SDD

System Design Document

SOMMARIO

1. Introduzione

1.1 Obiettivo del sistema

DiscoverEasy nasce da una necessità pratica: facilitare la compravendita di prodotti.

L'obbiettivo è realizzare un sistema che permetta una semplice visualizzazione degli annunci e una intuitività delle funzionalità connesse al sistema. Sarà possibile ricerca un annuncio per vari parametri come: nome, regione e categoria. Ci sarà una pagina dedicata agli inserzionisti per tenere sotto controllo i loro annunci, inserirli od eliminarli. Il sistema sarà integrato con una chat per facilitare la compravendita e per una miglior chiarezza degli annunci. Il gestore del sito avrà le funzionalità di bannare un qualsiasi annuncio o utente.

1.2 Design Goal

Rank/Priorità	ID Design Goal	Descrizione design goal	Categoria	Origine	Motivazione
Alta	DG_01: Tempi di risposta	Il sito web deve garantire un tempo di risposta massimo di 6 secondi, nel caso l'utente sia connesso mediante una rete con velocità di almeno 1Mb/s in download.	Performance	RNF_1; RNF_3;	Tempi di risposta brevi sono fondamentali per permettere agli utenti una buona esperienza d'uso del sistema.
Alta	DG_02: Robustezza	Eventuali input non validi	Affidabilità	RNF_2;	È importante che il sito web sia

		immessi dall'utente dovranno essere opportunamente riconosciuti e segnalati attraverso messaggi di errore.			robusto, per evitare danni e attacchi al sistema e per evitare problemi durante l'utilizzo da parte degli utenti.
Alta	DG_03: Disponibilità	Una volta realizzato il sistema, esso dovrà essere disponibile ogni qualvolta si voglia, escludendo periodi di manutenzione brevi opportunamente segnalati.	Affidabilità	RNF_2;	È importante che il sistema sia sempre disponibile, poiché ciò comporta una maggiore possibilità di utilizzo, obiettivo fondamentale del sistema.
Alta	DG_04: Tolleranza ai guasti	Anticipando l'eventuale presenza di guasti, la tolleranza ai guasti sarà garantita attraverso opportuno backup dei dati	Affidabilità	RNF_2;	La tolleranza ai malfunzionamenti è importante per garantire, quanto più possibile, sia la disponibilità del sistema, sia la consistenza e la persistenza dei dati.

		persistenti ogni 15 giorni.			
Alta	DG_5: Portabilità	La portabilità è garantita dal momento che l'interazione sarà effettuata utilizzando il browser, che fornirà l'accesso alle funzionalità del sistema sottostante, garantendo l'indipendenza dal sistema operativo.	Manutenzione	RNF_1; RNF_5;	È importante che il sistema sia portabile poiché si vuole che sia utilizzato dal maggior numero di utenti.
Alta	DG_6: Usabilità	Il sito deve avere un' altra efficienza d'uso, presentando obiettivi facili da raggiungere anche per intuitività.	End-User	RNF_1;	Una buona usabilità della piattaforma è necessaria per permettere agli utenti di fruire al meglio della stessa.
Alta	DG_7: Utilità	Utilizzando il sito l'utente potrà ridurre i tempi di ricerca del prodotto da lui desiderato.	End-User	N/A	N/A

* ogni design goal appartiene ad una delle seguenti macrocategorie: Permormance, usabilità, affidabilità, manutenibilità, implementazione, packaging e legali.

** ID del requisito non funzionale che ha condotto alla definizione del rispettivo design goal.

*** motivazioni che hanno spinto all'introduzione del design goal.

1.2.1 Trade offs

La seguente sezione descrive eventuali conflitti che sorgono tra due o più design goals. In questo documento si tratteranno soltanto i più significativi.

Memoria vs Efficienza:

I design goals DG_01, DG_02 e DG_08, rispettivamente Tempi di risposta, Memoria e Costi di distribuzione, sono ovviamente in conflitto tra di loro: una memoria più ampia, se ben utilizzata, favorisce tempi di risposta più brevi, a fronte di un costo di implementazione più alto.
Nell'implementazione di questo progetto, tenendo conto sia del poco spazio richiesto per la memorizzazione delle singole informazioni, sia della numerosità delle stesse, si preferirà fare uso di qualche ridondanza, ovvero di qualche spreco di memoria, per puntare a massimizzare la velocità nella ricerca delle informazioni.

1.3 Acronimi

- **DG:** Design Goal
- **RF:** Requisito Funzionale.
- **RNF:** Requisito Non Funzionale.
- **RNF_1:** Requisito Non Funzionale di usabilità.
- **RNF_2:** Requisito Non Funzionale di affidabilità.
- **RNF_3:** Requisito Non Funzionale di performance.
- **RNF_4:** Requisito Non Funzionale di manutenibilità.
- **RNF_5:** Requisito Non Funzionale di implementazione.
- **RNF_6:** Requisito Non Funzionale di packaging.
- **RNF_7:** Requisito Non Funzionale legale.

1.4 Riferimenti

Il presente SDD fa riferimento al progetto DiscoverEasy; per maggiori dettagli sui requisiti ai quali si fa riferimento, o per una panoramica più ad alto livello di questo progetto, consultare il RAD. Per orientarsi nella stesura dell'SDD, il team si è servito dei libri di testo consigliati durante il corso (sia Bruegge che Sommerville), delle slides e del materiale didattico fornito fino a questo punto del corso, ma soprattutto delle lezioni e delle spiegazioni del Docente.

1.5 Organizzazione del contenuto

Il presente documento illustra la struttura richiesta per il sito web DiscoverEasy. Si definiscono i dettagli dell'architettura del sistema proposto: si tratterà della decomposizione in sottosistemi, delle componenti hardware/software che verranno utilizzate ai fini della progettazione e di come verranno gestiti i dati persistenti.

La sezione riguardante l'architettura del sistema proposto si conclude con i dettagli su sicurezza e gestione degli accessi e sulla gestione delle condizioni limite. Il documento prosegue con una sezione che illustra i dettagli della decomposizione in sottosistemi spiegando, in modo più dettagliato, i servizi che ogni sottosistema offre agli altri sottosistemi. Il documento si conclude infine con un glossario.

2. Architettura del sistema proposto

2.1 Sintesi della sezione

Per realizzare il sistema, proponiamo un'applicazione web: tipologia software che, a nostro parere, calza perfettamente con l'idea proposta. L'obiettivo che il sistema si pone, come già ampiamente discusso nell'introduzione del presente documento e nei documenti precedenti, è quello di semplificare la compravendita di prodotti. Per raggiungere gli obiettivi del sistema ogni utente, che abbia effettuato una veloce procedura di registrazione, può essenzialmente inserire annunci.
L'architettura che verrà utilizzata nell'implementazione del sistema è la diffusissima architettura Client-Server: un server gestirà tutto ciò che riguarda i dati persistenti e la loro accessibilità da parte degli utenti; mentre la controparte client avrà il compito di presentare i vari contenuti in modo semplice ed efficace agli utenti finali.

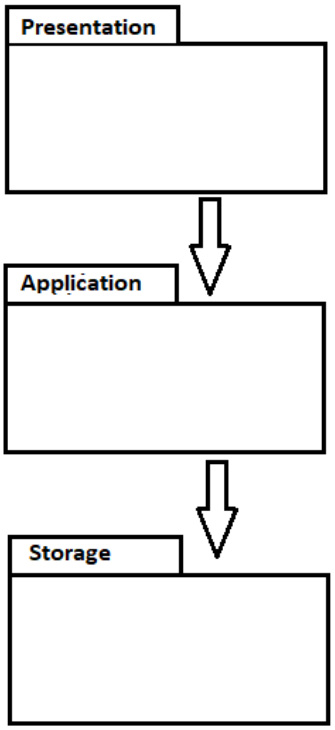
Il sistema è diviso in tre layer: Storage (gestione dati persistenti), Presentation (gestione interfaccia grafica), Application Layer (gestione della logica applicativa): i tre layer sono organizzati secondo il noto pattern "model – view – controller". Ad un livello di dettaglio maggiore, invece, è possibile identificare ben 9 sottosistemi, ognuno dei quali si occupa di una funzione specifica e ben definita, in modo da favorire una alta coesione ed un basso accoppiamento.

2.2 Decomposizione in sottosistemi

2.2.1 Decomposizione in layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

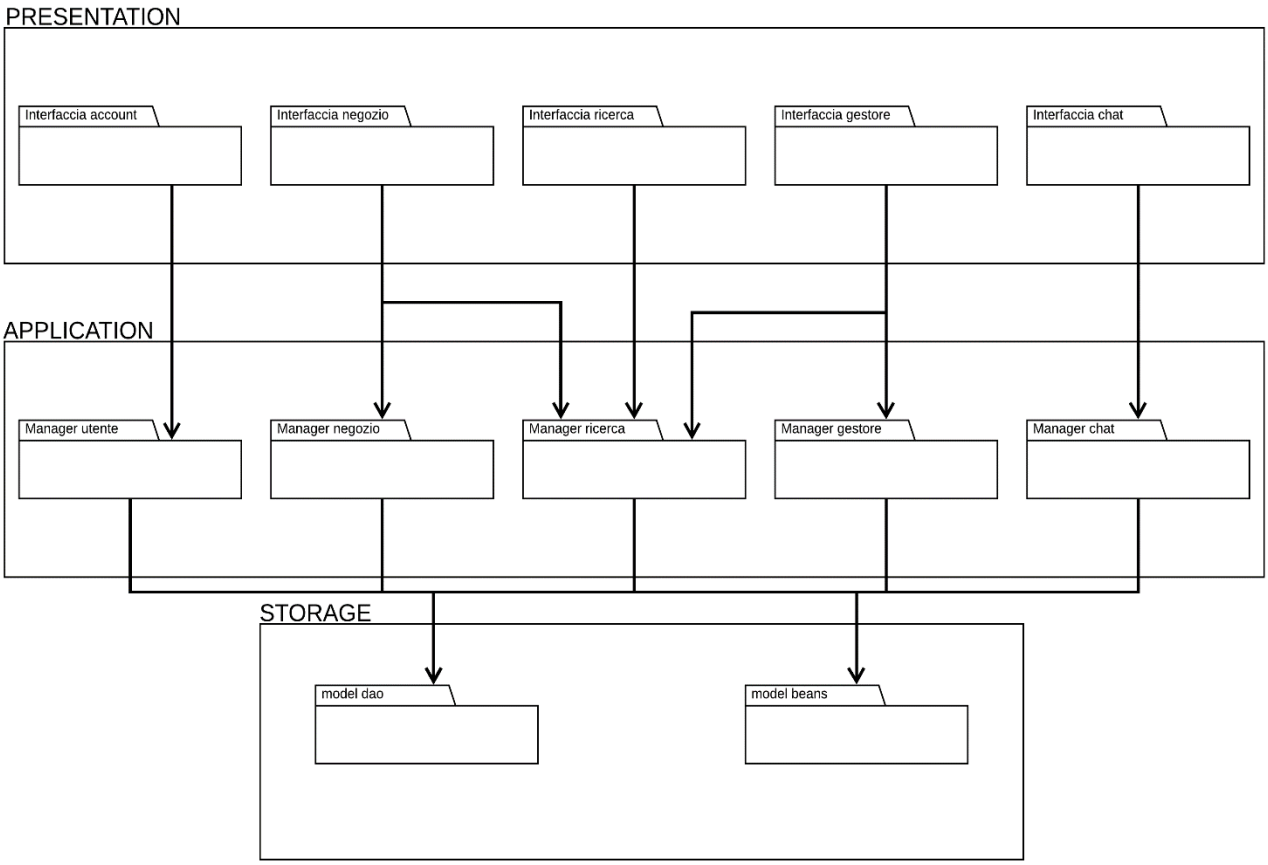
- Presentation (View): raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
- Application Layer: si occupa della gestione della logica del sistema, accettando l'input dall'utente e convertendolo in comandi per il modello e/o la vista.
- Storage (Model): si occupa della gestione e dello scambio dei dati tra i sottosistemi, indipendentemente dall'interfaccia utente;



2.2.2 Decomposizione in sottosistemi

Dopo un'attenta analisi funzionale, abbiamo scelto di dividere il nostro sistema nel seguente modo in quanto, per la divisione in componenti, avevamo bisogno di un basso accoppiamento ed un'elevata coesione tra i servizi offerti tra i componenti interni.

Il sistema si compone di dodici componenti che si occupano di gestirne aspetti e funzionalità differenti:



Il livello Presentation prevede 3 sottosistemi:

- Interfaccia account: gestisce le interfacce di registrazione, login e profilo;
- Interfaccia ricerca: gestisce le interfacce di ricerca;
- Interfaccia negozio: gestisce le interfacce di negozianti, in qualità di negoziante, nel sistema, saranno disponibili le funzionalità per gestire i propri annunci;
- Interfaccia chat: gestisce le interfacce di interazione tra i vari utenti;
- Interfaccia gestore: gestisce le interfacce di gestione del sistema.

Il livello Controller prevede a sua volta una suddivisione in 5 sottosistemi:

- Manager chat: sottosistema che gestisce l'interazione tra i vari utenti;
- Manager utente: sottosistema che permette all'utente di registrarsi, loggarsi e una volta registrato, di gestire le proprie informazioni e di modificarle in parte, se voluto;
- Manager ricerca: sottosistema che offre agli utenti di effettuare una ricerca, in base a vari parametri;
- Manager negozio: sottosistema che offre ai negozianti la possibilità di gestire i prodotti in vendita, quindi, di inserire prodotti, visualizzarli e rimuoverli;
- Manager gestore: sottosistema che permette al gestore di eliminare dal sistema inserzioni inappropriate o utenti non consoni alle politiche del sistema.

Il livello Storage prevede un unico sottosistema:

- Storage: formato da dao e beans, livello che gestisce ed immagazzina i dati persistenti.

2.3 Mapping hardware/software

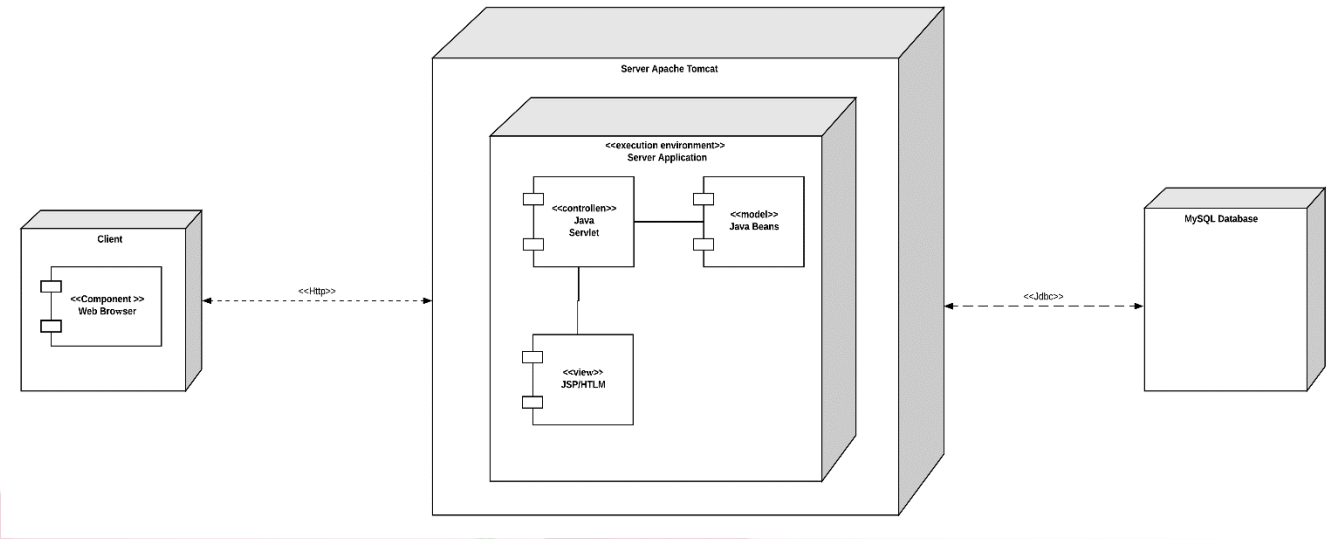
La struttura hardware proposta è costituita da un server centrale e dalle postazioni utenti (computer portatili o fissi), dotati di sistemi operativi che possono essere diversi fra loro.
All'application server, in cui risiedono gli oggetti di tipo control, entity e storage, si collegano le postazioni client; da una parte, la comunicazione tra client e server avverrà attraverso il protocollo HTTP, dall'altra, quella tra server e database si servirà di JDBC. Sarà il server stesso a garantire un'adeguata ripartizione delle operazioni consentite ai vari client, distinte in base alla modalità di autenticazione e considerando anche il caso in cui essa non avvenga.

In fase di accesso i dati sottomessi dall'utente saranno, infatti, inviati allo storage allo scopo di verificarne l'esattezza.

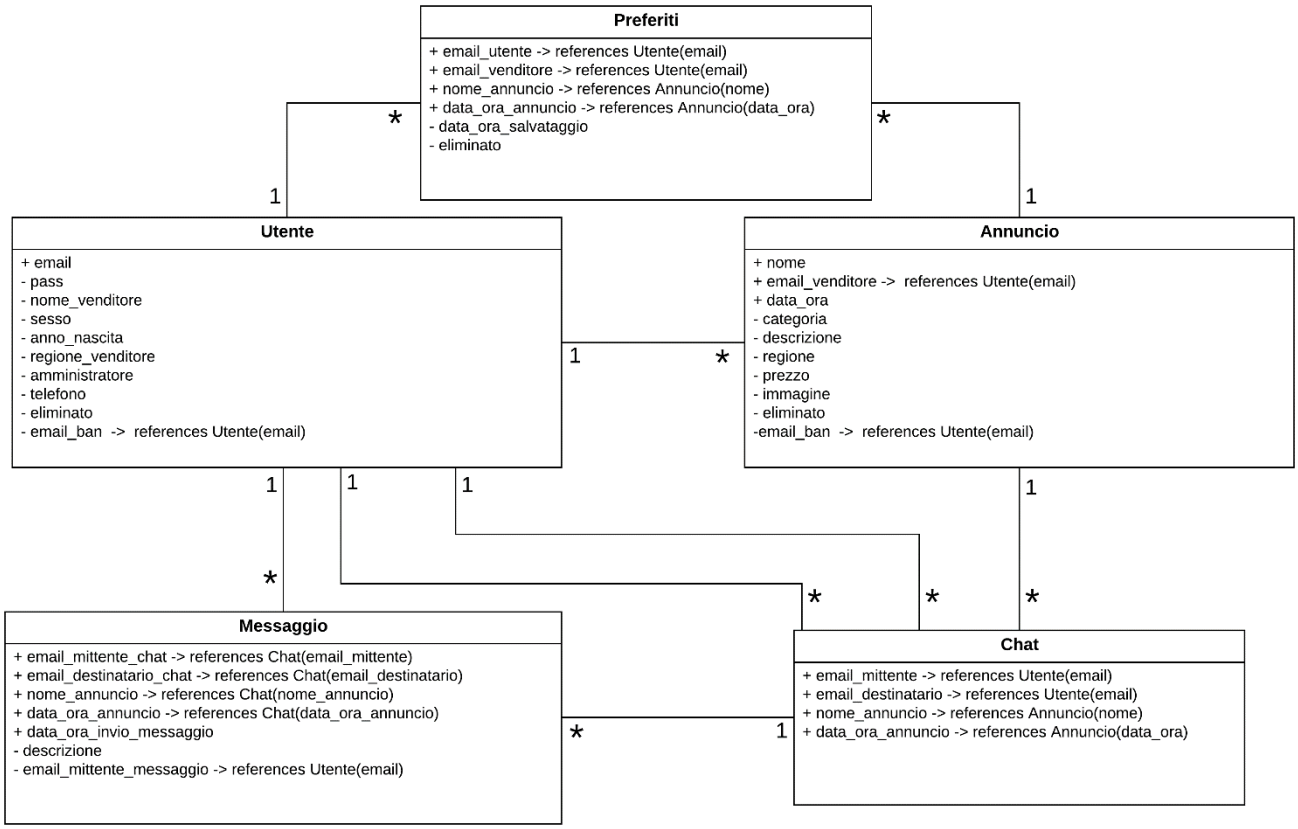
Al fine di assicurare una completa visione del sistema a "run time" è proposto un unico deployment diagram, mentre la comprensione della struttura statica a "compile time" potrà avvenire tramite i component diagram.

2.3.1 Deployment diagram

Il Deployment Diagram proposto descrive il sistema in termini di risorse hardware con le relative relazioni. Come mostrato dallo schema sottostante, gli utenti potranno interagire con DiscoverEasy tramite il browser su qualunque dispositivo semplicemente collegandosi all'indirizzo web del sito. Sarà compito poi del Controller interpretare gli eventi generati dall'utente, richiedere e prelevare le opportune risorse dal database ed inviare le informazioni richieste dagli utenti tramite il protocollo http, il quale si collegherà al Database tramite JDBC.



2.4 Gestione dei dati persistenti



Il sistema effettuerà query complesse sui dati. Si utilizzerà MySql come database relazionale per la gestione dei dati persistenti e SQL come linguaggio di programmazione per la creazione di tabelle e l'esecuzione di query.

2.5 Controllo degli accessi e sicurezza

L'accesso dell'utente e del gestore del sito verrà effettuato tramite l'inserimento di e-mail e password. Nella tabella sottostante sono riportate le operazioni che sono consentite agli utenti.

Attore / Oggetto	Annuncio	Lista preferiti	Messaggio	Utente
Inserzionista	Ricercare annunci Visualizzare lista annunci pubblicata Pubblicare annuncio Eliminare annuncio pubblicato	Aggiungere annuncio Eliminare annuncio Svuotare lista	Visualizza chat Visualizzare messaggi di una chat Inviare messaggio	Modifica password Eliminare account
Visitatore	Ricercare annunci	Aggiungere annuncio Eliminare annuncio Svuotare lista		
Gestore	Ricercare annunci Visualizzare lista annunci pubblicata Pubblicare annuncio Eliminare annuncio pubblicato Eliminare annuncio altro utente	Aggiungere annuncio Eliminare annuncio Svuotare lista	Visualizza chat Visualizzare messaggi di una chat Inviare messaggio	Modifica password Eliminare account Banna utente

2.6 Controllo flusso globale di sistema

Il web Server si occupa di gestire le varie richieste effettuate dal client. Il server smista le richieste alle classi java Servlet opportune che si occuperanno di gestire la richiesta ed eventualmente interagire con il model, e dare una risposta. Dopodiché il server crea la pagina jsp che verrà poi convertita in pagina html e visualizzata dall'utente

2.7 Condizioni limite

- Start-up: Per il primo start-up del sistema è necessario l'avvio di un web server e l'avvio di un database MySQL.
- Terminazione: Al momento della chiusura del software si ha la terminazione del sistema e verrà assicurata la consistenza dei dati.
- Fallimento: Possono verificarsi fallimenti nei seguenti casi:
    1. Nel caso in cui si verifichi un'interruzione dell'alimentazione. Non è previsto il ripristino dello stato del sistema prima del fallimento.
    2. Errore critico dell'hardware non è previsto una misura correttiva.
    3. Sovraccarico del database.

3. Servizi dei sottosistemi

PRESENTATION

- Interfacce che gestiscono l'interfaccia grafica e gli eventi generati dall'interazione dell'utente con il sistema.

GESTIONE UTENTE:

- Login
- Registrazione
- Visualizza pagina utente
- Logout

GESTORE RICERCA

- Ricerca
- Aggiungi inserzione a lista preferiti
- Visualizza lista preferiti
- Eliminare inserzione dalla lista preferiti
- Svuota lista preferiti

GESTORE NEGOZIO

- Visualizza inserzioni pubblicate
- Pubblica inserzione

- Cancella inserzione

GESTORE CHAT

- Visualizza chat
- Visualizza messaggi di una chat
- Invia messaggio

GESTORE ADMIN

- Cancella inserzione altro utente
- Banna utente

STORAGE

- Salvare dati
- Aggiornare dati
- Eliminare dati