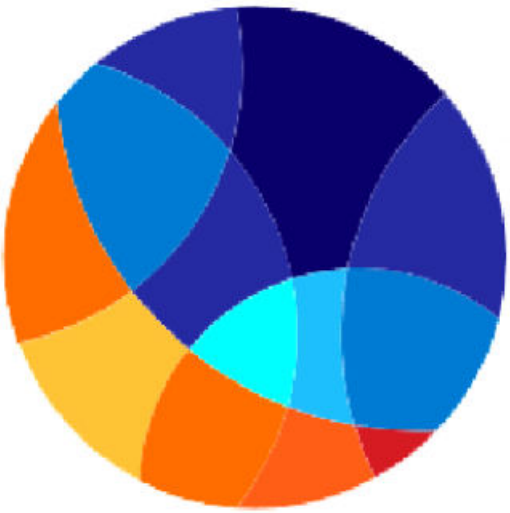
Università degli Studi di Salerno
Corso di Ingegneria del Software

Warehouse Connect
System Design Document



Introduzione

1.1. Intenti del sistema

Scopo di questo progetto è la realizzazione di un'applicazione generica che realizzi le funzionalità richieste per una gestione di un magazzino di un negozio di elettronica: il software che viene proposto è una sorta di database che va a sostituire i metodi cartacei utilizzati finora dal negozio per gestire il magazzino. Ciò che proponiamo è quindi un software web-based dotato di interfaccia grafica easy-use, in modo tale che qualsiasi figura professionale all'interno del negozio (che siano gestori, commessi, ecc) abbiano modo di utilizzare il software. Lo scopo del prodotto è fornire l'accesso ad un database in modo tale che dal negozio sia possibile conoscere quali prodotti si hanno a disposizione nel magazzino.

1.2. Obiettivi designati

1.2.1. Criteri di performance

Ogni azienda, piccola o grande che sia, durante il suo ciclo di vita ha bisogno di un modo per organizzare e schedulare il lavoro. In passato, le ricerche consistevano in un gran numero di documenti cartacei e venivano conservati in grandi archivi, causando problematiche quali la possibilità di perdita di dati o informazioni, e l'impossibilità, di fronte ad un vasto numero di archivi, di reperire documenti.
Per ovviare ai problemi sopraelencati vengono in aiuto lo sviluppo informatico e la tecnologia. Il software realizzato permette ad un negozio l'accesso diretto al proprio magazzino, cioè in ogni momento sarà possibile visionare quali prodotti sono in esso disponibili per la vendita. Il software che sviluppiamo prevede 2 figure ad utilizzarlo:

- *commessi*, in grado di accedere al database solo per la ricerca
- *gestori* (manager, admin, ...), in grado di accedere al database per la ricerca e per la modifica dello stesso database.

1.2.2. Criteri di manutenzione

Il software dovrà avere tutte quelle caratteristiche da consentire una semplice ed immediata manutenzione, ovvero la possibilità di apportare cambiamenti di qualsiasi genere senza la necessità di modificare l'organizzazione interna e l'implementazione stessa.

- **Estendibilità**: il sito dovrà presentare una struttura suddivisa per moduli, per poter consentire, nella maniera più semplice possibile, un'eventuale aggiunta di nuove features, che siano richieste o dal cliente o dall'effettivo uso del sistema.
- **Modificabilità**: la struttura interna del sistema dovrà essere concepita in modo tale da consentire una semplice e veloce implementazione di un'eventuale modifica richiesta relativa ad uno o più componenti che, eventualmente, hanno fatto registrare comportamenti anomali o necessitano di qualche miglioramento estremamente significativo. Viene quindi ribadita la necessità di avere un prodotto fortemente orientato alla modularità.

- **Portabilità**: trattandosi di un prodotto web-based, la portabilità è un fattore elementare e agevolato: chiaramente, in quanto web-based, il prodotto dovrà utilizzare un browser. Proprio per questo motivo è basilare il non essere, il software, browser-tied. Il prodotto non dev'essere quindi legato a nessun browser in particolare ma deve comportarsi nello stesso modo su qualsiasi browser e su qualsiasi macchina host.
- **Leggibilità**: per facilitare eventuali aggiunte, modifiche o correzioni al codice, tale codice sarà opportunamente prodotto nella maniera più corretta possibile. Questo permetterà a qualsiasi sviluppatore che leggerà il codice di leggerlo e comprenderlo con il minimo sforzo, ottimizzando il rendimento e le tempistiche di intervento sul codice stesso.

1.2.3. Criteri di costi

- **Costi di sviluppo**: in questa fase, utilizzando software open-source, non sono richiesti particolari costi. Non sono inoltre necessari specifici dispositivi hardware per la realizzazione dello stesso, sono sufficienti dei comuni Personal Computer.
- **Costi di distribuzione**: la natura del sistema software da realizzare, rende superfluo il costo aggiuntivo per quanto riguarda la fase di distribuzione del prodotto ultimato. I costi per un eventuale formazione dell'utenza sono praticamente nulli.

1.2.4. Criteri per l'utente finale

- **Utilità**: il prodotto finale deve garantire la possibilità di effettuare tutte e sole le operazioni previste in fase di sviluppo. Ogni volta che quindi un utente accederà al sistema, quest'ultimo dovrà rispettare il comportamento atteso. Il sistema dovrà far registrare una significativa utilità all'interno del contesto in cui è utilizzato.
- **Usabilità**: il prodotto finale dovrà risultare semplice e intuitivo, in modo da agevolare gli utenti durante l'utilizzo del sistema. Per questo motivo tale software viene dotato di un'interfaccia grafica user-friendly, basata su pulsanti da cliccare e form da compilare.

1.2.5. Criteri di dipendenza

Il sistema dovrà essere in grado di gestire in maniera efficace tutti gli imprevisti che potrebbero presentarsi al momento del suo utilizzo. La risposta del sistema a questi inaspettati eventi dovrà essere immediata e significativa in modo tale da garantire il suo corretto funzionamento in futuro.

- **Robustezza**: il sistema dovrà essere in grado di impedire all'utente di alterare, in maniera accidentale o volontaria, il suo corretto funzionamento, ad esempio tramite la modifica dell'URL per accedere illegalmente a porzioni di software dedicate esclusivamente all'admin. Tutte le possibili interazioni tra l'utente e il sistema dovranno quindi essere opportunamente controllate e gestite in modo tale da impedire il verificarsi di situazioni anomale che

potrebbero lasciare il sistema stesso in uno stato inconsistente.

- **Affidabilità**: il sistema dovrà garantire in qualsiasi momento l'utilizzo continuato delle sue applicazioni, anche in presenza di carichi di lavoro notevoli, permettendo all'utente di essere sempre a conoscenza dell'esito delle operazioni richieste, sia in caso di successo che di fallimento. In questo modo l'utente potrà avere una chiara percezione dello stato del sistema.
  Per quanto riguardo invece gli eventi eccezionali, come la caduta del database, non sarà il sistema a gestire gli eventuali problematiche da essi causati e, quindi, non sarà possibile in alcun modo recuperare eventuali dati andati perduti o operazioni non portate a termine.
- **Disponibilità**: l'applicazione dovrà essere disponibile 24/7, ed eventuali modifiche al database (aggiunta o rimozione di prodotti, ...) saranno rese visibili durante l'utilizzo stesso dell'applicativo.
- **Sicurezza**: il sistema dovrà garantire un livello di sicurezza molto alto, dal momento che, a seconda del proprio ruolo all'interno del negozio, si ha accesso a features diverse. A tal proposito il software utilizza un sistema di login basato sulla compilazione di form ID-Password.
  Non sarà possibile registrare nuovi utenti se non tramite codice (questo ovviamente per non permettere a figure non gestionali di registrarsi come tale).
- **Fault Tolerance**: il sistema dovrà essere in grado di rispondere in maniera adeguata ad eventuali comportamenti errati dell'utente che interagisce con esso. Il software guiderà l'utente tramite la compilazione di appositi campi a seconda dell'operazione da svolgere: il caso di non compilazione di campi richiesti verrà notificato all'utente il quale, eventualmente, li inserirà al fine di completare l'operazione da portare a termine.
  Invece, il sistema non dovrà gestire eventuali fallimenti causati da eventi come, ad esempio, la caduta del DBMS e, quindi, in questo caso, i dati inseriti andranno persi e le operazioni richieste non saranno portate a termine.

1.3. Definizioni, Acronimi, Abbreviazioni

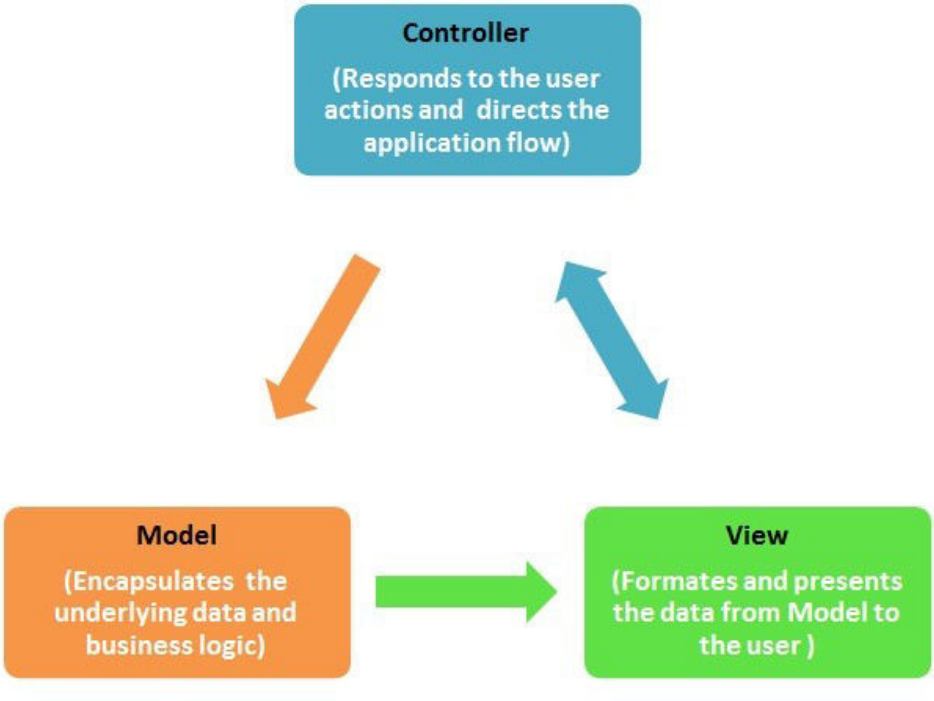
- **SDD**: System Design Document.
- **GUI**: Graphical User Interface.
- **RAD**: Requirement Analysis Document.

Sistema

2.1. Overview

L'architettura scelta è di tipo **MVC**, dove:

- il *Model* è rappresentato da un DB in Derby;
- la *View* avviene tramite JSP;
- il *Controller* gestisce la comunicazione tra DB e JSP tramite Java.



Per implementare il nostro sistema abbiamo scelto *Apache Derby*, data la sua disponibilità open source e la robustezza che ha dimostrato nel corso degli anni. Inoltre, nonostante presenti meno feature rispetto a MySQL (di cui abbiamo valutato e scartato la scelta), risulta essere più efficiente, veloce e di facile utilizzo.
Il sistema mantiene traccia dei dati suddividendoli ed incapsulando ciascun prodotto che viene inserito. I prodotti verranno suddivisi per categoria, e ciascun prodotto avrà inoltre un'immagine. Di tale immagine, che viene caricata in una cartella, ne viene memorizzato l'indirizzo e tramite l'indirizzo è possibile permettere al DB di creare l'associazione prodotto-immagine.
L'interazione con il database è completamente interfacciata tramite una GUI molto semplice, basata su pulsanti, form, e conferme.
La ricerca all'interno del database viene effettuata tramite query dinamiche generate al momento della richiesta dell'utente dopo aver compilato i campi appositi: ad esempio, per cercare un prodotto nel database, sarà necessario inserire o il codice del prodotto, o le sue caratteristiche, e in base ai campi inseriti verrà generata una query che cercherà secondo i campi compilati.

2.2. Hardware/Software Mapping

Il sistema è strutturato tramite un'architettura client/server.
Il client (Web browser) interagisce con il server attraverso il protocollo HTTP. Il server sarà l'unico ad aver accesso ai dati persistenti che saranno salvati su di un database gestito dal DBMS Apache Derby.
Il sistema sarà utilizzato da:

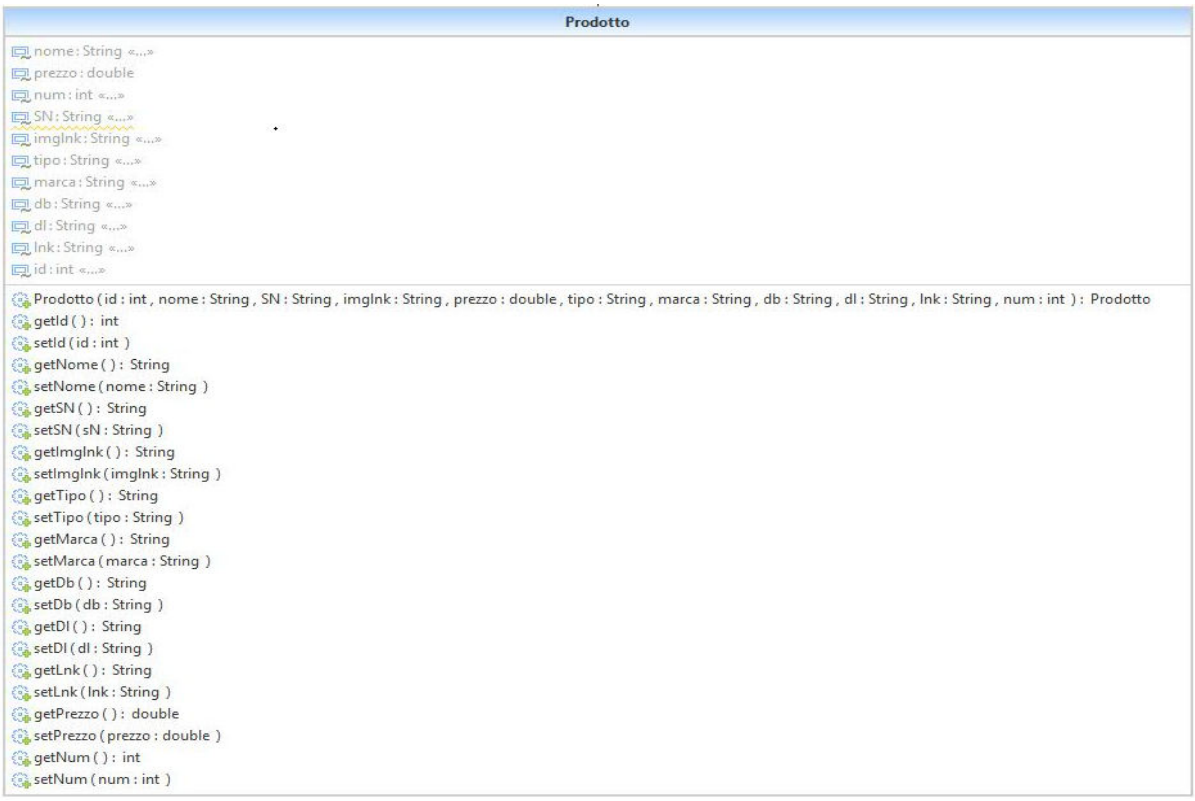
- *admin* (unica tipologia di utente per cui è prevista l'esistenza di una classe);
- *commesso*.

2.3. Gestione dei dati persistenti

Gli oggetti identificati come parti integranti del sistema sono i seguenti:

- Utente

- Prodotto



2.3.1. Utente

L'oggetto Utente racchiude l'ID e la Password relativi ad un utente con credenziali amministrative. Bisogna quindi intendere un oggetto Utente come un admin. Gli attributi di un oggetto Utente sono:

- *user*, che rappresenta l'ID per accedere alla sezione amministrativa;
- *pass*, che rappresenta la password per accedere alla sezione amministrativa;

Se entrambi i campi del form (ID-Password) verranno inseriti correttamente e faranno riferimento ad un utente realmente esistente, si potrà accedere alla sezione admin.

2.3.2. Prodotto

L'oggetto Prodotto racchiude tutte le caratteristiche associate ad un prodotto. Tali caratteristiche vengono assegnate tramite un form nel momento in cui viene, da un admin, inserito un prodotto nel database. Gli attributi di un oggetto Prodotto sono:

- *nome*, che rappresenta il nome del prodotto. Il nome di un prodotto è in genere ciò che meglio è conosciuto da un acquirente;
- *prezzo*, che rappresenta il prezzo attuale del prodotto;
- *num*, che rappresenta il numero di articoli presenti relativi ad un unico prodotto;
- *SN*, che rappresenta il Serial Number del prodotto stesso;
- *imglnk*, che rappresenta il link (della directory) da cui prendere l'immagine;
- *tipo*, che rappresenta la tipologia del prodotto (TV, Computer, ...);
- *marca*, che rappresenta la marca del prodotto;
- *db*, che rappresenta la descrizione breve del prodotto;
- *dl*, che rappresenta la descrizione lunga e completa del prodotto;
- *lnk*, che rappresenta parte del link generato dinamicamente ogni volta che viene aperto un prodotto per visualizzarne le caratteristiche;
- *id*, che rappresenta l'ID generato automaticamente ogni volta che viene inserito un prodotto. L'ID è utilizzato per organizzare in ordine cronologico i prodotti nella ricerca.

2.4. Controlli d'accesso e sicurezza

Ciascun utente supportato dal sistema deve essere concesso solo di determinate possibili operazioni. Pertanto, sarà descritta la politica di sicurezza in modo tale da evitare accessi non autorizzati alle sezioni riservate del sistema.

Oggetti \ ***Utente***	*Ricerca*	*Inserimento*	*Rimozione*	*Modifica*
Admin	Può ricercare prodotti all'interno del database.	Può inserire prodotti all'interno del database.	Può rimuovere prodotti dal database.	Può modificare prodotti nel database.
Commesso	Può ricercare prodotti all'interno del database.	N.A.*	N.A.*	N.A.*

*N.A. = Non Autorizzat

- Admin
  L'admin, prima di poter accedere all'area riservata, avrà bisogno di autenticarsi. L'autenticazione avviene tramite l'inserimento di ID e Password. Acceduto all'area gestionale, avrà accesso a features extra, quali l'inserimento e la rimozione di articolo nel database e la modifica degli articoli in esso già presenti.
- Commesso
  Il commesso avrà accesso ad una sola feature, ovvero la ricerca di articoli presenti nel database. Questo perché dovrà essere pronto a fornire informazioni sulla presenza o meno di articoli in magazzino. Inoltre di tali articolo potrà visualizzarne le informazioni e i dettagli.

2.5. Controlli software globali

Nel flusso di controllo relativo al software del sistema da realizzare, non esisterà alcuna sequenza di esecuzione prestabilita, ma di volta in volta, l'interazione dell'utente con le componenti dei vari sottosistemi identificati genererà degli specifici comportamenti del sistema.

2.6. Condizioni limite

- *Installazione*
  Startup: Il sistema lato server si avvia nel momento in cui è lanciato il web server. Il sistema lato client è inizializzato ogni volta che un candidato accede al portale (apertura della sessione) inserendo l'URL del sito nel browser.
- *Startup – Shutdown*
  Il sistema lato server non può terminare (a meno di guasti).
  Il sistema lato client termina con la chiusura del browser o con il logout.

Servizi di **S**istema

3.1. Sottosistemi

Sottosistemi	*Descrizione sottosistemi*
Servizio di Autenticazione	Si occupa di gestire dati relativi a: - Login/Logout
Gestione Dati	Si occupa di gestire dati relativi a: - Prodotti
Servizi di Ricerca	Si occupa di gestire dati relativi a: - Ricerca tramite query dinamiche - Ordinamento in ordine cronologico quando si ricerca per categoria