

Specifica Tecnica

Informazioni sul documento

Nome Documento Specifica Tecnica

 $\begin{array}{ccc} \textbf{Versione} & 1.0.0 \\ \textbf{Stato} & Formale \\ \textbf{Uso} & Esterno \\ \textbf{Data Creazione} & 18-05-2015 \\ \textbf{Data Ultima Modifica} & 18-05-2015 \\ \end{array}$

Redazione Fossa Manuel, Petrucci Mauro

ApprovazioneTollot PietroVerificaGabelli PietroLista distribuzioneLateButSafe

Prof. Tullio Vardanega Prof. Riccardo Cardin Proponente Zucchetti S.p.a.



Registro delle modifiche

Tab 1: Versionamento del documento

Versione	Autore	Data	Descrizione
1.0.0	Tollot Pietro	13-04-2015	Approvazione del documento
0.7.0	Petrucci Mauro	08-04-2015	Apportate le modifiche segnalate dal verificatore Fossa Manuel
0.3.0	Petrucci Mauro	25-03-2015	Aggiunta dei contenuti
0.2.0	Fossa Manuel	24-03-2015	Aggiunta dei contenuti
0.1.0	Busetto Matteo	20-03-2015	Stesura dello scheletro del documento



Storico

$\operatorname{pre-RR}$

Versione 1.0	Nominativo
Redazione	Fossa Manuel, Petrucci Mauro
Verifica	Gabelli Pietro
Approvazione	Tollot Pietro

Tab 2: Storico ruoli pre-RR



Indice

1	Intr	ntroduzione					
	1.1	Scopo del documento	5				
	1.2	Scopo del Prodotto					
	1.3	Glossario	1				
	1.4	Riferimenti	5				
		1.4.1 Normativi	1				
		1.4.2 Informativi	5				
2	Strı	rumenti					
	2.1	HTML	7				
	2.2	JavaScript	7				
	2.3	jQuery					
		Angular.js					
		Node.js					
		MongoDB					



Sommario

Il presente documento contiene la specifica tecnica delle componenti che costituiscono il prodotto software Premi.



1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di definire la progettazione ad alto livello del progetto Premi. Verrà presentata l'architettura generale secondo la quale saranno organizzate le varie componenti software e saranno descritti i Design Pattern utilizzati.

1.2 Scopo del Prodotto

Lo scopo del progetto_g è la realizzazione un software_g per la creazione ed esecuzione di presentazioni multimediali favorendo l'uso di tecniche di storytelling e visualizzazione non lineare dei contenuti.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento Glossario_v.1.0.0.pdf. Ogni occorrenza di vocaboli presenti nel Glossario è marcata da una "g" minuscola in pedice.

1.4 Riferimenti

1.4.1 Normativi

- Capitolato d'appalto C4: Premi: Software_g di presentazione "better than Prezi" http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf;
- Norme di Progetto_g: NormeDiProgetto_v.1.0.0.pdf;
- Analisi dei Requisiti: AnalisiDeiRequisiti_v.1.0.0.pdf;
- Piano di qualifica: PianoDiQualifica v.1.0.0.pdf;
- Piano di progetto: PianoDiProgetto v.1.0.0.pdf.

1.4.2 Informativi

- Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1995;
- Descrizione dei Design Pattern http://sourcemaking.com/design_patterns;
- Ingegneria del software_g Ian Sommerville 9a Edizione (2010):
- Slide del docente per l'anno accademico 2014/2015 reperibili al sito http://www.math.unipd.it/~tullio/IS-1/2014/;
- MongoDB: http://docs.mongodb.org/manual/;



- Node.js: https://nodejs.org/documentation/;
- Angular.js: https://docs.angularjs.org/tutorial;
- jQuery: http://api.jquery.com/;
- Impress.js: https://github.com/bartaz/impress.js/.



2 Strumenti

2.1 HTML

Si è deciso di utilizzare HTML5 e CSS3 per la presentazione grafica dell'applicazione web. Si è scelto di utilizzare HTML5 al posto di xHTML 1.1 perchè ormai è diventato uno standard de facto e permette una maggiore integrazione con i linguaggi di scripting e contenuti multimediali di cui questa applicazione fa forte uso.

• Vantaggi:

- Multi piattaforma: Poiché l'applicazione deve essere disponibile sia su dispositivi desktop che mobile HTML5 permette la creazione di strutture responsive in grado di adattarsi alle dimensioni dello schermo;
- Integrazione con linguaggi di scripting: Con HTML5 c'è una maggiore integrazione con i linguaggi di scripting come javacript questo permetterà di rendere l'applicazione dinamica;
- Nessuna installazione: Il fatto che l'applicazione sia sviluppata con tecnologie web quali HTML permetterà all'utente finale di poter utilizzare il prodotto senza doverlo scaricare e installare.

• Svantaggi:

- **Browser**: E' possibile che i browser meno recenti abbiano difficoltà ad interpretare correttamente le informazioni contenute nelle pagine, rendendo difficile, se non impossibile, l'utilizzo dell'applicazione con questo linguaggio.

2.2 JavaScript

JavaScript è un linguaggio di scripting lato client orientato agli oggetti, comunemente usato nei siti web, ed interpretato dai browser. Ciò permette di alleggerire il server dal peso della computazione, che viene eseguita dal client. Essendo molto popolare e ormai consolidato, JavaScript può essere eseguito dalla maggior parte dei browser, 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 deve passare attraverso un linguaggio che effettui esplicitamente la transazione, per poi restituire i risultati a JavaScript. Tale operazione richiede l'aggiornamento totale della pagina ma è stato superato questo problema adottando delle tecnologie con funzionamento asincrono.

- 2.3 jQuery
- 2.4 Angular.js
- 2.5 Node.js
- 2.6 MongoDB



3 Descrizione dei singoli componenti



3.1 View

3.1.1 Web View

Tipo, obiettivo e funzione del componente: questo livello costituisce l'interfaccia del software utilizzabile dagli utenti mediante pagine web.

Relazioni d'uso di altre componenti: il componente è costituito dal package Pages e comunica con il Controller per rendere possibile la gestione del proprio profilo, la gestione delle presentazioni e per controllare i dati in transito per il sistema, dovuti all'interazione dell'utente con lo stesso.

3.1.1.1 Premi.WebView.Pages.IndexPage

Tipo, obiettivo e funzione del componente: la classe IndexPage definisce la struttura, e la conseguente visualizazione, della pagina web che consente ad un utente di effettuare login e registrazione al sistema e di passare alla visualizzazione della classe Loader.

Attività svolte e dati trattati: la classe definisce la struttura della pagina web che consente agli utenti di autenticarsi e registrarsi al sistema. Essa resta in attesa che un utente inserisca i dati necessari per l'autenticazione o la registrazione al sistema oppure che l'utente decida di andare nella pagina Loader; il controllo poi passerà alla classe [[[[[[[CONTROLLER LOGIN]]]]]]]]].

3.1.1.2 Premi.WebView.Pages.Loader

Tipo, obiettivo e funzione del componente: la classe Loader definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra all'utente non autenticato le presentazioni scaricate precedentemente in locale permettondogli di eseguirle ed eliminarle e all'utente autenticato tutte le proprie presentazioni presenti sul proprio spazio nel server dandogli la possibilità di eseguirle, modificarle, eliminarle o scaricarle in locale.

Relazioni d'uso di altre componenti: la classe Loader utilizza i metodi messi a disposizione dalle classi [[[[[[[[CONTROLLER LOADER ESECUZIONE]]]]]]]]]] per il caricamento delle presentazioni presenti sul server che l'utente vuole eseguire e [[[[[[[CONTROLLER LOADER MODIFICA]]]]]]]] per il caricamento delle presentazioni che l'utente vuole modificare, presenti nel package Controller. La classe Loader, inoltre, utilizza i metodi messi a disposizione delle classi [[[[[[[CONTROLLER ELIMINAZIONE PRESENT.]]]]]]]]] per l'eliminazione delle presentazioni dal server e [[[[[[[CONTROLLER SCARICAMENTO MANIFEST]]]]]]]]]] per scaricare una presentazione in locale.

Interfacce con e relazioni d'uso e da altre componenti: la classe vienne utilizzata da



Attività svolte e dati trattati:

3.1.1.3 Premi.WebView.Pages.Home

Tipo, obiettivo e funzione del componente: Relazioni d'uso di altre componenti: Interfacce con e relazioni d'uso e da altre componenti: Attività svolte e dati trattati:

3.1.1.4 Premi.WebView.Pages.ProfileEditing

Tipo, obiettivo e funzione del componente: Relazioni d'uso di altre componenti: Interfacce con e relazioni d'uso e da altre componenti: Attività svolte e dati trattati:

3.1.1.5 Premi.WebView.Pages.Execution

Tipo, obiettivo e funzione del componente: Relazioni d'uso di altre componenti: Interfacce con e relazioni d'uso e da altre componenti: Attività svolte e dati trattati:

3.1.1.6 Premi.WebView.Pages.Editing

Tipo, obiettivo e funzione del componente: Relazioni d'uso di altre componenti: Interfacce con e relazioni d'uso e da altre componenti: Attività svolte e dati trattati: