



Obelix Group  
obelixswe@gmail.com

# Definizione di Prodotto

<b>Versione</b>	<i>v2_0_0</i>
<b>Data creazione</b>	2017-05-01
<b>Redattori</b>	Tomas Mali Emanuele Crespan Riccardo Saggese
<b>Verificatori</b>	Silvio Meneguzzo Federica Schifano
<b>Approvazione</b>	Nicolò Rigato
<b>Stato</b>	Approvato
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Obelix

## Sommario

Questo documento descrive l'architettura generale e di dettaglio del prodotto Monolith che verrà sviluppato dal gruppo Obelix.



## Diario delle revisioni

Versione	Modifica	Autore e Ruolo	Data
2.0.0	Approvazione documento	Nicolò Rigato Responsabile	2017-08-04
1.2.1	Modifica tracciamento	Tomas Mali Progettista	2017-08-03
1.2.0	Verifica sezioni "Diagrammi di attività" e "Diagrammi di sequenza"	Federica Schifano Verificatore	2017-08-02
1.1.2	Modifica sezione 6 "Diagrammi di sequenza"	Emanuele Crespan Progettista	2017-07-31
1.1.1	Modifica sezione 5 "Diagrammi di attività"	Riccardo Saggese Progettista	2017-07-25
1.1.0	Verifica sezione 3 "Descrizione Architettura"	Silvio Meneguzzo Verificatore	2017-07-23
1.0.10	Modifica sottosezione 3.5 "Architettura di dettaglio-classi delle bolle Demo"	Tomas Mali Progettista	2017-07-22
1.0.9	Modifica sottosezione 3.5 "Architettura di dettaglio-classi delle bolle Demo"	Riccardo Saggese Progettista	2017-07-21
1.0.8	Modifica sottosezione 3.3 "Architettura generale-Bolle Demo"	Tomas Mali Progettista	2017-07-20
1.0.7	Modifica sottosezione 3.3 "Architettura generale-Bolle Demo"	Riccardo Saggese Progettista	2017-07-18
1.0.6	Modifica sottosezione 3.4 "Architettura di Dettaglio-Classi del sistema Monolith"	Emanuele Crespan Progettista	2017-07-18
1.0.5	Modifica sottosezione 3.4 "Architettura di Dettaglio-Classi del sistema Monolith"	Tomas Mali Progettista	2017-07-18
1.0.4	Modifica sottosezione 3.2 "Architettura generale-Componenti del sistema"	Emanuele Crespan Progettista	2017-07-17



Versione	Modifica	Autore e Ruolo	Data
1.0.3	Modifica sottosezione 3.2 "Architettura generale-Componenti del sistema"	Tomas Mali Progettista	2017-07-16
1.0.2	Modifica sottosezione 3.2 "Architettura generale-Componenti del sistema"	Emanuele Crespan Progettista	2017-07-15
1.0.1	Modifica sottosezione 2.7 "Node.js"	Riccardo Saggese Progettista	2017-07-01
1.0.0	Approvazione documento	Riccardo Saggese Responsabile	2017-06-19
0.16.3	Aggiunta Appendice	Nicolò Rigato Progettista	2017-06-18
0.16.2	Aggiunta tracciamento Classi-Requisiti e Requisiti-Classi	Federica Schifano Progettista	2017-06-18
0.16.1	Aggiunta tracciamento Componenti-Requisiti e Requisiti-Componenti	Federica Schifano Progettista	2017-06-18
0.16.0	Verifica sezione Diagrammi di sequenza	Silvio Meneguzzo Verificatore	2017-06-17
0.0.10	Stesura sottosezione Voto sondaggio	Emanuele Crespan Progettista	2017-06-17
0.15.3	Stesura sottosezione Cancellazione bolla voto sondaggio	Riccardo Saggese Progettista	2017-06-17
0.15.2	Stesura sottosezione Creazione bolla	Nicolò Rigato Progettista	2017-06-17
0.15.1	Inizio stesura sezione Diagrammi di sequenza	Riccardo Saggese Progettista	2017-06-17
0.15.0	Verifica classi componenti MeteoBubble, SurveyBubble, TranslationBubble	Silvio Meneguzzo Verificatore	2017-06-16
0.14.0	Verifica classi componenti CurrencyBubble, DiceBubble, ListBubble	Nicolò Rigato Verificatore	2017-06-16
0.13.6	Descrizione classi componente TranslationBubble	Federica Schifano Progettista	2017-06-16



Versione	Modifica	Autore e Ruolo	Data
0.13.5	Descrizione classi componente SurveyBubble	Nicolò Rigato Progettista	2017-06-16
0.13.4	Descrizione classi componente MeteoBubble	Emanuele Crespan Progettista	2017-06-15
0.13.3	Descrizione classi componente ListBubble	Silvio Meneguzzo Progettista	2017-06-15
0.13.2	Descrizione classi componente DiceBubble	Tomas Mali Progettista	2017-06-14
0.13.1	Descrizione classi componente CurrencyBubble	Tomas Mali Progettista	2017-06-13
0.13.0	Verifica componenti Sideareas, UI-Layouts, UI-SingleComponents	Silvio Meneguzzo Verificatore	2017-06-11
0.12.3	Descrizione classi componente UI-SingleComponents	Tomas Mali Progettista	2017-06-10
0.12.2	Descrizione classi componente UI-Layouts	Riccardo Saggese Progettista	2017-06-06
0.12.1	Descrizione classi componente Sideareas	Federica Schifano Progettista	2017-06-06
0.12.0	Verifica componenti Checks e Bubbles	Emanuele Crespan Verificatore	2017-06-05
0.11.4	Fine descrizione classi componente Bubbles	Nicolò Rigato Progettista	2017-05-29
0.11.3	Fine descrizione classi componente Bubbles	Nicolò Rigato Progettista	2017-05-29
0.11.2	Inizio descrizione classi componente Bubbles	Nicolò Rigato Progettista	2017-05-27
0.11.1	Descrizione classi componente Checks	Silvio Meneguzzo Progettista	2017-05-26
0.11.0	Verifica componenti TranslationBubble e ListBubble	Riccardo Saggese Verificatore	2017-05-24
0.10.0	Verifica componenti Currency, Dice, Meteo, Survey Bubble	Federica Schifano Verificatore	2017-05-23



Versione	Modifica	Autore e Ruolo	Data
0.9.12	Descrizione componente List-Bubble::CheckListReading	Emanuele Crespan Progettista	2017-05-23
0.9.11	Descrizione componente List-Bubble::CheckListCreation	Federica Schifano Progettista	2017-05-23
0.9.10	Descrizione componente List-Bubble::DataManagement	Tomas Mali Progettista	2017-05-22
0.9.9	Descrizione componente ListBubble::Configuration	Tomas Mali Progettista	2017-05-21
0.9.8	Descrizione componente ListBubble::Receiver	Nicolò Rigato Progettista	2017-05-20
0.9.7	Descrizione componente ListBubble::Sender	Nicolò Rigato Progettista	2017-05-20
0.9.6	Descrizione componente ListBubble	Nicolò Rigato Progettista	2017-05-20
0.9.5	Descrizione componente TranslationBubble	Federica Schifano Progettista	2017-05-19
0.9.4	Descrizione componente SurveyBubble	Riccardo Saggese Progettista	2017-05-18
0.9.3	Descrizione componente MeteoBubble	Silvio Meneguzzo Progettista	2017-05-17
0.9.2	Descrizione componente DiceBubble	Emanuele Crespan Progettista	2017-05-17
0.9.1	Descrizione componente CurrencyBubble	Tomas Mali Progettista	2017-05-17
0.9.0	Verifica componente UI-SingeComponents	Federica Schifano Verificatore	2017-05-15
0.8.0	Verifica componente UI-Layouts	Federica Schifano Verificatore	2017-05-15
0.7.0	Verifica componente SideAreas	Riccardo Saggese Verificatore	2017-05-15
0.6.0	Verifica componente Bubbles	Riccardo Saggese Verificatore	2017-05-15
0.5.4	Descrizione componente UI-SingeComponents	Silvio Meneguzzo Progettista	2017-05-15
0.5.3	Descrizione componente UI-Layouts	Nicolò Rigato Progettista	2017-05-14



Versione	Modifica	Autore e Ruolo	Data
0.5.2	Descrizione componente SideAreas	Emanuele Crespan Progettista	2017-05-14
0.5.1	Descrizione componente Bubbles	Tomas Mali Progettista	2017-05-13
0.5.0	Verifica componente Ui	Federica Schifano Verificatore	2017-05-13
0.4.0	Verifica componenti Monolith e Database	Riccardo Saggese Verificatore	2017-05-13
0.3.7	Descrizione componente Ui	Riccardo Saggese Progettista	2017-05-12
0.3.6	Descrizione componente DatabaseSettings	Federica Schifano Progettista	2017-05-12
0.3.5	Descrizione componente Checks	Silvio Meneguzzo Progettista	2017-05-12
0.3.4	Descrizione componente Information Storage	Emanuele Crespan Progettista	2017-05-11
0.3.3	Descrizione componente Database	Federica Schifano Progettista	2017-05-11
0.3.2	Descrizione componente Monolith	Nicolò Rigato Progettista	2017-05-10
0.3.1	Stesura sezione Metodo e Formalismo di specifica	Tomas Mali Progettista	2017-05-09
0.3.0	Verifica sezione Diagrammi Attività	Riccardo Saggese Verificatore	2017-05-09
0.2.0	Verifica sezione Standard di Progetto	Federica Schifano Verificatore	2017-05-08
0.1.0	Verifica prime tre sezioni	Riccardo Saggese Verificatore	2017-05-08
0.0.6	Stesura sottosezione Configurazione ListBubble	Silvio Meneguzzo Progettista	2017-05-07
0.0.5	Stesura sottosezione Configurazione sondaggio	Federica Schifano Progettista	2017-05-06
0.0.4	Inizio stesura sezione Diagrammi Attività	Federica Schifano Progettista	2017-05-06
0.0.3	Stesura sezione Standard di Progetto	Riccardo Saggese Progettista	2017-05-05



Versione	Modifica	Autore e Ruolo	Data
0.0.4	Inizio stesura sezione Descrizione Architetture	Silvio Meneguzzo Progettista	2017-05-04
0.0.3	Stesura sezione Tecnologie Utilizzate	Emanuele Crespan Progettista	2017-05-03
0.0.2	Stesura sezione Introduzione	Federica Schifano Progettista	2017-05-02
0.0.1	Creto template	Nicolò Rigato Progettista	2017-05-01



## Indice

<b>1</b>	<b>Introduzione</b>	<b>14</b>
1.1	Scopo del documento . . . . .	14
1.2	Scopo del prodotto . . . . .	14
1.3	Glossario . . . . .	14
1.4	Riferimenti . . . . .	14
1.4.1	Normativi . . . . .	14
1.4.2	Informativi . . . . .	14
<b>2</b>	<b>Tecnologie Utilizzate</b>	<b>15</b>
2.1	Javascript 6th edition (ECMA SCRIPT 6) . . . . .	15
2.2	Meteor . . . . .	15
2.3	Mongo DB . . . . .	16
2.4	HTML5 . . . . .	17
2.5	SCSS . . . . .	17
2.6	React . . . . .	17
2.7	Node.js . . . . .	18
2.8	Rocket.chat . . . . .	19
2.9	Bootstrap . . . . .	19
2.10	polyglot.js . . . . .	20
2.11	Money.js . . . . .	20
2.12	weather.js . . . . .	21
2.13	classNames . . . . .	21
<b>3</b>	<b>Descrizione Architettura</b>	<b>21</b>
3.1	Metodo e formalismo di specifica . . . . .	21
3.2	Architettura generale - Componenti del sistema . . . . .	22
3.2.1	Monolith . . . . .	22
3.2.2	Monolith::Database . . . . .	23
3.2.3	Monolith::Database::InformationStorage . . . . .	24
3.2.4	Monolith::Database::informationStorage::Checks . . . . .	25
3.2.5	Monolith::Database::InformationStorage::DatabaseSettings . . . . .	26
3.2.6	Monolith::UI . . . . .	27
3.2.7	Monolith::UI::Bubbles . . . . .	28
3.2.8	Monolith::UI::SideAreas . . . . .	30
3.2.9	Monolith::UI::SideAreas::SideArea1_pkg . . . . .	31
3.2.10	Monolith::UI::SideAreas::SideArea2_pkg . . . . .	32
3.2.11	Monolith::UI::UI-Layouts . . . . .	33
3.2.12	Monolith::UI::UI-SingleComponents . . . . .	34
3.3	Architettura generale - Bolle Demo . . . . .	36
3.3.1	CurrencyBubble . . . . .	36
3.3.2	DiceBubble . . . . .	38
3.3.3	ListBubble . . . . .	40
3.3.4	ListBubble::CheckListCreation . . . . .	41
3.3.5	ListBubble::CheckListReading . . . . .	43
3.3.6	ListBubble::Configuration . . . . .	44
3.3.7	ListBubble::DataManagement . . . . .	45
3.3.8	ListBubble::Receiver . . . . .	46
3.3.9	ListBubble::Sender . . . . .	47





3.3.10	MeteoBubble . . . . .	48
3.3.11	SurveyBubble . . . . .	50
3.3.12	TranslationBubble . . . . .	52
3.4	Architettura di dettaglio - Classi del sistema Monolith . . . . .	54
3.4.1	check . . . . .	54
3.4.2	checkCreator . . . . .	55
3.4.3	concreteCheck . . . . .	56
3.4.4	checkDiscriminator . . . . .	57
3.4.5	concreteCheckCreator . . . . .	58
3.4.6	Bubble . . . . .	59
3.4.7	BubbleConfig . . . . .	60
3.4.8	bubbleCreator . . . . .	61
3.4.9	ConcreteBubble . . . . .	62
3.4.10	bubbleDiscriminator . . . . .	63
3.4.11	BubbleCreationButton . . . . .	64
3.4.12	ConcreteBubbleConfig . . . . .	65
3.4.13	concreteBubbleCreator . . . . .	66
3.4.14	ConcreteBubbleCreationButton . . . . .	67
3.4.15	SideArea1 . . . . .	68
3.4.16	SentBubbleHistory . . . . .	69
3.4.17	BubbleCreationMenu . . . . .	70
3.4.18	SideArea2 . . . . .	71
3.4.19	ReceivedBubbleHistory . . . . .	72
3.4.20	VerticalLayout . . . . .	73
3.4.21	ContainedElement . . . . .	74
3.4.22	HorizontalLayout . . . . .	75
3.4.23	ConditionalRendering . . . . .	76
3.4.24	Image . . . . .	77
3.4.25	ComboBox . . . . .	78
3.4.26	LineEdit . . . . .	79
3.4.27	LabelEdit . . . . .	80
3.4.28	PushButton . . . . .	81
3.4.29	CheckButton . . . . .	82
3.4.30	ImageButton . . . . .	83
3.4.31	CheckBoxList . . . . .	84
3.4.32	LabelComboBox . . . . .	85
3.4.33	TextAreaButton . . . . .	86
3.4.34	LabelPushButton . . . . .	87
3.4.35	LineEditComboBox . . . . .	88
3.4.36	RadioButtonGroup . . . . .	89
3.4.37	TextAreaComboBox . . . . .	90
3.4.38	LineEditPushButton . . . . .	91
3.4.39	LabelEditPushButton . . . . .	92
3.5	Architettura di dettaglio - Classi delle bolle demo . . . . .	92
3.5.1	CurrencyConversion . . . . .	92
3.5.2	CurrencyBubbleSender . . . . .	94
3.5.3	CurrencyBubbleCreator . . . . .	95
3.5.4	CurrencyBubbleReceiver . . . . .	96
3.5.5	CurrencyBubbleConfigMenu . . . . .	97
3.5.6	DiceRoller . . . . .	98



3.5.7	DiceBubbleSender . . . . .	99
3.5.8	DiceBubbleCreator . . . . .	100
3.5.9	DiceBubbleReceiver . . . . .	101
3.5.10	DiceBubbleConfigMenu . . . . .	102
3.5.11	CheckListCreator . . . . .	103
3.5.12	CheckListComponent . . . . .	104
3.5.13	CheckListItemsDefinition . . . . .	105
3.5.14	CheckList . . . . .	106
3.5.15	ListOfCheckLists . . . . .	107
3.5.16	ListCreationButton . . . . .	108
3.5.17	ListBubbleMenuConfig . . . . .	109
3.5.18	ListBubbleCreator . . . . .	110
3.5.19	ListBubbleReceiver . . . . .	111
3.5.20	ListBubbleSender . . . . .	112
3.5.21	MeteoItem . . . . .	113
3.5.22	MeteoDelivery . . . . .	114
3.5.23	MeteoBubbleSender . . . . .	115
3.5.24	MeteoBubbleCreator . . . . .	116
3.5.25	MeteoBubbleReceiver . . . . .	117
3.5.26	MeteoBubbleConfigMenu . . . . .	118
3.5.27	ResultsViewer . . . . .	119
3.5.28	SurveyManager . . . . .	120
3.5.29	SurveyBubbleSender . . . . .	121
3.5.30	SurveyBubbleCreator . . . . .	122
3.5.31	SurveyBubbleReceiver . . . . .	123
3.5.32	SurveyBubbleConfigMenu . . . . .	124
3.5.33	MessageTranslation . . . . .	125
3.5.34	TranslationBubbleSender . . . . .	126
3.5.35	TranslationBubbleCreator . . . . .	127
3.5.36	TranslationBubbleReceiver . . . . .	128
3.5.37	TranslationBubbleConfigMenu . . . . .	129
<b>4</b>	<b>Standard di Progetto</b>	<b>129</b>
4.1	Standard di documentazione del codice . . . . .	129
4.2	Standard di denominazione di entità e relazioni . . . . .	129
4.3	Strumenti di lavoro . . . . .	130
<b>5</b>	<b>Diagrammi di Attività</b>	<b>130</b>
5.1	Configurazione sondaggio . . . . .	130
5.2	Configurazione ListBubble . . . . .	131
<b>6</b>	<b>Diagramma di Sequenza</b>	<b>132</b>
6.1	Creazione Bolla . . . . .	132
6.2	Cancellazione Bolla . . . . .	133
6.3	Voto Bolla-Sondaggio . . . . .	134



<b>7</b>	<b>Tracciamento</b>	<b>134</b>
7.1	Tracciamento componenti-requisiti . . . . .	134
7.2	Tracciamento requisiti-componenti . . . . .	137
7.3	Tracciamento classi-requisiti . . . . .	140
7.4	Tracciamento requisiti-classi . . . . .	148
<b>A</b>	<b>Descrizione Design Pattern</b>	<b>162</b>
A.1	Design Pattern Utilizzati . . . . .	163
A.1.1	Factory Method . . . . .	163
A.1.2	Template Method . . . . .	163
A.1.3	Proxy Pattern . . . . .	164
A.1.4	Composite Pattern . . . . .	165



## Elenco delle figure

1	Diagramma per Monolith. . . . .	22
2	Diagramma per Monolith::Database::InformationStorage. . . . .	24
3	Diagramma per Monolith::Database::informationStorage::Checks. . . . .	25
4	Diagramma per Monolith::UI::Bubbles. . . . .	28
5	Diagramma per Monolith::UI::UI-Layouts. . . . .	33
6	Diagramma per Monolith::UI::UI-SingleComponents. . . . .	34
7	Diagramma per CurrencyBubble. . . . .	36
8	Diagramma per DiceBubble. . . . .	38
9	Diagramma per ListBubble. . . . .	40
10	Diagramma per ListBubble::CheckListCreation. . . . .	41
11	Diagramma per ListBubble::CheckListReading. . . . .	43
12	Diagramma per ListBubble::Configuration. . . . .	44
13	Diagramma per ListBubble::DataManagement. . . . .	45
14	Diagramma per ListBubble::Receiver. . . . .	46
15	Diagramma per ListBubble::Sender. . . . .	47
16	Diagramma per MeteoBubble. . . . .	48
17	Diagramma per SurveyBubble. . . . .	50
18	Diagramma per TranslationBubble. . . . .	52
19	Diagramma per check in Checks . . . . .	54
20	Diagramma per checkCreator in Checks . . . . .	55
21	Diagramma per concreteCheck in Checks . . . . .	56
22	Diagramma per checkDiscriminator in Checks . . . . .	57
23	Diagramma per concreteCheckCreator in Checks . . . . .	58
24	Diagramma per Bubble in Bubbles . . . . .	59
25	Diagramma per BubbleConfig in Bubbles . . . . .	60
26	Diagramma per bubbleCreator in Bubbles . . . . .	61
27	Diagramma per ConcreteBubble in Bubbles . . . . .	62
28	Diagramma per bubbleDiscriminator in Bubbles . . . . .	63
29	Diagramma per BubbleCreationButton in Bubbles . . . . .	64
30	Diagramma per ConcreteBubbleConfig in Bubbles . . . . .	65
31	Diagramma per concreteBubbleCreator in Bubbles . . . . .	66
32	Diagramma per ConcreteBubbleCreationButton in Bubbles . . . . .	67
33	Diagramma per SideArea1 in SideArea1_pkg . . . . .	68
34	Diagramma per SentBubbleHistory in SideArea1_pkg . . . . .	69
35	Diagramma per BubbleCreationMenu in SideArea1_pkg . . . . .	70
36	Diagramma per SideArea2 in SideArea2_pkg . . . . .	71
37	Diagramma per ReceivedBubbleHistory in SideArea2_pkg . . . . .	72
38	Diagramma per VerticalLayout in UI-Layouts . . . . .	73
39	Diagramma per ContainedElement in UI-Layouts . . . . .	74
40	Diagramma per HorizontalLayout in UI-Layouts . . . . .	75
41	Diagramma per ConditionalRendering in UI-Layouts . . . . .	76
42	Diagramma per Image in UI-SingleComponents . . . . .	77
43	Diagramma per ComboBox in UI-SingleComponents . . . . .	78
44	Diagramma per LineEdit in UI-SingleComponents . . . . .	79
45	Diagramma per LabelEdit in UI-SingleComponents . . . . .	80
46	Diagramma per PushButton in UI-SingleComponents . . . . .	81
47	Diagramma per CheckButton in UI-SingleComponents . . . . .	82
48	Diagramma per ImageButton in UI-SingleComponents . . . . .	83



49	Diagramma per CheckBoxList in UI-SingleComponents . . . . .	84
50	Diagramma per LabelComboBox in UI-SingleComponents . . . . .	85
51	Diagramma per TextAreaButton in UI-SingleComponents . . . . .	86
52	Diagramma per LabelPushButton in UI-SingleComponents . . . . .	87
53	Diagramma per LineEditComboBox in UI-SingleComponents . . . . .	88
54	Diagramma per RadioButtonGroup in UI-SingleComponents . . . . .	89
55	Diagramma per TextAreaComboBox in UI-SingleComponents . . . . .	90
56	Diagramma per LineEditPushButton in UI-SingleComponents . . . . .	91
57	Diagramma per LabelEditPushButton in UI-SingleComponents . . . . .	92
58	Diagramma per CurrencyConversion in CurrencyBubble . . . . .	93
59	Diagramma per CurrencyBubbleSender in CurrencyBubble . . . . .	94
60	Diagramma per CurrencyBubbleCreator in CurrencyBubble . . . . .	95
61	Diagramma per CurrencyBubbleReceiver in CurrencyBubble . . . . .	96
62	Diagramma per CurrencyBubbleConfigMenu in CurrencyBubble . . . . .	97
63	Diagramma per DiceRoller in DiceBubble . . . . .	98
64	Diagramma per DiceBubbleSender in DiceBubble . . . . .	99
65	Diagramma per DiceBubbleCreator in DiceBubble . . . . .	100
66	Diagramma per DiceBubbleReceiver in DiceBubble . . . . .	101
67	Diagramma per DiceBubbleConfigMenu in DiceBubble . . . . .	102
68	Diagramma per CheckListCreator in CheckListCreation . . . . .	103
69	Diagramma per CheckListComponent in CheckListCreation . . . . .	104
70	Diagramma per CheckListItemsDefinition in CheckListCreation . . . . .	105
71	Diagramma per CheckList in CheckListReading . . . . .	106
72	Diagramma per ListOfCheckLists in CheckListReading . . . . .	107
73	Diagramma per ListCreationButton in Configuration . . . . .	108
74	Diagramma per ListBubbleMenuConfig in Configuration . . . . .	109
75	Diagramma per ListBubbleCreator in DataManagement . . . . .	110
76	Diagramma per ListBubbleReceiver in Receiver . . . . .	111
77	Diagramma per ListBubbleSender in Sender . . . . .	112
78	Diagramma per MeteoItem in MeteoBubble . . . . .	113
79	Diagramma per MeteoDelivery in MeteoBubble . . . . .	114
80	Diagramma per MeteoBubbleSender in MeteoBubble . . . . .	115
81	Diagramma per MeteoBubbleCreator in MeteoBubble . . . . .	116
82	Diagramma per MeteoBubbleReceiver in MeteoBubble . . . . .	117
83	Diagramma per MeteoBubbleConfigMenu in MeteoBubble . . . . .	118
84	Diagramma per ResultsViewer in SurveyBubble . . . . .	119
85	Diagramma per SurveyManager in SurveyBubble . . . . .	120
86	Diagramma per SurveyBubbleSender in SurveyBubble . . . . .	121
87	Diagramma per SurveyBubbleCreator in SurveyBubble . . . . .	122
88	Diagramma per SurveyBubbleReceiver in SurveyBubble . . . . .	123
89	Diagramma per SurveyBubbleConfigMenu in SurveyBubble . . . . .	124
90	Diagramma per MessageTranslation in TranslationBubble . . . . .	125
91	Diagramma per TranslationBubbleSender in TranslationBubble . . . . .	126
92	Diagramma per TranslationBubbleCreator in TranslationBubble . . . . .	127
93	Diagramma per TranslationBubbleReceiver in TranslationBubble . . . . .	128
94	Diagramma per TranslationBubbleConfigMenu in Translation- Bubble . . . . .	129
95	Diagramma di attività per la bolla sondaggio . . . . .	131
96	Diagramma di attività per la bolla ListBubble . . . . .	132
97	Diagramma di sequenza per la creazione di una bolla . . . . .	132



98	Diagramma di sequenza per la cancellazione di una bolla . . . . .	133
99	Diagramma di sequenza per il voto in una bolla-sondaggio . . . . .	134
100	Diagramma del Factory method . . . . .	163
101	Diagramma del Template method . . . . .	164
102	Diagramma del Proxy Pattern . . . . .	164
103	Diagramma del Composite Pattern . . . . .	165

## Elenco delle tabelle

3	Tracciamento componenti - requisiti . . . . .	137
5	Tracciamento requisiti - componenti . . . . .	140
7	Tracciamento classi - requisiti . . . . .	148
9	Tracciamento requisiti - classi . . . . .	162



## 1 Introduzione

### 1.1 Scopo del documento

Questo documento ha come scopo quello di definire la progettazione ad alto e a basso livello per il prodotto Monolith. Verranno presentate l'architettura generale e di dettaglio secondo le quali saranno organizzate le varie componenti software e i *Design Pattern*<sub>[G]</sub> utilizzati nella creazione dell'SDK, delle bolle predefinite e della demo. Verrà inoltre dettagliato il tracciamento tra le componenti software individuate ed i requisiti, i requisiti e i componenti, le classi e i requisiti e i requisiti e le classi.

### 1.2 Scopo del prodotto

Lo scopo del prodotto è quello di permettere la creazione di bolle interattive, che dovranno funzionare nell'ambiente Rocket.chat. Queste bolle permetteranno di aumentare l'interattività tra gli utenti della chat e aggiungeranno nuove funzionalità accessibili direttamente dalla conversazione senza il bisogno di ricorrere all'apertura di applicazioni diverse. Il sistema offrirà agli sviluppatori un set di *API*<sub>[G]</sub> per creare e rilasciare nuove bolle e agli utenti finali la possibilità di usufruire di un insieme di bolle predefinite.

### 1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini che necessitano di essere chiariti saranno scritti in corsivo e marcati con una [G] in pedice alla prima occorrenza e saranno riportati nel *Glossario v3.0.0*.

### 1.4 Riferimenti

#### 1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v4.0.0*
- **Capitolato d'appalto C5:**  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C5.pdf>
- **Analisi dei Requisiti:** *Analisi dei Requisiti v3.0.0*

#### 1.4.2 Informativi

- **Slide del corso di Ingegneria del Software:**  
<http://www.math.unipd.it/~tullio/IS-1/2016/>
- **Documentazione React**
- **Documentazione Meteor**
- **Documentazione ECMAScript 6**
- **Libro Design Patterns** Design Patterns, Elementi per il riuso di software a oggetti. Gamma, Helm, Johnson, Vlissides.



## 2 Tecnologie Utilizzate

In questa sezione verranno descritte le tecnologie su cui si basa lo sviluppo del progetto. Per ognuna di esse, verranno indicati l'ambito di utilizzo della tecnologia, i vantaggi e gli svantaggi che ne derivano. Alcune delle tecnologie che saranno usate sono richieste come requisito dal capitolato scelto.

### 2.1 Javascript 6th edition (ECMA SCRIPT 6)

JavaScript è un *linguaggio di scripting*<sub>|G|</sub> orientato agli oggetti e agli eventi. È comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite l'uso di funzioni di script invocate da eventi innescati in vari modi dall'utente sulla pagina web in uso.

Come richiesto dal capitolato, per la realizzazione di Monolith, deve essere utilizzato *Javascript*<sub>|G|</sub> 6th edition (ECMA SCRIPT 6).

**Licenza** Non esiste una sola implementazione perché ECMAScript (o ES) è un linguaggio di programmazione standardizzato e mantenuto da Ecma International nell'ECMA-262 ed ISO/IEC 16262.

#### Vantaggi

- Gestione degli eventi asincroni tramite le promises
- Possibilità di dichiarare classi
- Supporto per le costanti (*const*)
- Possibilità di isolare la definizione di variabili ad un blocco (*let*)
- Possibilità di isolare lo scope di una funzione usando blocchi delimitati da parentesi graffe {} come ambienti isolati (vs closure)
- Uso di sintassi più espressiva per scrivere le funzioni anonime (*Arrow Functions*)

#### Svantaggi

- Il supporto di ES6 da parte dei browser è ancora incompleto
- L'assenza di tipizzazione potrebbe ostacolare la valutazione della correttezza del codice

### 2.2 Meteor

Meteor è un framework web JavaScript libero e open source per lo sviluppo di applicazioni web e mobile. È una piattaforma basata su Node.js. Meteor utilizza, dunque, JavaScript sia lato client che lato server.





**Licenza** MIT La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

#### Vantaggi

- Integrazione con diverse tecnologie utilizzate nello sviluppo web:
  - React
  - MongoDB
- Isomorfismo: il codice javascript scritto funziona in modo trasparente sul client (browser), sul server (Node.js) o in entrambi
- Ecosistema e modularità: la comunità di Meteor è molto attiva e molte funzionalità client o server potrebbero già essere pacchettizzate dal *package<sub>|G|</sub>* manager ufficiale

#### Svantaggi

- Inizialmente sconosciuto ai membri del gruppo.

### 2.3 Mongo DB

MongoDB è un database NoSQL orientato ai documenti, basato sul formato *BSON<sub>|G|</sub>* per la memorizzazione e la rappresentazione dei dati. È distribuito come software libero open source.

#### Licenza GNU AGPL v3.0

È una licenza pubblicata da Free Software Foundation. È simile alla capostipite GNU GPL, una licenza fortemente copyleft per software libero.

#### Vantaggi

- È più flessibile di un database *SQL<sub>|G|</sub>* e facilita la rappresentazione su un modello ad oggetti
- Supporta ricerche per campi, intervalli e regular expression. Le query possono restituire campi specifici del documento e anche includere funzioni definite dall'utente in JavaScript.
- Qualunque campo in MongoDB può essere indicizzato

#### Svantaggi

- Inizialmente sconosciuto ai membri del gruppo
- Non è possibile eseguire query complesse a causa dell'assenza di schema dei dati. A causa di questo parte della complessità non può essere demandata al sistema di database



## 2.4 HTML5

HTML5 è un linguaggio di markup per la strutturazione delle pagine web.

**Licenza** Non esiste una sola implementazione perché HTML5 è un linguaggio di markup standardizzato e mantenuto da W3C.

### Vantaggi

- Codice più pulito e sintassi semplificata rispetto alle versioni precedenti
- Interattività senza l'ausilio di plugin esterni valida per diversi formati multimediali
- Semantica intuitiva grazie ai nuovi  $TAG_{|G|}$  di formattazione
- Introduzione della geo localizzazione, dovuta ad una forte espansione di sistemi operativi mobili
- Sistema più efficiente alternativo ai normali cookie chiamato Web Storage

### Svantaggi

- Non tutti i browser supportano HTML5

## 2.5 SCSS

SCSS è una sintassi per i fogli di stile introdotta da Sass 3 (Syntactically Awesome StyleSheets). È un'estensione del CSS .

### Licenza MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

### Vantaggi

- Possibilità di utilizzare variabili
- Possibilità di creare funzioni
- Possibilità di organizzare il foglio di stile in più file
- Compatibilità completa con la sintassi del CSS

### Svantaggi

- Sintassi più complessa.

## 2.6 React

React è una libreria Javascript open source che permette di costruire interfacce utente.

**Licenza** BSD-3-Clause

Le licenze BSD sono una famiglia di licenze permissive, senza copyleft, per software. Le tre clausole della licenza BSD-3-Clause sono:

- Libertà di eseguire il programma per qualsiasi scopo
- Libertà di studiare il programma e modificarlo
- Libertà di ridistribuire copie del programma in modo da aiutare il prossimo

**Vantaggi**

- Semplificazione della realizzazione di interfacce UI dinamiche che possono reagire ai cambiamenti di dati in maniera autonoma attraverso opportuni componenti
- Possibilità di utilizzare le viste per creare codice più facile da comprendere e su cui è più semplice effettuare il debugging.

**Svantaggi**

- Implementa solo la parte puramente visuale dell'applicazione. Esistono alternative che possono gestire molti più aspetti.
- Curva di apprendimento ripida
- È una libreria relativamente nuova

## 2.7 Node.js

Node.js è una piattaforma event-driven per il motore JavaScript V8. Essa permette di realizzare applicazioni web utilizzando il linguaggio JavaScript, che tipicamente è usato client-side, per la scrittura anche della parte server-side delle applicazioni web.

**Licenza** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

**Vantaggi**

- Facile apprendimento
- Possibilità di realizzare la parte back-end di applicazioni utilizzando lo stesso linguaggio della parte front-end.

**Svantaggi**

- Non supporta database relazionali



## 2.8 Rocket.chat

Rocket.chat è una Web chat server sviluppata in Javascript utilizzando il *Framework*<sub>[G]</sub> Meteor.

### **Licenza** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

### **Vantaggi**

- Codice open source
- Possibilità di creare chat di gruppo
- Possibilità di inviare audio, video e file
- Possibilità di effettuare video chiamate
- Community molto attiva

### **Svantaggi**

- Parzialmente documentata

## 2.9 Bootstrap

Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.

### **Licenza** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

### **Vantaggi**

- Piattaforma ben standardizzata
- Non richiede l'appoggio né di un linguaggio di programmazione server side, né di un database
- Ottima documentazione
- *Responsive*<sub>[G]</sub> Design
- É supportato dai browser moderni

**Svantaggi**

- I plugin di jQuery sono limitati
- Le modifiche dovute al continuo sviluppo non sono sempre facili da integrare

**2.10 polyglot.js**

Polyglot.js è una libreria per la traduzione scritta in JavaScript, eseguita sia per il browser che per gli ambienti CommonJS(Node).

**Licenza MIT**

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

**Vantaggi**

- Non è richiesta una iscrizione per l'utilizzo
- È un servizio non a pagamento
- Il Polyglot ha zero dipendenze
- Copre una traduzione di 30 lingue diverse

**2.11 Money.js**

Money.js è una libreria semplice con l'unico obiettivo di convertire un valore di denaro da qualsiasi valuta in qualsiasi altra valuta. Money.js utilizza una fusione algoritmica per calcolare un insieme di tassi costantemente preciso per 165 valute mondiali.

**Licenza MIT**

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

**Vantaggi**

- Non è richiesta una iscrizione per l'utilizzo
- È un servizio non a pagamento
- È una libreria semplice da integrare nel codice JavaScript



### 2.12 weather.js

Weather.js è una libreria che recupera i dati da [openweathermap.org](https://openweathermap.org) e fa la ricerca di tutti i tipi di informazioni relative alle condizioni meteo.

**Licenza** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

**Vantaggi**

- Non è richiesta una iscrizione per l'utilizzo
- É un servizio non a pagamento
- É una libreria semplice da integrare nel codice JavaScript

**Svantaggi**

- Ha bisogno di 11 dipendenze

### 2.13 classNames

classNames è una semplice utility raccomandata per l'uso con *React*<sub>|G|</sub> per l'unione condizionata di className

**Licenza** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

**Vantaggi**

- Semplifica la gestione dei className dinamici
- Non possiede ulteriori dipendenze

## 3 Descrizione Architettura

### 3.1 Metodo e formalismo di specifica

Nell'esposizione dell'architettura dell'applicazione si procederà con un approccio top-down, descrivendo l'architettura iniziando dal generale ed andando al particolare. Si procederà quindi alla descrizione dei componenti software, per poi descrivere nel dettaglio le singole classi, specificando per ognuna il tipo, l'obiettivo, la funzione e le relazioni in ingresso ed in uscita. Successivamente si illustreranno degli esempi di uso dei Design Pattern nell'architettura del



sistema. L'architettura dell'  $SDK_{|G|}$  e della  $demo_{|G|}$  sono state progettate separatamente. Per i diagrammi delle componenti di classe e di attività, si utilizza il formalismo  $UML_{|G|}$  2.0. Le classi e componenti presenti in librerie o framework esterni vengono contraddistinte da colori diversi. I framework esterni verranno rappresentati con un colore azzurro, invece le classi e componenti proprie saranno rappresentate con un colore giallo. Nella demo le componenti dell'SDK sono in verde. L'intera applicazione è progettata utilizzando il framework  $Meteor_{|G|}$  che permette di utilizzare il linguaggio JavaScript sia per il lato client che per quello server (tramite NodeJS). I diagrammi delle classi che permettono di mostrare l'architettura generale del sistema vengono affiancati anche dai diagrammi di sequenza e attività, che permettono di definire le interazioni tra le componenti, senza preoccuparsi della loro classificazione.

## 3.2 Architettura generale - Componenti del sistema

### 3.2.1 Monolith

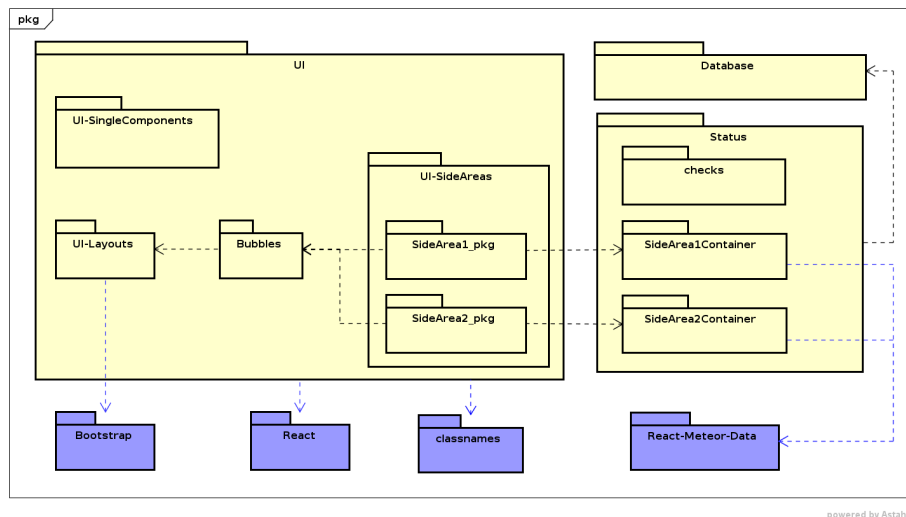


Figura 1: Diagramma per Monolith.

#### Descrizione:

Componente che rappresenta l'intera SDK di Monolith



### 3.2.2 Monolith::Database

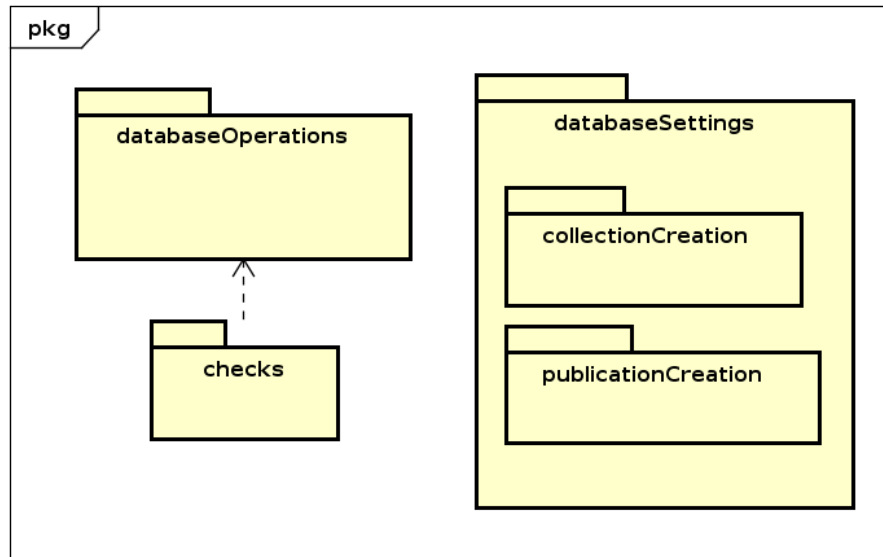
**Descrizione:**

Componente contenente i pacchetti che vengono utilizzati per interagire con il database





### 3.2.3 Monolith::Database::InformationStorage



powered by Astah

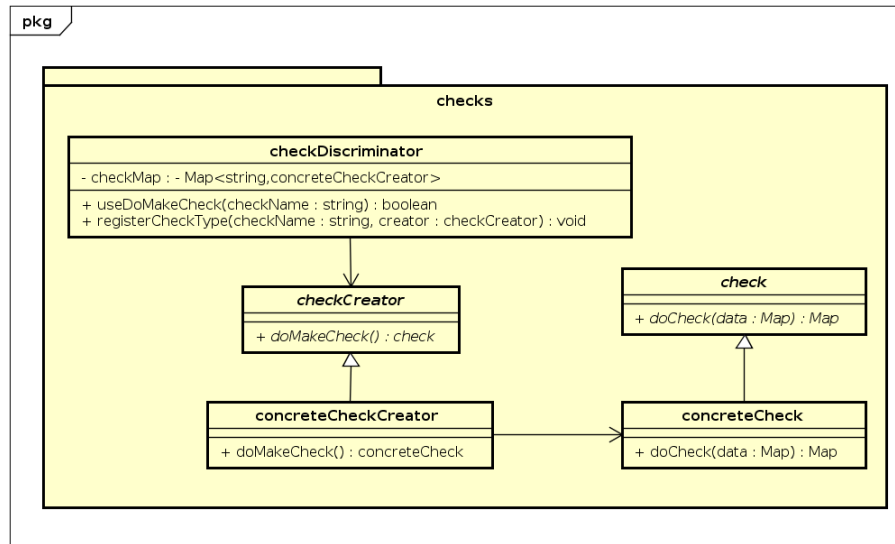
Figura 2: Diagramma per Monolith::Database::InformationStorage.

**Descrizione:**

Componente per la configurazione dell'utilizzo del database



### 3.2.4 Monolith::Database::informationStorage::Checks



powered by Astah

Figura 3: Diagramma per Monolith::Database::informationStorage::Checks.

**Descrizione:**

Componente per la creazione dei controlli da effettuare sul client prima di effettuare l'inserimento dei dati nel database

**Classi contenute:**

- check
- checkCreator
- checkDiscriminator
- concreteCheck
- concreteCheckCreator



### 3.2.5 Monolith::Database::InformationStorage::DatabaseSettings

**Descrizione:**

Modulo per la configurazione delle collection



### 3.2.6 Monolith::UI

**Descrizione:**

Componente contenente tutti i pacchetti che servono per comporre e gestire la parte visuale dell'applicazione delle bolle

**Dipendenze**

- React
- classNames

### 3.2.7 Monolith::UI::Bubbles

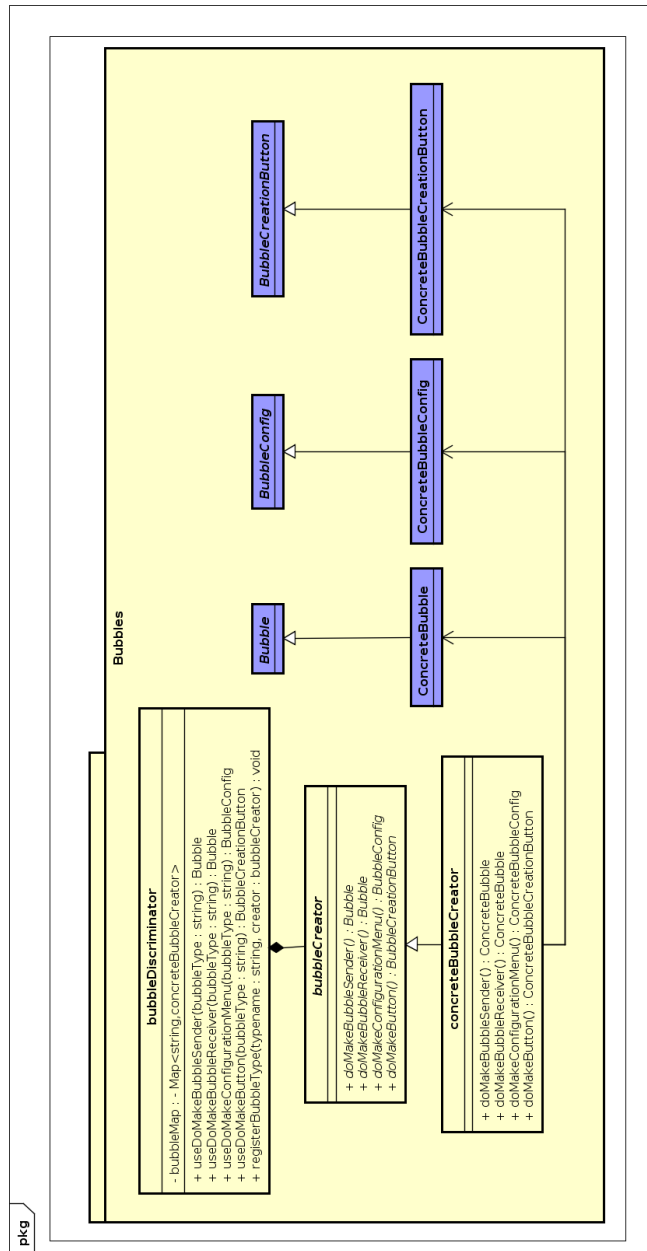


Figura 4: Diagramma per Monolith::UI::Bubbles.

#### Descrizione:

Componente per la creazione delle bolle da visualizzare

Classi contenute:

- Bubble
- BubbleConfig



- BubbleCreationButton
- bubbleCreator
- bubbleDiscriminator
- ConcreteBubble
- ConcreteBubbleConfig
- ConcreteBubbleCreationButton
- concreteBubbleCreator



### 3.2.8 Monolith::UI::SideAreas

**Descrizione:**

Contiene i package per la visualizzazione delle bolle nelle side-bar



### 3.2.9 Monolith::UI::SideAreas::SideArea1\_pkg

**Descrizione:**

Componente per la visualizzazione delle bolle inviate e del menù di creazione delle bolle nella prima side area

**Classi contenute:**

- BubbleCreationMenu
- SentBubbleHistory
- SideArea1





### 3.2.10 Monolith::UI::SideAreas::SideArea2\_pkg

**Descrizione:**

Componente per la visualizzazione delle bolle ricevute nella seconda side area

**Classi contenute:**

- ReceivedBubbleHistory
- SideArea2



### 3.2.11 Monolith::UI::UI-Layouts

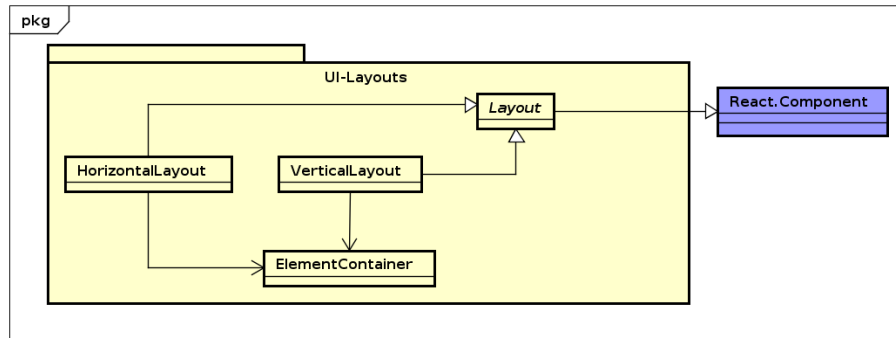


Figura 5: Diagramma per Monolith::UI::UI-Layouts.

**Descrizione:**

Componente che contiene le classi React per la gestione dei layout

**Dipendenze**

Bootstrap

**Classi contenute:**

- ConditionalRendering
- ContainedElement
- HorizontalLayout
- VerticalLayout



### 3.2.12 Monolith::UI::UI-SingleComponents

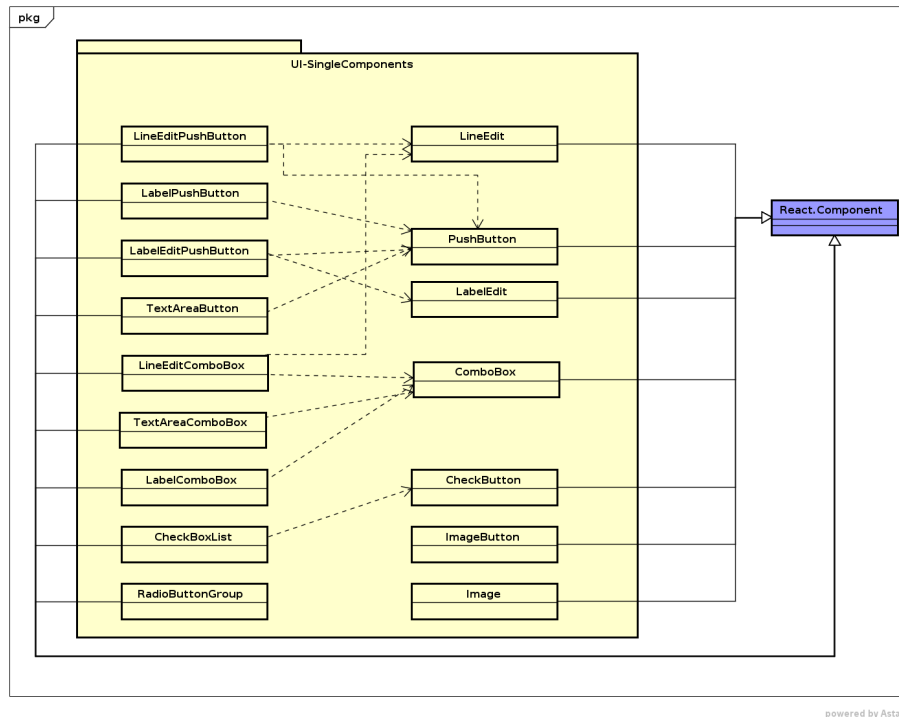


Figura 6: Diagramma per Monolith::UI::UI-SingleComponents.

**Descrizione:**

Componente che contiene tutti i componenti React per la composizione della GUI

**Classi contenute:**

- CheckBoxList
- CheckButton
- ComboBox
- Image
- ImageButton
- LabelComboBox
- LabelEdit
- LabelEditPushButton
- LabelPushButton
- LineEdit



- LineEditComboBox
- LineEditPushButton
- PushButton
- RadioButtonGroup
- TextAreaButton
- TextAreaComboBox



### 3.3 Architettura generale - Bolle Demo

#### 3.3.1 CurrencyBubble

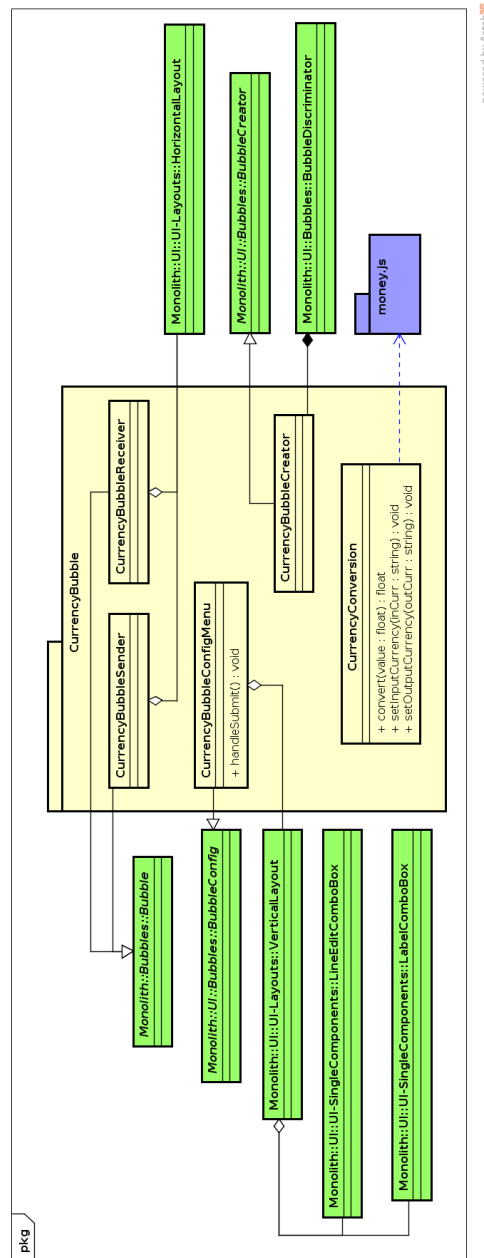


Figura 7: Diagramma per CurrencyBubble.

#### Descrizione:

Componente contenente le classi necessarie per la creazione della bolla convertitore valute.

#### Classi contenute:

- CurrencyBubbleConfigMenu



- CurrencyBubbleCreator
- CurrencyBubbleReceiver
- CurrencyBubbleSender
- CurrencyConversion



### 3.3.2 DiceBubble

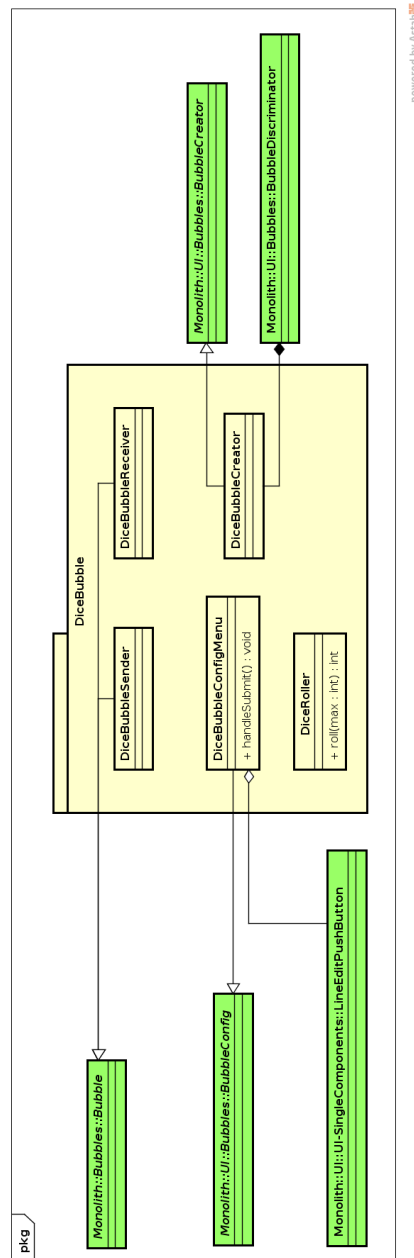


Figura 8: Diagramma per DiceBubble.

**Descrizione:**

Componente contenente le classi necessarie per la creazione della bolla estrazione numero casuale.

**Classi contenute:**

- DiceBubbleConfigMenu
- DiceBubbleCreator



- DiceBubbleReceiver
- DiceBubbleSender
- DiceRoller





### 3.3.3 ListBubble

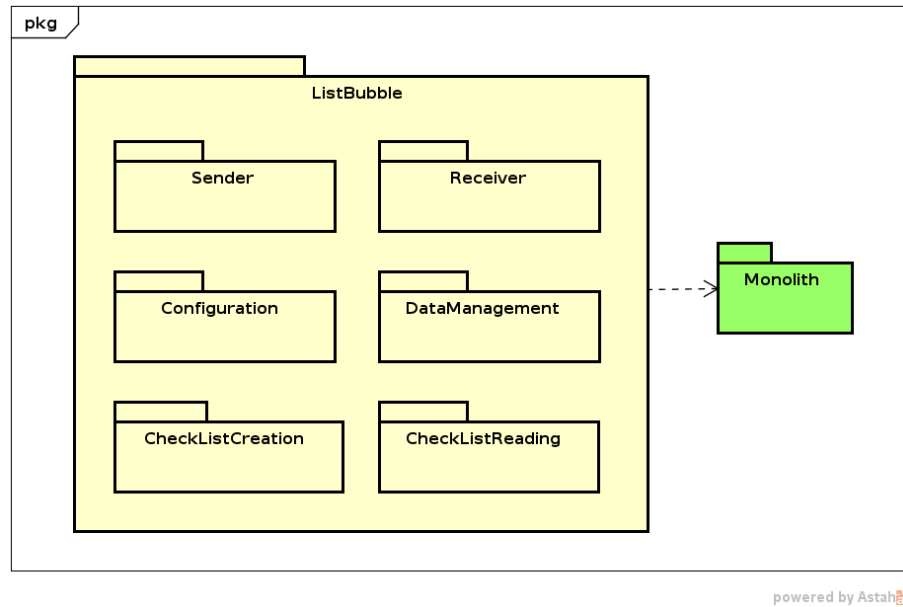


Figura 9: Diagramma per ListBubble.

**Descrizione:**

Componente contenente i pacchetti necessari per la creazione della bolla lista.

### 3.3.4 ListBubble::CheckListCreation

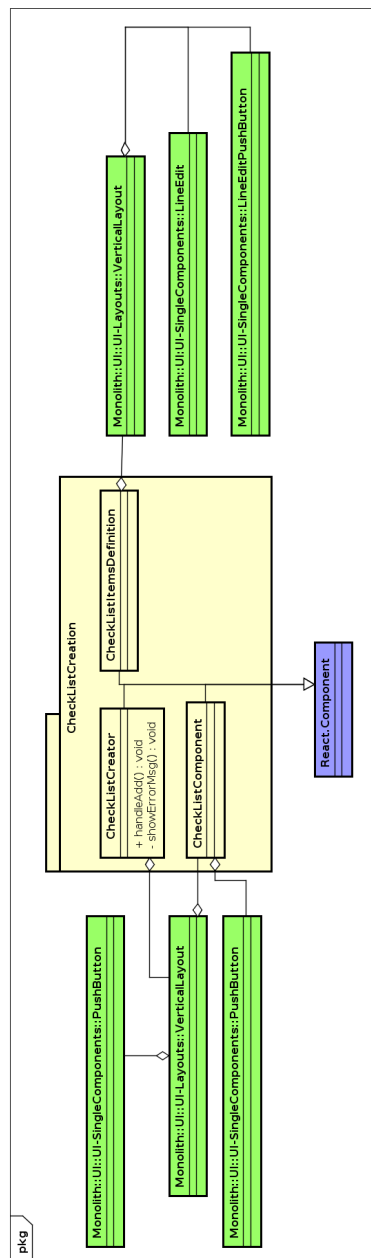


Figura 10: Diagramma per ListBubble::CheckListCreation.

**Descrizione:**

Componente che si occupa della creazione delle check list.

**Classi contenute:**

- CheckListComponent
- CheckListCreator



- CheckListItemsDefinition

### 3.3.5 ListBubble::CheckListReading

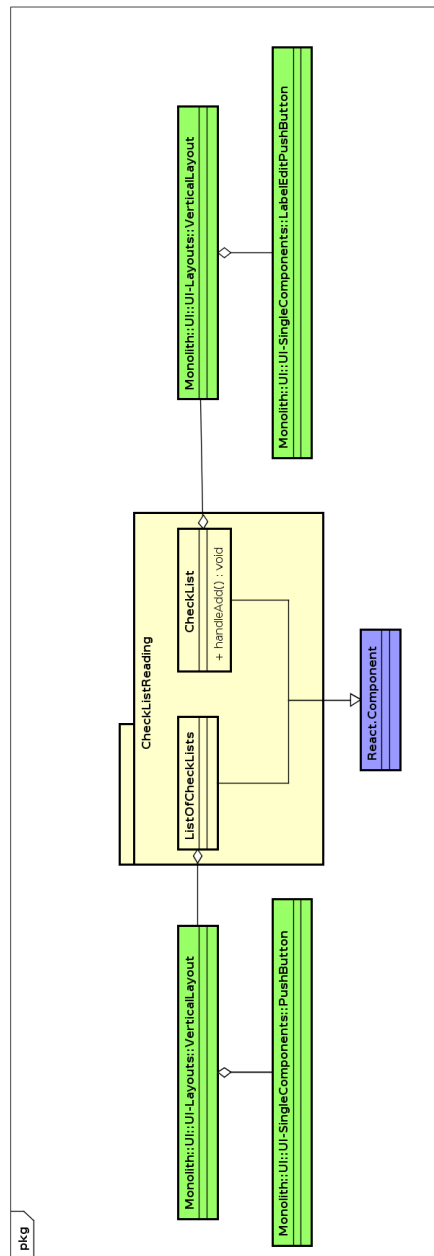


Figura 11: Diagramma per ListBubble::CheckListReading.

**Descrizione:**

Componente che si occupa della lettura e dell'utilizzo delle check list.

**Classi contenute:**

- CheckList
- ListOfCheckLists

### 3.3.6 ListBubble::Configuration

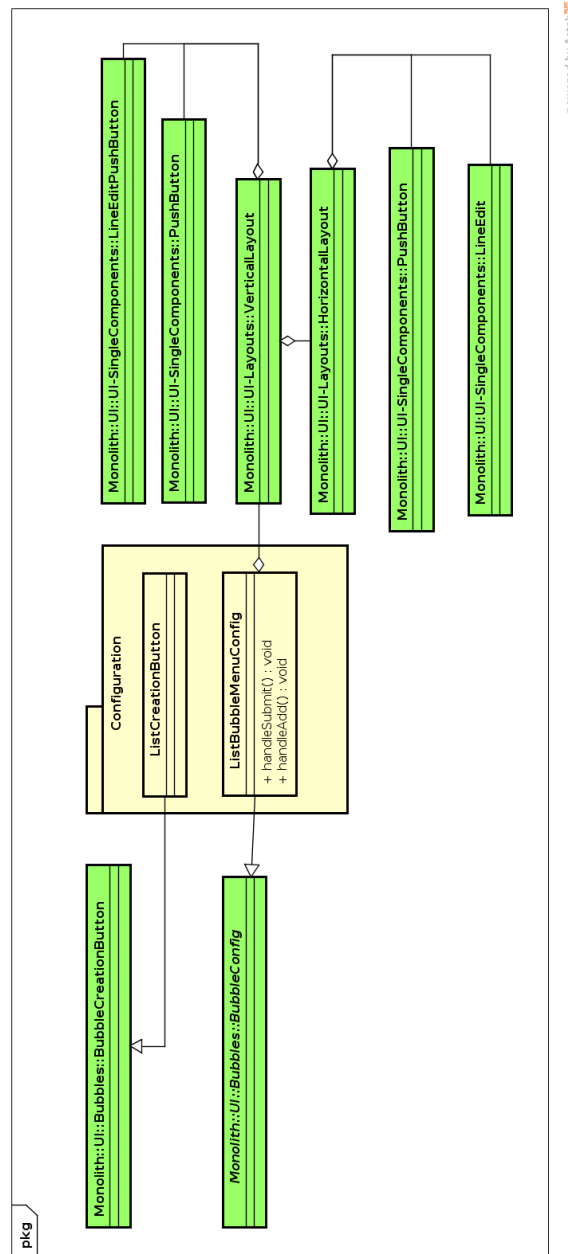


Figura 12: Diagramma per ListBubble::Configuration.

**Descrizione:**

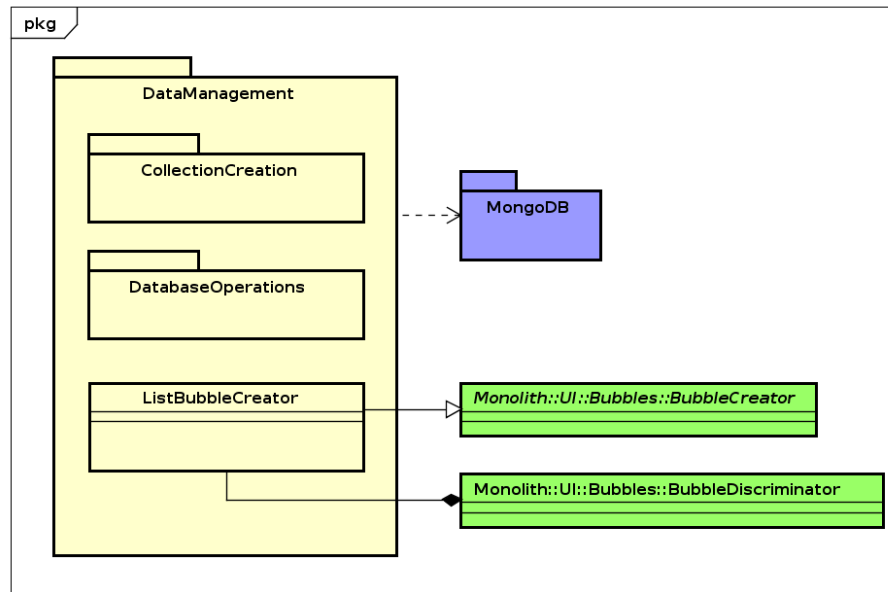
Componente che gestisce l'area di configurazione della bolla e il pulsante apposito da inserire nel menu iniziale di creazione.

**Classi contenute:**

- ListBubbleMenuConfig
- ListCreationButton



### 3.3.7 ListBubble::DataManagement



powered by Astah

Figura 13: Diagramma per ListBubble::DataManagement.

**Descrizione:**

Componente che si occupa di tutte le operazioni di gestione dei dati che non sono gestite da Monolith. Usa il database MongoDB.

**Classi contenute:**

- ListBubbleCreator



### 3.3.8 ListBubble::Receiver

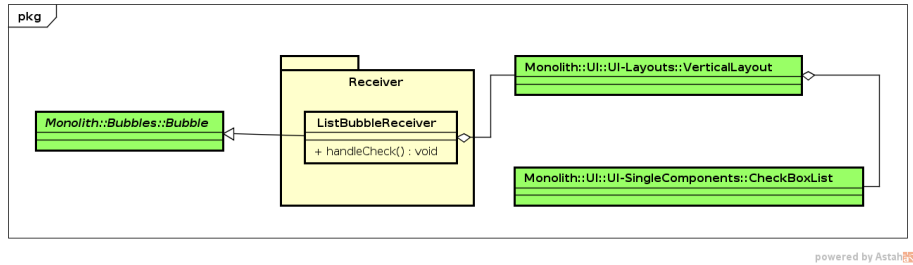


Figura 14: Diagramma per ListBubble::Receiver.

**Descrizione:**

Componente che gestisce la visualizzazione della bolla da parte del ricevente.

**Classi contenute:**

- ListBubbleReceiver



### 3.3.9 ListBubble::Sender

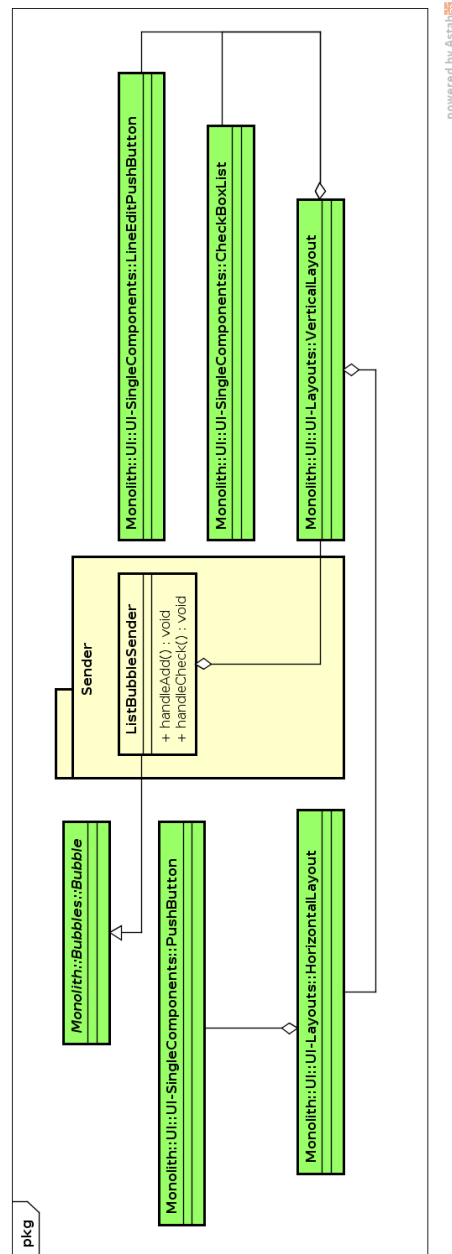


Figura 15: Diagramma per ListBubble::Sender.

**Descrizione:**

Componente che gestisce la visualizzazione della bolla da parte del mittente.

**Classi contenute:**

- ListBubbleSender



Figura 16: Diagramma per MeteoBubble.



- MeteoBubbleReceiver
- MeteoBubbleSender
- MeteoDelivery
- MeteoItem

### 3.3.11 SurveyBubble

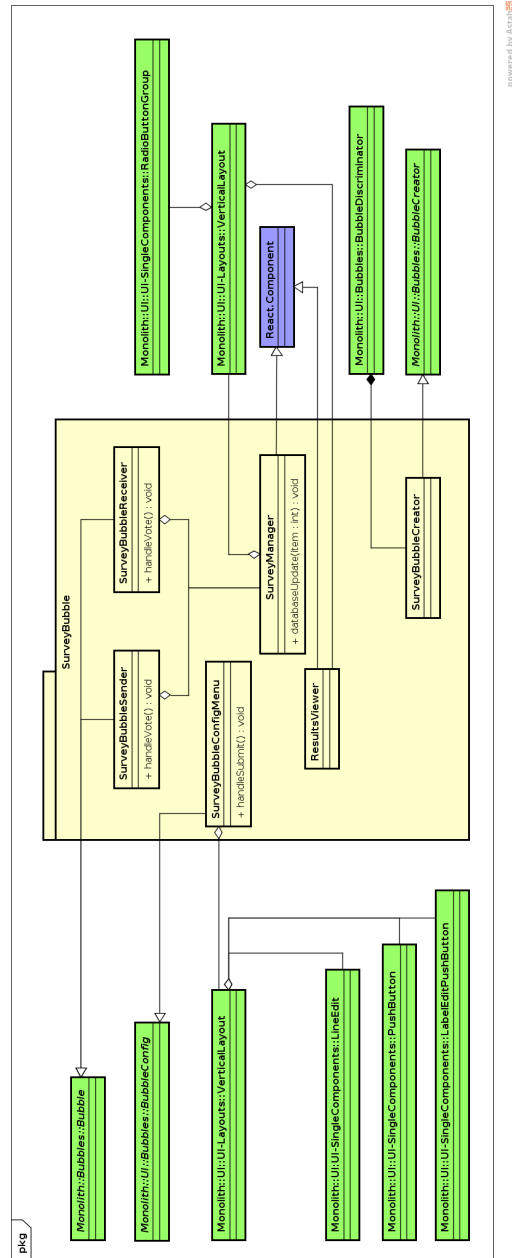


Figura 17: Diagramma per SurveyBubble.

**Descrizione:**

Componente contenente le classi necessarie per la creazione della bolla sondaggio.

**Classi contenute:**

- ResultsViewer
- SurveyBubbleConfigMenu



- SurveyBubbleCreator
- SurveyBubbleReceiver
- SurveyBubbleSender
- SurveyManager

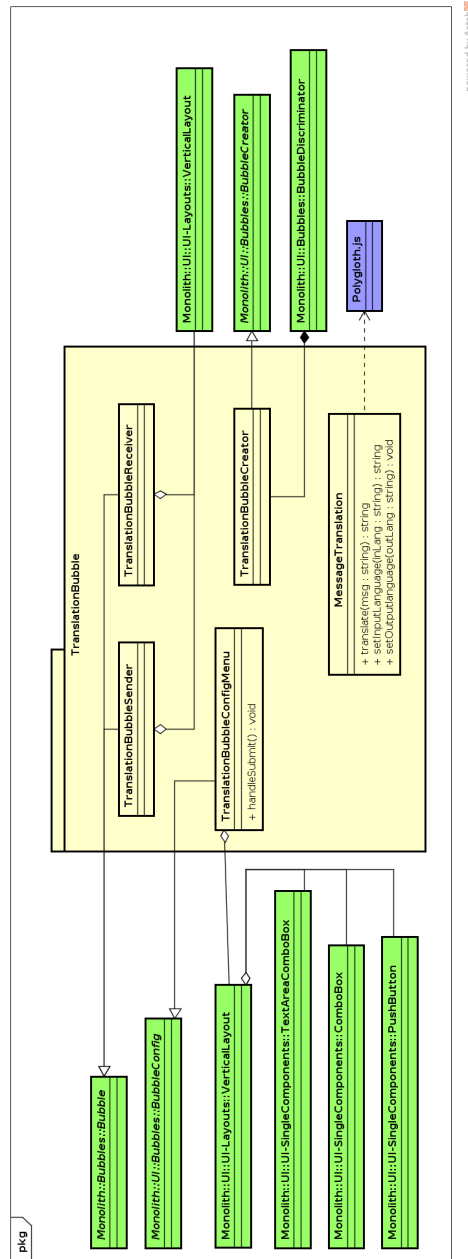


Figura 18: Diagramma per TranslationBubble.

- MessageTranslation
- TranslationBubbleConfigMenu



- TranslationBubbleCreator
- TranslationBubbleReceiver
- TranslationBubbleSender



### 3.4 Architettura di dettaglio - Classi del sistema Monolith

#### 3.4.1 check

**Componente:** Monolith::Database::informationStorage::Checks

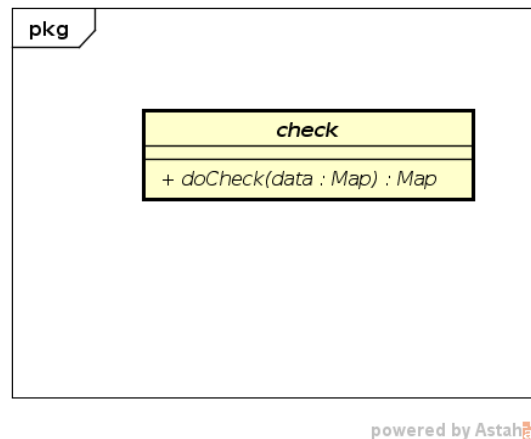


Figura 19: Diagramma per check in Checks

#### Descrizione

Classe concreta di concreteCheck che serve per effettuare controlli sui dati.

#### Metodi:

- +doCheck() : boolean  
Ritorna il risultato di un controllo, true se positivo false se negativo.

#### Applicazioni

Interfaccia che viene utilizzata come rappresentazione di concreteCheck.



### 3.4.2 checkCreator

**Componente:** Monolith::Database::informationStorage::Checks

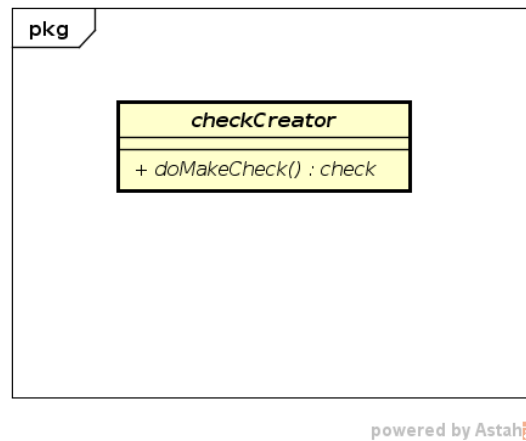


Figura 20: Diagramma per checkCreator in Checks

#### Descrizione

Classe astratta di concreteCheckCreator.

#### Metodi:

- +doMakeCheck() : check  
Ritorna un'istanza di check.

#### Applicazioni

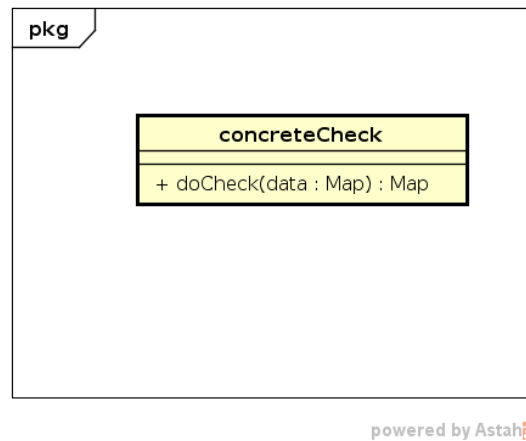
Viene utilizzata quando viene richiesto un controllo.





### 3.4.3 concreteCheck

**Componente:** Monolith::Database::informationStorage::Checks



powered by Astah

Figura 21: Diagramma per concreteCheck in Checks

#### Descrizione

Classe che effettua un controllo e ne ritorna il risultato.

#### Metodi:

- +doCheck() : bool  
Ritorna il risultato di un controllo, true se positivo false se negativo.

#### Applicazioni

Viene utilizzata per effettuare controlli sui dati.



#### 3.4.4 checkDiscriminator

**Componente:** Monolith::Database::informationStorage::Checks

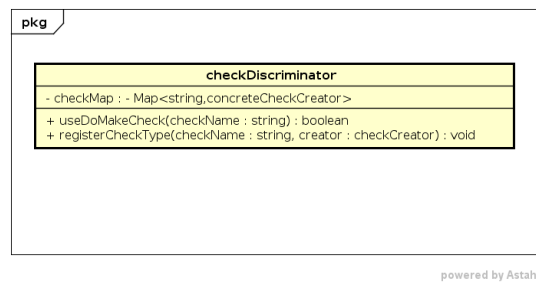


Figura 22: Diagramma per checkDiscriminator in Checks

##### Descrizione

Classe che effettua il controllo in base alla stringa considerata.

##### Attributi:

- -checkMap : Map< string , concreteCheckCreator >  
Struttura che mappa il nome del check con l'istanza del checkCreator.

##### Metodi:

- +useDoMakeCheck(checkName : string) : bool  
Ritorna il risultato del controllo corrispondete alla stringa passata.

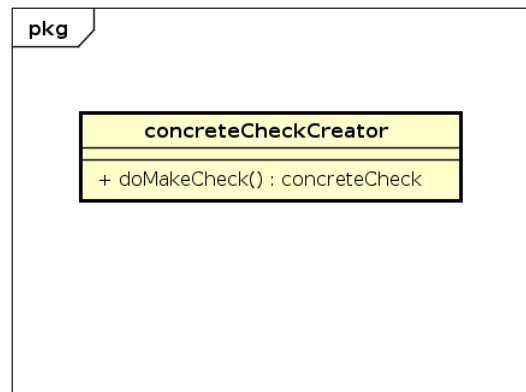
##### Applicazioni

Viene utilizzata quando viene richiesto un controllo.



### 3.4.5 concreteCheckCreator

**Componente:** Monolith::Database::informationStorage::Checks



powered by Astah

Figura 23: Diagramma per concreteCheckCreator in Checks

#### Descrizione

Classe che rappresenta istanze concrete di tipo check.

#### Metodi:

- +doMakeCheck() : concreteCheck  
Ritorna un'istanza di concreteCheck.

#### Applicazioni

Viene utilizzata per creare istanze di controlli.



### 3.4.6 Bubble

**Componente:** Monolith::UI::Bubbles

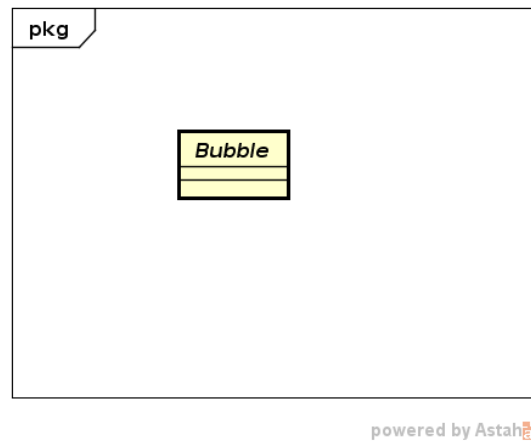


Figura 24: Diagramma per Bubble in Bubbles

#### **Descrizione**

Classe Astratta di ConcreteBubble.

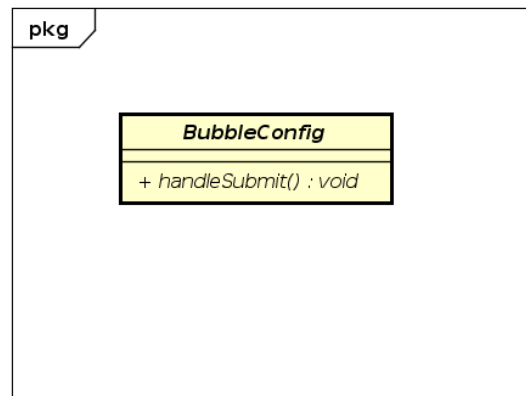
#### **Applicazioni**

Viene utilizzata come interfaccia generica di bolla.



### 3.4.7 BubbleConfig

**Componente:** Monolith::UI::Bubbles



powered by Astah

Figura 25: Diagramma per BubbleConfig in Bubbles

#### **Descrizione**

Classe Astratta di ConcreteBubbleConfig.

#### **Applicazioni**

Interfaccia che viene utilizzata come rappresentazione di ConcreteBubbleConfig.



### 3.4.8 bubbleCreator

**Componente:** Monolith::UI::Bubbles

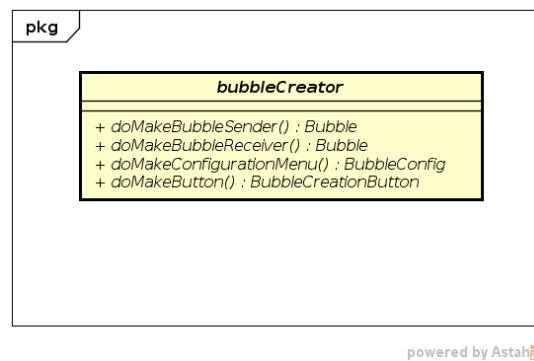


Figura 26: Diagramma per bubbleCreator in Bubbles

#### Descrizione

Classe astratta di concreteBubbleCreator.

#### Metodi:

- +doMakeBubbleSender() : ConcreteBubble  
Ritorna il componente React per visualizzare una bolla inviata.
- +doMakeBubbleReceiver() : ConcreteBubble  
Ritorna il componente React per visualizzare una bolla ricevuta.
- +doMakeConfigurationMenu() : ConcreteBubbleConfig  
Ritorna il componente React per visualizzare il menù di creazione di una bolla da inviare.
- +doMakeButton() : ConcreteBubbleCreationButton  
Ritorna il componente React per visualizzare il bottone di creazione del menù di configurazione di una bolla da inviare.

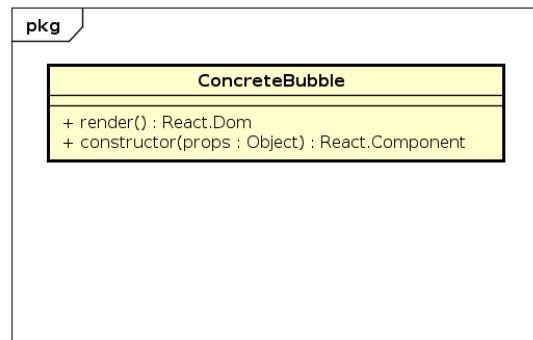
#### Applicazioni

Interfaccia che viene utilizzata come rappresentazione di concreteBubbleCreator.



### 3.4.9 ConcreteBubble

**Componente:** Monolith::UI::Bubbles



powered by Astah

Figura 27: Diagramma per ConcreteBubble in Bubbles

#### Descrizione

Classe React che rappresenta una bolla inviata o ricevuta.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata da concreteBubbleCreator per creare una bolla inviata o ricevuta dai dati recuperati da meteor.



### 3.4.10 bubbleDiscriminator

**Componente:** Monolith::UI::Bubbles

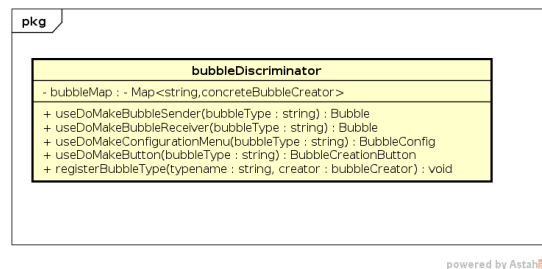


Figura 28: Diagramma per bubbleDiscriminator in Bubbles

#### Descrizione

Classe che contiene i metodi che ritornano le funzionalità necessarie per la rappresentazione delle bolle.

#### Attributi:

- -bubbleMap : Map< string,concreteBubbleCreator>  
Struttura che mappa il nome di una bolla con l'istanza di concreteBubbleCreator per quella bolla.

#### Metodi:

- +useDoMakeBubbleSender( bubbleType: string) : ConcreteBubble  
Ritorna il componente React della stringa passata come parametro per visualizzare una bolla inviata.
- +useDoMakeBubbleReceiver( bubbleType: string) : ConcreteBubble  
Ritorna il componente React della stringa passata come parametro per visualizzare una bolla ricevuta.
- +useDoMakeBubbleConfigurationMenu( bubbleType: string) : ConcreteBubbleConfig  
Ritorna il componente React della stringa passata come parametro per visualizzare il menù di creazione di una bolla da inviare.
- +useDoMakeButton( bubbleType: string) : ConcreteBubbleCreationButton  
Ritorna il componente React della stringa passata come parametro per visualizzare il bottone di creazione del menù di configurazione di una bolla da inviare.

#### Applicazioni

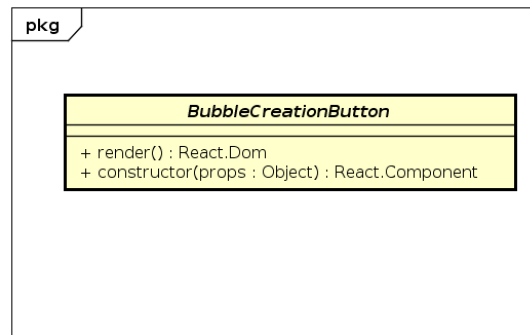
Viene utilizzata quando si deve creare una nuova bolla, ritornando l'oggetto della bolla appena selezionata.





### 3.4.11 BubbleCreationButton

**Componente:** Monolith::UI::Bubbles



powered by Astah

Figura 29: Diagramma per BubbleCreationButton in Bubbles

#### **Descrizione**

Classe Astratta di ConcreteBubbleCreationButton.

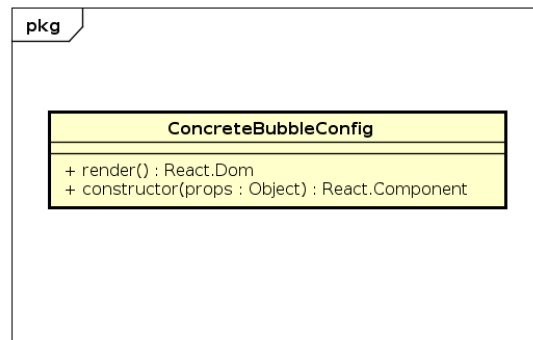
#### **Applicazioni**

Interfaccia che viene utilizzata come rappresentazione di ConcreteBubbleCreationButton.



### 3.4.12 ConcreteBubbleConfig

**Componente:** Monolith::UI::Bubbles



powered by Astah

Figura 30: Diagramma per ConcreteBubbleConfig in Bubbles

#### Descrizione

Classe che rappresenta il menù di creazione di una bolla.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata da concreteBubbleCreator per creare un'istanza del menù di creazione di una bolla.



### 3.4.13 concreteBubbleCreator

**Componente:** Monolith::UI::Bubbles

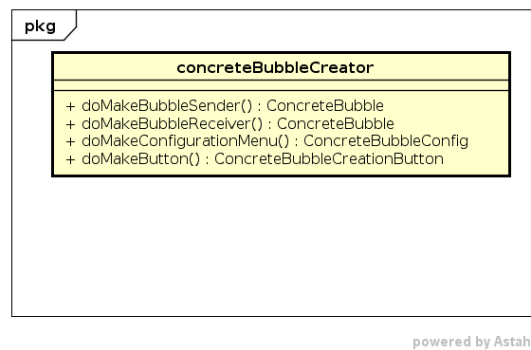


Figura 31: Diagramma per concreteBubbleCreator in Bubbles

#### Descrizione

Classe che contiene i factory method utilizzati per la creazione degli oggetti concreti di tipo Bubble, BubbleConfig e BubbleCreationButton.

#### Metodi:

- +doMakeBubbleSender() : ConcreteBubble  
Ritorna il componente React per visualizzare una bolla inviata.
- +doMakeBubbleReceiver() : ConcreteBubble  
Ritorna il componente React per visualizzare una bolla ricevuta.
- +doMakeConfigurationMenu() : ConcreteBubbleConfig  
Ritorna il componente React per visualizzare il menù di creazione di una bolla da inviare.
- +doMakeButton() : ConcreteBubbleCreationButton  
Ritorna il componente React per visualizzare il bottone di creazione del menù di configurazione di una bolla da inviare.

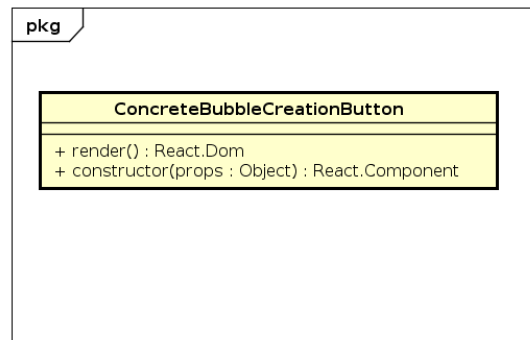
#### Applicazioni

Viene utilizzata da bubbleDiscriminator per creare le bolle ricevute e inviate, i menù di configurazione delle bolle e i bottoni per aprire il menù di configurazione delle bolle.



### 3.4.14 ConcreteBubbleCreationButton

**Componente:** Monolith::UI::Bubbles



powered by Astah

Figura 32: Diagramma per ConcreteBubbleCreationButton in Bubbles

#### Descrizione

Classe che rappresenta il bottone per aprire il menù di creazione di una bolla.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

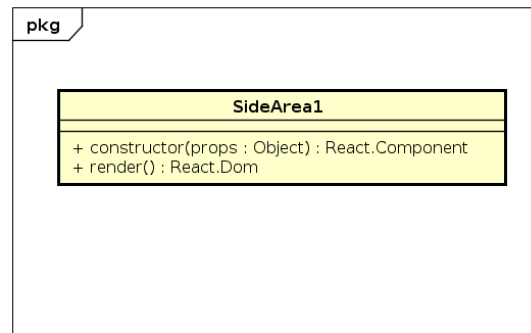
#### Applicazioni

Viene utilizzata da concreteBubbleCreator per creare un'istanza di un bottone nel menù di selezione di una bolla.



### 3.4.15 SideArea1

**Componente:** Monolith::UI::SideAreas::SideArea1\_pkg



powered by Astah

Figura 33: Diagramma per SideArea1 in SideArea1\_pkg

#### Descrizione

Classe che rappresenta il componente React per la visualizzazione del contenuto della prima area laterale.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata all'apertura della prima area laterale per visualizzare il menù di creazione delle bolle e lo storico delle bolle inviate.



### 3.4.16 SentBubbleHistory

**Componente:** Monolith::UI::SideAreas::SideArea1\_pkg

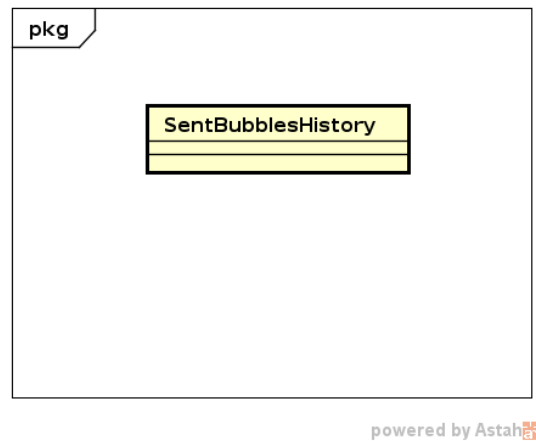


Figura 34: Diagramma per SentBubbleHistory in SideArea1\_pkg

#### Descrizione

Classe che rappresenta il componente React per la visualizzazione dello storico delle bolle inviate.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

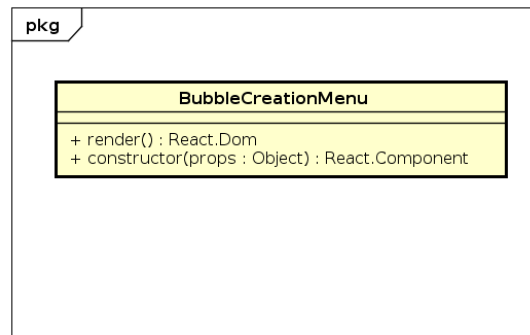
#### Applicazioni

Viene utilizzata dalla `SideArea1` per la visualizzazione dello storico delle bolle inviate.



### 3.4.17 BubbleCreationMenu

**Componente:** Monolith::UI::SideAreas::SideArea1\_pkg



powered by Astah

Figura 35: Diagramma per BubbleCreationMenu in SideArea1\_pkg

#### Descrizione

Classe che rappresenta il componente React per la configurazione delle bolle.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

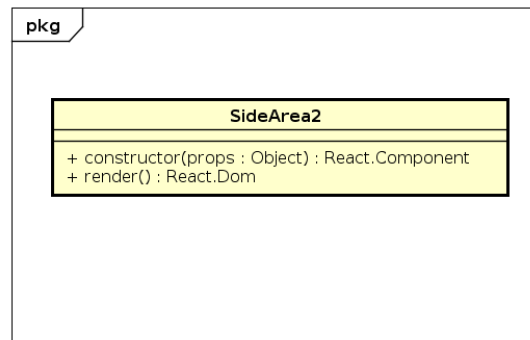
#### Applicazioni

Viene utilizzata dalla `SideArea1` per la visualizzazione del menù di creazione delle bolle.



### 3.4.18 SideArea2

**Componente:** Monolith::UI::SideAreas::SideArea2\_pkg



powered by Astah

Figura 36: Diagramma per SideArea2 in SideArea2\_pkg

#### Descrizione

Classe che rappresenta il componente React per la visualizzazione del contenuto della seconda area laterale.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

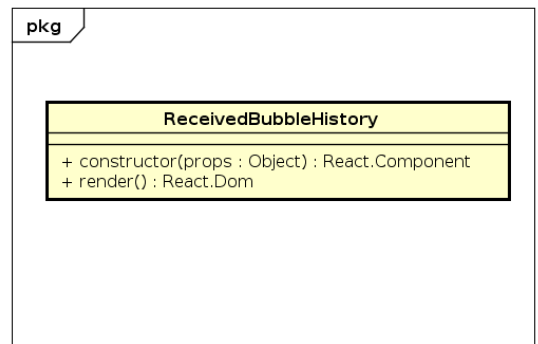
Viene utilizzata all'apertura della seconda area laterale per visualizzare dello storico delle bolle ricevute.





### 3.4.19 ReceivedBubbleHistory

**Componente:** Monolith::UI::SideAreas::SideArea2\_pkg



powered by Astah

Figura 37: Diagramma per ReceivedBubbleHistory in SideArea2\_pkg

#### Descrizione

Classe che rappresenta il componente React per la visualizzazione dello storico delle bolle ricevute.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

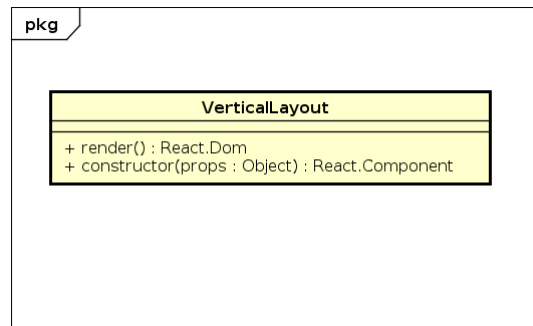
#### Applicazioni

Viene utilizzata dalla SideArea2 per la visualizzazione dello storico delle bolle ricevute.



### 3.4.20 VerticalLayout

**Componente:** Monolith::UI::UI-Layouts



powered by Astah

Figura 38: Diagramma per VerticalLayout in UI-Layouts

#### Descrizione

Classe contenitore che dispone i componenti contenuti uno sotto l'altro

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

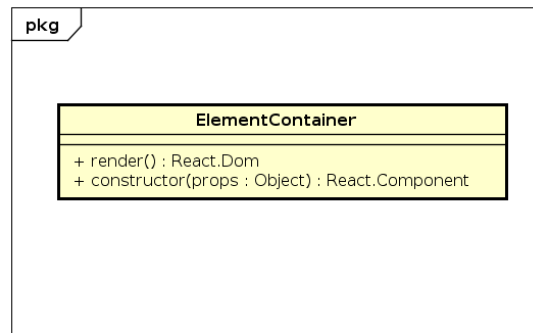
#### Applicazioni

Utilizzata per assegnare l'attributo className bootstrap a tutti i componenti figli, in modo che vengano visualizzati allineati in verticale con la dimensione adeguata.



### 3.4.21 ContainedElement

**Componente:** Monolith::UI::UI-Layouts



powered by Astah

Figura 39: Diagramma per ContainedElement in UI-Layouts

#### Descrizione

Classe che rappresenta un oggetto figlio dei Layout

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata dalle classi HorizontalLayout, VerticalLayout e ConditionalRendering per contenere un componente generico



### 3.4.22 HorizontalLayout

**Componente:** Monolith::UI::UI-Layouts

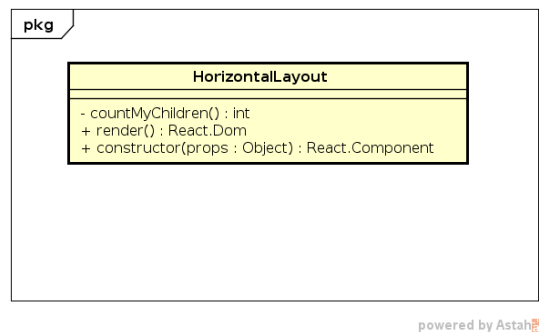


Figura 40: Diagramma per HorizontalLayout in UI-Layouts

#### Descrizione

Classe contenitore che dispone i componenti contenuti in orizzontale.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+countMyChildren() : int**  
Metodo che ritorna un intero contenente il numero di componenti figli, utilizzato per impostare la classe bootstrap corretta.

#### Applicazioni

Utilizzata per assegnare l'attributo className bootstrap a tutti i componenti figli, in modo che vengano visualizzati allineati in orizzontale con la dimensione adeguata.



### 3.4.23 ConditionalRendering

**Componente:** Monolith::UI::UI-Layouts

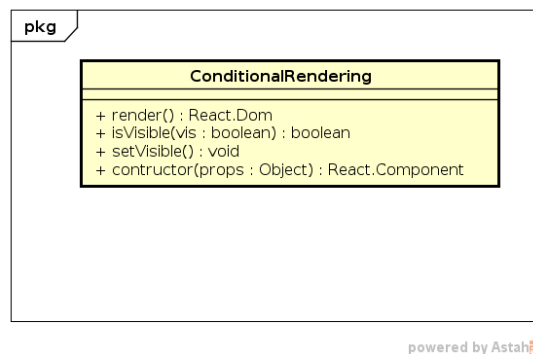


Figura 41: Diagramma per ConditionalRendering in UI-Layouts

#### Descrizione

Classe che fornisce i metodi per nascondere un componente contenuto all'esecuzione di un suo metodo

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+setVisible(vis : boolean): void`  
Imposta l'attributo che indica se il componente è visibile o meno in base al valore di `vis`.
- `+isVisible() : boolean`  
Metodo che ritorna `true` se il componente è visibile altrimenti ritorna `false`.
- `+setVisible( vis : boolean ) : void`  
Imposta l'attributo che indica se il componente è visibile o meno in base al valore del parametro.
- `+isVisible() : bool`  
Metodo che ritorna `true` se il componente è visibile altrimenti ritorna `false`.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.

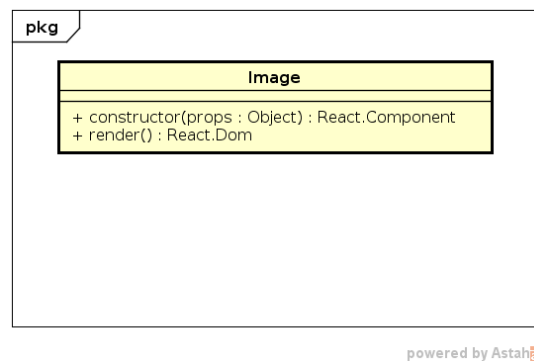
#### Applicazioni

Viene utilizzata per nascondere un componente a necessità.



### 3.4.24 Image

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 42: Diagramma per Image in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un'immagine.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.25 ComboBox

**Componente:** Monolith::UI::UI-SingleComponents

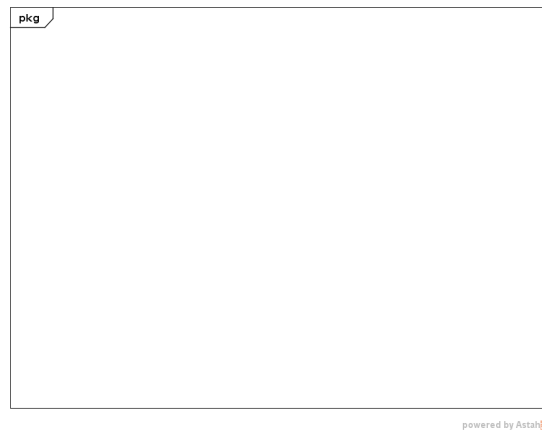


Figura 43: Diagramma per ComboBox in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un combobox.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+optChange(event : String) : void`  
Comunica l'opzione selezionata tramite il metodo del genitore.
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.26 LineEdit

**Componente:** Monolith::UI::UI-SingleComponents

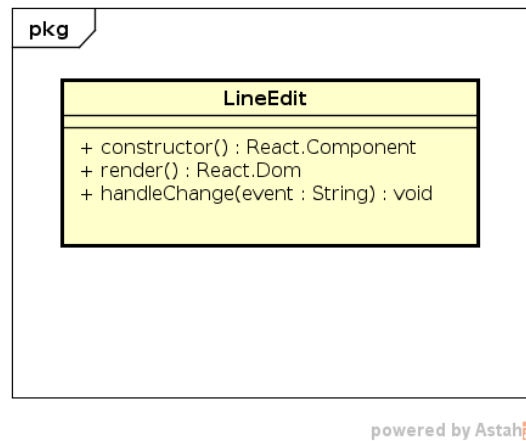


Figura 44: Diagramma per LineEdit in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un input di testo.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleChange(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.





### 3.4.27 LabelEdit

**Componente:** Monolith::UI::UI-SingleComponents

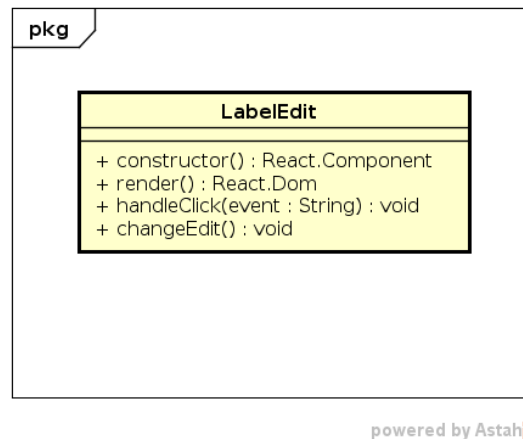


Figura 45: Diagramma per LabelEdit in UI-SingleComponents

#### Descrizione

Componente React che rappresenta una label che quando viene cliccata diventa un campo di inserimento testo.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+handleClick(event : String) : void`  
Viene invocato il metodo del genitore per trasferirvi il cambiamento di stato
- `+changeEdit() : void`  
Cambia lo stato del componente rendendolo (o meno) modificabile
- `+render() : React.DOM`  
Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

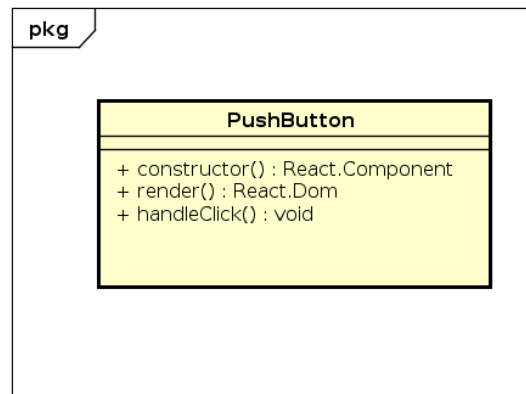
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.28 PushButton

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 46: Diagramma per PushButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un pulsante cliccabile.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleClick(): void**  
Viene invocato il metodo passato dal genitore
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

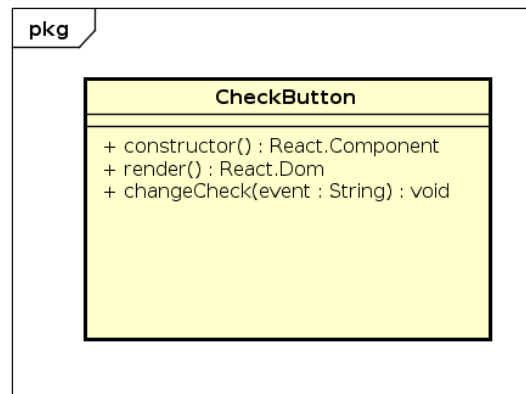
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.29 CheckButton

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 47: Diagramma per CheckButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un elemento di checkbox.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+changeCheck(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

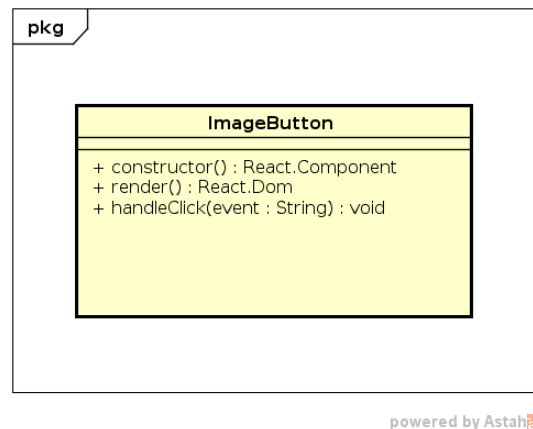
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.30 ImageButton

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 48: Diagramma per ImageButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un immagine che funge da pulsante.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleClick(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.31 CheckBoxList

**Componente:** Monolith::UI::UI-SingleComponents

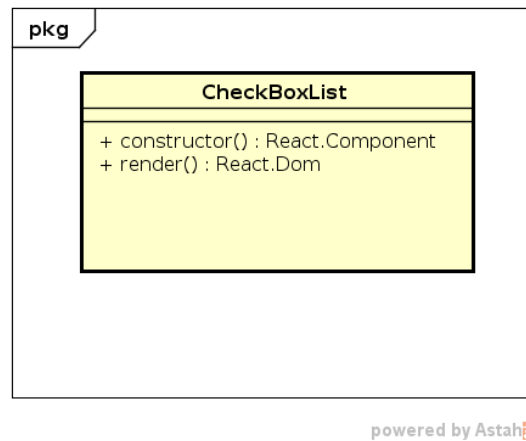


Figura 49: Diagramma per CheckBoxList in UI-SingleComponents

#### Descrizione

Componente React che rappresenta una lista di CheckBox.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.32 LabelComboBox

**Componente:** Monolith::UI::UI-SingleComponents

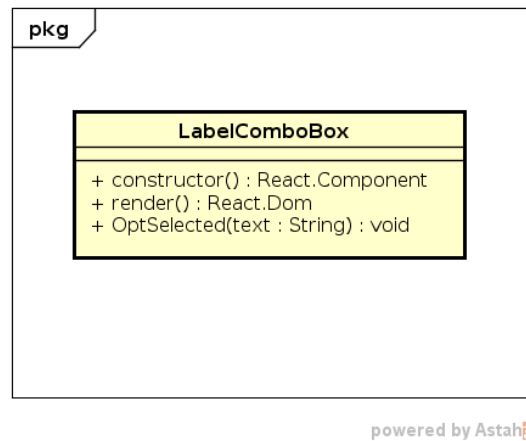


Figura 50: Diagramma per LabelComboBox in UI-SingleComponents

#### Descrizione

Componente React che rappresenta una lista di item selezionabile affiancata da una Label.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+OptSelected(text : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.33 TextAreaButton

**Componente:** Monolith::UI::UI-SingleComponents

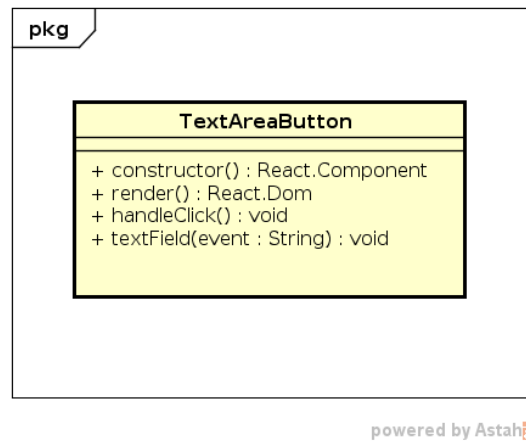


Figura 51: Diagramma per TextAreaButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un'area di testo con un pulsante.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+textField(text : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+handleClick(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.34 LabelPushButton

**Componente:** Monolith::UI::UI-SingleComponents

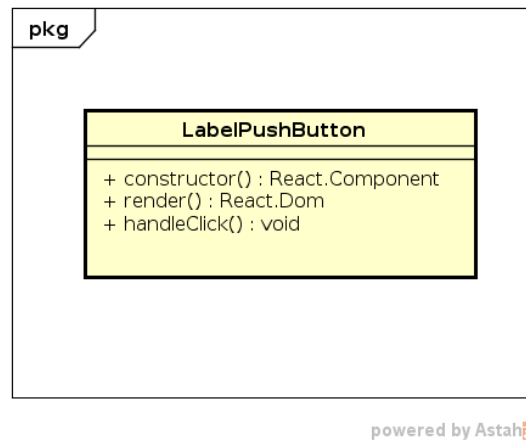


Figura 52: Diagramma per LabelPushButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un Label affiancato da un bottone cliccabile.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleClick(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.





### 3.4.35 LineEditComboBox

**Componente:** Monolith::UI::UI-SingleComponents

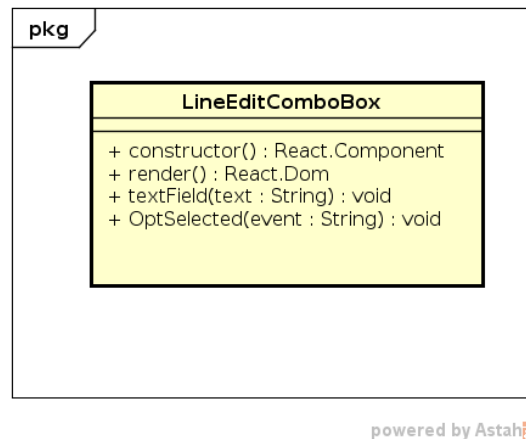


Figura 53: Diagramma per LineEditComboBox in UI-SingleComponents

#### Descrizione

Componente React che rappresenta una LineEdit con a fianco un ComboBox.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+textField(text : String) : void**  
Comunica il contenuto della textarea tramite il metodo del genitore.
- **+OptSelected(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

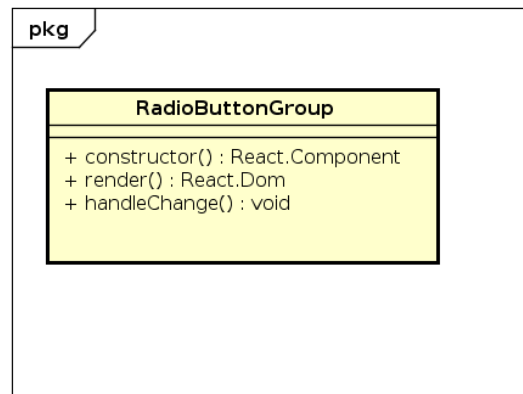
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.36 RadioButtonGroup

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 54: Diagramma per RadioButtonGroup in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un insieme di radio button tra cui è possibile scegliere tra opzioni mutualmente esclusive.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleChange(): void**  
Viene invocato il metodo passato dal genitore
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

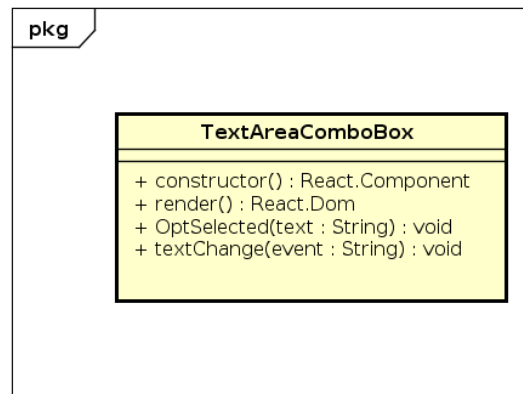
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.37 TextAreaComboBox

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 55: Diagramma per TextAreaComboBox in UI-SingleComponents

#### Descrizione

Componente React che rappresenta una TextArea con a fianco un ComboBox

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+OptSelected(text : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+textChange(event : String) : void**  
Comunica il contenuto della textarea tramite il metodo del genitore
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

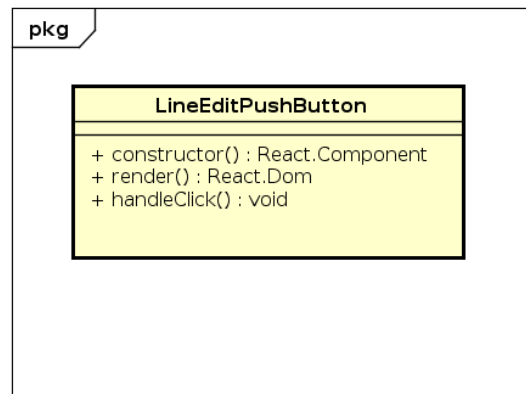
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.38 LineEditPushButton

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 56: Diagramma per LineEditPushButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta un campo di inserimento testo affiancato ad un pulsante.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+handleClick(event : String) : void**  
Viene invocato il metodo passato dal genitore
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

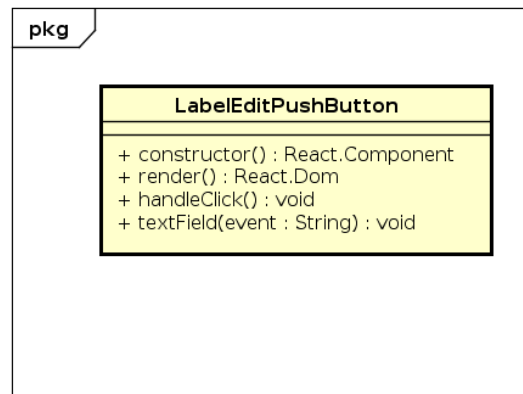
#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.



### 3.4.39 LabelEditPushButton

**Componente:** Monolith::UI::UI-SingleComponents



powered by Astah

Figura 57: Diagramma per LabelEditPushButton in UI-SingleComponents

#### Descrizione

Componente React che rappresenta del testo con un pulsante.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+textField(text : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+handleClick(event : String) : void**  
Comunica l'opzione selezionata tramite il metodo del genitore.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per costruire le interfacce grafiche delle bolle.

## 3.5 Architettura di dettaglio - Classi delle bolle demo

### 3.5.1 CurrencyConversion

**Componente:** CurrencyBubble

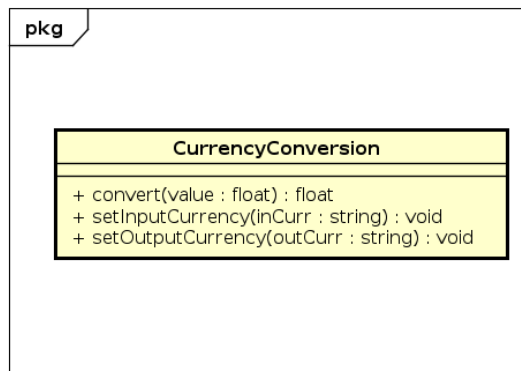


Figura 58: Diagramma per CurrencyConversion in CurrencyBubble

### Descrizione

Classe predisposta per la conversione di un importo da una valuta di partenza a una valuta di arrivo.

### Metodi:

- **+convert(value : float) : float**  
Metodo che effettua la conversione del valore utilizzando l'utility money.js.
- **+setInputCurrency(inCurr : string) : void**  
Metodo che imposta la valuta di partenza.
- **+setOutputCurrency(outCurr : string) : void**  
Metodo che imposta la valuta di arrivo.

### Applicazioni

Viene utilizzata per effettuare la conversione di un importo dalla valuta di partenza a quella di arrivo. Si serve dell'utility money.js.



### 3.5.2 CurrencyBubbleSender

**Componente:** CurrencyBubble

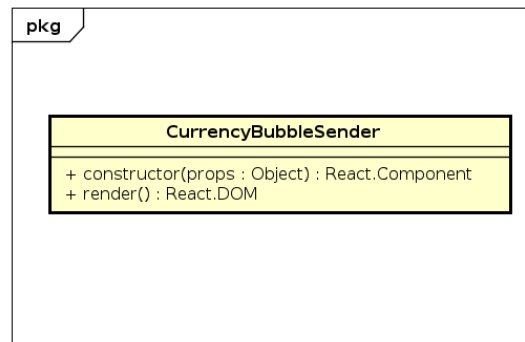


Figura 59: Diagramma per CurrencyBubbleSender in CurrencyBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal mittente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del mittente.



### 3.5.3 CurrencyBubbleCreator

**Componente:** CurrencyBubble

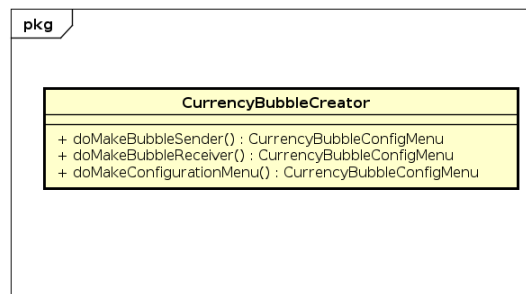


Figura 60: Diagramma per CurrencyBubbleCreator in CurrencyBubble

#### Descrizione

Istanziamento concreto della classe Monolith::UI::Bubbles::BubbleCreator che gestisce la creazione della singola istanza di bolla convertitore di valuta, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe Monolith::UI::Bubbles::BubbleDiscriminator.

#### Metodi:

- +doMakeBubbleSender() : CurrencyBubble  
Metodo che gestisce la creazione della bolla vista dal mittente.
- +doMakeBubbleReceiver() : CurrencyBubble  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- +doMakeConfigurationMenu() : CurrencyBubbleConfig  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- +doMakeButton() : CurrencyBubbleCreationButton  
Metodo che gestisce la creazione della singola istanza di pulsante da inserire nel menu iniziale di creazione. Viene lasciata l'implementazione della super classe.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla convertitore di valuta, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.





### 3.5.4 CurrencyBubbleReceiver

**Componente:** CurrencyBubble

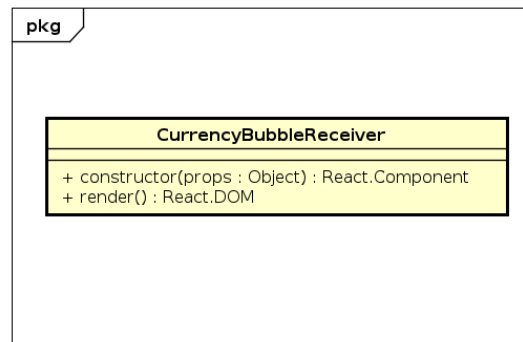


Figura 61: Diagramma per CurrencyBubbleReceiver in CurrencyBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal ricevente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.



### 3.5.5 CurrencyBubbleConfigMenu

**Componente:** CurrencyBubble

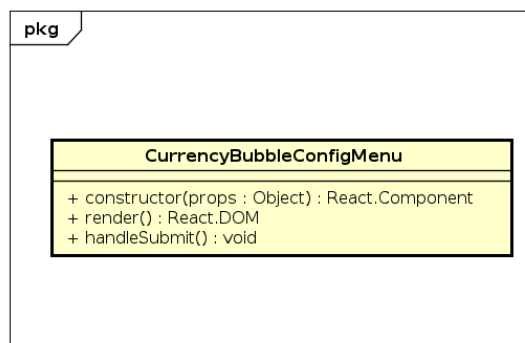


Figura 62: Diagramma per CurrencyBubbleConfigMenu in CurrencyBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleConfig` che rappresenta l'area di configurazione della bolla.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+ handleSubmit() : void` Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.



### 3.5.6 DiceRoller

**Componente:** DiceBubble

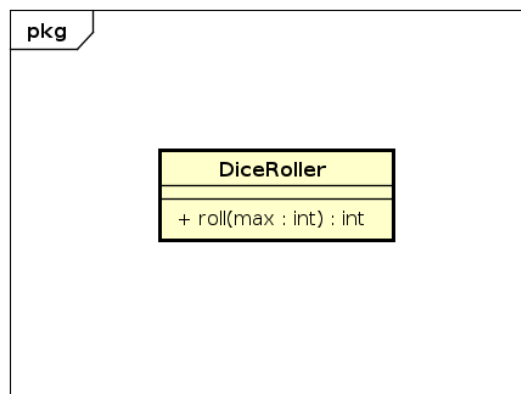


Figura 63: Diagramma per DiceRoller in DiceBubble

#### **Descrizione**

Classe predisposta per l'estrazione del numero casuale.

#### **Metodi:**

- **+roll(max : int) : int** Metodo che effettua l'estrazione del numero casuale utilizzando la funzione Javascript Math.random().

#### **Applicazioni**

Viene utilizzata per l'estrazione del numero casuale compreso tra 0 e max. Si serve della funzione Javascript Math.random().



### 3.5.7 DiceBubbleSender

**Componente:** DiceBubble

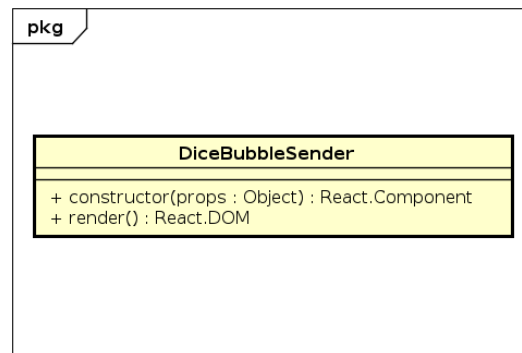


Figura 64: Diagramma per DiceBubbleSender in DiceBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal mittente.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del mittente.



### 3.5.8 DiceBubbleCreator

**Componente:** DiceBubble

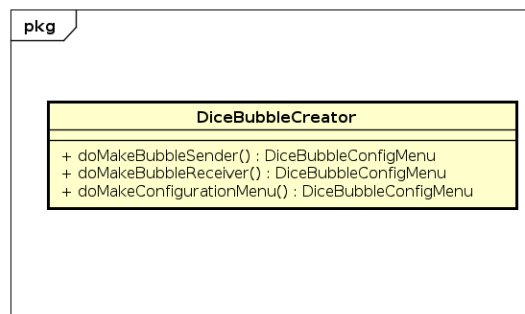


Figura 65: Diagramma per DiceBubbleCreator in DiceBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleCreator` che gestisce la creazione della singola istanza di bolla estrazione di numero casuale, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe `Monolith::UI::Bubbles::BubbleDiscriminator`.

#### Metodi:

- `+doMakeBubbleSender() : DiceBubble`  
Metodo che gestisce la creazione della bolla vista dal mittente.
- `+doMakeBubbleReceiver() : DiceBubble`  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- `+doMakeConfigurationMenu() : DiceBubbleConfig`  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- `+doMakeButton() : DiceBubbleCreationButton`  
Metodo che gestisce la creazione della singola istanza di pulsante da inserire nel menu iniziale di creazione. Viene lasciata l'implementazione della super classe.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla estrazione di numero casuale, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.



### 3.5.9 DiceBubbleReceiver

**Componente:** DiceBubble

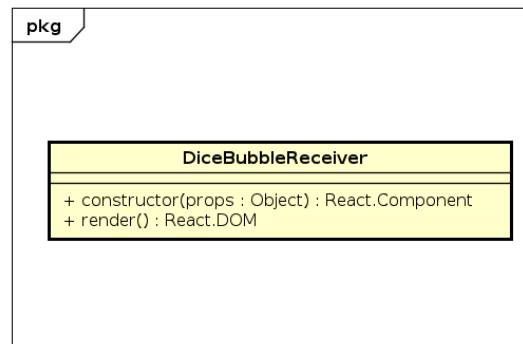


Figura 66: Diagramma per DiceBubbleReceiver in DiceBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal ricevente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.



### 3.5.10 DiceBubbleConfigMenu

**Componente:** DiceBubble

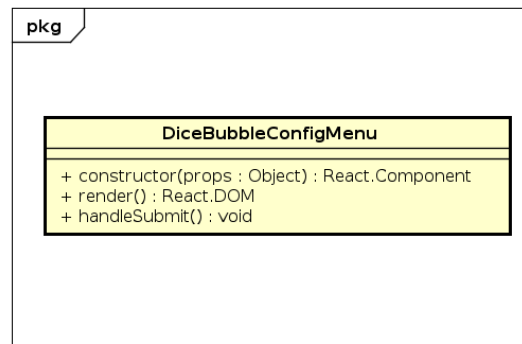


Figura 67: Diagramma per DiceBubbleConfigMenu in DiceBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleConfig` che rappresenta l'area di configurazione della bolla.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+handleSubmit() : void` Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.



### 3.5.11 CheckListCreator

**Componente:** ListBubble::CheckListCreation

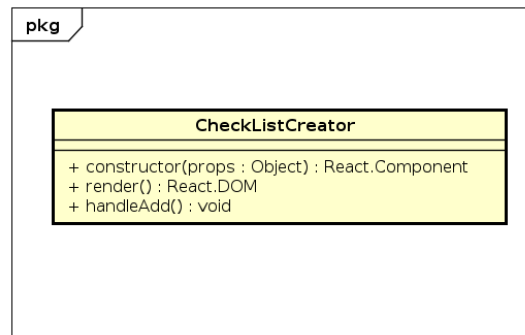


Figura 68: Diagramma per CheckListCreator in CheckListCreation

#### Descrizione

Componente React che rappresenta l'area di visualizzazione delle check list e un pulsante per poterne aggiungere di nuove.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleAdd() : void** Metodo che gestisce l'aggiunta di una nuova check list.

#### Applicazioni

Viene utilizzato per rappresentare l'area di visualizzazione delle check list e un pulsante per poterne aggiungere di nuove.





### 3.5.12 CheckListComponent

**Componente:** ListBubble::CheckListCreation

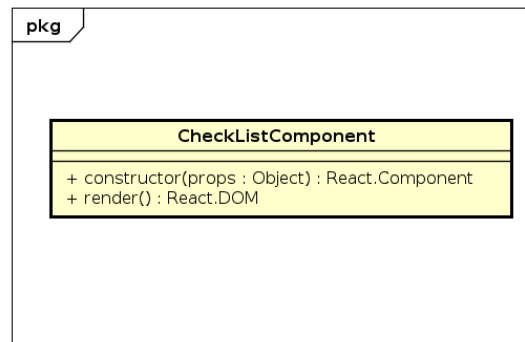


Figura 69: Diagramma per CheckListComponent in CheckListCreation

#### Descrizione

Componente React che rappresenta una lista di check list.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per rappresentare una lista di check list.



### 3.5.13 CheckListItemsDefinition

**Componente:** ListBubble::CheckListCreation

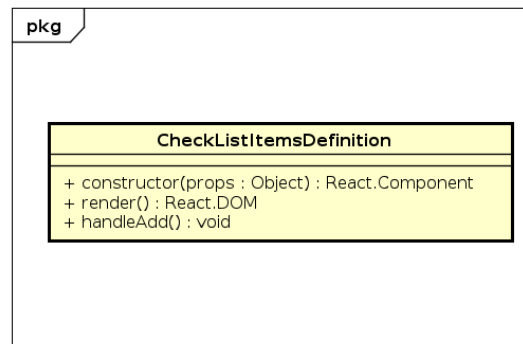


Figura 70: Diagramma per CheckListItemsDefinition in CheckListCreation

#### Descrizione

Componente React che rappresenta l'area di configurazione di una check list.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per rappresentare l'area di configurazione di una check list.



### 3.5.14 CheckList

**Componente:** ListBubble::CheckListReading

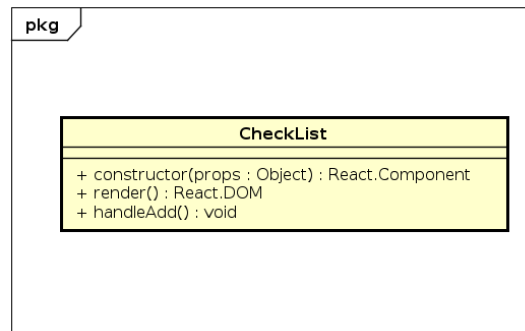


Figura 71: Diagramma per CheckList in CheckListReading

#### Descrizione

Componente React che gestisce l'aggiunta di un elemento alla lista.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleAdd() : void**  
Metodo che gestisce l'aggiunta di un elemento alla lista.

#### Applicazioni

Viene utilizzata per gestire l'aggiunta di un elemento alla lista.



### 3.5.15 ListOfCheckLists

**Componente:** ListBubble::CheckListReading

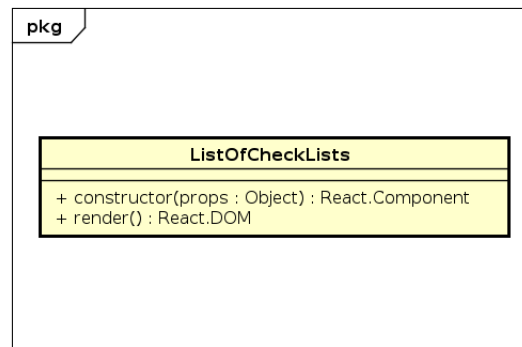


Figura 72: Diagramma per ListOfCheckLists in CheckListReading

#### Descrizione

Componente React che rappresenta una lista di check list.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della lista di check list.



### 3.5.16 ListCreationButton

**Componente:** ListBubble::Configuration

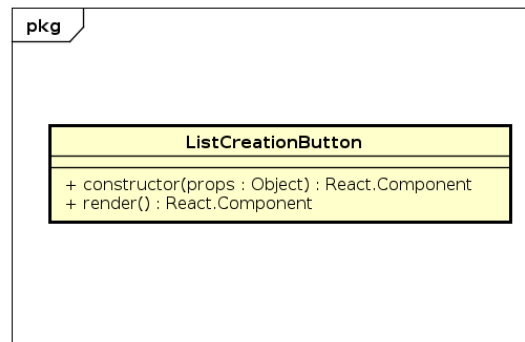


Figura 73: Diagramma per ListCreationButton in Configuration

#### Descrizione

Istanziamento della classe Monolith::UI::Bubbles::BubbleCreationButton che rappresenta la singola istanza di pulsante da inserire nel menu iniziale di creazione.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per rappresentare la singola istanza di pulsante da inserire nel menu iniziale di creazione.



### 3.5.17 ListBubbleMenuConfig

**Componente:** ListBubble::Configuration

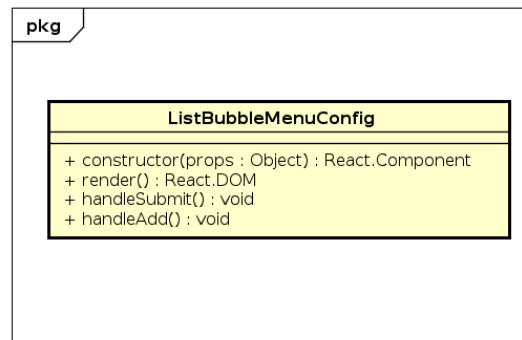


Figura 74: Diagramma per ListBubbleMenuConfig in Configuration

#### Descrizione

Istanziatura concreta della classe Monolith::UI::Bubbles::BubbleConfig che rappresenta l'area di configurazione della bolla.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleSubmit() : void**  
Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.
- **+handleAdd() : void**  
Metodo che viene invocato quando si vuole aggiungere un elemento alla lista.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.



### 3.5.18 ListBubbleCreator

**Componente:** ListBubble::DataManagement

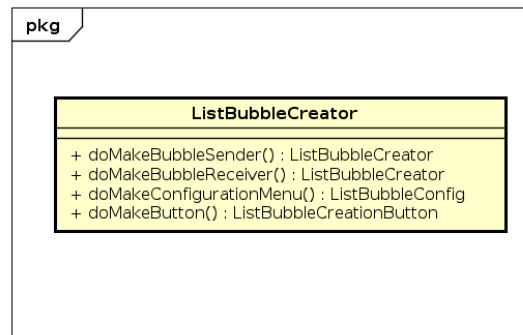


Figura 75: Diagramma per ListBubbleCreator in DataManagement

#### Descrizione

Istanziatura concreta della classe Monolith::UI::Bubbles::BubbleCreator che gestisce la creazione della singola istanza di bolla lista, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe Monolith::UI::Bubbles::BubbleDiscriminator.

#### Metodi:

- +doMakeBubbleSender() : ListBubble  
Metodo che gestisce la creazione della bolla vista dal mittente.
- +doMakeBubbleReceiver() : ListBubble  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- +doMakeConfigurationMenu() : ListBubbleConfig  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- +doMakeButton() : ListBubbleCreationButton  
Metodo che gestisce la creazione dello specifico pulsante da inserire nel menu iniziale di creazione.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla lista, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.



### 3.5.19 ListBubbleReceiver

**Componente:** ListBubble::Receiver

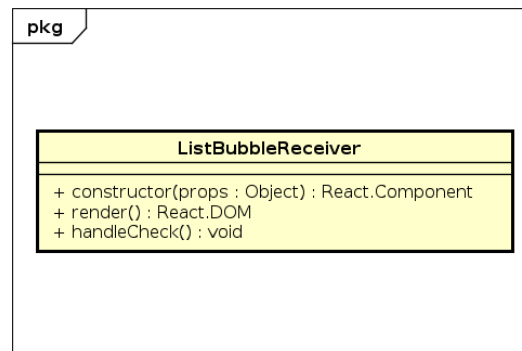


Figura 76: Diagramma per ListBubbleReceiver in Receiver

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal ricevente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleCheck() : void**  
Metodo che gestisce la spunta di un elemento della lista.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.





### 3.5.20 ListBubbleSender

**Componente:** ListBubble::Sender

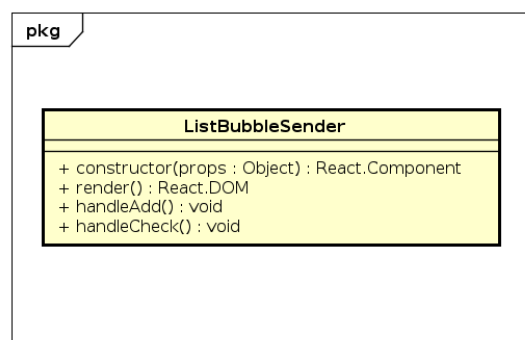


Figura 77: Diagramma per ListBubbleSender in Sender

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal mittente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM**  
Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleAdd() : void**  
Metodo che gestisce l'aggiunta di un nuovo elemento alla lista.
- **+handleCheck() : void**  
Metodo che gestisce la spunta di un elemento della lista.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del mittente.



### 3.5.21 MeteoItem

**Componente:** MeteoBubble

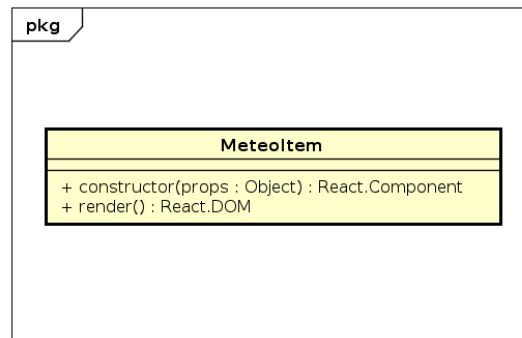


Figura 78: Diagramma per MeteoItem in MeteoBubble

#### Descrizione

Componente React predisposto per la visualizzazione delle previsioni meteo.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione delle previsioni meteo di un determinato giorno.



### 3.5.22 MeteoDelivery

**Componente:** MeteoBubble

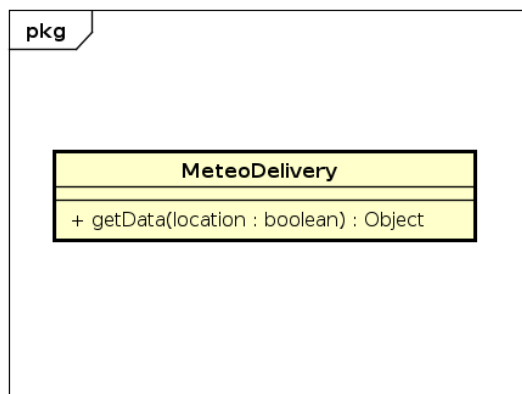


Figura 79: Diagramma per MeteoDelivery in MeteoBubble

#### Descrizione

Classe predisposta per il recupero dei dati relativi alle previsioni meteo.

#### Metodi:

- **+getData(location : string) : Object** Metodo che recupera i dati relativi alle previsioni meteo della località selezionata utilizzando l'utility `weather.js`.

#### Applicazioni

Viene utilizzata per recuperare i dati relativi alle previsioni meteo della località scelta. Si serve dell'utility `weather.js`.



### 3.5.23 MeteoBubbleSender

**Componente:** MeteoBubble

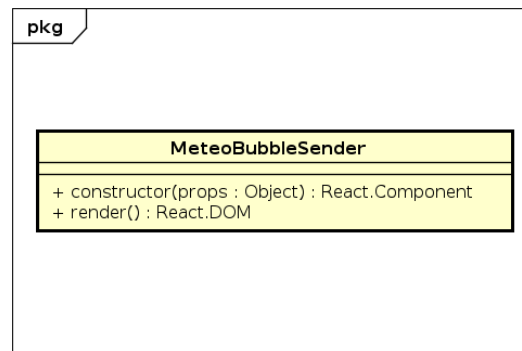


Figura 80: Diagramma per MeteoBubbleSender in MeteoBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal mittente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del mittente.



### 3.5.24 MeteoBubbleCreator

**Componente:** MeteoBubble

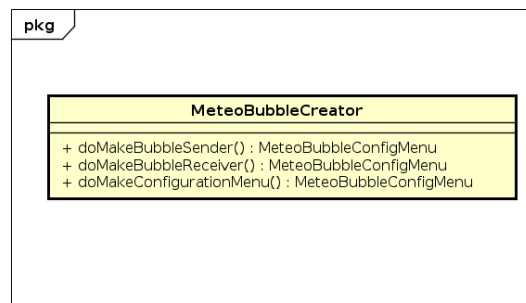


Figura 81: Diagramma per MeteoBubbleCreator in MeteoBubble

#### Descrizione

Istanziamento concreto della classe `Monolith::UI::Bubbles::BubbleCreator` che gestisce la creazione della singola istanza di bolla meteo, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe `Monolith::UI::Bubbles::BubbleDiscriminator`.

#### Metodi:

- `+doMakeBubbleSender() : MeteoBubble`  
Metodo che gestisce la creazione della bolla vista dal mittente.
- `+doMakeBubbleReceiver() : MeteoBubble`  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- `+doMakeConfigurationMenu() : MeteoBubbleConfig`  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- `+doMakeButton() : MeteoBubbleCreationButton`  
Metodo che gestisce la creazione dello specifico pulsante da inserire nel menu iniziale di creazione. Viene lasciata l'implementazione della super classe.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla meteo, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.



### 3.5.25 MeteoBubbleReceiver

**Componente:** MeteoBubble

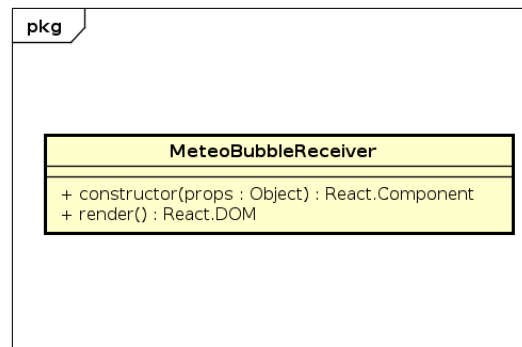


Figura 82: Diagramma per MeteoBubbleReceiver in MeteoBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal ricevente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.



### 3.5.26 MeteoBubbleConfigMenu

**Componente:** MeteoBubble

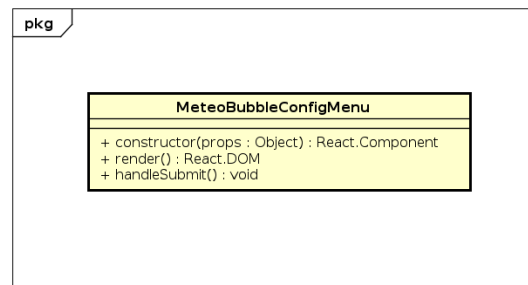


Figura 83: Diagramma per MeteoBubbleConfigMenu in MeteoBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleConfig` che rappresenta l'area di configurazione della bolla.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+handleSubmit() : void` Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.



### 3.5.27 ResultsViewer

**Componente:** SurveyBubble

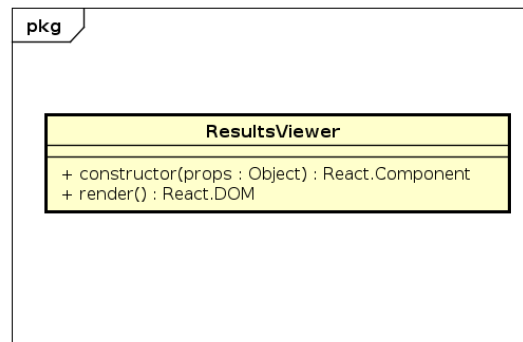


Figura 84: Diagramma per ResultsViewer in SurveyBubble

#### Descrizione

Componente React che rappresenta il risultato del sondaggio.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per la visualizzazione del risultato del sondaggio.





### 3.5.28 SurveyManager

**Componente:** SurveyBubble

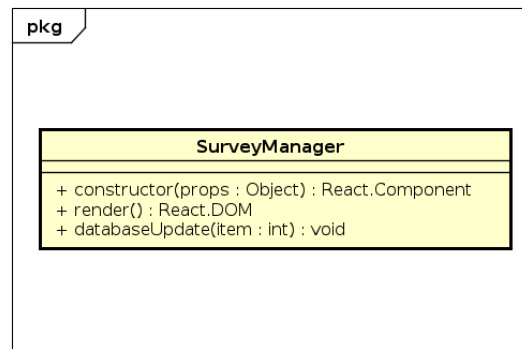


Figura 85: Diagramma per SurveyManager in SurveyBubble

#### Descrizione

Classe predisposta per la gestione del sondaggio.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+databaseUpdate(item : int) : void** Metodo che gestisce l'aggiornamento del database al cambiare dei voti nel sondaggio.

#### Applicazioni

Viene utilizzata per gestire l'aggiornamento del database al cambiare dei voti nel sondaggio.



### 3.5.29 SurveyBubbleSender

**Componente:** SurveyBubble

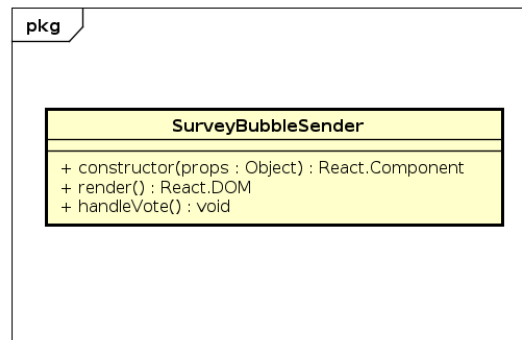


Figura 86: Diagramma per SurveyBubbleSender in SurveyBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::Bubbles::Bubble` che rappresenta la bolla vista dal mittente.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+handleVote() : void` Metodo che viene invocato quando è stata selezionata un'opzione.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del mittente.



### 3.5.30 SurveyBubbleCreator

**Componente:** SurveyBubble

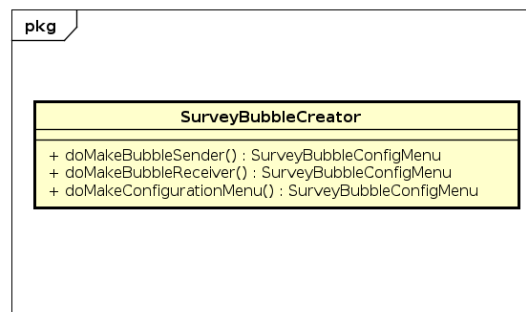


Figura 87: Diagramma per SurveyBubbleCreator in SurveyBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleCreator` che gestisce la creazione della singola istanza di bolla sondaggio, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe `Monolith::UI::Bubbles::BubbleDiscriminator`.

#### Metodi:

- `+doMakeBubbleSender() : SurveyBubble`  
Metodo che gestisce la creazione della bolla vista dal mittente.
- `+doMakeBubbleReceiver() : SurveyBubble`  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- `+doMakeConfigurationMenu() : SurveyBubbleConfig`  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- `+doMakeButton() : SurveyBubbleCreationButton`  
Metodo che gestisce la creazione della singola istanza di pulsante da inserire nel menu iniziale di creazione. Viene lasciata l'implementazione della super classe.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla sondaggio, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.



### 3.5.31 SurveyBubbleReceiver

**Componente:** SurveyBubble

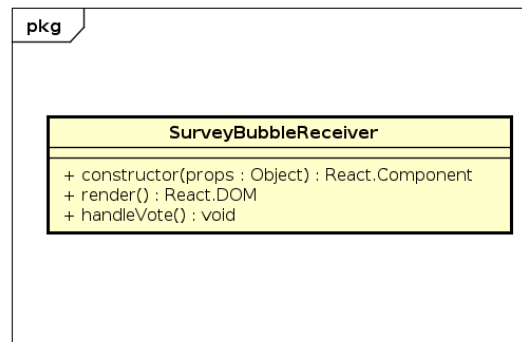


Figura 88: Diagramma per SurveyBubbleReceiver in SurveyBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::Bubbles::Bubble` che rappresenta la bolla vista dal ricevente.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+handleVote() : void` Metodo che viene invocato quando è stata selezionata un'opzione.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.



### 3.5.32 SurveyBubbleConfigMenu

**Componente:** SurveyBubble

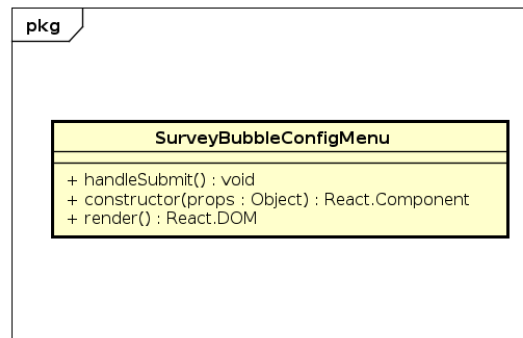


Figura 89: Diagramma per SurveyBubbleConfigMenu in SurveyBubble

#### Descrizione

Istanziamento concreto della classe `Monolith::UI::Bubbles::BubbleConfig` che rappresenta l'area di configurazione della bolla.

#### Metodi:

- `+constructor(props : Object) : React.Component`  
Costruttore della sottoclasse di `React.Component` in cui è necessario chiamare `super (props)` ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- `+render() : React.DOM` Metodo che esamina `this.props` e `this.state` e restituisce un singolo elemento `React` che può essere una rappresentazione di un componente `DOM` nativo o un altro componente composito.
- `+handleSubmit() : void` Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.



### 3.5.33 MessageTranslation

**Componente:** TranslationBubble

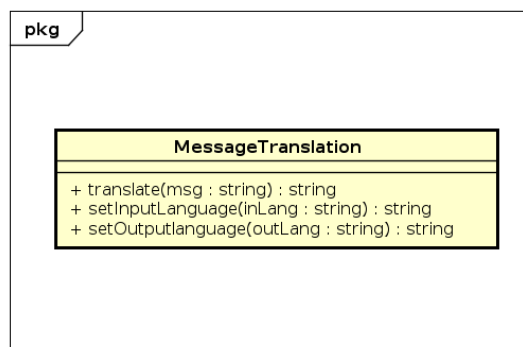


Figura 90: Diagramma per MessageTranslation in TranslationBubble

#### Descrizione

Classe predisposta per l'effettiva traduzione del messaggio.

#### Metodi:

- `+translate(msg : string) : string`  
Metodo che effettua la traduzione del messaggio utilizzando l'utility Polyglot.js.
- `+setInputLanguage(inLang : string) : void`  
Metodo che imposta la lingua di partenza.
- `+setOutputLanguage(outLang : string) : void`  
Metodo che imposta la lingua di arrivo.

#### Applicazioni

Viene utilizzata per effettuare la traduzione del messaggio di testo dalla lingua di partenza a quella di arrivo. Si serve dell'utility Polyglot.js.



### 3.5.34 TranslationBubbleSender

**Componente:** TranslationBubble

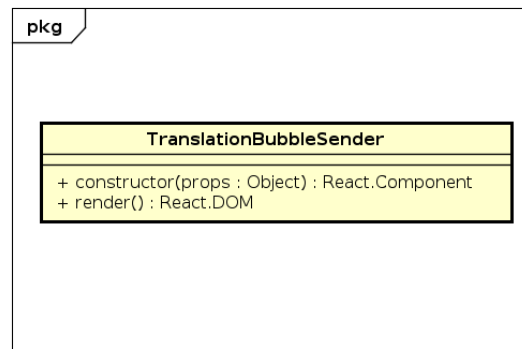


Figura 91: Diagramma per TranslationBubbleSender in TranslationBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal mittente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzato per la visualizzazione della bolla da parte del mittente.



### 3.5.35 TranslationBubbleCreator

**Componente:** TranslationBubble

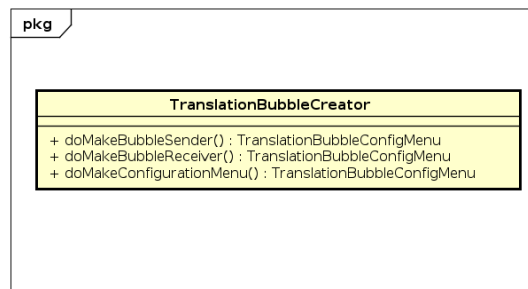


Figura 92: Diagramma per TranslationBubbleCreator in TranslationBubble

#### Descrizione

Istanziatura concreta della classe `Monolith::UI::Bubbles::BubbleCreator` che gestisce la creazione della singola istanza di bolla traduttore, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante tramite l'utilizzo della classe `Monolith::UI::Bubbles::BubbleDiscriminator`.

#### Metodi:

- `+doMakeBubbleSender() : TranslationBubble`  
Metodo che gestisce la creazione della bolla vista dal mittente.
- `+doMakeBubbleReceiver() : TranslationBubble`  
Metodo che gestisce la creazione della bolla vista dal ricevente.
- `+doMakeConfigurationMenu() : TranslationBubbleConfig`  
Metodo che gestisce la creazione dell'area di configurazione della bolla.
- `+doMakeButton() : TranslationBubbleCreationButton`  
Metodo che gestisce la creazione della singola istanza di pulsante da inserire nel menu iniziale di creazione. Viene lasciata l'implementazione della super classe.

#### Applicazioni

Viene utilizzata per gestire la creazione della singola istanza di bolla traduttore, della singola istanza di menù di configurazione della bolla e della singola istanza di pulsante.





### 3.5.36 TranslationBubbleReceiver

**Componente:** TranslationBubble

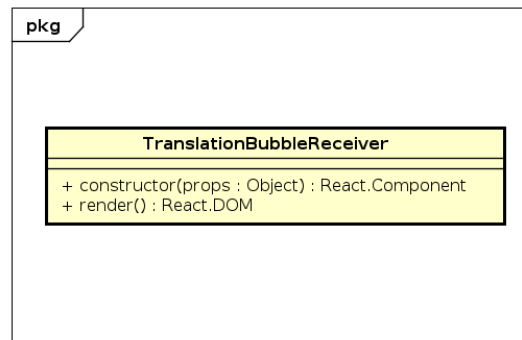


Figura 93: Diagramma per TranslationBubbleReceiver in TranslationBubble

#### Descrizione

Istanziatura concreta della classe Monolith::Bubbles::Bubble che rappresenta la bolla vista dal ricevente.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.

#### Applicazioni

Viene utilizzata per la visualizzazione della bolla da parte del ricevente.



### 3.5.37 TranslationBubbleConfigMenu

**Componente:** TranslationBubble

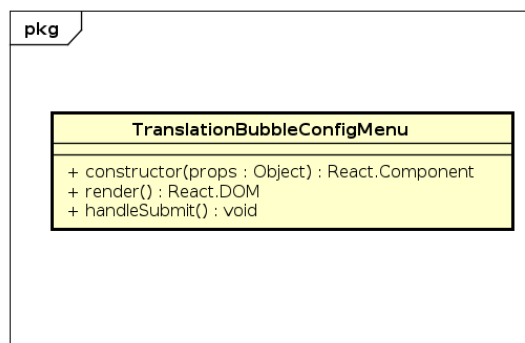


Figura 94: Diagramma per TranslationBubbleConfigMenu in TranslationBubble

#### Descrizione

Istanziatura concreta della classe Monolith::UI::Bubbles::BubbleConfig che rappresenta l'area di configurazione della bolla.

#### Metodi:

- **+constructor(props : Object) : React.Component**  
Costruttore della sottoclasse di React.Component in cui è necessario chiamare super (props) ed è possibile inizializzare lo stato per i dati soggetti a cambiamento.
- **+render() : React.DOM** Metodo che esamina this.props e this.state e restituisce un singolo elemento React che può essere una rappresentazione di un componente DOM nativo o un altro componente composito.
- **+handleSubmit() : void** Metodo che viene invocato quando la configurazione della bolla è terminata ed è pronta per essere inviata.

#### Applicazioni

Viene utilizzata per la visualizzazione dell'area di configurazione della bolla.

## 4 Standard di Progetto

### 4.1 Standard di documentazione del codice

Gli standard di documentazione del codice sono definiti nel documento Norme di Progetto v4.0.0 .

### 4.2 Standard di denominazione di entità e relazioni

Gli standard di denominazione sono definiti nel nel documento Norme di Progetto v4.0.0 .



### 4.3 Strumenti di lavoro

Gli strumenti da utilizzare e le procedure da seguire sono descritti nel documento Norme di Progetto v4.0.0 .

## 5 Diagrammi di Attività

Il diagramma delle attività è un diagramma definito all'interno dello Unified Modeling Language (UML) che definisce le attività da svolgere per realizzare una data funzionalità. Può essere utilizzato durante la progettazione del software per dettagliare un determinato algoritmo. Più in dettaglio, un activity diagram definisce una serie di attività o flusso, anche in termini di relazioni tra le attività, i responsabili per le singole attività e i punti di decisione. L'activity diagram è spesso usato come modello complementare allo Use Case Diagram, per descrivere le dinamiche con cui si sviluppano i diversi use case.

### 5.1 Configurazione sondaggio

Questo diagramma rappresenta la configurazione e l'invio della bolla sondaggio. L'utente inserisce il titolo del sondaggio, dopodiché avviene l'inserimento delle opzioni. L'utente ha la possibilità di inserire più opzioni (rappresentato nel diagramma con la freccia che parte dal nodo decisione e raggiunge il nodo azione chiamato inserisci opzioni). Una volta compiuto il processo di configurazione l'utente decide di inviare la bolla (rappresentato dal nodo invia). Se il numero delle opzioni inserite è minore di due allora il flusso ritorna al nodo azione chiamato "inserisci opzione" per poter permettere all'utente di incrementare il numero delle opzioni del sondaggio. Altrimenti (il numero di opzioni è maggiore di uno) il flusso arriva al nodo di fine dell'attività.

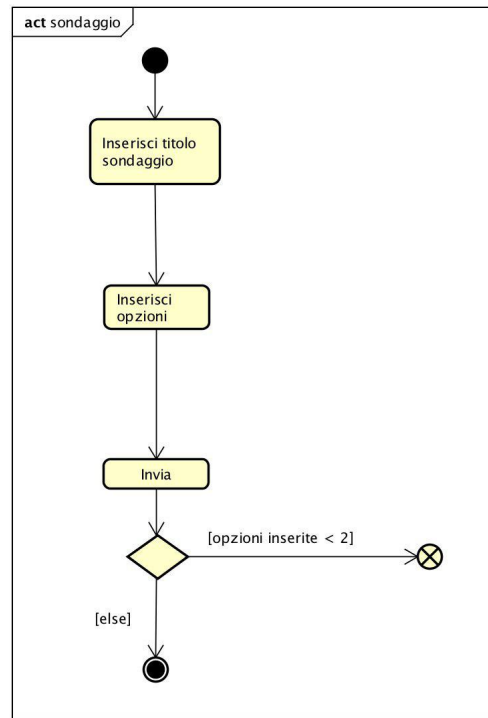


Figura 95: Diagramma di attività per la bolla sondaggio

## 5.2 Configurazione ListBubble

Questo diagramma rappresenta la configurazione della bolla ListBubble. Il flusso inizia con l'inserimento del titolo della lista. Dopo questo processo il flusso raggiunge il nodo decisione dove l'utente sceglie il modo di inserimento degli elementi. L'inserimento degli elementi avviene o manualmente (inserisci elemento manualmente) oppure con la selezione nella lista degli elementi predefiniti (inserisci elemento da checklist). Nel caso l'utente scegliesse di inserire un elemento dalla checklist predefinita, il flusso raggiunge il nodo decisione. Da questo punto l'utente può scegliere di inviare la bolla già configurata e il flusso raggiunge il nodo terminale, oppure scegliere di ritornare nel nodo decisione raggiunto precedentemente per poter inserire un altro elemento. Anche nel caso in cui l'utente scegliesse di inserire un elemento manualmente il flusso segue lo stesso percorso, ovvero l'utente inserisce l'elemento e decide di inviare la bolla con il raggiungere il nodo terminale oppure inserire un altro elemento.

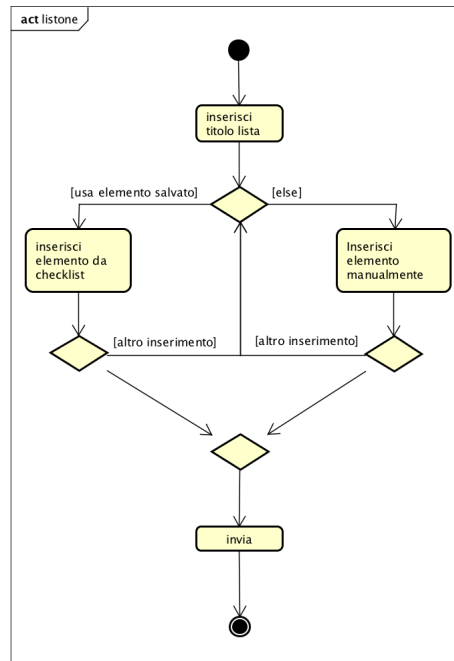


Figura 96: Diagramma di attività per la bolla ListBubble

## 6 Diagramma di Sequenza

### 6.1 Creazione Bolla

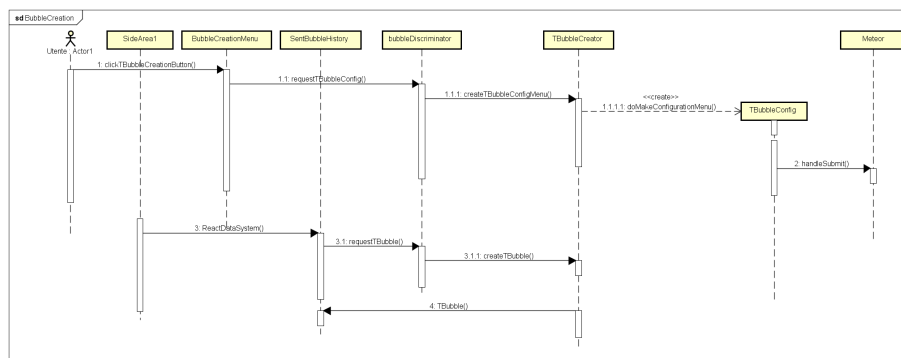


Figura 97: Diagramma di sequenza per la creazione di una bolla



**Scenario:** Questo scenario rappresenta la creazione di una bolla con l'utente che invia la richiesta per accedere al menù di creazione. La richiesta passa per il bubbleDiscriminator che capisce la tipologia di bolla e passa la richiesta al TBubbleCreator e TBubbleConfig che costruiscono il menù di configurazione adatto. Dopodichè SideArea1 si aggiorna automaticamente (tramite il meccanismo di aggiornamento di Minimongo contenuto in Meteor) e una volta compilati i campi la richiesta torna al bubbleDiscriminator e al TBubbleCreator che creano la bolla aggiungendola alla SentBubbleHistory.

## 6.2 Cancellazione Bolla

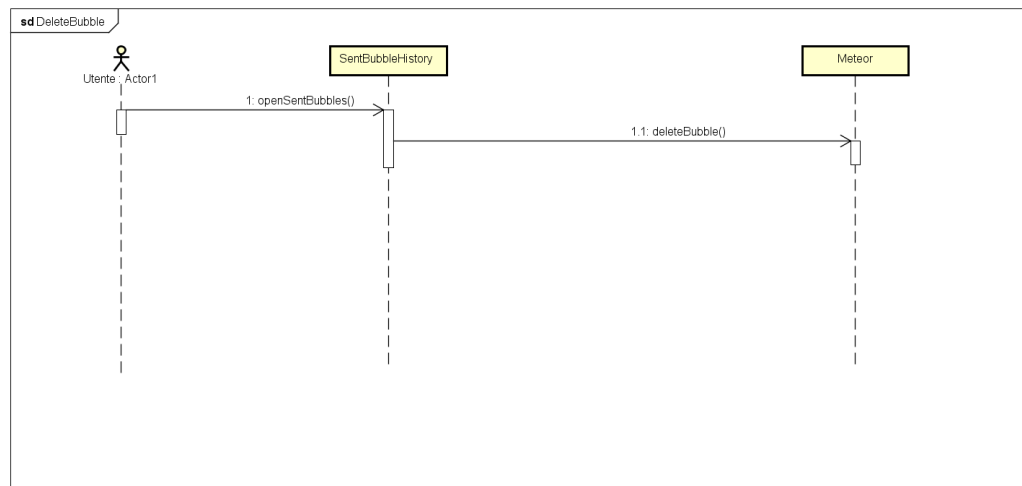


Figura 98: Diagramma di sequenza per la cancellazione di una bolla

**Scenario:** Questo scenario rappresenta la cancellazione, da parte di un utente, di una delle proprie bolle inviate precedentemente. L'utente in questione accede alla propria area di bolle inviate e seleziona la bolla da cancellare. La richiesta arriva al database dove avviene la cancellazione, successivamente, Meteor, gestisce l'aggiornamento dei dati nello storico delle bolle inviate (dell'utente attore) e in quello delle bolle ricevute degli altri utenti.

### 6.3 Voto Bolla-Sondaggio

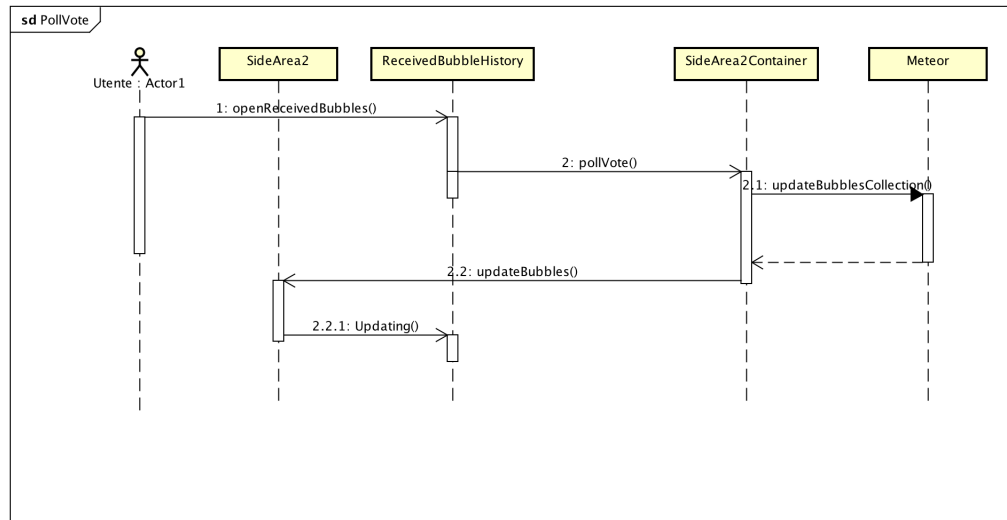


Figura 99: Diagramma di sequenza per il voto in una bolla-sondaggio

**Scenario:** Questo scenario rappresenta il voto di un utente ad una bolla sondaggio. L'utente accede all'area delle bolle ricevute e vota l'opzione che desidera, il voto viene salvato nel database e, attraverso Meteor, vengono aggiornati i dati della bolla in questione di ogni utente.

## 7 Tracciamento

### 7.1 Tracciamento componenti-requisiti

Componente	Requisiti
CurrencyBubble	RObFu01-cv RObFu01.1-cv RObFu01.2-cv RObFu01.3-cv RObFu01.4-cv ROpFu02-cv
DiceBubble	RObFu01-dd RObFu01.1-dd RObFu01.2-dd ROpFu01.2.1-dd RObFu01.3-dd



Componente	Requisiti
ListBubble	RObFu01-ls RObFu01.1-ls RObFu01.2-ls RObFu02-ls RObFu03-ls RObFu04-ls
ListBubble::CheckListCreation	RObFu02-ls
ListBubble::CheckListReading	RObFu01.2-ls
ListBubble::Configuration	RObFu01-ls RObFu01.1-ls
ListBubble::Receiver	RObFu03-ls RObFu04-ls
ListBubble::Sender	RObFu03-ls RObFu04-ls
MeteoBubble	RObFu01-mt RObFu01.1-mt RObFu01.2-mt RObFu01.3-mt
Monolith	RObFu11 RObFu11.1 RObFu11.2 RObFu11.2.1
Monolith::Database	RObFu10.2.5 RObFu10.2.5.1 RObFu10.2.7 RObFu10.3 RObFu10.3.1 RObFu10.3.1.1 RObFu11.1 RObFu11.2 RObFu11.2.2 RObFu11.2.3 RObFu11.2.3.1 RObFu11.2.3.2 RObFu11.2.3.3 RObFu20 RObFu21
Monolith::Database::InformationStorage	RObFu11.2.3 RObFu11.2.3.1 RObFu11.2.3.2 RObFu11.2.3.3 RObFu20
Monolith::Database::InformationStorage:: DatabaseSettings	RObFu20
Monolith::Database::informationStorage:: Checks	RObFu22 RObFu22.1





Componente	Requisiti
Monolith::UI	RObFu11.1 RObFu11.2.1 RObFu11.2.1.7 RObFu23
Monolith::UI::Bubbles	RObFu10.2.1 RObFu10.2.2 RObFu10.2.3 RObFu10.2.4 RObFu10.2.5 RObFu10.3 RObFu10.3.1 RObFu11.1 RObFu11.2.1.6 RObFu11.2.3 RObFu22
Monolith::UI::SideAreas	RObFu10 RObFu10.1
Monolith::UI::SideAreas::SideArea1_pkg	RObFu10 RObFu10.2 RObFu10.2.1 RObFu10.2.2 RObFu10.2.3 RObFu10.2.4 RObFu10.2.5 RObFu10.2.5.1 RObFu10.2.7
Monolith::UI::SideAreas::SideArea2_pkg	RObFu10 RObFu10.3 RObFu10.3.1 RObFu10.3.1.1
Monolith::UI::UI-Layouts	RObFu11.2.1.2 RObFu11.2.1.2.1 RObFu11.2.1.2.2 RObFu11.2.1.2.3



Componente	Requisiti
Monolith::UI::UI-SingleComponents	RObFu11.2.1 RObFu11.2.1.1 RObFu11.2.1.1.1 RObFu11.2.1.1.2 RObFu11.2.1.1.3 RObFu11.2.1.1.4 RObFu11.2.1.1.5 RObFu11.2.1.1.6 RObFu11.2.1.3 RObFu11.2.1.3.1 RObFu11.2.1.3.2 RObFu11.2.1.3.3 RObFu11.2.1.3.4 RObFu11.2.1.3.5 RObFu11.2.1.3.6 RObFu23
SurveyBubble	RObFu01-sd RObFu02-sd RObFu04-sd
TranslationBubble	RObFu01-tr RObFu01.1-tr RObFu01.2-tr RObFu01.4-tr RObFu01.5-tr

Tabella 3: Tracciamento componenti - requisiti

## 7.2 Tracciamento requisiti-componenti

Requisito	Componente
RObFu01-cv	CurrencyBubble
RObFu01-dd	DiceBubble
RObFu01-ls	ListBubble ListBubble::Configuration
RObFu01-mt	MeteoBubble
RObFu01-sd	SurveyBubble
RObFu01-tr	TranslationBubble
RObFu01.1-cv	CurrencyBubble
RObFu01.1-dd	DiceBubble
RObFu01.1-ls	ListBubble ListBubble::Configuration
RObFu01.1-mt	MeteoBubble
RObFu01.1-tr	TranslationBubble
RObFu01.2-cv	CurrencyBubble



Requisito	Componente
RObFu01.2-dd	DiceBubble
RObFu01.2-ls	ListBubble ListBubble::CheckListReading
RObFu01.2-mt	MeteoBubble
RObFu01.2-tr	TranslationBubble
ROpFu01.2.1-dd	DiceBubble
RObFu01.3-cv	CurrencyBubble
RObFu01.3-dd	DiceBubble
RObFu01.3-mt	MeteoBubble
RObFu01.4-cv	CurrencyBubble
RObFu01.4-tr	TranslationBubble
RObFu01.5-tr	TranslationBubble
ROpFu02-cv	CurrencyBubble
RObFu02-ls	ListBubble ListBubble::CheckListCreation
RObFu02-sd	SurveyBubble
RObFu03-ls	ListBubble ListBubble::Receiver ListBubble::Sender
RObFu04-ls	ListBubble ListBubble::Receiver ListBubble::Sender
RObFu04-sd	SurveyBubble
RObFu10	Monolith::UI::SideAreas Monolith::UI::SideAreas::SideArea1_pkg Monolith::UI::SideAreas::SideArea2_pkg
RObFu10.1	Monolith::UI::SideAreas
RObFu10.2	Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.1	Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.2	Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.3	Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.4	Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.5	Monolith::Database Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.5.1	Monolith::Database Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.2.7	Monolith::Database Monolith::UI::SideAreas::SideArea1_pkg
RObFu10.3	Monolith::Database Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea2_pkg



Requisito	Componente
RObFu10.3.1	Monolith::Database Monolith::UI::Bubbles Monolith::UI::SideAreas::SideArea2_pkg
RObFu10.3.1.1	Monolith::Database Monolith::UI::SideAreas::SideArea2_pkg
RObFu11	Monolith
RObFu11.1	Monolith Monolith::Database Monolith::UI Monolith::UI::Bubbles
RObFu11.2	Monolith Monolith::Database
RObFu11.2.1	Monolith Monolith::UI Monolith::UI::UI-SingleComponents
RObFu11.2.1.1	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.1	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.2	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.3	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.4	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.5	Monolith::UI::UI-SingleComponents
RObFu11.2.1.1.6	Monolith::UI::UI-SingleComponents
RObFu11.2.1.2	Monolith::UI::UI-Layouts
RObFu11.2.1.2.1	Monolith::UI::UI-Layouts
RObFu11.2.1.2.2	Monolith::UI::UI-Layouts
RObFu11.2.1.2.3	Monolith::UI::UI-Layouts
RObFu11.2.1.3	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.1	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.2	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.3	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.4	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.5	Monolith::UI::UI-SingleComponents
RObFu11.2.1.3.6	Monolith::UI::UI-SingleComponents
RObFu11.2.1.6	Monolith::UI::Bubbles
RObFu11.2.1.7	Monolith::UI
RObFu11.2.2	Monolith::Database
RObFu11.2.3	Monolith::Database Monolith::Database::InformationStorage Monolith::UI::Bubbles
RObFu11.2.3.1	Monolith::Database Monolith::Database::InformationStorage
RObFu11.2.3.2	Monolith::Database Monolith::Database::InformationStorage
RObFu11.2.3.3	Monolith::Database Monolith::Database::InformationStorage



Requisito	Componente
RObFu20	Monolith::Database Monolith::Database::InformationStorage Monolith::Database::InformationStorage:: DatabaseSettings
RObFu21	Monolith::Database
RObFu22	Monolith::Database::informationStorage:: Checks Monolith::UI::Bubbles
RObFu22.1	Monolith::Database::informationStorage:: Checks
RObFu23	Monolith::UI Monolith::UI::UI-SingleComponents

Tabella 5: Tracciamento requisiti - componenti

### 7.3 Tracciamento classi-requisiti

Class	Requisiti
CurrencyBubble::CurrencyBubbleConfigMenu	RObFu01-cv RObFu01.2-cv RObFu01.1-cv ROpFu02-cv
CurrencyBubble::CurrencyBubbleCreator	RObFu01-cv
CurrencyBubble::CurrencyBubbleReceiver	RObFu01.4-cv RObFu01-cv
CurrencyBubble::CurrencyBubbleSender	RObFu01.4-cv RObFu01-cv
CurrencyBubble::CurrencyConversion	RObFu01-cv ROpFu02-cv RObFu01.3-cv
DiceBubble::DiceBubbleConfigMenu	RObFu01.1-dd RObFu01-dd
DiceBubble::DiceBubbleCreator	RObFu01-dd
DiceBubble::DiceBubbleReceiver	RObFu01.2-dd RObFu01-dd ROpFu01.2.1-dd
DiceBubble::DiceBubbleSender	RObFu01-dd ROpFu01.2.1-dd RObFu01.2-dd
DiceBubble::DiceRoller	RObFu01.3-dd RObFu01-dd
ListBubble::CheckListCreation:: CheckListComponent	RObFu02-ls RObFu01-ls



Class	Requisiti
ListBubble::CheckListCreation:: CheckListCreator	RObFu01-ls RObFu02-ls
ListBubble::CheckListCreation:: CheckListItemsDefinition	RObFu01-ls RObFu02-ls
ListBubble::CheckListReading::CheckList	RObFu01-ls RObFu01.2-ls
ListBubble::CheckListReading:: ListOfCheckLists	RObFu01.2-ls RObFu01-ls
ListBubble::Configuration:: ListBubbleMenuConfig	RObFu01.1-ls RObFu01-ls
ListBubble::Configuration::ListCreationButton	RObFu01.1-ls RObFu01-ls
ListBubble::Receiver::ListBubbleReceiver	RObFu04-ls RObFu03-ls
ListBubble::Sender::ListBubbleSender	RObFu04-ls RObFu03-ls
MeteoBubble::MeteoBubbleConfigMenu	RObFu01-mt RObFu01.1-mt
MeteoBubble::MeteoBubbleCreator	RObFu01-mt
MeteoBubble::MeteoBubbleReceiver	RObFu01.3-mt RObFu01-mt
MeteoBubble::MeteoBubbleSender	RObFu01.3-mt RObFu01-mt
MeteoBubble::MeteoDelivery	RObFu01-mt RObFu01.2-mt
MeteoBubble::MeteoItem	RObFu01-mt
Monolith::Database::informationStorage:: Checks::check	RObFu11.2 RObFu22.1 RObFu11 RObFu22
Monolith::Database::informationStorage:: Checks::checkCreator	RObFu22 RObFu11.2 RObFu22.1 RObFu11
Monolith::Database::informationStorage:: Checks::checkDiscriminator	RObFu11 RObFu22 RObFu11.2 RObFu22.1
Monolith::Database::informationStorage:: Checks::concreteCheck	RObFu11.2 RObFu22.1 RObFu11 RObFu22
Monolith::Database::informationStorage:: Checks::concreteCheckCreator	RObFu11.2 RObFu22.1 RObFu11 RObFu22



Class	Requisiti
Monolith::UI::Bubbles::Bubble	RObFu11 RObFu11.2 RObFu11.2.1.7 RObFu10.3.1 RObFu11.1 RObFu11.2.1 RObFu10.2.5
Monolith::UI::Bubbles::BubbleConfig	RObFu11 RObFu11.2 RObFu11.2.1.6 RObFu10.2.4 RObFu11.1 RObFu11.2.1 RObFu11.2.1.7 RObFu10.2.3
Monolith::UI::Bubbles::BubbleCreationButton	RObFu11.1 RObFu11.2.1 RObFu11.2.1.7 RObFu10.2.1 RObFu11 RObFu11.2 RObFu11.2.1.6 RObFu10.2.2
Monolith::UI::Bubbles::ConcreteBubble	RObFu11.2.1.7 RObFu10.2.5 RObFu11 RObFu11.2.1 RObFu10.3.1 RObFu11.2
Monolith::UI::Bubbles::ConcreteBubbleConfig	RObFu11.2.1.7 RObFu10.2.4 RObFu11.2 RObFu11.2.1.6 RObFu10.2.3 RObFu11 RObFu11.2.1
Monolith::UI::Bubbles::ConcreteBubbleCreationButton	RObFu10.2.2 RObFu11.2 RObFu11.2.1.6 RObFu10.2.1 RObFu11 RObFu11.2.1 RObFu11.2.1.7



Class	Requisiti
Monolith::UI::Bubbles::bubbleCreator	RObFu10.2.3 RObFu10.3.1 RObFu11.1 RObFu11.2.1 RObFu10.2.2 RObFu11.2.1.7 RObFu10.2.5 RObFu11 RObFu11.2 RObFu10.2.1 RObFu11.2.1.6
Monolith::UI::Bubbles::bubbleDiscriminator	RObFu11.2 RObFu10.2.1 RObFu11.2.1.6 RObFu10.2.3 RObFu10.3.1 RObFu11.1 RObFu11.2.1 RObFu10.2.2 RObFu11.2.1.7 RObFu10.2.5 RObFu11
Monolith::UI::Bubbles::concreteBubbleCreator	RObFu11 RObFu10.2.3 RObFu11.2.1 RObFu10.3.1 RObFu11.2.1.7 RObFu10.2.1 RObFu11.2 RObFu10.2.5 RObFu11.2.1.6
Monolith::UI::SideAreas::SideArea1_pkg:: BubbleCreationMenu	RObFu11.2 RObFu11.2.1.7 RObFu10.2.1 RObFu11 RObFu11.2.1 RObFu10.2 RObFu10.2.2
Monolith::UI::SideAreas::SideArea1_pkg:: SentBubbleHistory	RObFu11.2.1 RObFu10.2.5 RObFu11.2 RObFu11.2.1.7 RObFu10.2 RObFu11





Class	Requisiti
Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1	RObFu11.2.1 RObFu10 RObFu10.2 RObFu10.2.7 RObFu11.2 RObFu11.2.1.7 RObFu10.1 RObFu10.2.5.1 RObFu11
Monolith::UI::SideAreas::SideArea2_pkg:: ReceivedBubbleHistory	RObFu11.2.1.7 RObFu10.3 RObFu11 RObFu11.2.1 RObFu10.3.1 RObFu11.2
Monolith::UI::SideAreas::SideArea2_pkg:: SideArea2	RObFu10.1 RObFu10.3.1.1 RObFu11.2 RObFu11.2.1.7 RObFu10 RObFu10.3 RObFu11 RObFu11.2.1
Monolith::UI::UI-Layouts:: ConditionalRendering	RObFu11 RObFu11.2.1 RObFu11.2.1.2.3 RObFu11.2 RObFu11.2.1.2 RObFu11.2.1.7
Monolith::UI::UI-Layouts::ContainedElement	RObFu11.2 RObFu11.2.1.2 RObFu11 RObFu11.2.1 RObFu11.2.1.7
Monolith::UI::UI-Layouts::HorizontalLayout	RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.2.1 RObFu11.2 RObFu11.2.1.2



Class	Requisiti
Monolith::UI::UI-Layouts::VerticalLayout	RObFu11.2 RObFu11.2.1.2 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.2.2
Monolith::UI::UI-SingleComponents:: CheckBoxList	RObFu11 RObFu11.2.1 RObFu11.2.1.1.5 RObFu11.2.1.3.6 RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.3 RObFu11.2.1.7
Monolith::UI::UI-SingleComponents:: CheckButton	RObFu23 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu11.2.1.7 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.3.6
Monolith::UI::UI-SingleComponents::ComboBox	RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23
Monolith::UI::UI-SingleComponents::Image	RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.3 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.1.2 RObFu11.2.1.3.3



Class	Requisiti
Monolith::UI::UI-SingleComponents::ImageButton	RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23
Monolith::UI::UI-SingleComponents::LabelComboBox	RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23
Monolith::UI::UI-SingleComponents::LabelEdit	RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23
Monolith::UI::UI-SingleComponents::LabelEditPushButton	RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23 RObFu11.2
Monolith::UI::UI-SingleComponents::LabelPushButton	RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23 RObFu11.2
Monolith::UI::UI-SingleComponents::LineEdit	RObFu11.2.1.7 RObFu11.2.1.1.3 RObFu11.2.1.3.2 RObFu11 RObFu11.2.1 RObFu11.2.1.1.1 RObFu23 RObFu11.2.1.3 RObFu11.2 RObFu11.2.1.1



Class	Requisiti
Monolith::UI::UI-SingleComponents:: LineEditComboBox	RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7
Monolith::UI::UI-SingleComponents:: LineEditPushButton	RObFu11 RObFu11.2.1 RObFu11.2.1.3 RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7
Monolith::UI::UI- SingleComponents::PushButton	RObFu11 RObFu23 RObFu11.2.1 RObFu11.2.1.1.4 RObFu11.2.1.3.4 RObFu11.2.1.7 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.3 RObFu11.2.1.3.5
Monolith::UI::UI-SingleComponents:: RadioButtonGroup	RObFu11.2.1.3.1 RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.3 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.1.6
Monolith::UI::UI-SingleComponents:: TextAreaButton	RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3



Class	Requisiti
Monolith::UI::UI-SingleComponents:: TextAreaComboBox	RObFu23 RObFu11.2 RObFu11.2.1.1 RObFu11.2.1.7 RObFu11 RObFu11.2.1 RObFu11.2.1.3
SurveyBubble::SurveyBubbleConfigMenu	RObFu01-sd
SurveyBubble::SurveyBubbleReceiver	RObFu02-sd RObFu04-sd
SurveyBubble::SurveyBubbleSender	RObFu04-sd
TranslationBubble::MessageTranslation	RObFu01.4-tr RObFu01-tr
TranslationBubble:: TranslationBubbleConfigMenu	RObFu01.2-tr RObFu01.1-tr RObFu01-tr
TranslationBubble::TranslationBubbleCreator	RObFu01-tr
TranslationBubble::TranslationBubbleReceiver	RObFu01-tr RObFu01.5-tr
TranslationBubble::TranslationBubbleSender	RObFu01-tr RObFu01.5-tr

Tabella 7: Tracciamento classi - requisiti

#### 7.4 Tracciamento requisiti-classi

Requisiti	Classi
RObFu01-cv	CurrencyBubble::CurrencyBubbleConfigMenu CurrencyBubble::CurrencyConversion CurrencyBubble::CurrencyBubbleCreator CurrencyBubble::CurrencyBubbleReceiver CurrencyBubble::CurrencyBubbleSender
RObFu01-dd	DiceBubble::DiceBubbleSender DiceBubble::DiceBubbleReceiver DiceBubble::DiceBubbleConfigMenu DiceBubble::DiceRoller DiceBubble::DiceBubbleCreator



Requisiti	Classi
RObFu01-ls	ListBubble::CheckListCreation:: CheckListCreator ListBubble::CheckListCreation:: CheckListItemsDefinition ListBubble::CheckListCreation:: CheckListComponent ListBubble::CheckListReading::CheckList ListBubble::CheckListReading:: ListOfCheckLists ListBubble::Configuration:: ListBubbleMenuConfig ListBubble::Configuration::ListCreationButton
RObFu01-mt	MeteoBubble::MeteoDelivery MeteoBubble::MeteoBubbleConfigMenu MeteoBubble::MeteoItem MeteoBubble::MeteoBubbleCreator MeteoBubble::MeteoBubbleReceiver MeteoBubble::MeteoBubbleSender
RObFu01-sd	SurveyBubble::SurveyBubbleConfigMenu
RObFu01-tr	TranslationBubble::TranslationBubbleReceiver TranslationBubble::TranslationBubbleSender TranslationBubble::TranslationBubbleCreator TranslationBubble::MessageTranslation TranslationBubble:: TranslationBubbleConfigMenu
RObFu01.1-cv	CurrencyBubble::CurrencyBubbleConfigMenu
RObFu01.1-dd	DiceBubble::DiceBubbleConfigMenu
RObFu01.1-ls	ListBubble::Configuration:: ListBubbleMenuConfig ListBubble::Configuration::ListCreationButton
RObFu01.1-mt	MeteoBubble::MeteoBubbleConfigMenu
RObFu01.1-tr	TranslationBubble:: TranslationBubbleConfigMenu
RObFu01.2-cv	CurrencyBubble::CurrencyBubbleConfigMenu
RObFu01.2-dd	DiceBubble::DiceBubbleReceiver DiceBubble::DiceBubbleSender
RObFu01.2-ls	ListBubble::CheckListReading:: ListOfCheckLists ListBubble::CheckListReading::CheckList
RObFu01.2-mt	MeteoBubble::MeteoDelivery
RObFu01.2-tr	TranslationBubble:: TranslationBubbleConfigMenu
ROpFu01.2.1-dd	DiceBubble::DiceBubbleSender DiceBubble::DiceBubbleReceiver
RObFu01.3-cv	CurrencyBubble::CurrencyConversion
RObFu01.3-dd	DiceBubble::DiceRoller
RObFu01.3-mt	MeteoBubble::MeteoBubbleReceiver MeteoBubble::MeteoBubbleSender



Requisiti	Classi
RObFu01.4-cv	CurrencyBubble::CurrencyBubbleReceiver CurrencyBubble::CurrencyBubbleSender
RObFu01.4-tr	TranslationBubble::MessageTranslation
RObFu01.5-tr	TranslationBubble::TranslationBubbleReceiver TranslationBubble::TranslationBubbleSender
ROpFu02-cv	CurrencyBubble::CurrencyConversion CurrencyBubble::CurrencyBubbleConfigMenu
RObFu02-ls	ListBubble::CheckListCreation:: CheckListComponent ListBubble::CheckListCreation:: CheckListCreator ListBubble::CheckListCreation:: CheckListItemsDefinition
RObFu02-sd	SurveyBubble::SurveyBubbleReceiver
RObFu03-ls	ListBubble::Receiver::ListBubbleReceiver ListBubble::Sender::ListBubbleSender
RObFu04-ls	ListBubble::Receiver::ListBubbleReceiver ListBubble::Sender::ListBubbleSender
RObFu04-sd	SurveyBubble::SurveyBubbleSender SurveyBubble::SurveyBubbleReceiver
RObFu10	Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1 Monolith::UI::SideAreas::SideArea2_pkg:: SideArea2
RObFu10.1	Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1 Monolith::UI::SideAreas::SideArea2_pkg:: SideArea2
RObFu10.2	Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1 Monolith::UI::SideAreas::SideArea1_pkg:: SentBubbleHistory Monolith::UI::SideAreas::SideArea1_pkg:: BubbleCreationMenu
RObFu10.2.1	Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::BubbleCreationButton Monolith::UI::Bubbles::concreteBubbleCreator Monolith::UI::Bubbles:: ConcreteBubbleCreationButton Monolith::UI::Bubbles::bubbleCreator Monolith::UI::SideAreas::SideArea1_pkg:: BubbleCreationMenu



Requisiti	Classi
RObFu10.2.2	Monolith::UI::Bubbles::ConcreteBubbleCreationButton Monolith::UI::Bubbles::bubbleCreator Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::BubbleCreationButton Monolith::UI::SideAreas::SideArea1_pkg::BubbleCreationMenu
RObFu10.2.3	Monolith::UI::Bubbles::bubbleCreator Monolith::UI::Bubbles::concreteBubbleCreator Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::ConcreteBubbleConfig Monolith::UI::Bubbles::BubbleConfig
RObFu10.2.4	Monolith::UI::Bubbles::ConcreteBubbleConfig Monolith::UI::Bubbles::BubbleConfig
RObFu10.2.5	Monolith::UI::Bubbles::ConcreteBubble Monolith::UI::Bubbles::bubbleCreator Monolith::UI::Bubbles::concreteBubbleCreator Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::Bubble Monolith::UI::SideAreas::SideArea1_pkg::SentBubbleHistory
RObFu10.2.5.1	Monolith::UI::SideAreas::SideArea1_pkg::SideArea1
RObFu10.2.7	Monolith::UI::SideAreas::SideArea1_pkg::SideArea1
RObFu10.3	Monolith::UI::SideAreas::SideArea2_pkg::ReceivedBubbleHistory Monolith::UI::SideAreas::SideArea2_pkg::SideArea2
RObFu10.3.1	Monolith::UI::Bubbles::bubbleCreator Monolith::UI::Bubbles::concreteBubbleCreator Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::Bubble Monolith::UI::Bubbles::ConcreteBubble Monolith::UI::SideAreas::SideArea2_pkg::ReceivedBubbleHistory
RObFu10.3.1.1	Monolith::UI::SideAreas::SideArea2_pkg::SideArea2





Requisiti	Classi
RObFu11	Monolith::Database::informationStorage:: Checks::checkDiscriminator
	Monolith::Database::informationStorage:: Checks::concreteCheck
	Monolith::Database::informationStorage:: Checks::check
	Monolith::Database::informationStorage:: Checks::concreteCheckCreator
	Monolith::Database::informationStorage:: Checks::checkCreator
	Monolith::UI::Bubbles::concreteBubbleCreator
	Monolith::UI::Bubbles::BubbleConfig
	Monolith::UI::Bubbles::Bubble
	Monolith::UI::Bubbles::ConcreteBubble
	Monolith::UI::Bubbles::BubbleCreationButton
	Monolith::UI::Bubbles::bubbleCreator
	Monolith::UI::Bubbles:: ConcreteBubbleCreationButton
	Monolith::UI::Bubbles::ConcreteBubbleConfig
	Monolith::UI::Bubbles::bubbleDiscriminator
	Monolith::UI::SideAreas::SideArea1_pkg:: BubbleCreationMenu
	Monolith::UI::SideAreas::SideArea1_pkg:: SentBubbleHistory
	Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1
	Monolith::UI::SideAreas::SideArea2_pkg:: ReceivedBubbleHistory
	Monolith::UI::SideAreas::SideArea2_pkg:: SideArea2
	Monolith::UI::UI-Layouts:: ConditionalRendering
	Monolith::UI::UI-Layouts::HorizontalLayout
	Monolith::UI::UI-Layouts::ContainedElement
	Monolith::UI::UI-Layouts::VerticalLayout
	Monolith::UI::UI-SingleComponents:: CheckBoxList
	Monolith::UI::UI-SingleComponents::PushButton
	Monolith::UI::UI-SingleComponents:: CheckButton
	Monolith::UI::UI-SingleComponents:: LineEditPushButton
	Monolith::UI::UI-SingleComponents:: LineEditComboBox



Requisiti	Classi
RObFu11	Monolith::UI::UI-SingleComponents::LineEdit Monolith::UI::UI-SingleComponents:: LabelPushButton Monolith::UI::UI-SingleComponents:: LabelEditPushButton Monolith::UI::UI-SingleComponents::LabelEdit Monolith::UI::UI-SingleComponents::Image Monolith::UI::UI-SingleComponents:: LabelComboBox Monolith::UI::UI-SingleComponents::ComboBox Monolith::UI::UI-SingleComponents:: ImageButton Monolith::UI::UI-SingleComponents:: RadioButtonGroup Monolith::UI::UI-SingleComponents:: TextAreaComboBox Monolith::UI::UI-SingleComponents:: TextAreaButton
RObFu11.1	Monolith::UI::Bubbles::BubbleCreationButton Monolith::UI::Bubbles::bubbleCreator Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::BubbleConfig Monolith::UI::Bubbles::Bubble



Requisiti	Classi
RObFu11.2	Monolith::Database::informationStorage:: Checks::concreteCheck
	Monolith::Database::informationStorage:: Checks::check
	Monolith::Database::informationStorage:: Checks::concreteCheckCreator
	Monolith::Database::informationStorage:: Checks::checkCreator
	Monolith::Database::informationStorage:: Checks::checkDiscriminator
	Monolith::UI::Bubbles::bubbleDiscriminator
	Monolith::UI::Bubbles::BubbleConfig
	Monolith::UI::Bubbles::Bubble
	Monolith::UI::Bubbles:: ConcreteBubbleCreationButton
	Monolith::UI::Bubbles::ConcreteBubbleConfig
	Monolith::UI::Bubbles::concreteBubbleCreator
	Monolith::UI::Bubbles::BubbleCreationButton
	Monolith::UI::Bubbles::bubbleCreator
	Monolith::UI::Bubbles::ConcreteBubble
	Monolith::UI::SideAreas::SideArea1_pkg:: BubbleCreationMenu
	Monolith::UI::SideAreas::SideArea1_pkg:: SentBubbleHistory
	Monolith::UI::SideAreas::SideArea1_pkg:: SideArea1
	Monolith::UI::SideAreas::SideArea2_pkg:: SideArea2
	Monolith::UI::SideAreas::SideArea2_pkg:: ReceivedBubbleHistory
	Monolith::UI::UI-Layouts::ContainedElement
	Monolith::UI::UI-Layouts::VerticalLayout
	Monolith::UI::UI-Layouts:: ConditionalRendering
	Monolith::UI::UI-Layouts::HorizontalLayout
	Monolith::UI::UI-SingleComponents::LabelEdit
	Monolith::UI::UI-SingleComponents::Image
	Monolith::UI::UI-SingleComponents:: LabelComboBox
	Monolith::UI::UI-SingleComponents::ComboBox
	Monolith::UI::UI-SingleComponents:: ImageButton



Requisiti	Classi
RObFu11.2	Monolith::UI::UISingleComponents:: RadioButtonGroup
	Monolith::UI::UISingleComponents:: TextAreaComboBox
	Monolith::UI::UISingleComponents:: TextAreaButton
	Monolith::UI::UISingleComponents:: CheckBoxList
	Monolith::UI::UISingleComponents::PushButton
	Monolith::UI::UISingleComponents:: CheckBox
	Monolith::UI::UISingleComponents:: LineEditPushButton
	Monolith::UI::UISingleComponents:: LineEditComboBox
	Monolith::UI::UISingleComponents::LineEdit
	Monolith::UI::UISingleComponents:: LabelPushButton
	Monolith::UI::UISingleComponents:: LabelEditPushButton



Requisiti	Classi
RObFu11.2.1	Monolith::UI::Bubbles::concreteBubbleCreator
	Monolith::UI::Bubbles::BubbleCreationButton
	Monolith::UI::Bubbles::bubbleCreator
	Monolith::UI::Bubbles::ConcreteBubble
	Monolith::UI::Bubbles::bubbleDiscriminator
	Monolith::UI::Bubbles::BubbleConfig
	Monolith::UI::Bubbles::Bubble
	Monolith::UI::Bubbles::
	ConcreteBubbleCreationButton
	Monolith::UI::Bubbles::ConcreteBubbleConfig
	Monolith::UI::SideAreas::SideArea1_pkg::
	SentBubbleHistory
	Monolith::UI::SideAreas::SideArea1_pkg::
	SideArea1
	Monolith::UI::SideAreas::SideArea1_pkg::
	BubbleCreationMenu
	Monolith::UI::SideAreas::SideArea2_pkg::
	ReceivedBubbleHistory
	Monolith::UI::SideAreas::SideArea2_pkg::
	SideArea2
	Monolith::UI::UI-Layouts::
	ConditionalRendering
	Monolith::UI::UI-Layouts::HorizontalLayout
	Monolith::UI::UI-Layouts::ContainedElement
	Monolith::UI::UI-Layouts::VerticalLayout
	Monolith::UI::UI-SingleComponents::
	CheckBoxList
	Monolith::UI::UI-SingleComponents::PushButton
	Monolith::UI::UI-SingleComponents::
	CheckButton
	Monolith::UI::UI-SingleComponents::
	LineEditPushButton
	Monolith::UI::UI-SingleComponents::
	LineEditComboBox
	Monolith::UI::UI-SingleComponents::LineEdit
	Monolith::UI::UI-SingleComponents::
	LabelPushButton
	Monolith::UI::UI-SingleComponents::
	LabelEditPushButton
	Monolith::UI::UI-SingleComponents::LabelEdit
	Monolith::UI::UI-SingleComponents::Image



Requisiti	Classi
RObFu11.2.1	Monolith::UI::UI-SingleComponents:: LabelComboBox Monolith::UI::UI-SingleComponents::ComboBox Monolith::UI::UI-SingleComponents:: ImageButton Monolith::UI::UI-SingleComponents:: RadioButtonGroup Monolith::UI::UI-SingleComponents:: TextAreaComboBox Monolith::UI::UI-SingleComponents:: TextAreaButton
RObFu11.2.1.1	Monolith::UI::UI-SingleComponents:: LabelPushButton Monolith::UI::UI-SingleComponents:: LabelEditPushButton Monolith::UI::UI-SingleComponents::LabelEdit Monolith::UI::UI-SingleComponents::Image Monolith::UI::UI-SingleComponents:: LabelComboBox Monolith::UI::UI-SingleComponents::ComboBox Monolith::UI::UI-SingleComponents:: ImageButton Monolith::UI::UI-SingleComponents:: RadioButtonGroup Monolith::UI::UI-SingleComponents:: TextAreaComboBox Monolith::UI::UI-SingleComponents:: TextAreaButton Monolith::UI::UI-SingleComponents:: CheckBoxList Monolith::UI::UI-SingleComponents::PushButton Monolith::UI::UI-SingleComponents:: CheckButton Monolith::UI::UI-SingleComponents:: LineEditPushButton Monolith::UI::UI-SingleComponents:: LineEditComboBox Monolith::UI::UI-SingleComponents::LineEdit
RObFu11.2.1.1.1	Monolith::UI::UI-SingleComponents::LineEdit
RObFu11.2.1.1.2	Monolith::UI::UI-SingleComponents::Image
RObFu11.2.1.1.3	Monolith::UI::UI-SingleComponents::LineEdit
RObFu11.2.1.1.4	Monolith::UI::UI-SingleComponents::PushButton
RObFu11.2.1.1.5	Monolith::UI::UI-SingleComponents:: CheckBoxList
RObFu11.2.1.1.6	Monolith::UI::UI-SingleComponents:: RadioButtonGroup



Requisiti	Classi
RObFu11.2.1.2	Monolith::UI::UI-Layouts::ContainedElement Monolith::UI::UI-Layouts::VerticalLayout Monolith::UI::UI-Layouts:: ConditionalRendering Monolith::UI::UI-Layouts::HorizontalLayout
RObFu11.2.1.2.1	Monolith::UI::UI-Layouts::HorizontalLayout
RObFu11.2.1.2.2	Monolith::UI::UI-Layouts::VerticalLayout
RObFu11.2.1.2.3	Monolith::UI::UI-Layouts:: ConditionalRendering
RObFu11.2.1.3	Monolith::UI::UI-SingleComponents:: CheckBox Monolith::UI::UI-SingleComponents:: LineEditPushButton Monolith::UI::UI-SingleComponents::Image Monolith::UI::UI-SingleComponents:: LineEditComboBox Monolith::UI::UI-SingleComponents:: RadioButtonGroup Monolith::UI::UI-SingleComponents:: LabelPushButton Monolith::UI::UI-SingleComponents:: LabelEditPushButton Monolith::UI::UI-SingleComponents::LineEdit Monolith::UI::UI-SingleComponents::LabelEdit Monolith::UI::UI-SingleComponents:: LabelComboBox Monolith::UI::UI-SingleComponents:: CheckBoxList Monolith::UI::UI-SingleComponents::ComboBox Monolith::UI::UI-SingleComponents::PushButton Monolith::UI::UI-SingleComponents:: ImageButton Monolith::UI::UI-SingleComponents:: TextAreaComboBox Monolith::UI::UI-SingleComponents:: TextAreaButton
RObFu11.2.1.3.1	Monolith::UI::UI-SingleComponents:: RadioButtonGroup
RObFu11.2.1.3.2	Monolith::UI::UI-SingleComponents::LineEdit
RObFu11.2.1.3.3	Monolith::UI::UI-SingleComponents::Image
RObFu11.2.1.3.4	Monolith::UI::UI-SingleComponents::PushButton
RObFu11.2.1.3.5	Monolith::UI::UI-SingleComponents::PushButton

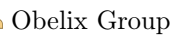


Requisiti	Classi
RObFu11.2.1.3.6	Monolith::UI::UI-SingleComponents:: CheckBoxList Monolith::UI::UI-SingleComponents:: CheckButton
RObFu11.2.1.6	Monolith::UI::Bubbles::bubbleDiscriminator Monolith::UI::Bubbles::BubbleConfig Monolith::UI::Bubbles:: ConcreteBubbleCreationButton Monolith::UI::Bubbles::ConcreteBubbleConfig Monolith::UI::Bubbles::concreteBubbleCreator Monolith::UI::Bubbles::BubbleCreationButton Monolith::UI::Bubbles::bubbleCreator





Requisiti	Classi
RObFu11.2.1.7	Monolith::UI::Bubbles::ConcreteBubbleConfig
	Monolith::UI::Bubbles::ConcreteBubble
	Monolith::UI::Bubbles::concreteBubbleCreator
	Monolith::UI::Bubbles::BubbleCreationButton
	Monolith::UI::Bubbles::Bubble
	Monolith::UI::Bubbles::bubbleCreator
	Monolith::UI::Bubbles::bubbleDiscriminator
	Monolith::UI::Bubbles::BubbleConfig
	Monolith::UI::Bubbles::
	ConcreteBubbleCreationButton
	Monolith::UI::SideAreas::SideArea1_pkg::
	BubbleCreationMenu
	Monolith::UI::SideAreas::SideArea1_pkg::
	SentBubbleHistory
	Monolith::UI::SideAreas::SideArea1_pkg::
	SideArea1
	Monolith::UI::SideAreas::SideArea2_pkg::
	ReceivedBubbleHistory
	Monolith::UI::SideAreas::SideArea2_pkg::
	SideArea2
	Monolith::UI::UI-Layouts::HorizontalLayout
	Monolith::UI::UI-Layouts::VerticalLayout
	Monolith::UI::UI-Layouts::ContainedElement
	Monolith::UI::UI-Layouts::
	ConditionalRendering
	Monolith::UI::UI-SingleComponents::LineEdit
	Monolith::UI::UI-SingleComponents::
	LabelPushButton
	Monolith::UI::UI-SingleComponents::
	LabelEditPushButton
	Monolith::UI::UI-SingleComponents::LabelEdit
	Monolith::UI::UI-SingleComponents::
	LabelComboBox
	Monolith::UI::UI-SingleComponents::ComboBox
	Monolith::UI::UI-SingleComponents::
	ImageButton
	Monolith::UI::UI-SingleComponents::
	CheckButton
	Monolith::UI::UI-SingleComponents::Image
	Monolith::UI::UI-SingleComponents::
	TextAreaComboBox



Requisiti	Classi
RObFu11.2.1.7	Monolith::UI::UISingleComponents::PushButton Monolith::UI::UISingleComponents::TextAreaButton Monolith::UI::UISingleComponents::RadioButtonGroup Monolith::UI::UISingleComponents::LineEditPushButton Monolith::UI::UISingleComponents::CheckBoxList Monolith::UI::UISingleComponents::LineEditComboBox
RObFu22	Monolith::Database::informationStorage::Checks::checkCreator Monolith::Database::informationStorage::Checks::checkDiscriminator Monolith::Database::informationStorage::Checks::concreteCheck Monolith::Database::informationStorage::Checks::check Monolith::Database::informationStorage::Checks::concreteCheckCreator
RObFu22.1	Monolith::Database::informationStorage::Checks::concreteCheck Monolith::Database::informationStorage::Checks::check Monolith::Database::informationStorage::Checks::concreteCheckCreator Monolith::Database::informationStorage::Checks::checkCreator Monolith::Database::informationStorage::Checks::checkDiscriminator



Requisiti	Classi
RObFu23	Monolith::UI::UISingleComponents::
	CheckBox
	Monolith::UI::UISingleComponents::Image
	Monolith::UI::UISingleComponents::
	TextAreaComboBox
	Monolith::UI::UISingleComponents::PushButton
	Monolith::UI::UISingleComponents::
	TextAreaButton
	Monolith::UI::UISingleComponents::
	RadioButtonGroup
	Monolith::UI::UISingleComponents::
	LineEditPushButton
	Monolith::UI::UISingleComponents::
	CheckBoxList
	Monolith::UI::UISingleComponents::
	LineEditComboBox
	Monolith::UI::UISingleComponents::LineEdit
	Monolith::UI::UISingleComponents::
	LabelPushButton
	Monolith::UI::UISingleComponents::
	LabelEditPushButton
	Monolith::UI::UISingleComponents::LabelEdit
	Monolith::UI::UISingleComponents::
	LabelComboBox
	Monolith::UI::UISingleComponents::ComboBox
	Monolith::UI::UISingleComponents::
	ImageButton

Tabella 9: Tracciamento requisiti - classi

## A Descrizione Design Pattern

Un design pattern è una soluzione progettuale elegante e generale ad un problema ricorrente. In particolare si tratta di un modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante la fase di progettazione e sviluppo del software, ancora prima della definizione dell'algoritmo risolutivo.

Essi si suddividono in quattro categorie:

- **Architetturali** : esprimono schemi di base per impostare l'organizzazione strutturale di un sistema software;
- **Creazionali** : forniscono un'astrazione del processo di istanziazione degli oggetti;
- **Strutturali** : si occupano delle modalità di composizione di classi e oggetti per formare strutture complesse;



- **Comportamentali** : si occupano di algoritmi e dell'assegnamento di responsabilità tra oggetti collaboranti.

## A.1 Design Pattern Utilizzati

### A.1.1 Factory Method

Rappresenta uno dei pattern creazionali adottati dal gruppo Obelix, esso indirizza il problema della creazione di oggetti senza specificarne l'esatta classe. Questo pattern raggiunge il suo scopo fornendo un'interfaccia per creare un oggetto, ma lascia che le sottoclassi decidano quale oggetto istanziare.

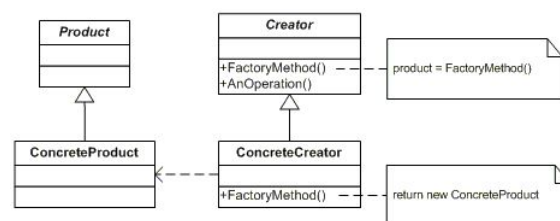


Figura 100: Diagramma del Factory method

Product definisce l'interfaccia implementata da ConcreteProduct, creator definisce il factory method che restituisce una interfaccia di tipo product, ConcreteCreator definisce il metodo factory effettivo per la creazione di un'istanza particolare di tipo Product. Utilizzato in tal modo nella classe BubbleCreator dal gruppo Obelix.

I motivi che portano alla scelta del suo utilizzo sono:

- La creazione di un oggetto preclude il suo riuso senza una significativa duplicazione di codice
- La creazione di un oggetto richiede l'accesso ad informazioni o risorse che non dovrebbero essere contenute nella classe di composizione
- La gestione del ciclo di vita degli oggetti gestiti deve essere centralizzata in modo da assicurare un comportamento coerente all'interno dell'applicazione

### A.1.2 Template Method

Questo pattern permette di definire la struttura di un algoritmo lasciando alle sottoclassi il compito di implementarne alcuni passi come preferiscono. In questo modo si può ridefinire e personalizzare parte del comportamento nelle varie sottoclassi senza dover riscrivere più volte il codice in comune.

Questo pattern è composto dai seguenti partecipanti:

- **AbstractClass**: definisce il metodo concreto ed i metodi primitivi astratti. Il metodo concreto richiama i metodi primitivi implementati nelle sottoclassi
- **ConcreteClass**: implementa i metodi primitivi per svolgere i passi specifici dell'algoritmo ed eventualmente i metodi hook

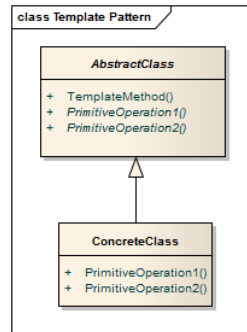


Figura 101: Diagramma del Template method

Tale pattern presenta i seguenti vantaggi/svantaggi:

- definizione di metodi comuni: occorre comprendere quali sono i comportamenti comunemente implementati dalle sottoclassi e “promuoverli” come comportamenti della superclasse in modo da centralizzarli ed evitare la duplicazione del codice.
- definizione dei metodi hook: occorre permettere alle sottoclassi di poter definire un comportamento durante l’esecuzione dell’algoritmo tramite l’introduzione di un metodo “hook” che POSSONO ma NON SONO OBBLIGATE a ridefinire.

Utilizzato dal gruppo nella classe BubbleDatabase con una funzione polimorfa che ritorna il nome della bolla in base all’istanza. Questa scelta è dovuta al fatto di dover sviluppare comportamenti comuni a più bolle, quindi si è deciso di far in modo da centralizzarli ed evitare la duplicazione del codice.

### A.1.3 Proxy Pattern

Invece che chiamare direttamente i metodi di un oggetto li facciamo chiamare da un terzo attore che si occupa di invocare concretamente i metodi sull’oggetto e di restituirci i valori in caso di metodi con valori di ritorno. Di solito questo è utilizzato quando esiste una dipendenza da una risorsa remota (con conseguente latenza di rete) o quando un oggetto richiede molto tempo per caricare. Quindi si applica il modello Proxy e si crea un oggetto proxy che “si accende” per l’oggetto originale. Il proxy inoltra la richiesta a un oggetto di destinazione. L’interfaccia dell’oggetto Proxy è uguale all’oggetto originale e i client potrebbero non essere nemmeno consapevoli di trattare con un proxy piuttosto che l’oggetto reale

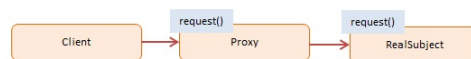


Figura 102: Diagramma del Proxy Pattern

Partecipanti:

- Client



- Chiama Proxy richiedendo un'operazione
- Proxy
  - Fornisce un'interfaccia simile all'oggetto reale
  - Mantiene un riferimento che consente l'accesso proxy all'oggetto reale
  - Gestisce le richieste e li inoltra all'oggetto reale
- RealSubject
  - Definisce l'oggetto reale per cui è richiesto il servizio

Meteor call utilizza il Pattern Proxy, infatti simula lato client una risorsa che si trova lato server gestendo in modo trasparente la comunicazione tra le due.

#### A.1.4 Composite Pattern

Questo pattern permette di trattare un gruppo di oggetti come se fossero l'istanza di un oggetto singolo. Il design pattern Composite organizza gli oggetti in una struttura ad albero, nella quale i nodi sono delle composite e le foglie sono oggetti semplici.

È utilizzato per dare la possibilità ai client di manipolare oggetti singoli e composizioni in modo uniforme.

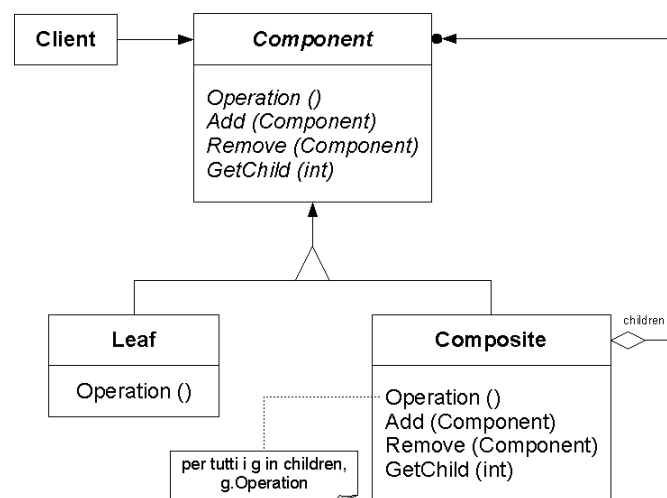


Figura 103: Diagramma del Composite Pattern

Struttura:

- Client: manipola gli oggetti attraverso l'interfaccia Component
- Component: dichiara l'interfaccia per gli oggetti nella composizione, per l'accesso e la manipolazione di questi, imposta un comportamento di default per l'interfaccia comune a tutte le classi e può definire un'interfaccia per l'accesso al padre del componente e la implementa se è appropriata



- Composite: definisce il comportamento per i componenti aventi figli, salva i figli e implementa le operazioni ad essi connesse nell'interfaccia Component
- Leaf: definisce il comportamento degli oggetti primitivi, cioè delle foglie.

Attraverso l'interfaccia Component, il Client interagisce con gli oggetti della composite. Se l'oggetto desiderato è una Leaf, la richiesta è processata direttamente; altrimenti, se è una Composite, viene rimandata ai figli cercando di svolgere le operazioni prima e dopo del rimando.

In questo modo, si semplifica il Client, si creano delle gerarchie di classi, si semplifica l'aggiunta di nuovi componenti, anche se il design diventa troppo generale.

Questo Design Pattern è adottato da React, libreria utilizzata dal gruppo Obelix all'interno di Monolith.