System Design Document

SmartBlog

Progetto di Ingegneria del Software 2020/2021

Felice De Chiara	0512105866
Vincenzo Emanuele Martone	0512105758
Antonio Russo	0512106058
Alfonso Graziano	0512105776

Sommario

1. Introduzione

1.1 Obiettivi del sistema

Il software che andiamo a proporre mira a migliorare le scelte degli utenti e a renderli utenti *consapevoli*. La nostra proposta si basa sull'idea che valutare lo smartphone adatto a sé debba essere semplice. Diamo anche molta importanza a ciò che pensa la community, ecco perché ogni utente è libero di lasciare una recensione su uno smartphone.

1.2 Design Goals

Id	Criteria	Design Criterion	Priority	Origin
DG1	Performance	**Response time**: Si prevede un tempo di risposta non superiore a 5s.	Media	NF3.2
DG2	Performance	**Throughput:** Il sistema deve essere in grado di supportare 500 utenti contemporaneamente	Media	NF3.1
DG3	Dependability	**Robustness**: Il sistema deve essere in grado di sopravvivere ad input errati dell'utente tramite precisi controlli e filtraggi sui campi immessi	Alta	Application Domain
DG4	Dependability	**Availability**: Il sistema garantisce un uptime che si aggira attorno al 98%.	Media	NF2.1
DG5	Dependability	**Security**: Il sistema garantisce sicurezza relativa ai dati persistenti tramite filtraggio di interrogazioni alla base di dati. Inoltre il sistema deve gestire la sicurezza in modo	Alta	NF2.3 NF2.2

		programmatico per evitare che utenti non autorizzati accedano a risorse riservate.		
DG6	Cost	**Development cost:** Il sistema deve essere sviluppato entro le deadlines definite con il committente	Bassa	Application Domain
DG7	Maintenance	**Extensibility:** È importante che il sistema sia estendibile per prevedere l'aggiunta di successive funzionalità	Media	Application Domain
DG8	Maintenance	**Readability:** La leggibilità del codice sarà garantita dalla suddivisione del sistema nei relativi sottosistemi tramite i packages.	Alta	NF4.3
DG9	Maintenance	**Portability:** La portabilità del sistema è garantita dall'utilizzo di Java. È, dunque, possibile spostarlo da una macchina all'altra, a patto che su di essa sia installata la JDK.	Alta	NF4.4
DG10	End User	**Utility:** Il sistema supporta l'utente tramite l'indicazione precisa dei dati scorretti durante l'immissione degli stessi	Media	Application Domain
DG11	End User	**Usability:** Il sistema è progettato per essere facilmente fruibile dall'utente grazie a diverse scelte riguardanti l'interfaccia grafica, in particolare la navigazione della piattaforma sarà supportata da una gerarchia delle pagine principali visitate, posta in alto. La semantica delle operazioni sarà supportata dalla palette di colori usata e in particolare le	Alta	NF1.1 NF1.2 NF1.3 NF1.4

		azioni che indicano conferma saranno di colore azzurro, quelle che indicano cancellazione saranno di colore grigio. Quando un utente si trova ad esprimere il proprio gradimento nei confronti di una caratteristica durante una recensione, essa sarà espressa da stelline colorate che rendono immediata la comprensione da parte dell'utente. Infine l'utente meno esperto che vuole fruire del tool di Intelligenza Artificiale, non sarà costretto a ricorrere a tecnicismi per descrivere lo Smartphone dei suoi sogni, ma potrà limitarsi a termini a lui familiari (*Display*, *Fotocamera*, *Prestazioni* e *Batteria*).		

1.2.1 Trade-offs

Trade-off	Rational
Development cost vs Functionality	Per rientrare nei tempi di rilascio previsti, sarà sviluppato un sottoinsieme principale di funzionalità per fornire all'utente una demo funzionante del sistema
Development cost vs Quality	Per curare nel dettaglio determinate funzionalità che verranno realizzate, i costi di sviluppo potrebbero variare circa del 14%
Development cost vs Extensibility	Per garantire l'estendibilità del sistema sarà necessario un costo di sviluppo maggiore, il quale

	verrà tuttavia ripagato grazie alla semplicità di eventuali estensioni future
Functionality vs Usability	Per garantire la Usability del sistema anche agli utenti meno esperti, sono state previste funzionalità ridotte
Usability vs Development cost	Per mantenere bassi i costi e il tempo di sviluppo si è deciso di non rendere responsive l'interfaccia grafica

1.3 Definizioni, acronimi ed abbreviazioni

- Piattaforma → Applicazione web
- CRUD → Create, Read, Update, Delete
- DG → Design Goal
- DB → Database

1.4 Riferimenti

Nel corso del documento verranno fatti riferimenti a numerosi aspetti trattati nel RAD. In particolare i *Requisiti non funzionali* e il *Class Diagram.*

1.5 Panoramica

Nel documento:

1. Affronteremo l'analisi di sistemi simili.
2. Valuteremo la scomposizione del sistema in sottosistemi.
3. Definiremo la strategia di deploy e le condizioni limite.
4. Verranno esplicitati tutti i servizi di ciascun sottosistema.

2. Architettura di sistemi simili

Un sistema simile a quello proposto potrebbe essere ad esempio il sito web hdblog.it.

Il sistema HDBlog sfrutta server Microsoft, il webserver è IIS, utilizza tool di Analytics come Google Analytics.

Non ci è dato sapere come gestiscono i dati persistenti. Possiamo ipotizzare che i dati siano salvati su un Database gestito da un DBMS.

3. Architettura del sistema proposto

3.1 Panoramica

SmartBlog è un sistema *distribuito*.

Le componenti principali sono tre:

1. Un *database relazionale* per il salvataggio dei dati persistenti
2. Un *framework front-end* (React) per la realizzazione delle interfacce utente
3. Servizi *Java* per la realizzazione del back-end

3.2 Scomposizione in sottosistemi

Abbiamo deciso di realizzare *SmartBlog* seguendo un'architettura *three-tier* al fine di minimizzare l'accoppiamento tra i vari componenti.

Ecco la suddivisione dei sottosistemi:

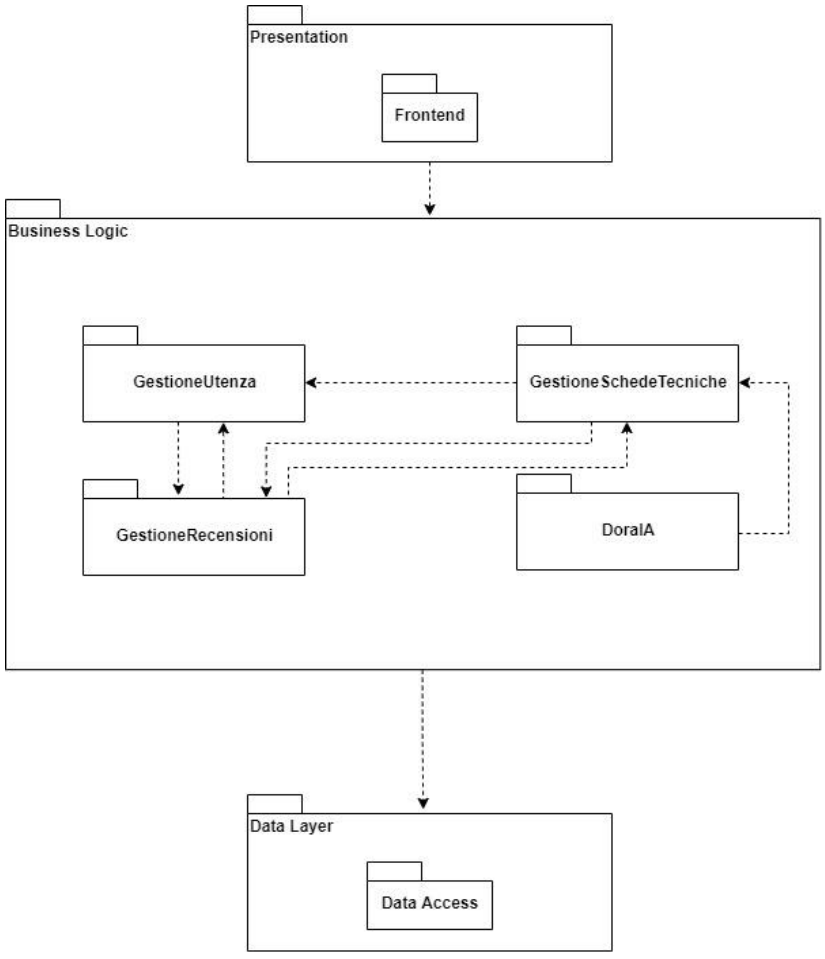
1. *Presentation* contiene il front-end. Questa è l'interfaccia utente che gli utenti useranno per interagire con il sistema.
2. *Business Logic*. All'interno troviamo tutta la logica di business con la gestione degli utenti e relativa gestione delle autorizzazioni, la

gestione delle schede tecniche, la gestione delle recensioni e DoraIA, il tool di Intelligenza Artificiale.

I sottosistemi individuati sono i seguenti:

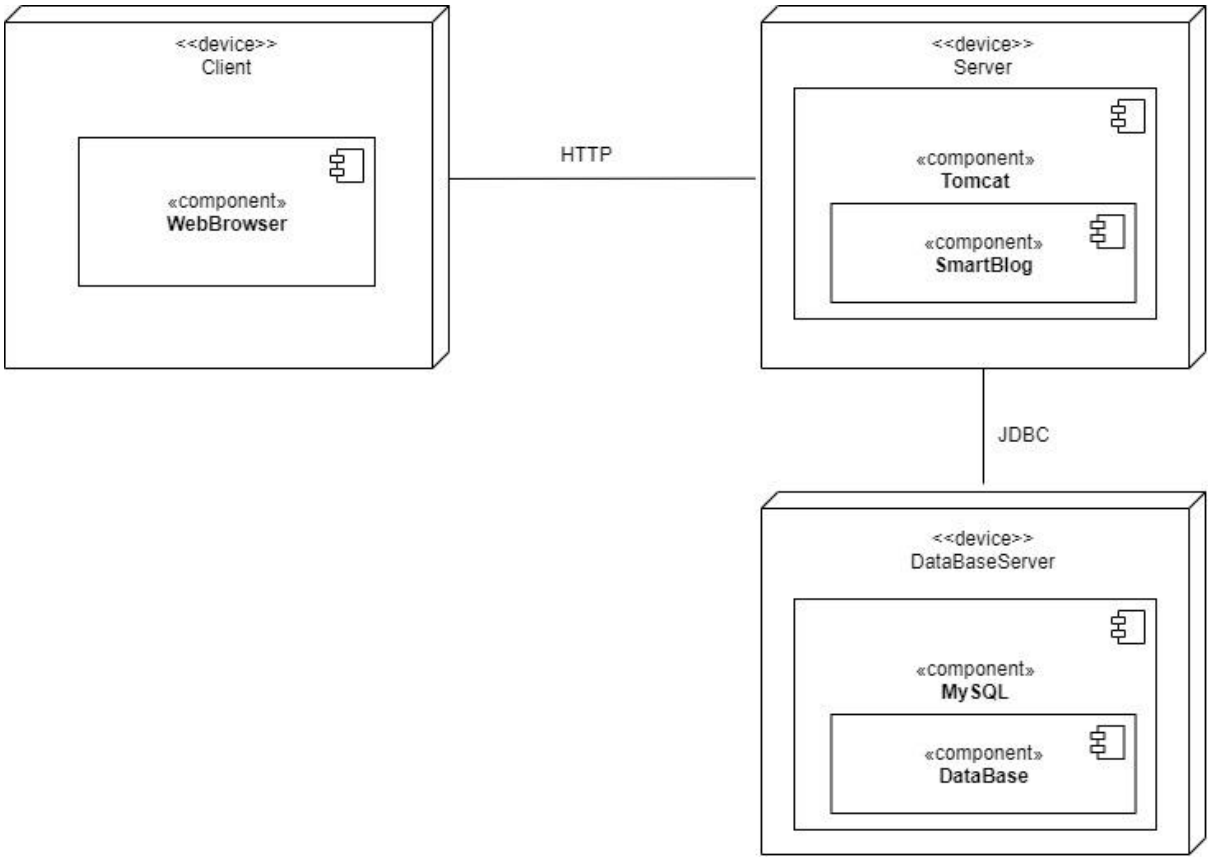
a. *Gestione Utenza*: si occupa della creazione, modifica ed eliminazione degli account degli utenti.
b. *Gestione Recensioni*: si occupa della creazione, approvazione/rifiuto e visualizzazione delle recensioni effettuate dagli User
c. *Gestione Schede Tecniche*: si occupa della creazione, modifica, eliminazione e visualizzazione delle schede tecniche
d. *DoraIA*: si occupa di fornire agli User un tool di Intelligenza Artificiale che gli consiglia una lista di Smartphone in base ai parametri da essi inseriti

3. Data layer contiene solo il sottosistema Data Access che accede direttamente al DB per effettuare le operazioni sui dati persistenti



3.3 Hardware/software mapping

Il sistema che si va a realizzare è distribuito e consente l'accesso da parte di un *Web Browser* installato su un dispositivo client. Quest'ultimo si collegherà al server, tramite HTTP, su cui è presente il *WebServer Tomcat* sul quale è *deployata* la nostra piattaforma. I dati persistenti vengono mantenuti all'interno di una base di dati posta su una terza macchina (un *DataBase Server*) sulla quale è installato *MySQL* che gestisce la base di dati del sistema.



3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti si è deciso di utilizzare MySQL per le seguenti ragioni:

- E' un database relazionale con una vastissima community
- Contiene un'ampia documentazione
- E' molto scalabile
- E' pratico da utilizzare

Grazie a queste caratteristiche ci consente di gestire i dati persistenti con semplicità.

Usando come riferimento il *Class Diagram* sono state identificate i seguenti dati che dovranno essere resi persistenti:

- User
- Manager
- Reviewer
- Spec
- Reviews
- SpecScore

Non è prevista la gestione di dati persistenti tramite dei file.

3.5 Controllo degli accessi e sicurezza

La sicurezza dei dati è garantita dall'impossibilità per ogni visitatore del sito di creare,modificare o cancellare gli oggetti che modellano le entità. Tale compito viene assegnato solo ad alcuni tipi di account, i *Manager* e i *Reviewer*. Questi ultimi possono accedere a tali funzioni solo attraverso il proprio username e la propria password, che verrà cifrata. È previsto per tutti gli utenti il recupero della password.

	Gestione Utenza	Gestione Schede Tecniche	Gestione recensioni	DoraIA
User	createUser auth signOut getUserInfoByEmail updateUserInfo deleteAccount recoverPassword	searchByName searchAll searchById	createReview searchReviewsByUser	findSpecsByParams
Reviewer	auth	setScores	searchPendingR	

	signOut getUserInfoByEmail updateUserInfo deleteAccount recoverPassword		eviews searchReviewInfo approvation	
Manager	auth signOut getUserInfoByEmail updateUserInfo deleteAccount recoverPassword	createSpec getSpecsFromFile editSpec deleteSpec		

3.6 Controllo flusso globale del sistema

Il flusso del sistema è gestito tramite invocazioni da parte di più *Client* verso il *Server* che smista le richieste verso una *Servlet* specifica, la quale restituisce i dati richiesti al *Client*. Tali chiamate sono gestite in modo concorrente, dunque ogni richiesta del client avrà un Thread dedicato, grazie al *Web Container Tomcat* che gestisce tale concorrenza.

3.7 Boundary condition

La seguente sezione del documento illustra le principali *Boundary condition* del sistema. In particolare si affronterà la questione della *configurazione iniziale* in cui vengono creati gli account di Manager e Reviewer all'interno della base di dati, dello *startup*, dello *shutdown* e del comportamento in caso di eventuali *failure*. In tali procedure verranno coinvolti due attori particolari:

- **WebServerAdmin**: è l'*Amministratore* del *Web Server* che si occupa dell'avvio e dello spegnimento del *Web Server*
- **DatabaseAdmin**: è l'*Amministratore* del *Database* che si occupa dell'avvio e dello spegnimento del *Database Server* e dell'inserimento, in fase di prima configurazione, degli account *Manager* e *Reviewer.*

3.7.1 Startup sistema

Use Case Name	Web Server Startup
Participating actors	Iniziato da WebServerAdmin
Flow of events	1. WebServerAdmin apre l'interfaccia di configurazione di "Apache Tomcat" 2. Clicca sul pulsante "Start" 3. Attende che il Server completi la procedura di avvio 4. Al termine della procedura, se tutto è andato a buon fine, la voce "Service Status" riporterà "Started"
Entry Condition	Admin apre l'interfaccia di configurazione di "Apache Tomcat"
Exit conditions	Admin ha avviato correttamente il Web Server

Use Case Name	Database Server Startup
Participating actors	Iniziato da DatabaseAdmin
Flow of events	1. DatabaseAdmin avvia il tool “MySQL Notifier” 2. Clicca sul pulsante relativo al tool e visualizza i DatabaseServer attivi sulla sua macchina 3. Sposta il cursore sul DatabaseServer da avviare e dal menù a tendina che compare clicca su “Start” 4. Al termine della procedura, se tutto è andato a buon fine, lo status del server sarà “Running”
Entry Condition	DatabaseAdmin avvia il tool MySQL Notifier
Exit conditions	DatabaseAdmin ha avviato correttamente il Database Server

3.7.2 Shutdown Sistema

Use Case Name	Web Server Shutdown
Participating actors	Iniziato da WebServerAdmin
Flow of events	1. WebServerAdmin apre l’interfaccia di configurazione di “Apache Tomcat” 2. Clicca sul pulsante “Stop” 3. Attende che il Server completi la procedura di arresto 4. Al termine della procedura, se tutto è andato a buon fine, la voce “Service Status” riporterà “Stopped”
Entry Condition	Admin apre l’interfaccia di configurazione di “Apache Tomcat”
Exit conditions	Admin ha stoppato correttamente il Web Server

Use Case Name	Database Server Shutdown
Participating actors	Iniziato da DatabaseAdmin
Flow of events	1. DatabaseAdmin clicca sul pulsante relativo al tool “MySQLNotifier” e visualizza i DatabaseServer attivi sulla sua macchina 2. Sposta il cursore sul DatabaseServer da stoppare e dal menù a tendina che compare clicca su “Stop” 3. Al termine della procedura, se tutto è andato a buon fine, lo status del server sarà “Stopped”
Entry Condition	DatabaseAdmin clicca sul pulsante relativo al tool “MySQLNotifier”
Exit conditions	DatabaseAdmin ha stoppato correttamente il Database Server

3.7.3 Prima configurazione

Use Case Name	Creazione account “Manager”
Participating actors	Iniziato da DatabaseAdmin
Flow of events	1. DatabaseAdmin accede all’interfaccia “MySQL Command Line Client” 2. Sceglie il database da usare tramite il comando “USE” 3. Scrive la query per l’inserimento dell’account specificando e-mail, username, password per la crezione dell’User e numero di telefono per la successiva creazione del relativo Manager. 4. Visualizza l’esito della query
Entry Condition	DatabaseAdmin apre l’interfaccia “MySQL Command Line Client”

Exit conditions	DatabaseAdmin ha creato correttamente l'utente Manager

Use Case Name	Creazione account "Reviewer"
Participating actors	Iniziato da DatabaseAdmin
Flow of events	1. DatabaseAdmin accede all'interfaccia "MySQL Command Line Client" 2. Sceglie il database da usare tramite il comando "USE" 3. Scrive la query per l'inserimento dell'account specificando e-mail, username, password per la crezione dell'User, numero di telefono e rank per la successiva creazione del relativo Reviewer. 4. Visualizza l'esito della query
Entry Condition	DatabaseAdmin apre l'interfaccia "MySQL Command Line Client"
Exit conditions	DatabaseAdmin ha creato correttamente l'utente Reviewer

3.7.4 Fallimento

Il sistema può incorrere in diverse *failure* che verranno gestite in vari modi. Se durante l'esecuzione del sistema, il *Database Server* dovesse avere dei malfunzionamenti, il sistema avviserà l'utente dell'errore chiedendogli di provare a ripetere l'operazione in seguito. In particolare, se il malfunzionamento del *Database Server* dovesse verificarsi durante un'interazione con la base di dati stessa, l'utente verrà avvisato dell'errore, ma verranno evitate situazioni inconsistenti in quanto ogni accesso in scrittura alla base di dati verrà gestito con una transazione. Non sono previsti supporti per la ridondanza dei Server, dunque un malfunzionamento al *Web Server* comporterà la completa incapacità da parte dell'utente di accedere ai diversi servizi offerti.

4. Subsystem services

4.1 Gestione Utenti

Servizio	**Descrizione**
createUser	Permette a User di registrarsi alla piattaforma tramite l’inserimento di email e password
auth	Permette agli User della piattaforma dotati di credenziali di effettuare il Login
signOut	Permette ad un User autenticato alla piattaforma di effettuare il LogOut terminando la propria sessione
getUserInfoByEmail	Permette ad un User autenticato di visualizzare le proprie informazioni riguardanti il proprio profilo, e le recensioni scritte da quest'ultimo.
updateUserInfo	Permette ad un User autenticato di modificare le informazioni del proprio account (quali e-mail, username e password)
deleteAccount	Permette ad un User autenticato di cancellare il proprio account
recoverPassword	Permette ad un User, che non ricorda la propria password, di recuperarla tramite l’invio di un'email per impostarne una nuova

4.2 Gestione Schede Tecniche

Servizio	Descrizione
searchByName	Permette a User di cercare una scheda tecnica specificando una keyword relativa al nome
searchById	Permette a un User di visualizzare tutte le informazioni riguardante una spec
createSpec	Permette ad un Manager di inserire i dati delle nelle Spec, con lo scopo di creare una singola Spec
getSpecsFromFile	Permette ad un Manager di ottenere una lista di Spec dato un file formattato
editSpec	Permette ad un Manager di modificare le informazioni relative ad una Spec
deleteSpec	Permette ad un Manager di eliminare una Spec
setScores	Permette ad un Reviewer di inserire i punteggi relativi ad una Spec
editSpec	Permette ad un Reviewer di modificare i punteggi relativi ad un Spec

4.3 Gestione recensioni

Servizio	Descrizione
createReview	Permette a un User di aggiungere una Review ad una Spec
searchPendingReviews	Permette ad un Reviewer di cercare tutte le recensioni in stato pending
searchReviewInfo	Permette a un Reviewer di leggere una Review di un User
approvation	Permette a un Reviewer di approvare o rifiutare una Review
searchReviewsByUser	Permette ad un User di ottenere tutte le Review che ha scritto

4.4 DoraIA

Servizio	Descrizione
findSpecsByParams	Consente a User di ottenere una lista di Specs in base al costo massimo scelto da User e al grado di importanza che dà alle diverse caratteristiche dello Smartphone