System Design

1. Introduzione

1.1. Scopo del sistema

TomMASO è una piattaforma **semplice e user-friendly** che permette agli utenti registrati di creare un proprio spazio tramite l'utilizzo di un'intuitiva interfaccia grafica. La piattaforma permette agli utenti di creare blog, scrivere, modificare i propri articoli (tramite l'utilizzo del linguaggio **Markdown**) e di interagire con gli altri utenti registrati.

1.2. Obiettivi del sistema

La piattaforma è stata progettata con i seguenti obiettivi:

1.2.1. Obiettivi di usabilità

Il sistema deve essere fruibile al maggior numero di persone, pertanto si auspica l'utilizzo di un'interfaccia user-friendly e un editor che consenta anche a chi si approccia per la prima volta al linguaggio Markdown di creare i propri articoli sul blog.

1.2.2. Obiettivi di performance

Frequenti rallentamenti della piattaforma potrebbero scoraggiare l'utilizzo da parte di alcuni utenti, pertanto si vuole garantire che il servizio sia prestante e garantisca una risposta entro un secondo. Il server dovrebbe garantire un uptime di almeno il 95%.

1.2.3. Obiettivi di affidabilità

Una failure durante la gestione di una richiesta effettuata da un client non deve compromettere il funzionamento del resto del sistema ed esso deve poter continuare a servire gli altri utenti normalmente.

Il sistema deve essere protetto dalle vulnerabilità più comuni quali XSS e SQL-injection in quanto potrebbero compromettere la sicurezza degli utenti e portare al furto di dati sensibili.

Le password non devono essere salvate in chiaro in quanto esistono utenti che riutilizzano le password su altre piattaforme e in caso di attacchi potrebbero essere compromessi anche account esterni alla piattaforma.

1.2.4. Criteri di portabilità

Il sistema deve essere installabile su tutti i dispositivi con sistemi Windows, MacOS e Linux per garantire la possibilità in futuro di distribuire il sistema su server diversi, pertanto si è deciso di utilizzare un sistema di virtualizzazione.

1.3. Definizioni, acronimi e abbreviazioni

RAD - Requirement Analysis Document

MVC - Model View Control

JSP - Java Server Page

GUI – Graphical User Interface

JDBC – Java DataBase Connectivity

SHA – Secure Hash Algorithm

HTTP - Hypertext Transfer Protocol

Salt - Una sequenza casuale di bit utilizzata assieme ad una password come input a una funzione unidirezionale

1.4. Riferimenti

Requirement Analysis Document di TomMASO

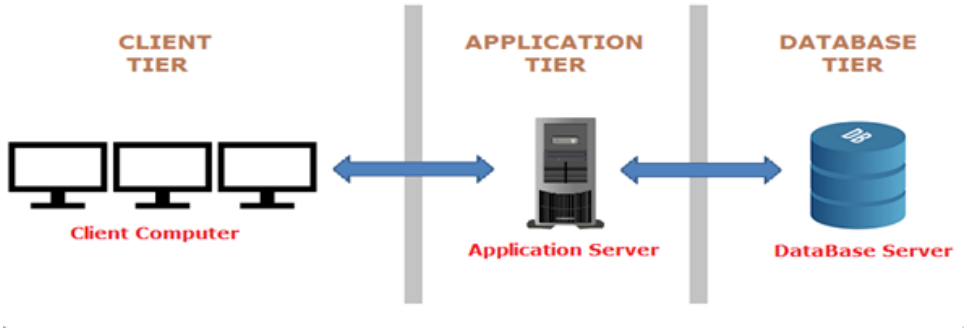
1.5. Panoramica

Il documento mostra i dettagli della progettazione della piattaforma TomMASO. Verranno analizzati nel dettaglio i sottosistemi che lo compongono.

2. Architettura Software Proposta

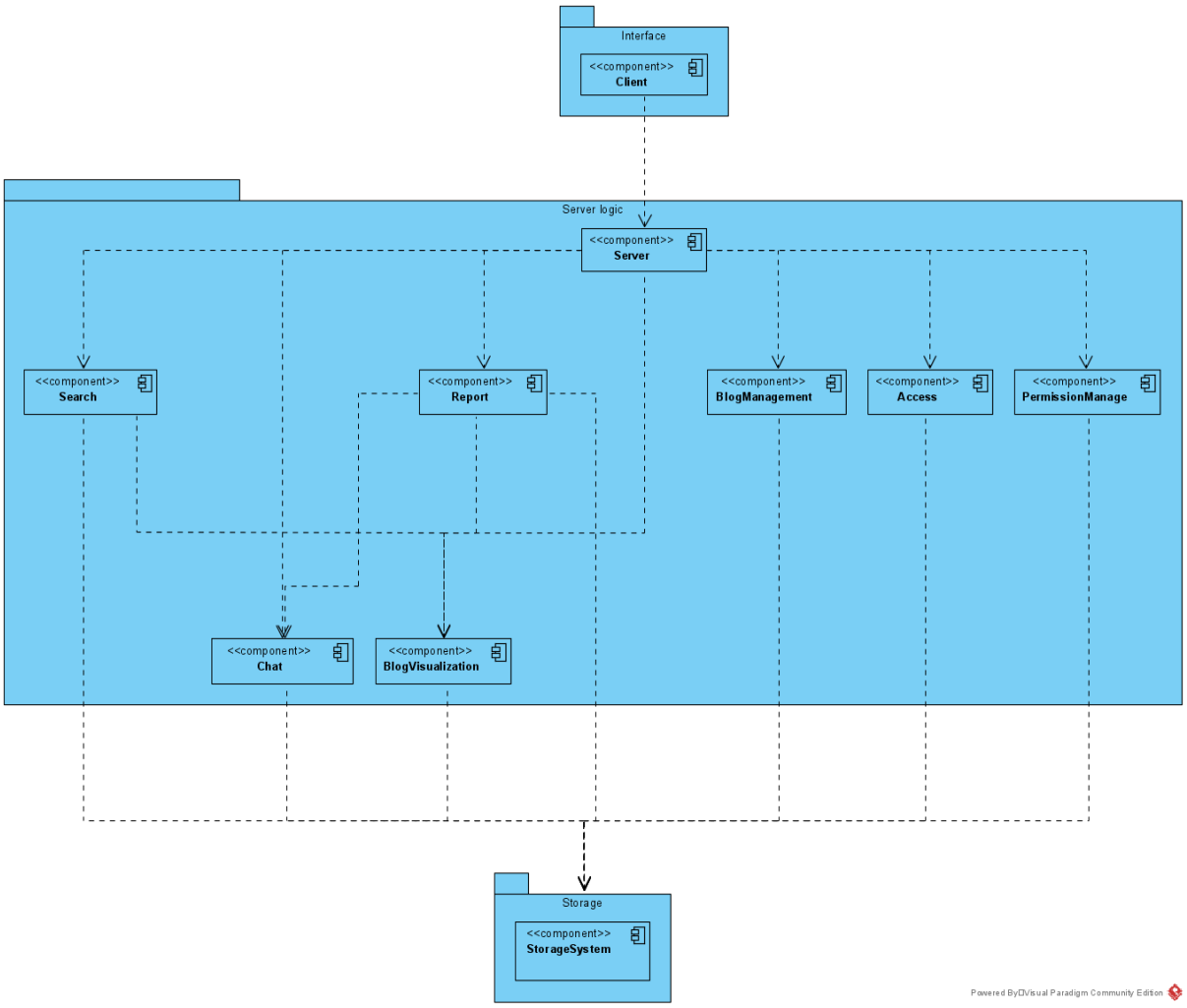
2.1. Panoramica

Per il sistema si è proposta un'**architettura a tre livelli**.



2.2. Decomposizione in sottosistemi:

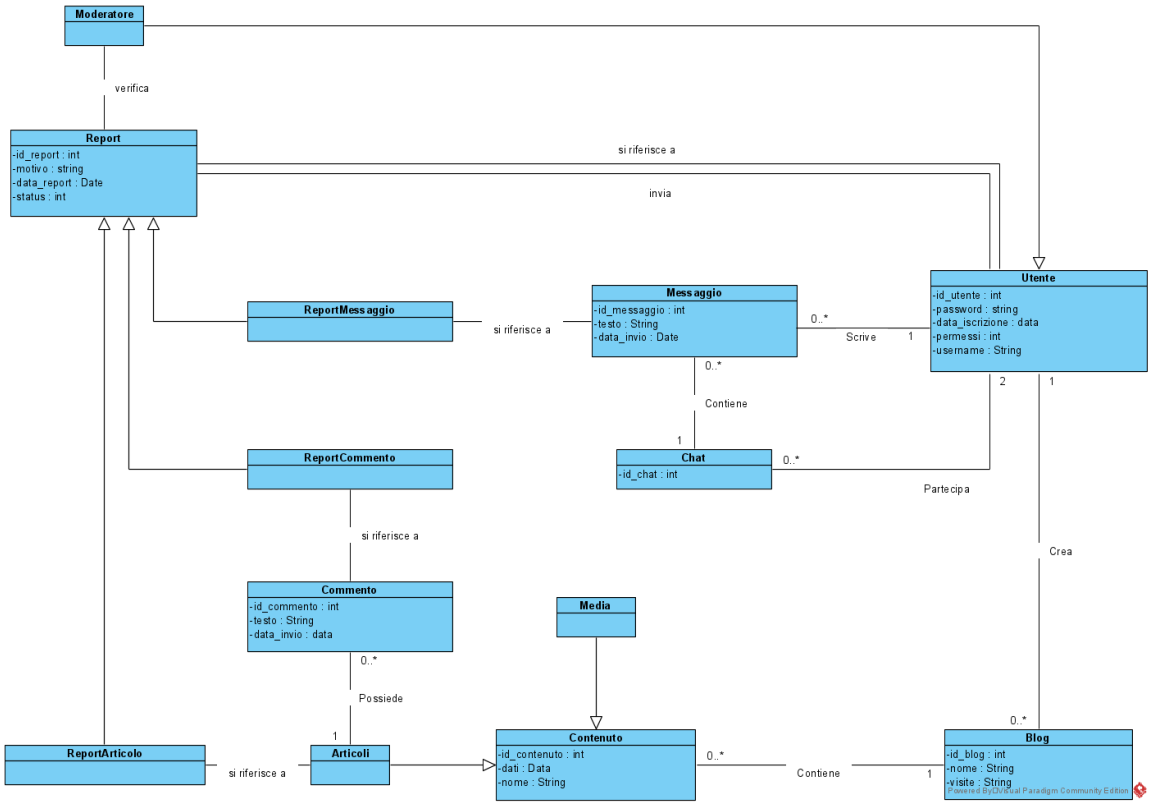
- Il sottosistema **Chat** è responsabile della creazione delle chat tra gli utenti, dell'invio dei messaggi e del recupero delle chat già avviate
- Il sottosistema **Server** gestisce il controllo degli accessi e della concorrenza
- Il sottosistema **Client** fornisce un frontend per l'utente che inizializza tutti i casi d'uso
- Il sottosistema **StorageSystem** è responsabile della memorizzazione di tutti gli oggetti persistenti
- Il sottosistema **PermissionManage** si occupa del controllo delle autorizzazioni
- Il sottosistema **Access** si occupa della registrazione e dell'autenticazione dell'utente
- Il sottosistema **BlogManagement** si occupa del recupero dei contenuti di un blog, del caricamento dei file relativi ad un blog, della creazione e cancellazione di contenuti o blog
- Il sottosistema **BlogVisualization** si occupa di visualizzare i contenuti dei blog con la corretta formattazione e della gestione dei commenti relativi agli articoli
- Il sottosistema **Report** gestisce le modalità di segnalazione, controllo e verifica report
- Il sottosistema **Search** è responsabile della ricerca di utenti e blog, della visualizzazione del profilo utente e del recupero dei migliori blog


Interface
<<component>>
Client
Server logic
<<component>>
Server
<<component>>
Search
<<component>>
Report
<<component>>
BlogManagement
<<component>>
Access
<<component>>
PermissionManage
<<component>>
Chat
<<component>>
BlogVisualization
Storage
<<component>>
StorageSystem
Powered By Visual Paradigm Community Edition

2.3. Gestione dei dati persistenti

Il sistema si connetterà ad un database gestito con MariaDB, dove saranno memorizzati i dati personali degli utenti, le chat, i report e le informazioni relative ai blog.

I contenuti caricati dagli utenti verranno memorizzati sul file system del database server poiché potrebbero essere di grandi dimensioni e la gestione di essi tramite database implicherebbe delle perdite in termine di performance



3. Mappatura Hardware/Software

Il sistema è basato su una architettura di tipo Client/Server.

Lato server, Apache Tomcat 9.0 ricopre il ruolo di Web Server, tramite Java Servlet viene gestita la logica applicativa, mentre le JSP forniranno l'interfaccia utente del sistema.

Lato client, il web browser dell'utente gestirà l'interazione con il sistema.

Il client e il server comunicano utilizzando il protocollo HTTP.

Il server comunica con i dati persistenti attraverso le API JDBC.

4. Controllo degli accessi e sicurezza

Gli utenti potranno accedere al loro spazio tramite login con username e password.

Le password verranno crittografate prima di essere memorizzate sul DB tramite l'utilizzo di salt e SHA2 a 512 bit.

Il sistema dovrà garantire che i file caricati dagli utenti non compromettano la sicurezza.

È necessario sanificare tutti i dati immessi in input dall'utente per evitare l'esecuzione di codice arbitrario e altre vulnerabilità.

Per documentare i diritti di accesso usiamo la matrice di controllo degli accessi.

Nella prima colonna vengono riportati gli oggetti, mentre nella prima riga le entità.

Oggetto/Entità	Utente	UtenteRegistrato	UtenteAutenticato	ModeratoreChat	ModeratoreBlog	GestoreUtenti
Utente	registrazionevisitaProfilo					managePermessi
Articolo	viewArticolo		createArticolomodifyArticolo			
Chat			createChatviewPrivateChat	viewReportedChat		
Messaggio			createMessaggio			
ReportMessaggio			reportMessaggio	manageReportMessaggio		
ReportCommento			reportCommento		manageReportCommento	
ReportArticolo			reportArticolo		manageReportArticolo	
Contenuto	viewContenuto		uploadFiledeleteContenuto			
Commento	viewCommenti		createCommento			
Blog	viewBlog		createBlogdeleteBlog			
Sistema		login				

È POSSIBILE VEDERE LA TABELLA CON UNA FORMATTAZIONE MIGLIORE IN MATRICEACCESSI