UNIVERSITÀ DEGLI STUDI
DI SALERNO

Ingegneria del Software

A.A. 2019/2020

T

SDD

Versione 1.1

Top Manager:

Professore
Andrea De Lucia

Partecipanti:

Nome	Matricola
Vittorio Ventura	05121 5766

1. Introduction

1.1 Purpose of the system

il sistema che si vuole realizzare ha come scopo quello di facilitare la vita a studenti che affrontano per la prima volta (e non) l'università
In questo modo offriamo quindi una piattaforma comune sia a studenti che a chi concede i servizi offerti.

1.2 Design goals

Criteri di performance:

- Tempi di risposta: il sistema deve rispondere alle operazioni dell'utente in massimo 3 secondi anche se il sistema è sotto stress (picco di carico elevato). Il sistema deve essere capace di visualizzare le pagine in massimo 2 secondi nonostante le condizioni di connessione dell'utente per garantire una miglior esperienza di navigazione
- Memoria: la memoria dipende dalla quantità di richieste che il sistema riceve e dalla memoria utilizzata per mantenere le informazioni memorizzate nel DB (ovviamente quelle necessarie al funzionamento). Si cercherà di ottimizzare lo spazio occupato in memoria con ottimizzazione sul DB

Criteri di Affidabilità:

- Robustezza: il sistema informerà l'utente, attraverso l'utilizzo di appositi messaggi d'errore, in presenza di input che non rispettano i formati definite nelle tabelle presenti all'interno del RAD
- Affidabilità: il sistema deve garantire l'affidabilità dei servizi offerti; la piattaforma sarà sviluppata in modo da controllare tutti gli input inseriti sia da utenti che autori, e i processi di registrazione e autenticazione saranno gestiti in modo affidabile.
- Disponibilità: il sistema sarà disponibile all'utente secondo le sue esigenze ad eccezione dei periodi di manutenzione che verranno comunicati.
- Sicurezza: il sistema fornisce le principali tecniche per poter resistere ad attacchi come ad esempio l' SQLinjection

Criteri di costo:

- Costo di sviluppo: viene stimato un costo complessivo di 200 ore per lo sviluppo del sistema divise tra i vari componenti del team.
- Costo di distribuzione: viene stimato un costo di pochi minuti per le formazioni degli utenti per poter utilizzare la nuova piattaforma
- Costo di manutenzione: viene richiesto un costo di 24 ore per la correzione di problemi sia software che hardware

Criteri di manutenzione:

- Estensibilità: è possibile aggiungere nuove funzionalità in base alle esigenze dell'utente o delle nuove tecnologie
- Portabilità: l'utilizzo del sistema avviene mediante l'utilizzo di un browser quindi il sistema può funzionare su qualsiasi dispositivo avente un browser web. Inoltre, l'utilizzo di funzioni di base supportate all'interno di qualsiasi browser web facilità ancor di più la portabilità

Criteri di Usabilità:

Usabilità: l'utilizzo del sistema risulta molto semplice visto l'utilizzo di un'interfaccia user-friendly e mediante apposite sezioni di supporto presenti nell'interfaccia, da suggerimenti e campi d'esempio.

1.3 Definition,acronyms and abbreviations

DB: Database

RAD: Requirements Analisys document

DBMS: Database Management system

JDBC: Java Database Connectivity

SDD: system design document

1.4 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

1.5 Overview

Il seguente documento SDD comprende le seguenti sezioni:

1. Introduzione: in questa sezione sarà presentata una breve descrizione degli obiettivi del sistema e vengono forniti i design goal. Successivamente vengono presentate le definizioni, acronimi e abbreviazioni per facilitare al lettore la comprensione di alcuni termini.
2. Architettura del sistema corrente: il suo scopo è fornire la descrizione dell'architettura del sistema, se esiste, da sostituire.
3. Architettura del sistema proposto: lo scopo di questa sezione è quello di documentare il modello di progettazione del nuovo sistema. E' diviso in sette sezioni: overview, subsystem decomposition, Hardaware/software mapping , persistent data management, access control and securety, global software control, boundary condiction.
4. Servizi di sottosistemi: questa sezione descrive i servizi forniti da ciascun sottosistema.

2. Current software Architecture

Attualmente non esiste una piattaforma dove gli utenti possano reperire informazioni che sono verificate, agevolandolo nella ricerca e nella comprensione di esse. Infatti, la ricerca di informazioni prevede i seguenti passi: l'utente effettua una ricerca all'interno di un browser web dove vengono prodotti come risultato di ricerca informazioni che in alcuni casi possono essere errati. Inoltri nella maggior parte dei casi le informazioni ricercate sono difficili da individuare poiché molti articoli contenenti le informazioni desiderate sono scritti da autori sconosciuti..

3. Proposed software architecture

3.1 Overview

Il sistema proposto è un application-web che segue il pattern MVC separando la parte di logica da quella dei dati e dalla logica di business. Si tratta di un'architettura multi-tier dove le varie funzionalità sono separate su più layer. Tutti gli utenti potranno autenticarsi e registrarsi alla piattaforma.

Subsystem decomposition

3.1.1 Decomposizione in Layer

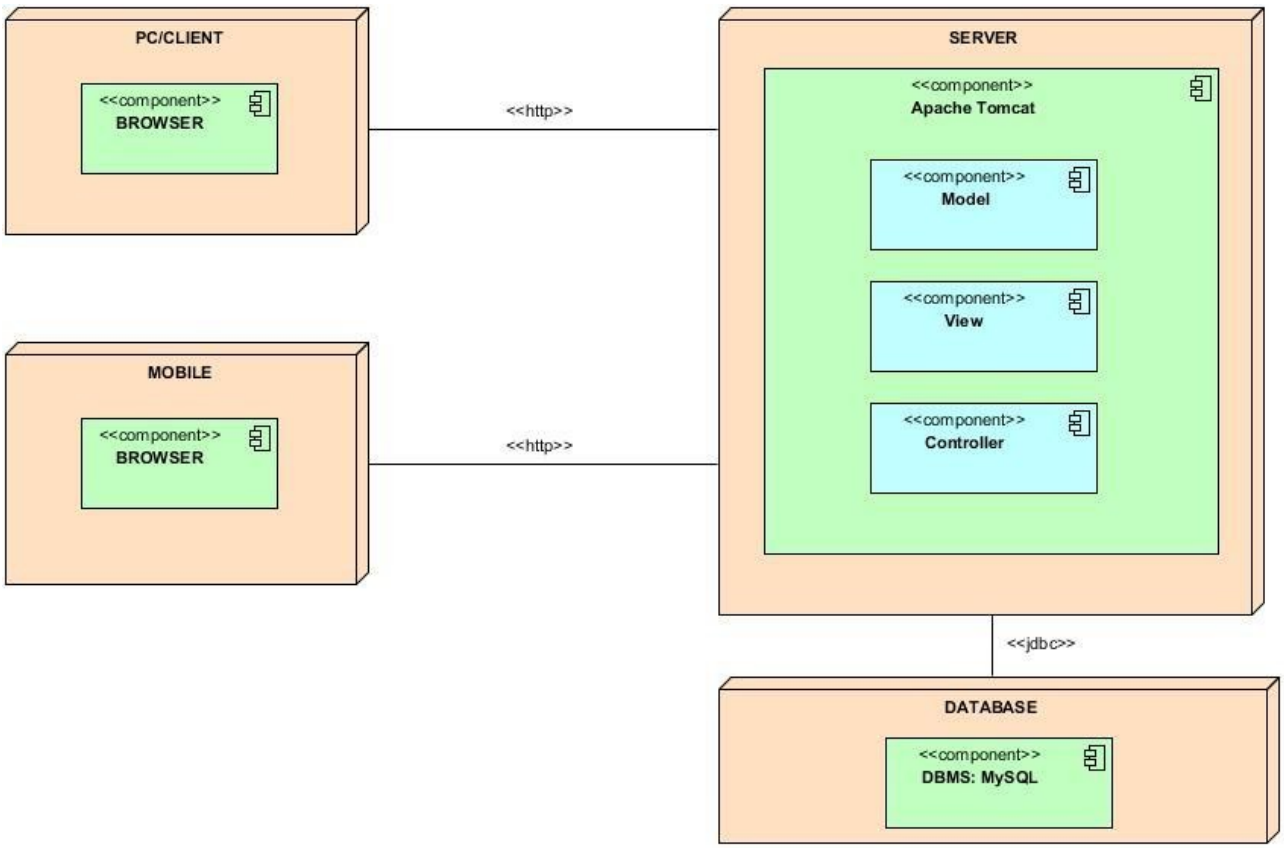
La decomposizione prevista per il sistema è quella basata su tre livelli. Quindi abbiamo tre livelli che si occupano di gestire diverse funzionalità del sistema:

- Model: che si occupa della gestione e dell'accesso ai dati
- Control: gestisce la sequenza di interazioni con l'utente
- View: si occupa della parte di visualizzazione delle informazioni della piattaforma

  Il livello View prevedere la gestione di quattro sottosistemi
  - UtenteView
  - AdminView

3.2 Hardware/software mapping

Si è scelto di utilizzare una struttura hardware costituita da un server (unico) che risponderà ai servizi richiesti dal client. Il client è un qualsiasi dispositivo dotato di un web browser, che supporta le funzionalità di base di JavaScript, e di una connessione ad Internet che verranno utilizzati per connettersi alla piattaforma. Per il server si è deciso di utilizzare una macchina che include un web server in particolare Apache Tomcat e un DBMS in questo caso MySql per gestire sia la logica che la persistenza dei dati. Il client e il server saranno connessi tramite una rete che utilizzerà il protocollo TCP/IP attraverso cui verranno inoltrate le richieste di un servizio. Il web server comunicherà con il DBMS tramite JDBC.



3.3 Persistent data management

3.3.1 Selecting a storage management stategy

Per memorizzare i dati si è scelto un database relazione che consente la gestione della concorrenza, il controllo degli accessi e il ripristino degli accessi anomali. Inoltre vista la quantità di dati che continuerà ad aumentare visto lo scopo della piattaforma un database relazionale risulta essere scalabile e quindi ideale per grandi quantità di dati. Di conseguenza questo ci permetterà di avere accessi efficienti, tempi di risposta brevi e un'elevata quantità di spazio di archiviazione.

Ho optato per l'utilizzo di database MySQL. A differenza dei suoi competitors, MySQL offre servizi altamente ottimizzati per l'utilizzo in grandi sistemi, con uno sguardo sempre attento alle ottimizzazioni circa l'indicizzazione e la richiesta dei dati. Nello specifico, abbiamo trovato adatto al nostro sistema la possibilità di utilizzare:

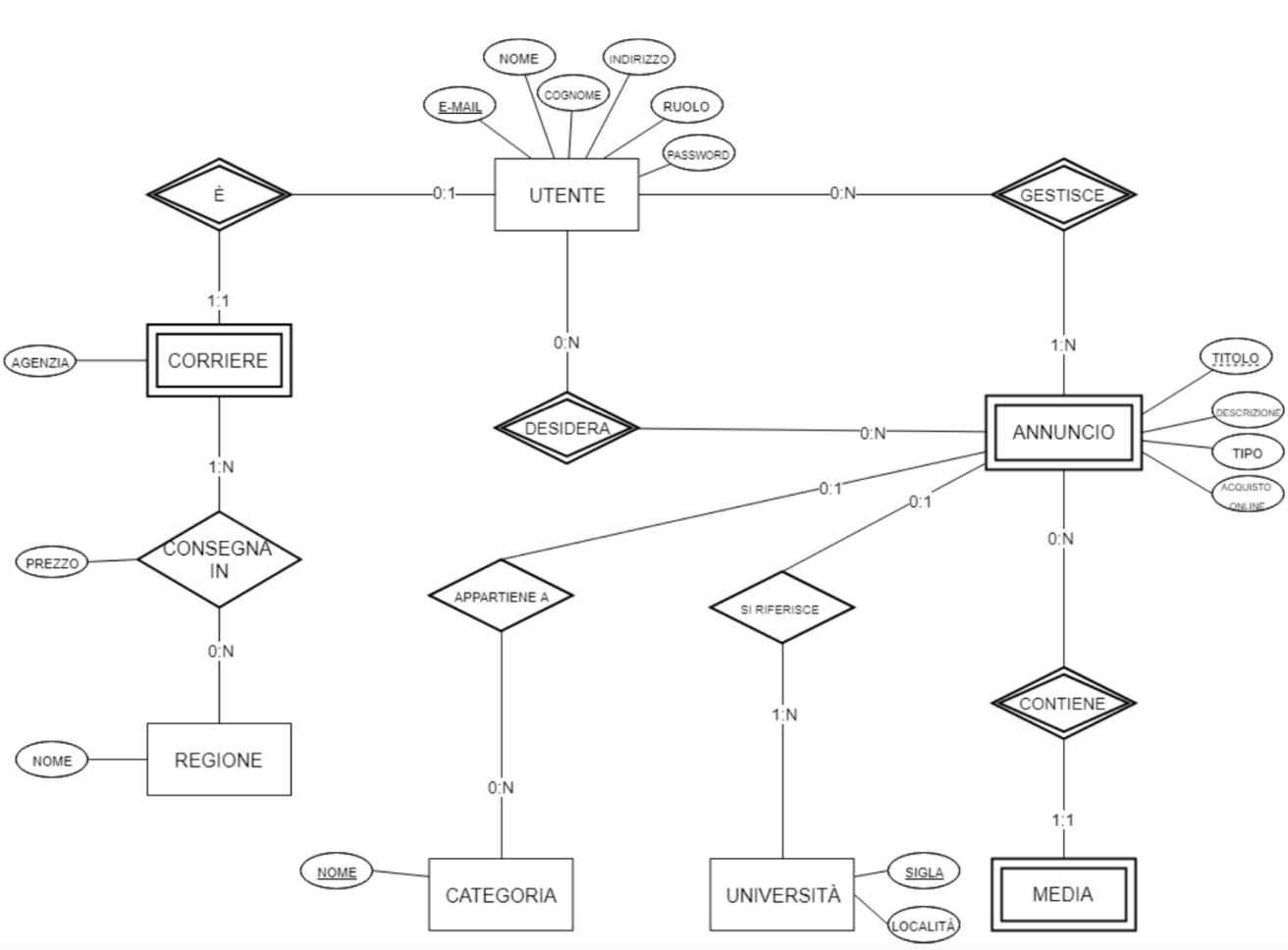
• Un'interfaccia grafica completa (detta Workbench) per modificare il database on-thefly;

• Un potente sistema di indicizzazione dei dati che permette di scrivere e prelevare dal database in maniera rapida ed efficiente;

• Una libreria (JDBC-Connector) che ha il pieno supporto del linguaggio che utilizzeremo lato server (Java);

• Un sistema per la gestione dei vincoli referenziali chiara ed efficiente.

3.3.2 Base di Dati (modello EER)



3.4 Global software control

Il flusso del sistema UniAds fornisce delle funzionalità che richiedono continua interazione da parte degli utenti, per cui il controllo del flusso globale del sistema è di tipo event-driven, dunque guidato dagli eventi. Il sistema è caratterizzato da un sito web accessibile da un browser e da un webServer, attivo 24h, che deve provvedere a gestire gli accessi concorrenti da parte di utenti. Quando un utente fa il log in 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 viene gestito da Java Servlets che, interagendo con il client , svolgono varie operazioni. Il server smista ogni nuova richista alla servlet adeguata, inoltrando poi la risposta al client sotto forma di JSP.