

# KALEIDOSCODE

## SWEDesigner

### SOFTWARE PER DIAGRAMMI UML

DEFINIZIONE DI PRODOTTO V1.0.0



#### Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Data Redazione</b>	15/06/2017
<b>Redazione</b>	Bonato Enrico Pace Giulio Pezzuto Francesco Sovilla Matteo
<b>Verifica</b>	Sanna Giovanni
<b>Approvazione</b>	Bonolo Marco
<b>Uso</b>	Esterno
<b>Distribuzione</b>	<i>Prof. Vardanega Tullio</i> <i>Prof. Cardin Riccardo</i> <i>Zucchetti s.p.a.</i>

## Diario delle Modifiche

Versione	Data	Autore	Descrizione
1.0.0	04/07/2017	Bonolo Marco	Approvazione documento
0.2.0	03/07/2017	Sanna Giovanni	Verifica documento
0.1.3	01/07/2017	Pace Giulio	Modifiche e correzioni in tracciamenti
0.1.1	28/06/2017	Pezzuto Francesco	Modifiche e correzioni in Specifica delle componenti
0.1.0	17/06/2017	Sanna Giovanni	Verifica documento
0.0.6	15/06/2017	Sovilla Matteo	Stesura sezione Tracciamenti
0.0.5	14/06/2017	Bonato Enrico	Stesura parziale sezione Specifica delle componenti
0.0.4	14/06/2017	Pace Giulio	Stesura parziale sezione Specifica delle componenti
0.0.3	13/06/2017	Pezzuto Francesco	Stesura parziale sezione Specifica delle componenti
0.0.2	15/05/2017	Pace Giulio	Stesura capitolo 2, Architettura dell'applicazione
0.0.1	01/05/2017	Pace Giulio	Creazione scheletro del documento e stesura della sezione Introduzione

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del prodotto . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti utili . . . . .	1
1.4.1	Riferimenti normativi . . . . .	1
1.4.2	Riferimenti informativi . . . . .	1
<b>2</b>	<b>Architettura dell'applicazione</b>	<b>3</b>
2.1	Architettura client . . . . .	3
2.1.1	Diagrammi editabili . . . . .	4
2.2	Architettura server . . . . .	4
2.2.1	Comunicazioni server-client . . . . .	5
<b>3</b>	<b>Specifica delle componenti</b>	<b>6</b>
3.1	SWEDesigner . . . . .	6
3.2	SWEDesigner::Client . . . . .	6
3.3	SWEDesigner::Client::Model . . . . .	6
3.3.1	SWEDesigner::Client::Model::DataManager . . . . .	7
3.3.2	SWEDesigner::Client::Model::ProjectModel . . . . .	8
3.3.3	SWEDesigner::Client::Model::ToolBarModel . . . . .	10
3.3.4	SWEDesigner::Client::Model::Project . . . . .	11
3.3.5	SWEDesigner::Client::Model::RequestHandler . . . . .	12
3.4	SWEDesigner::Client::Model::Items . . . . .	13
3.4.1	SWEDesigner::Client::Model::Items::Swedesigner . . . . .	13
3.4.2	Swedesigner.model.packageDiagram.items.Base . . . . .	13
3.4.3	Swedesigner.model.packageDiagram.items.BaseView . . . . .	14
3.4.4	Swedesigner.model.packageDiagram.items.Package . . . . .	14
3.4.5	Swedesigner.model.packageDiagram.items.PkgComment . . . . .	15
3.4.6	Swedesigner.model.packageDiagram.items.PkgCommentView . . . . .	16
3.4.7	Swedesigner.model.packageDiagram.items.packageDiagramLink . . . . .	16
3.4.8	Swedesigner.model.packageDiagram.items.PkgCommentLink . . . . .	17
3.4.9	Swedesigner.model.packageDiagram.items.PkgDependency . . . . .	17
3.4.10	Swedesigner.model.classDiagram.items.Base . . . . .	17
3.4.11	Swedesigner.model.classDiagram.items.BaseView . . . . .	18
3.4.12	Swedesigner.model.classDiagram.items.Class . . . . .	18
3.4.13	Swedesigner.model.classDiagram.items.Interface . . . . .	19
3.4.14	Swedesigner.model.classDiagram.items.ClComment . . . . .	20
3.4.15	Swedesigner.model.classDiagram.items.ClCommentView . . . . .	21
3.4.16	Swedesigner.model.classDiagram.items.classDiagramLink . . . . .	22
3.4.17	Swedesigner.model.classDiagram.items.ClCommentLink . . . . .	22
3.4.18	Swedesigner.model.classDiagram.items.Generalization . . . . .	22
3.4.19	Swedesigner.model.classDiagram.items.Implementation . . . . .	23
3.4.20	Swedesigner.model.classDiagram.items.Aggregation . . . . .	23
3.4.21	Swedesigner.model.classDiagram.items.Composition . . . . .	23

3.4.22	Swedesigner.model.classDiagram.items.Association . . . . .	24
3.4.23	Swedesigner.model.bubbleDiagram.items.Base . . . . .	24
3.4.24	Swedesigner.model.bubbleDiagram.items.BaseView . . . . .	25
3.4.25	Swedesigner.model.bubbleDiagram.items.customBubble . . . . .	25
3.4.26	Swedesigner.model.bubbleDiagram.items.bubbleIf . . . . .	26
3.4.27	Swedesigner.model.bubbleDiagram.items.bubbleElse . . . . .	27
3.4.28	Swedesigner.model.bubbleDiagram.items.bubbleFor . . . . .	27
3.4.29	Swedesigner.model.bubbleDiagram.items.bubbleReturn . . . . .	28
3.4.30	Swedesigner.model.bubbleDiagram.items.bubbleStart . . . . .	29
3.4.31	Swedesigner.model.bubbleDiagram.items.bubbleWhile . . . . .	29
3.4.32	Swedesigner.model.bubbleDiagram.items.bubbleDiagramLink . . . .	30
3.4.33	Swedesigner.model.bubbleDiagram.items.bubbleLink . . . . .	30
3.5	SWEDesigner::Client::View . . . . .	31
3.5.1	SWEDesigner::Client::View::ProjectView . . . . .	31
3.5.2	SWEDesigner::Client::View::TitlebarView . . . . .	34
3.5.3	SWEDesigner::Client::View::ToolBarView . . . . .	35
3.5.4	SWEDesigner::Client::View::PathView . . . . .	36
3.5.5	SWEDesigner::Client::View::EditPanelView . . . . .	36
3.6	SWEDesigner::Server . . . . .	38
3.7	SWEDesigner::Server::CodeGenerator . . . . .	38
3.7.1	SWEDesigner::Server::CodeGenerator::CodeGenerator . . . . .	39
3.8	SWEDesigner::Server::CodeGenerator::Parser . . . . .	40
3.8.1	SWEDesigner::Server::CodeGenerator::Parser::Parser . . . . .	40
3.9	SWEDesigner::Server::CodeGenerator::Coder . . . . .	41
3.9.1	SWEDesigner::Server::CodeGenerator::Coder::JavaCoder . . . . .	41
3.9.2	SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder . . .	42
3.9.3	SWEDesigner::Server::CodeGenerator::Coder::CoderOperation . . .	43
3.9.4	SWEDesigner::Server::CodeGenerator::Coder::CoderParameter . . .	44
3.9.5	SWEDesigner::Server::CodeGenerator::Coder::CoderAttribute . . .	45
3.9.6	SWEDesigner::Server::CodeGenerator::Coder::CoderActivity . . . .	45
3.9.7	SWEDesigner::Server::CodeGenerator::Coder::CodedProg . . . . .	48
3.10	SWEDesigner::Server::CodeGenerator::Builder . . . . .	48
3.10.1	SWEDesigner::Server::CodeGenerator::Builder::Builder . . . . .	48
3.11	SWEDesigner::Server::CodeGenerator::Zipper . . . . .	50
3.11.1	SWEDesigner::Server::CodeGenerator::Zipper::Zipper . . . . .	50
3.12	SWEDesigner::Server::DataManager . . . . .	50
3.12.1	SWEDesigner::Server::DataManager::DataManager . . . . .	50
3.13	SWEDesigner::Server::RequestHandler . . . . .	52
3.14	SWEDesigner::Server::RequestHandler . . . . .	52
<b>4</b>	<b>Tracciamenti</b>	<b>53</b>
4.1	Tracciamento classi - requisiti . . . . .	53
4.2	Tracciamento requisiti - classi . . . . .	64

## Elenco delle tabelle

2	Tracciamento Classi-Requisiti . . . . .	63
3	Tracciamento Requisiti-Classi . . . . .	74

---

## Elenco delle figure

1	Architettura del client . . . . .	3
2	Architettura del server . . . . .	5
3	Esempi delle possibili comunicazioni client-server . . . . .	5
4	Architettura del client . . . . .	6
5	Architettura di Model . . . . .	8
6	Architettura di View . . . . .	31
7	Architettura del server . . . . .	38
8	Architettura di Coder . . . . .	41

# 1 Introduzione

## 1.1 Scopo del documento

Con il presente documento si intende definire la progettazione in dettaglio della struttura e del funzionamento delle componenti del progetto *SWEDesigner*.

Verrà presentato innanzi tutto l'architettura secondo la quale verranno organizzate le componenti software. Successivamente verranno specificate le componenti nel dettaglio per la parte client e server. Infine verranno mostrati i diagrammi di sequenza delle principali azioni lato back-end al fine di facilitare la comprensione del funzionamento del programma. Il documento ha la funzione di servire da guida ai *Programimatori* del gruppo.

## 1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un software di costruzione di diagrammi UML<sub>G</sub> con la relativa generazione di codice Java<sub>G</sub> e Javascript<sub>G</sub> utilizzando tecnologie web. Il prodotto deve essere conforme ai vincoli qualitativi richiesti dal committente.

## 1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite sono riportate nel documento *Glossario v2.0.0*.

La prima occorrenza di ciascuno di questi vocaboli è marcata da una "G" maiuscola in pedice.

## 1.4 Riferimenti utili

### 1.4.1 Riferimenti normativi

- Capitolato<sub>G</sub> d'appalto:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf> (09/03/2017);
- Norme di progetto: *Norme di progetto v3.0.0*;
- Analisi dei requisiti: *Analisi dei requisiti v3.0.0*;
- Specifica tecnica: *Specifica tecnica v2.0.0*;
- Verbali esterni:
  - Verbale incontro con *Zucchetti s.p.a.* in data 05/05/2017.

### 1.4.2 Riferimenti informativi

- Slide dell'insegnamento di Ingegneria del Software 1° semestre:
  - Design pattern strutturali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E04.pdf> (02/05/2017);

- Design pattern creazionali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E05.pdf> (02/05/2017);
- Design pattern comportamentali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E06.pdf> (02/05/2017);
- Design pattern architetturali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E07.pdf> (02/05/2017),  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E08.pdf> (02/05/2017);
- Stili architetturali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E09.pdf> (02/05/2017);
- **Design Patterns: Elements of reusable object-oriented software**  
E. Gamma, R. Helm, R. Johnson, J. Vlissides - 1st Edition (2002)
  - Capitolo 3: Creational patterns;
  - Capitolo 4: Structural patterns;
  - Capitolo 5: Behavioral patterns.
- **Jointjs**: <https://www.jointjs.com/opensource> - 02/05/2017
- **jQuery**: <https://jquery.com/> - 02/05/2017
- **Lodash**: <https://lodash.com/> - 02/05/2017
- **Backbone.js**: <http://backbonejs.org/> - 02/05/2017
- **Node.js**: <https://nodejs.org/it/> - 02/05/2017
- **RequireJS**: <http://requirejs.org/> - 02/05/2017
- **MySQL**: <https://www.mysql.com/> - 02/05/2017



## 2 Architettura dell'applicazione

*SWEDesigner* è realizzato utilizzando un'architettura client-server, in particolare:

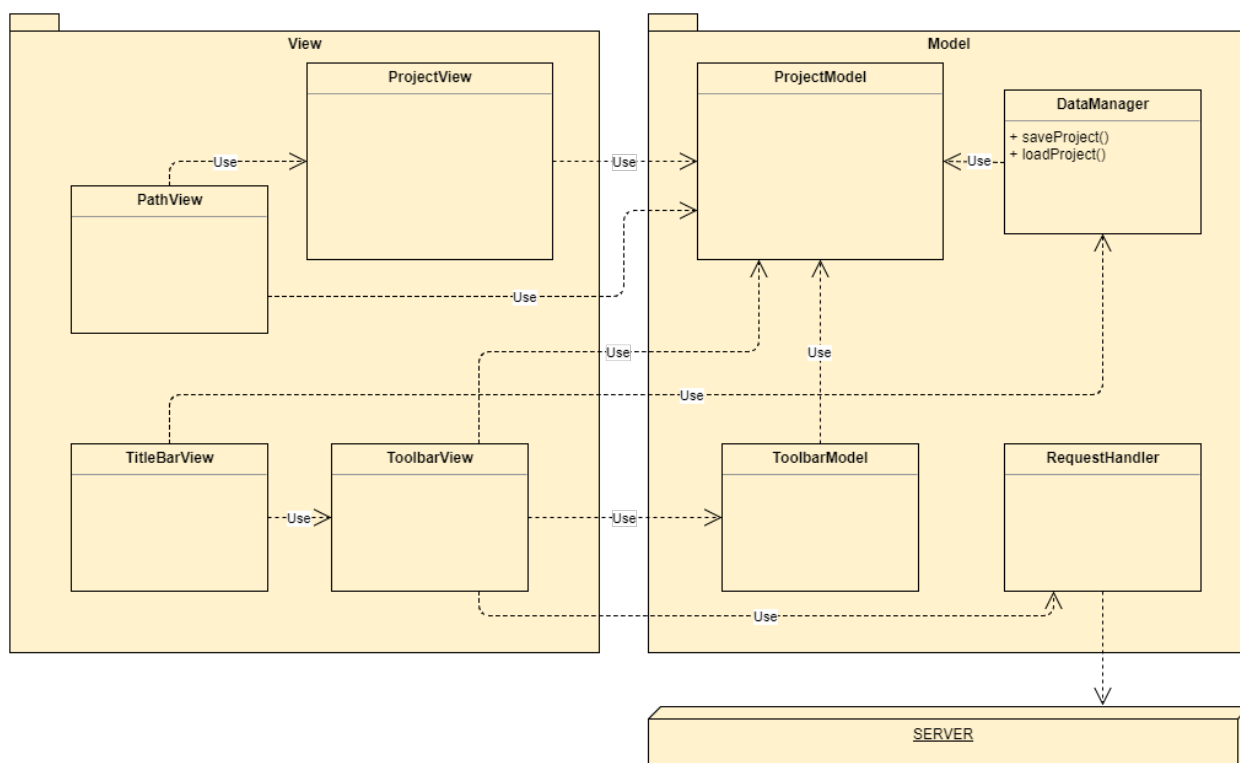
- Il **client** corrisponde alla parte dell'applicativo che funzionerà nel browser dell'utente;
- Il **server** avrà il compito di fornire la pagina dell'applicativo al client e ne gestirà le richieste ricevute riguardanti la generazione del codice sorgente o le attività "bubble" da inserire nell'editor.

### 2.1 Architettura client

Il client (parte front-end<sub>G</sub>) è una Single Page Application (SPA<sub>G</sub>) scritta con i linguaggi HTML5, CSS<sub>G</sub> e Javascript.

La sua architettura è costruita utilizzando il framework Backbone.js che offre un'architettura di tipo Model-View ed è quindi principalmente suddivisa nei seguenti moduli:

- **Model:** organizza la logica alla base dei diagrammi dell'editor.
- **View:** gestisce l'interfaccia grafica dell'editor e, seguendo la struttura definita da Backbone.js, "contiene" la componente controller per la gestione degli eventi;



**Figura 1:** Architettura del client

### 2.1.1 Diagrammi editabili

In ogni diagramma creabile all'interno dell'applicazione è offerto solamente un sottoinsieme del totale dei formalismi definiti dal linguaggio UML standard. Si possono individuare tre tipi di diagrammi:

- Diagramma dei package<sub>G</sub>;
- Diagramma delle classi;
- Diagramma delle bubble.

Il diagramma dei package è logicamente correlato con il diagramma delle classi. Per ogni elemento (package o classe) all'interno di questi diagrammi è possibile assegnare un livello di importanza attraverso il quale si può "filtrare" gli oggetti a schermo visualizzabili nell'editor.

Per la corretta generazione del codice, nei diagrammi delle attività è previsto che l'utente approfondisca il loro livello di astrazione fino ad arrivare ad un diagramma costituito solamente da bubble (diagramma delle bubble) che verranno fornite nell'editor come se fossero delle attività specifiche.

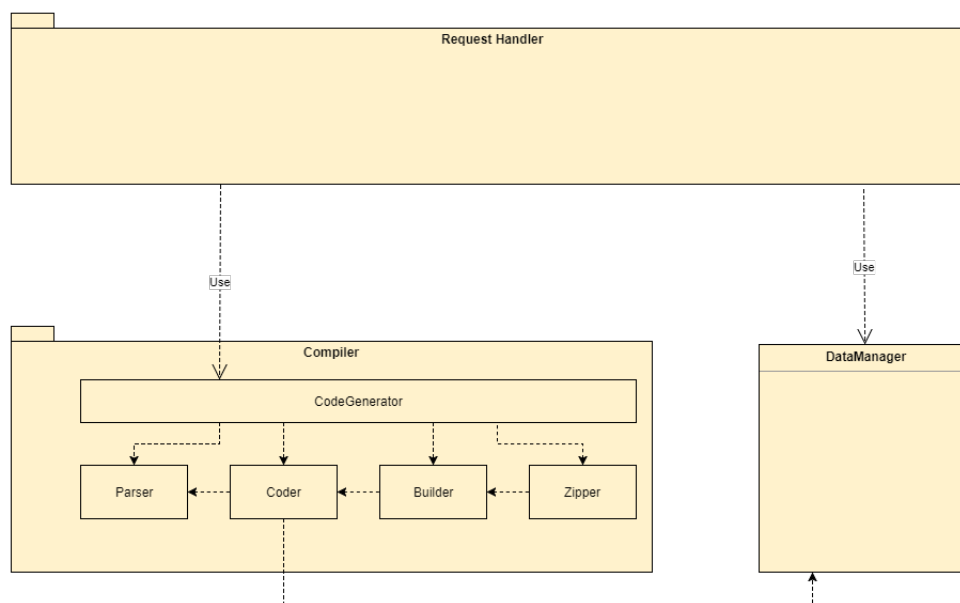
## 2.2 Architettura server

Il server (parte back-end<sub>G</sub>) è sviluppato in Node.js ed offre i seguenti servizi:

- Fornire la Single Page Application ai client che la richiedono;
- Fornire la lista di bubble utilizzabili nell'editor;
- Generare il codice sorgente, nel formato desiderato dal client, del progetto inviatogli.

In particolare, la componente che genera il codice sorgente è stata realizzata utilizzando un'architettura di tipo Pipe And Filter, in modo tale da assegnare un compito ben preciso ad ogni modulo per attuare una procedura sequenziale a "catena di montaggio". L'ultimo modulo ha il compito di creare un file compresso .zip del codice generato che sarà poi inviato al client.

Le bubble saranno salvate in una base dati per poter garantire una futura estendibilità del numero di queste ultime, eventualmente anche in altri domini da quello considerato al momento (i giochi da tavolo).



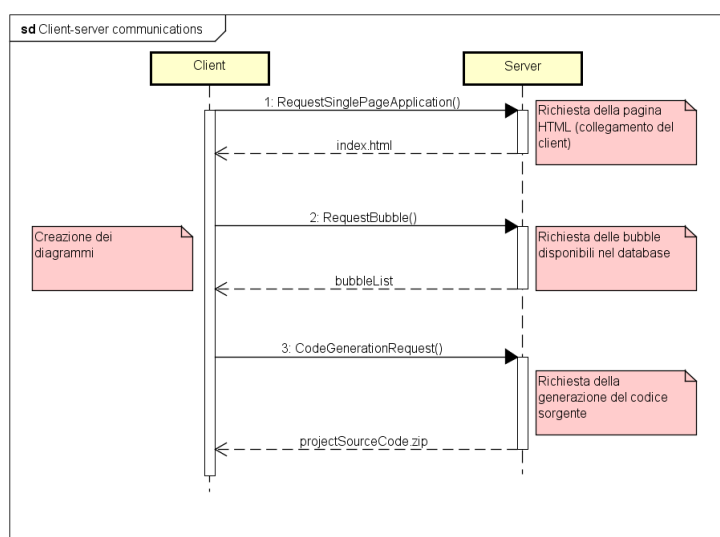
**Figura 2:** Architettura del server

### 2.2.1 Comunicazioni server-client

La Single Page Application viene fornita al client semplicemente attraverso una pagina HTML.

Per la richiesta e fornitura delle bubble, client e server utilizzano AJAX per lo scambio di dati in formato JSON in modo tale da alleggerire il traffico.

Per la richiesta della generazione del codice, il client invia i dati del progetto in formato JSON utilizzando AJAX ed il server una volta elaborata la richiesta procede con l'invviare il file .zip precedentemente descritto.



**Figura 3:** Esempi delle possibili comunicazioni client-server

## 3 Specifica delle componenti

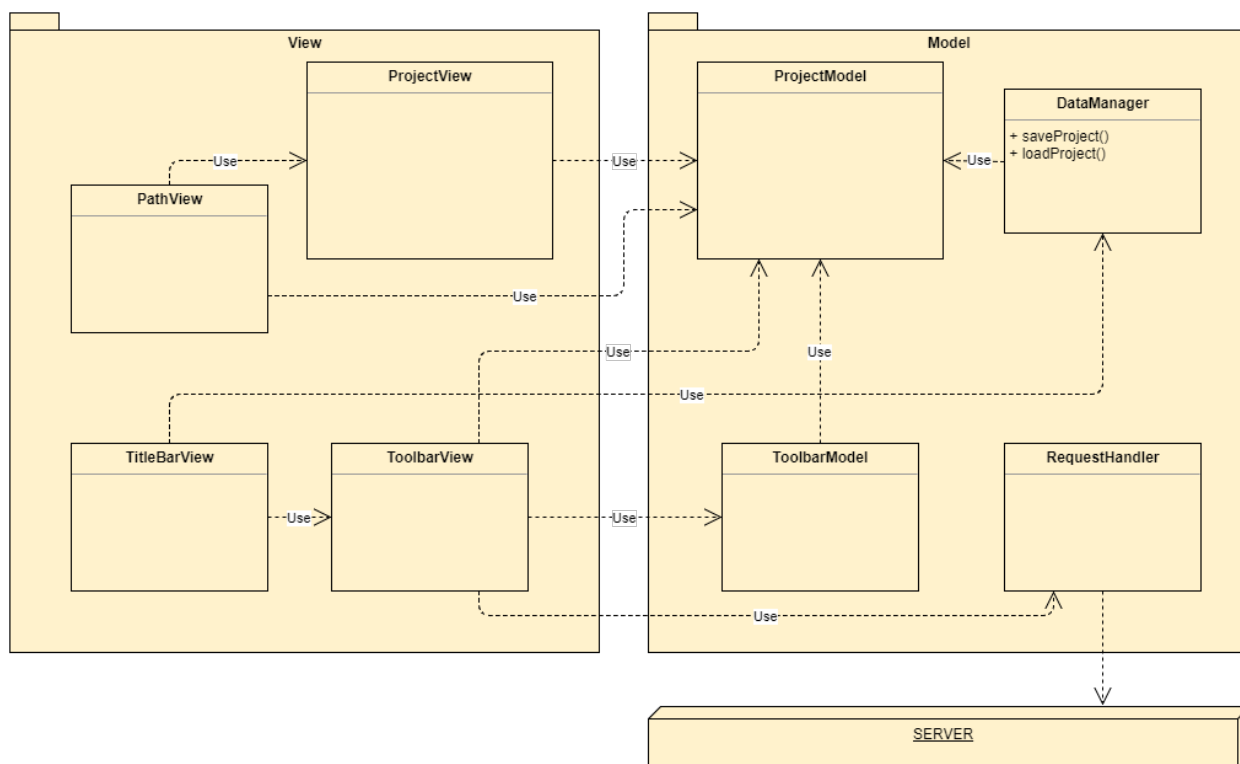
### 3.1 SWEDesigner

I package contenuti al suo interno sono:

- SWEDesigner::Client;
- SWEDesigner::Server.

Questo package non contiene delle classi.

### 3.2 SWEDesigner::Client



**Figura 4:** Architettura del client

I package contenuti al suo interno sono:

- SWEDesigner::Client::Model;
- SWEDesigner::Client::View.

Questo package non contiene delle classi.

### 3.3 SWEDesigner::Client::Model

I package contenuti al suo interno sono:

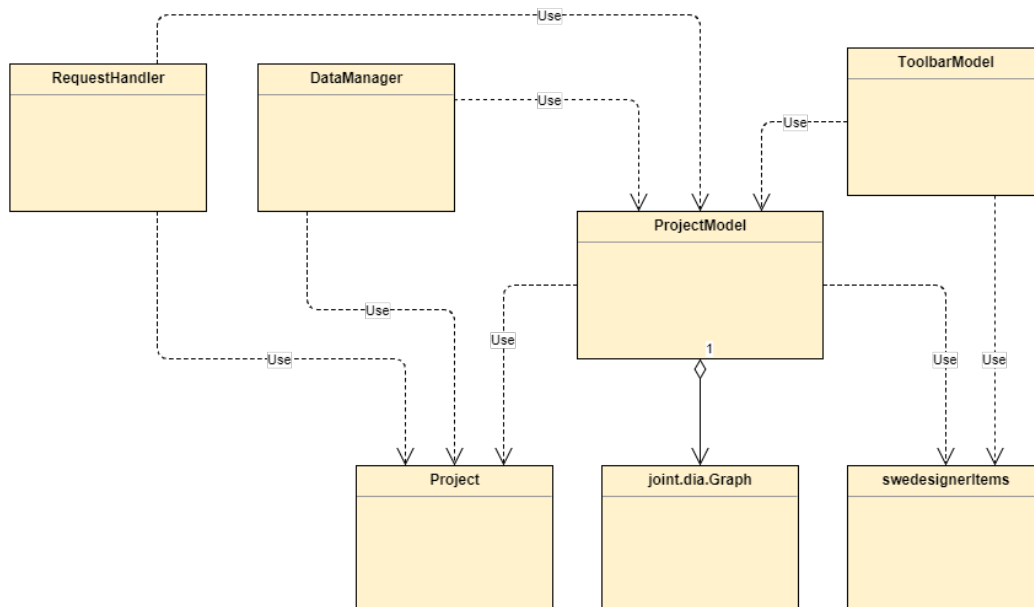
- SWEDesigner::Client::Model::Items.

Le classi contenute al suo interno verranno elencate qui di seguito.

### 3.3.1 SWEDesigner::Client::Model::DataManager

- **Tipo:** *Classe statica*;
- **Descrizione:** Si occupa della persistenza dei dati, in particolare del salvataggio su file system locale del progetto già esistente.  
;
- **Metodi:**
  - *newProject(): void*  
Dopo aver chiesto conferma all'utente, crea un nuovo progetto sovrascrivendo quello correntemente aperto;
  - *openProject(): void*  
Legge un file JSON e ne salva il contenuto in project e nel projectModel come progetto attualmente aperto;
  - *save(fileName: String): void*  
Salva i dati del progetto, li converte in formato JSON e avvia la procedura di download in locale del browser;  
**Parametri:**
    - \* *fileName: String* Nome del file generato da scaricare;
  - *saveAs(): void*  
Estrae la stringa inserita dall'utente nella schermata per il salvataggio con nome e invoca la il metodo per il salvataggio del progetto corrente in un file con il nome desiderato;
- **Relazioni con le altre classi:**
  - OUT *SWEDesigner::Client::Model::ProjectModel*: si occupa di gestire la parte logica dell'editor;
  - OUT *SWEDesigner::Client::Model::Project*: si occupa di gestire gli elementi contenuti nel diagramma.

### 3.3.2 SWEDesigner::Client::Model::ProjectModel



**Figura 5:** Architettura di Model

- **Tipo:** *Classe*;
- **Descrizione:** Model del progetto corrente. Si occupa di gestire il graph (`joint.dia.Graph`) e tutti gli eventi ad esso associati;
- **Padre:** *Backbone.model*;
- **Attributi:**
  - *currentDiagram: String*  
L'id del diagramma correntemente caricato nel graph (null se è il diagramma dei package);
  - *currentDiagramType: String*  
Il tipo del diagramma correntemente caricato nel graph ("packageDiagram", "classDiagram" o "bubbleDiagram");
  - *graph: joint.dia.Graph*  
Il model dell'area di disegno associata al paper della *SWEDesigner::Client::View::ProjectView*;
  - *itemToBeAdded: String*  
Salvataggio temporaneo dell'elemento da aggiungere al graph corrente;
- **Metodi:**

- *addItem(item: Object): void*

Salva in itemToBeAdded l'elemento passato in input che è un oggetto di Swe-designer::Client::Model::Items;

**Parametri:**

- \* *item: Object* Elemento del diagramma;

- *addItemToGraph(): void*

Aggiunge un elemento al grafo del diagramma corrente;

- *changedPosition(graph: joint.dia.Graph, cell: joint.dia.Cell, newPosition: Object, opt: Object): void*

Gestisce la traslazione di un elemento selezionato nel grafo;

**Parametri:**

- \* *graph: joint.dia.Graph* Grafo del diagramma corrente;
- \* *cell: joint.dia.Cell* Elemento correntemente selezionato;
- \* *newPosition: Object* Posizione attuale dell'oggetto nel grafo;
- \* *opt: Object* Traslazione dell'oggetto dalla posizione iniziale alla posizione "newPosition";

- *deleteCell(): void*

Rimuove un elemento dal grafo eliminando anche gli eventuali diagrammi derivati (classi o bubble);

- *deleteOperation(): void*

Rimuove un'operazione ed eventualmente anche il diagramma delle bubble associato;

- *getCellFromId(cellId: String): void*

Ritorna l'elemento del graph avente l'id passato come parametro in input;

**Parametri:**

- \* *cellId: String* Identificativo dell'elemento nel graph;

- *graphSwitched(): void*

Genera l'evento "switchgraph";

- *initialize(): void*

Inizializzazione del ProjectModel: inizializzazione del graph, del currentDiagramType, degli eventi verificabili;

- *resizeParent(parent: Object): void*

Esegue il resize di un elemento del diagramma ingrandendolo;

**Parametri:**

- \* *parent: Object* Elemento del diagramma;

- *saveCurrentDiagram(): void*

Salva il diagramma correntemente aperto all'interno della struttura definita nella classe Project;

- *switchInGraph(): void*

Esegue lo switch in profondità al diagramma selezionato svuotando il graph dagli elementi correntemente presenti e caricando gli eventuali nuovi elementi;

- *switchOutGraph(): void*

Esegue lo switch all'antistante tipo di diagramma selezionato svuotando il graph dagli elementi correntemente presenti e caricando gli eventuali nuovi elementi;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Client::Model::DataManager*: si occupa della persistenza dei dati, in particolare del salvataggio su file system locale del progetto e del caricamento di un progetto già esistente; *SWEDesigner::Client::Model::ToolbarModel*: È il componente del programma che si occupa di gestire la parte logica della toolbar;
- IN *SWEDesigner::Client::Model::RequestHandler*: si occupa di gestire i dati ricevuti dal server;
- OUT *SWEDesigner::Client::Model::Project*: si occupa di gestire gli elementi contenuti nel diagramma;
- OUT *SWEDesigner::Client::Model::Items::Swedesigner*: è il contenitore degli elementi che si possono inserire in un diagramma.

### 3.3.3 SWEDesigner::Client::Model::ToolbarModel

- **Tipo:** *Classe*;
- **Descrizione:** È il componente del programma che si occupa di gestire la parte logica della toolbar;
- **Padre:** *Backbone.model*;
- **Attributi:**
  - *items: Object*  
Contiene tutti gli elementi definibili nel diagramma corrente;



- **Metodi:**

- *addElement(id: String): void*

Salva lo strumento selezionato interagendo con il *SWEDesigner::Client::Model::ProjectModel*;

**Parametri:**

- \* *id: String* Identificativo del tipo di strumento/elemento da inserire;

- *createItems(): void*

Assegna al campo dati "items" il set di strumenti utilizzabili nel diagramma corrente;

- *currentDiagram(): String*

Ritorna il tipo del diagramma corrente;

- *initialize(): void*

Inizializzazione del ToolbarModel: chiama il metodo createItems;

- **Relazioni con le altre classi:**

- OUT *SWEDesigner::Client::Model::ProjectModel*: si occupa di gestire la parte logica dell'editor;
- OUT *SWEDesigner::Client::Model::Items::Swedesigner*: è il contenitore degli elementi che si possono inserire in un diagramma.

### 3.3.4 SWEDesigner::Client::Model::Project

- **Tipo:** *Classe*;

- **Descrizione:** Contenitore di tutti gli elementi del progetto correntemente aperto nella Single Page Application;

- **Padre:** *Backbone.model*;

- **Attributi:**

- *classes: Object*

Contiene: *classesArray* (array contenente diagrammi delle classi; in ogni indice è presente un oggetto id: *idPackagePadre*, items: [arrayClassiDelDiagramma]) e *dependenciesArray* (array contenente i link del corrispondente diagramma delle classi; in ogni indice è presente un oggetto id: *idPackagePadre*, items: [arrayLinkDelDiagramma]);

- *operations: Array<Object>*

Contiene un array di oggetti; in ogni indice è presente un oggetto id: id dell'operazione, items: [arrayBubbleDelDiagramma]);

- *packages: Object*  
Contiene: *packagesArray* (array contenente i package item del diagramma dei package) e *dependenciesArray* (array contenente i link del diagramma dei package);

- **Metodi:**

- *deleteClassesDiagramOfPkg(id: String): void*  
Elimina il diagramma delle classi associato al package e tutti i diagrammi delle bubble associati alle operazioni delle relative classi;

**Parametri:**

- \* *id: String* Identificativo del package;

- *deleteOperationDiagram(id: String): void*  
Elimina il diagramma delle bubble associato all'operazione;

**Parametri:**

- \* *id: String* Identificativo dell'operazione;

- *getClassIndex(id: String): Number*  
Cerca ed eventualmente ritorna l'indice dell'array *classesArray* del diagramma delle classi associato al package;

**Parametri:**

- \* *id: String* Identificativo del package;

- *getOperationIndex(id: String): Number*  
Cerca ed eventualmente ritorna l'indice dell'array *operations* del diagramma delle bubble associato all'operazione;

**Parametri:**

- \* *id: String* Identificativo dell'operazione;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Client::Model::ProjectModel*: si occupa di gestire la parte logica dell'editor;
- IN *SWEDesigner::Client::Model::DataManager*: si occupa della persistenza dei dati, in particolare del salvataggio su file system locale del progetto e del caricamento di un progetto già esistente;
- IN *SWEDesigner::Client::Model::RequestHandler*: si occupa di gestire i dati ricevuti dal server;

### 3.3.5 SWEDesigner::Client::Model::RequestHandler

- **Tipo:** *Classe*;

- **Descrizione:** Si occupa della gestione delle comunicazioni tra client e server (lato client);
- **Padre:** *Backbone.model*;
- **Metodi:**
  - *caricaJa(): void*  
Carica il file json nel server e ne genera il codice Java restituendo il nome della cartella compressa;
  - *caricaJs(): void*  
Carica il file json nel server e ne genera il codice Javascript restituendo il nome della cartella compressa;
- **Relazioni con le altre classi:**
  - OUT *SWEDesigner::Client::Model::ProjectModel*: si occupa di gestire la parte logica dell'editor;
  - OUT *SWEDesigner::Client::Model::Project*: si occupa di gestire gli elementi contenuti nel diagramma.

### 3.4 SWEDesigner::Client::Model::Items

Questo package non contiene dei sottopackage. Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.4.1 SWEDesigner::Client::Model::Items::Swedesigner

- **Tipo:** *Class*;
- **Descrizione:** Collezione di oggetti che si possono inserire all'interno di un diagramma suddivisi per tipo di diagramma;
- **Relazioni con le altre classi:**
  - IN *SWEDesigner::Client::Model::ProjectModel*
  - IN *SWEDesigner::Client::Model::toolbarModel*

#### 3.4.2 SWEDesigner::Client::Model::Items:: Swedesigner.model.packageDiagram.items.Base

- **Tipo:** *Class*;
- **Descrizione:** Elemento base generico per diagramma dei package UML;
- **Padre:** *joint.shapes.basic.Generic*;
- **Attributi:**

- *toolMarkup: string*  
Markup HTML per la rappresentazione grafica;
- *defaults: Object*  
Attributi di default per l'oggetto;

- **Metodi:**

- *initialize(): void*  
Inizializzazione di Base: imposta evento al verificarsi del cambio dei valori e chiama il metodo per la renderizzazione dell'item;
- *updateRectangles(): void*  
Render dell'item;
- *getValues(): Object*  
Ritorna i valori dell'item;
- *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
  - \* *value: Object*  
Valore da assegnare;
  - \* *path: string*  
Percorso al membro;

### 3.4.3 SWEDesigner::Client::Model::Items::

**Swedesigner.model.packageDiagram.items.BaseView**

- **Tipo:** *Class*;
- **Descrizione:** View per item "Base";
- **Padre:** *joint.dia.ElementView*;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di BaseView: chiama il metodo "initialize" della classe "Base" e imposta un evento alla reazione del model chiamando sequenzialmente i metodi "update" e "resize";
  - *render(): Object*  
Render dell'item;
  - *renderTools(): Object*  
Assistenza al metodo "render" per la renderizzazione dell'item;

### 3.4.4 SWEDesigner::Client::Model::Items::

**Swedesigner.model.packageDiagram.items.Package**

- **Tipo:** *Class*;
- **Descrizione:** Elemento package per diagramma dei package UML;

- **Padre:** *Swedesigner.model.packageDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto Package (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di Package: chiama il metodo "initialize" della classe base e crea l'istanza di Diagram associata al diagramma delle classi relativo al package;
  - *getPackageName(): string*  
Ritorna il nome del Package;
  - *updateRectangles(): void*  
Render del package;

### 3.4.5 SWEDesigner::Client::Model::Items:: Swedesigner.model.packageDiagram.items.PkgComment

- **Tipo:** *Class*;
- **Descrizione:** Commento per diagramma dei package UML;
- **Padre:** *joint.shapes.basic.TextBlock*;
- **Attributi:**
  - *toolMarkup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto PkgComment;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di PkgComment;
  - *getPackageName(): string*  
Ritorna il nome del Package;
  - *getValues(): Objects*  
Ritorna i valori dell'item PkgComment;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;

- \* *path: string*  
Percorso al membro;
- *updateContent(): void*  
Aggiorna il contenuto dell'item PkgComment;

#### 3.4.6 SWEDesigner::Client::Model::Items::

**Swedesigner.model.packageDiagram.items.PkgCommentView**

- **Tipo:** *Class*;
- **Descrizione:** View per oggetto "PkgComment";
- **Padre:** *joint.shapes.basic.TextBlockView*;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di PkgCommentView;
  - *render(): Object*  
Render dell'item PkgCommentView;
  - *renderTools(): Objects*  
Assistenza al metodo "render" per la renderizzazione dell'item;

#### 3.4.7 SWEDesigner::Client::Model::Items::

**Swedesigner.model.packageDiagram.items.packageDiagramLink**

- **Tipo:** *Class*;
- **Descrizione:** Collegamento tra due componenti di un diagramma dei package UML;
- **Padre:** *joint.dia.Link*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di PackageDiagramLink;
  - *getValues(): Object*  
Ritorna i valori del collegamento;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

### 3.4.8 SWEDesigner::Client::Model::Items::

#### Swedesigner.model.packageDiagram.items.PkgCommentLink

- **Tipo:** *Class*;
- **Descrizione:** Link tra un commento e un componente UML del diagramma dei package;
- **Padre:** *Swedesigner.model.packageDiagram.items.packageDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

### 3.4.9 SWEDesigner::Client::Model::Items::

#### Swedesigner.model.packageDiagram.items.PkgDependency

- **Tipo:** *Class*;
- **Descrizione:** Dipendenza tra due package UML del diagramma dei package;
- **Padre:** *Swedesigner.model.packageDiagram.items.packageDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

### 3.4.10 SWEDesigner::Client::Model::Items::

#### Swedesigner.model.classDiagram.items.Base

- **Tipo:** *Class*;
- **Descrizione:** Elemento base generico per diagramma dei package UML;
- **Padre:** *joint.shapes.basic.Generic*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Object*  
Attributi di default per l'oggetto;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di Base: imposta evento al verificarsi del cambio dei valori e chiama il metodo per la renderizzazione dell'item;
  - *getValues(): Object*  
Ritorna i valori dell'item;

- *updateRectangles(): void*  
Render dell'item;
- *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
  - \* *value: Object*  
Valore da assegnare;
  - \* *path: string*  
Percorso al membro;
- *executeMethod(met: function): void*  
Esegue il metodo avente il nome passato in input; Parametri:
  - \* *met: function*  
Metodo da essere eseguito;

#### 3.4.11 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.BaseView

- **Tipo:** *Class*;
- **Descrizione:** View per oggetto "Base";
- **Padre:** *joint.dia.ElementView*;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di BaseView: chiama il metodo "initialize" della classe base e imposta un evento alla reazione del model chiamando sequenzialmente i metodi "update" e "resize";
  - *render(): Object*  
Render dell'item;
  - *renderTools(): Object*  
Assistenza al metodo "render" per la renderizzazione dell'item;

#### 3.4.12 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Class

- **Tipo:** *Class*;
- **Descrizione:** Elemento classe per diagramma delle classi UML;
- **Padre:** *Swedesigner.model.classDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;



- *defaults: Objects*  
Attributi di default per l'oggetto Class (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);

- **Metodi:**

- *initialize(): void*  
Inizializzazione di Class: chiama il metodo "initialize" della classe base;
- *updateRectangles(): void*  
Render della classe;
- *addAttribute(): void*  
Aggiunge un nuovo attributo alla classe;
- *addOperation(): void*  
Aggiunge una nuova operazione alla classe;
- *addParameter(ind: Number): void*  
Aggiunge un nuovo parametro alla classe; Parametri:
  - \* *ind: Number*  
Indice dell'operazione;
- *deleteParameter(ind: Number): void*  
Rimuove un parametro dall'operazione passata in input; Parametri:
  - \* *ind: Number*  
Indice dell'operazione;
- *deleteAttribute(ind: Number): void*  
Rimuove un attributo alla classe; Parametri:
  - \* *ind: Number*  
Indice dell'operazione;
- *deleteOperation(ind: Number): void*  
Rimuove un'operazione dalla classe; Parametri:
  - \* *ind: Number*  
Indice dell'operazione;
- *getAttrsDesc(): Object[]*  
Ritorna la lista di attributi della classe;
- *getOpDesc(): Object[]*  
Ritorna la lista di operazioni della classe;
- *getItemDesc(): Object[]*  
Ritorna le informazioni della classe;
- *getWidth(): Number*  
Ritorna la larghezza dell'oggetto grafico;

### 3.4.13 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Interface

- **Tipo:** *Class*;

- **Descrizione:** Interfaccia UML;
- **Padre:** *Swedesigner.model.classDiagram.items.Class*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di Interface;
  - *updateRectangles(): void*  
Render dell'interfaccia;
  - *addOperation(): void*  
Aggiunge una nuova operazione alla classe;
  - *addParameter(ind: Number): void*  
Aggiunge un parametro all'operazione passata in input; Parametri:
    - \* *ind: Number*  
Indice dell'operazione;
  - *deleteParameter(ind: Number): void*  
Rimuove un parametro dall'operazione passata in input; Parametri:
    - \* *ind: Number*  
Indice dell'operazione;
  - *deleteOperation(ind: Number): void*  
Rimuove un'operazione dalla classe; Parametri:
    - \* *ind: Number*  
Indice dell'operazione;
  - *getOpDesc(): Object[]*  
Ritorna la lista di operazioni della classe;
  - *getItemDesc(): Object[]*  
Ritorna le informazioni della classe;
  - *getWidth(): Number*  
Ritorna la larghezza dell'oggetto grafico;

#### 3.4.14 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.ClComment

- **Tipo:** *Class*;
- **Descrizione:** Commento per diagramma delle classi UML;

- **Padre:** *joint.shapes.basic.TextBlock*;
- **Attributi:**
  - *toolMarkup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto ClComment 1211
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di ClComment;
  - *getValues(): void*  
Ritorna i valori dell'item ClComment;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;
  - *updateContent(): void*  
Aggiorna il contenuto dell'item ClComment;

#### 3.4.15 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.ClCommentView

- **Tipo:** *Class*;
- **Descrizione:** View per oggetto "ClComment";
- **Padre:** *joint.shapes.basic.TextBlockView*;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di ClCommentView;
  - *render(): Object*  
Render dell'item ClCommentView;
  - *renderTools(): Objects*  
Assistenza al metodo "render" per la renderizzazione dell'item;

**3.4.16 SWEDesigner::Client::Model::Items::****Swedesigner.model.classDiagram.items.classDiagramLink**

- **Tipo:** *Class*;
- **Descrizione:** Collegamento tra due componenti di un diagramma delle classi UML;
- **Padre:** *joint.dia.Link*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di classDiagramLink;
  - *getValues(): Object*  
Ritorna i valori del collegamento;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

**3.4.17 SWEDesigner::Client::Model::Items::****Swedesigner.model.classDiagram.items.ClCommentLink**

- **Tipo:** *Class*;
- **Descrizione:** Link tra un commento e un componente UML del diagramma delle classi;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

**3.4.18 SWEDesigner::Client::Model::Items::****Swedesigner.model.classDiagram.items.Generalization**

- **Tipo:** *Class*;
- **Descrizione:** Generalizzazione tra due componenti UML;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;

- **Attributi:**

- *defaults: Objects*  
Attributi di default per l'oggetto;

#### 3.4.19 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Implementation

- **Tipo:** *Class*;
- **Descrizione:** Implementazione tra due componenti UML;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

#### 3.4.20 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Aggregation

- **Tipo:** *Class*;
- **Descrizione:** Aggregazione tra due componenti UML;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

#### 3.4.21 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Composition

- **Tipo:** *Class*;
- **Descrizione:** Composizione tra due componenti UML;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;

### 3.4.22 SWEDesigner::Client::Model::Items:: Swedesigner.model.classDiagram.items.Association

- **Tipo:** *Class*;
- **Descrizione:** Associazione tra due componenti UML;
- **Padre:** *Swedesigner.model.classDiagram.items.classDiagramLink*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;
  - **Metodi:**
    - \* *updatelabel(): void*  
Aggiornamento della label;
    - \* *getcard(): Number*  
Ritorna la cardinalità della label;
    - \* *getAttribute(): string*  
Ritorna l'attributo della label;
    - \* *initialize(): void*  
Inizializzazione della Association;
    - \* *setTo Value(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
      - *value: Object*  
Valore da assegnare;
      - *path: string*  
Percorso al membro;

### 3.4.23 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.Base

- **Tipo:** *Class*;
- **Descrizione:** Elemento base generico per il diagramma delle bubble;
- **Padre:** *joint.shapes.basic.Generic*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Object*  
Attributi di default per l'oggetto;
- **Metodi:**

- *initialize(): void*  
Inizializzazione di Base: imposta evento al verificarsi del cambio dei valori e chiama il metodo per la renderizzazione dell'item;
- *getValues(): Object*  
Ritorna i valori dell'item;
- *updateRectangles(): void*  
Render dell'item;

#### 3.4.24 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.BaseView

- **Tipo:** *Class*;
- **Descrizione:** Elemento view base generico per il diagramma delle bubble;
- **Padre:** *joint.dia.ElementView*;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di BaseView: chiama il metodo "initialize" della classe base e imposta un evento alla reazione del model chiamando sequenzialmente i metodi "update" e "resize";
  - *render(): Object*  
Render dell'item;
  - *renderTools(): Object*  
Assistenza al metodo "render" per la renderizzazione dell'item;

#### 3.4.25 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.customBubble

- **Tipo:** *Class*;
- **Descrizione:** Elemento custom bubble per il diagramma delle bubble;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto customBubble (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**

- *initialize(): void*  
Inizializzazione di customBubble: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto customBubble;
- *updateRectangles(): void*  
Render della custom bubble;
- *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
  - \* *value: Object*  
Valore da assegnare;
  - \* *path: string*  
Percorso al membro;

### 3.4.26 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleIf

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta un'istruzione condizionale;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleIf (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di bubbleIf: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleIf;
  - *updateRectangles(): void*  
Render della bubbleIf;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;



### 3.4.27 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleElse

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta il ramo 'else' di un'istruzione condizionale;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleElse (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di bubbleIf: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleElse;
  - *updateRectangles(): void*  
Render della bubbleElse;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

### 3.4.28 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleFor

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta un'iterazione lungo una sequenza di istruzioni;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleFor (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**

- *initialize(): void*  
Inizializzazione di bubbleFor: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleFor;
- *updateRectangles(): void*  
Render della bubbleFor;
- *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
  - \* *value: Object*  
Valore da assegnare;
  - \* *path: string*  
Percorso al membro;

### 3.4.29 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleReturn

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta un'istruzione per uscire da un metodo e ritornare degli argomenti al chiamante;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleReturn (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di bubbleReturn: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleReturn;
  - *updateRectangles(): void*  
Render della bubbleReturn;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

### 3.4.30 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleStart

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta la prima istruzione di un metodo;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleStart (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di bubbleStart: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleStart;
  - *updateRectangles(): void*  
Render della bubbleStart;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

### 3.4.31 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleWhile

- **Tipo:** *Class*;
- **Descrizione:** Rappresenta un loop con controllo di condizione lungo una sequenza di istruzioni;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.Base*;
- **Attributi:**
  - *markup: string*  
Markup HTML per la rappresentazione grafica;
  - *defaults: Objects*  
Attributi di default per l'oggetto bubbleWhile (tipo, posizione, dimensione, attributi CSS, stato e contenuto dell'oggetto);

- **Metodi:**

- *initialize(): void*  
Inizializzazione di bubbleWhile: chiama il metodo "initialize" della classe base e crea l'istanza dell'oggetto bubbleWhile;
- *updateRectangles(): void*  
Render della bubbleWhile;
- *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
  - \* *value: Object*  
Valore da assegnare;
  - \* *path: string*  
Percorso al membro;

### 3.4.32 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleDiagramLink

- **Tipo:** *Class*;
- **Descrizione:** Collegamento tra due componenti di un diagramma delle bubble;
- **Padre:** *joint.dia.Link*;
- **Attributi:**
  - *defaults: Objects*  
Attributi di default per l'oggetto;
- **Metodi:**
  - *initialize(): void*  
Inizializzazione di bubbleDiagramLink;
  - *setToValue(value: Object, path: string): void*  
Imposta "values.path" a "value"; Parametri:
    - \* *value: Object*  
Valore da assegnare;
    - \* *path: string*  
Percorso al membro;

### 3.4.33 SWEDesigner::Client::Model::Items:: Swedesigner.model.bubbleDiagram.items.bubbleLink

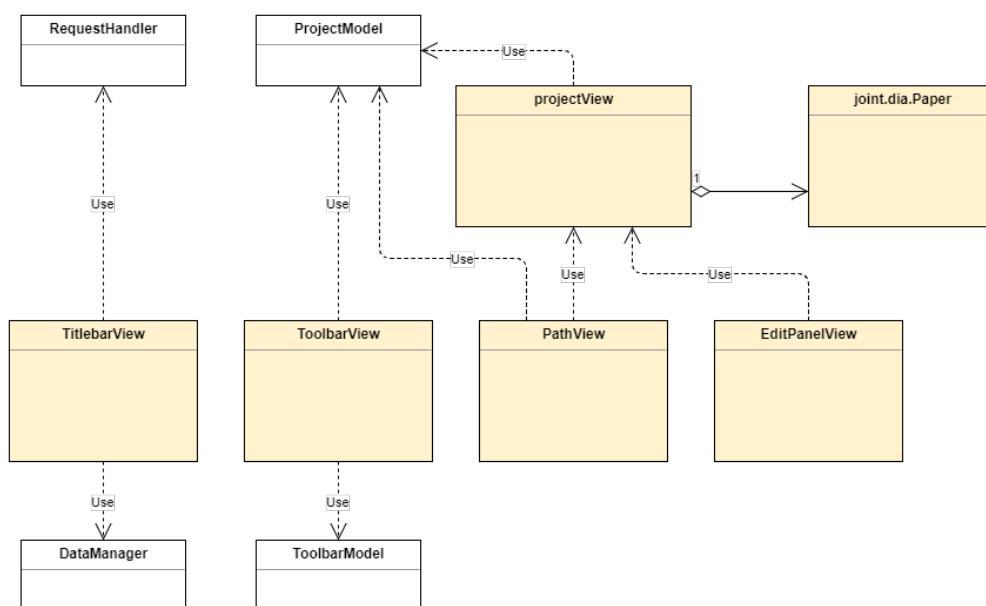
- **Tipo:** *Class*;
- **Descrizione:** Link tra due elementi del diagramma delle bubble;
- **Padre:** *Swedesigner.model.bubbleDiagram.items.bubbleDiagramLink*;

- **Attributi:**

- *defaults: Objects*  
Attributi di default per l'oggetto;

### 3.5 SWEDesigner::Client::View

È il componente del programma che si occupa di gestire l'interfaccia grafica. Nella particolare declinazione MVC adottata da Backbone.js, si occupa anche di gestire gli input dell'utente.



**Figura 6:** Architettura di View

Questo package non contiene dei sottopackage. Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.5.1 SWEDesigner::Client::View::ProjectView

- **Tipo:** *Class*;
- **Descrizione:** Questa classe gestisce il diagramma disegnato e le interazioni dell'utente con esso;
- **Padre:** *Backbone.View*;
- **Attributi:**
  - *model*  
Istanza di *ProjectModel* del programma;
  - *paper*  
Oggetto *joint.dia.Paper* della libreria esterna JointJS;

- **Metodi:**

- *resetSelectedCell(): void*  
Pone this.paper.selectedCell a null e genera l'evento "changed-selected-cell";
- *mouseMoveFunction(event: JavaScriptEvent): void*  
Provoca la traslazione del paper nella direzione del trascinamento del mouse;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *blankPointerDown(elem: CellView, event: JavaScriptEvent, x: Double, y: Double): void*  
Salva le correnti coordinate al click del mouse nello spazio vuoto del paper;  
Parametri:
  - \* *elem: CellView*: Elemento cellView;
  - \* *event: JavaScriptEvent*: Evento;
  - \* *x: Double*: Coordinata dell'asse delle ascisse;
  - \* *y: Double*: Coordinata dell'asse delle ordinate;
- *blankPointerUp(elem: CellView, event: JavaScriptEvent, x: Double, y: Double): void*  
Elimina le coordinate iniziali al click del mouse nello spazio vuoto del paper;  
Parametri:
  - \* *elem: CellView*: Elemento cellView;
  - \* *event: JavaScriptEvent*: Evento;
  - \* *x: Double*: Coordinata dell'asse delle ascisse;
  - \* *y: Double*: Coordinata dell'asse delle ordinate;
- *onMouseWheel(elem: CellView, event: JavaScriptEvent): void*  
Trasla verticalmente il paper effettuando uno zoom in avanti o indietro a seconda della rotazione della ruota del mouse;  
Parametri:
  - \* *elem: CellView*: Elemento cellView;
  - \* *event: JavaScriptEvent*: Evento;
- *render(): void*  
Provoca il render della projectView;
- *addCell(elem: CellView, event: JavaScriptEvent, x: Double, y: Double): void*  
Aggiunge un nuovo elemento al graph chiamando il relativo metodo di [Project-Model](#);  
Parametri:
  - \* *elem: cellView*: Elemento CellView;
  - \* *event: JavaScriptEvent*: Evento;
  - \* *x: Double*: Coordinata dell'asse delle ascisse;
  - \* *y: Double*: Coordinata dell'asse delle ordinate;

- *deleteCell(event: JavaScriptEvent): void*  
Elimina un elemento dal graph chiamando il relativo metodo di ProjectModel;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *unembedCell(event: JavaScriptEvent): void*  
Rimuove l'innestamento della cella selezionata;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *pointerDownFunction(prView: [projectView](#), elem: cellView, event: JavaScriptEvent, x: double, y: double): void*  
Gestisce l'evento generato dal click (non rilasciato) del mouse nel paper. Se viene cliccato un elemento, genera a sua volta l'evento "changed-selected-cell" gestito da [EditPanelView](#);  
Parametri:
  - \* *prView: [ProjectView](#)*: Istanza di ProjectView;
  - \* *elem: cellView*: Elemento CellView;
  - \* *event: JavaScriptEvent*: Evento;
  - \* *x: Double*: Coordinata dell'asse delle ascisse;
  - \* *y: Double*: Coordinata dell'asse delle ordinate;
- *pointerUpFunction(prView: [ProjectView](#), elem: CellView, event: JavaScriptEvent, x: Double, y: Double): void*  
Gestisce l'evento generato dal click (al rilascio) del mouse nel paper (rimozione di un elemento, nesting di un elemento in un'altro, collegamento di una relazione tra elementi);  
Parametri:
  - \* *prView: [projectView](#)*: Istanza di projectView;
  - \* *elem: CellView*: Elemento cellView;
  - \* *event: JavaScriptEvent*: Evento;
  - \* *x: Double*: Coordinata dell'asse delle ascisse;
  - \* *y: Double*: Coordinata dell'asse delle ordinate;
- *switchIn(id: String): void*  
Gestisce lo switch in profondità (dall'elemento selezionato il cui id è parametro in input) invocando il relativo metodo di [ProjectModel](#);  
Parametri:
  - \* *id: String*: Identificativo dell'elemento;
- *switchOut(diagramType: String): void*  
Gestisce lo switch verso un diagramma (il cui tipo è parametro in input) antistante da quello corrente invocando il relativo metodo di [ProjectModel](#).  
Parametri:
  - \* *diagramType: String*: Tipo di diagramma di destinazione;

- *deleteOperationAt(ind: Int): void*  
Gestisce l'eliminazione di un diagramma delle bubble invocando il relativo metodo di [projectModel](#).

Parametri:

- \* *ind: Int*: Indice dell'array di operazioni del diagramma delle bubble da eliminare;

- **Relazioni con le altre classi:**

- IN [PathView](#): gestisce l'interfaccia grafica della barra di indirizzo;
- IN [EditPanelView](#): gestisce l'interfaccia grafica del pannello di editing.
- OUT [ProjectModel](#): si occupa di gestire la parte logica dell'editor;
- OUT `joint.dia.Paper`: gestisce l'interfaccia grafica dell'area dei diagrammi.

### 3.5.2 SWEDesigner::Client::View::TitlebarView

- **Tipo:** *Class*;

- **Descrizione:** È il componente del programma che fa la funzione di view per la barra del titolo, dove saranno collocati il menu dell'applicazione e gli shortcut;

- **Padre:** *Backbone.View*;

- **Attributi:**

- *el: String*  
Il tag HTML popolato dalla Titlebar;
- *events: Object*  
Gli eventi verificabili nella titlebar;

- **Metodi:**

- *generateJava(event: JavaScriptEvent): void*  
Richiede al server di generare il codice in linguaggio Java del progetto correntemente aperto invocando il rispettivo metodo di [RequestHandler](#);  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *generateJavascript(event: JavaScriptEvent): void*  
Richiede al server di generare il codice in linguaggio JavaScript del progetto correntemente aperto invocando il rispettivo metodo di [RequestHandler](#);  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *newProject(event: JavaScriptEvent): void*  
Crea un nuovo progetto invocando il rispettivo metodo di [DataManager](#)  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;



- *openProject(event: JavaScriptEvent): void*  
Apri un progetto invocando il rispettivo metodo di *DataManager*  
Parametri:
    - \* *event: JavaScriptEvent*: Evento;
  - *saveProject(event: JavaScriptEvent): void*  
Salva il progetto correntemente aperto invocando il rispettivo metodo di *DataManager*  
Parametri:
    - \* *event: JavaScriptEvent*: Evento;
  - *saveProjectAs(event: JavaScriptEvent): void*  
Salva il progetto correntemente aperto con nome specificato dall'utente invocando il rispettivo metodo di *DataManager*  
Parametri:
    - \* *event: JavaScriptEvent*: Evento;
- **Relazioni con le altre classi:**
    - OUT *RequestHandler*: gestisce la comunicazione con il server;
    - OUT *DataManager*: gestisce la persistenza dei dati su file system.

### 3.5.3 SWEDesigner::Client::View::ToolBarView

- **Tipo:** *Class*;
- **Descrizione:** È il componente del programma che fa la funzione di view per la toolbar dove saranno collocati gli strumenti per editare i diagrammi;
- **Padre:** *Backbone.View*;
- **Attributi:**
  - *el: String*  
Il tag HTML popolato dalla Toolbar;
  - *events: Object*  
Gli eventi verificabili nella Toolbar;
- **Metodi:**
  - *addElement(event: JavaScriptEvent): void*  
Aggiunge un elemento al diagramma alla selezione di uno strumento invocando il rispettivo metodo di *ToolBarModel*;  
Parametri:
    - \* *event: JavaScriptEvent*: Evento;
  - *initialize(): void*  
Inizializza ToolBarView;

- *render(): void*  
Provoca il render della toolbar in base al diagramma correntemente visualizzato;

- **Relazioni con le altre classi:**

- OUT *ProjectModel*: si occupa di gestire la parte logica dell'editor;
- OUT *ToolbarModel*: si occupa di gestire la parte logica della toolbar.

### 3.5.4 SWEDesigner::Client::View::PathView

- **Tipo:** *Class*;

- **Descrizione:** È il componente del programma che fa la funzione di view per il cosiddetto breadcrumb dove viene inserita la posizione corrente;

- **Padre:** *Backbone.View*;

- **Attributi:**

- *el: String*  
Il tag HTML popolato dalla path;
- *events: Object*  
Gli eventi verificabili nella path;

- **Metodi:**

- *initialize(): void*  
Inizializza PathView;
- *render(): void*  
Provoca il render del path in base al diagramma correntemente visualizzato;
- *switchDiagram(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato. Switch verso un tipo antistante di diagramma.;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;

- **Relazioni con le altre classi:**

- OUT *ProjectModel*: si occupa di gestire la parte logica dell'editor;
- OUT *ProjectView*: si occupa di gestire la parte grafica del model.

### 3.5.5 SWEDesigner::Client::View::EditPanelView

- **Tipo:** *Class*;

- **Descrizione:** ;

- **Padre:** *Backbone.View*;

- **Attributi:**

- *currentTemplate: Object*  
Il template correntemente caricato e renderizzato;
- *el: String*  
Il tag HTML popolato dalla path;
- *events: Object*  
Gli eventi verificabili nella path;
- *tagname: String*  
Il tag HTML popolato dal pannello;

- **Metodi:**

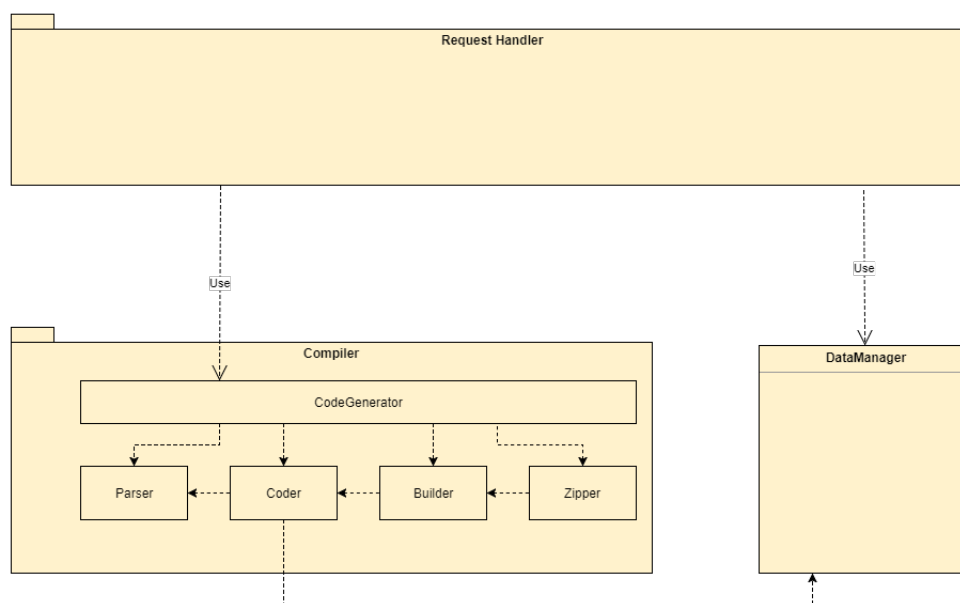
- *confirmEdit(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, salva le modifiche apportate ad una proprietà del contenuto selezionato nel pannello;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *execCommand(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, esegue il metodo definito dal nome dell'elemento generante l'evento sul contenuto selezionato nel pannello;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *initialize(): void*  
Inizializza la EditPanelView;
- *render(): void*  
Provoca il render del pannello in base all'elemento del paper cliccato;
- *reset(): void*  
Provoca il reset del pannello;
- *saveCode(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, salva il codice interno ad una custom-Bubble;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *switch(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, esegue lo switch in profondità del tipo di diagramma;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;
- *switchDiagram(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, esegue lo switch verso un tipo antistante di diagramma;  
Parametri:

- \* *event: JavaScriptEvent*: Evento;
- *unembedCell(event: JavaScriptEvent): void*  
Metodo chiamato da evento generato, rimuove la bubble selezionata nel pannello dall'innesto;  
Parametri:
  - \* *event: JavaScriptEvent*: Evento;

- **Relazioni con le altre classi:**

- OUT *ProjectView*: si occupa di gestire la parte grafica del model.

### 3.6 SWEDesigner::Server



**Figura 7:** Architettura del server

I package contenuti al suo interno sono:

- SWEDesigner::Server::CodeGenerator;
- SWEDesigner::Server::DataManager;
- SWEDesigner::Server::RequestHandler.

Questo package non contiene delle classi.

### 3.7 SWEDesigner::Server::CodeGenerator

I package contenuti al suo interno sono:

- SWEDesigner::Server::CodeGenerator::Builder;

- SWEDesigner::Server::CodeGenerator::Coder;
- SWEDesigner::Server::CodeGenerator::Parser;
- SWEDesigner::Server::CodeGenerator::Zipper.

Le classi contenute al suo interno verranno elencate qui di seguito.

### 3.7.1 SWEDesigner::Server::CodeGenerator::CodeGenerator

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità per cui, dato un file in formato JSON che contiene le informazioni necessarie a codificare un programma, restituisce un pacchetto in formato .zip contenente i file del codice sorgente che costituiscono il programma rappresentato dal file in input. I file prodotti sono organizzati in packages, come indicato nel file JSON in input;
- **Metodi:**
  - *generateJsProgram(jsonProgram:JSON, nomeZip:string):void*  
Funzione statica che codifica, in Javascript, il programma corrispondente al contenuto del JSON in input e costruisce un pacchetto compresso in formato .zip contenente il programma codificato e strutturato in file e directory come specificato nel file JSON di input;  
Parametri:
    - \* *jsonProgram:JSON*: Contiene le informazioni, in formato JSON, necessarie a codificare un programma.
    - \* *nomeZip:string*: Specifica il nome con cui verrà nominato il pacchetto zip prodotto.
  - *generateJavaProgram(jsonProgram:JSON, nomeZip:string):void*  
Funzione statica che codifica, in Java, il programma corrispondente al contenuto del JSON in input e costruisce un pacchetto compresso in formato .zip contenente il programma codificato e strutturato in file e directory come specificato nel file JSON di input;  
Parametri:
    - \* *jsonProgram:JSON*: Contiene le informazioni, in formato JSON, necessarie a codificare un programma.
    - \* *nomeZip:string*: Specifica il nome con cui verrà nominato il pacchetto zip prodotto.

- **Relazioni con le altre classi:**

- OUT *SWEDesigner::Server::CodeGenerator::Parser::Parser*
- OUT *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*
- OUT *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*
- OUT *SWEDesigner::Server::CodeGenerator::Builder::Builder*
- OUT *SWEDesigner::Server::CodeGenerator::Zipper::Zipper*
- IN *SWEDesigner::Server::RequestHandler::RequestHandler*

### 3.8 SWEDesigner::Server::CodeGenerator::Parser

Questo package non contiene dei sottopackage. Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.8.1 SWEDesigner::Server::CodeGenerator::Parser::Parser

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità, dato un file JSON valido in input, di ottenere un oggetto contenente le informazioni che costituiscono il file JSON di input;
- **Metodi:**
  - *parse(jsonFile:JSON):Object*  
Trasforma il file JSON ricevuto in input in un oggetto Javascript contenente tutte le sue informazioni;  
Parametri:
    - \* *jsonFile:JSON* Contiene le informazioni, in formato JSON, necessarie a codificare un programma.
- **Relazioni con le altre classi:**
  - IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.9 SWEDesigner::Server::CodeGenerator::Coder

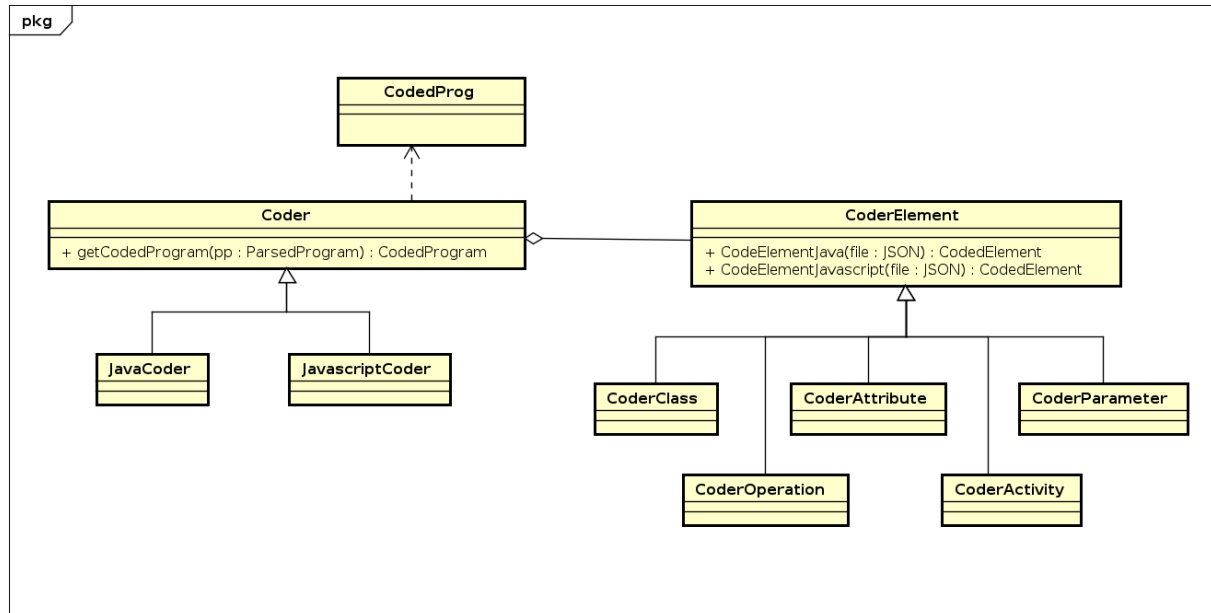


Figura 8: Architettura di Coder

Questo package non contiene dei sottopackage.

Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.9.1 SWEDesigner::Server::CodeGenerator::Coder::JavaCoder

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità, dato un oggetto in input che contiene le informazioni necessarie a codificare un programma, di ottenere un oggetto contenente il codice sorgente, in linguaggio Java, corrispondente all'oggetto in input;
- **Metodi:**
  - *coderParameters(operationObj:Object):string*  
Riceve in input operationObj, un oggetto che contiene le informazioni necessarie a codificare una operazione; restituisce la stringa del codice sorgente, in Java, della lista dei parametri dell'operazione operationObj di input;  
Parametri:
    - \* *operationObj:Object* Contiene le informazioni necessarie a codificare una operazione.
  - *coderAttributes(classObj:Object):string*  
Riceve in input classObj, un oggetto che contiene le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Java, di tutti gli attributi della classe classObj di input;  
Parametri:

- \* *classObj:Object* Contiene le informazioni necessarie a codificare una classe.
- *coderOperations(classObj,operations):string*  
Riceve in input classObj, un oggetto che le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Java, relativa alla definizione e implementazione di tutte le operazioni di classObj di input;  
Parametri:
  - \* *classObj:Object* Contiene le informazioni necessarie a codificare una classe;
  - \* *operations:Object* Contiene le informazioni necessarie a codificare l'implementazione di tutte le operazioni di classObj.
- *getCodedProgram(parsedProgram:Object):CodedProgram*  
Riceve in input parsedProgram, un oggetto che contiene le informazioni necessarie a codificare un programma; restituisce un oggetto istanza di CodedProgram, contenente il codice sorgente in Java di tutte le classi presenti nell'oggetto parsedProgram di input;  
Parametri:
  - \* *parsedProgram:Object* Contiene le informazioni necessarie a codificare un programma.

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.9.2 SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità, dato un oggetto in input che contiene le informazioni necessarie a codificare un programma, di ottenere un oggetto contenente il codice sorgente, in linguaggio Javascript, corrispondente all'oggetto in input;
- **Metodi:**
  - *coderParameters(operationObj:Object):string*  
Riceve in input operationObj, un oggetto che contiene le informazioni necessarie a codificare una operazione; restituisce la stringa del codice sorgente, in Javascript, della lista dei parametri dell'operazione operationObj di input;  
Parametri:
    - \* *operationObj:Object* Contiene le informazioni necessarie a codificare una operazione.
  - *coderInstanceAttributes(classObj:Object):string*  
Riceve in input classObj, un oggetto che contiene le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Javascript, di tutti gli attributi non statici della classe classObj di input;  
Parametri:



- \* *classObj: Object* Contiene le informazioni necessarie a codificare una classe.
- *coderInstanceOperations(classObj, operations): string*  
Riceve in input *classObj*, un oggetto che le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Javascript, relativa alla definizione e implementazione di tutte le operazioni non statiche di *classObj* di input;  
Parametri:
  - \* *classObj: Object* Contiene le informazioni necessarie a codificare una classe;
  - \* *operations: Object* Contiene le informazioni necessarie a codificare l'implementazione di tutte le operazioni di *classObj*.
- *coderStaticAttributes(classObj: Object): string*  
Riceve in input *classObj*, un oggetto che contiene le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Javascript, di tutti gli attributi statici della classe *classObj* di input;  
Parametri:
  - \* *classObj: Object* Contiene le informazioni necessarie a codificare una classe.
- *coderStaticOperations(classObj, operations): string*  
Riceve in input *classObj*, un oggetto che le informazioni necessarie a codificare una classe; restituisce la stringa del codice sorgente, in Javascript, relativa alla definizione e implementazione di tutte le operazioni statiche di *classObj* di input;  
Parametri:
  - \* *classObj: Object* Contiene le informazioni necessarie a codificare una classe;
  - \* *operations: Object* Contiene le informazioni necessarie a codificare l'implementazione di tutte le operazioni di *classObj*.
- *getCodedProgram(parsedProgram: Object): CodedProgram*  
Riceve in input *parsedProgram*, un oggetto che contiene le informazioni necessarie a codificare un programma; restituisce un oggetto istanza di *CodedProgram*, contenente il codice sorgente in Javascript di tutte le classi presenti nell'oggetto *parsedProgram* di input;  
Parametri:
  - \* *parsedProgram: Object* Contiene le informazioni necessarie a codificare un programma.

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.9.3 SWEDesigner::Server::CodeGenerator::Coder::CoderOperation

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità che permette di codificare l'istanza dell'operazione corrispondente al contenuto dell'oggetto *operationObj* di input;

- **Metodi:**

- *codeElementJava(operationObj:Object):string*

- Riceve in input operationObj, un oggetto che contiene le informazioni necessarie a codificare una operazione; restituisce la stringa codice sorgente in Java relativa all'intestazione dell'operazione operationObj di input;

- Parametri:

- \* *operationObj:Object* Contiene le informazioni necessarie a codificare una operazione.

- *codeElementJavascript(operationObj:Object, className:string):string*

- Riceve in input operationObj, un oggetto che contiene le informazioni necessarie a codificare una operazione; restituisce la stringa codice sorgente in Javascript relativa all'intestazione dell'operazione operationObj di input;

- Parametri:

- \* *operationObj:Object* Contiene le informazioni necessarie a codificare una operazione;

- \* *className:string* Il nome della classe che possiede l'operazione operationObj. Necessaria solo se tale operazione è statica;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*

- IN *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*

### 3.9.4 SWEDesigner::Server::CodeGenerator::Coder::CoderParameter

- **Tipo:** *Class*;

- **Descrizione:** Rende disponibile la funzionalità che permette di codificare un parametro, corrispondente al contenuto dell'oggetto parameterObj di input;

- **Metodi:**

- *codeElementJava(parameterObj:Object):string*

- Riceve in input parameterObj, un oggetto che contiene le informazioni necessarie a codificare un parametro di operazione; restituisce la stringa codice sorgente in Java relativa al parametro parameterObj di input;

- Parametri:

- \* *parameterObj:Object* Contiene le informazioni necessarie a codificare un parametro di operazione.

- *codeElementJavascript(parameterObj:Object):string*

- Riceve in input parameterObj, un oggetto che contiene le informazioni necessarie a codificare un parametro di operazione; restituisce la stringa codice sorgente in Javascript relativa al parametro parameterObj di input;

- Parametri:

- \* *parameterObj:Object* Contiene le informazioni necessarie a codificare un parametro di operazione.

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*
- IN *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*

### 3.9.5 SWEDesigner::Server::CodeGenerator::Coder::CoderAttribute

- **Tipo:** *Class*;

- **Descrizione:** Rende disponibile la funzionalità che permette di codificare un attributo corrispondente al contenuto dell'oggetto attributeObj di input;

- **Metodi:**

- *codeElementJava(attributeObj:Object):string*  
Riceve in input attributeObj, un oggetto che contiene le informazioni necessarie a codificare un attributo di classe; restituisce la stringa codice sorgente in Java relativa all'attributo attributeObj di input;  
Parametri:
  - \* *attributeObj:Object* Contiene le informazioni necessarie a codificare un attributo di classe.
- *codeElementJavascript(attributeObj:Object, className:string):string*  
Riceve in input attributeObj, un oggetto che contiene le informazioni necessarie a codificare un attributo; restituisce la stringa codice sorgente in Javascript relativa all'attributo attributeObj di input;  
Parametri:
  - \* *attributeObj:Object* Contiene le informazioni necessarie a codificare un attributo di classe;
  - \* *className:string* Il nome della classe che possiede l'attributo attributeObj. Necessaria solo se tale attributo è statico;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*
- IN *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*

### 3.9.6 SWEDesigner::Server::CodeGenerator::Coder::CoderActivity

- **Tipo:** *Class*;

- **Descrizione:** Rende disponibile la funzionalità che permette di codificare l'implementazione di un'operazione, corrispondente al contenuto dell'oggetto activityObj di input;

- **Metodi:**

- *getBubbleLinks(activityObj:Object):Array*

Estrae, per ogni elemento in activityObj, le informazioni relative al collegamento con un'altro elemento (se esiste) contenuto in ActivityObj e le inserisce in un array che viene restituito;

Parametri:

- \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.

- *getBubbleId(bubbleId:string, activityObj:Object):Object*

Ritorna la bubble (oggetto contenente le informazioni necessarie a codificare una porzione dell'implementazione di activityObj) contenuta in activityObj e il cui identificativo corrisponde al bubbleId di input;

Parametri:

- \* *bubbleId:string* Stringa identificativa della bubble che si vuole ottenere;
    - \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.

- *getNextBubble(bubbleObj:Object, activityObj:Object):Object*

Ritorna la bubble (oggetto contenente le informazioni necessarie a codificare una porzione dell'implementazione di activityObj) contenuta in activityObj, successiva alla bubble di input anch'essa contenuta in activityObj;

Parametri:

- \* *bubbleObj:Object* Contiene le informazioni necessarie a codificare una porzione dell'implementazione di un'operazione;
    - \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.

- *getStartBubble(bubbleArray:Array, parent:string):Object*

Ritorna la bubble (oggetto contenente le informazioni necessarie a codificare una porzione dell'implementazione di una operazione) contenuta in bubbleArray, corrispondente alla bubble iniziale, ovvero la bubble da cui comincia la sotto attività rappresentata da bubbleArray;

Parametri:

- \* *bubbleArray:Array* Contiene le informazioni necessarie a codificare una sotto attività di un'operazione;
    - \* *parent:string* Contiene l'identificativo della bubble in cui sono innestate le bubble presenti in bubbleArray.

- *codeEmbeddedBubbles(bubbleObj:Object, activityObj:Object):Object*

Codifica la bubbleObj (oggetto contenente le informazioni necessarie a codificare una porzione dell'implementazione di una operazione) di input e tutte le

bubble innestate in essa; tale bubbleObj dev'essere contenuta in activityObj  
Parametri:

- \* *bubbleObj:Object* Contiene le informazioni necessarie a codificare una porzione dell'implementazione di un'operazione;
  - \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.
- *codeBubble(bubbleObj:Object, activityObj:Object, parent:string):Object*  
Codifica la bubbleObj (oggetto contenente le informazioni necessarie a codificare una porzione dell'implementazione di una operazione) di input; tale bubbleObj dev'essere contenuta in activityObj  
Parametri:
- \* *bubbleObj:Object* Contiene le informazioni necessarie a codificare una porzione dell'implementazione di un'operazione;
  - \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.
  - \* *parent:string* Contiene l'identificativo della bubble in cui è innestata la bubbleObj di input.
- *codeElementJava(activityObj:Object):string*  
Riceve in input activityObj, un oggetto che contiene le informazioni necessarie a codificare l'implementazione di un operazione; restituisce la stringa codice sorgente in Java corrispondente all'implementazione della activityObj di input;  
Parametri:
- \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.
- *codeElementJavascript(activityObj:Object):string*  
Riceve in input activityObj, un oggetto che contiene le informazioni necessarie a codificare l'implementazione di un operazione; restituisce la stringa codice sorgente in Javascript corrispondente all'implementazione della activityObj di input;  
Parametri:
- \* *activityObj:Object* Contiene le informazioni necessarie a codificare l'implementazione di un operazione. E costituito da elementi (bubble) ognuno contenente le informazioni necessarie a codificare una porzione dell'implementazione dell'operazione.

● **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*
- IN *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*

### 3.9.7 SWEDesigner::Server::CodeGenerator::Coder::CodedProg

- **Tipo:** *Class*;
- **Descrizione:** È il componente che contiene il codice sorgente prodotto dal Coder;
- **Attributi:**
  - *\_classes: Array*: Array contenente le informazioni di ogni classe codificata dal Coder. Tali informazioni comprendono il nome della classe, il suo codice sorgente, il nome del package in cui è contenuta e l'array delle dipendenze OUT;
- **Metodi:**
  - *add(\_class:Object):void*  
Inserisce nell'array *\_classes* le informazioni della classe *\_class* codificata dal Coder;  
Parametri:
    - \* *\_class:Object* Contiene il nome della classe, il suo codice sorgente, il nome del package in cui è contenuta e l'array delle dipendenze OUT.
  - *getSource(i:int):string*  
Restituisce il codice sorgente della classe contenuta nell'attributo *\_classes* all'indice *i*;  
Parametri:
    - \* *i:int* Indice della classe, contenuta nell'attributo *\_classes*, di cui si vuole il codice sorgente;
- **Relazioni con le altre classi:**
  - IN *SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder*
  - IN *SWEDesigner::Server::CodeGenerator::Coder::JavaCoder*

### 3.10 SWEDesigner::Server::CodeGenerator::Builder

Questo package non contiene dei sottopackage.

Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.10.1 SWEDesigner::Server::CodeGenerator::Builder::Builder

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità che permette di creare su disco la directory contenente il programma corrispondente all'oggetto program di tipo CodedProgram;
- **Attributi:**

- *generalPath: string*: Contiene il path alla directory dove vengono creati tutti i programmi (directories, file sorgenti, ...);

- **Metodi:**

- *deleteFolderRecursive(path:string):void*

Rimuove la directory passata in input se esistente e tutto il suo contenuto ricorsivamente;

Parametri:

- \* *path:string* Il percorso alla directory da rimuovere.

- *mkJavaFile(progDir:string, name:string, pkg:string, dependencies:Array, source:string):void*

Crea un file sorgente in Java e/o scrive ulteriore codice in append se già esistente;

Parametri:

- \* *progDir:string* La directory indicante dove creare i file sorgenti del programma;
- \* *name:string* Il nome del file;
- \* *pkg:string* Il package in cui è contenuto il file;
- \* *dependencies:Array* L'array delle dipendenze del package;
- \* *source:string* Il codice sorgente da scrivere nel file;

- *mkJavascriptFile(progDir:string, name:string, pkg:string, dependencies:Array, source:string):void*

Crea un file sorgente in Javascript e/o scrive ulteriore codice in append se già esistente;

Parametri:

- \* *progDir:string* La directory indicante dove creare i file sorgenti del programma;
- \* *name:string* Il nome del file;
- \* *pkg:string* Il package in cui è contenuto il file;
- \* *dependencies:Array* L'array delle dipendenze del package;
- \* *source:string* Il codice sorgente da scrivere nel file;

- *javaBuild(program:CodedProgram):Object*

Crea directory e file su disco contenente il programma in linguaggio java corrispondente all'oggetto program di input di tipo CodedProgram;;

Parametri:

- \* *program:CodedProgram* L'oggetto che contiene il codice sorgente prodotto dal Coder;.

- *javascriptBuild(program:CodedProgram):Object*

Crea directory e file su disco contenente il programma in linguaggio javascript corrispondente all'oggetto program di input di tipo CodedProgram;;

Parametri:

- \* *program:CodedProgram* L'oggetto che contiene il codice sorgente prodotto dal Coder;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.11 SWEDesigner::Server::CodeGenerator::Zipper

Questo package non contiene dei sottopackage.

Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.11.1 SWEDesigner::Server::CodeGenerator::Zipper::Zipper

- **Tipo:** *Class*;
- **Descrizione:** Rende disponibile la funzionalità che permette di creare un pacchetto in formato zip della directory specificata in input;
- **Metodi:**
  - *mkJavaFile(progDir:string, name:string, pkg:string, dependencies:Array, source:string):void*  
Crea un file sorgente in Java e/o scrive ulteriore codice in append se già esistente;  
Parametri:
    - \* *name:string* Il nome dell'archivio da creare;
    - \* *namepath:string* Il percorso alla directory del progetto da comprimere;
    - \* *callback:function* Callback di risposta;
- **Relazioni con le altre classi:**
  - IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.12 SWEDesigner::Server::DataManager

Questo package non contiene dei sottopackage.

Le classi contenute al suo interno verranno elencate qui di seguito.

#### 3.12.1 SWEDesigner::Server::DataManager::DataManager

- **Tipo:** *Class*;
- **Descrizione:** Espone le funzionalità che permettono di interagire con il database delle bubbles;
- **Attributi:**



- *host\_:string*: L’hostname del database a cui connettersi;
- *user:string*: L’username per il login al database;
- *password:string*: La password per il login al database ;
- *database:string*: Il nome del database a cui connettersi;
- *connection:Object*: La connessione al database;

- **Metodi:**

- *setConnection(host:string, user:string, psw:string, db:string):void*  
Setta gli attributi necessari a creare una connessione;  
Parametri:
  - \* *host\_:string*: L’hostname del database a cui connettersi;
  - \* *user:string*: L’username per il login al database;
  - \* *psw:string*: La password per il login al database ;
  - \* *db:string*: Il nome del database a cui connettersi;
- *\_startConnection():void*  
Inizializza l’attributo connection e esegue la connessione al database;
- *insertBubble(name:string, source:string, language:string, descr:string):void*  
Inserisce una nuova bubble nel database;  
Parametri:
  - \* *name:string*: Attributo ‘Name’ della bubble da inserire;
  - \* *source:string*: Attributo ‘Source’ della bubble da inserire;
  - \* *language:string*: Attributo ‘Language’ della bubble da inserire ;
  - \* *descr:string*: Attributo ‘Description’ della bubble da inserire;
- *deleteBubble(name:string, source:string, language:string, descr:string):void*  
Elimina una bubble se presente nel database;  
Parametri:
  - \* *name:string*: Attributo ‘Name’ della bubble da rimuovere;
  - \* *source:string*: Attributo ‘Source’ della bubble da rimuovere;
  - \* *language:string*: Attributo ‘Language’ della bubble da rimuovere ;
- *getBubble(name:string, language:string):void*  
Restituisce le informazioni di una bubble se presente nel database;  
Parametri:
  - \* *name:string*: Attributo ‘Name’ della bubble che si vuole ottenere;
  - \* *language:string*: Attributo ‘Language’ della bubble che si vuole ottenere;
- *getAllBubble():void*  
Restituisce le informazioni di tutte le bubble presenti nel database;

- **Relazioni con le altre classi:**

- IN *SWEDesigner::Server::CodeGenerator::CodeGenerator*

### 3.13 SWEDesigner::Server::RequestHandler

Questo package non contiene dei sottopackage.

Le classi contenute al suo interno verranno elencate qui di seguito

### 3.14 SWEDesigner::Server::RequestHandler

::RequestHandler

- **Tipo:** *Class*;
- **Descrizione:** Espone le funzionalità che permettono di interagire con il database delle bubbles;
- **Metodi:**
  - *getIndex(req:string, res:string):void*  
Invia il file index della Single Page Application;  
Parametri:
    - \* *res:string*: Risposta alla richiesta descritta in req;
    - \* *req:string*: Contiene informazioni sulla richiesta HTTP;
  - *caricaJs(req:string, res:string):void*  
Carica il file json nel server e ne genera il codice Javascript restituendo il nome della cartella compressa;  
Parametri:
    - \* *res:string*: Risposta alla richiesta descritta in req;
    - \* *req:string*: Contiene informazioni sulla richiesta HTTP;
  - *caricaJa(req:string, res:string):void*  
Carica il file json nel server e ne genera il codice Java restituendo il nome della cartella compressa;  
Parametri:
    - \* *res:string*: Risposta alla richiesta descritta in req;
    - \* *req:string*: Contiene informazioni sulla richiesta HTTP;
- **Relazioni con le altre classi:**
  - IN *SWEDesigner::Client::Model::RequestHandler*

## 4 Tracciamenti

### 4.1 Tracciamento classi - requisiti

Nome classe	Codice Requisito
SWEDesigner::Client::Model::DataManager	R0F14.2 R0F14.3.1 R0F14.8 R0F14.8.1 R0F14.8.2
SWEDesigner::Client::Model::DataManager	R0F14 R0F14.1 R0F14.1.1 R0F14.2 R0F14.3 R0F14.3.1 R0F14.8 R0F14.8.1 R0F14.8.2 R0F5 R0F6 R1V3
SWEDesigner::Client::Model::ToolbarModel	R0F11 R0F15 R0F15.1 R0F15.10 R0F15.11 R0F15.4 R0F15.4.1 R0F15.4.2 R0F15.4.3 R0F15.4.4 R0F15.4.5 R0F15.4.7 R0F15.7 R0F16.8 R0F17.8 R0F20 R0F20.1

Nome classe	Codice Requisito
	R0F20.4 R0F4 R0F4.1 R1F18 R1F18.1 R1F18.5 R1F8
SWEDesigner::Client::Model::Project	R0F14 R0F15 R0F20 R0F5 R0F6 R1F18
SWEDesigner::Client::Model::ProjectModel	R0F11 R0F15 R0F15.1 R0F15.10 R0F15.11 R0F15.13 R0F15.16 R0F15.3 R0F15.4 R0F15.4.1 R0F15.4.2 R0F15.4.3 R0F15.4.4 R0F15.4.5 R0F15.4.7 R0F15.5 R0F15.5.1 R0F15.5.2 R0F15.5.3 R0F15.5.4 R0F15.5.5 R0F15.6 R0F15.7 R0F15.9

Nome classe	Codice Requisito
	R0F16.8 R0F16.9 R0F17.7 R0F17.8 R0F2 R0F20 R0F20.1 R0F20.3 R0F20.4 R0F20.6 R0F20.8 R0F4 R0F4.1 R1F18 R1F18.1 R1F18.2.4 R1F18.3 R1F18.5 R1F18.6 R1F18.7 R1F8
SWEDesigner::Client::Model::Items::Package	R1F18 R1F18.1 R1F18.2 R1F18.2.1 R1F18.3 R1F18.7
SWEDesigner::Client::Model::Items::PkgComment	R1F18 R1F18.7
SWEDesigner::Client::Model::Items::PkgCommentLink	R1F18 R1F18.7
SWEDesigner::Client::Model::Items::PkgDependencies	R1F18 R1F18.5 R1F18.6 R1F18.7
SWEDesigner::Client::Model::Items::Class	R0F15.1 R0F15.16

Nome classe	Codice Requisito
	R0F15.2
	R0F15.2.1
	R0F15.2.3
	R0F15.2.4
	R0F15.2.4.1
	R0F15.2.4.2
	R0F15.2.4.3
	R0F15.3
	R0F15.8.2.1
	R0F15.8.2.2
	R0F15.8.2.3
	R0F16.1
	R0F16.10
	R0F16.10.1
	R0F16.10.2
	R0F16.10.3
	R0F16.10.4
	R0F16.10.5
	R0F16.10.6
	R0F16.2
	R0F16.2.2
	R0F16.2.3
	R0F16.2.4
	R0F16.3
	R0F16.4
	R0F16.5
	R0F16.5.1
	R0F16.5.2
	R0F16.5.3
	R0F16.5.3.1
	R0F16.5.3.2
	R0F16.5.3.2.1
	R0F16.5.3.2.2
	R0F16.5.3.2.3
	R0F16.5.3.2.4
	R0F16.5.3.3
	R0F16.5.4
	R0F16.6

Nome classe	Codice Requisito
	R0F16.7
SWEDesigner::Client::Model::Items::Interface	R0F15.16 R0F15.4.7 R0F15.5.7 R0F15.7 R0F15.8 R0F15.8.1 R0F15.8.2 R0F15.8.3 R0F15.9 R0F17 R0F17.1 R0F17.2 R0F17.2.1 R0F17.2.2 R0F17.2.3 R0F17.2.3.1 R0F17.2.3.2 R0F17.2.3.2.1 R0F17.2.3.2.2 R0F17.2.3.2.3 R0F17.2.3.2.4 R0F17.2.3.3 R0F17.2.4 R0F17.2.5 R0F17.3 R0F17.4 R0F17.5 R0F17.6
SWEDesigner::Client::Model::Items::ClComment	R0F15.10 R0F15.12 R0F15.13 R0F15.16
SWEDesigner::Client::Model::Items::ClCommentLink	R0F15.11 R0F15.16 R0F16.8
SWEDesigner::Client::Model::Items::Generalization	R0F15.16

Nome classe	Codice Requisito
	R0F15.4.1 R0F15.5.1
SWEDesigner::Client::Model::Items::Association	R0F15.16 R0F15.4.2 R0F15.5.2
SWEDesigner::Client::Model::Items::Implementation	R0F15.16 R0F15.4.1 R0F15.5.1
SWEDesigner::Client::Model::Items::Aggregation	R0F15.16 R0F15.4.4 R0F15.5.4
SWEDesigner::Client::Model::Items::Composition	R0F15.16 R0F15.4.5 R0F15.5.5
SWEDesigner::Client::Model::Items::bubbleLink	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::Model::Items::customBubble	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::Model::Items::bubbleIf	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.4 R0F20.5 R0F20.6 R0F20.8
SWEDesigner::Client::Model::Items::bubbleElse	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8



Nome classe	Codice Requisito
SWEDesigner::Client::Model::Items::bubbleFor	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::Model::Items::bubbleReturn	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::Model::Items::bubbleStart	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::Model::Items::bubbleWhile	R0F20 R0F20.1 R0F20.2 R0F20.3 R0F20.8
SWEDesigner::Client::View::TitlebarView	R0F14 R0F14.1 R0F14.1.1 R0F14.2 R0F14.3 R0F14.3.1 R0F14.8 R0F14.8.1 R0F14.8.2 R0F5 R0F6 R1V3
SWEDesigner::Client::View::ToolbarView	R0F11 R0F15 R0F15.1 R0F15.10 R0F15.11

Nome classe	Codice Requisito
	R0F15.4 R0F15.4.1 R0F15.4.2 R0F15.4.3 R0F15.4.4 R0F15.4.5 R0F15.4.7 R0F15.7 R0F16.8 R0F17.8 R0F20 R0F20.1 R0F20.4 R0F4 R0F4.1 R1F18 R1F18.1 R1F18.5 R1F8
SWEDesigner::Client::View::PathView	R0F15.15 R0F15.2.5 R0F16.9 R0F17.7 R1F18.4
SWEDesigner::Client::View::EditPanelView	R0F15.12 R0F15.2 R0F15.2.1 R0F15.2.3 R0F15.2.4 R0F15.2.4.1 R0F15.2.4.2 R0F15.2.4.3 R0F15.5 R0F15.5.7 R0F15.8 R0F15.8.1 R0F15.8.2

Nome classe	Codice Requisito
	R0F15.8.2.1
	R0F15.8.2.2
	R0F15.8.2.3
	R0F15.8.3
	R0F16
	R0F16.1
	R0F16.10
	R0F16.10.1
	R0F16.10.2
	R0F16.10.3
	R0F16.10.4
	R0F16.10.5
	R0F16.10.6
	R0F16.2
	R0F16.2.2
	R0F16.2.3
	R0F16.2.4
	R0F16.3
	R0F16.4
	R0F16.5
	R0F16.5.1
	R0F16.5.2
	R0F16.5.3
	R0F16.5.3.1
	R0F16.5.3.2
	R0F16.5.3.2.1
	R0F16.5.3.2.2
	R0F16.5.3.2.3
	R0F16.5.3.2.4
	R0F16.5.3.3
	R0F16.5.4
	R0F16.6
	R0F16.7
	R0F17
	R0F17.1
	R0F17.2
	R0F17.2.1
	R0F17.2.2

Nome classe	Codice Requisito
	R0F17.2.3 R0F17.2.3.1 R0F17.2.3.2 R0F17.2.3.2.1 R0F17.2.3.2.2 R0F17.2.3.2.3 R0F17.2.3.2.4 R0F17.2.3.3 R0F17.2.4 R0F17.2.5 R0F17.2.6 R0F17.3 R0F17.4 R0F17.5 R0F17.6 R0F20.2 R0F20.5 R1F18.2 R1F18.2.1 R1F18.2.3
SWEDesigner::Client::View::ProjectView	R0F11 R0F15 R0F15.1 R0F15.10 R0F15.11 R0F15.13 R0F15.16 R0F15.3 R0F15.4 R0F15.4.1 R0F15.4.2 R0F15.4.3 R0F15.4.4 R0F15.4.5 R0F15.4.7 R0F15.5 R0F15.5.1

Nome classe	Codice Requisito
	R0F15.5.2 R0F15.5.3 R0F15.5.4 R0F15.5.5 R0F15.6 R0F15.7 R0F15.9 R0F16.8 R0F16.9 R0F17.7 R0F17.8 R0F2 R0F20 R0F20.1 R0F20.3 R0F20.4 R0F20.6 R0F20.8 R0F4 R0F4.1 R1F18 R1F18.1 R1F18.2.4 R1F18.3 R1F18.5 R1F18.6 R1F18.7 R1F8
SWEDesigner::Server::CodeGenerator::CodeGenerator	R0F5 R0F6
SWEDesigner::Server::CodeGenerator::Coder::JavaCoder	R0F5
SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder	R0F6
SWEDesigner::Server::CodeGenerator::Coder::Coder	R0F5 R0F6
SWEDesigner::Server::RequestHandler::RequestHandler	R0F14.7

**Tabella 2:** Tracciamento Classi-Requisiti

## 4.2 Tracciamento requisiti - classi

Codice Requisito	Nome classe
R0F11	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F14	SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.1	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.1.1	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.2	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.3	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.3.1	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.7	SWEDesigner::Server::RequestHandler::RequestHandler
R0F14.8	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.8.1	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F14.8.2	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView
R0F15	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.1	SWEDesigner::Client::Model::ProjectModel

Codice Requisito	Nome classe
	SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.10	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::ClComment SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.11	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::ClCommentLink SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.12	SWEDesigner::Client::Model::Items::ClComment SWEDesigner::Client::View::EditPanelView
R0F15.13	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::ClComment SWEDesigner::Client::View::ProjectView
R0F15.15	SWEDesigner::Client::View::PathView
R0F15.16	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::ClCommentLink SWEDesigner::Client::Model::Items::Composition SWEDesigner::Client::Model::Items::ClComment SWEDesigner::Client::Model::Items::Aggregation SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::Model::Items::Implementation SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::Model::Items::Association SWEDesigner::Client::Model::Items::Generalization SWEDesigner::Client::View::ProjectView
R0F15.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView

Codice Requisito	Nome classe
R0F15.2.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.4.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.4.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.4.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.2.5	SWEDesigner::Client::View::PathView
R0F15.3	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::ProjectView
R0F15.4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.4.1	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::Implementation SWEDesigner::Client::Model::Items::Generalization SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.4.2	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::Association SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.4.3	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.4.4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::Aggregation SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F15.4.5	SWEDesigner::Client::Model::ProjectModel



Codice Requisito	Nome classe
	SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::Composition SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.4.7	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.5	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::EditPanelView
R0F15.5.1	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Implementation SWEDesigner::Client::Model::Items::Generalization SWEDesigner::Client::View::ProjectView
R0F15.5.2	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Association SWEDesigner::Client::View::ProjectView
R0F15.5.3	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::ProjectView
R0F15.5.4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Aggregation SWEDesigner::Client::View::ProjectView
R0F15.5.5	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Composition SWEDesigner::Client::View::ProjectView
R0F15.5.7	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F15.6	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::ProjectView
R0F15.7	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F15.8	SWEDesigner::Client::Model::Items::Interface

Codice Requisito	Nome classe
	SWEDesigner::Client::View::EditPanelView
R0F15.8.1	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F15.8.2	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F15.8.2.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.8.2.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.8.2.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F15.8.3	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F15.9	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::ProjectView
R0F16	SWEDesigner::Client::View::EditPanelView
R0F16.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.5	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.10.6	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.2.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView

Codice Requisito	Nome classe
R0F16.2.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.2.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.2.1	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.2.2	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.2.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.2.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.3.3	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.5.4	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.6	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.7	SWEDesigner::Client::Model::Items::Class SWEDesigner::Client::View::EditPanelView
R0F16.8	SWEDesigner::Client::Model::ProjectModel

Codice Requisito	Nome classe
	SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::ClCommentLink SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R0F16.9	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::PathView
R0F17	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.1	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.1	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.2	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.1	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.2	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.2.1	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.2.2	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.2.3	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.2.4	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.3.3	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.4	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.2.5	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView

Codice Requisito	Nome classe
R0F17.2.6	SWEDesigner::Client::View::EditPanelView
R0F17.3	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.4	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.5	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.6	SWEDesigner::Client::Model::Items::Interface SWEDesigner::Client::View::EditPanelView
R0F17.7	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::Project View SWEDesigner::Client::View::PathView
R0F17.8	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::Project View SWEDesigner::Client::View::ToolbarView
R0F2	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::Project View
R0F20	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::bubbleLink SWEDesigner::Client::Model::Items::bubbleReturn SWEDesigner::Client::Model::Items::bubbleFor SWEDesigner::Client::Model::Items::bubbleElse SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::Model::Items::bubbleWhile SWEDesigner::Client::Model::Items::customBubble SWEDesigner::Client::Model::Items::bubbleStart SWEDesigner::Client::View::Project View SWEDesigner::Client::View::ToolbarView
R0F20.1	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::customBubble SWEDesigner::Client::Model::Items::bubbleStart SWEDesigner::Client::Model::Items::bubbleLink SWEDesigner::Client::Model::Items::bubbleReturn

Codice Requisito	Nome classe
	SWEDesigner::Client::Model::Items::bubbleFor SWEDesigner::Client::Model::Items::bubbleElse SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::Model::Items::bubbleWhile SWEDesigner::Client::View::Project View SWEDesigner::Client::View::ToolBar View
R0F20.2	SWEDesigner::Client::Model::Items::bubbleFor SWEDesigner::Client::Model::Items::bubbleElse SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::Model::Items::bubbleWhile SWEDesigner::Client::Model::Items::customBubble SWEDesigner::Client::Model::Items::bubbleStart SWEDesigner::Client::Model::Items::bubbleLink SWEDesigner::Client::Model::Items::bubbleReturn SWEDesigner::Client::View::EditPanelView
R0F20.3	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::bubbleElse SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::Model::Items::bubbleWhile SWEDesigner::Client::Model::Items::customBubble SWEDesigner::Client::Model::Items::bubbleStart SWEDesigner::Client::Model::Items::bubbleLink SWEDesigner::Client::Model::Items::bubbleReturn SWEDesigner::Client::Model::Items::bubbleFor SWEDesigner::Client::View::Project View
R0F20.4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::View::Project View SWEDesigner::Client::View::ToolBar View
R0F20.5	SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::View::EditPanelView
R0F20.6	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::View::Project View
R0F20.8	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::bubbleElse

Codice Requisito	Nome classe
	SWEDesigner::Client::Model::Items::bubbleIf SWEDesigner::Client::Model::Items::bubbleWhile SWEDesigner::Client::Model::Items::customBubble SWEDesigner::Client::Model::Items::bubbleStart SWEDesigner::Client::Model::Items::bubbleLink SWEDesigner::Client::Model::Items::bubbleReturn SWEDesigner::Client::Model::Items::bubbleFor SWEDesigner::Client::View::ProjectView
R0F4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F4.1	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView
R0F5	SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView SWEDesigner::Server::CodeGenerator::CodeGenerator SWEDesigner::Server::CodeGenerator::Coder::Coder SWEDesigner::Server::CodeGenerator::Coder::JavaCoder
R0F6	SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView SWEDesigner::Server::CodeGenerator::CodeGenerator SWEDesigner::Server::CodeGenerator::Coder::Coder SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder
R1F18	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Project SWEDesigner::Client::Model::ToolbarModel SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::Model::Items::PkgDependencies SWEDesigner::Client::Model::Items::PkgCommentLink SWEDesigner::Client::Model::Items::PkgComment SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolbarView

Codice Requisito	Nome classe
R1F18.1	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R1F18.2	SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::View::EditPanelView
R1F18.2.1	SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::View::EditPanelView
R1F18.2.3	SWEDesigner::Client::View::EditPanelView
R1F18.2.4	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::View::ProjectView
R1F18.3	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::View::ProjectView
R1F18.4	SWEDesigner::Client::View::PathView
R1F18.5	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::Model::Items::PkgDependencies SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R1F18.6	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::PkgDependencies SWEDesigner::Client::View::ProjectView
R1F18.7	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::Items::PkgDependencies SWEDesigner::Client::Model::Items::PkgCommentLink SWEDesigner::Client::Model::Items::PkgComment SWEDesigner::Client::Model::Items::Package SWEDesigner::Client::View::ProjectView
R1F8	SWEDesigner::Client::Model::ProjectModel SWEDesigner::Client::Model::ToolBarModel SWEDesigner::Client::View::ProjectView SWEDesigner::Client::View::ToolBarView
R1V3	SWEDesigner::Client::Model::DataManager SWEDesigner::Client::View::TitlebarView

**Tabella 3:** Tracciamento Requisiti-Classi