

404NotFound

Premi: better than Prezi.



Specifica Tecnica

Versione	1.0
Redazione	Vegro Federico Cossu Mattia Camborata Marco Manuto Monica Rettore Andrea Gobbo Ismaele De Lazzari Enrico
Verifica	Manuto Monica Rettore Andrea
Responsabile	Gobbo Ismaele
Uso	Esterno
Stato	Formale
Ultima modifica	XXX
Lista di distribuzione	404NotFound prof. Tullio Vardanega prof. Riccardo Cardin Zucchetti S.p.a.

Registro delle modifiche

Versione	Autore	Data	Descrizione
1.0	Autore vers 1.0	12-01-2015	Scrittura versione finale ecc.
0.1	Autore vers 0.1	19-12-2014	Stesura scheletro ecc.

Tabella 1: Storico versioni del documento.

Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo del prodotto	6
1.3	Glossario	6
2	Definizione del Prodotto	7
2.1	Metodo e formalismo di specifica	7
2.2	Presentazione dell'architettura generale del sistema	7
3	Diagrammi dei Package	7
4	Descrizione dei singoli componenti	8
4.1	Premi	8
4.1.1	Premi.Utility	8
4.1.2	Premi.View	8
4.1.3	Premi.PremiCtrl	9
4.2	Premi.UserManager	9
4.2.1	Premi.UserManager.User	9
4.2.2	Premi.UserManager.View	9
4.2.3	Premi.UserManager.UserManagerCtrl	10
4.2.4	Premi.UserManager.LoginView	10
4.2.5	Premi.UserManager.LoginController	10
4.2.6	Premi.UserManager.RegistrationView	10
4.2.7	Premi.UserManager.RegistrationController	11
4.2.8	Premi.UserManager.ChangePasswordView	11
4.2.9	Premi.UserManager.ChangePasswordController	11
4.3	Premi.Viewer	11
4.3.1	Premi.Viewer.View	11
4.3.2	Premi.ViewerCtrl	12
4.4	Premi.Presentation	12
4.4.1	Premi.Presentation.GraphicObject	12
4.4.2	Premi.Presentation.GoContent	12
4.4.3	Premi.Presentation.Text	12
4.4.4	Premi.Presentation.Image	13
4.4.5	Premi.Presentation.Shape	13
4.4.6	Premi.Presentation.Frame	13
4.4.7	Premi.Presentation.Trail	13
4.4.8	Premi.Presentation.Presentation	14
4.5	Premi.PresentationManager	14
4.5.1	Premi.PresentationManager.Export	14
4.5.2	Premi.PresentationManager.Portable	14
4.5.3	Premi.PresentationManager.View	15
4.5.4	Premi.PresentationManager.PresentationManagerCtrl	15
4.5.5	Premi.PresentationManager.RemoveView	15
4.5.6	Premi.PresentationManager.RemoveController	15
4.5.7	Premi.PresentationManager.PublicView	16

4.5.8	Premi.PresentationManager.PublicController	16
4.5.9	Premi.PresentationManager.EditTitleDescrView	16
4.5.10	Premi.PresentationManager.EditTitleDescrController	16
4.5.11	Premi.PresentationManager.NewView	17
4.5.12	Premi.PresentationManager.NewController	17
4.5.13	Premi.PresentationManager.ListPresView	17
4.5.14	Premi.PresentationManager.ListPresController	17
4.6	Premi.Editor	18
4.6.1	Premi.Editor.View	18
4.6.2	Premi.Editor.EditorCtrl	18
4.6.3	Premi.Editor.TextView	18
4.6.4	Premi.Editor.TextController	18
4.6.5	Premi.Editor.ShapeView	19
4.6.6	Premi.Editor.ShapeController	19
4.6.7	Premi.Editor.ImageView	19
4.6.8	Premi.Editor.ImageController	19
4.6.9	Premi.Editor.StyleView	19
4.6.10	Premi.Editor.StyleController	20
4.6.11	Premi.Editor.FrameListView	20
4.7	Premi.Editor.FrameEditor	20
4.7.1	Premi.Editor.FrameEditor.View	20
4.7.2	Premi.Editor.FrameEditor.FrameEditorCtrl	20
4.7.3	Premi.Editor.FrameEditor.FrameListController	21
4.8	Premi.Editor.InfographicEditor	21
4.8.1	Premi.Editor.InfographicEditor.View	21
4.8.2	Premi.Editor.InfographicEditor.InfographicEditorCtrl	21
4.8.3	Premi.Editor.InfographicEditor.FrameListController	21
4.9	Premi.Editor.TrailsEditor	22
4.9.1	Premi.Editor.TrailsEditor.View	22
4.9.2	Premi.Editor.TrailsEditor.TrailsEditorCtrl	22
4.9.3	Premi.Editor.TrailsEditor.FrameListController	22
4.9.4	Premi.Editor.TrailsEditor.EditTrailView	22
4.9.5	Premi.Editor.TrailsEditor.EditTrailController	23
4.9.6	Premi.Editor.TrailsEditor.NewView	23
4.9.7	Premi.Editor.TrailsEditor.NewController	23
4.9.8	Premi.Editor.TrailsEditor.EditTitleView	23
4.9.9	Premi.Editor.TrailsEditor.EditTitleController	24
4.9.10	Premi.Editor.TrailsEditor.RemoveView	24
4.9.11	Premi.Editor.TrailsEditor.RemoveController	24
5	Diagrammi delle attività	24
5.1	Attività principali	25
5.2	Lista presentazioni	25
5.3	Login	27
5.4	Registrazione	28
5.5	Cambio password	29
5.6	Visualizzatore	30

5.7	Esegui presentazione proprietario	31
5.8	Esegui presentazione non proprietario	33
5.9	Creazione presentazione	34
5.10	Modifica titolo e descrizione presentazione	35
5.11	Pubblicazione presentazione	36
5.12	Elimina presentazione	37
5.13	Esportazione presentazione	38
5.14	Rendi portable	39
5.15	Modifica presentazione	40
5.16	Frame editor	41
5.17	Modifica frame	43
5.18	Infografica editor	44
5.19	Modifica infografica	45
5.20	Editor percorsi	46
5.21	Modifica percorso	47
5.22	Aggiungi frame	48
5.23	Rimuovi frame da percorso	49
6	Stime di fattibilità e di bisogno di risorse	49
7	Tracciamento della relazione componenti - requisiti	49
8	Descrizione delle tecnologie utilizzate	49
8.1	JavaScript	49
8.2	HTML5	50
8.3	CSS3	50
8.4	Angular	50
8.5	Meteor	52
9	Mockup Interfaccia [FACOLTATIVO]	53

Elenco delle tabelle

1	Storico versioni del documento.	1
---	---	---

Elenco delle figure

1	Attività principali	25
2	Lista presentazioni	26
3	Login utente	27
4	Registrazione utente	28
5	Cambio password	29
6	Visualizzatore	30
7	Esegui presentazione proprietario	31
8	Esegui presentazione non proprietario	33
9	Creazione presentazione	34
10	Modifica titolo e descrizione della presentazione	35
11	Pubblicazione presentazione	36
12	Elimina presentazione	37
13	Esportazione presentazione	38
14	Rendi portable	39
15	Modifica presentazione	40
16	Frame editor	42
17	Modifica frame	43
18	Infographic editor	44
19	Modifica infografica	45
20	Editor percorsi	46
21	Modifica percorso	47
22	Aggiungi frame	48
23	Rimuovi frame dal percorso	49

1 Introduzione

1.1 Scopo del documento

Questo documento definisce la progettazione ad alto livello di Premi. Viene prima descritta la struttura generale del sistema e successivamente vengono analizzate le varie componenti software in relazione alle loro attività principali. Segue poi la descrizione delle tecnologie e dei Design Pattern_G utilizzati, e un mockup_G dell'interfaccia grafica lato utente.

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un software di presentazione di slide non basato sul modello di PowerPoint_G, sviluppato in tecnologia HTML5_G e che funzioni sia su desktop che su dispositivo mobile. Il software dovrà permettere la creazione da parte dell'autore e la successiva presentazione del lavoro, fornendo effetti grafici di supporto allo storytelling e alla creazione di mappe mentali.

1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio e ai termini utilizzati nei documenti formali tutti i termini e gli acronimi presenti nel seguente documento che necessitano di definizione saranno seguiti da una "G" in pedice e saranno riportati in un documento esterno denominato Glossario.pdf. Tale documento accompagna e completa il presente e consiste in un listato ordinato di termini e acronimi con le rispettive definizioni e spiegazioni.

2 Definizione del Prodotto

2.1 Metodo e formalismo di specifica

Verrà qui esposta l'architettura di Premi ad alto livello seguendo un approccio top-down_G: verranno prima descritti i package_G e le loro dipendenze e successivamente le singole classi contenute al loro interno. I diagrammi delle classi e dei package_G seguono il formalismo UML_G2.0 e la struttura dei package segue una prassi ("best practice_G") di AngularJS_G che propone una suddivisione dei componenti per funzionalità dell'applicazione in alternativa alla classica suddivisione Model-View-Controller_G, più difficile da mantenere per applicazioni di medie o grandi dimensioni. Per ulteriori approfondimenti consultare la guida al sito scotch.io oppure il tutorial di [urigo:angular-meteor](http://urigo.angular-meteor.com). Si illustreranno poi i Design Pattern utilizzati nella fase di progettazione ad alto livello e si descriveranno le interazioni dell'utente con l'applicazione attraverso i diagrammi di attività_G.

2.2 Presentazione dell'architettura generale del sistema

I componenti sono stati suddivisi prima in base al loro contributo a specifiche funzionalità del software e solo successivamente per appartenenza ai ruoli del pattern MVC_G. Questo aumenta la chiarezza espositiva dei diagrammi, evita la creazione di package_G contenenti un numero eccessivo di classi e aiuta a compiere verifiche mirate a singoli componenti.

È importante specificare che il framework AngularJS unisce view e controller attraverso una dichiarazione esterna a entrambi, che fa parte del meccanismo detto di *routing* o di reindirizzamento dell'utente; view e controller inoltre non fanno di essere collegati tra loro e comunicano attraverso un oggetto chiamato *\$scope*. Questo rende l'architettura sia di tipo Model-View-Controller_G che di tipo Model-View-ViewModel_G. Per motivi di leggibilità *\$scope* e *routing* non verranno rappresentati in modo esplicito nei diagrammi dei package e delle classi di questo documento, ma sono comunque da considerarsi impliciti nelle dipendenze tra i view e controller dei componenti.

3 Diagrammi dei Package

Di seguito vengono descritti componenti principali del sistema e le loro dipendenze.

_____pkg.jpg

L'applicazione è costituita da un solo package_G principale chiamato **Premi**; al suo interno sono presenti:

- **Premi** racchiude tutti i package_G e la view e il controller principali;
- **Premi.UserManager** è il package_G di gestione dei dati dell'utente;
- **Premi.Viewer** racchiude gli elementi necessari alla visualizzazione della presentazione nei vari contesti previsti;
- **Premi.Presentation** racchiude la struttura generale della presentazione;

- `Premi.PresentationManager` contiene gli elementi necessari alla gestione delle presentazioni da parte dell'utente;
- `Premi.Editor` è il `packageG` dedicato alla modifica interna delle presentazioni; possiede al suo interno tre ulteriori `packageG`:
 - `Premi.Editor.FrameEditor` si occupa di creare, modificare o cancellare i `FrameG` contenuti nella presentazione;
 - `Premi.Editor.InfographicEditor` posiziona `FrameG` o altri elementi all'interno di un poster;
 - `Premi.Editor.TrailsEditor` ordina i `Frame` per la creazione di uno o più percorsi di presentazione.

4 Descrizione dei singoli componenti

Tipo, obiettivo e funzione del componente

Relazioni d'uso di altre componenti

Interfacce con le relazioni di uso da altre componenti

4.1 Premi

—————premi-class.jpg

4.1.1 Premi.Utility

Nome: Utility

Tipo: class

Package: Premi

Descrizione: classe statica di utilità che fornisce strumenti per interagire con la base di dati

4.1.2 Premi.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi
- **Descrizione:** view generale che funge da sfondo per le altre view

4.1.3 Premi.PremiCtrl

- **Nome:** PremiCtrl
- **Tipo:** class
- **Package:** Premi
- **Descrizione:** controller generale che modifica la view principale in base alle scelte dell'utente
- **Relazioni con altri componenti:** la classe altera l'aspetto di `Premi.View` caricando le view di cui l'utente ha bisogno

4.2 Premi.UserManager

—————premi-class.jpg

4.2.1 Premi.UserManager.User

—————userManager-class.jpg (mostra le relazioni che possiede ESTERNE al package)

- **Nome:** User
- **Tipo:** class
- **Package:** Premi.UserManger
- **Descrizione:** classe che definisce un utente e fornisce le funzionalità necessarie alla sua creazione, al suo login e ad un eventuale cambio di password
- **Relazioni con altri componenti:** si collega a `Premi.Utility` per le interazioni con la base di dati

4.2.2 Premi.UserManager.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.UserManger
- **Descrizione:** vista generale del package `UserManager`. Può contenere le view abbinate alle funzionalità di `User`

4.2.3 Premi.UserManager.UserManagerCtrl

- **Nome:** UserManagerCtrl
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** controller generale del package `UserManager` dedicato a fornire alla view strumenti per l'interazione con l'utente
- **Relazioni con altri componenti:** si collega a `Premi.UserManager.View` per mostrare i dati dell'utente che va a prelevare tramite `Premi.UserManager.User`

4.2.4 Premi.UserManager.LoginView

- **Nome:** LoginView
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** view che permette all'utente di effettuare la login

4.2.5 Premi.UserManager.LoginController

- **Nome:** LoginController
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** fornisce gli strumenti necessari alla view per effettuare la login
- **Relazioni con altri componenti:** con `Premi.UserManager.LoginView` e con `Premi.UserManager.User` per trasmettere le informazioni relative al login

4.2.6 Premi.UserManager.RegistrationView

- **Nome:** RegistrationView
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** view che permette all'utente di registrarsi nel sistema

4.2.7 Premi.UserManager.RegistrationController

- **Nome:** RegistrationController
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** fornisce gli strumenti necessari alla view per registrare l'utente
- **Relazioni con altri componenti:** con `Premi.UserManager.RegistrationView` e con `Premi.UserManager.User` per trasmettere le informazioni relative alla registrazione

4.2.8 Premi.UserManager.ChangePasswordView

- **Nome:** ChangePasswordView
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** view che permette all'utente di cambiare la propria password

4.2.9 Premi.UserManager.ChangePasswordController

- **Nome:** ChangePassword
- **Tipo:** class
- **Package:** Premi.UserManager
- **Descrizione:** fornisce gli strumenti necessari alla view per cambiare la password dell'utente
- **Relazioni con altri componenti:** con `Premi.UserManager.ChangePasswordView` e con `Premi.UserManager.User` per trasmettere le informazioni relative al cambio password

4.3 Premi.Viewer

————viewer-class.jpg

4.3.1 Premi.Viewer.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.Viewer
- **Descrizione:** view che mostra la presentazione all'utente, offrendo funzionalità dipendenti dalla visibilità(pubblica o privata) o dal contesto in cui si trova(presentazione live)

4.3.2 Premi.ViewerCtrl

- **Nome:** ViewerCtrl
- **Tipo:** class
- **Package:** Premi.Viewer
- **Descrizione:** abilita funzionalità nella view in base all'utente
- **Relazioni con altri componenti:** con `Premi.Viewer.View`, con `Premi.UserManager.User` per verificare se l'utente è il proprietario della presentazione, e con la libreria esterna `ImpressJs` per aggiungere animazioni alla presentazione

4.4 Premi.Presentation

—————-presentation-class.jpg

4.4.1 Premi.Presentation.GraphicObject

- **Nome:** GraphicObject
- **Tipo:** *abstract class*
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta gli oggetti grafici nella presentazione

4.4.2 Premi.Presentation.GoContent

- **Nome:** GoContent
- **Tipo:** *abstract class*
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta gli oggetti grafici con contenuto di una presentazione
- **Relazioni con altri componenti:** estende `Premi.Presentation.GraphicObject`

4.4.3 Premi.Presentation.Text

- **Nome:** Text
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta un'area di testo nella presentazione
- **Relazioni con altri componenti:** estende `Premi.Presentation.GoContent`

4.4.4 Premi.Presentation.Image

- **Nome:** Image
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta un'immagine nella presentazione
- **Relazioni con altri componenti:** estende `Premi.Presentation.GoContent`

4.4.5 Premi.Presentation.Shape

- **Nome:** Shape
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta una figura nella presentazione
- **Relazioni con altri componenti:** estende `Premi.Presentation.GoContent`

4.4.6 Premi.Presentation.Frame

- **Nome:** Frame
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta un frame nella presentazione
- **Relazioni con altri componenti:** estende `Premi.Presentation.GraphicObject` e possiede un insieme di oggetti `Premi.Presentation.GoContent`

4.4.7 Premi.Presentation.Trail

- **Nome:** Trail
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta un percorso nella presentazione
- **Relazioni con altri componenti:** possiede una lista di riferimenti ad oggetti di tipo `Premi.Presentation.Frame`, ma non è in possesso degli oggetti stessi

—dipendenze di presentation.jpg CITARE L'IMMAGINE DELLE DIPENDENZE CON DI PRESENTATION Completo-Dettagliato.jpg

4.4.8 Premi.Presentation.Presentation

- **Nome:** Presentation
- **Tipo:** class
- **Package:** Premi.Presentation
- **Descrizione:** rappresenta una presentazione
- **Relazioni con altri componenti:** possiede una lista di oggetti `premi.Presentation.Graphic` che possono essere sia Frame che oggetti grafici generici e assieme compongono l'infografica della presentazione, possiede inoltre una lista di percorsi e quindi di oggetti `Premi.Presentation.Trail`, identifica l'appartenenza di una presentazione ad un determinato `Premi.UserManager.User` e si collega alla base di dati tramite la classe statica `Premi.Utility`

4.5 Premi.PresentationManager

——-usermanager-classe.jpg

4.5.1 Premi.PresentationManager.Export

- **Nome:** Export
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** permette di esportare la presentazione in formato poster
- **Relazioni con altri componenti:** preleva le informazioni da `Premi.Presentation.Presentation` per costruire il poster

4.5.2 Premi.PresentationManager.Portable

- **Nome:** Portable
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** permette di rendere portabile una presentazione, e quindi di essere indipendente dal software Premi e di essere caricata come pagina HTML da un browser
- **Relazioni con altri componenti:**

4.5.3 Premi.PresentationManager.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.PresentationController
- **Descrizione:** è la view generale delle operazioni che l'utente può effettuare sul

4.5.4 Premi.PresentationManager.PresentationManagerCtrl

- **Nome:** PresentationManagerCtrl
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view gli strumenti necessari alla gestione delle presentazioni
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.View`

4.5.5 Premi.PresentationManager.RemoveView

- **Nome:** RemoveView
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** mostra una finestra di conferma eliminazione della presentazione selezionata

4.5.6 Premi.PresentationManager.RemoveController

- **Nome:** RemoveController
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view gli strumenti per rimuovere la presentazione o annullare l'azione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.RemoveView` e sfrutta le funzionalità di `Premi.Presentation.Presentation` per l'eliminazione della presentazione

4.5.7 Premi.PresentationManager.PublicView

- **Nome:** PublicView
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** mostra una finestra di conferma per rendere pubblica o privata una presentazione

4.5.8 Premi.PresentationManager.PublicController

- **Nome:** PublicController
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view gli strumenti per rendere pubblica o privata una presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.PublicView` e sfrutta le funzionalità di `Premi.Presentation.Presentation` per accedere alla base di dati

4.5.9 Premi.PresentationManager.EditTitleDescrView

- **Nome:** EditTitleDescrView
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** mostra una finestra per la modifica del titolo e della descrizione della presentazione

4.5.10 Premi.PresentationManager.EditTitleDescrController

- **Nome:** EditTitleDescrController
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view la possibilità di modificare titolo e descrizione della presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.EditTitleDescrView` e sfrutta le funzionalità di `Premi.Presentation.Presentation` per accedere alla base di dati

4.5.11 Premi.PresentationManager.NewView

- **Nome:** NewView
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** mostra una finestra per la creazione di una nuova presentazione

4.5.12 Premi.PresentationManager.NewController

- **Nome:** NewController
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view gli strumenti per creare una nuova presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.New` e sfrutta le funzionalità di `Premi.Presentation.Presentation` per accedere alla base di dati

4.5.13 Premi.PresentationManager.ListPresView

- **Nome:** ListPresView
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** mostra all'utente la lista delle sue presentazioni, e bottoni aggiuntivi per accedere alle altre funzionalità del package

4.5.14 Premi.PresentationManager.ListPresController

- **Nome:** ListPresController
- **Tipo:** class
- **Package:** Premi.PresentationManager
- **Descrizione:** fornisce alla view una lista preformata delle presentazioni dell'utente
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.PresentationManager.Lis` e sfrutta le funzionalità di `Premi.Presentation.Presentation` per recuperare la lista delle presentazioni

4.6 Premi.Editor

4.6.1 Premi.Editor.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** view generica dell'editor che funge da contenitore

4.6.2 Premi.Editor.EditorCtrl

- **Nome:** EditorCtrl
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** fornisce alla view le funzionalità per la modifica di una presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.EditorView` e sfrutta `Premi.Presentation.Presentation` per recuperare la presentazione dalla base di dati

4.6.3 Premi.Editor.TextView

- **Nome:** TextView
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** mostra le opzioni per l'aggiunta di un'area di testo

4.6.4 Premi.Editor.TextController

- **Nome:** TextController
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** fornisce alla view le funzionalità per creare e modificare un'area di testo
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.TextView` e utilizza la classe `Premi.Presentation.Text` per rappresentare le aree di testo

4.6.5 Premi.Editor.ShapeView

- **Nome:** ShapeView
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** mostra le opzioni per l'aggiunta di uno shape

4.6.6 Premi.Editor.ShapeController

- **Nome:** ShapeController
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** fornisce alla view le funzionalità per creare e modificare uno shape
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.ShapeView` e utilizza la classe `Premi.Presentation.Shape` per rappresentare le Shape

4.6.7 Premi.Editor.ImageView

- **Nome:** ImageView
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** mostra le opzioni per l'aggiunta di un'immagine

4.6.8 Premi.Editor.ImageController

- **Nome:** ImageController
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** fornisce alla view le funzionalità per aggiungere un'immagine
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.ImageView` e utilizza la classe `Premi.Presentation.Image` per rappresentare le immagini

4.6.9 Premi.Editor.StyleView

- **Nome:** StyleView
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** permette la modifica dello stile di un Frame oppure dello sfondo dell'infografica

4.6.10 Premi.Editor.StyleController

- **Nome:** StyleController
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** fornisce alla view le funzionalità per la modifica dello stile
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.StyleView`, e utilizza la classe `Premi.Presentation.Frame` per rappresentare lo stile dei Frame e dell'infografica

4.6.11 Premi.Editor.FrameListView

- **Nome:** FrameListView
- **Tipo:** class
- **Package:** Premi.Editor
- **Descrizione:** view di base per la rappresentazione di una lista dei Frame contenuti all'interno della presentazione

4.7 Premi.Editor.FrameEditor

4.7.1 Premi.Editor.FrameEditor.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.Editor.FrameEditor.View
- **Descrizione:** è la view generale della fase di modifica dei Frame

4.7.2 Premi.Editor.FrameEditor.FrameEditorCtrl

- **Nome:** FrameEditorCtrl
- **Tipo:** class
- **Package:** Premi.Editor.FrameEditor.FrameEditorCtrl
- **Descrizione:** fornisce alla view le funzionalità per la gestione delle altre view contenute al suo interno
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.FrameEditor.View`

4.7.3 Premi.Editor.FrameEditor.FrameListController

- **Nome:** `FrameListController`
- **Tipo:** `class`
- **Package:** `Premi.Editor.FrameEditor`
- **Descrizione:** fornisce alla view `FrameListView` le funzionalità per rappresentazione della lista dei `Frame`
- **Relazioni con altri componenti:** è uno dei tre controller dedicati di `Premi.Editor.FrameList` e utilizza la classe `Premi.Presentation.Frame` per rappresentare un'anteprima dei `Frame`

4.8 Premi.Editor.InfographicEditor

4.8.1 Premi.Editor.InfographicEditor.View

- **Nome:** `View`
- **Tipo:** `class`
- **Package:** `Premi.Editor.InfographicEditor`
- **Descrizione:** è la view generale della fase di creazione o modifica dell'infografica

4.8.2 Premi.Editor.InfographicEditor.InfographicEditorCtrl

- **Nome:** `InfographicEditorCtrl`
- **Tipo:** `class`
- **Package:** `Premi.Editor.InfographicEditor`
- **Descrizione:** fornisce alla view le funzionalità per la gestione delle altre view contenute al suo interno, che permetteranno all'utente di produrre un'infografica attraverso l'utilizzo di oggetti grafici, tra cui i `Frame` creati nella fase precedente
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.InfoGraphicEdito`

4.8.3 Premi.Editor.InfographicEditor.FrameListController

- **Nome:** `FrameListController`
- **Tipo:** `class`
- **Package:** `Premi.Editor.InfographicEditor`
- **Descrizione:** fornisce alla view `FrameListView` le funzionalità per rappresentazione della lista dei `Frame`
- **Relazioni con altri componenti:** è uno dei tre controller dedicati di `Premi.Editor.FrameList` e utilizza la classe `Premi.Presentation.Frame` per rappresentare un'anteprima dei `Frame`

4.9 Premi.Editor.TrailsEditor

4.9.1 Premi.Editor.TrailsEditor.View

- **Nome:** View
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** view generale della della fase di creazione o modifica dei percorsi di presentazione

4.9.2 Premi.Editor.TrailsEditor.TrailsEditorCtrl

- **Nome:** TrailsEditorCtrl
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view le funzionalità per la gestione delle altre view contenute al suo interno, che permetteranno di gestire i percorsi di presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.TrailsEditor.View`, si collega a `Premi.Presentation.Presentation` per recuperare oggetti di tipo `Premi.Presentation.Trail`

4.9.3 Premi.Editor.TrailsEditor.FrameListController

- **Nome:** FrameListController
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view `FrameListView` le funzionalità per rappresentazione della lista dei Frame
- **Relazioni con altri componenti:** è uno dei tre controller dedicati di `Premi.Editor.FrameList` e utilizza la classe `Premi.Presentation.Frame` per rappresentare un'anteprima dei Frame

4.9.4 Premi.Editor.TrailsEditor.EditTrailView

- **Nome:** EditTrailView
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** view dedicata alla modifica di un percorso di presentazione

4.9.5 Premi.Editor.TrailsEditor.EditTrailController

- **Nome:** EditTrailController
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view le funzionalità per la gestione delle altre view contenute al suo interno, che permetteranno all'utente di produrre un percorso di presentazione con i Frame da lui creati
- **Relazioni con altri componenti:** è il controller dedicato di Premi.Editor.TrailsEditor.EditTrailController

4.9.6 Premi.Editor.TrailsEditor.NewView

- **Nome:** NewView
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** mostra la finestra di creazione di un percorso di presentazione

4.9.7 Premi.Editor.TrailsEditor.NewController

- **Nome:** NewController
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view le funzionalità per la creazione di un nuovo percorso di presentazione
- **Relazioni con altri componenti:** è il controller dedicato di Premi.Editor.TrailsEditor.NewController

4.9.8 Premi.Editor.TrailsEditor.EditTitleView

- **Nome:** EditTitleView
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** mostra la finestra di modifica del titolo di un percorso di presentazione

4.9.9 Premi.Editor.TrailsEditor.EditTitleController

- **Nome:** EditTileController
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view le funzionalità per la modifica del titolo di un percorso di presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.TrailsEditor.EditTitleController`

4.9.10 Premi.Editor.TrailsEditor.RemoveView

- **Nome:** RemoveView
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** mostra la finestra di conferma rimozione di un percorso di presentazione

4.9.11 Premi.Editor.TrailsEditor.RemoveController

- **Nome:** RemoveController
- **Tipo:** class
- **Package:** Premi.Editor.TrailsEditor
- **Descrizione:** fornisce alla view le funzionalità per la rimozione di un percorso di presentazione
- **Relazioni con altri componenti:** è il controller dedicato di `Premi.Editor.TrailsEditor.RemoveController`

5 Diagrammi delle attività

Vengono illustrati ora i diagrammi di attività. Viene illustrato il diagramma principale ad alto livello e i diversi sotto-diagrammi specifici.

5.1 Attività principali

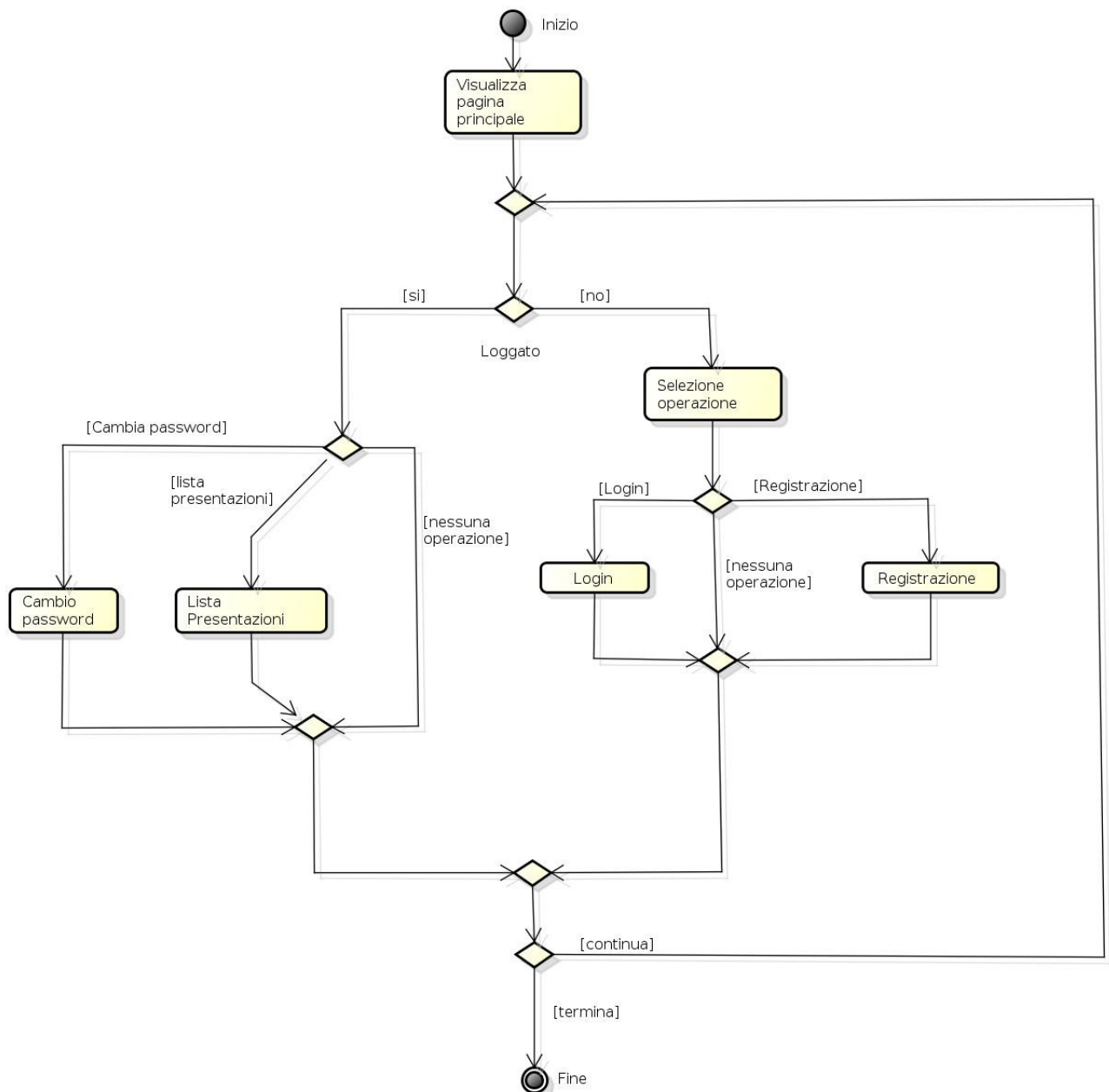


Figura 1: Attività principali

L'utente nel momento in cui accede nel programma ha la possibilità di effettuare la login o di registrarsi nel sistema. L'utente loggato può invece effettuare il logout, andare nella lista presentazioni ed effettuare il cambio password.

5.2 Lista presentazioni

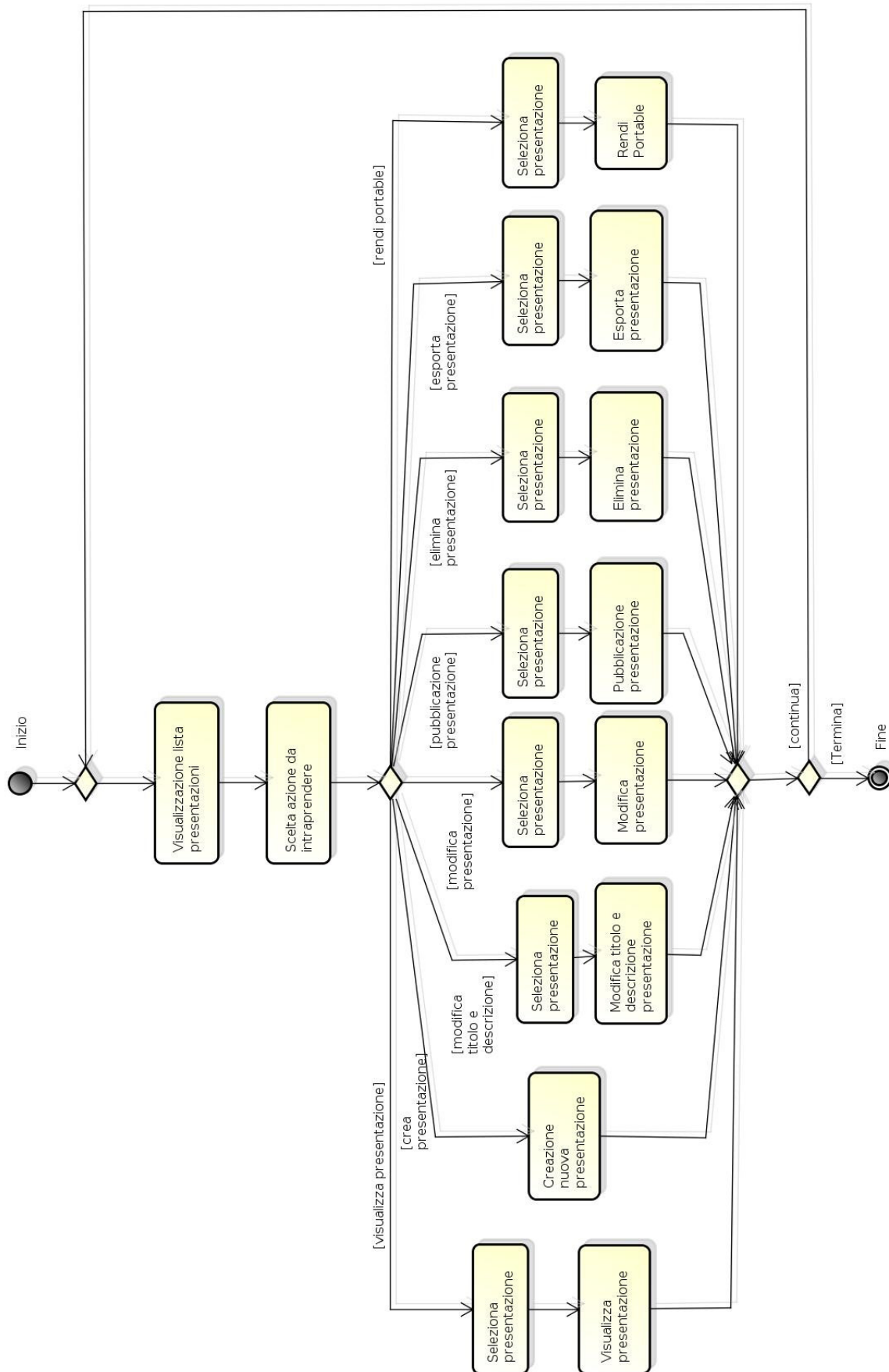


Figura 2: Lista presentazioni

Le scelte che ha l'utente una volta entrato nella lista presentazioni sono: Visualizza presentazione, creazione nuova presentazione, modifica titolo e descrizione presentazione, modifica presentazione, pubblicazione presentazione, elimina presentazione, esporta presentazione, rendi portable.

5.3 Login

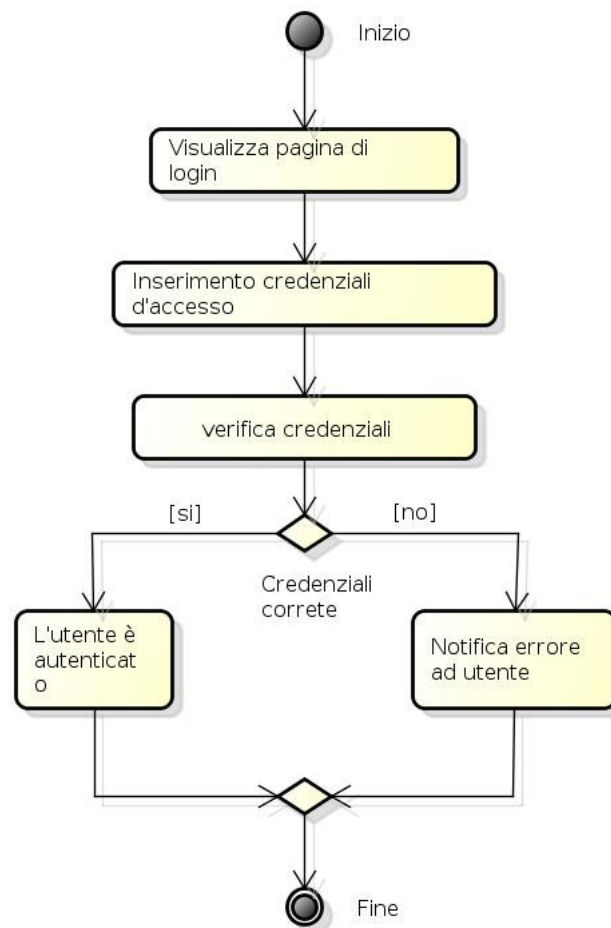


Figura 3: Login utente

L'utente quando accede nella pagina di login inserisce le credenziali che corrispondono a email e password. Se sono corrette viene autenticato altrimenti viene restituito un messaggio d'errore.

5.4 Registrazione

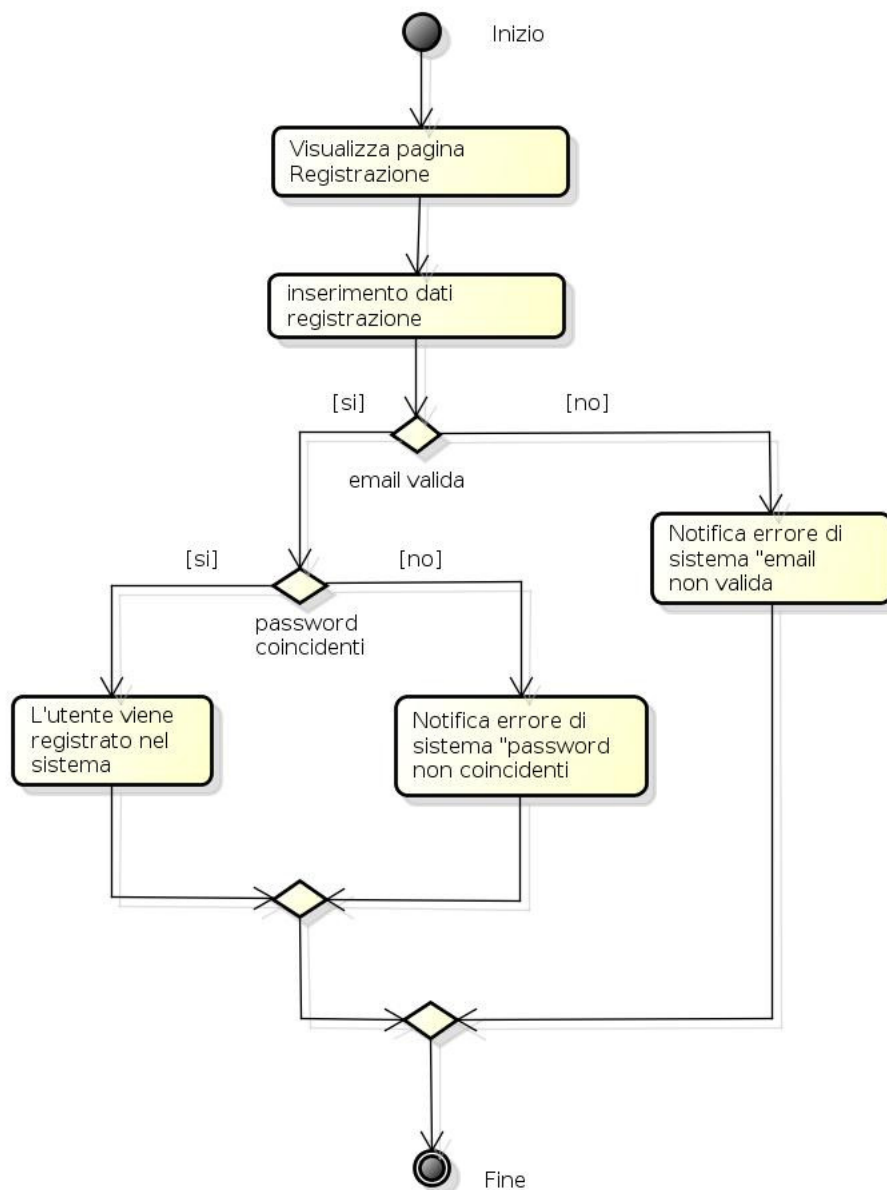


Figura 4: Registrazione utente

L'utente quando accede nella parte di registrazione inserisce l'email e la password. Se l'email non ha un formato valido o se è già presente nel sistema viene restituito un errore altrimenti viene verificato che le password inserite coincidano. In caso affermativo l'utente viene registrato nel sistema, altrimenti viene restituito un messaggio d'errore.

5.5 Cambio password

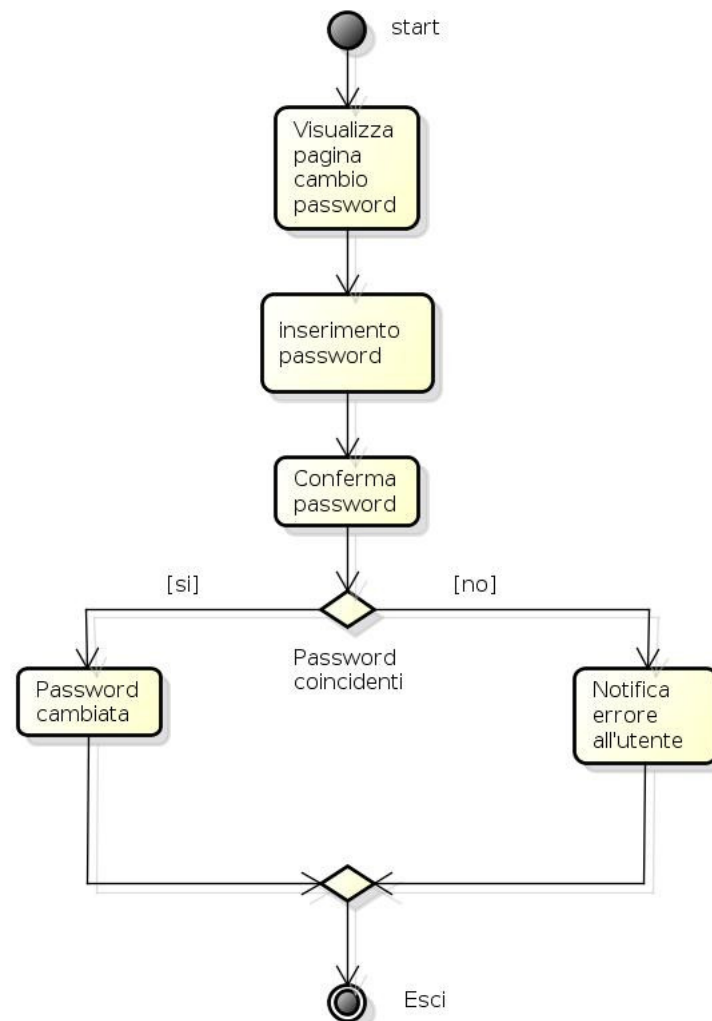


Figura 5: Cambio password

Per effettuare il cambio password l'utente inserisce la nuova password e la conferma della nuova. Se sono coincidenti viene effettuato il cambio password altrimenti viene notificato un errore all'utente.

5.6 Visualizzatore

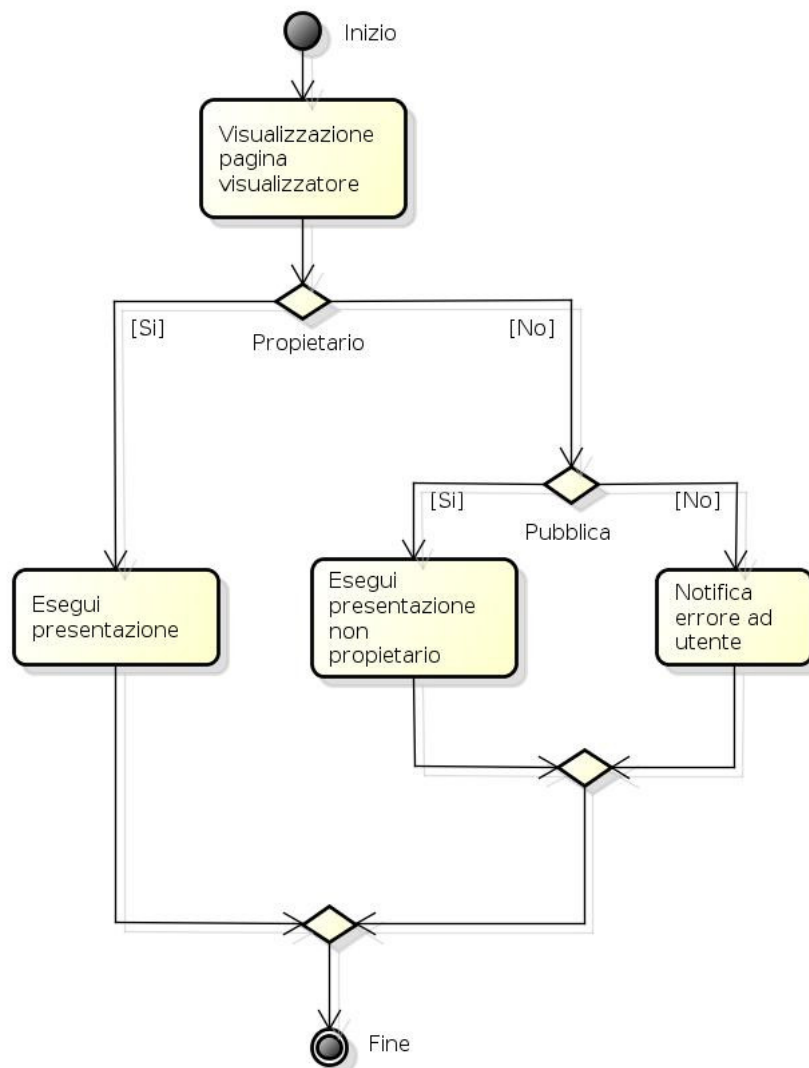


Figura 6: Visualizzatore

Se l'utente che visualizza la presentazione è l'utente proprietario viene eseguita la presentazione in modalità proprietario altrimenti viene controllato se la presentazione è pubblica. In caso affermativo viene eseguita la presentazione in modalità non proprietario altrimenti viene notificato un errore, in quanto se la presentazione non è pubblica non può essere visualizzata. L'utente non proprietario per accedere ad una presentazione deve ottenere il link generato dall'utente proprietario nel momento in cui la rende pubblica. L'utente proprietario può rendere privata una presentazione in ogni momento perciò il controllo se è pubblica, deve essere effettuato per controllare se il link è ancora valido.

5.7 Esegui presentazione proprietario

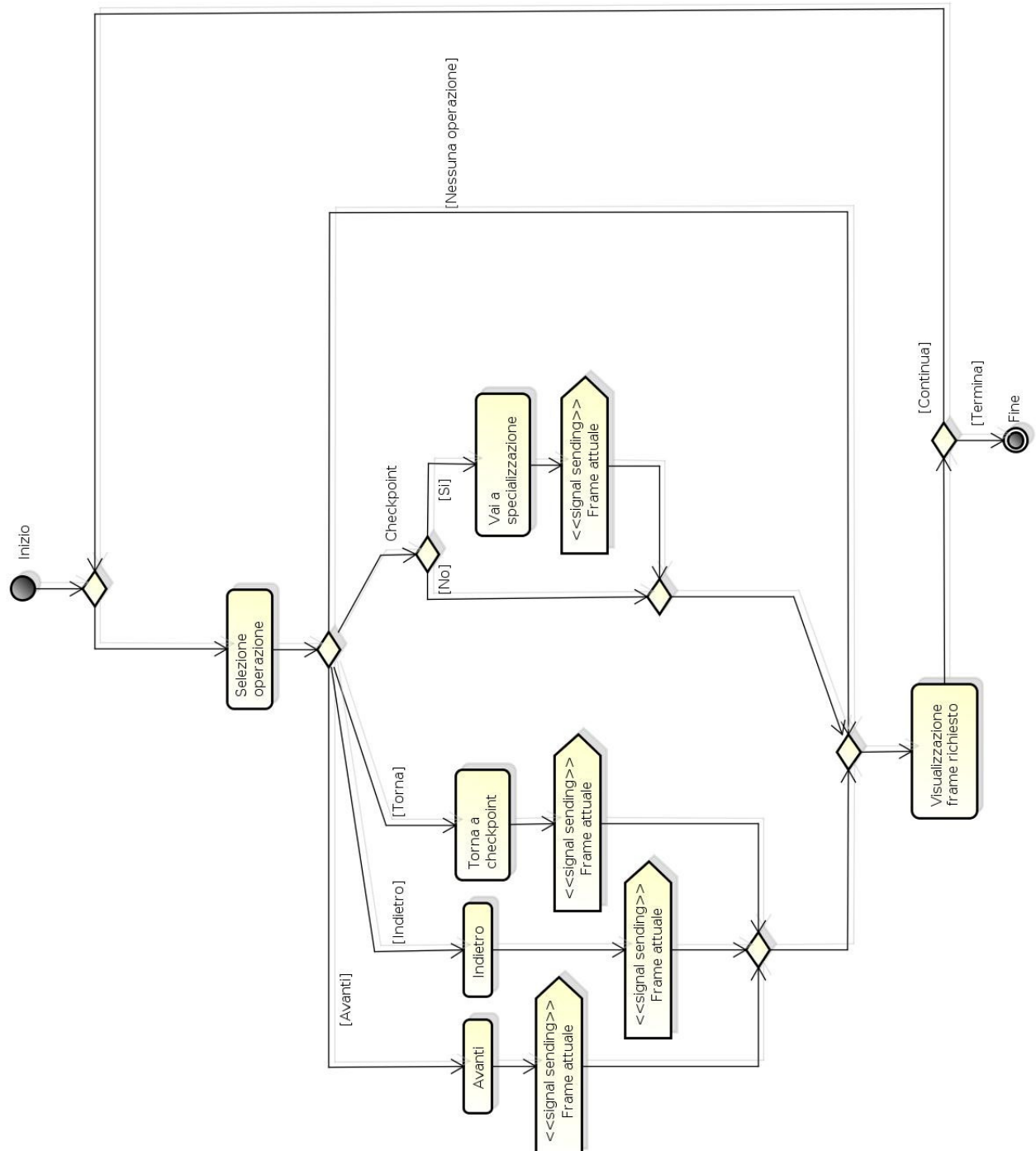


Figura 7: Esegui presentazione proprietario

Se la modalità di visualizzazione presentazione è in modalità proprietario si hanno le seguenti scelte:

- avanti: per andare avanti di una frame;
- indietro: per tornare indietro di un frame;

- torna a checkpoint: permette di tornare al frame iniziale o di tornare al checkpoint se si è entrati in un percorso di specializzazione;
- checkpoint: se il frame corrente è un checkpoint l'utente con questa scelta entra nel percorso di specializzazione.

Il segnale Frame attuale inviato permette agli utenti non proprietari di visualizzare il frame corrente scelto dal proprietario. L'utente ha la possibilità di uscire quando lo desidera.

- torna a checkpoint: permette di tornare al frame iniziale o di tornare al checkpoint se si è entrati in un percorso di specializzazione;
- checkpoint: se il frame corrente è un checkpoint l'utente con questa scelta entra nel percorso di specializzazione;
- presentazione live: con questa scelta l'utente visualizza il frame corrente che l'utente proprietario ha scelto di visualizzare.

L'utente ha la possibilità di uscire quando lo desidera.

5.9 Creazione presentazione

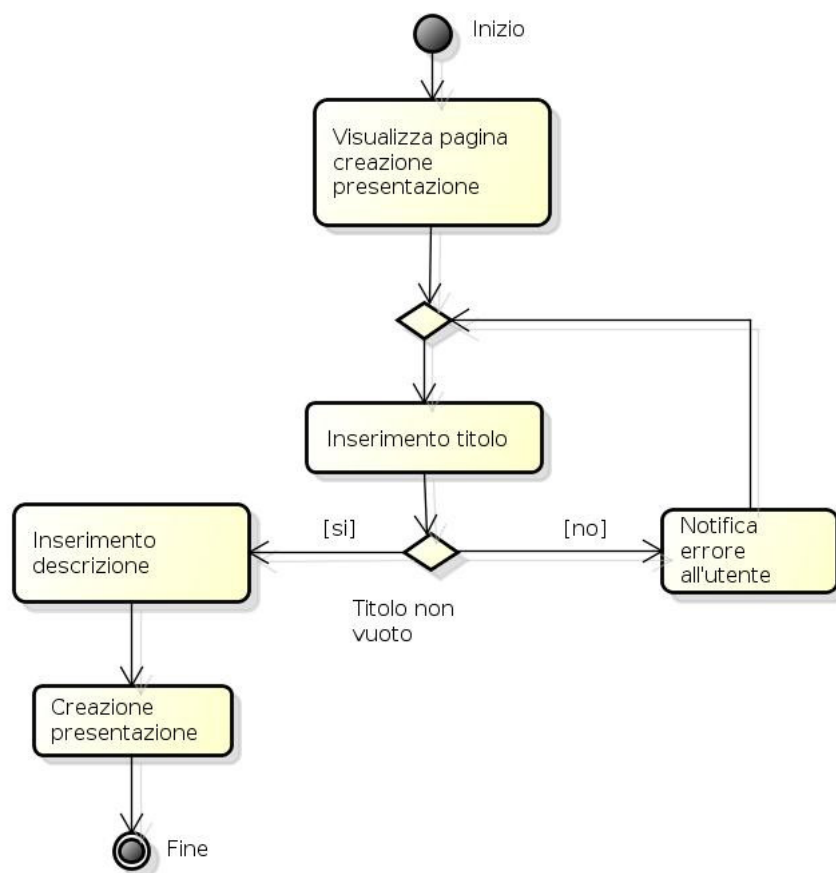


Figura 9: Creazione presentazione

L'utente può creare una nuova presentazione inserendo titolo e descrizione. Se non inserisce il titolo viene visualizzata una notifica di errore altrimenti la presentazione viene inserita nel sistema.

5.10 Modifica titolo e descrizione presentazione

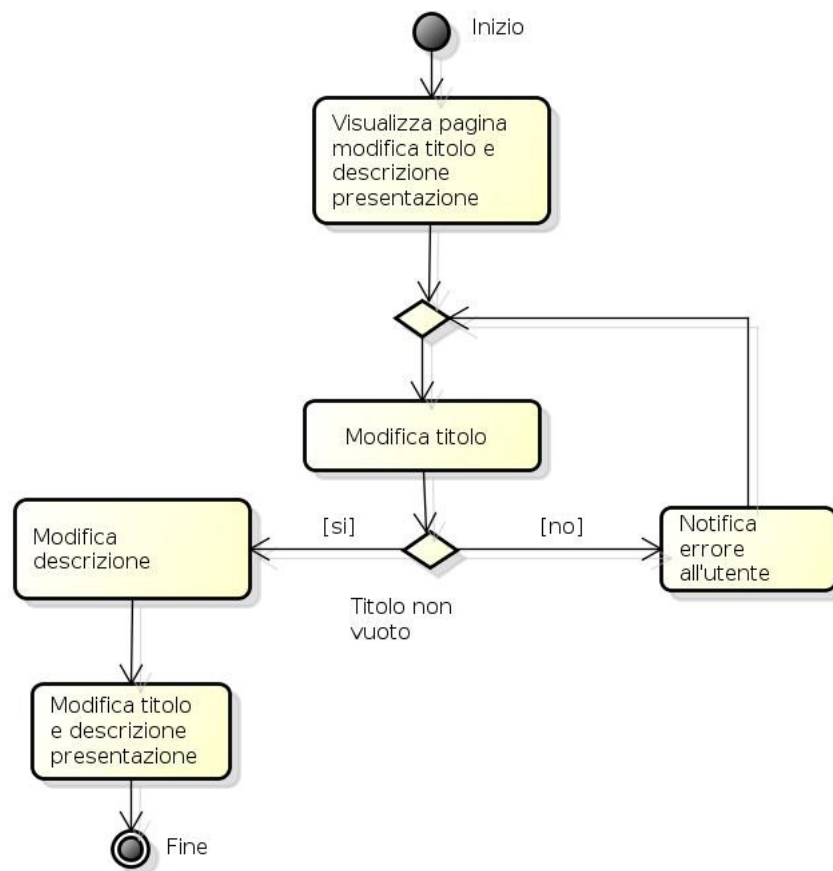


Figura 10: Modifica titolo e descrizione della presentazione

L'utente può modificare sia il titolo che la descrizione. Se il titolo non è vuoto le modifiche vengono salvate altrimenti viene restituita una notifica di errore.

5.11 Pubblicazione presentazione

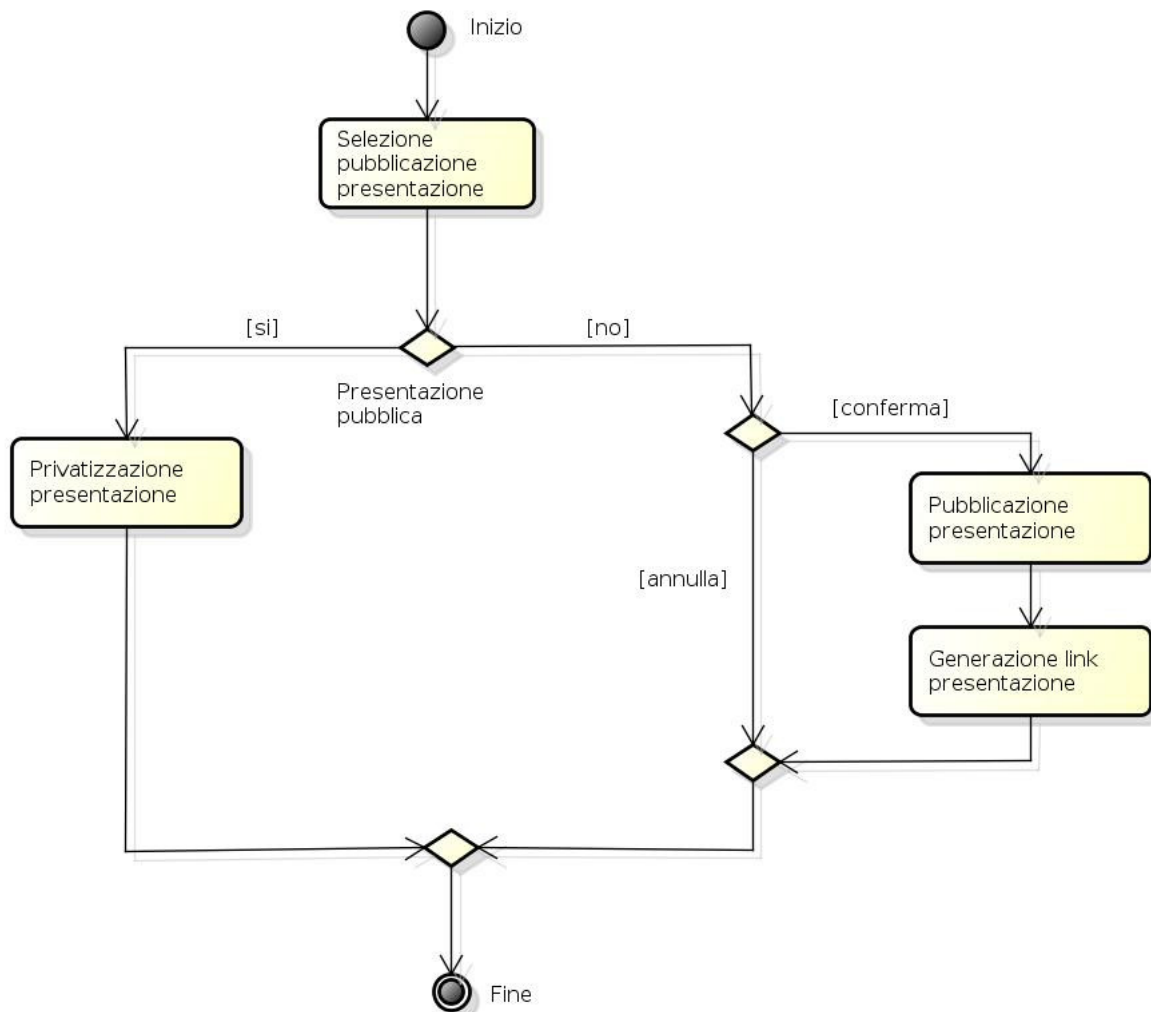


Figura 11: Pubblicazione presentazione

Se la presentazione è già pubblica rende privata la stessa, altrimenti se conferma la pubblicazione la rende visibile al pubblico e viene generato un link da inviare a chi voglia visualizzarla.

5.12 Elimina presentazione

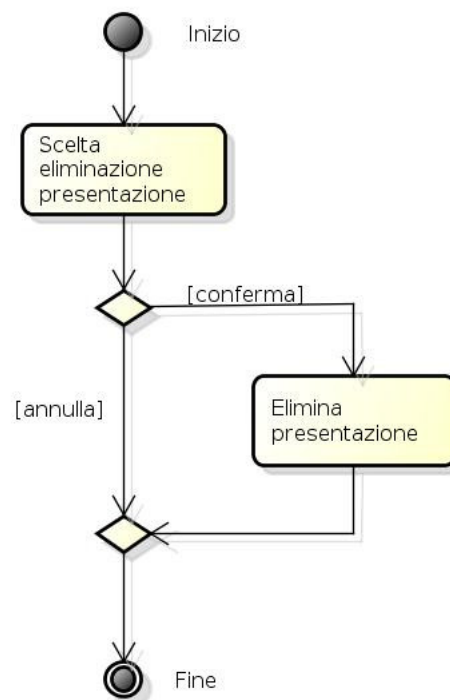


Figura 12: Elimina presentazione

L'utente deve confermare l'eliminazione altrimenti l'operazione viene annullata.

5.13 Esportazione presentazione

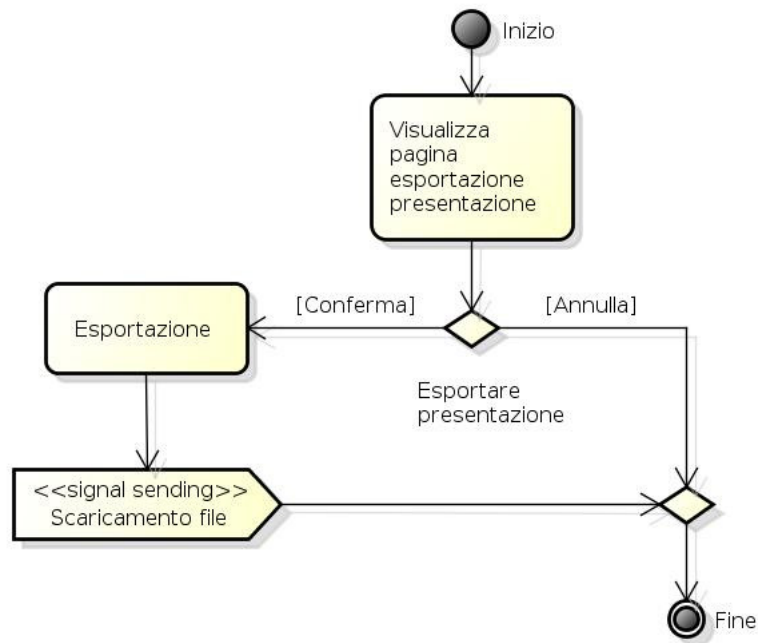


Figura 13: Esportazione presentazione

L'utente può esportare la presentazione in modo da ottenere un poster. Se l'operazione è confermata vengono esportati i dati e si procede allo scaricamento del file. Altrimenti viene annullata.

5.14 Rendi portable

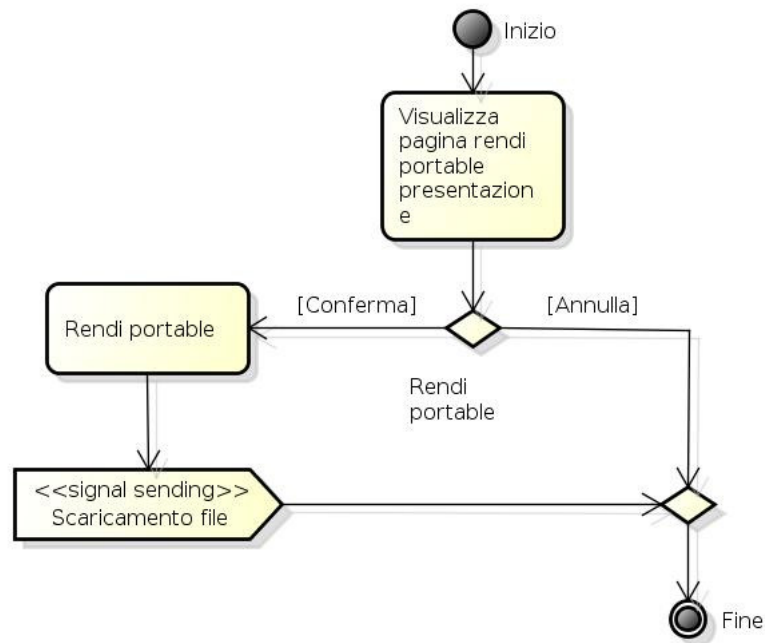


Figura 14: Rendi portable

L'utente può rendere portabile la presentazione in modo da vederla offline. Se l'operazione è confermata la presentazione viene resa portabile e si procede allo scaricamento dei file. Altrimenti viene annullata.

5.15 Modifica presentazione

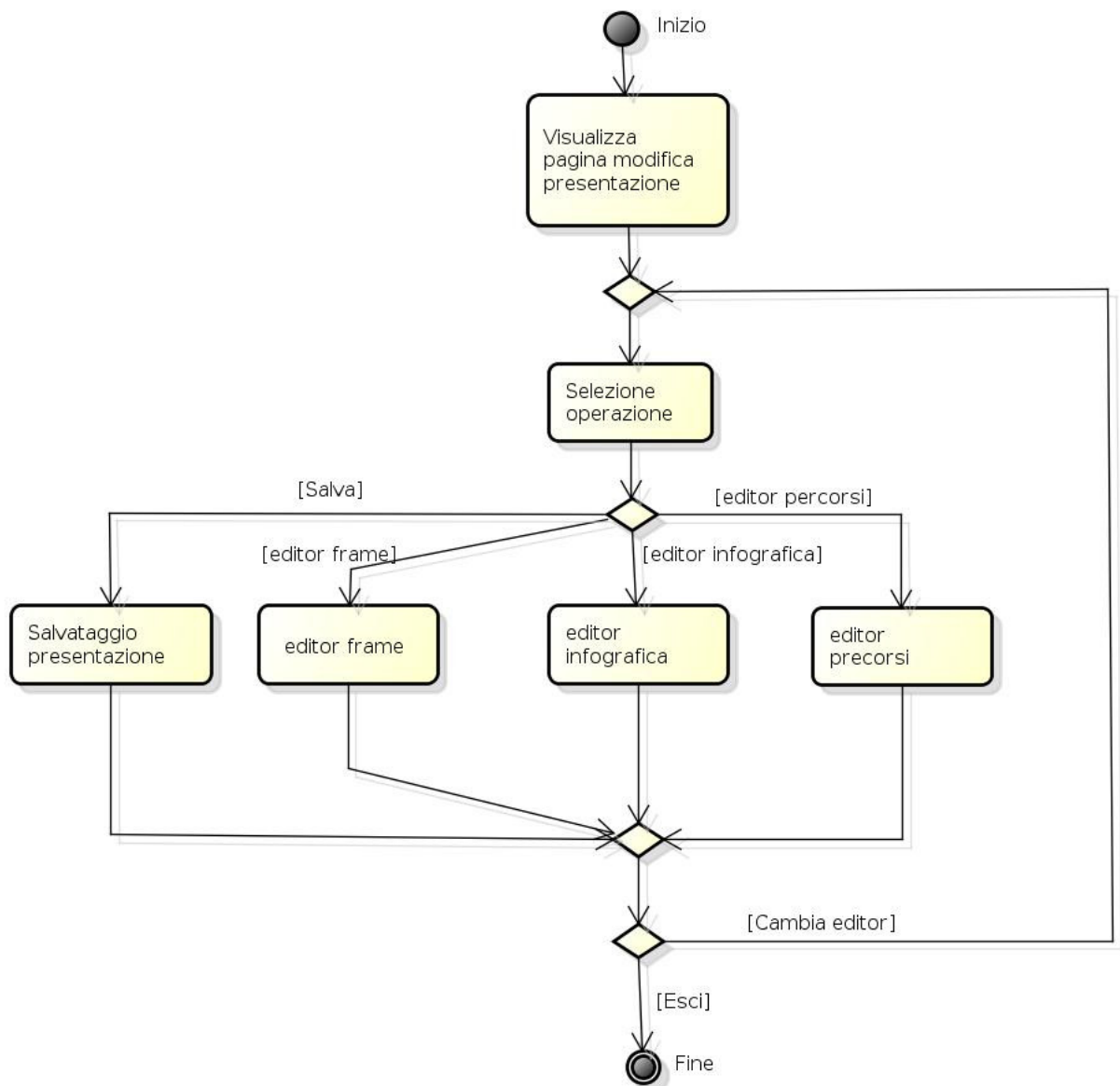


Figura 15: Modifica presentazione

L'utente può scegliere di: salvare la presentazione, andare nel frame editor, andare nell'infografica editor o nell'editor percorsi. L'utente può in ogni momento cambiare editor.

5.16 Frame editor

L'utente procede alla creazione del frame oppure può selezionarne uno. Una volta selezionato viene visualizzato nell'editor e a questo punto può scegliere se eliminarlo o modificarlo.

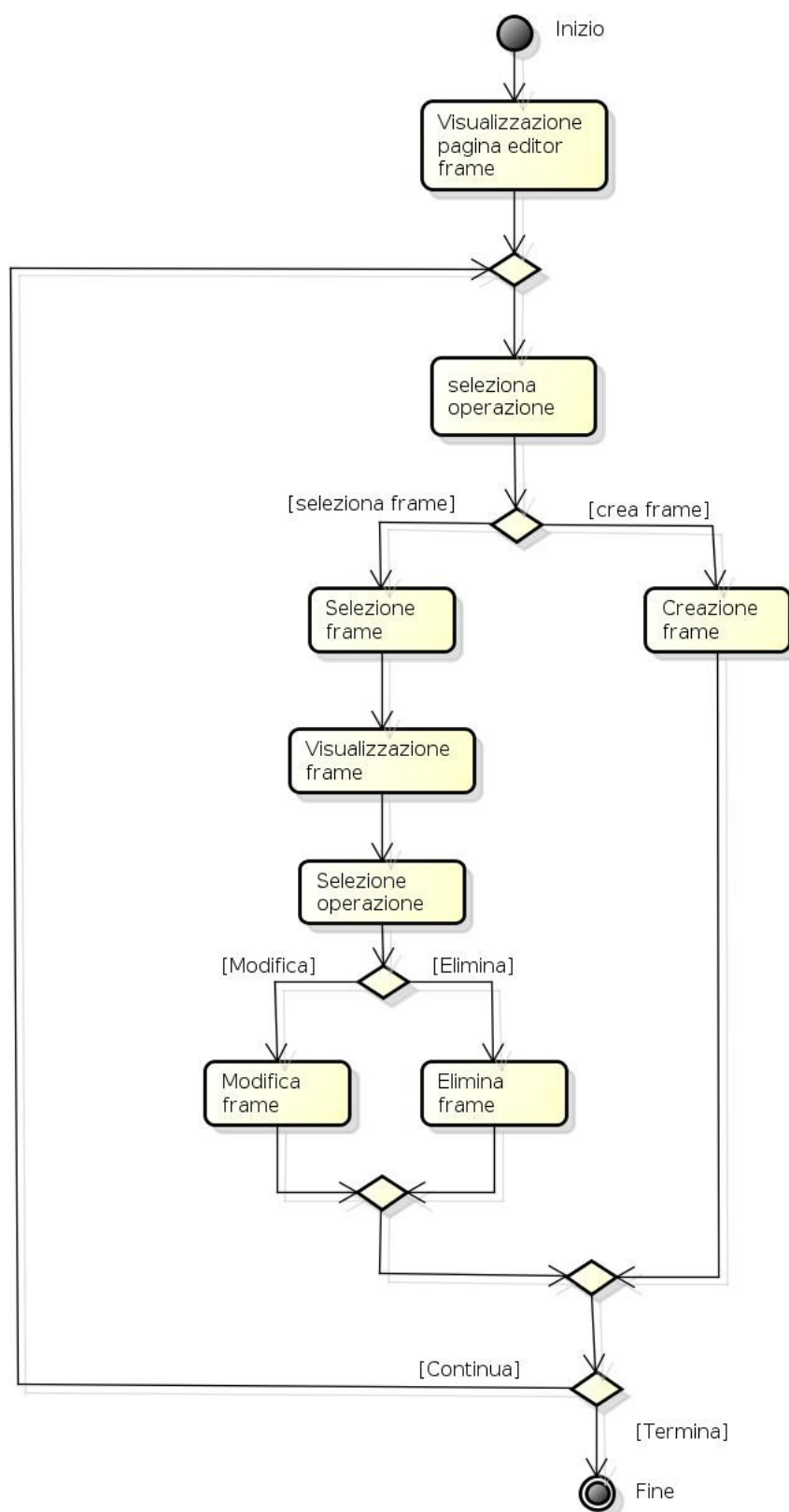


Figura 16: Frame editor

5.17 Modifica frame

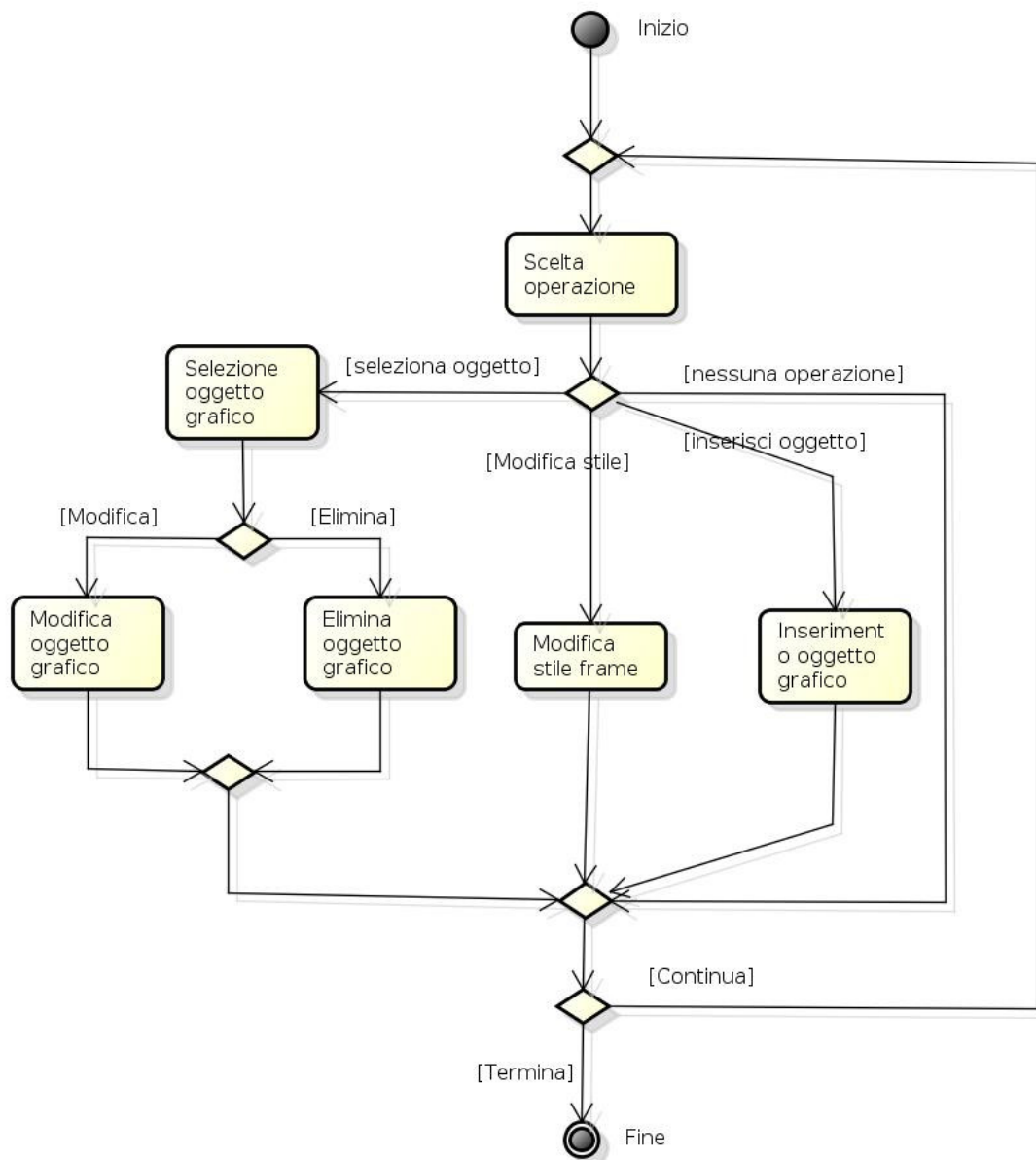


Figura 17: Modifica frame

L'utente può effettuare le seguenti operazioni:

- Selezione oggetto grafico: una volta selezionato l'oggetto può essere eliminato o modificato;
- Modificare lo stile del frame;
- Inserire un oggetto grafico nel frame;
- Non effettuare nessuna operazione.

5.18 Infografica editor

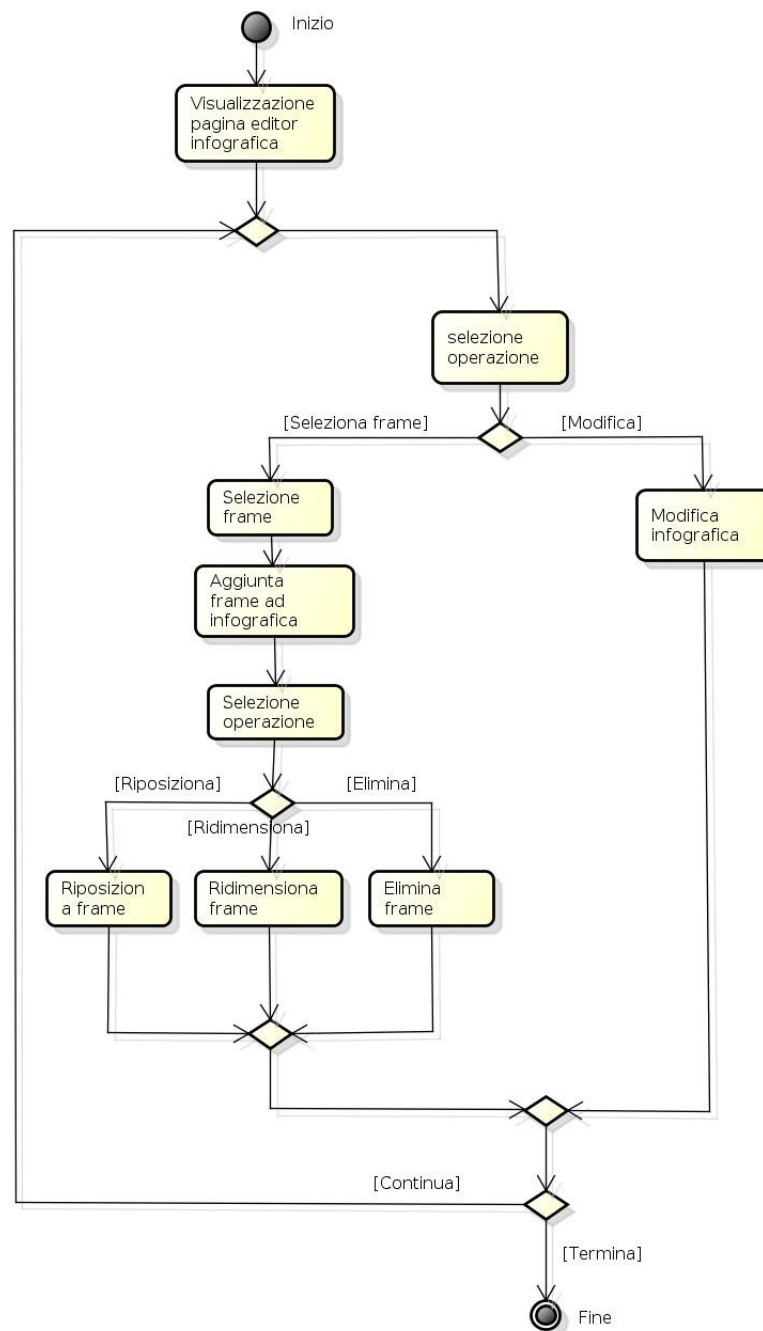


Figura 18: Infographic editor

L'utente può scegliere di:

- modificare l'infografica
- selezionare un frame e successivamente aggiungerlo all'infografica, eliminarlo dall'infografica o cambiargli la posizione e la grandezza e altezza.

5.19 Modifica infografica

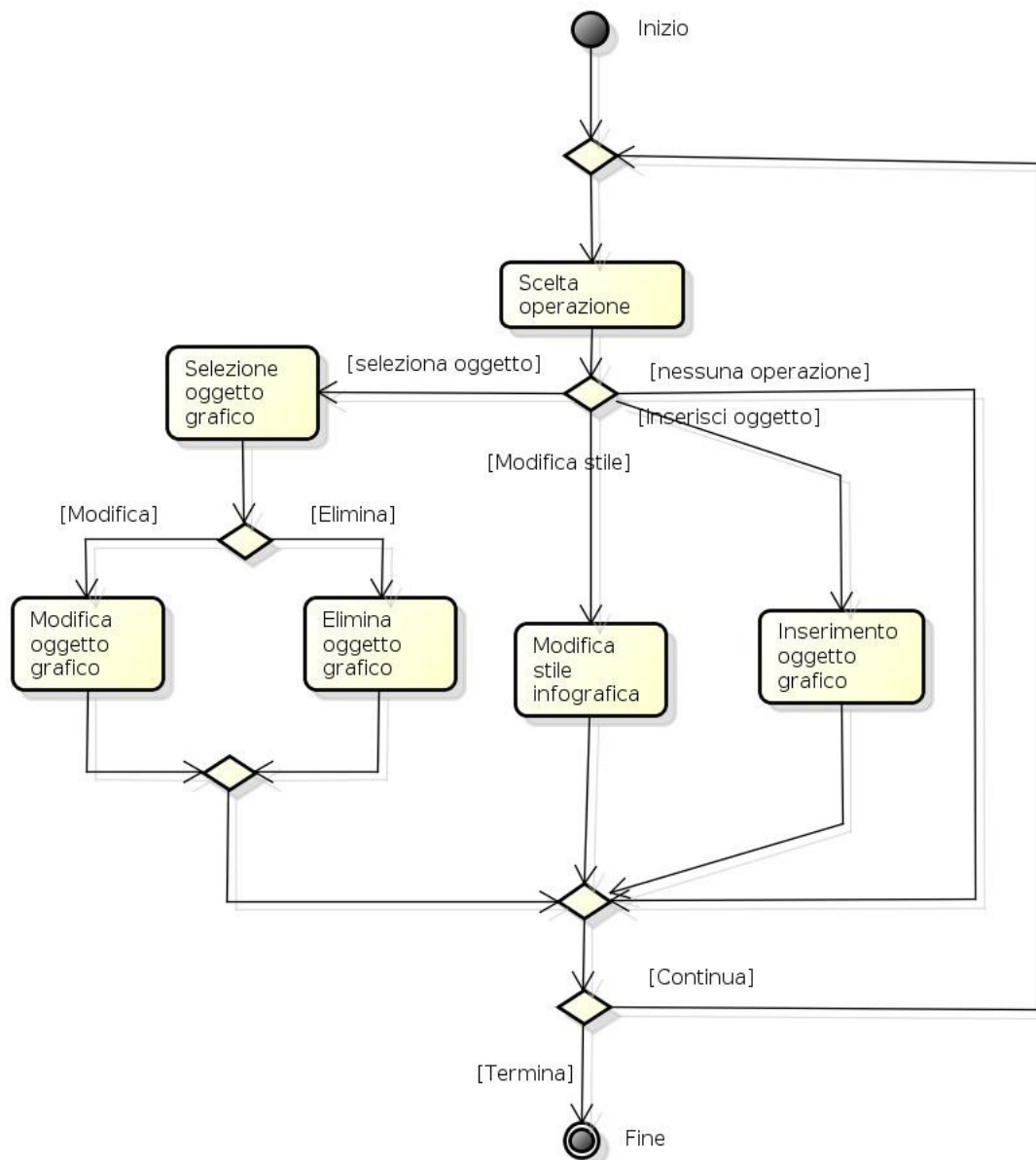


Figura 19: Modifica infografica

L'utente può effettuare le seguenti operazioni:

- Selezione oggetto grafico: una volta selezionato l'oggetto può essere eliminato o modificato;
- Modificare lo stile dell'infografica;
- Inserire un oggetto grafico nell'infografica;
- Non effettuare nessuna operazione.

5.20 Editor percorsi

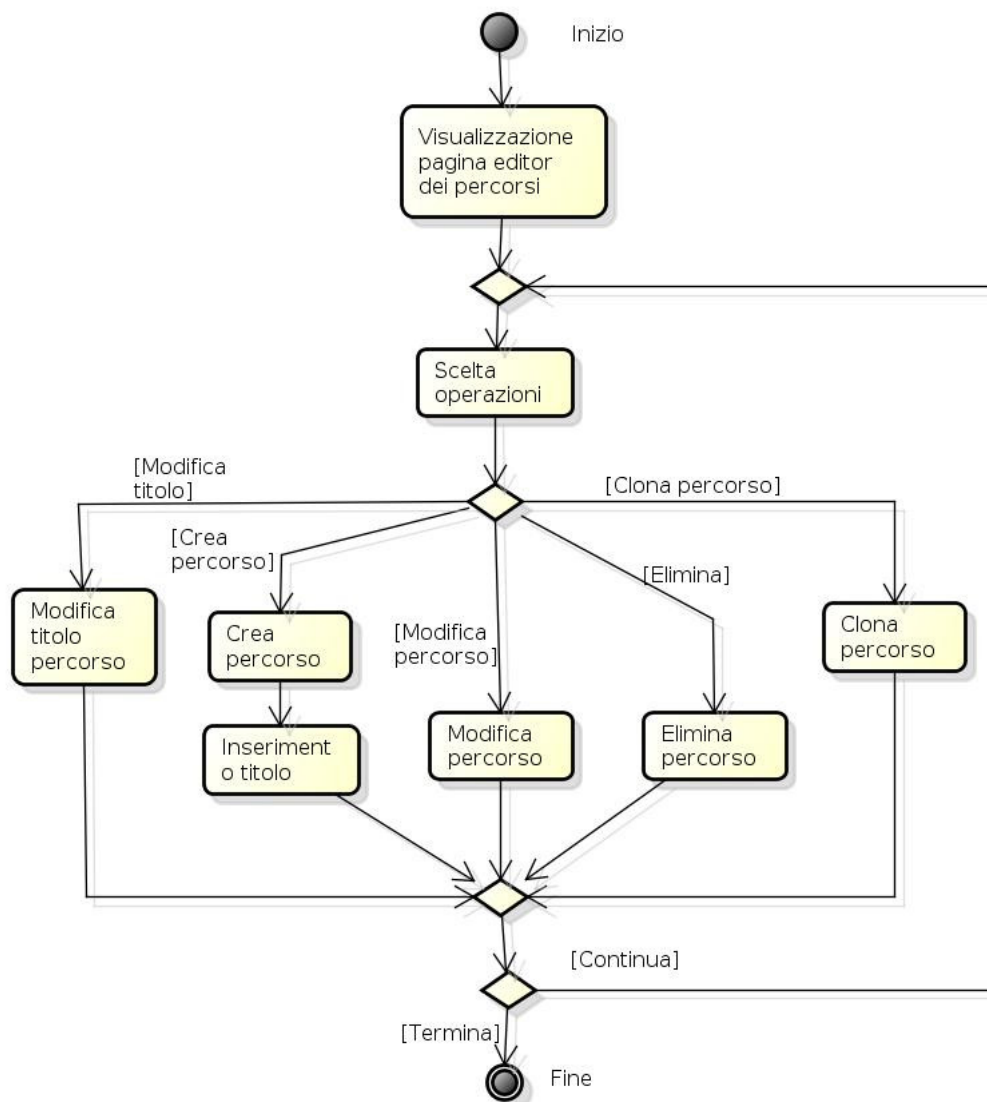


Figura 20: Editor percorsi

Le operazioni che può fare l'utente sono:

- Modificare il titolo del percorso selezionato;
- Creare un nuovo percorso inserendo il titolo;
- Modificare il percorso selezionato;
- Eliminare il percorso selezionato;
- Clonare il percorso selezionato.

5.21 Modifica percorso

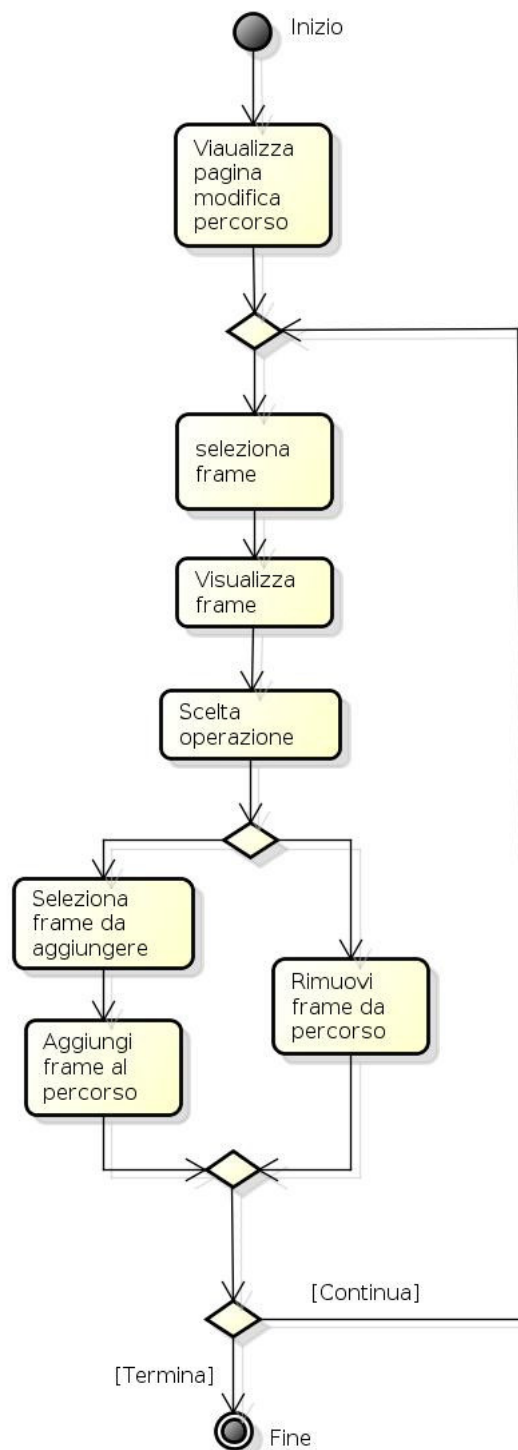


Figura 21: Modifica percorso

Per modificare il percorso l'utente seleziona un frame e questo viene visualizzato nell'editor. Dopodiché ha la possibilità di rimuovere il frame dal percorso o di selezionarne uno da aggiungerlo al percorso.

5.22 Aggiungi frame

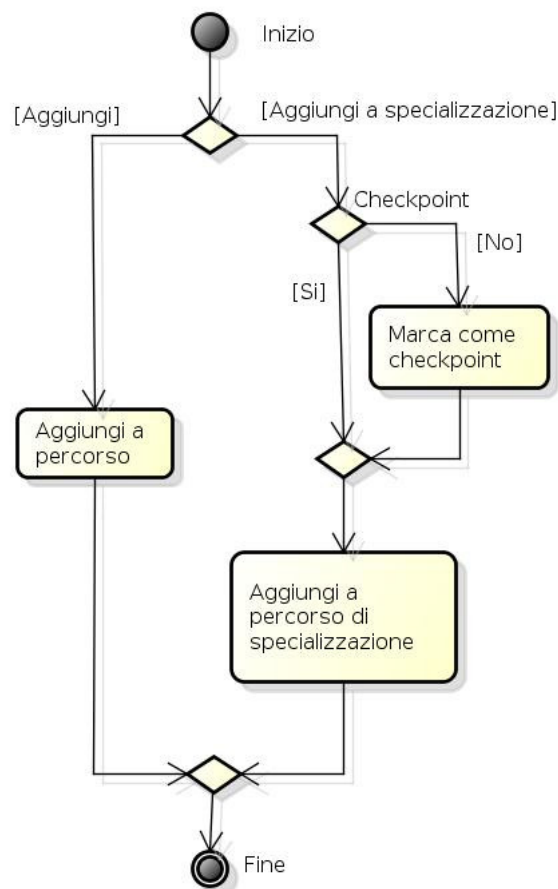


Figura 22: Aggiungi frame

Per aggiungere un frame al percorso si hanno due possibilità:

- Aggiungere il frame in coda al frame visualizzato nell'editor;
- Marcare il frame corrente visualizzato come checkpoint, se non già marcato, e aggiungerlo al percorso di specializzazione.

5.23 Rimuovi frame da percorso

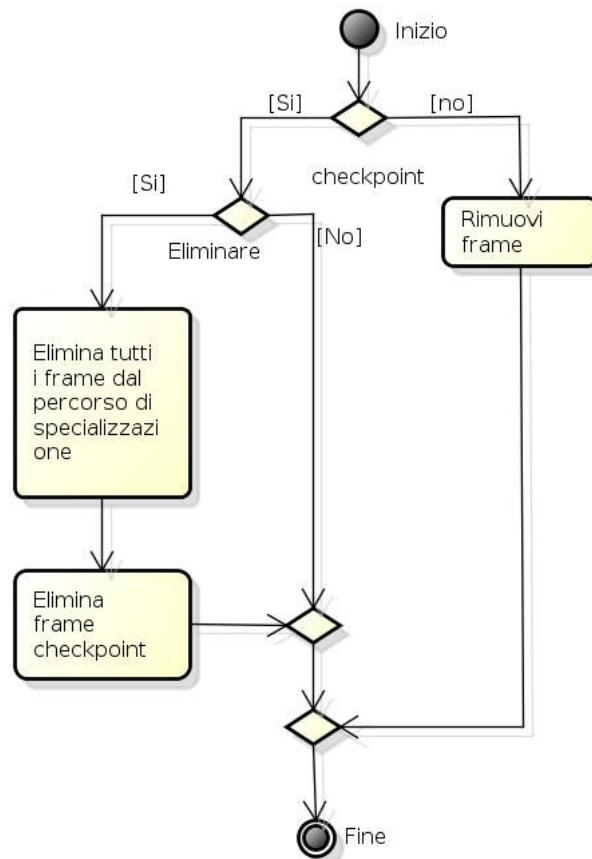


Figura 23: Rimuovi frame dal percorso

Quando l'utente rimuove un frame viene controllato se è un checkpoint. In caso negativo si rimuove il frame dal percorso, viceversa se c'è la conferma dell'utente si eliminano prima tutti i frame del percorso di specializzazione e successivamente si elimina il frame dal percorso.

6 Stime di fattibilità e di bisogno di risorse

7 Tracciamento della relazione componenti - requisiti

8 Descrizione delle tecnologie utilizzate

8.1 JavaScript

JavaScript_G è un linguaggio di scripting lato client_G orientato agli oggetti, comunemente usato nei siti web, ed interpretato dai browser_G. Ciò permette di alleggerire il server_G dal peso della computazione, che viene eseguita dal client_G. Questo è un

aspetto molto importante per lo sviluppo del capitolato Premi. La caratteristica principale di JavaScript_G ‘e, appunto, quella di essere un linguaggio interpretato: il codice non viene compilato, ma interpretato, dal browser_G. Essendo molto popolare e ormai consolidato, JavaScript_G può essere eseguito dalla maggior parte dei browser_G, sia desktop che mobile, grazie anche alla sua leggerezza. Uno degli svantaggi di questo linguaggio è che ogni operazione che richieda informazioni che devono essere recuperate da un database_G deve passare attraverso un linguaggio che effettui esplicitamente la transazione, per poi restituire i risultati a JavaScript_G. Tale operazione richiede l’aggiornamento totale della pagina, ma grazie ad Meteor è possibile superare questo limite.

8.2 HTML5

HTML5_G verrà utilizzato per definire la struttura dell’applicazione web Premi. Tale struttura sarà completamente separata dalla presentazione, che verrà realizzata tramite CSS3_G. HTML5_G presenta, rispetto ad HTML—_G 4, diversi vantaggi per lo svolgimento del progetto:

- Introduzione di elementi di controllo per i menu di navigazione (tag `<nav>`);
- Introduzione di elementi specifici per l’inserimento di contenuti multimediali (tag `<video>` e `<audio>`)

] e molti altri.

8.3 CSS3

CSS_G (Cascading Style Sheet) è un linguaggio pensato con lo scopo di definire l’aspetto di pagine HTML_G e non solo, che devono presentare un collegamento al loro foglio di stile nell’header (la parte del documento HTML_G che introduce un gruppo di ausili introduttivi o di navigazione). Grazie ai CSS_G, ‘e possibile una completa separazione tra la presentazione (cioè l’aspetto grafico delle pagine web) ed i contenuti delle pagine stesse. Ciò semplifica la comprensione, la manutenzione e la portabilità’. Rispetto a CSS2, CSS3_G introduce funzionalità grafiche più avanzate.

8.4 Angular

Nello sviluppo software AngularJS (più comunemente noto come “Angular”) è un framework_G per applicazioni web open-source_G gestito da Google e da una comunità di singoli sviluppatori e aziende per affrontare molte delle sfide incontrate nello sviluppo di applicazioni una sola pagina. Angular mira a semplificare lo sviluppo e la sperimentazione di tali applicazioni, fornendo un quadro di riferimento per l’architettura Model-View-Controller (MVC) lato client_G, insieme ai componenti comunemente utilizzati in applicazioni.

Le librerie di AngularJS funzionano leggendo prima la pagina HTML_G, che ha incorporati in essa tag attributo personalizzati aggiuntivi. Angular interpreta quegli attributi come direttive per legare parti della pagina (in ingresso o in uscita) a un modello che è rappresentato da variabili JavaScript_G standard. I valori di tali variabili

JavaScript_G possono essere impostati manualmente all'interno del codice, o recuperati da risorse JSON_G statiche o dinamiche.

AngularJS è costruito attorno alla convinzione che la programmazione dichiarativa deve essere utilizzata per la costruzione di interfacce utente e il collegamento dei componenti software, mentre la programmazione imperativa è più adatta per definire la logica di business di un'applicazione. Il framework_G adatta ed estende il tradizionale HTML_G per presentare contenuti dinamici attraverso il "Two-Way Data Binding" che consente la sincronizzazione automatica di modelli e viste, con il risultato di migliorare la testabilità e le prestazioni.

I nostri obiettivi nella scelta di Angular:

- Disaccoppiare manipolazione del DOM_G dalla logica dell'applicazione.
La difficoltà di questo è notevolmente influenzata dal modo in cui il codice è strutturato.
- Disaccoppiare il lato client_G di un'applicazione dal lato server_G.
Questo permette allo sviluppo di progredire in parallelo, e permette il riutilizzo del codice di entrambe le parti.
- Fornire la struttura per il percorso di creazione di un'applicazione:
dalla progettazione dell'interfaccia utente, attraverso la scrittura della logica, al collaudo.
- Angular implementa il pattern MVC_G per separare la presentazione, i dati e le componenti logiche. Usando la Dependency Injection, che verrà descritta dettagliatamente più avanti, Angular porta servizi tradizionalmente lato server, come i Controllers dipendenti dalle Viste, al lato client_G delle applicazioni Web. Di conseguenza, la maggior parte del carico sul server può essere ridotto.

Perchè Angular:

- **Data Binding:**
è un modo automatico di aggiornamento della vista ogni volta che il modello cambia, così come l'aggiornamento del modello ogni volta che cambia la vista. Ciò elimina la manipolazione del DOM_G dalla lista delle cose di cui occuparsi.
- **Controller:**
definiscono il comportamento dietro gli elementi del DOM_G. AngularJS permette di esprimere il comportamento in una forma leggibile pulita, registrando callback_G o guardando le modifiche dei modelli.
- **JavaScript:**
A differenza di altri framework_G, non vi è alcuna necessità di ereditare da tipi di proprietà, per wrappare i modelli. I modelli in Angular sono semplici vecchi oggetti JavaScript_G. Questo rende il codice facile testare, mantenere, e facilita il riutilizzo..

- **Comunicazione con il Server:**

AngularJS fornisce servizi integrati basati su XHR_G , nonché vari altri backends, utilizzando librerie di terze parti. Le Promises semplificano ulteriormente il codice per la gestione di ritorno asincrona dei dati.

- **Direttive:**

Le direttive sono una caratteristica unica e potente disponibile solo in Angular. Consentono di inventare nuova sintassi $HTML_G$, specifica per l'applicazione.

- **Componenti Riutilizzabili:**

Usando le direttive per creare componenti riutilizzabili. Un componente consente di nascondere la complessa struttura del DOM_G , CSS_G , e il comportamento. Questo permette di concentrarsi sia su ciò che l'applicazione deve fare o su come l'applicazione appare separatamente.

- **Integrabile:**

AngularJS lavora molto bene con altre tecnologie. E' possibile aggiungere tanto o poco di AngularJS a una pagina esistente a seconda delle esigenze. Molte altri $framework_G$ richiedono di essere totalmente inclusi. Poichè AngularJS non ha uno stato globale più applicazioni possono essere eseguite su una singola pagina.

- **Iniettabile:**

La dependency injection in AngularJS consente di descrivere in modo dichiarativo come l'applicazione è collegata. Ciò significa che l'applicazione non ha bisogno del metodo `main()`. Inoltre ogni componente che non si adatta alle nostre esigenze può essere facilmente sostituita.

- **Testabile:** AngularJS è stato progettato da zero per essere verificabile.

8.5 Meteor

Sebbene Meteor sia frequentemente comparato a Backbone.js e AngularJS per il suo design reattivo, esso è invece un $framework_G$ completo, in grado di utilizzare entrambi come moduli.

Le sue principali motivazioni progettuali sono elencate di seguito:

- Al posto di essere il $server_G$ ad inviare interi file HTML al client, Meteor invia solo i dati minimi necessari per rirenderizzare la parte della pagina che è cambiata. Ciò consente la creazione di applicazioni a bassa latenza di una sola pagina che evitano il totale refresh della pagina.
- Unifica il linguaggio (Javascript $_G$) utilizzato sul $client_G$ e sul $server_G$.
- La stessa API può essere utilizzata sia sul $server_G$ e il $client_G$ per interrogare il database. Nel browser $_G$, un'implementazione di MongoDB in memoria chiamata Minimongo permette l'interrogazione una cache di documenti che sono stati inviati al $client_G$.

- La compensazione di latenza: sul client_G , Meteor effettua il prefetch dei dati e simula modelli facendo sembrare che le chiamate di metodo sul server ritornino istantaneamente.
- Assoluta reattività: Tutti i livelli, dal database ai template, si aggiornano automaticamente quando necessario.
- Atmosfera, repository di pacchetti di Meteor, detiene più di 5.200.
- Meteor è stato progettato per essere facile da imparare, anche per i principianti.

9 Mockup Interfaccia [FACOLTATIVO]

Riferimenti bibliografici