

Exigram

**System
Design
Document**

Membri del team

Iannaccone Davide 05121 05292

Esposito Domenico 05121 05478

Data	Cambiamenti	Autori
05/Dicembre/2019	Creato il documento	Iannaccone Davide Esposito Domenico
06/Dicembre/2019	Revisione documento e modifica formattazione	Esposito Domenico
12/Dicembre/2019	Modifica documento	Iannaccone Davide
12/Febbraio/2020	Modifica e Revisione documento	Iannaccone Davide

Sommario

1. Introduzione.....	4
1.1 Obiettivi del Sistema	4
1.2 Design Goals	4
1.3 Riferimenti.....	5
1.4 Panoramica.....	6
2. Architettura del Sistema proposto.....	6
2.1 Panoramica.....	6
2.2 Decomposizione in Sistemi	6
2.3 Mapping HW/SW	9
2.4 Gestione dati persistenti.....	10
2.5 Controllo accessi e sicurezza.....	10
2.6 Controllo flusso globale del sistema.....	11
2.7 Condizioni limite.....	11
2.7.1 Avvio del sistema.....	11
2.7.2 Terminazione del sistema.....	11
2.7.3 Fallimento del sistema.....	12
3 Glossario	12

1. Introduzione

1.1 Obiettivi del Sistema

Una tendenza corrente nella società odierna è quello di mettere in comunicazione migliaia di persone di tutto il mondo e di tutte le nazionalità attraverso un social network. Questi ultimi forniscono da molto tempo numerosi servizi per formare e rafforzare rapporti con amici presenti nelle zone circostanti, ma anche con persone estranee molto lontane.

Se ad una persona piace condividere le sue esperienze e i suoi stati d'animo per farsi conoscere, lo strumento migliore è un social, ed Exigram è uno di questi.

1.2 Design Goals

I design goals identificati per il sistema Exigram sono i seguenti:

Criteri di Performance

- Tempo di risposta:
Exigram deve essere più reattivo possibile per ogni azione che all'utente comune è permesso compiere all'interno del sistema. Per la conferma di creazione di un post e la sua pubblicazione il tempo di risposta deve essere inferiore ai 3 secondi, anche se il sistema è sottoposto ad un picco elevato. Il tempo di reindirizzamento per la visualizzazione di pagine utente o post deve essere inferiore ai 2 secondi e per il caricamento dei dati all'interno del database il sistema deve poter rispondere in tempi brevi, anche se tutto questo è condizionato dalla qualità della connessione dell'utente al sistema.
- Memoria:
Il sistema utilizza un database relazionale per memorizzare tutti i dati. Le dimensioni del database non influenzeranno le prestazioni, fino ad un determinato limite.

Criteri di Affidabilità

- Affidabilità:
Il sistema deve proteggere i pochi dati sensibili richiesti degli utenti registrati tramite un servizio di autenticazione sicuro. Eventuali input inseriti dall'utente incorrettamente o non validi verranno segnalati come errori. I dati relativi ai post dei propri Companion che verranno visualizzati all'utente verranno salvate in modo sicuro nel database del sistema.
- Disponibilità:
Exigram deve poter essere disponibile 24 ore su 24, 7 giorni su 7 per gli utenti che vogliono usufruirne, con un server dedicato al sistema.
- Security:
L'intero sistema è accessibile a tutti gli utenti che hanno eseguito una registrazione ad Exigram, fatto eccezione per l'area riservata agli amministratori. Per accedere alle funzionalità di Exigram è necessario eseguire l'accesso al sistema, inserendo le proprie credenziali.

Criteri di Costo

- Costi di Sviluppo:

Il costo complessivo del progetto sarà di circa 150 ore per persona nel progetto e lo sviluppo del sistema. Il server esterno verrà messo su un server pagato annualmente ad una cifra modica.

Criteri di Manutenzione

- Estendibilità:

Il sistema sarà progettato in modo tale da poter essere il più flessibile possibile, in cui si possono inserire facilmente delle modifiche o nuove implementazioni, utilizzando un linguaggio di markup HTML5.

- Modificabilità:

Bisogna garantire che il codice implementato possa essere leggibile e commentato per i futuri programmatori che lavoreranno sul progetto Exigram.

- Portabilità:

Il sistema non necessita di nessun particolare sistema operativo, ma necessita esclusivamente di un browser su un dispositivo che possa usufruire di un ISP.

Criteri utenti finali

- Usabilità:

Il sistema deve essere user friendly e molto intuitivo, dove anche l'utente meno esperto può facilmente apprendere il suo funzionamento.

1.3 Riferimenti

Il materiale di riferimento utilizzato per la realizzazione del progetto e per la stesura di questo stesso documento comprende:

Libro di Testo: Object-Oriented Software Engineering Bruegge, A.H. Dutoit.

Slide fornite dal Professore Andrea De Lucia reperibili sulla piattaforma e-learning

1.4 Panoramica

Exigram non riceverà alcuna informazione da qualsiasi dispositivo supportato, le uniche informazioni che riceverà saranno quelle che l'utente, di propria iniziativa, inserisce per farsi conoscere dagli altri utenti. Ogni informazione inserita può essere modificata per essere sempre aggiornata.

Exigram può essere visto come un tuo caro amico che può farti conoscere innumerevoli persone come e quando vuoi, scambiando informazioni come si preferisce.

Dovute restrizioni aiuteranno l'utente ad utilizzare e gestire il proprio dispositivo e le proprie informazioni in modo anche più responsabile per evitare possibili disagi.

2. Architettura del Sistema proposto

2.1 Panoramica

Exigram è una piattaforma nella quale è possibile condividere fotografie con i propri Companion.

Le tipologie di utenti si dividono in: Amministratore, Utente registrato e Utente non registrato.

L'Amministratore avrà la possibilità di bandire i profili degli Utenti che sono stati segnalati e di eliminare i post che sono stati segnalati, inoltre avrà la possibilità visionale la lista di tutti gli Utenti esistenti e avrà la possibilità di modificare il proprio profilo. L'Utente registrato avrà la possibilità di aggiungere ed eliminare Utenti dalla propria lista companion, di caricare post che saranno visti da altri Utenti, di modificare il proprio profilo per mantenerlo aggiornato e segnalare post e Utenti se ritenuti inappropriati. L'Utente non registrato avrà la possibilità di creare un account per entrare a far parte della community di Exigram.

L'architettura del sistema si presenta con una architettura Client/Server per i seguenti motivi:

- Portabilità: il sistema potrà essere utilizzato su diversi sistemi operativi;
- Trasparenza: il sistema è in grado di fornire i servizi agli Utenti senza interferenze;
- Performance: il sistema garantirà reattività agli utenti collegati ma essendo un sito web molto dipenderà dalla qualità della connessione;
- Scalabilità: il sistema sarà in grado di gestire un gran numero di Utenti connessi in contemporanea;
- Flessibilità: il sistema garantirà un'interfaccia grafica intuitiva.

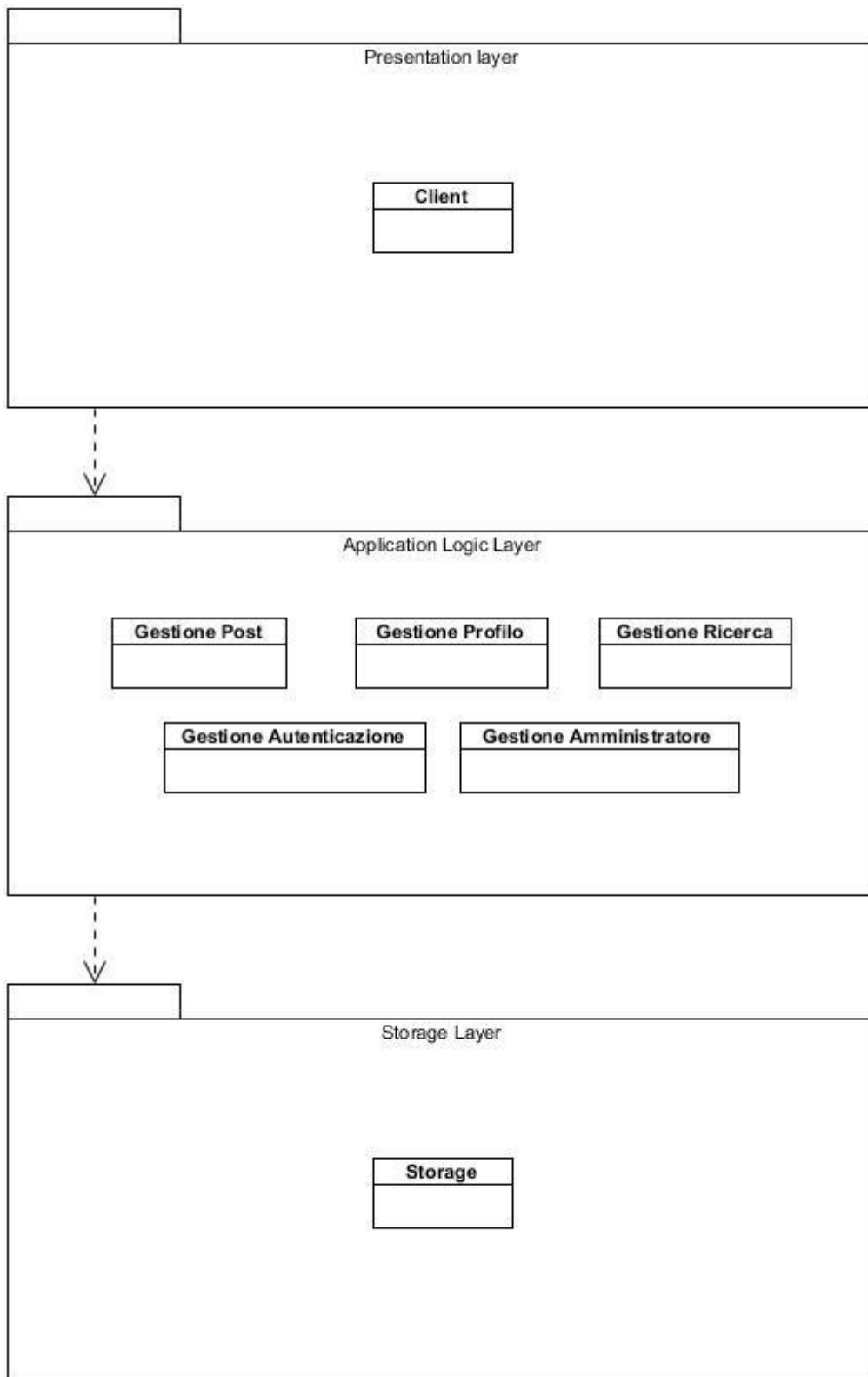
2.2 Decomposizione in Sistemi

Per realizzare il sistema EXIGRAM è stato utilizzato lo stile a tre strati Client/Server. In particolare i tre strati si occupano di:

- Il Presentation Layer:
Include tutte le interfacce grafiche e in generale i boundary object con cui interagisce l'utente;
- L'Application Logic Layer:
Si occupa della gestione della logica del sistema;

- Lo Storage Layer:

Effettua la memorizzazione degli oggetti persistenti tramite DBMS

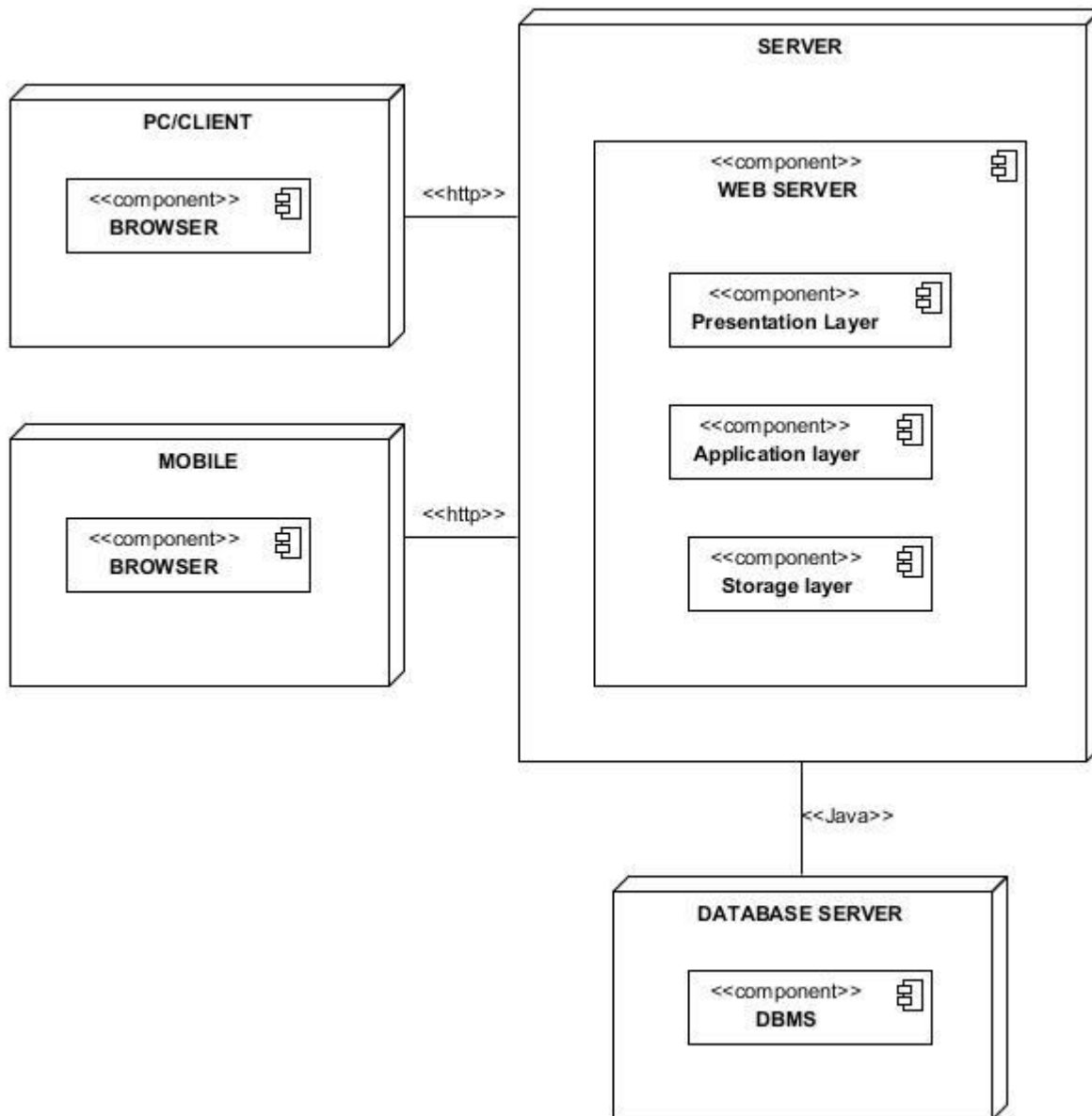


Il sistema EXIGRAM avrà le seguenti gestioni:

- Gestione Autenticazione:
Serie di funzionalità per l'autenticazione sulla piattaforma e che permette di creare un nuovo account a EXIGRAM inserendo i propri dati.
- Gestione Profilo:
Serie di funzionalità per la gestione del Profilo sulla piattaforma.
- Gestione Post:
Serie di funzionalità per la gestione dei Post sulla piattaforma.
- Gestione Amministratore:
Serie di funzionalità per la gestione delle segnalazioni sulla piattaforma.
- Gestione Ricerca:
Serie di funzionalità per la ricerca di utenti e post sulla piattaforma.

2.3 Mapping HW/SW

Il sistema sarà costituito da un server unico e da una serie di browser (client) per accedere al sistema EXIGRAM. Il client ed il server saranno connessi tramite una rete che utilizzerà il protocollo TCP/IP. Al server si collegano i client ed il database, il tipo di utente è determinato in fase di autenticazione controllando nel database il tipo di utente che corrisponde allo Username inserito in quel client. I client dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità.



2.4 Gestione dati persistenti

Si rimanda al documento “SDD-GestioneDatiPersistenti-EXIGRAM” allegato al presente.

2.5 Controllo accessi e sicurezza

Al sistema EXIGRAM possono accedere diversi attori con vari compiti e permessi. Il controllo degli accessi è garantito tramite l'utilizzo di username e password per ogni tipo di utente, che verranno richieste per ogni singolo accesso. Per schematizzare al meglio il controllo degli accessi abbiamo suddiviso per tipologia di utente le azioni consentite, al fine di ottenere una visione più compatta e dettagliata grazie ad una matrice degli accessi riportata di seguito:

Attori Oggetti	Utente non Registrato	Utente Registrato	Amministratore
Autenticazione		✓ Login ✓ Logout ✓ Recupera password	✓ Login ✓ Logout ✓ Recupera password
Registrazione	✓ Registrazione		
Account		✓ Visualizza utente ✓ Modifica profilo ✓ Modifica password ✓ Modifica immagine ✓ Aggiungi companion ✓ Rimuovi companion ✓ Visualizza notifiche ✓ Elimina utente ✓ AggiungiSegnalazioneUtente	✓ Visualizza utente ✓ Modifica profilo ✓ Modifica immagine ✓ Modifica password
Post		✓ Carica post ✓ Visualizza post ✓ Rimuovi post ✓ Inserisci commento ✓ Aggiungi votazione ✓ Modifica didascalia ✓ Rimuovi commento ✓ AggiungiSegnalazionePost	✓ Visualizza post ✓ Rimuovi post ✓ Aggiungi votazione
Amministrazione			✓ Visualizza lista utenti ✓ Visualizza segnalazioni ✓ Rimuovi segnalazione ✓ Bandisci utente
Ricerca		✓ Ricerca utente ✓ Ricerca tag	

2.6 Controllo flusso globale del sistema

Il sistema è caratterizzato da un sito web accessibile da browser e da un WebServer, attivo 24h, che deve provvedere a gestire gli accessi concorrenti da parte degli utenti. Quando un utente fa il login e sottomette i propri dati, vi è un accesso al DB per il controllo dell'esistenza dell'utente. Dopo la conferma, l'utente può accedere a diverse operazioni messe a disposizione dal sistema. Ogni operazione è indipendente dalle altre ed è attivabile dalla pressione di un bottone (onClick).

Il controllo del flusso è gestito da Spring MVC: quando un client manda una richiesta ad uno specifico URL, la richiesta viene ricevuta dal Web Container, nel nostro caso Tomcat, che visualizza il web.xml e cerca la Servlet o i Filters mappati su quel particolare URL. Il Web Container delega anche Servlet o Filter nel processare la richiesta.

Il Web Container (Tomcat) è responsabile anche della creazione di istanze di Servlet e Filter e nella invocazione dei loro vari metodi che compongono il proprio ciclo vita (init(), service(), destroy()). Nel caso delle richieste HTTP, le gestisce HttpServlet e dipendentemente ai vari metodi di richiesta HTTP, doSomething() viene invocato dal container, come ad esempio: doGet() per processare la richiesta GET e doPost() per processare la richiesta POST.

Per abilitare Spring MVC bisogna dichiarare il DispatcherServlet dal jar di Spring MVC all'interno del web.xml. Questa Servlet ascolta un path URL '/*', indicando che tutte le richieste sono mappate al DispatcherServlet. Se la richiesta corrisponde al percorso URL del DispatcherServlet, allora la richiesta verrà gestita da Spring MVC, altrimenti no. Il DispatcherServlet passa la richiesta ad un controller specifico dipendente dal tipo di URL richiesto, per sapere a chi passare queste richieste viene utilizzato il @RequestMapping.

È possibile utilizzare anche i decorator @GetMapping, @PutMapping e @PostMapping per indicare richieste specifiche, utilizzate accuratamente per specificare i tipo di operazione di cui abbiamo bisogno. Per le classi Controller viene utilizzato invece il decorator @RestController dato che il nostro caso è quello di fornire una applicazione RESTful.

Dopo aver eseguito la richiesta, il Controller ritorna un Logical View Name e un modello (model) al DispatcherServlet e consulta i risolutori view finché una view effettiva viene determinata per visualizzare l'output. Il DispatcherServlet allora contatta la view scelta, nel nostro caso Angular, e determina l'output in base al modello dei dati (model data).

Questo output renderizzato viene restituito al client come una risposta HTTP. Durante il suo processo per ritornare indietro, può passare per ogni Filtro configurato, per trasformare la risposta in un JSON.

2.7 Condizioni limite

Le condizioni limite riguardano l'accensione e lo spegnimento del sistema per il solo lato Server.

2.7.1 Avvio del sistema

Il sistema dopo essersi avviato presenta un'interfaccia ai client.

2.7.2 Terminazione del sistema

Prima della disattivazione del sistema verranno disconnessi tutti i client connessi al sistema. Per non incorrere in problemi basta spegnere il server tramite riga di comando.

2.7.3 Fallimento del sistema

In caso di errori di tipo hardware o software la persistenza dei dati è comunque garantita in quanto i dati sono gestiti dal DBMS.

3 Glossario

1. **UPVOTES:** Sistema per definire se la foto visionata è piaciuta all'utente; non sono presenti riferimenti agli utenti quindi il voto è totalmente anonimo.
2. **DOWNVOTES:** Sistema per definire se la foto visionata non è piaciuta all'utente; non sono presenti riferimenti agli utenti quindi il voto è totalmente anonimo.
3. **TAG:** Etichetta testuale che viene assegnata dall'utente per categorizzare l'oggetto caricato, utilizzabile nella ricerca per trovare solo gli oggetti desiderati.
4. **SEGUIRE:** Sistema per il quale un utente riceve notifiche riguardanti azioni compiute da un altro utente, ad esempio caricare file multimediali sulla piattaforma.
5. **COMMENTO:** Sistema secondo il quale è possibile scrivere sotto l'oggetto desiderato per esprimere un proprio parere o per rispondere ad un altro commento.
6. **COMPANION:** Utente aggiunto alla lista delle persone seguite, si visualizzeranno di questa nella pagina principale.
7. **POST:** È un'immagine con annessi tag e, opzionalmente, anche una didascalia, che può essere pubblicata sul sito.
8. **DIDASCALIA:** Stinga di testo che può essere affiancata ad un post per descriverne il contenuto o per lasciare un messaggio ai companion.

9. **PUBBLICARE:** Aggiungere un post visibile a tutti sul sito.