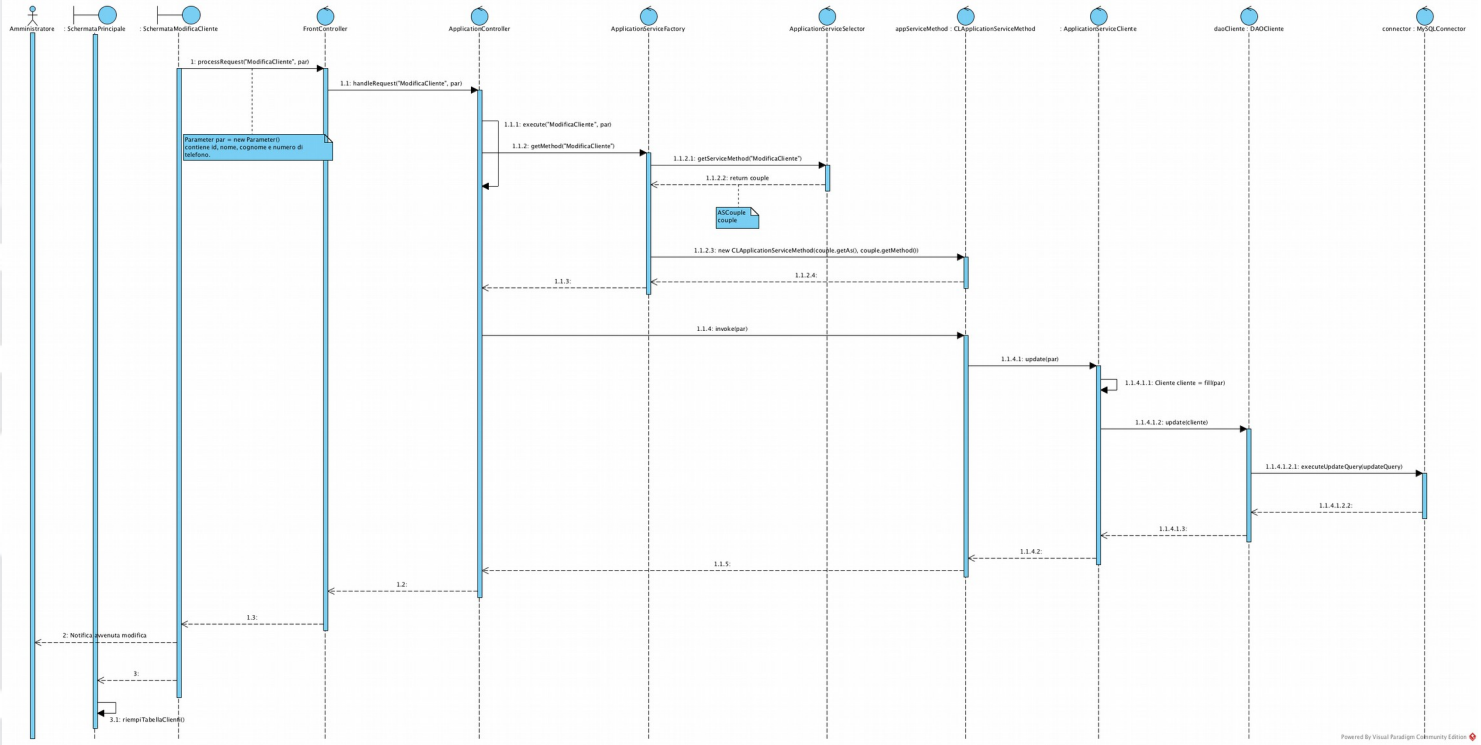
Inserisci Agenzia 1-2



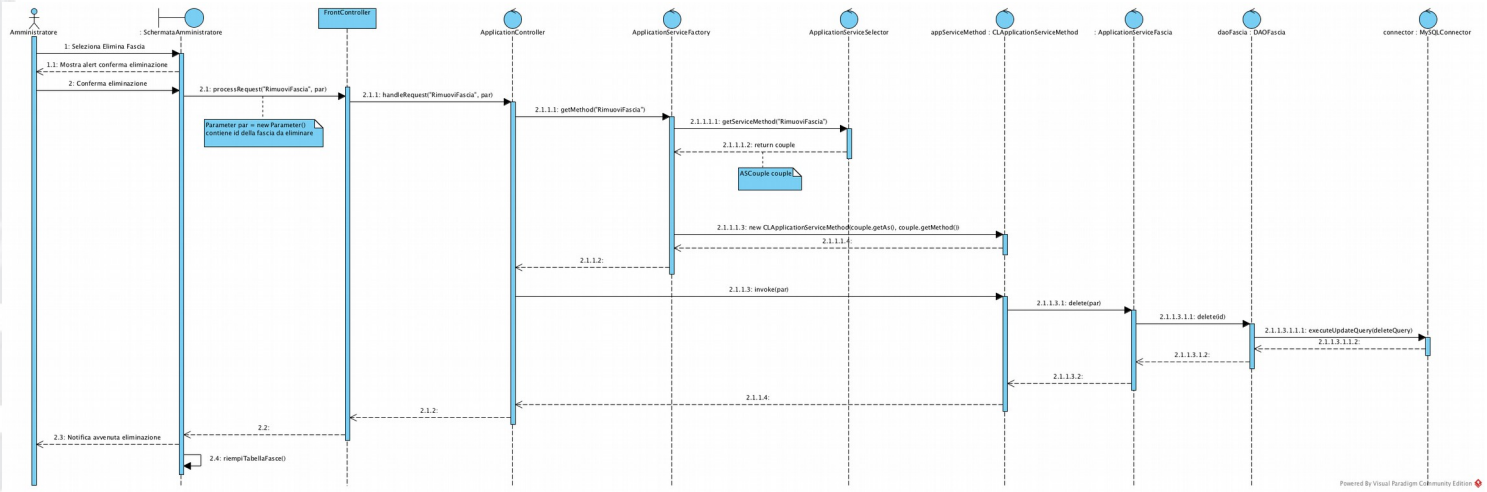
Inserisci Agenzia 2-2



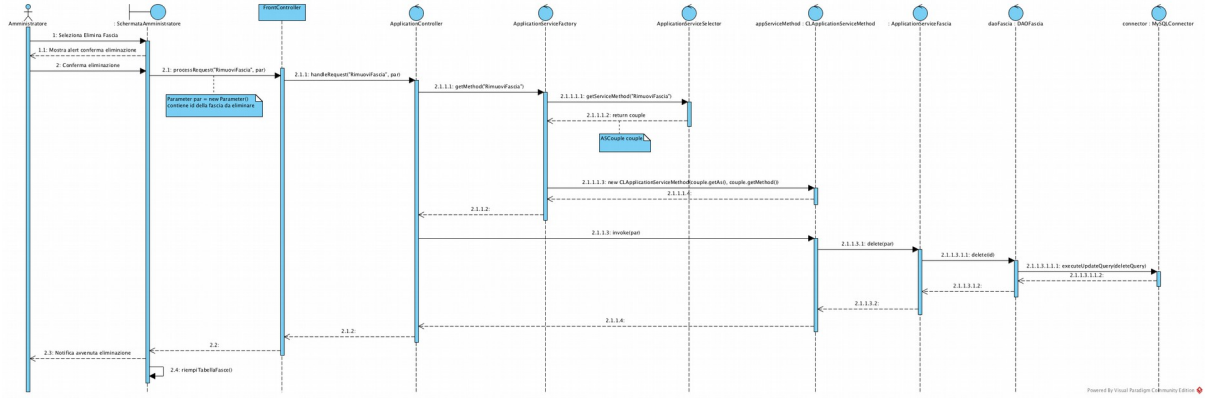
Modifica Cliente 1-2



Modifica cliente 2-2



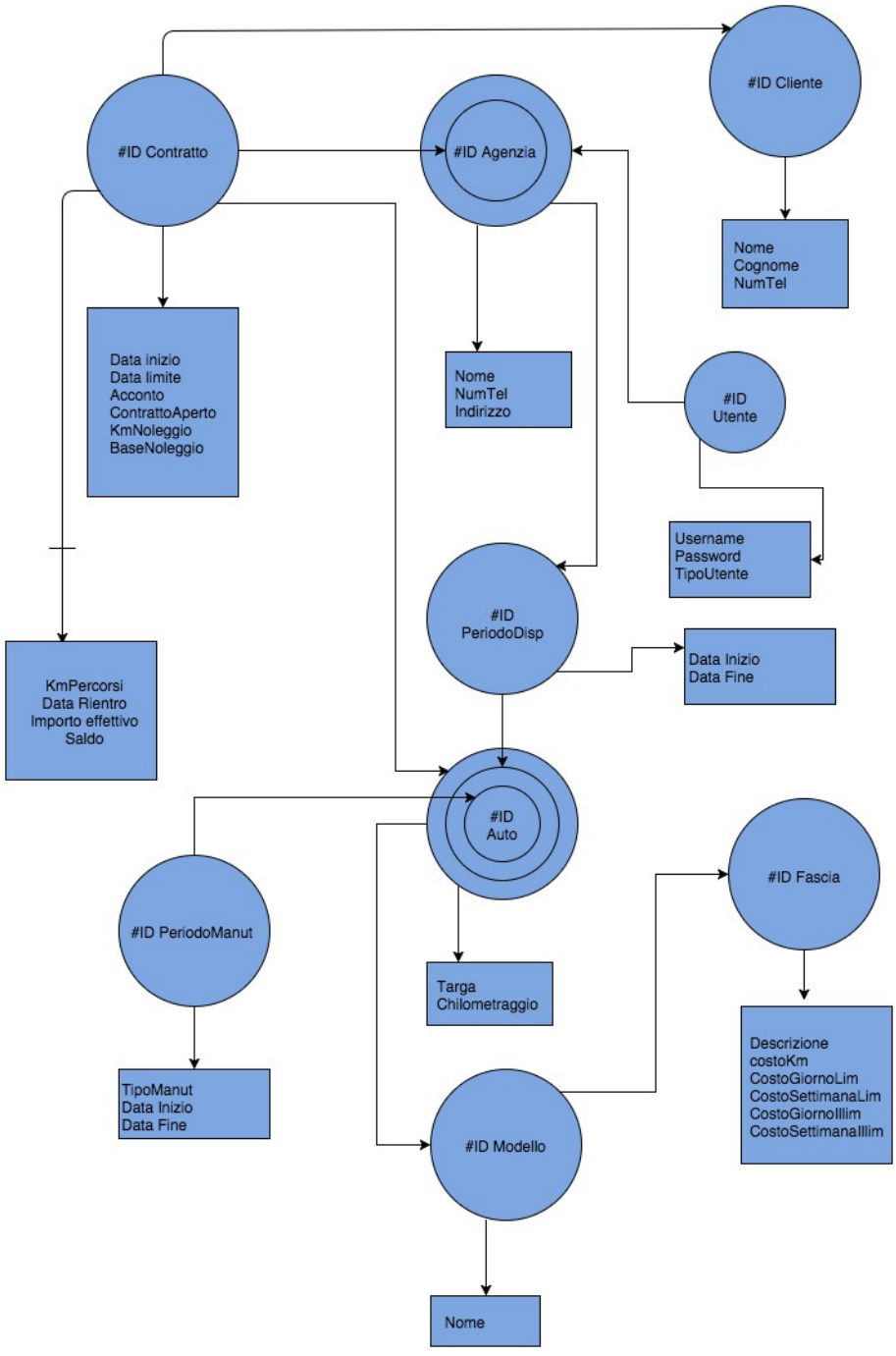
Elimina Fascia 1-1



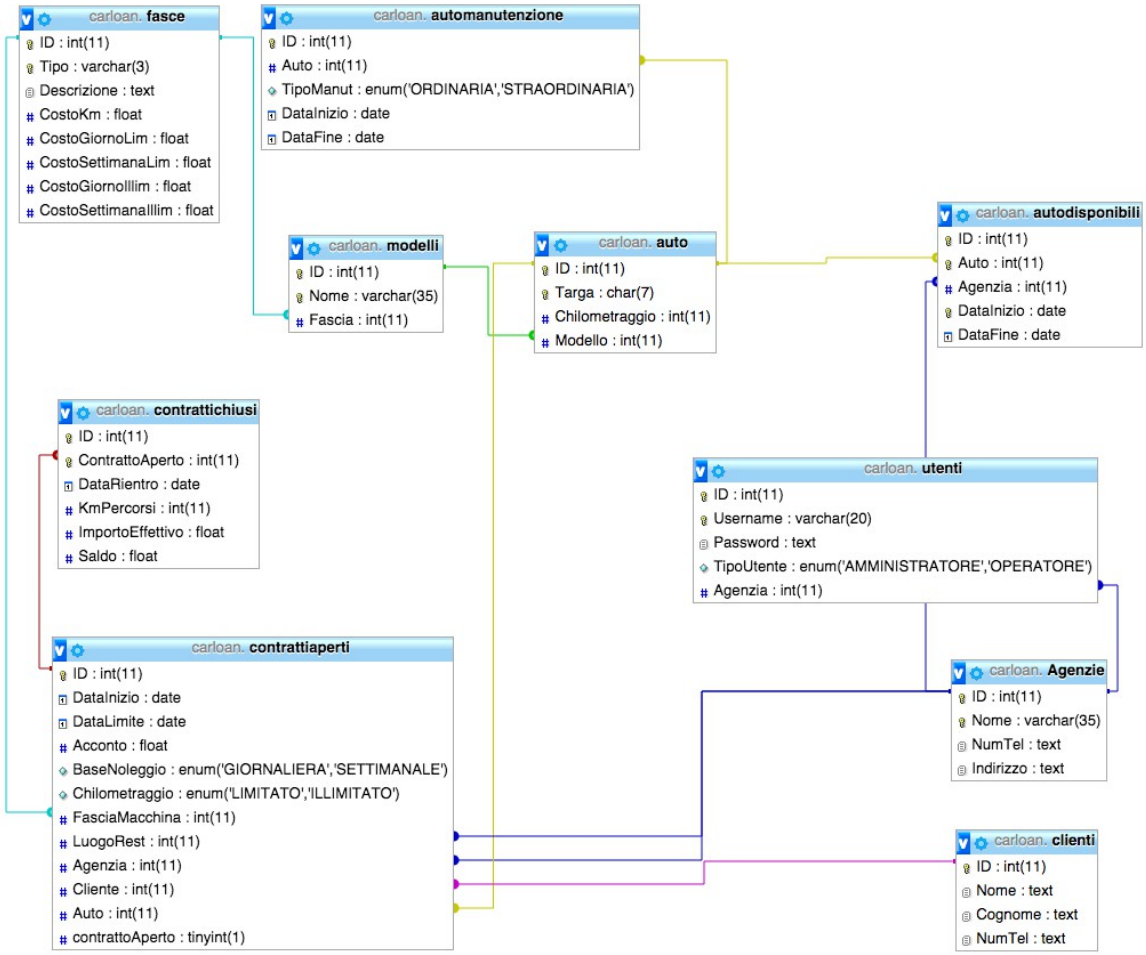
3. PROGETTO DEI DATI

3.1 Database

3.1.1 Diagramma delle Dipendenze dei Dati



3.1.2 Modello del Database



3.1.3 Dettaglio dei Dati

Agenzie

Colonna	Tipo	Null	Predefinito
ID	int(11)	No	
Nome	varchar(35)	No	
NumTel	text	No	
Indirizzo	text	No	

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	4	A	No
Nome	BTREE	Sì	No	Nome	4	A	No

auto

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
Targa	char(7)	No		
Chilometraggio	int(11)	No		
Modello	int(11)	No		modelli -> ID

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	8	A	No
Targa	BTREE	Sì	No	Targa	8	A	No

autodisponibili

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
Auto	int(11)	No		auto -> ID

Agenzia	int(11)	No		Agenzie -> ID
DataInizio	date	No		
DataFine	date	Sì	*NULL*	

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Null	Commenti
PRIMARY	BTREE	Sì	No	ID	10	No	
Auto	BTREE	Sì	No	Auto	10	No	
				DataInizio	10	No	

automanutenzione

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
Auto	int(11)	No		auto -> ID
TipoManut	enum('ORDINARIA', 'STRAORDINARIA')	No		
DataInizio	date	No		
DataFine	date	No		

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	0	A	No

clienti

Colonna	Tipo	Null	Predefinito
ID	int(11)	No	
Nome	text	No	
Cognome	text	No	

NumTel	text	No	

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	4	A	No

contrattiaperti

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
DataInizio	date	No		
DataLimite	date	No		
Acconto	float	No		
BaseNoleggio	enum('GIORNALIERA', 'SETTIMANALE')	No		
Chilometraggio	enum('LIMITATO', 'ILLIMITATO')	No		
FasciaMacchina	int(11)	No		fasce -> ID
LuogoRest	int(11)	Sì	*NULL*	Agenzie -> ID
Agenzia	int(11)	No		Agenzie -> ID
Cliente	int(11)	No		clienti -> ID
Auto	int(11)	Sì	*NULL*	auto -> ID
contrattoAperto	tinyint(1)	Sì	*NULL*	

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	5	A	No

contrattichiusi

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
ContrattoAperto	int(11)	No		contrattiaperti -> ID
DataRientro	date	No		
KmPercorsi	int(11)	No		
ImportoEffettivo	float	No		
Saldo	float	No		

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Null
PRIMARY	BTREE	Sì	No	ID	2	No
ContrattoAperto	BTREE	Sì	No	ContrattoAperto	2	No

fasce

Colonna	Tipo	Null	Predefinito
ID	int(11)	No	
Tipo	varchar(3)	No	
Descrizione	text	No	
CostoKm	float	No	
CostoGiornoLim	float	No	
CostoSettimanaLim	float	No	
CostoGiornoIllim	float	No	
CostoSettimanaIllim	float	No	

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	4	A	No
Tipo	BTREE	Sì	No	Tipo	4	A	No

modelli

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
Nome	varchar(35)	No		
Fascia	int(11)	No		fasce -> ID

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	8	A	No
Nome	BTREE	Sì	No	Nome	8	A	No

utenti

Colonna	Tipo	Null	Predefinito	Collegamenti a
ID	int(11)	No		
Username	varchar(20)	No		
Password	text	No		
TipoUtente	enum('AMMINISTRATORE', 'OPERATORE')	No		
Agenzia	int(11)	No		Agenzie -> ID

Indici

Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null
PRIMARY	BTREE	Sì	No	ID	2	A	No
Username	BTREE	Sì	No	Username	2	A	No

4. APPENDICE

4.1 Pattern utilizzati

4.1.1 Pattern architetturali

Front Controller: Consente la centralizzazione delle richieste da parte delle interfacce in modo che la gestione della logica di business sia demandata ai livelli sottostanti.

Application Controller: Ad esso giungono le richieste smistate dal front controller. Ha il compito di gestirle.

Business Object: Racchiude i dati di un certo tipo di entità del dominio applicativo.

Application Service: Racchiude le operazioni per le entità del sistema e la logica di business per gestirle.

Transfer Object: Permette di incapsulare dati che vanno scambiati tra i vari tier del sistema. Nel caso di studio è stato implementato come un HashMap<String, Object>, un dizionario in cui inserire di volta in volta i dati necessari all'operazione.

Data Access Object: Consente di incapsulare l'accesso alla sorgente dei dati per effettuare su essa le diverse operazioni.

4.1.2 Pattern di design

Command: Incapsula il singolo metodo di uno specifico application service individuato dalla corrispondente factory e le modalità per invocarlo.

4.2 Altro

Sicurezza

Per la crittografia dei dati all'interno del database, è stata utilizzata una libreria che implementa l'algoritmo di hashing **PBKDF2** (Password-Based Key Derivation Function 2). Inoltre per un ulteriore accorgimento di sicurezza è stata utilizzata la tecnica del *salting* per le password. In questa maniera se un *attacker* dovesse avere possesso del database gli hash delle password risulterebbero più difficoltosi da violare.