



# MaaP: MongoDB as an admin Platform

---

## Specifica Tecnica

<b>Versione</b>	3.2.0
<b>Data creazione</b>	xxxx-xx-xx2014-01-24
<b>Data ultima modifica</b>	xxxx-xx-xx
<b>Stato del Documento</b>	Formale
<b>Uso del Documento</b>	Esterno
<b>Redazione</b>	Pinato Giacomo,nome2,...
<b>Verifica</b>	nome1,nome2,...
<b>Approvazione</b>	nome1,nome2,...
<b>Distribuzione</b>	Aperture Software

### Sommario

Questo documento si propone di presentare la Specifica tecnica e architetturale per la Realizzazione del prodotto **MaaP**.

---

Diario delle modifiche

Versione	Data	Autore	Modifiche effettuate
1.2.0	2014-02-xx	... (RE)	Approvazione documento
1.1.0	2014-02-xx	... (VE)	Verifica documento
1.0.0	2014-03-05	Giacomo Pinato (PR)	Componenti e Classi
1.0.0	2014-01-25	Giacomo Pinato (PR)	Tecnologie Utilizzate
1.0.0	2014-01-25	Fabio Miotto (PR)	Tecnologie Utilizzate
1.0.0	2014-01-24	Giacomo Pinato (PR)	Prima stesura del documento

Tabella 1: Registro delle modifiche

## Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Scopo del documento . . . . .	7
1.2	Scopo del prodotto . . . . .	7
1.3	Glossario . . . . .	7
1.4	Riferimenti . . . . .	7
1.4.1	Normativi . . . . .	7
1.4.2	Informativi . . . . .	7
<b>2</b>	<b>Tecnologie utilizzate</b>	<b>8</b>
2.1	MongoDB . . . . .	8
2.2	Javascript . . . . .	8
2.3	NodeJs . . . . .	8
2.4	JSON . . . . .	8
2.5	AngularJs . . . . .	9
2.6	HTML5 . . . . .	9
<b>3</b>	<b>Descrizione architettura</b>	<b>10</b>
3.1	Metodo e formalismo di specifica . . . . .	10
3.2	Architettura generale . . . . .	10
3.2.1	MaaPCLI . . . . .	11
3.2.1.1	Informazioni sul package . . . . .	11
3.2.1.2	Descrizione . . . . .	11
3.2.1.3	Classi . . . . .	11
3.2.1.3.0.1	CLI . . . . .	11
3.2.1.3.0.2	Installer . . . . .	12
3.2.1.3.0.3	InstanceManager . . . . .	12
3.2.1.3.0.4	ProjectFacade . . . . .	12
3.2.1.3.0.5	ProjectCreate . . . . .	13
3.2.1.3.0.6	ProjectClone . . . . .	13
3.2.1.3.0.7	ProjectRemove . . . . .	13
3.2.2	Package . . . . .	14
3.2.3	Classi . . . . .	15
3.2.3.1	ModelServer . . . . .	16
3.2.3.2	Controller . . . . .	17
3.2.3.3	Client . . . . .	18
<b>4</b>	<b>Componenti e Classi</b>	<b>19</b>
4.1	MaaP . . . . .	19
4.1.1	Informazioni sul package . . . . .	19
4.1.1.1	Descrizione . . . . .	19
4.1.1.2	Sotto-componenti . . . . .	19
4.2	MaaP::ModelServer . . . . .	20
4.2.1	Informazioni sul package . . . . .	20
4.2.1.1	Descrizione . . . . .	20
4.2.1.2	Sottocomponenti . . . . .	20
4.2.2	MaaP::ModelServer::DataManager . . . . .	21
4.2.2.1	Informazioni sul package . . . . .	21
4.2.2.2	Descrizione . . . . .	21
4.2.2.3	Sotto-componenti . . . . .	21
4.2.2.4	Classi . . . . .	21

4.2.2.4.1	JSonComposer . . . . .	21
4.2.2.4.2	IDatabaseManager . . . . .	22
4.2.2.4.3	IDataRetriever . . . . .	22
4.2.2.5	MaaP::ModelServer::DataManager::DatabaseAnalysisManager . . . . .	23
4.2.2.5.1	Informazioni sul package . . . . .	23
4.2.2.5.2	Descrizione . . . . .	23
4.2.2.5.3	Classi . . . . .	23
4.2.2.5.3.1	DatabaseAnalysisManager . . . . .	23
4.2.2.5.3.2	DatabaseRetrieverAnalysis . . . . .	24
4.2.2.6	MaaP::ModelServer::DataManager::DatabaseUserManager . . . . .	25
4.2.2.6.1	Informazioni sul package . . . . .	25
4.2.2.6.2	Descrizione . . . . .	25
4.2.2.6.3	Classi . . . . .	25
4.2.2.6.3.1	DatabaseUserManager . . . . .	25
4.2.2.6.3.2	DataRetrieverUsers . . . . .	26
4.2.2.7	MaaP::ModelServer::DataManager::IndexManager . . . . .	26
4.2.2.7.1	Informazioni sul package . . . . .	26
4.2.2.7.2	Descrizione . . . . .	26
4.2.2.7.3	Classi . . . . .	27
4.2.2.7.3.1	IndexManager . . . . .	27
4.2.3	MaaP::ModelServer::Database . . . . .	28
4.2.3.1	Informazioni sul package . . . . .	28
4.2.3.2	Descrizione . . . . .	28
4.2.3.3	Classi . . . . .	28
4.2.3.3.1	MongooseDBAnalysis . . . . .	28
4.2.3.3.2	DBAnalysis . . . . .	29
4.2.3.3.3	Mongoose . . . . .	29
4.2.3.3.4	MongooseDBFramework . . . . .	29
4.2.3.3.5	DBFramework . . . . .	30
4.2.3.3.6	User . . . . .	30
4.2.3.3.7	Query . . . . .	31
4.2.4	MaaP::ModelServer::DSL . . . . .	32
4.2.4.1	Informazioni sul package . . . . .	32
4.2.4.2	Descrizione . . . . .	32
4.2.4.3	Sotto-componenti . . . . .	33
4.2.4.4	Classi . . . . .	33
4.2.4.4.1	ParserInterface . . . . .	33
4.2.4.4.2	DSLParser . . . . .	33
4.2.4.4.3	DSLManager . . . . .	33
4.2.4.4.4	CollectionData . . . . .	34
4.3	MaaP::Controller . . . . .	34
4.3.1	Informazioni sul package . . . . .	34
4.3.1.1	Descrizione . . . . .	34
4.3.1.2	Classi . . . . .	35
4.3.1.2.1	IPassport . . . . .	35
4.3.1.2.2	PassportAdapter . . . . .	35
4.3.1.2.3	Passport . . . . .	35
4.3.1.2.4	FrontController . . . . .	35
4.3.1.2.5	Dispatcher . . . . .	36
4.4	MaaP::Client . . . . .	37

4.4.1	Informazioni sul package . . . . .	37
4.4.1.1	Descrizione . . . . .	37
4.4.1.2	Sottocomponenti . . . . .	37
4.4.2	MaaP::Client::View . . . . .	38
4.4.2.1	Informazioni sul package . . . . .	38
4.4.2.2	Descrizione . . . . .	38
4.4.2.3	Sotto-componenti . . . . .	38
4.4.2.4	MaaP::Client::View::Template . . . . .	39
4.4.2.5	Informazioni sul package . . . . .	39
4.4.2.6	Descrizione . . . . .	39
4.4.2.7	Classi . . . . .	39
4.4.2.7.1	SignIn . . . . .	39
4.4.2.7.2	SignUp . . . . .	40
4.4.2.7.3	AdminMainPageCollection . . . . .	40
4.4.2.7.4	UserMainPageCollection . . . . .	40
4.4.2.7.5	MainPageDocument . . . . .	41
4.4.2.7.6	MainPageDocumentEdit . . . . .	41
4.4.2.7.7	UserProfileEdit . . . . .	41
4.4.2.7.8	UserProfile . . . . .	42
4.4.2.7.9	AdminProfile . . . . .	42
4.4.2.7.10	PasswordRecovery . . . . .	42
4.4.3	MaaP::Client::ControllerModelView . . . . .	43
4.4.3.1	Informazioni sul package . . . . .	43
4.4.3.2	Descrizione . . . . .	43
4.4.3.3	Sotto-componenti . . . . .	43
4.4.3.4	MaaP::Client::ControllerModelView::ControllerClient . . . . .	44
4.4.3.4.1	Informazioni sul package . . . . .	44
4.4.3.4.2	Descrizione . . . . .	44
4.4.3.4.3	Classi . . . . .	44
4.4.3.4.3.1	ControllerAutenticazione . . . . .	44
4.4.3.4.3.2	ControllerCollection . . . . .	45
4.4.3.4.3.3	ControllerDocument . . . . .	45
4.4.3.4.3.4	ControllerProfilo . . . . .	46
4.4.3.4.3.5	ControllerMenu . . . . .	47
4.4.3.5	MaaP::Client::ControllerModelView::Scope . . . . .	48
4.4.3.5.1	Informazioni sul package . . . . .	48
4.4.3.5.2	Descrizione . . . . .	48
4.4.3.5.3	Classi . . . . .	49
4.4.3.5.3.1	Collection . . . . .	49
4.4.3.5.3.2	Query . . . . .	49
4.4.3.5.3.3	Document . . . . .	49
4.4.3.5.3.4	Profilo . . . . .	49
4.4.3.5.3.5	Menu . . . . .	50
4.4.4	MaaP::Client::ModelClient . . . . .	51
4.4.4.1	Informazioni sul package . . . . .	51
4.4.4.2	Descrizione . . . . .	51
4.4.4.3	Sotto-componenti . . . . .	51
4.4.4.4	MaaP::Client::ModelClient::Services . . . . .	52
4.4.4.4.1	Informazioni sul package . . . . .	52
4.4.4.4.2	Descrizione . . . . .	52

4.4.4.4.3	Classi . . . . .	52
4.4.4.4.3.1	HTTP . . . . .	52
4.4.4.5	MaaP::Client::ModelClient::Model . . . . .	53
4.4.4.5.1	Informazioni sul package . . . . .	53
4.4.4.5.2	Descrizione . . . . .	53
4.4.4.5.3	Classi . . . . .	53
4.4.4.5.3.1	SessionData . . . . .	53
<b>5</b>	<b>Diagrammi di attività</b>	<b>54</b>
<b>6</b>	<b>Design Pattern Utilizzati</b>	<b>55</b>
6.1	Design Pattern architetturali . . . . .	55
6.1.1	MVVM . . . . .	55
6.1.2	Three-tier . . . . .	56
6.2	Design Pattern creazionali . . . . .	57
6.2.1	Singleton . . . . .	57
6.3	Design Pattern comportamentali . . . . .	58
6.3.1	Strategy . . . . .	58
6.4	Design Pattern strutturali . . . . .	59
6.4.1	Adapter . . . . .	59
6.4.2	Facade . . . . .	60
<b>7</b>	<b>Stime di fattibilità e di bisogno di risorse</b>	<b>61</b>
<b>8</b>	<b>Tracciamento</b>	<b>62</b>
8.1	Tracciamento componenti - requisiti . . . . .	62
8.2	Tracciamento requisiti - componenti . . . . .	62
<b>A</b>	<b>Descrizione Design Pattern</b>	<b>63</b>
A.1	Design Pattern architetturali . . . . .	63
A.1.1	MVVM . . . . .	63
A.1.2	Three-Tier . . . . .	64
A.2	Design Pattern creazionali . . . . .	65
A.2.1	Singleton . . . . .	65
A.3	Design Pattern strutturali . . . . .	65
A.3.1	Adapter . . . . .	65
A.3.2	Facade . . . . .	66
A.4	Design Pattern comportamentali . . . . .	67
A.4.1	Strategy . . . . .	67

## 1 Introduzione

### 1.1 Scopo del documento

Il presente documento ha lo scopo di definire la progettazione ad alto livello del progetto **MaaP**, a partire dai requisiti individuati durante l'Analisi. Verrà presentata l'architettura generale secondo la quale saranno organizzate le varie componenti software, i *Design Pattern* e le tecnologie utilizzate per poi descrivere più dettagliatamente le varie componenti e relative dipendenze.

### 1.2 Scopo del prodotto

Lo scopo del prodotto è produrre un framework per generare interfacce web di amministrazione dei dati di business basati sullo stack Node.js e MongoDB.

L'obiettivo è quello di semplificare il lavoro allo sviluppatore che dovrà rispondere in modo rapido e standard alle richieste degli esperti di business.

### 1.3 Glossario

Al fine di evitare ogni ambiguità nella comprensione del linguaggio utilizzato nel presente documento e, in generale, nella documentazione fornita dal gruppo Aperture Software, ogni termine tecnico, di difficile comprensione o di necessario approfondimento verrà inserito nel documento *Glossario\_v3.2.0.pdf*.

Saranno in esso definiti e descritti tutti i termini in corsivo e allo stesso tempo marcati da una lettera "G" maiuscola in pedice nella documentazione fornita.

### 1.4 Riferimenti

#### 1.4.1 Normativi

- **Analisi dei requisiti:** *Analisi\_dei\_Requisiti\_v3.2.0.pdf*
- **Norme di Progetto:** *Norme\_di\_Progetto\_v3.2.0.pdf* (allegato alla presente documentazione)

#### 1.4.2 Informativi

- Learning Node: O'Reilly Shelley Powers
- AngularJS: O'Reilly Brad Green e Shyam Seshadri
- Software Engineering (8th edition), Ian Sommerville, Pearson Education — Addison-Wesley
- Design Patterns, E. Gamma, R. Helm, R. Johnson, J. Vlissides, Pearson Education — Addison-Wesley
- Dall'idea al codice con UML 2 L. Baresi, L. Lavazza, M. Pianciamore, Pearson Education

## 2 Tecnologie utilizzate

In questa sezione verranno elencate e descritte le tecnologie che si utilizzeranno durante lo sviluppo del progetto. In particolare la colonna portante del progetto sarà lo stack MEAN, ovvero MongoDB, Express, Angular e Node.js.

### 2.1 MongoDB

Il database con il quale la nostra applicazione dovrà interagire è realizzato con MongoDB, come specificato nel capitolato. Questa tecnologia offre i seguenti vantaggi:

- Facile indicizzazione: Ogni campo in MongoDB può diventare un indice;
- Bilanciamento di carico: MongoDB scala orizzontalmente molto facilmente grazie all'utilizzo di shard;
- Integrazione con Javascript: Query o altre funzioni scritte in Javascript possono essere eseguite direttamente dal database.

### 2.2 Javascript

Si è deciso di utilizzare Javascript in quanto è il linguaggio su cui si basano tutte le altre tecnologie che andremo ad utilizzare, e offre quindi una facile integrazione, oltre ad essere un ottimo linguaggio per applicazioni web e client side.

### 2.3 NodeJs

Si è deciso di utilizzare il linguaggio Node.js in quanto consigliato dal capitolato e adatto al progetto. Le sue caratteristiche più vantaggiose sono:

- Modello event-driven: ovvero programmazione ad eventi, che si basa su un concetto semplice: il flusso del programma non segue un corso specifico ma è guidato dalle azioni dell'utilizzatore;
- Modello asincrono: grazie a questa caratteristica è possibile ridurre al minimo i tempi di morti in quanto, nell'attesa del completamento di una operazione, si procede con altri flussi logici.
- Grande scalabilità: Grazie al modo in cui è implementato, Node.js riesce ad essere largamente scalabile con minimo sforzo.

### 2.4 JSON

Rappresenta il tipo di messaggi con cui client e server si scambiano informazioni. I vantaggi offerti sono:

- Semplicità: i messaggi JSON sono più corti rispetto ad altri formati di interscambio, e vengono eseguiti più velocemente dal parser. JSON inoltre risulta più semplice e immediato rispetto ad esempio a XML.

Svantaggi:

- Restrittività: JSON è meno restrittivo rispetto ad XML, e questo può permettere di inserire errori nello scambio di messaggi.



## 2.5 AngularJs

- Two Way Data-Binding: Una delle caratteristiche principali di angular. Le modifiche apportate al model si riflettono direttamente sugli elementi del DOM, e le modifiche al DOM si ripercuotono automaticamente sul model. Questo alleggerisce tremendamente il codice necessario a controllare ed ascoltare e gestire il DOM, automatizzando il processo. E noi sappiamo che automatico è bene.
- Templates: I template HTML sono parsati dal browser nel DOM, il quale costituisce poi l'input per il compilatore Angular. Quest'ultimo poi crea il data binding tra il DOM e lo scope dei dati. Uno dei più grandi vantaggi di questa tecnica è che separa presentazione da implementazione, in quanto i template HTML possono essere modificati senza alterare il modo in cui sono inseriti i dati.
- Dependency Injection: Angular possiede nativamente una "dependency injection", che aiuta gli sviluppatori facilitando la creazione, la comprensione e il testing dell'applicazione.
- Directives: Le directives possono essere usate per definire tag HTML personalizzati che fungono da widget. Possono inoltre essere usate per decorare elementi con comportamenti personalizzati o per manipolare attributi del DOM.

## 2.6 HTML5

TODO\*\*\*\*\*DA completare.....

## **3 Descrizione architettura**

### **3.1 Metodo e formalismo di specifica**

Si è deciso di procedere utilizzando un approccio top-down per l'esposizione dell'architettura dell'applicazione, ovvero descrivendo inizialmente le componenti in generale per poi arrivare a trattarle al particolare. Si descriveranno i package e i componenti per poi dettagliare le singole classi, specificando per ciascuna di esse il tipo, l'obiettivo e la funzionalità. Poi si passerà ad illustrare degli esempi d'uso di Design Pattern (descritti approfonditamente nell'Appendice A) e le tecnologie utilizzate.

### **3.2 Architettura generale**

L'architettura del framework segue un modello di architettura in stile three-tier che prevede la suddivisione dell'applicazione in tre diversi strati dedicati rispettivamente all'interfaccia utente (Client), alla business logic (Controller) e alla gestione dei dati persistenti (Model). La parte Client segue il design pattern MVVM utilizzato da AngularJS ed è quindi suddivisa in Model, View, ViewModel.

I seguenti diagrammi rappresentano l'architettura ad alto livello del framework, indicando i package e le relazioni che intercorrono tra questi.

### 3.2.1 MaaPCLI

#### 3.2.1.1 Informazioni sul package

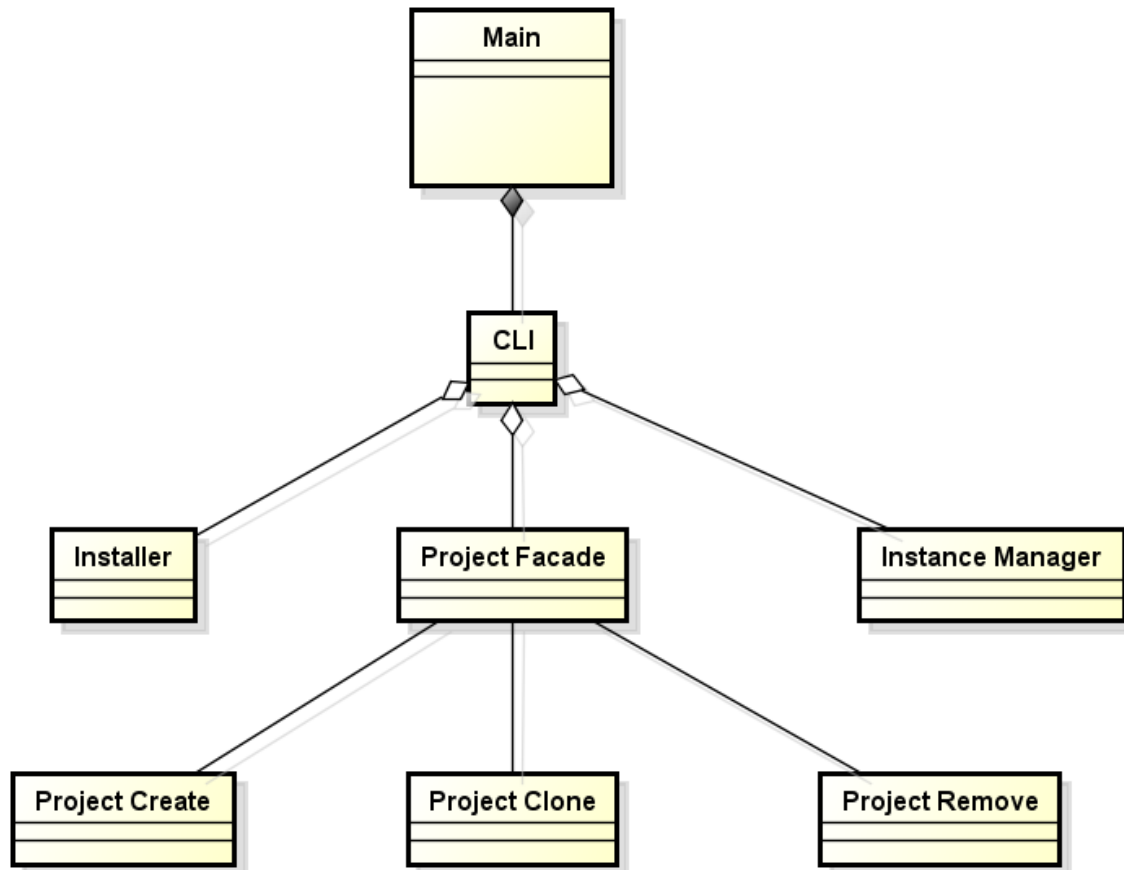


Figura 1: Diagramma delle classi relativo alla gestione del framework da parte dell'utente sviluppatore

#### 3.2.1.2 Descrizione

Descrizione...

#### 3.2.1.3 Classi

##### 3.2.1.3.0.1 CLI

**Nome**

MaaPCLI::CLI

**Descrizione**

Classe che rappresenta l'interfaccia a riga di comando.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per interagire con il framework.

**Relazioni con altre classi**

- **MaaPCLI::Installer**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo Installer per avviare l'installazione del framework;
- **MaaPCLI::ProjectFacade**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo ProjectFacade per creare un nuovo progetto, clonare uno esistente oppure eliminarlo;
- **MaaPCLI::InstanceManager**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo InstanceManager per istanziare una istanza di progetto MaaP precedentemente creato.

**3.2.1.3.0.2 Installer****Nome**

MaaPCLI::Installer

**Descrizione**

Classe che rappresenta lo script di installazione del framework.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per installare il framework e relative dipendenze nel sistema in uso.

**Relazioni con altre classi**

- **MaaPCLI::CLI**  
Relazione entrante, interazione con l'interfaccia a riga di comando.

**3.2.1.3.0.3 InstanceManager****Nome**

MaaPCLI::InstanceManager

**Descrizione**

Classe che rappresenta lo script per l'avvio di un'istanza di un progetto esistente.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per avviare il server caricando un determinato progetto.

**Relazioni con altre classi**

- **MaaPCLI::CLI**  
Relazione entrante, interazione con l'interfaccia a riga di comando.

**3.2.1.3.0.4 ProjectFacade****Nome**

MaaPCLI::ProjectFacade

**Descrizione**

Classe che rappresenta la classe Facade nel design pattern Facade

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per interagire con il framework per la creazione di un nuovo progetto e/o per la clonazione, eliminazione di un progetto esistente.

**Relazioni con altre classi**

- **MaaPCLI::ProjectCreate**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo ProjectCreate per avviare la creazione di un nuovo progetto;
- **MaaPCLI::ProjectClone**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo ProjectClone per avviare la clonazione di un progetto esistente;
- **MaaPCLI::ProjectRemove**  
Relazione uscente, contiene un riferimento ad un oggetto di tipo ProjectRemove per eliminare un progetto esistente;

#### 3.2.1.3.0.5 ProjectCreate

**Nome**

MaaPCLI::ProjectCreate

**Descrizione**

Classe che rappresenta una classe del design patter Facade.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per avviare la creazione di un nuovo progetto.

**Relazioni con altre classi**

- **MaaPCLI::ProjectFacade**  
Relazione entrante, interazione con la facciata ProjectFacade.

#### 3.2.1.3.0.6 ProjectClone

**Nome**

MaaPCLI::ProjectClone

**Descrizione**

Classe che rappresenta una classe del design patter Facade.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per avviare la clonazione di un progetto esistente.

**Relazioni con altre classi**

- **MaaPCLI::ProjectFacade**  
Relazione entrante, interazione con la facciata ProjectFacade.

#### 3.2.1.3.0.7 ProjectRemove

**Nome**

MaaPCLI::ProjectRemove

**Descrizione**

Classe che rappresenta una classe del design patter Facade.

**Utilizzo**

Viene utilizzata dall'utente sviluppatore per eliminare un progetto esistente.

**Relazioni con altre classi**

- **MaaPCLI::ProjectFacade**  
Relazione entrante, interazione con la facciata ProjectFacade.

### 3.2.2 Package

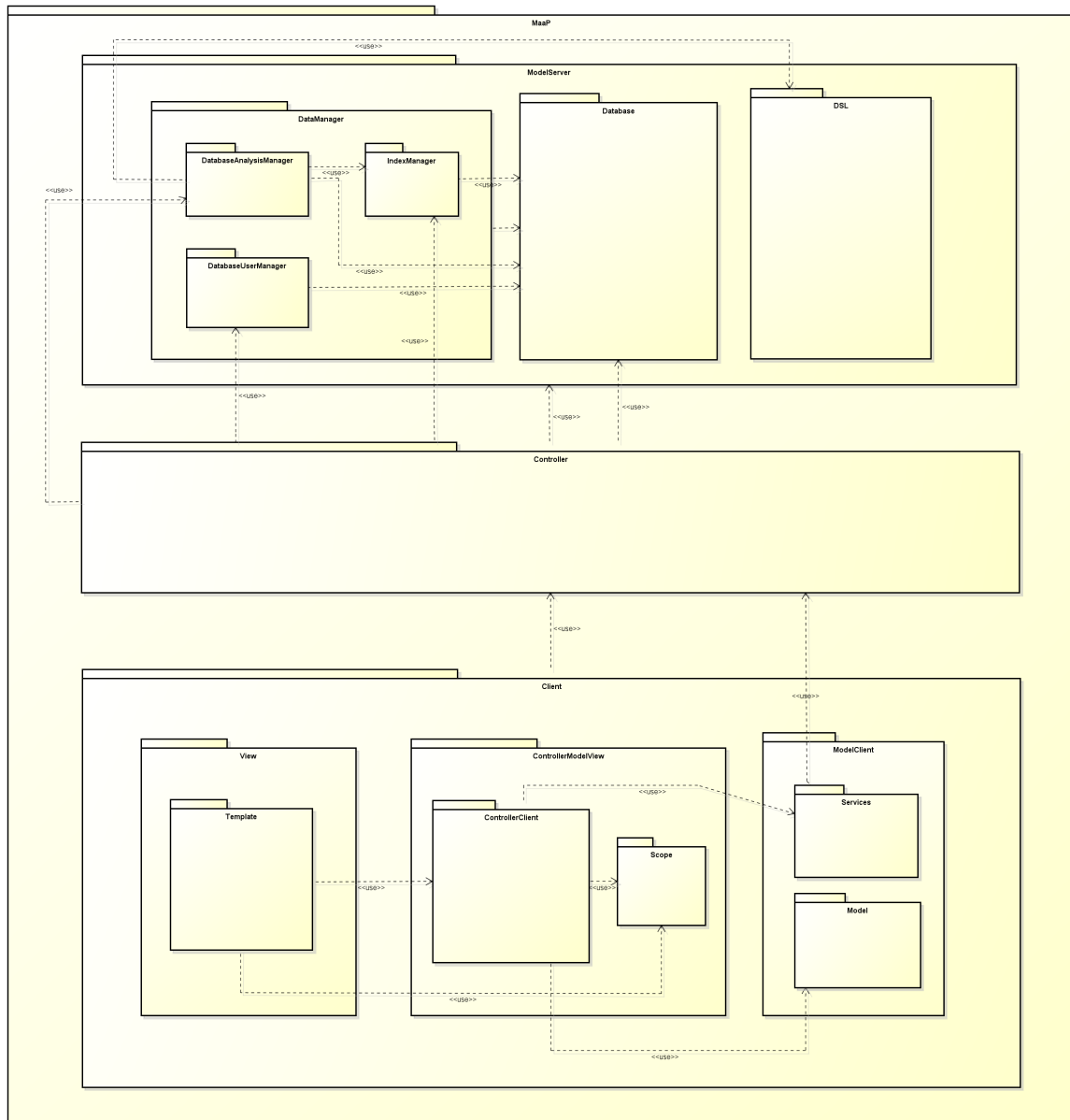


Figura 2: Architettura generale del software - vista package

Nel precedente diagramma sono presenti le relazioni tra i package Client, Controller e ModelServer. Vengono inoltre presentati tutti i sotto-package così da facilitare la comprensione dell'intero sistema.

### 3.2.3 Classi

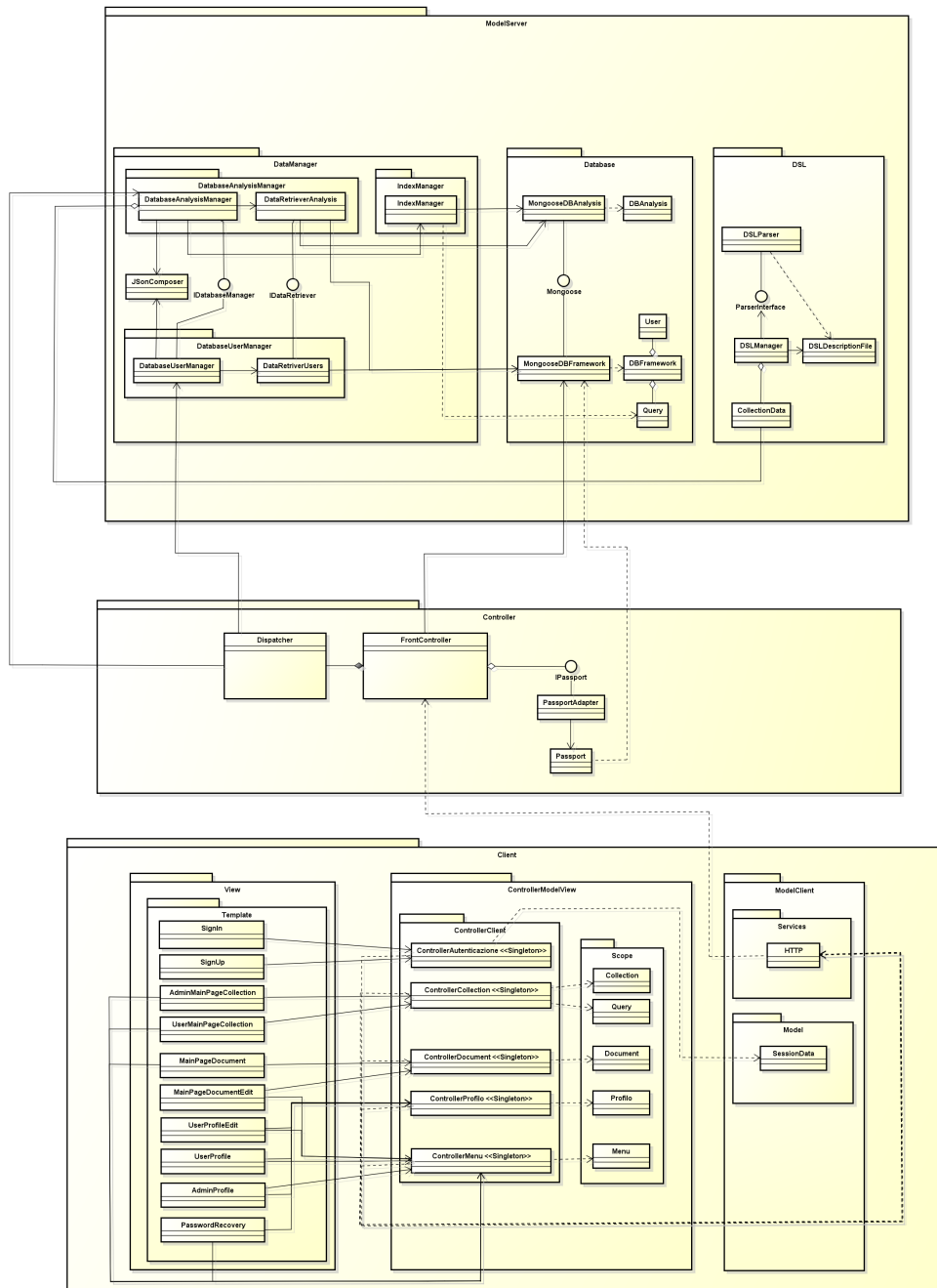


Figura 3: Architettura generale del software

Nel precedente diagramma è presente l'architettura ad alto livello del software e vengono indicate le classi fondamentali per rappresentare le relazioni del modello three-tier. \*\*\*\*\*  
DO DESCRIVERE 3TIER IN APPENDICE. I diagrammi di sequenza relativi allo scambio di segnali, lo scopo ed il contesto di utilizzo sono presenti nella sezione ????.

### 3.2.3.1 ModelServer

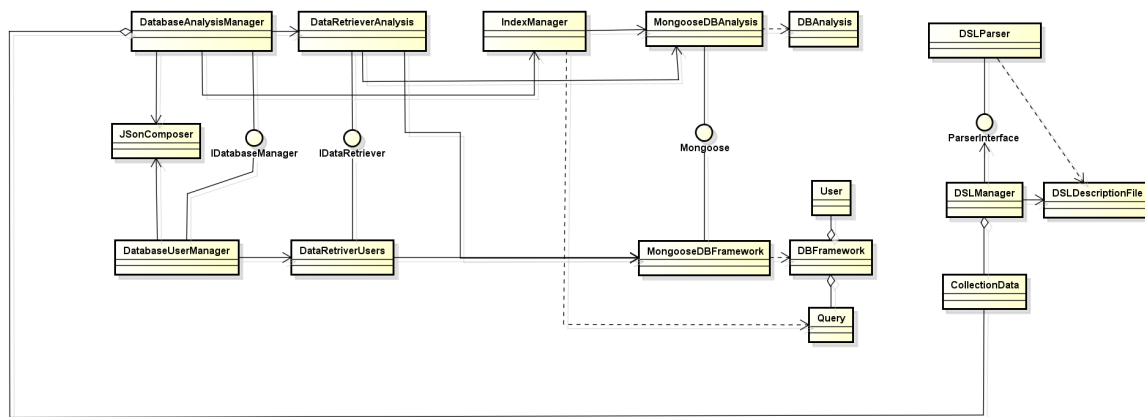


Figura 4: Diagramma delle classi del ModelServer

Nel ModelServer sono presenti oggetti che rappresentano:

- Il database di analisi e quello degli utenti;
- La gestione del file DSL e il suo parsing;
- La gestione dei dati richiesti dal controller.

Tutte le operazioni di gestione, modifica e recupero dei dati vengono messe a disposizione dal model. In tal modo il controller è responsabile solamente di gestire la logica dell'applicazione.



### 3.2.3.2 Controller

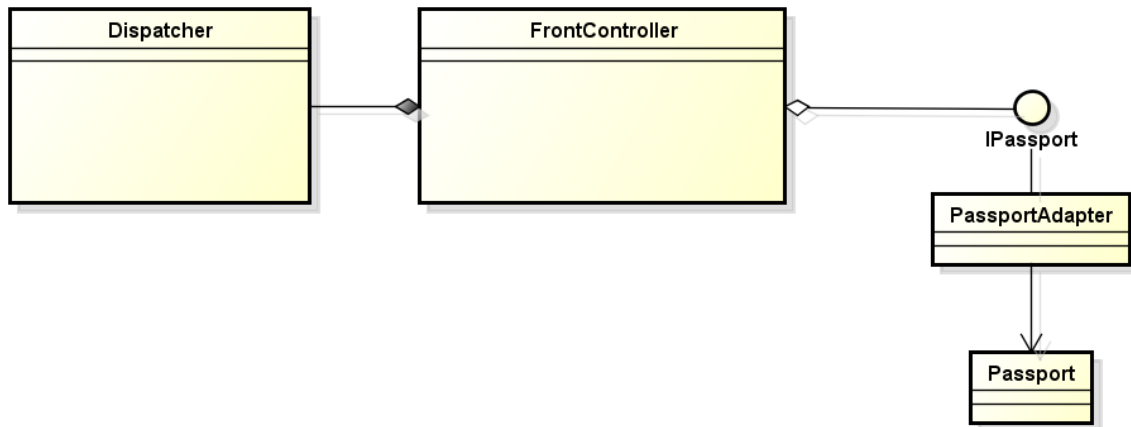


Figura 5: Diagramma delle classi del Controller

Il controller è responsabile dell'autenticazione delle richieste e del loro routing da Client a Model-Server e viceversa.

### 3.2.3.3 Client

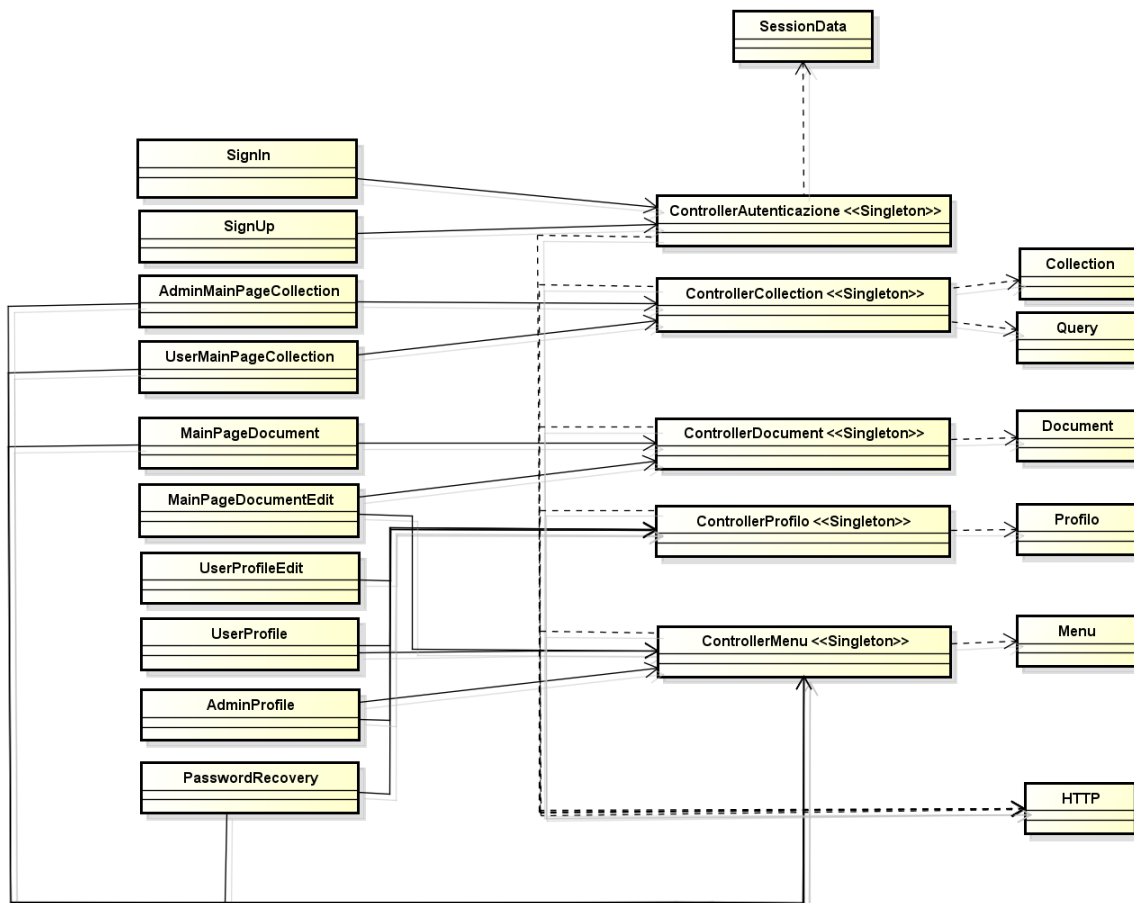


Figura 6: Diagramma delle classi del Client

Nel Client sono presenti oggetti che rappresentano:

- I template per le pagine web;
- I Controller per la gestione dei template;
- Lo Scope per l'aggiornamento dei dati dei template;
- I Servizi utilizzati dai Controller.

## 4 Componenti e Classi

### 4.1 MaaP

#### 4.1.1 Informazioni sul package

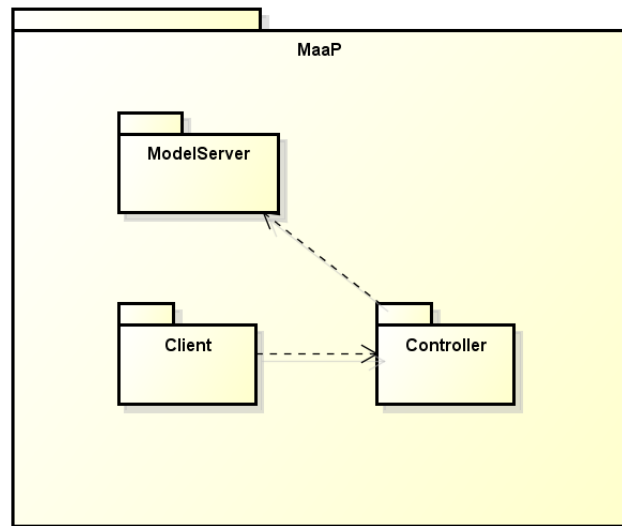


Figura 7: Componenti MaaP

##### 4.1.1.1 Descrizione

Namespace globale per il progetto. Le relazioni tra i package ModelServer, Controller e Client identificano il modello di architettura three-tier.

##### 4.1.1.2 Sotto-componenti

- MaaP::ModelServer
- MaaP::Controller
- MaaP::Client

## 4.2 MaaP::ModelServer

### 4.2.1 Informazioni sul package

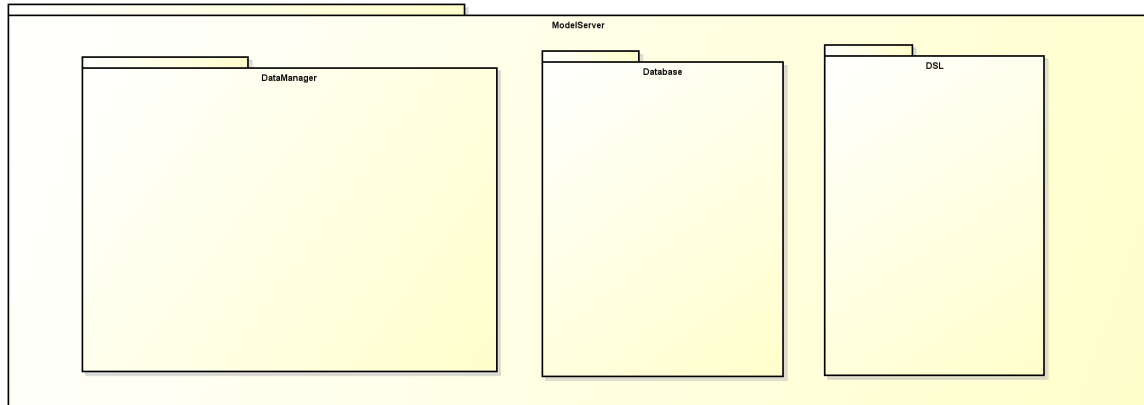


Figura 8: Componente MaaP::ModelServer

#### 4.2.1.1 Descrizione

Package per il componente ModelServer del modello di architettura three-tier.

#### 4.2.1.2 Sottocomponenti

- MaaP::ModelServer::DataManager;
- MaaP::ModelServer::Database;
- MaaP::ModelServer::DSL.

## 4.2.2 MaaP::ModelServer::DataManager

### 4.2.2.1 Informazioni sul package

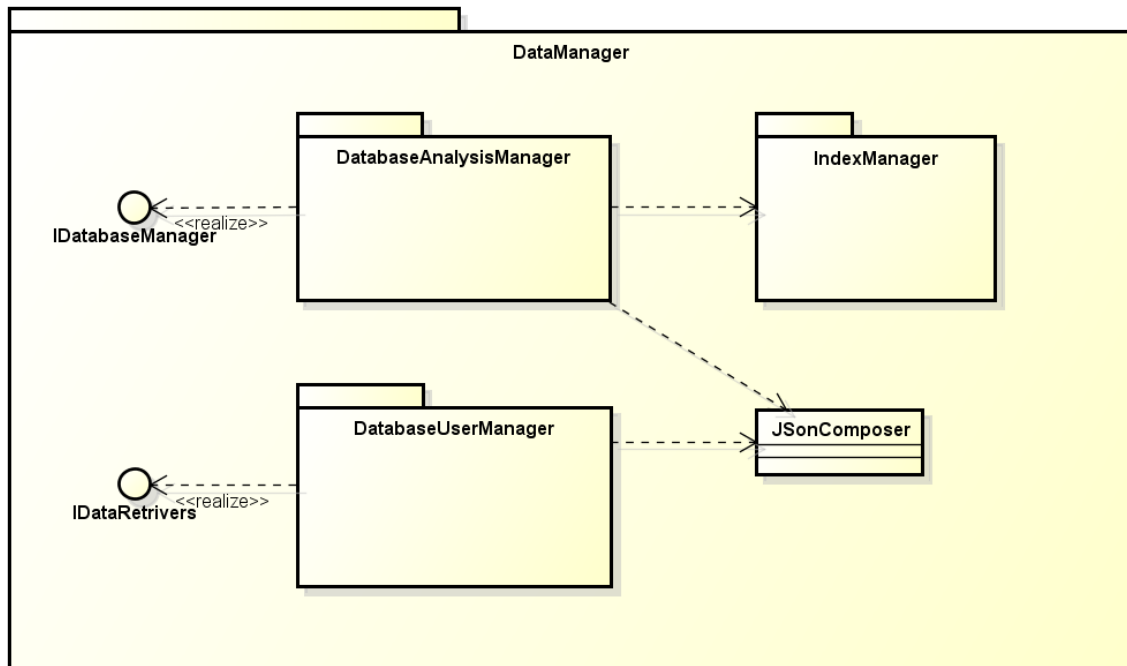


Figura 9: Componente MaaP::ModelServer::DataManager

### 4.2.2.2 Descrizione

Componente parte del ServerModel per la gestione dei dati.

### 4.2.2.3 Sotto-componenti

- MaaP::ModelServer::DataManager::DataBaseAnalysisManager;
- MaaP::ModelServer::DataManager::DatabaseUserManager;
- MaaP::ModelServer::DataManager::IndexManager.

### 4.2.2.4 Classi

#### 4.2.2.4.1 JSonComposer

##### Nome

MaaP::ModelServer::DataManager::JSonComposer

##### Descrizione

Classe che costruisce un file JSON a partire dalla struttura di una Collection, o di un Document, e dai suoi dati.

**Utilizzo**

Viene utilizzata dai DatabaseManager per costruire il file JSON da inviare al Controller.

**4.2.2.4.2 IDatabaseManager****Nome**

MaaP::ModelServer::DataManager::IDatabaseManager

**Descrizione**

Interfaccia che rappresenta il gestore dei database. Contiene tutte le operazioni che si possono effettuare sul database e l'elaborazione dei dati recuperati da essi.

**Utilizzo**

Viene utilizzata per la gestione delle richieste inoltrate dal Controller.

**Classi che ereditano**

- MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager;
- MaaP::ModelServer::DataManager::DatabaseUserManager::DatabaseUserManager.

**4.2.2.4.3 IDataRetriever****Nome**

MaaP::ModelServer::DataManager::IDataRetriever

**Descrizione**

Interfaccia attraverso cui i DatabaseManager dialogano con i database. Contiene le operazioni di lettura e scrittura nei database.

**Utilizzo**

Viene utilizzata per recuperare e inserire dati, sui database, su richiesta dei DataManager.

**Classi che ereditano**

- MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DataRetrieverAnalysis;
- MaaP::ModelServer::DataManager::DatabaseUserManager::DataRetrieverUsers.

#### 4.2.2.5 MaaP::ModelServer::DataManager::DatabaseAnalysisManager

##### 4.2.2.5.1 Informazioni sul package

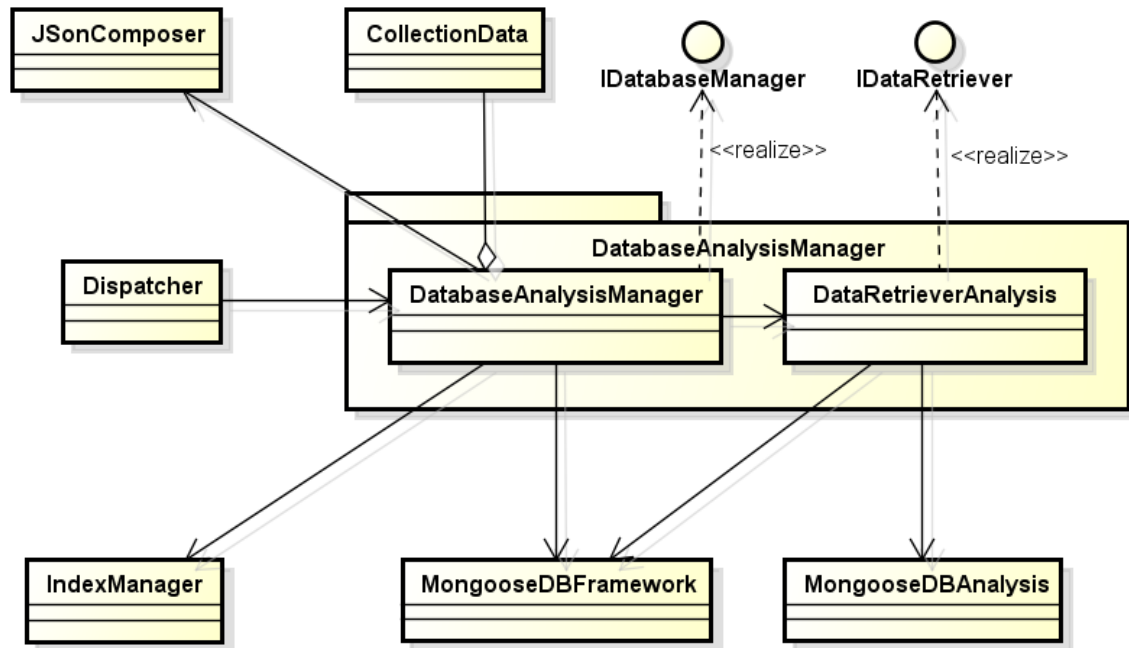


Figura 10: Componente MaaP::ModelServer::DataManager::DatabaseAnalysisManager

##### 4.2.2.5.2 Descrizione

Componente parte del DataManager per la gestione dei dati del database di analisi.

##### 4.2.2.5.3 Classi

###### 4.2.2.5.3.1 DatabaseAnalysisManager

###### Nome

MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager

###### Descrizione

Classe che rappresenta il gestore dei database di analisi. Contiene tutte le operazioni che si possono effettuare sul database di analisi e l'elaborazione dei dati recuperati da essi.

###### Utilizzo

Viene utilizzata per la gestione delle richieste, relative al database di analisi, inoltrate dal Controller.

###### Classi da cui eredita

- MaaP::ModelServer::DataManager::IDatabaseManager;

###### Relazioni con altre classi

- **MaaP::ModelServer::DataManager::JsonComposer**  
Relazione uscente, utilizza un riferimento a un oggetto di tipo JsonComposer per ottenere il JSON da spedire;
- **MaaP::ModelServer::DSL::CollectionData**  
Relazione uscente, utilizza un riferimento a un oggetto CollectionData che contiene la struttura di un file di descrizione;
- **MaaP::ModelServer::DataManager::DataAnalysisManager::DataRetrieverAnalysis**  
Relazione uscente, utilizza un riferimento a un oggetto DataRetrieverAnalysis per relazionarsi con il database di analisi;
- **MaaP::ModelServer::DataManager::IndexManager::IndexManager**  
Relazione uscente, utilizza un riferimento a un oggetto IndexManager per la creazione degli indici;
- **MaaP::Controller::Dispatcher**  
Relazione entrante, interazioni con le funzionalità del gestore del database di analisi.

#### 4.2.2.5.3.2 DatabaseRetrieverAnalysis

**Nome**

MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseRetrieverAnalysis

**Descrizione**

Classe che rappresenta l'oggetto per interagire con i database.

**Utilizzo**

Viene utilizzata per inserire e leggere dati sui database di analisi e framework.

**Classi da cui eredita**

- MaaP::ModelServer::DataManager::IDataRetriever;

**Relazioni con altre classi**

- **MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager**  
Relazione entrante, interazione con il database;
- **MaaP::ModelServer::Database::MongooseDBAnalysis**  
Relazione uscente, utilizza un riferimento a un oggetto di tipo MongooseDBAnalysis per creare lo schema dei dati del database di analisi e per interagire con essi;
- **MaaP::ModelServer::Database::MongooseDBFramework**  
Relazione uscente, utilizza un riferimento a un oggetto di tipo MongooseDBFramework per creare lo schema dei dati del database del framework e per interagire con essi;



#### 4.2.2.6 MaaP::ModelServer::DataManager::DatabaseUserManager

##### 4.2.2.6.1 Informazioni sul package

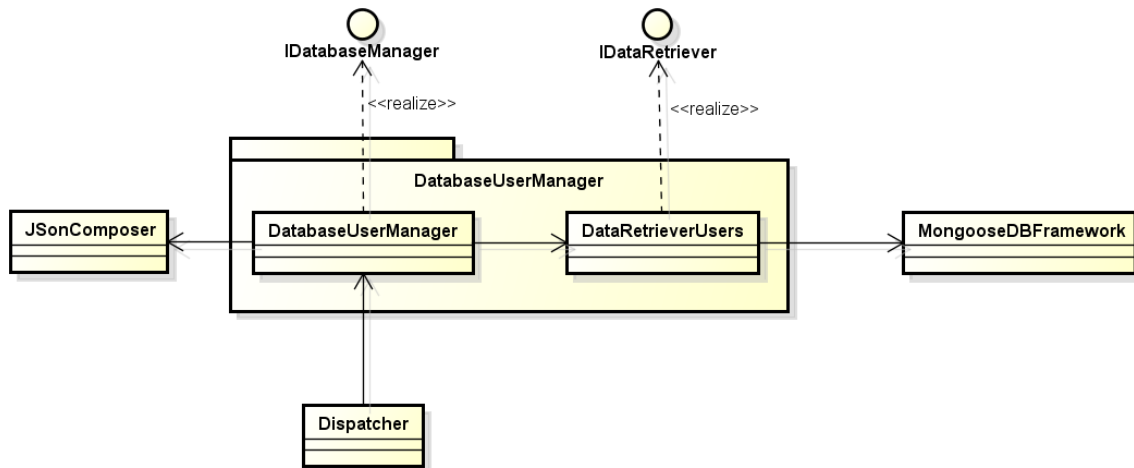


Figura 11: Componente MaaP::ModelServer::DataManager::DatabaseUserManager

##### 4.2.2.6.2 Descrizione

Componente parte del DataManager per la gestione dei dati del database del framwork che comprende sia dati utente che impostazioni del sistema.

##### 4.2.2.6.3 Classi

###### 4.2.2.6.3.1 DatabaseUserManager

###### Nome

MaaP::ModelServer::DataManager::DatabaseUserManager::DatabaseUserManager

###### Descrizione

Classe che rappresenta il gestore del database del framework. Contiene tutte le operazioni che si possono effettuare sul database del framework e l'elaborazione dei dati recuperati da esso.

###### Utilizzo

Viene utilizzata per la gestione delle richieste relative al database del framework inoltrate dal Controller.

###### Classi da cui eredita

- MaaP::ModelServer::DataManager::IDatabaseManager;

###### Relazioni con altre classi

- MaaP::ModelServer::DataManager::JsonComposer  
Relazione uscente, utilizza un riferimento a un oggetto di tipo JsonComposer per ottenere il JSON da spedire;

- **MaaP::ModelServer::DataManager::DataUserManager::DataRetrieverUsers**  
Relazione uscente, utilizza un riferimento a un oggetto DataRetrieverUsers per relazionarsi con il database del framework;
- **MaaP::Controller::Dispatcher**  
Relazione entrante, interazioni con le funzionalità del gestore del database di analisi.

#### 4.2.2.6.3.2 DataRetrieverUsers

##### Nome

MaaP::ModelServer::DataManager::DatabaseUserManager::DataRetrieverUsers

Classe che rappresenta l'oggetto per interagire con il database del framework.

##### Utilizzo

Viene utilizzata per inserire e leggere dati sul database del framework.

##### Classi da cui eredita

- MaaP::ModelServer::DataManager::IDataRetriever;

##### Relazioni con altre classi

- **MaaP::ModelServer::DataManager::DatabaseUserManager::DatabaseUserManager**  
Relazione entrante, interazione con il database;
- **MaaP::ModelServer::Database::MongooseDBFramework**  
Relazione uscente, utilizza un riferimento a un oggetto di tipo MongooseDBFramework per creare lo schema dei dati del database del framework e per interagire con essi.

#### 4.2.2.7 MaaP::ModelServer::DataManager::IndexManager

##### 4.2.2.7.1 Informazioni sul package

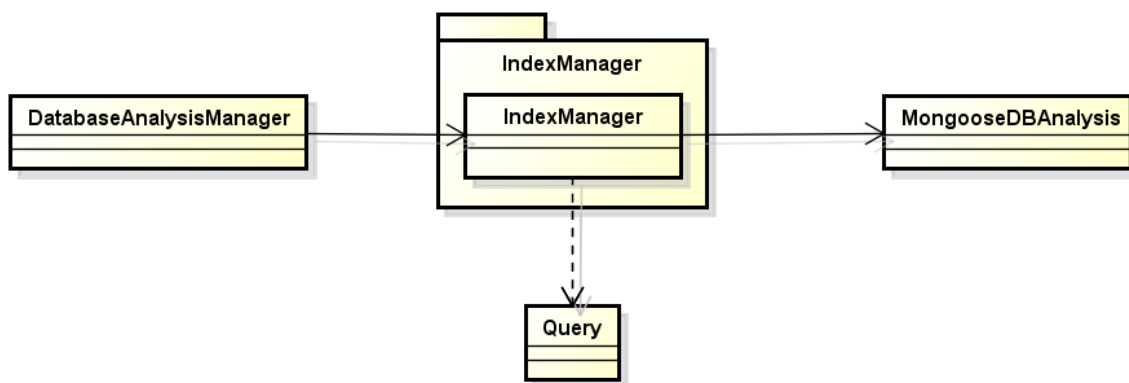


Figura 12: Componente MaaP::ModelServer::DataManager::IndexManager

##### 4.2.2.7.2 Descrizione

Componente parte del DataManager per la creazione e gestione degli indici.

#### 4.2.2.7.3 Classi

##### 4.2.2.7.3.1 IndexManager

**Nome**

MaaP::ModelServer::DataManager::IndexManager::IndexManager

**Descrizione**

Classe che rappresenta il gestore degli indici. Contiene tutte le operazioni per la creazione degli indici.

**Utilizzo**

Viene utilizzata per la creazione di indici personalizzati su richiesta del DatabaseAnalysisManager.

**Relazioni con altre classi**

- **MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager**  
Relazione entrante, interazione con il database;
- **MaaP::ModelServer::DataManager::Database::MongooseDBAnalysis**  
Relazione uscente, utilizza un riferimento ad un oggetto di tipo MongooseDBAnalysis per creare lo schema dei dati del database di analisi e per interagire con essi;
- **MaaP::ModelServer::DataManager::Database::Query**  
Relazione uscente debole, utilizza un riferimento ad un oggetto Query per il recupero delle query più utilizzate.

### 4.2.3 MaaP::ModelServer::Database

#### 4.2.3.1 Informazioni sul package

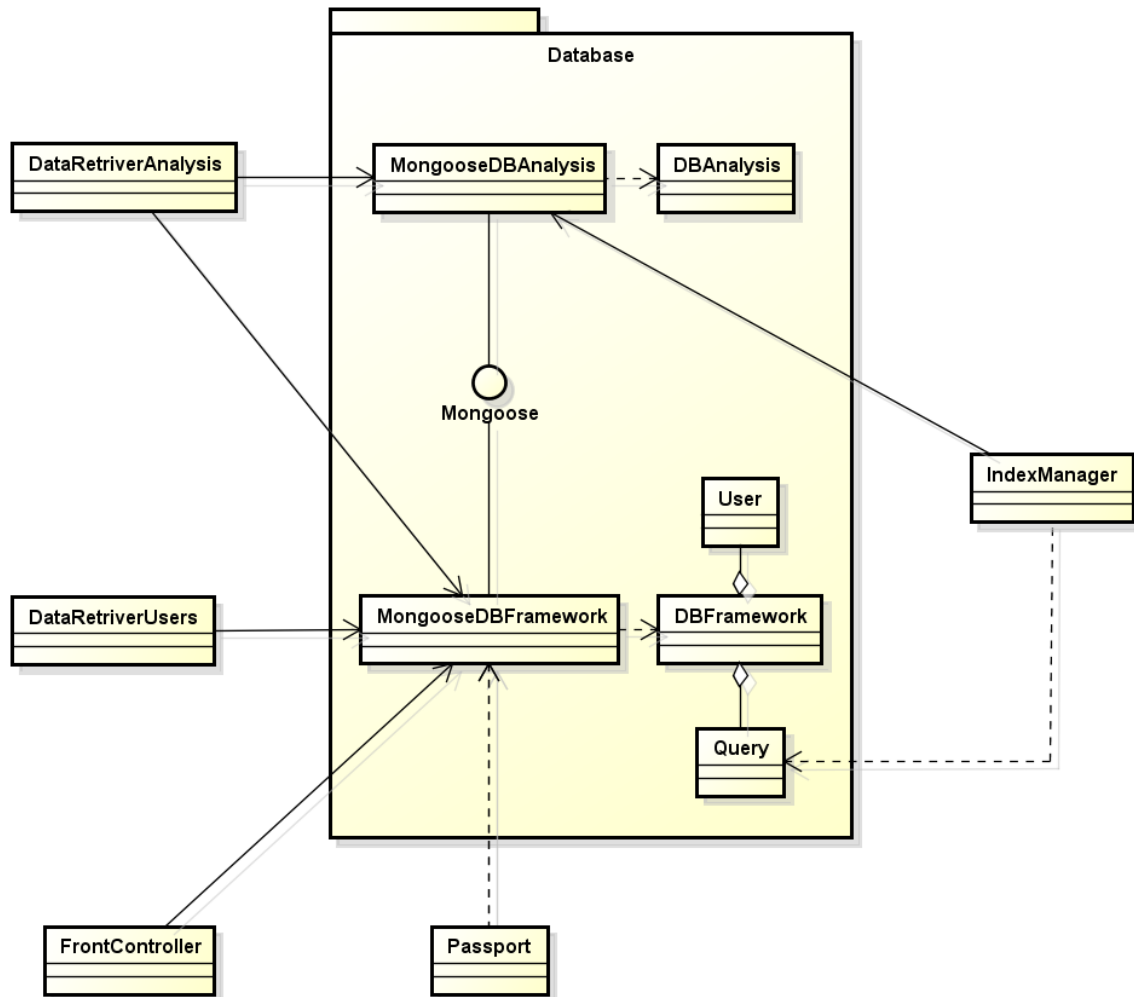


Figura 13: Componente MaaP::ModelServer::Database

#### 4.2.3.2 Descrizione

Componente parte del ModelServer per la gestione dei dati.

#### 4.2.3.3 Classi

##### 4.2.3.3.1 MongooseDBAnalysis

Nome

MaaP::ModelServer::Database::MongooseDBAnalysis

**Descrizione**

Classe che rappresenta l'interfaccia di connessione con il database di analisi.

**Utilizzo**

Viene utilizzata per interfacciarsi con il database di analisi fornendo uno schema adeguato.

**Classi da cui eredita**

- `MaaP::ModelServer::Database::Mongoose`;

**Relazioni con altre classi**

- **`MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DataRetrieverAnalysis`**  
Relazione entrante, interazione con il database di analisi;
- **`MaaP::ModelServer::DataManager::IndexManager::IndexManager`**  
Relazione entrante, interazione con il database di analisi;
- **`MaaP::ModelServer::Database::DBAnalysis`**  
Relazione uscente debole, utilizza un riferimento al database di analisi a cui connettersi.

**4.2.3.3.2 DBAnalysis****Nome**

`MaaP::ModelServer::Database::DBAnalysis`

**Descrizione**

Classe che rappresenta il database di analisi.

**Utilizzo**

Viene utilizzata per contenere i dati di analisi.

**Relazioni con altre classi**

- **`MaaP::ModelServer::Database::MongooseDBAnalysis`**  
Relazione entrante debole, interazione con il database di analisi.

**4.2.3.3.3 Mongoose****Nome**

`MaaP::ModelServer::Database::Mongoose`

**Descrizione**

Interfaccia che permette di dialogare con i database utilizzando Mongoose.

**Utilizzo**

Viene utilizzata per interfacciarsi con i vari database.

**Classi che ereditano**

- `MaaP::ModelServer::Database::MongooseDBAnalysis`;
- `MaaP::ModelServer::Database::MongooseDBFramework`.

**4.2.3.3.4 MongooseDBFramework****Nome**

`MaaP::ModelServer::Database::MongooseDBMongooseDBFramework`

**Descrizione**

Classe che rappresenta l'interfaccia di connessione con il database del framework.

**Utilizzo**

Viene utilizzata per interfacciarsi con il database del framework fornendo uno schema adeguato.

**Classi da cui eredita**

- **MaaP::ModelServer::Database::Mongoose;**

#### Relazioni con altre classi

- **MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DataRetrieverAnalysis**  
Relazione entrante, interazione con il database del framework;
- **MaaP::ModelServer::DataManager::DatabaseUserManager::DataRetrieverUsers**  
Relazione entrante, interazione con il database del framework;
- **MaaP::ModelServer::Database::DBFramework**  
Relazione uscente debole, utilizza un riferimento al database del framework a cui connettersi.

#### 4.2.3.3.5 DBFramework

##### Nome

MaaP::ModelServer::Database::DBFramework

##### Descrizione

Classe che rappresenta il database del framework.

##### Utilizzo

Viene utilizzata per contenere i dati utente ed impostazioni varie del sistema.

##### Relazioni con altre classi

- **MaaP::ModelServer::Database::User**  
Relazione uscente, utilizza un riferimento ad un oggetto User per gestire i dati utente.
- **MaaP::ModelServer::Database::Query**  
Relazione uscente, utilizza un riferimento ad un oggetto Query per gestire la lista di query fin'ora effettuate dal sistema;
- **MaaP::Controller::FrontController**  
Relazione entrante, interazione con il database del framework;
- **MaaP::Controller::Passport**  
Relazione entrante debole, interazione con il database del framework.

#### 4.2.3.3.6 User

##### Nome

MaaP::ModelServer::Database::User

##### Descrizione

Classe che rappresenta la parte contenuta nel database del framework relativa ai dati utenti.

##### Utilizzo

Viene utilizzata per contenere i dati utente.

##### Relazioni con altre classi

- **MaaP::ModelServer::Database::DBFramework**  
Relazione entrante, interazione con i dati utente.

#### 4.2.3.3.7 Query

**Nome**

MaaP::ModelServer::Database::Query

**Descrizione**

Classe che rappresenta la parte contenuta nel database del framework relativa alle query effettuate del sistema.

**Utilizzo**

Viene utilizzata per contenere le query effettuate del sistema.

**Relazioni con altre classi**

- **MaaP::ModelServer::Database::DBFramework**  
Relazione entrante, interazione con le query effettuate del sistema.
- **MaaP::ModelServer::DataManager::IndexManager::IndexManager**  
Relazione entrante debole, interazione con le query effettuate del sistema.

#### 4.2.4 MaaP::ModelServer::DSL

##### 4.2.4.1 Informazioni sul package

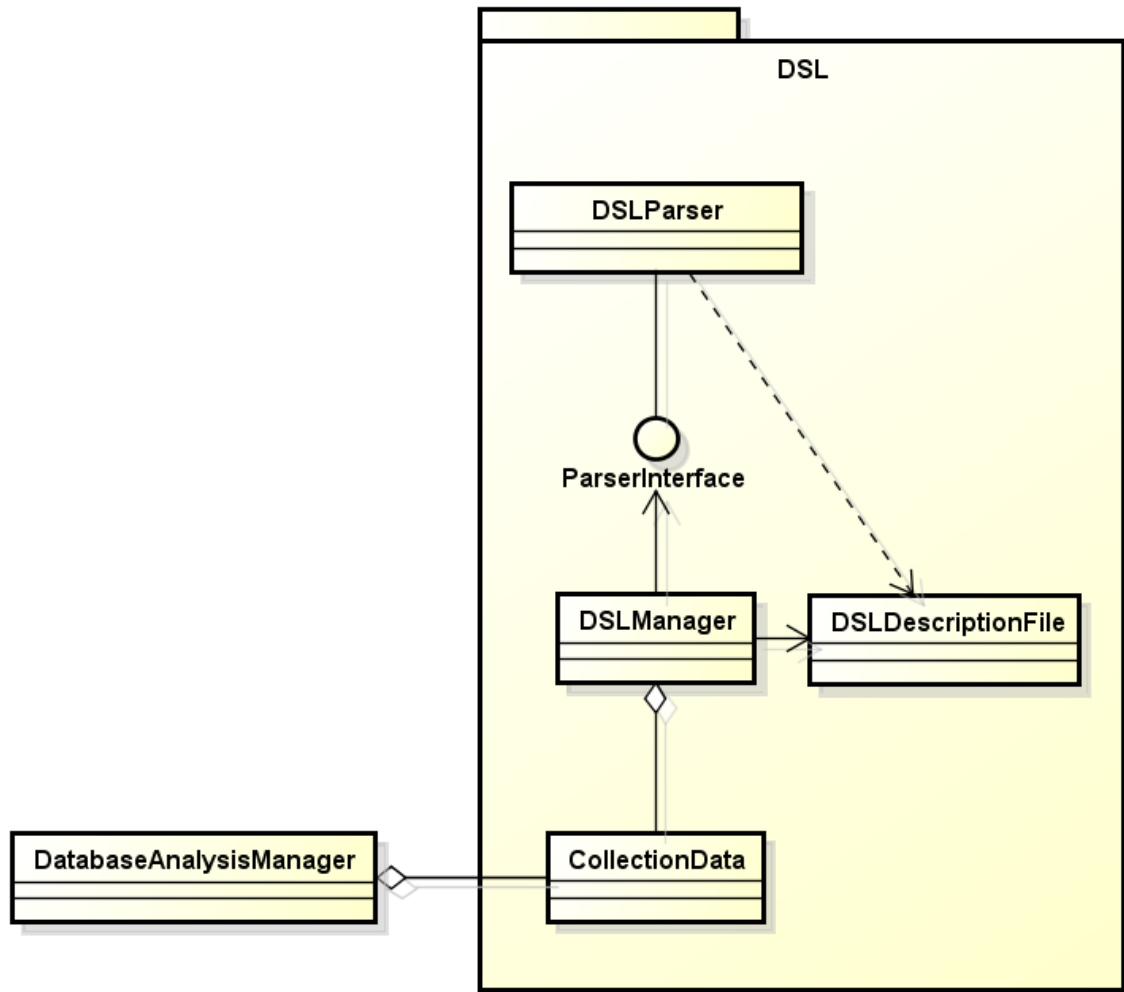


Figura 14: Componente MaaP::ModelServer::DSL

##### 4.2.4.2 Descrizione

Componente parte del ServerModel per la gestione dei file di descrizione.



#### 4.2.4.3 Sotto-componenti

##### 4.2.4.4 Classi

###### 4.2.4.4.1 ParserInterface

**Nome**

MaaP::ModelServer::DSL::ParserInterface

**Descrizione**

Interfaccia che rappresenta la componente interfaccia del design pattern strategy per il parser di un linguaggio DSL.

**Utilizzo**

Viene utilizzata per la effettuare il parsing di un file di descrizione.

**Classi che ereditano**

- MaaP::ModelServer::DSL::DSLParser.

###### 4.2.4.4.2 DSLParser

**Nome**

MaaP::ModelServer::DSL::DSLParser

**Descrizione**

Classe che rappresenta l'algoritmo per il parser DSL del design pattern strategy.

**Utilizzo**

Viene utilizzata all'avvio del sistema per eseguire il parsing dei file di descrizione. **Classi da cui eredita**

- MaaP::ModelServer::DSL::ParserInterface;

**Relazioni con altre classi**

- **MaaP::ModelServer::DSL::DSLDescriptionFile**

Relazione uscente debole, utilizza un riferimento ad un oggetto DSLDescriptionFile per leggere il file di descrizione;

###### 4.2.4.4.3 DSLManager

**Nome**

MaaP::ModelServer::DSL::DSLManager

**Descrizione**

Classe che rappresenta il gestore dei file di descrizione. Contiene tutte le operazioni per eseguire il parsing dei file di descrizione e per salvare il risultato su appositi file di tipo CollectionData

**Utilizzo**

Viene utilizzata all'avvio del sistema per eseguire il parsing dei file di descrizione e salvare il risultato su file. **Classi da cui eredita Relazioni con altre classi**

- **MaaP::ModelServer::DSL::ParserInterface**

Relazione uscente, utilizza un riferimento ad un oggetto ParserInterface per eseguire il parsing del file di descrizione;

- **MaaP::ModelServer::DSL::DSLDescriptionFile**

Relazione uscente, utilizza un riferimento ad un oggetto DSLDescriptionFile per leggere il file di descrizione;

- **MaaP::ModelServer::DSL::CollectionData**

Relazione uscente, utilizza un riferimento ad un oggetto CollectionData per salvare i risultati dell'operazione di parsing.

#### 4.2.4.4.4 CollectionData

##### Nome

MaaP::ModelServer::DSL::CollectionData

##### Descrizione

Classe che rappresenta il file contenente il risultato dell'operazione di parsing.

##### Utilizzo

Viene utilizzata all'avvio del sistema per salvare il risultato dell'operazione di parsing del file di descrizione. **Relazioni con altre classi**

- **MaaP::ModelServer::DSL::DSLManager**  
Relazione entrante, interazione con il file;
- **MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager**  
Relazione entrante, interazione con il file.

### 4.3 MaaP::Controller

#### 4.3.1 Informazioni sul package

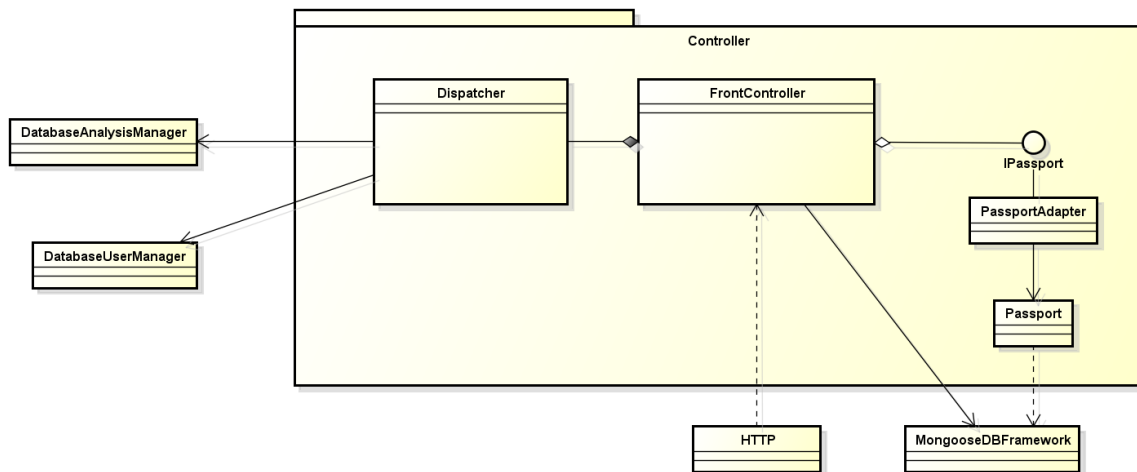


Figura 15: Componente MaaP::Controller

#### 4.3.1.1 Descrizione

Package per il componente Controller del modello di architettura three-tier.

#### 4.3.1.2 Classi

##### 4.3.1.2.1 IPassport

**Nome**

MaaP::Controller::IPassport

**Descrizione**

Interfaccia che rappresenta il componente target del design pattern object adapter.

**Utilizzo**

Viene utilizzata per gestire l'autenticazione utente. **Relazioni con altre classi**

- **MaaP::Controller::FrontController**  
Relazione entrante, interazione con il gestore dell'autenticazione.

**Classi che ereditano**

- MaaP::Controller::PassportAdapter.

##### 4.3.1.2.2 PassportAdapter

**Nome**

MaaP::Controller::PassportAdapter

**Descrizione**

Classe che rappresenta il componente adapter del design pattern object adapter.

**Utilizzo**

Viene utilizzata per gestire l'autenticazione utente. **Classi da cui eredita**

- MaaP::Controller::IPassport.

**Relazioni con altre classi**

- **MaaP::Controller::Passport**  
Relazione uscente, utilizza un riferimento ad un oggetto di tipo Passport per gestire l'autenticazione utente.

##### 4.3.1.2.3 Passport

**Nome**

MaaP::Controller::Passport

**Descrizione**

Classe che rappresenta il componente adaptee del design patter object adapter.

**Utilizzo**

Viene utilizzata per gestire l'autenticazione utente. **Relazioni con altre classi**

- **MaaP::ModelServer::Database::MongooseDBFramework**  
Relazione uscente debole, utilizza un riferimento ad un oggetto MongooseDBFramework per accedere ai dati utente.

##### 4.3.1.2.4 FrontController

**Nome**

MaaP::Controller::FrontController

**Descrizione**

Classe che rappresenta il componente controller del design patter Front Controller.

**Utilizzo**

Viene utilizzata per gestire le richieste del client ed inoltrarle al dispatcher. **Relazioni con altre classi**

- **MaaP::Controller::IPassport**  
Relazione uscente, contiene un riferimento ad un oggetto IPassport per gestire l'autenticazione utente;
- **MaaP::Controller::Dispatcher**  
Relazione uscente, contiene un riferimento ad un oggetto Dispatcher per smistare le richieste del client ai vari manager;
- **MaaP::ModelServer::Database::MongooseDBFramework**  
Relazione uscente, contiene un riferimento ad un oggetto MongooseDBFramework per inserire nuovi dati nel database del framework relativi a nuovi utenti;
- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione entrante debole, interazione con il servizio HTTP.

**4.3.1.2.5 Dispatcher****Nome**

MaaP::Controller::Dispatcher

**Descrizione**

Classe che rappresenta il componente dispatcher del design patter Front Controller.

**Utilizzo**

Viene utilizzata per smistare le richieste del client ai vari gestori dei dati. **Relazioni con altre classi**

- **MaaP::Controller::FrontController**  
Relazione entrante, interazione con il FrontController;
- **MaaP::ModelServer::DataManager::DatabaseAnalysisManager::DatabaseAnalysisManager**  
Relazione uscente, contiene un riferimento ad un oggetto DatabaseAnalysisManager per richiedere azioni relative ai dati di analisi;
- **MaaP::ModelServer::DataManager::DatabaseUserManager::DatabaseUserManager**  
Relazione uscente, contiene un riferimento ad un oggetto DatabaseUserManager per richiedere azioni relative ai dati utenti ed impostazioni di sistema.

## 4.4 MaaP::Client

### 4.4.1 Informazioni sul package

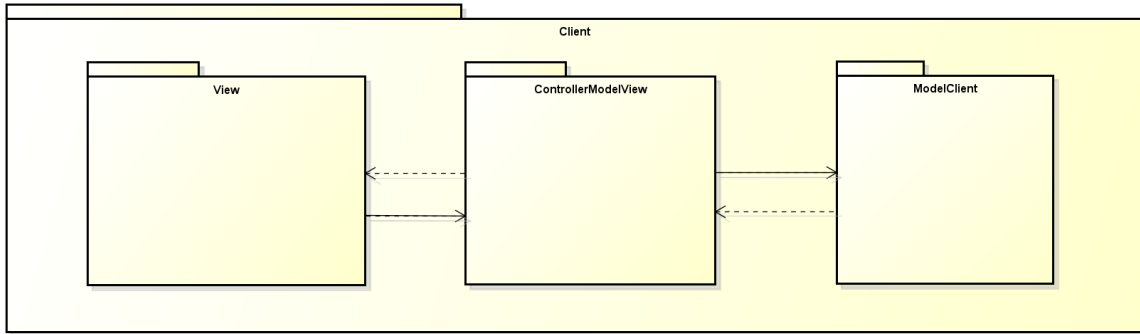


Figura 16: Componente MaaP::Client

#### 4.4.1.1 Descrizione

Package per il componente Client del modello di architettura three-tier.

#### 4.4.1.2 Sottocomponenti

- MaaP::Client::View;
- MaaP::Client::ControllerModelView;
- MaaP::Client::ModelClient.

#### 4.4.2 MaaP::Client::View

##### 4.4.2.1 Informazioni sul package

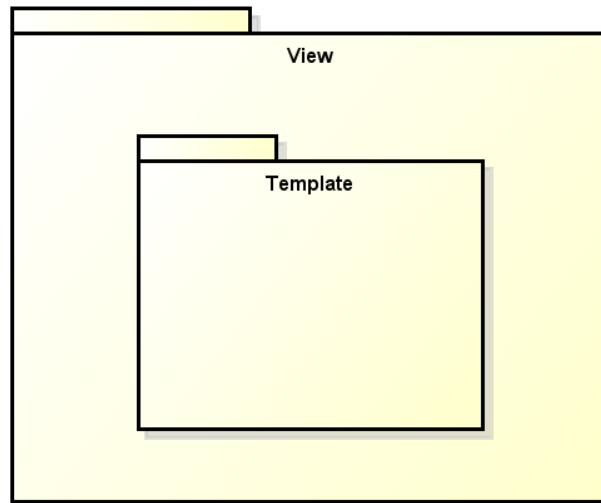


Figura 17: Componente MaaP::Client::View

##### 4.4.2.2 Descrizione

Componente view del design pattern MVVM.

##### 4.4.2.3 Sotto-componenti

- MaaP::Client::Template.

#### 4.4.2.4 MaaP::Client::View::Template

#### 4.4.2.5 Informazioni sul package

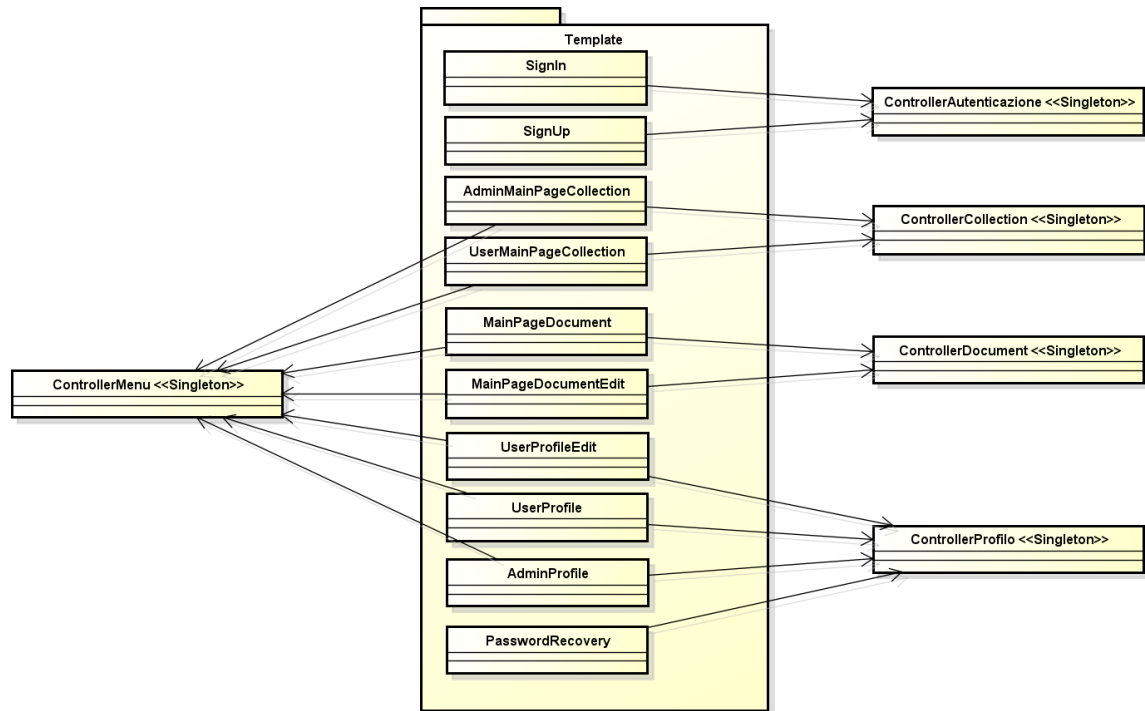


Figura 18: Componente MaaP::Client::View::Template

#### 4.4.2.6 Descrizione

Componente che contiene i template per la visualizzazione delle pagine web.

#### 4.4.2.7 Classi

##### 4.4.2.7.1 SignIn

###### Nome

MaaP::Client::View::Template::SignIn

###### Descrizione

Classe che rappresenta il template per la pagina di login.

###### Utilizzo

Viene utilizzata per renderizzare la pagina web di login.

###### Relazioni con altre classi

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerAutenticazione**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerAutenticazione per gestire il login utente.

#### 4.4.2.7.2 SignUp

**Nome**

MaaP::Client::View::Template::SignUp

**Descrizione**

Classe che rappresenta il template per la pagina di registrazione.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di registrazione utente. **Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerAutenticazione**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerAutenticazione per gestire la registrazione di un nuovo utente.

#### 4.4.2.7.3 AdminMainPageCollection

**Nome**

MaaP::Client::View::Template::AdminMainPageCollection

**Descrizione**

Classe che rappresenta il template per la pagina di visualizzazione Collection per l'utente amministratore.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di visualizzazione Collection per l'utente amministratore.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerCollection per gestire la visualizzazione della pagina Collection;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.4 UserMainPageCollection

**Nome**

MaaP::Client::View::Template::UserMainPageCollection

**Descrizione**

Classe che rappresenta il template per la pagina di visualizzazione Collection per l'utente.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di visualizzazione Collection per l'utente.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerCollection per gestire la visualizzazione della pagina Collection;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.



#### 4.4.2.7.5 MainPageDocument

**Nome**

MaaP::Client::View::Template::MainPageDocument

**Descrizione**

Classe che rappresenta il template per la pagina di visualizzazione Document.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di visualizzazione del Document.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerDocument**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerDocument per gestire la visualizzazione della pagina Document;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.6 MainPageDocumentEdit

**Nome**

MaaP::Client::View::Template::MainPageDocumentEdit

**Descrizione**

Classe che rappresenta il template per la pagina di modifica dei Document.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di modifica dei Document.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerDocument**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerDocument per gestire la visualizzazione della pagina di modifica dei Document;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.7 UserProfileEdit

**Nome**

MaaP::Client::View::Template::UserProfileEdit

**Descrizione**

Classe che rappresenta il template per la pagina di modifica del profilo utente.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di modifica del profilo utente.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerProfilo per gestire la visualizzazione della pagina di modifica del profilo utente;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.8 UserProfile

**Nome**

MaaP::Client::View::Template::UserProfile

**Descrizione**

Classe che rappresenta il template per la pagina di visualizzazione del profilo utente.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di visualizzazione del profilo utente.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerProfilo per gestire la visualizzazione della pagina del profilo utente;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.9 AdminProfile

**Nome**

MaaP::Client::View::Template::AdminProfile

**Descrizione**

Classe che rappresenta il template per la pagina di visualizzazione del profilo utente amministratore.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web di visualizzazione del profilo utente amministratore.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerProfilo per gestire la visualizzazione della pagina del profilo utente amministratore;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.2.7.10 PasswordRecovery

**Nome**

MaaP::Client::View::Template::UserProfile

**Descrizione**

Classe che rappresenta il template per la pagina di recupero password.

**Utilizzo**

Viene utilizzata per renderizzare la pagina web recupero password. **Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerProfilo per gestire la visualizzazione della pagina di recupero password;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione uscente, contiene un riferimento ad un oggetto ControllerMenu per gestire la visualizzazione del menù.

#### 4.4.3 MaaP::Client::ControllerModelView

##### 4.4.3.1 Informazioni sul package

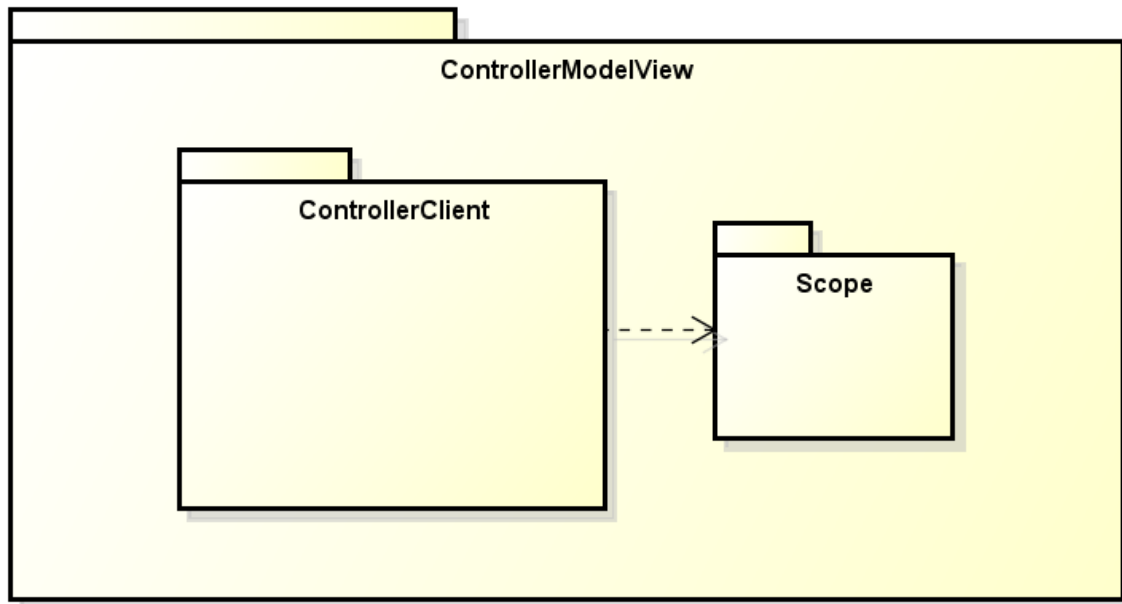


Figura 19: Componente MaaP::Client::ControllerModelView

##### 4.4.3.2 Descrizione

Componente ModelView del design pattern MVVM.

##### 4.4.3.3 Sotto-componenti

- MaaP::Client::ControllerModelView::ControllerClient;
- MaaP::Client::ControllerModelView::Scope.

#### 4.4.3.4 MaaP::Client::ControllerModelView::ControllerClient

##### 4.4.3.4.1 Informazioni sul package

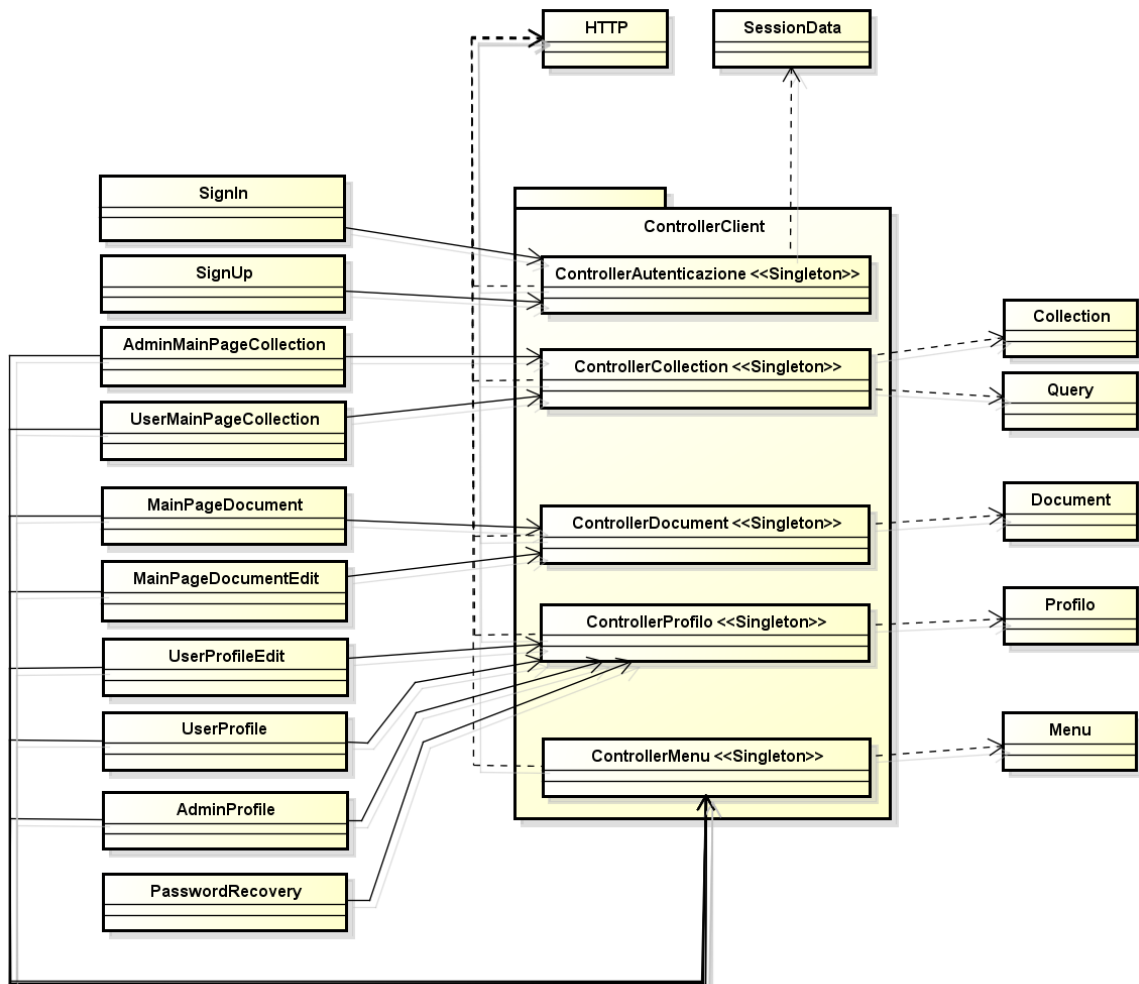


Figura 20: Componente MaaP::Client::ControllerModelView::ControllerClient

##### 4.4.3.4.2 Descrizione

Componente parte del ControllerModelView contenente i vari controller.

##### 4.4.3.4.3 Classi

###### 4.4.3.4.3.1 ControllerAutenticazione

Nome

MaaP::Client::ControllerModelView::ControllerClient::ControllerAutenticazione

**Descrizione**

Classe che rappresenta il controller per indirizzare le richieste di autenticazione e registrazione.

**Utilizzo**

Viene utilizzata per la indirizzare le richieste di autenticazione e registrazione.

**Relazioni con altre classi**

- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione uscente debole, contiene un riferimento ad un oggetto HTTP per utilizzare il relativo servizio;
- **MaaP::Client::ModelClient::Model::SessionData**  
Relazione uscente debole, contiene un riferimento ad un oggetto SessionData per utilizzare i dati di sessione;
- **MaaP::Client::View::Template::SignIn**  
Relazione entrante, interazione con il template;
- **MaaP::Client::View::Template::SignUp**  
Relazione entrante, interazione con il template.

**4.4.3.4.3.2 ControllerCollection****Nome**

MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection

**Descrizione**

Classe che rappresenta il controller per indirizzare le richieste di visualizzazione di una pagina Collection.

**Utilizzo**

Viene utilizzata per indirizzare le richieste di visualizzazione di una pagina Collection.

**Relazioni con altre classi**

- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione uscente debole, contiene un riferimento ad un oggetto HTTP per utilizzare il relativo servizio;
- **MaaP::Client::ControllerModelView::Scope::Collection**  
Relazione uscente debole, contiene un riferimento ad un oggetto Collection per accedere allo scope relativo ai dati di una Collection;
- **MaaP::Client::ControllerModelView::Scope::Query**  
Relazione uscente debole, contiene un riferimento ad un oggetto Query per accedere allo scope relativo ai dati relativi alle query;
- **MaaP::Client::View::Template::AdminMainPageCollection**  
Relazione entrante, interazione con il template;
- **MaaP::Client::View::Template::UserMainPageCollection**  
Relazione entrante, interazione con il template.

**4.4.3.4.3.3 ControllerDocument****Nome**

MaaP::Client::ControllerModelView::ControllerClient::ControllerDocument

**Descrizione**

Classe che rappresenta il controller per indirizzare le richieste di visualizzazione di una pagina Document.

**Utilizzo**

Viene utilizzata per indirizzare le richieste di visualizzazione di una pagina Document.

**Relazioni con altre classi**

- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione uscente debole, contiene un riferimento ad un oggetto HTTP per utilizzare il relativo servizio;
- **MaaP::Client::ControllerModelView::Scope::Document**  
Relazione uscente debole, contiene un riferimento ad un oggetto Document per accedere allo scope relativo ai dati di un Document;
- **MaaP::Client::View::Template::MainPageDocument**  
Relazione entrante, interazione con il template;
- **MaaP::Client::View::Template::MainPageDocumentEdit**  
Relazione entrante, interazione con il template.

**4.4.3.4.3.4 ControllerProfilo****Nome**

MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo

**Descrizione**

Classe che rappresenta il controller per indirizzare le richieste di visualizzazione di una pagina profilo utente.

**Utilizzo**

Viene utilizzata per indirizzare le richieste di visualizzazione di una pagina profilo utente.

**Relazioni con altre classi**

- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione uscente debole, contiene un riferimento ad un oggetto HTTP per utilizzare il relativo servizio;
- **MaaP::Client::ControllerModelView::Scope::Profilo**  
Relazione uscente debole, contiene un riferimento ad un oggetto Profilo per accedere allo scope relativo ai dati del profilo;
- **MaaP::Client::View::Template::UserProfileEdit**  
Relazione entrante, interazione con il template;
- **MaaP::Client::View::Template::UserProfile**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::AdminProfile**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::PasswordRecovery**  
Relazione entrante, interazione con il template.

#### 4.4.3.4.3.5 ControllerMenu

**Nome**

MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu

**Descrizione**

Classe che rappresenta il controller per indirizzare le richieste di visualizzazione della parte di pagina relativa al menù.

**Utilizzo**

Viene utilizzata per indirizzare le richieste di visualizzazione della parte di pagina relativa al menù.

**Relazioni con altre classi**

- **MaaP::Client::ModelClient::Services::HTTP**  
Relazione uscente debole, contiene un riferimento ad un oggetto HTTP per utilizzare il relativo servizio;
- **MaaP::Client::ControllerModelView::Scope::Menu**  
Relazione uscente debole, contiene un riferimento ad un oggetto Menu per accedere allo scope relativo ai dati del menù;
- **MaaP::Client::View::Template::AdminMainPageCollection**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::UserMainPageCollection**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::MainPageDocument**  
Relazione entrante, interazione con il template;
- **MaaP::Client::View::Template::MainPageDocumentEdit**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::UserProfileEdit**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::UserProfile**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::AdminProfile**  
Relazione entrante, interazione con il template.
- **MaaP::Client::View::Template::PasswordRecovery**  
Relazione entrante, interazione con il template.

#### 4.4.3.5 MaaP::Client::ControllerModelView::Scope

##### 4.4.3.5.1 Informazioni sul package

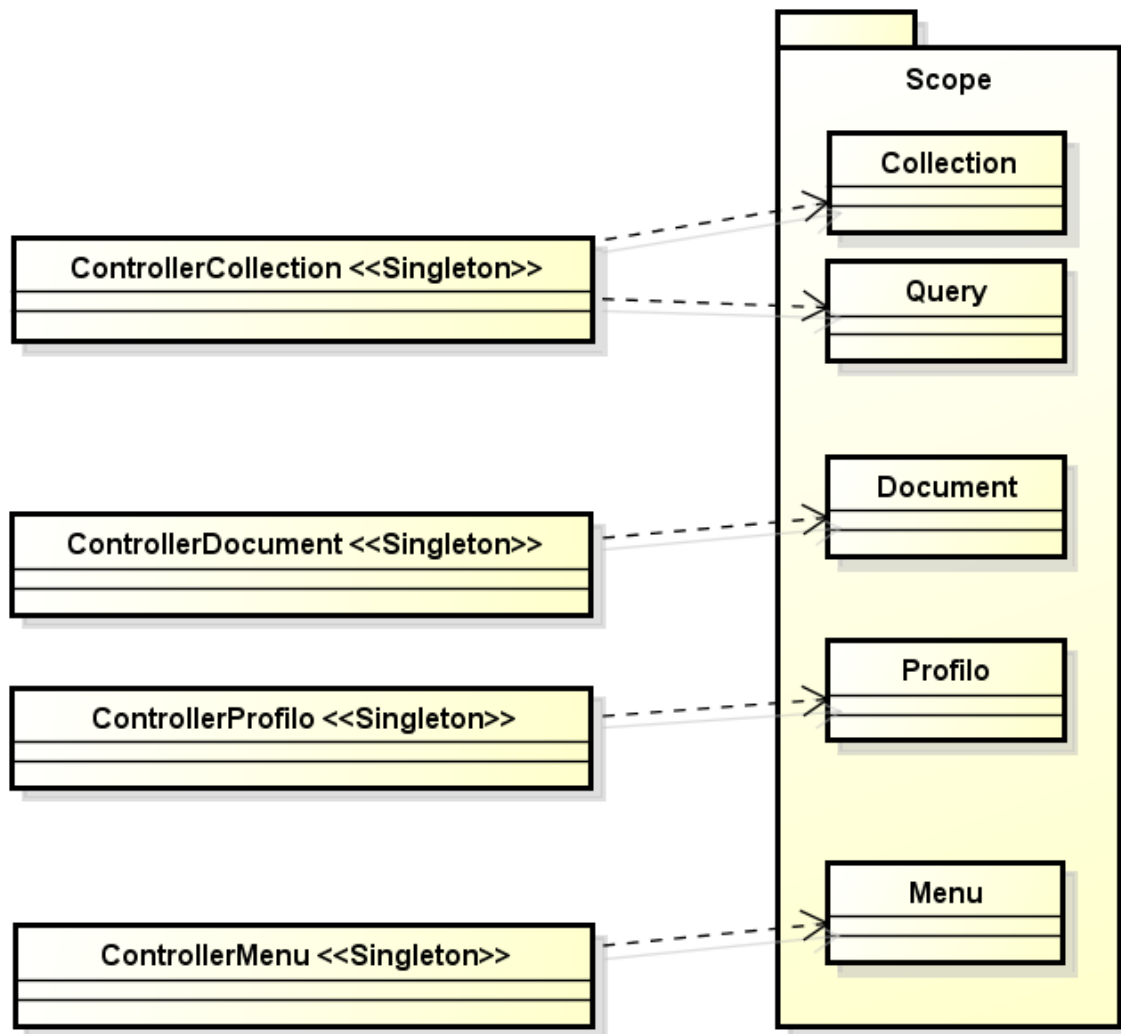


Figura 21: Componente MaaP::Client::ControllerModelView::Scope

##### 4.4.3.5.2 Descrizione

Componente parte del ControllerModelView contenente i dati per renderizzare i template.



#### 4.4.3.5.3 Classi

##### 4.4.3.5.3.1 Collection

**Nome**

MaaP::Client::ControllerModelView::Scope::Collection

**Descrizione**

Classe che rappresenta i dati relativi alla Collection da visualizzare.

**Utilizzo**

Viene utilizzata per memorizzare i dati relativi alla Collection da visualizzare i quali saranno successivamente visualizzati nella pagina web.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection**  
Relazione entrante debole, interazione con il controller della Collection;

##### 4.4.3.5.3.2 Query

**Nome**

MaaP::Client::ControllerModelView::Scope::Query

**Descrizione**

Classe che rappresenta i dati relativi alle query più utilizzare.

**Utilizzo**

Viene utilizzata per memorizzare i dati relativi alle query più utilizzate, le quali saranno successivamente visualizzate nella pagina web.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection**  
Relazione entrante debole, interazione con il controller della Collection;

##### 4.4.3.5.3.3 Document

**Nome**

MaaP::Client::ControllerModelView::Scope::Document

**Descrizione**

Classe che rappresenta i dati relativi al Document da visualizzare.

**Utilizzo**

Viene utilizzata per memorizzare i dati relativi al Document da visualizzare i quali saranno successivamente visualizzati nella pagina web.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerDocument**  
Relazione entrante debole, interazione con il controller del Document;

##### 4.4.3.5.3.4 Profilo

**Nome**

MaaP::Client::ControllerModelView::Scope::Profilo

**Descrizione**

Classe che rappresenta i dati relativi al profilo utente da visualizzare.

**Utilizzo**

Viene utilizzata per memorizzare i dati relativi al profilo utente da visualizzare i quali saranno

successivamente visualizzati nella pagina web.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**

Relazione entrante debole, interazione con il controller del profilo;

**4.4.3.5.3.5 Menu****Nome**

MaaP::Client::ControllerModelView::Scope::Menu

**Descrizione**

Classe che rappresenta i dati relativi al menù da visualizzare.

**Utilizzo**

Viene utilizzata per memorizzare i dati relativi al menù da visualizzare i quali saranno successivamente visualizzati nella pagina web.

**Relazioni con altre classi**

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**

Relazione entrante debole, interazione con il controller del menù;

#### 4.4.4 MaaP::Client::ModelClient

##### 4.4.4.1 Informazioni sul package

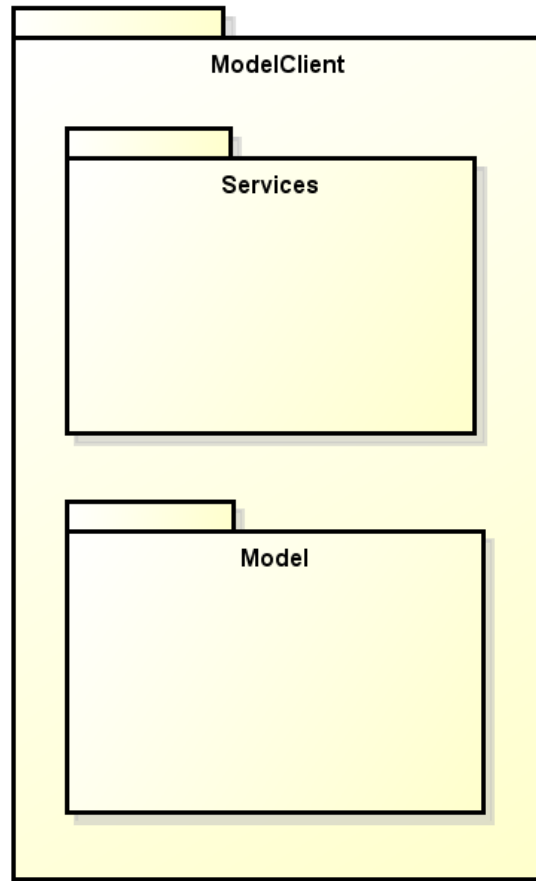


Figura 22: Componente MaaP::Client::ModelClient

##### 4.4.4.2 Descrizione

Componente Model del design pattern MVVM.

##### 4.4.4.3 Sotto-componenti

- MaaP::Client::ModelClient::Services;
- MaaP::Client::ModelClient::Model.

#### 4.4.4.4 MaaP::Client::ModelClient::Services

##### 4.4.4.4.1 Informazioni sul package

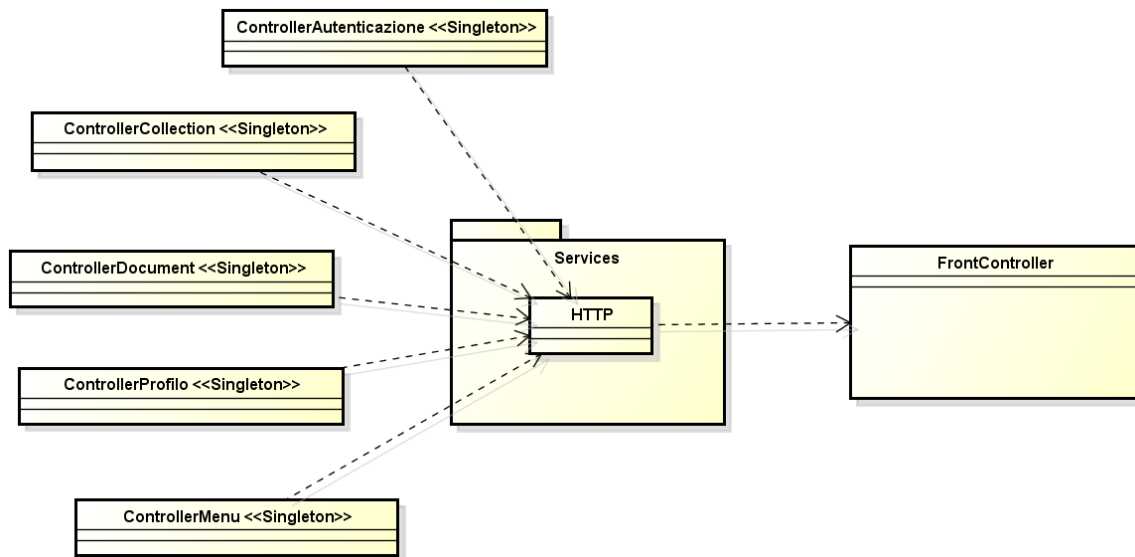


Figura 23: Componente MaaP::Client::ModelClient::Services

##### 4.4.4.4.2 Descrizione

Componente parte del ModelClient contenente i servizi per la comunicazione con il server.

##### 4.4.4.4.3 Classi

###### 4.4.4.4.3.1 HTTP

###### Nome

MaaP::Client::ModelClient::Services::HTTP

###### Descrizione

Classe che rappresenta il servizio di comunicazione HTTP con il server.

###### Utilizzo

Viene utilizzata per inviare richieste HTTP al server.

###### Relazioni con altre classi

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerAutenticazione**  
Relazione entrante debole, interazione con il controller dell'Autenticazione;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerCollection**  
Relazione entrante debole, interazione con il controller della Collection;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerDocument**  
Relazione entrante debole, interazione con il controller del Document;

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerProfilo**  
Relazione entrante debole, interazione con il controller del profilo;
- **MaaP::Client::ControllerModelView::ControllerClient::ControllerMenu**  
Relazione entrante debole, interazione con il controller del menù;
- **MaaP::Controller::FrontController**  
Relazione uscente debole, contiene un riferimento ad un oggetto di tipo FrontController per inviare richieste HTTP al server.

#### 4.4.4.5 MaaP::Client::ModelClient::Model

##### 4.4.4.5.1 Informazioni sul package

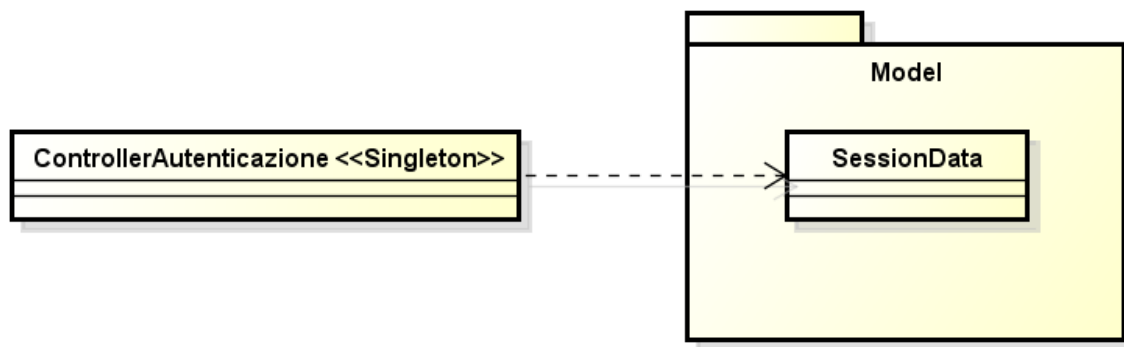


Figura 24: Componente MaaP::Client::ModelClient::Model

##### 4.4.4.5.2 Descrizione

Componente parte del ModelClient contenente i dati di sessione.

##### 4.4.4.5.3 Classi

###### 4.4.4.5.3.1 SessionData

###### Nome

MaaP::Client::ModelClient::Model::SessionData

###### Descrizione

Classe che rappresenta i dati di sessione utente.

###### Utilizzo

Viene utilizzata memorizzare i dati di sessione del client.

###### Relazioni con altre classi

- **MaaP::Client::ControllerModelView::ControllerClient::ControllerAutenticazione**  
Relazione entrante debole, interazione con il controller dell'Autenticazione.

## **5 Diagrammi di attività**

## 6 Design Pattern Utilizzati

### 6.1 Design Pattern architetturali

#### 6.1.1 MVVM

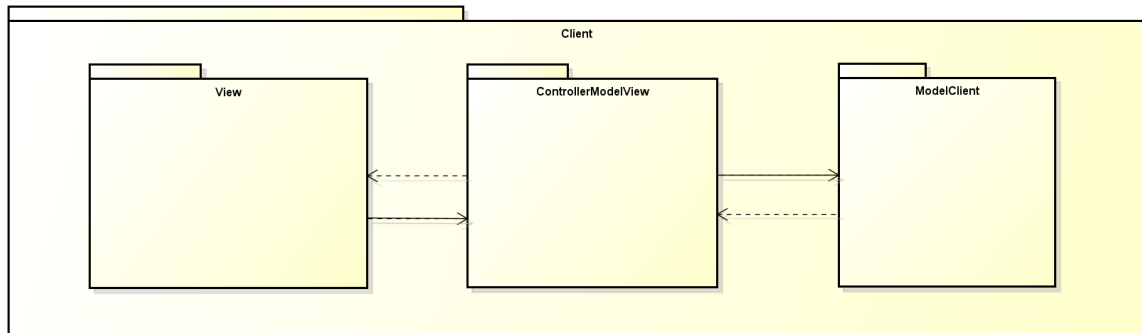


Figura 25: Applicazione di MVVM in MaaP

- **Scopo:** Viene usato il pattern Model-View-ViewModel per... ;
- **Utilizzo:** .

### 6.1.2 Three-tier

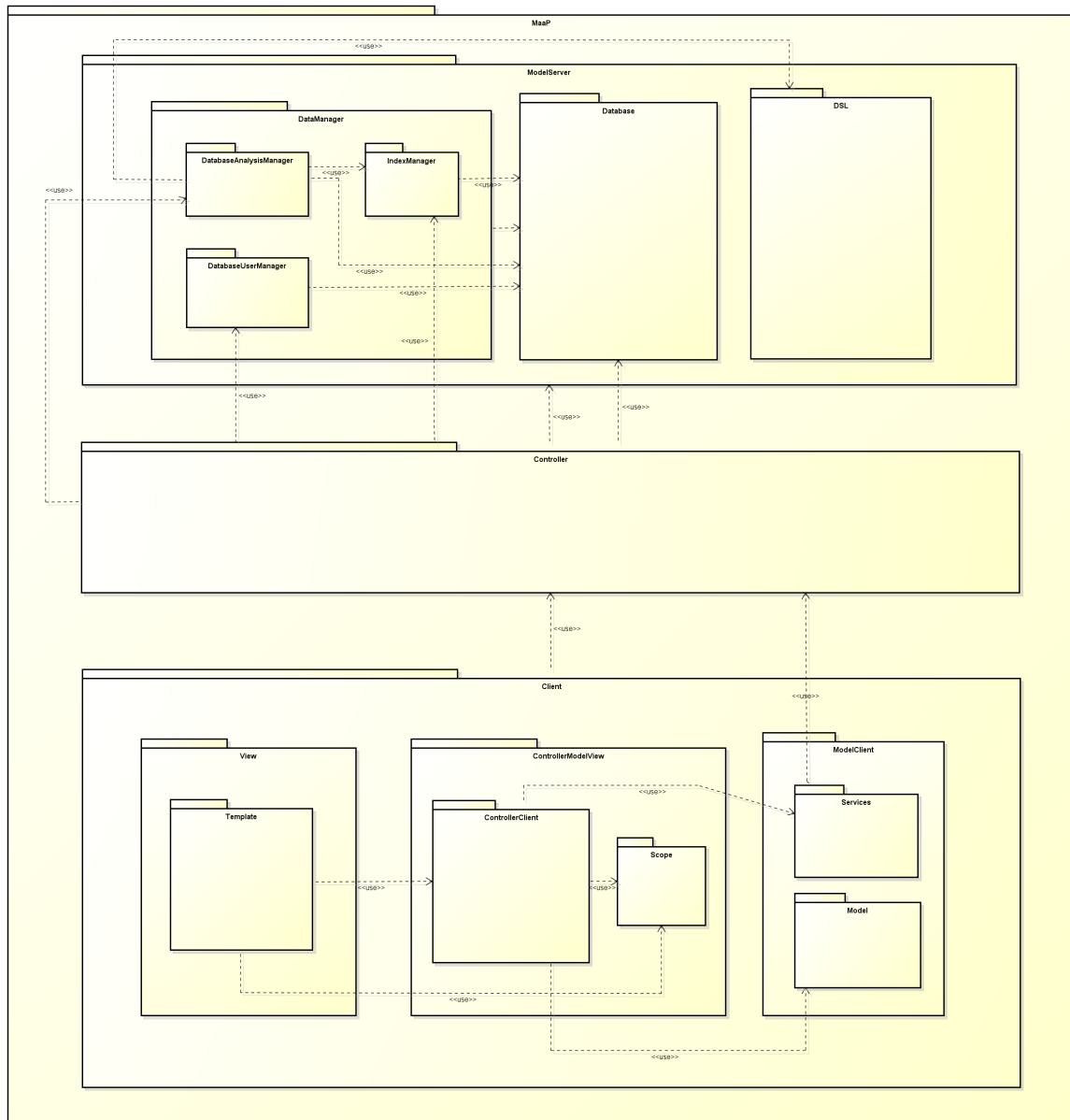


Figura 26: Applicazione di Three-tier in MaaP

- **Scopo:** Viene usato il pattern Three-tier per... ;
- **Utilizzo:** .



## 6.2 Design Pattern creazionali

### 6.2.1 Singleton

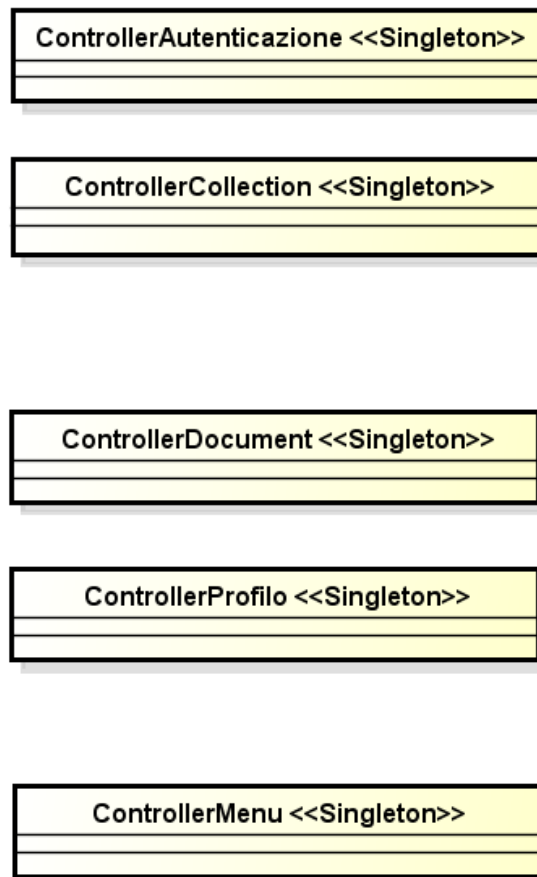


Figura 27: Applicazione di Singleton in MaaP

- **Scopo:** Viene usato il pattern Singleton per le classi che devono avere un'unica istanza durante l'esecuzione dell'applicazione;
- **Utilizzo:** Le classi che devono avere un'unica istanza sono i controller lato client.

## 6.3 Design Pattern comportamentali

### 6.3.1 Strategy

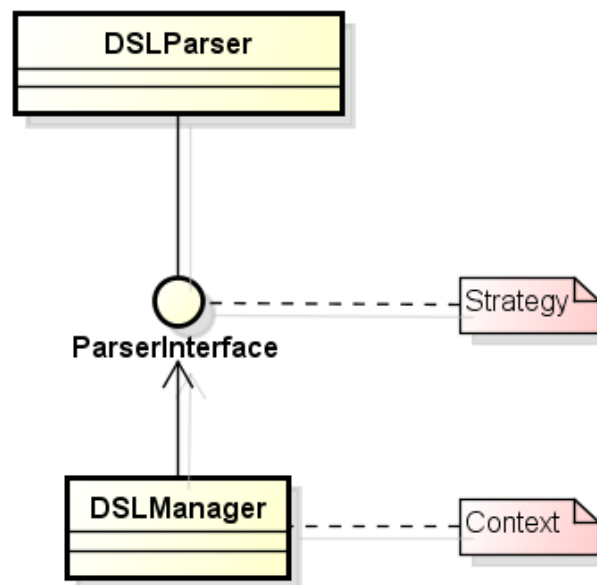


Figura 28: Applicazione di Strategy in MaaP

- **Scopo:** Il pattern Strategy viene usato per isolare più algoritmi che svolgono la stessa funzione dal codice che esegue la funzione;
- **Utilizzo:** In MaaP è stato usato gestire inizialmente un singolo algoritmo di parsing del file di descrizione, ma permetterà in futuro di aggiungere nuovi algoritmi di parsing differenziati senza modificare le classi che ne fanno uso.  
La concrete strategy attualmente presente è: **DSLParser**.

## 6.4 Design Pattern strutturali

### 6.4.1 Adapter

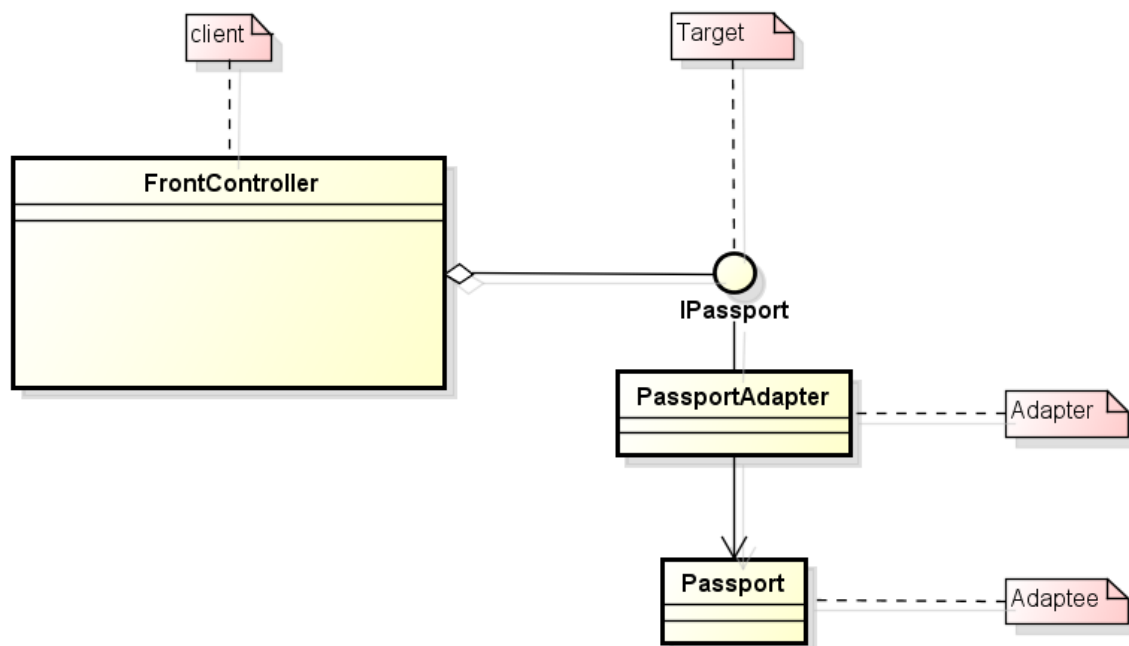


Figura 29: Applicazione di Adapter in MaaP

- **Scopo:** Il pattern Adapter viene utilizzato per adattare una classe riutilizzando un oggetto già esistente. Questo semplifica l'eventuale processo di sostituzione dell'oggetto esistente, creando un'interfaccia stabile per il resto dell'applicazione;
- **Utilizzo:** In MaaP è stato usato per adattare la classe Passport nel Controller. PassportAdapter adatta Passport.

#### 6.4.2 Facade

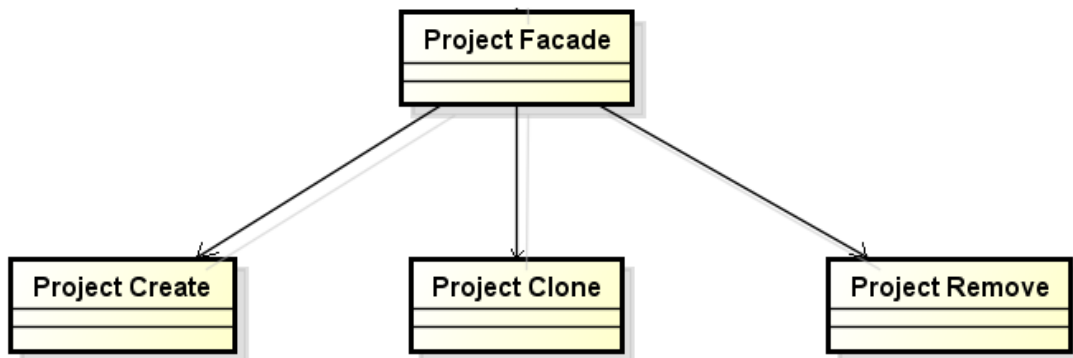


Figura 30: Applicazione di Facade in MaaP

- **Scopo:** Il pattern Facade viene usato per fornire un'interfaccia unica a più classi;
- **Utilizzo:** In MaaP, ProjectFacade è una Facade che presenta un'interfaccia per tutti gli oggetti gestiscono la creazione e/o modifica di un progetto:
  - ProjectCreate;
  - ProjectClone;
  - ProjectRemove.

## 7 Stime di fattibilità e di bisogno di risorse

## 8 Tracciamento

### 8.1 Tracciamento componenti - requisiti

### 8.2 Tracciamento requisiti - componenti

## A Descrizione Design Pattern

### A.1 Design Pattern architetturali

#### A.1.1 MVVM

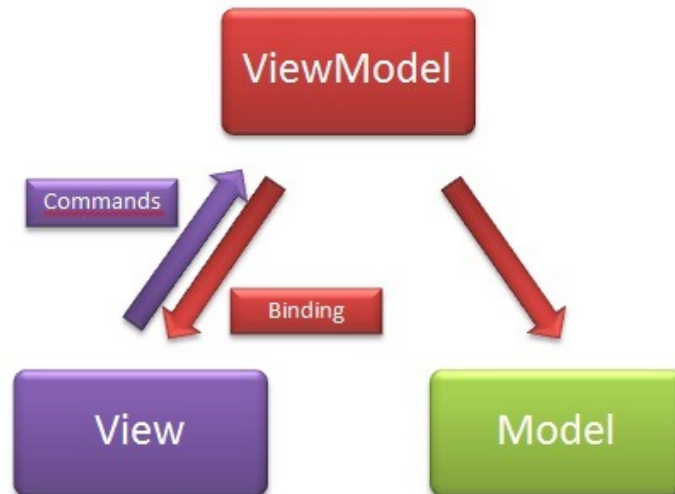


Figura 31: Diagramma del design pattern MVVM

- **Scopo:**
- **Motivazione:**
- **Applicabilità:**

### A.1.2 Three-Tier

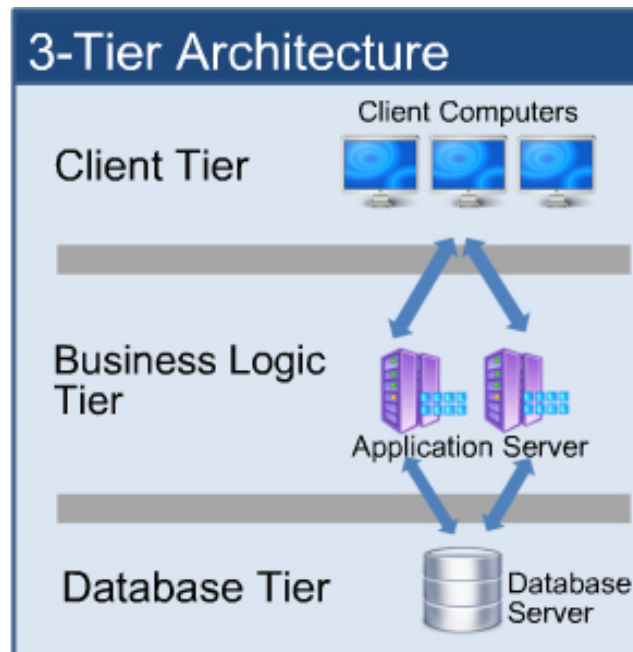


Figura 32: Diagramma del design pattern Three-tier

- **Scopo:**
- **Motivazione:**
- **Applicabilità:**



## A.2 Design Pattern creazionali

### A.2.1 Singleton



Figura 33: Diagramma del design pattern Singleton

- Scopo:
- Motivazione:
- Applicabilità:

## A.3 Design Pattern strutturali

### A.3.1 Adapter

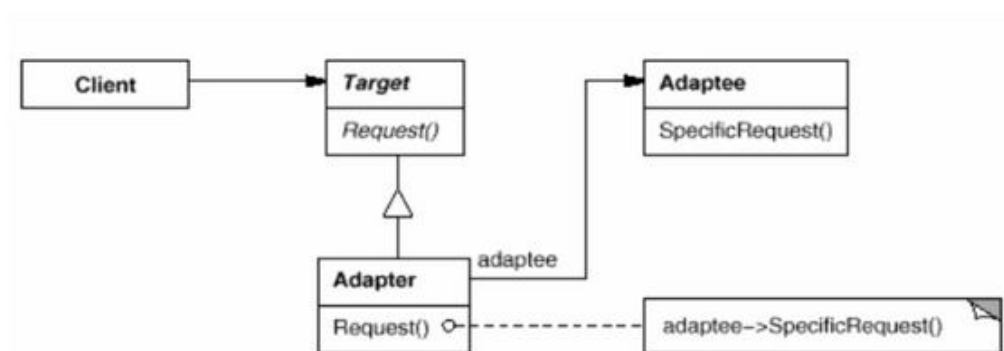


Figura 34: Diagramma del design pattern Adapter

- Scopo:

- Motivazione:
- Applicabilità:

### A.3.2 Facade

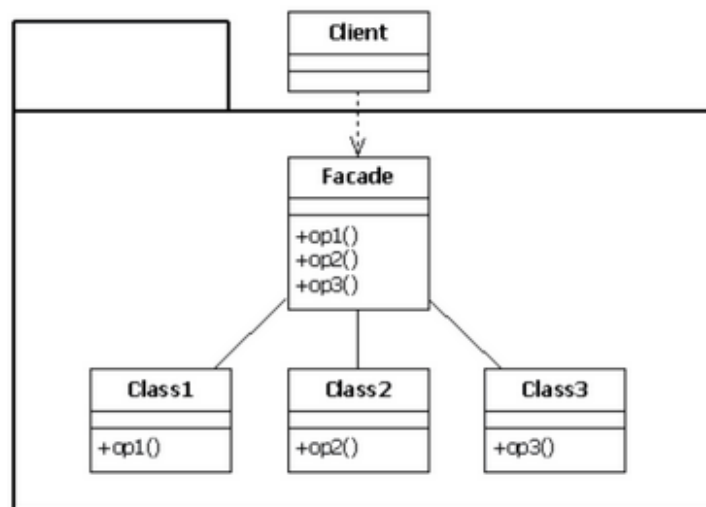


Figura 35: Diagramma del design pattern Facade

- Scopo:
- Motivazione:
- Applicabilità:

## A.4 Design Pattern comportamentali

### A.4.1 Strategy

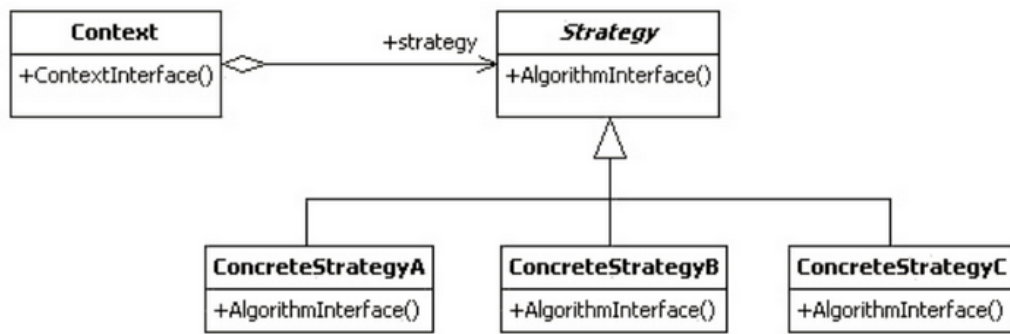


Figura 36: Diagramma del design pattern Strategy

- **Scopo:**
- **Motivazione:**
- **Applicabilità:**