

System Design Document

1.0 Introduzione

1.1 Scopo del Sistema

Manga Up è una piattaforma web based che si occuperà di mettere a disposizione dei propri clienti, manga e altri prodotti nipponici che siano sempre disponibili, stando al passo con le nuove uscite di volumi e serie. I prodotti che Manga Up mette a disposizione sono maggiormente richiesti da un pubblico avente età compresa tra i 12 e i 25 anni, tuttavia non mancano clienti che hanno una passione tale da ricercare prodotti che erano molto popolari qualche anno fa e che Manga Up mette a loro disposizione. Gli utenti del sito sono: ospiti che visitano il sito e prendono visione dei prodotti che esso mette a loro disposizione, clienti che possono acquistare i prodotti forniti dal sito, gestori degli ordini che gestiscono eventuali ordini effettuati dai clienti, gestori del catalogo che si occupano di gestire il catalogo dei prodotti che il sito mette a disposizione.

1.2 Design Goals

Durante la fase di System Design abbiamo deciso di dare maggiore priorità ai seguenti obiettivi di progettazione:

Robustezza

Per fare in modo che in presenza di input non validi inseriti dall'utente oppure in presenza di situazioni limite a livello di applicazione, il sistema resista in modo tale da rispondere con appropriati messaggi di errore nel primo caso, e che riesca a trovare l'alternativa giusta ad una situazione non favorevole e non convenzionale nel secondo. Questo obiettivo di progettazione estende il concetto espresso dal requisito non funzionale richiesto dal client relativo alla reliability sugli input forniti dall'utente

Usabilità

Per fornire un'adeguata interfaccia utente per i vari tipi di attori previsti dal sistema, in maniera tale da rendere agevole e quanto più semplice possibile l'utilizzo delle funzionalità del sistema ed anche per rendere quanto più breve possibile il tempo medio impiegato da un

utente per imparare ad operare con l'applicazione. Questo obiettivo di progettazione è una riconferma di quanto richiesto dal client sulla usability nel problem statement.

Manutenibilità

Per incentivare l'evoluzione dell'architettura software in un' eventuale reiterazione dello sviluppo del sistema ed anche per agevolare gli sviluppatori nella modifica delle componenti contenute in esso. Ciò avverrà perseguendo l'obiettivo di sviluppare il software in termini di sottosistemi quanto più coesi e quanto meno accoppiati tra di loro, i quali saranno poi raggruppati in strati di funzionalità fornite allo stesso livello di astrazione.

La coesione interna ai sottosistemi verrà massimizzata cercando di inserirvi tutte e sole quelle classi che forniscono operazioni che hanno scopi comuni tra di esse. Questo obiettivo di progettazione considera ed estende il requisito non funzionale di supportability riguardo alla stratificazione dell'applicazione su tre livelli di responsabilità.

1.3 Riferimenti

- Riferimento al Requirement Analysis Document

1.4 Panoramica

Il seguente documento di System Design (SDD) mostra i dettagli tecnici del design del sistema Manga-UP. Altri dettagli riguardanti le funzionalità e le caratteristiche del sistema possono essere trovati nel documento dell'analisi dei requisiti (RAD) mentre una panoramica generale può essere trovata nel Problem Statement. In questo documento viene specificata un'introduzione generale all'architettura e agli obiettivi di design che il sistema si propone di raggiungere, viene proposta una suddivisione del sistema in sottosistemi definendone inoltre il mapping Hardware/Software, in modo da assegnare ogni sottosistema ad uno specifico hardware. Sono inoltre descritti il controllo dell'accesso e i problemi di sicurezza legati al sistema, evidenziando anche il controllo generale del software e il controllo dei Boundary, trattando gli stati iniziali e la gestione del sistema.

2.0 Architettura del Sistema Corrente

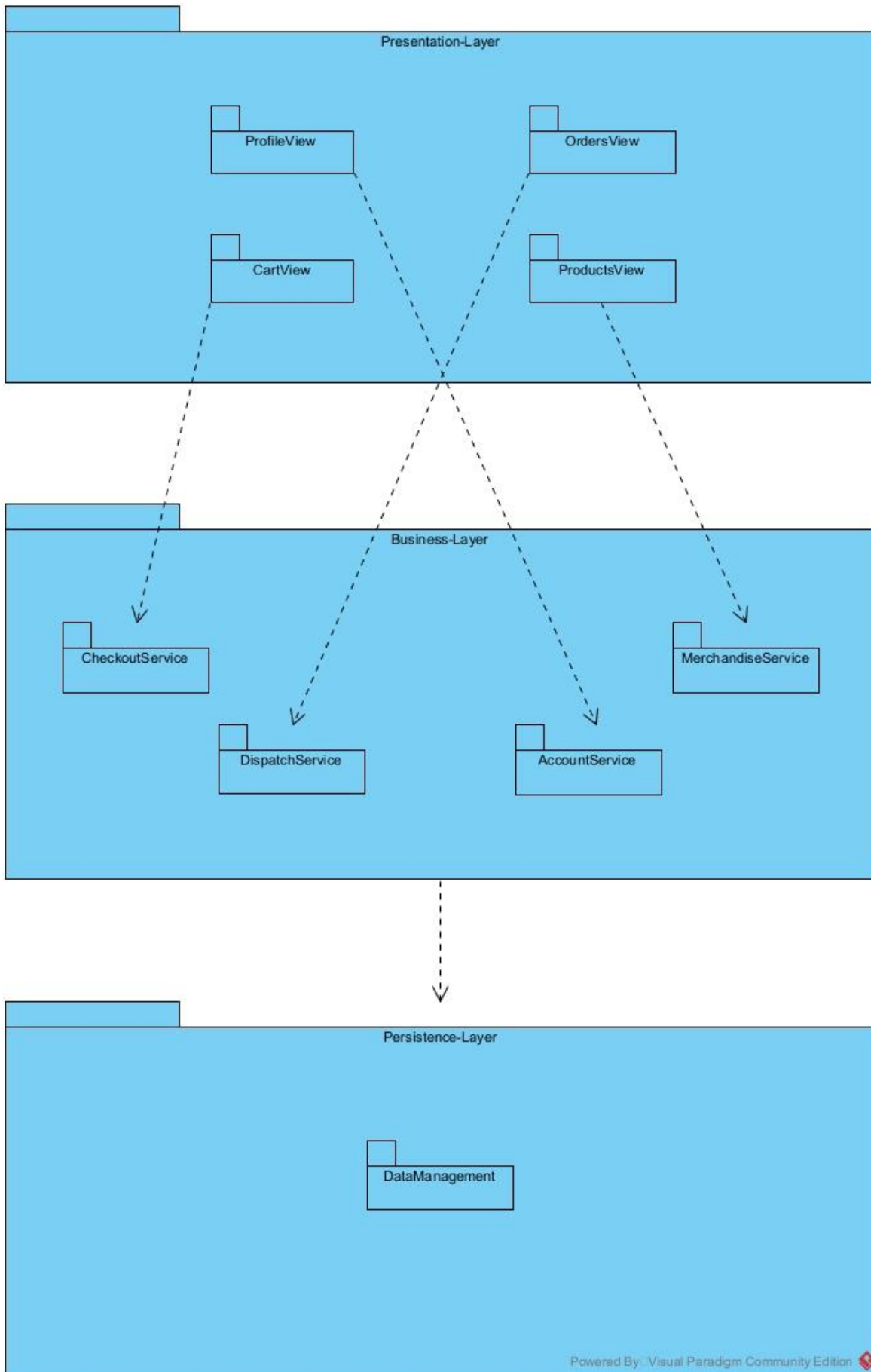
Data l'assenza di un sistema esistente, non c'è alcun riferimento ad un'architettura corrente

3.0 Architettura del Sistema Proposto

3.1 Panoramica

L'architettura del sistema proposto sarà basata sulla suddivisione dei sottosistemi in 3 strati, ognuno dei quali è in parte responsabile della fornitura delle funzionalità offerte dal sistema. In particolare abbiamo un architettura chiusa (*Closed Architecture*) organizzata in maniera gerarchica al cui apice si trova lo strato di presentazione o *Presentation Layer* il quale ha il compito di occuparsi della visualizzazione delle pagine richieste dai client web; subito dopo si trova lo strato di controllo o di business (*Business Logic Layer*) il quale ha la responsabilità di mediare tra le richieste del client web e la parte model del sistema, con l'obiettivo di eseguire le operazioni di controllo del sistema e di fornire i dati appropriati alle view (le quali si occupano della visualizzazione) che li richiedono; infine abbiamo lo strato di persistenza o *Persistence Layer* il quale si occuperà di gestire i dati persistenti del sistema, effettuando l'aggiornamento dei dati e garantendo la consistenza di essi quando vengono recuperati.

3.2 Scomposizione In Sottosistemi



Nome Componente	Servizi
ProfileView	Visualizzazione delle pagine dell'applicazione preposte ad accedere ai dati personali dell'end-user, modificare i dati dell'account dell'end-user, gestire gli account dei gestori del sistema.
OrdersView	Visualizzazione delle pagine dell'applicazione preposte ad accedere agli ordini effettuati dall'end-user, gestire ordini in arrivo da parte del gestore degli ordini, visualizzazione della lista degli ordini.
CartView	Visualizzazione delle pagine dell'applicazione preposte ad accedere agli elementi presenti nel carrello dell'end-user, aggiungere/rimuovere elementi dal carrello, procedere con l'acquisto del/dei prodotto/i.
ProductsView	Visualizzazione delle pagine dell'applicazione preposte ad accedere ai prodotti presenti nel catalogo del sistema, overview del singolo prodotto, aggiunta/modifica/rimozione di un prodotto da parte del gestore del catalogo.
CheckoutService	Delegazione delle richieste relative all'aggiunta, rimozione, modifica degli elementi nel carrello, acquisto del/i prodotto/i nel carrello
DispatchService	Visualizzazione degli ordini effettuati da un particolare utente, gestione di un ordine da parte del gestore degli ordini
AccountService	gestione dei profili utente da parte del gestore degli utenti
MerchandiseService	Delegazione delle richieste relative alla fornitura dei prodotti trovati nel catalogo in base ad una particolare ricerca, aggiunta, modifica o rimozione di un prodotto dal catalogo
DataManagement	Si occupa di gestire il lato persistenza garantendo le proprietà ACID di una transazione e fornendo le operazioni CRUD per tutti i dati persistenti che lo permettono

Dato il diagramma presentato e le relative dipendenze presenti in esso abbiamo individuato 4 sottosistemi principali basati sulle funzionalità e sono : **Account**, **Merchandising**, **Order**, e **Cart** ognuno dei quali fornisce uno o più servizi che elenchiamo di seguito per ogni sottosistema:

Account	
Registrazione	Consente all'utente di registrarsi potendo successivamente accedere a tutte le funzionalità di Manga-UP previste per gli end-user
Accesso	Consente all'utente di accedere al sistema tramite il login e di fare il logout
Gestione profilo personale	Consente all'utente di modificare le informazioni relative al suo account
Gestione account	Consente all'utente di aggiungere e rimuovere gli account dei gestori del sito

Merchandising	
Visualizzazione catalogo	Consente all'utente visualizzare il catalogo dei prodotti a cui è interessato
Ricerca nel catalogo	Consente all'utente finale e non di filtrare gli elementi del catalogo in base a più criteri
Visualizza prodotto	Consente all'utente di visualizzare l'overview di uno specifico prodotto presente nel catalogo
Gestione prodotto	Consente all'utente di aggiungere un prodotto al catalogo, rifornire le quantità in magazzino e rimuoverli.

Ordine	
Visualizzazione Ordini	Consente all'utente di visualizzare lo storico degli ordini effettuati a suo nome
Gestione ordini	Consente all'utente di assegnare un ordine al relativo corriere e quindi di commissionarne la sua spedizione

Cart

Visualizza informazioni carrello

Consente all'utente di visualizzare l'insieme dei prodotti che ha precedentemente inserito nel carrello

Gestione Carrello

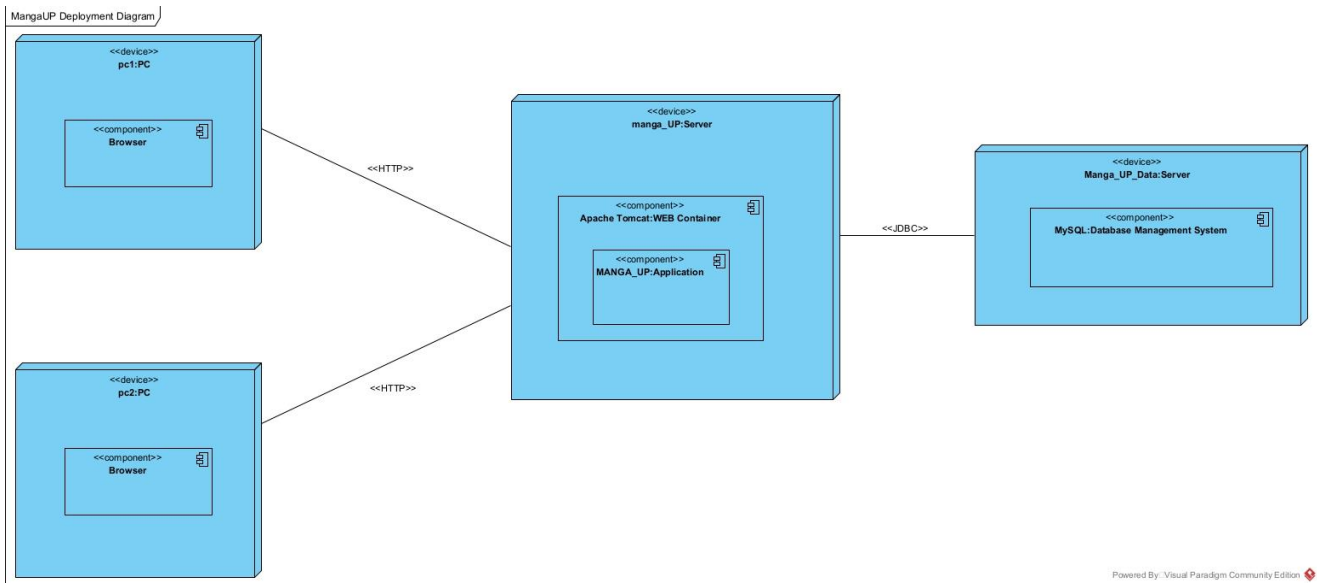
Consente all'utente di aggiungere,rimuovere e modificare la quantità di un prodotto presente nel proprio carrello

Checkout

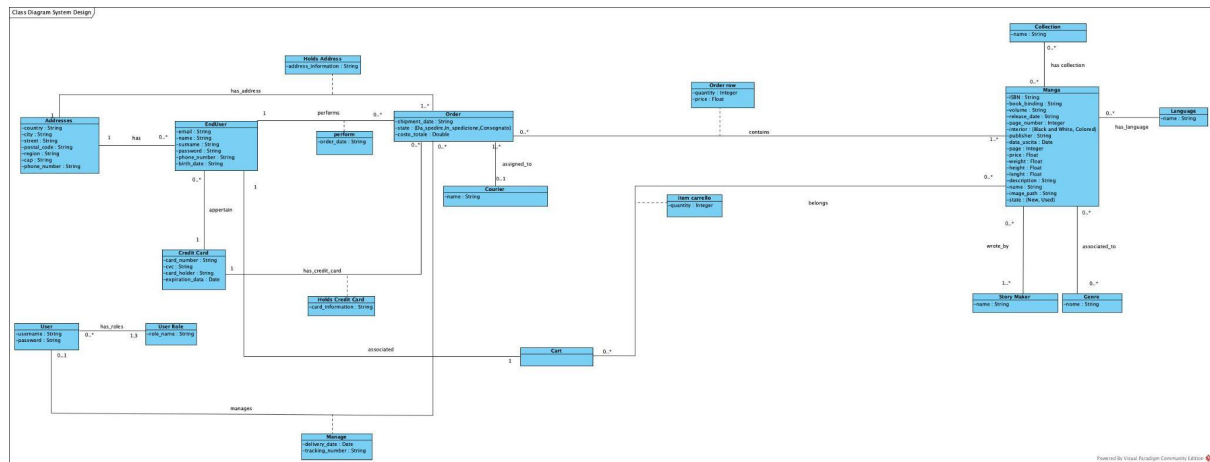
Consente all'utente procedere all'acquisto dei prodotti presenti nella sua istanza di carrello

3.3 Hardware/Software Mapping

Il sistema utilizza un'architettura Client/Server. Il Web Server è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet mentre l'interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page). Il Client è rappresentato dal Web Browser utilizzato dall'utente. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server, e da query in JDBC tra server e database.



3.4 Gestione Dati Persistenti



3.5 Controllo degli accessi e sicurezza

	End-User non registrato	End-User registrato	Gestore Catalogo	Gestore Ordini	Gestore Utenti
Account	Registrazione	Accesso, Gestione profilo personale	Accesso	Accesso	Accesso, Gestione Account
Merchandising	Visualizzazione Catalogo, Visualizzazione prodotto	Visualizzazione Catalogo, visualizzazione prodotto	Gestione prodotto, ricerca nel catalogo	-	-
Order	-	Visualizzazione Ordini	-	Visualizzazione ordine da gestire /Gestione ordine	-
Cart	-	Visualizzazione Carrello, Gestione carrello, Checkout	-	-	-

Ogni tipo di utente: end-user non registrato, end-user registrato, gestore catalogo, gestore degli ordini e gestore utenti avranno a disposizione diverse interfacce grafiche (ad eccezione per l'end-user non registrato e registrato che hanno alcune interfacce in comune) in modo che ogni utente possa accedere solo alle rispettive funzionalità che il sistema gli permette di controllare. Gli utenti gestori possono manipolare gli oggetti del sistema solo se autenticati.

L'end-user non registrato ha un'interazione parziale con gli oggetti riguardanti gli end-user registrati, per poter usufruire di funzionalità aggiuntive è necessaria l'autenticazione. Per poter autenticare gli end-user è necessaria una mail e una password, mentre per i gestori è necessaria un username ed una password. Per l'autenticazione degli end-user utilizziamo due pagine differenti. Password e dati della carta di credito vengono criptati tramite una funzione SHA-6.

3.6 Controllo Software Globale

La componente **Server** si occupa di gestire le varie richieste dei **client**. Il server smista le richieste alle classi **Java Servlet** opportune che si occuperanno di gestire la richiesta, eventualmente interagire con il **DBMS**, e dare una risposta. Dopodiché il server cattura la risposta della java servlet e la invia al client sotto forma di pagina html.

Per quanto riguarda la decentralizzazione del controllo del nostro sistema, abbiamo a disposizione diversi oggetti control che vengono eseguiti in thread indipendenti, i quali sono creati dal Web Container alla ricezione di richieste HTTP per assicurare l'accesso concorrente alla nostra infrastruttura.