



UNIVERSITÀ DEGLI STUDI DI SALERNO  
**DIPARTIMENTO DI INFORMATICA**  
INGEGNERIA DEL SOFTWARE 2017-2018

System Design Document versione 2.0

# Easybook

Data: 14/01/2018

## Sommario

1.PARTECIPANTI:	2
2.REVISION HISTORY:	2
3.INTRODUZIONE:	3
3.1. SCOPO DEL SISTEMA:	3
3.2. OBIETTIVI DI PROGETTAZIONE:	3
3.3. OBIETTIVI DI PROGETTAZIONE:	5
3.4. RIFERIMENTI:	6
3.5. PANORAMICA:	6
4.ARCHITETTURA SOFTWARE CORRENTE:	6
5.ARCHITETTURA SOFTWARE PROPOSTA:	6
5.1 PANORAMICA	7
5.2 DECOMPOSIZIONE SUBSYSTEM	7
5.3 HARDWARE / SOFTWARE DI MAPPATURA	7
5.4 COMPONENT/DEPLOYMENT DIAGRAM	9
5.5 GESTIONE DATI PERSISTENTI	37
5.6 CONTROLLO ACCESSI E SICUREZZA	38
5.7 SOFTWARE DI CONTROLLO GLOBALE	39
5.8 BOUNDARY CONDITIONS	39
6.SUBSYSTEM SERVICES:	40

## 1.PARTECIPANTI:

Nome	Matricola
Andrea Di Lucia	0512103222

## 2.REVISION HISTORY:

Data	Versione	Descrizione	Autore
08/12/2017	1.0	Prima versione “System Design Document ”	Andrea Di Lucia
14/01/2018	2.0	Revisione finale documento	

### 3. INTRODUZIONE:

Il sistema è stato creato per rendere possibile la consultazione del catalogo della biblioteca e la prenotazione di libri.

Con tale sito il bibliotecario spera di aumentare l'utenza della biblioteca rendendola accessibile anche a chi per vari motivi non ci si può recare.

#### 3.1. SCOPO DEL SISTEMA:

"Easybook" è un sito che permette agli utenti tesserati di accedere con il numero di tessera per poter sfogliare il catalogo dei libri disponibili in biblioteca e di prenotare un libro senza doversi recare in biblioteca.

Inoltre premette di effettuare la ricerca per titolo, di controllare i libri che si possiede in prestito e di gestire un carrello per effettuare l'ordine.

Il bibliotecario può registrare nuovi utenti e avere una lista completa di tutti gli utenti registrati, aggiungere libri al catalogo o rimuoverli.

#### 3.2. OBIETTIVI DI PROGETTAZIONE:

Il sistema deve rispettare i seguenti obiettivi di progettazione:

##### 3.2.3 CRITERI DI PERFORMANCE

###### TEMPO DI RISPOSTA

Chiaramente, i tempi di risposta per le operazioni che deve eseguire l'utente, deve essere inferiore ai 3 secondi.

###### THROUGHPUT

Le prestazioni del sistema hanno come unica limitazione l'utilizzo di un database relazionale utilizzato come repository centrale e del tipo di web server utilizzato per la gestione del portale.

###### MEMORIA

Il sistema ha bisogno di un database relazionale di tipo MySQL per memorizzare i dati relativi agli utenti, agli ordini e ai prodotti.

##### 3.2.4 CRITERI DI AFFIDABILITÀ

###### ROBUSTEZZA

Il sistema garantisce il funzionamento anche nel caso in cui vengano inseriti input errati: in questo caso, l'utente verrà avvisato dell'errore nell'inserimento.

#### **AFFIDABILITÀ**

Il sistema deve portare a termine ogni sua operazione. Le operazioni, quindi, devono essere atomiche.

#### **SICUREZZA**

L'accesso utente e amministratore sarà protetto dall'inserimento di userID e password.

### **3.2.5 CRITERI DI COSTO**

#### **DEVELOPMENT COST**

Non sono presenti costi di sviluppo, in quanto utilizzeremo Apache Tomcat come web server e MySQL come DBMS e i componenti lavorano gratuitamente.

#### **DEPLOYMENT COST**

Non sono previsti costi di deployment.

### **3.2.6 CRITERI DI MANTENIMENTO**

#### **ESTENDIBILITÀ**

L'utilizzo di linguaggi, quali HTML, JAVASCRIPT, CSS, SQL e JAVA, permetterà la facile introduzione di nuove funzionalità.

#### **LEGGIBILITÀ**

Grazie ad appositi commenti al codice e documentazione, si potrà facilmente intervenire su di esso.

#### **MODIFICABILITÀ**

Chiaramente, il codice sarà modificabile, in maniera tale da migliorarlo e da correggere eventuali imperfezioni.

#### **TRACCIABILITÀ**

In tutte le fasi di sviluppo, fino alla codifica, si potrà risalire a requisiti funzionali per ogni funzione.

### **1.2.7 CRITERI DELL'UTENTE FINALE**

#### **USABILITY**

Il sistema dovrà essere user-friendly, cioè non richiede competenze specifiche. Le interfacce dovranno essere intuitive, in maniera tale da non confondere l'utente.

#### UTILITY

Grazie ad opportuni messaggi, l'utente sarà sempre avvisato degli errori.

### 3.3. OBIETTIVI DI PROGETTAZIONE:

ACRONIMO	DEFINIZIONE
DB	Database (sistema di memorizzazione per l'archiviazione dei dati permanenti)
DBMS	Database Management System (gestore del sistema di memorizzazione)
HTML	HyperText Markup Language (linguaggio di markup utilizzato per la definizione della struttura della pagina web)
HTTP	HyperTextTransferProtocol (Protocollo per la gestione di richieste e risposte scambiate tra client e server)
JAVA	Linguaggio orientato agli oggetti
JAVASCRIPT	Linguaggio di scripting utilizzato lato client per rendere il portale web dinamico
Layout	Impaginazione struttura grafica del portale e dell'applicazione
MYSQL	DBMS utilizzato
SQL	Structured Query Language (linguaggio per l'invocazione delle richieste al database)
Query	Interrogazione al database
RAD	Requirements Analysis Document

SDD	System Design Document
-----	------------------------

### 3.4. RIFERIMENTI:

Per realizzare il progetto, sono stati utilizzati:

- slide del docente;
- B. BRUEGGE, A.H. DUTOIT, OBJECT ORIENTED SOFTWARE ENGINEERING – USING UML, PATTERNS AND JAVA, PRENTICE HALL, 3D EDITION, 2009;
- RAD di EasyBook.

### 3.5. PANORAMICA:

Nella prima parte del documento vengono esposti gli obiettivi di progettazione.

Al secondo e al terzo punto del documento vengono espresse l'architettura software corrente e quella da noi proposta.

Poi, viene mostrata la decomposizione del sistema in sottosistemi. Ad ogni sottosistema è associata una funzione. Sono analizzati i dati persistenti necessari e l'approccio alla loro gestione. Sono individuati le categorie di utenti che useranno il sistema e viene formalizzata la politica di accesso. Vengono formalizzate le condizioni di eccezione e i casi limite.

## 4. ARCHITETTURA SOFTWARE CORRENTE:

Un sito concorrente è il sito della rete bibliotecaria bergamasca <http://www.rbbg.it/> che permette la consultazione del proprio catalogo e la prenotazione di libri e altri contenuti dopo essersi iscritti al sito.

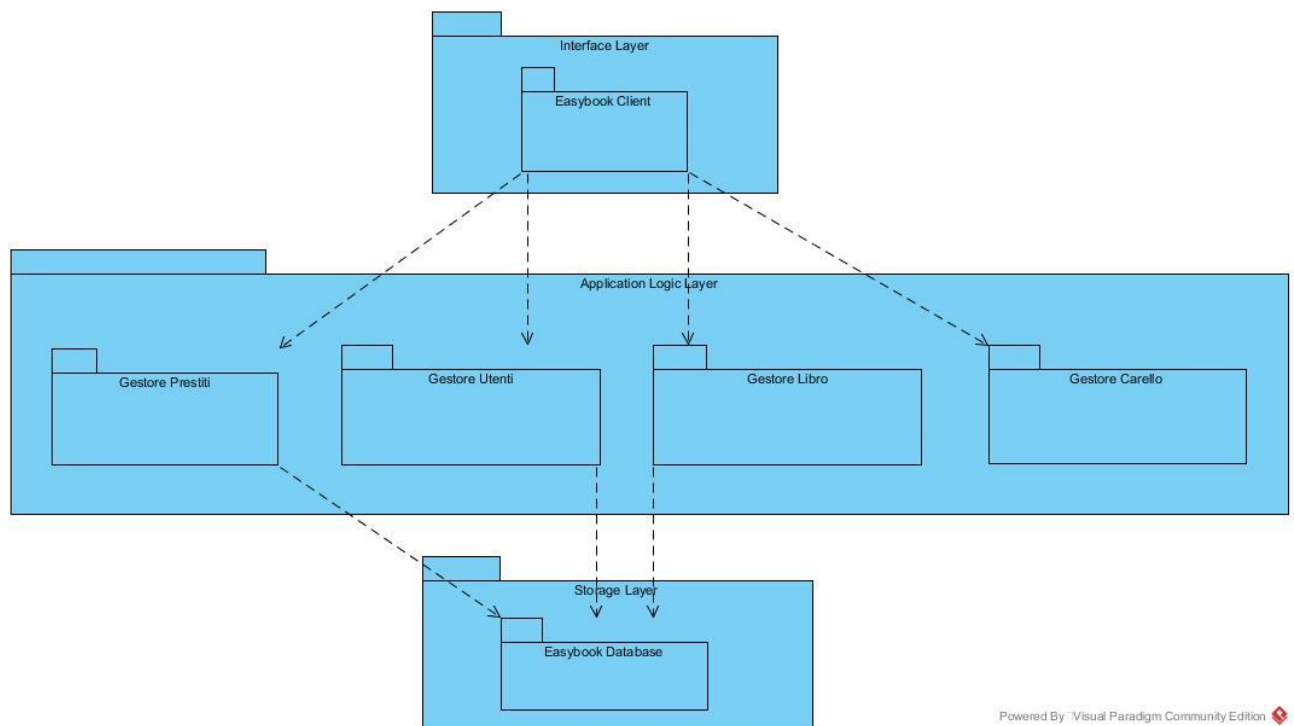
## 5. ARCHITETTURA SOFTWARE PROPOSTA:

## 5.1 PANORAMICA

*Easybook è un sito web e, come tale, si basa su di un'architettura client server in cui il client è il browser dell'utente: il client chiede l'erogazione di servizi al server che provvederà a fornire una risposta. Per memorizzare i dati persistenti è stato scelto di utilizzare un database relazionale. Si è preferito l'utilizzo di un DBMS per garantire la consistenza dei dati e un rapido accesso agli stessi. Trattandosi di un applicazione web abbiamo optato per un'architettura three-tier: vengono cioè individuati i 3 livelli di presentazione delle informazioni, logica dell'applicazione e sistema di memorizzazione.*

## 5.2 DECOMPOSIZIONE SUBSYSTEM

Essendo questa un'applicazione web client server, si individuano 3 layer per la gestione del servizio:



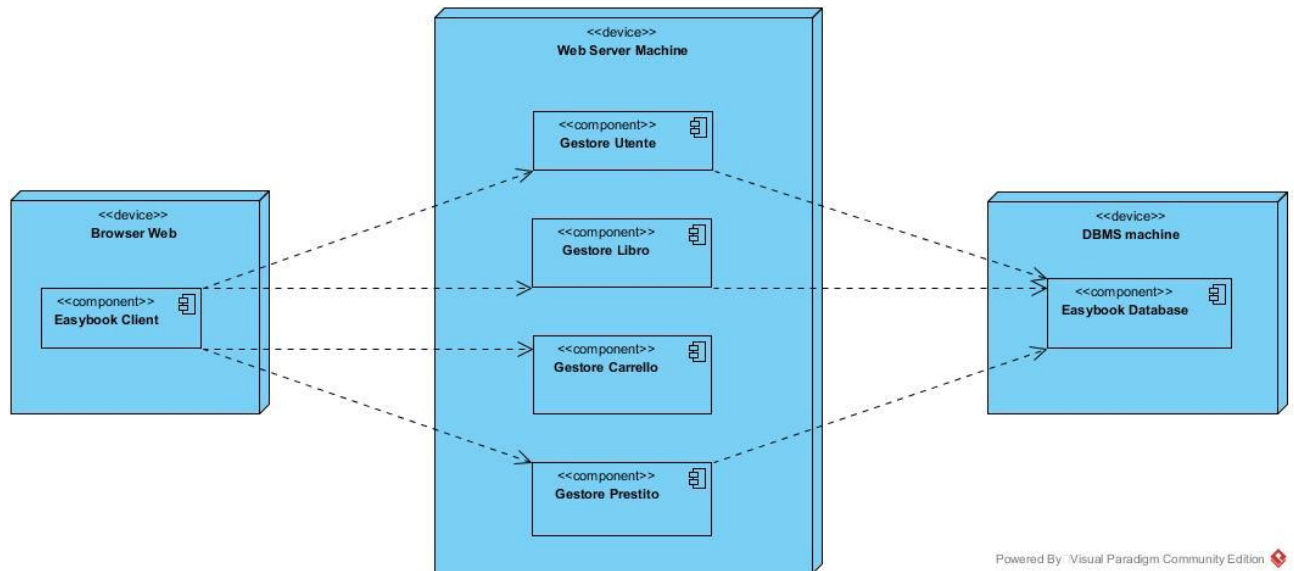
Si individuano i sottosistemi di

- "Gestore Utente (Registrazione Utente, Login , Logout , Cancellazione profilo utente)",
- "Gestore Prodotto (Visualizzazione catalogo, Ricerca libri, aggiunta libri al catalogo, rimozione libri catalogo)",
- "Gestore Carrello (Visualizza carrello , Aggiunta di un prodotto al carrello, Rimozione dei prodotti dal carrello)",
- "Gestore Prestito (registrazione prestito, visualizzazione prestiti, chiusura prestiti)",

## 5.3 HARDWARE / SOFTWARE DI MAPPATURA

Easybook è un'applicazione web. Per questa sua natura si possono distinguere due tipologie di macchine coinvolte nella sua esecuzione: una macchina server, garante della logica del sito e

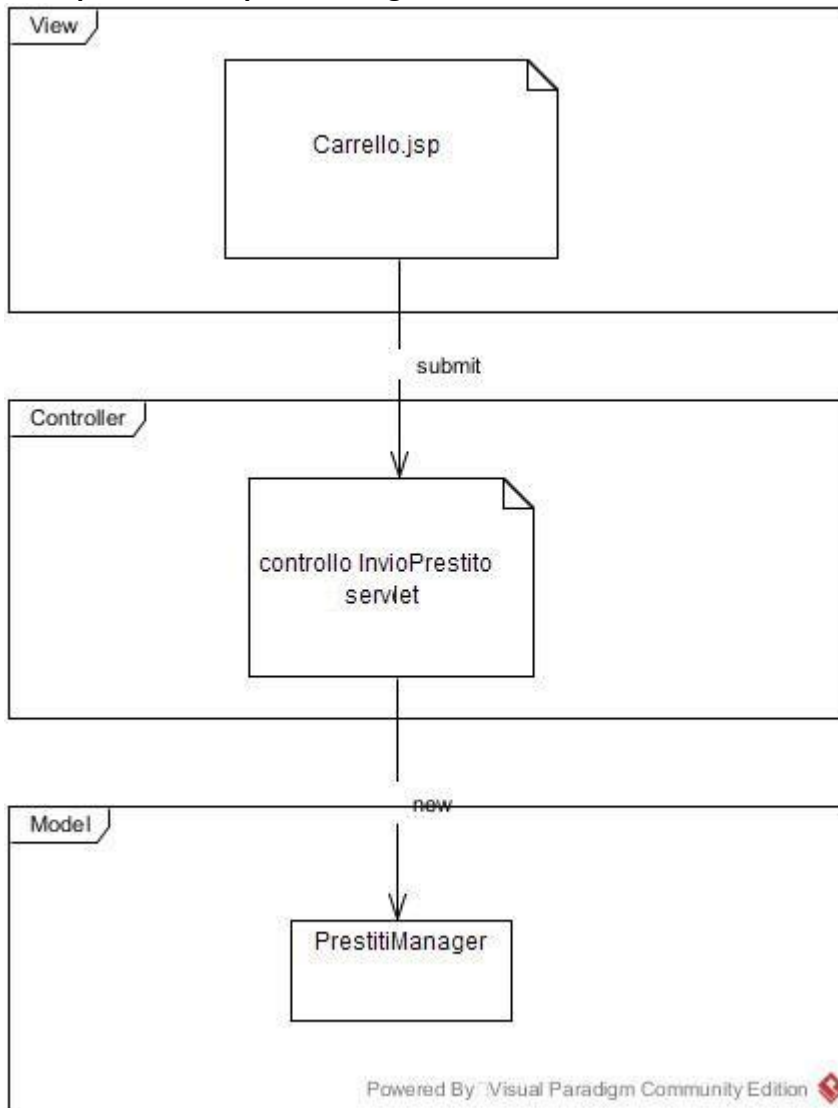
responsabile dell'accesso al database e di tutte le operazioni, e più macchine client, adibite alla visualizzazione dei contenuti e all'invio di "richieste" al server. Il client usa un browser per visualizzare i contenuti offerti dal server tramite protocollo http. Per la memorizzazione dei dati si sfrutta un DBMS (in particolare MySQL) relazionale e il linguaggio SQL.



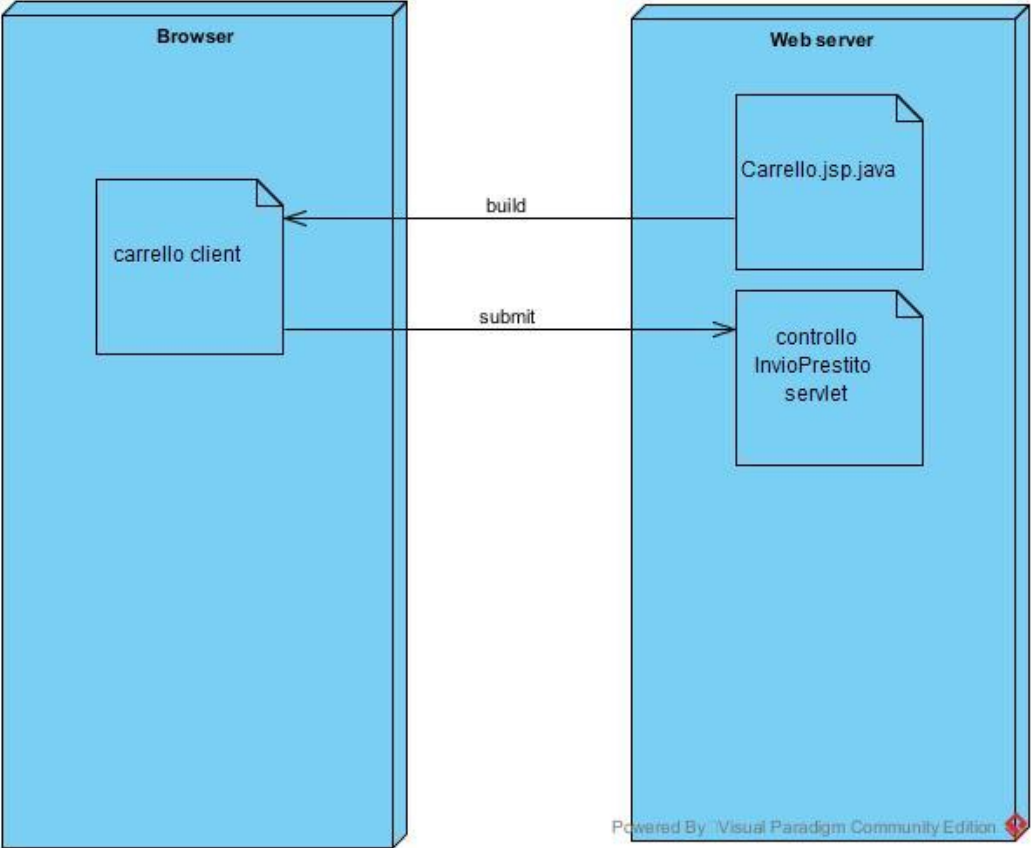


## 5.4 COMPONENT/DEPLOYMENT DIAGRAM

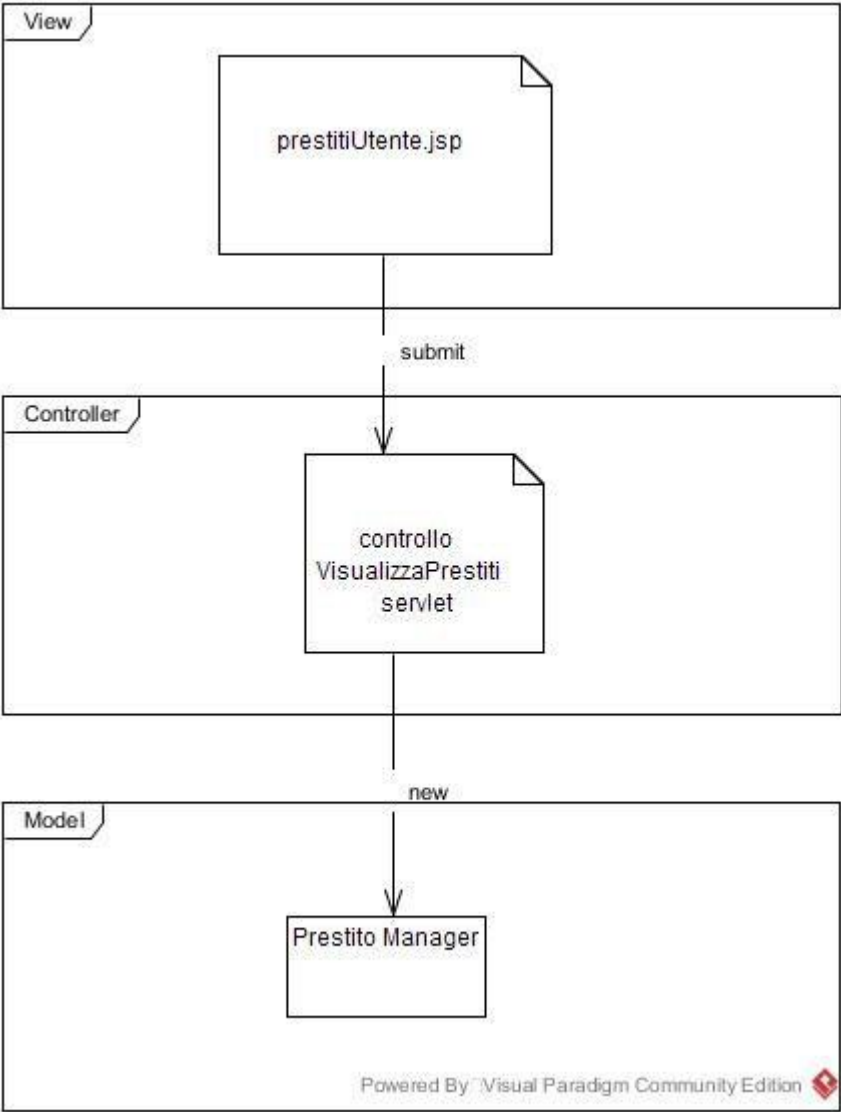
### Invio prestito component diagram



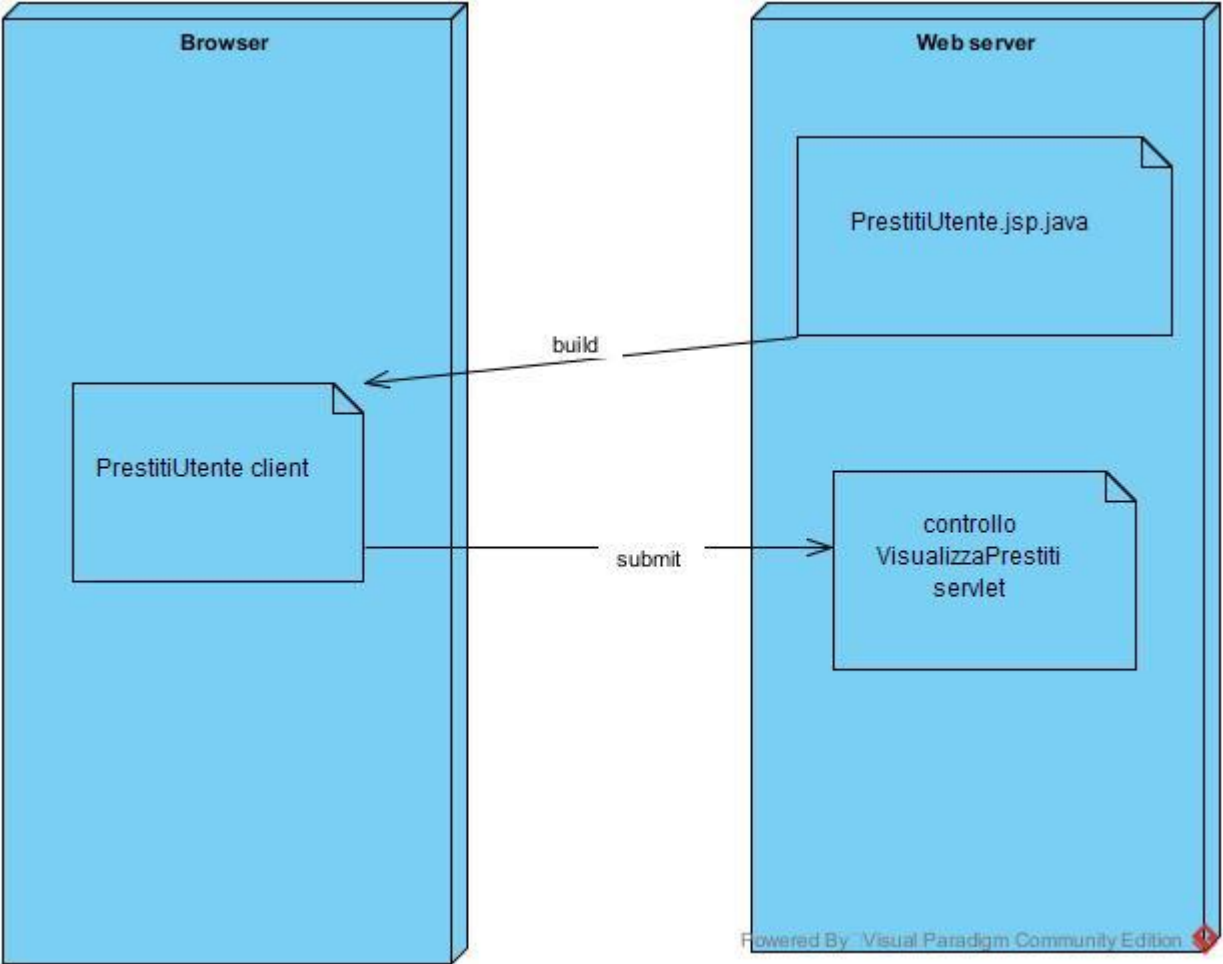
**Invio prestito deployment diagram**



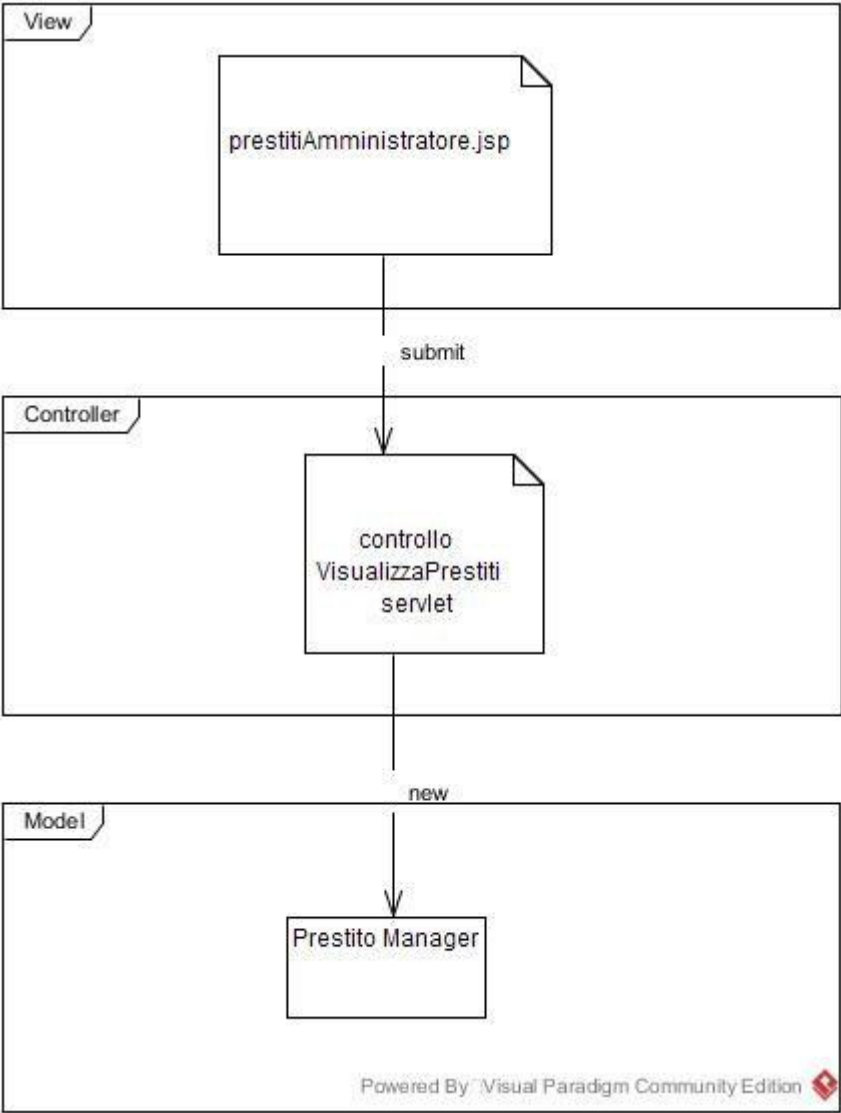
Visualizza prestiti utente component diagram



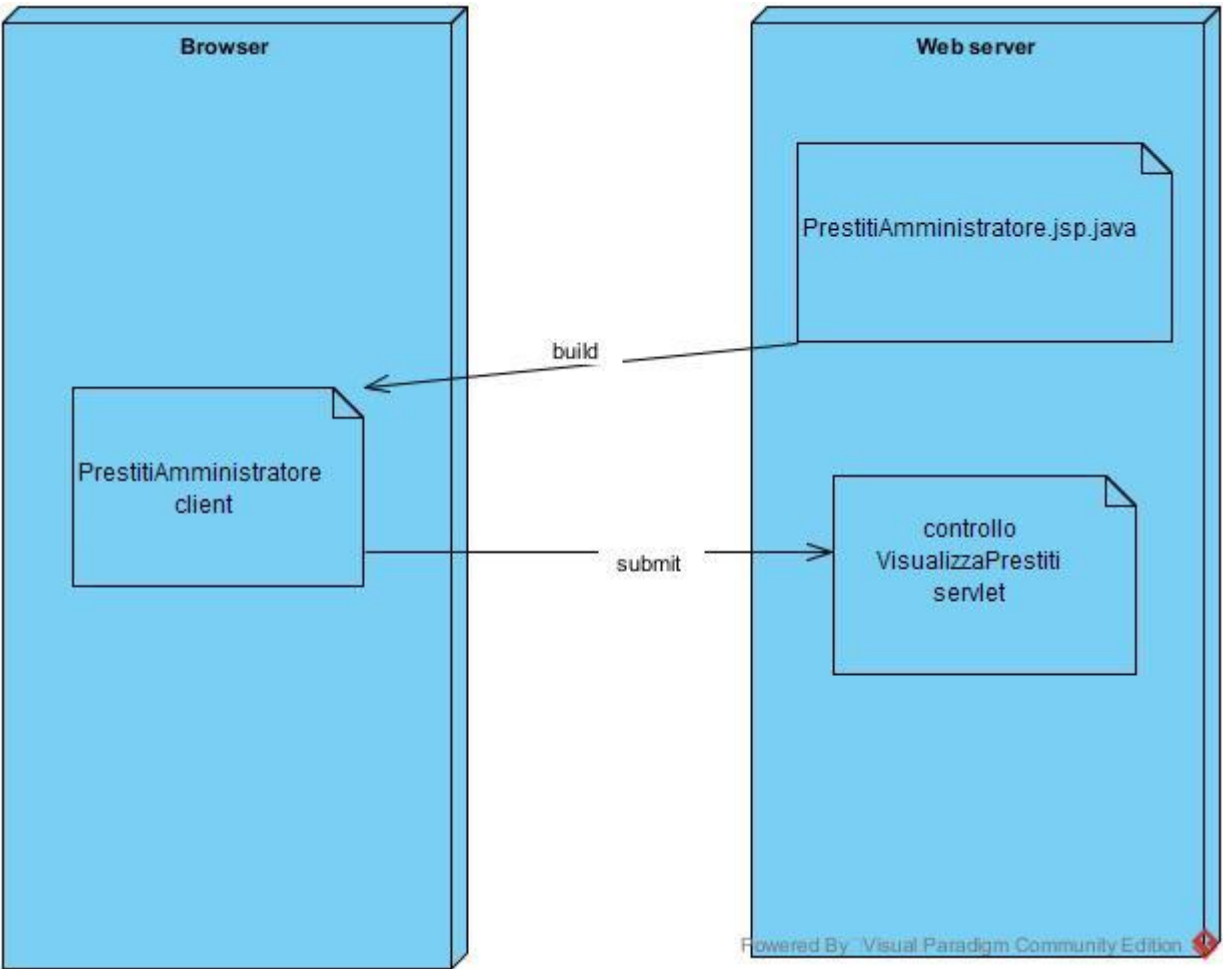
Visualizza prestiti utente deployment diagram



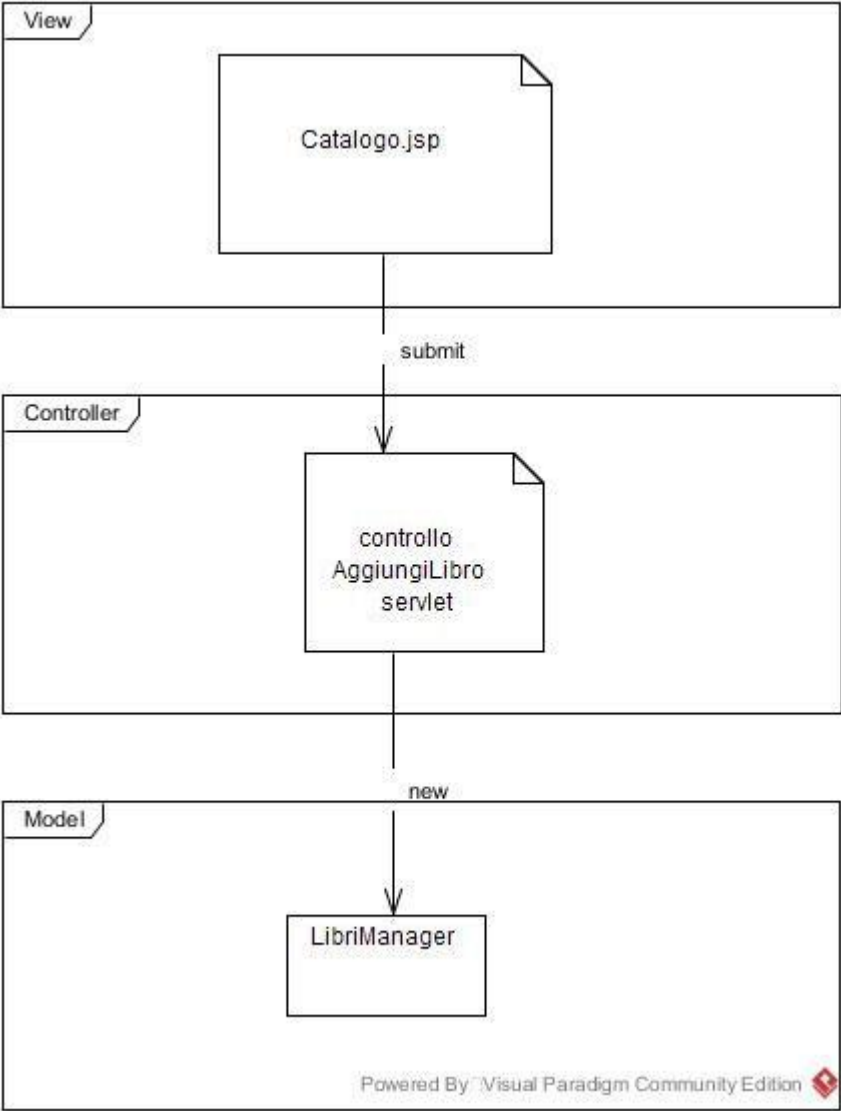
Visualizza prestiti amministratore component diagram



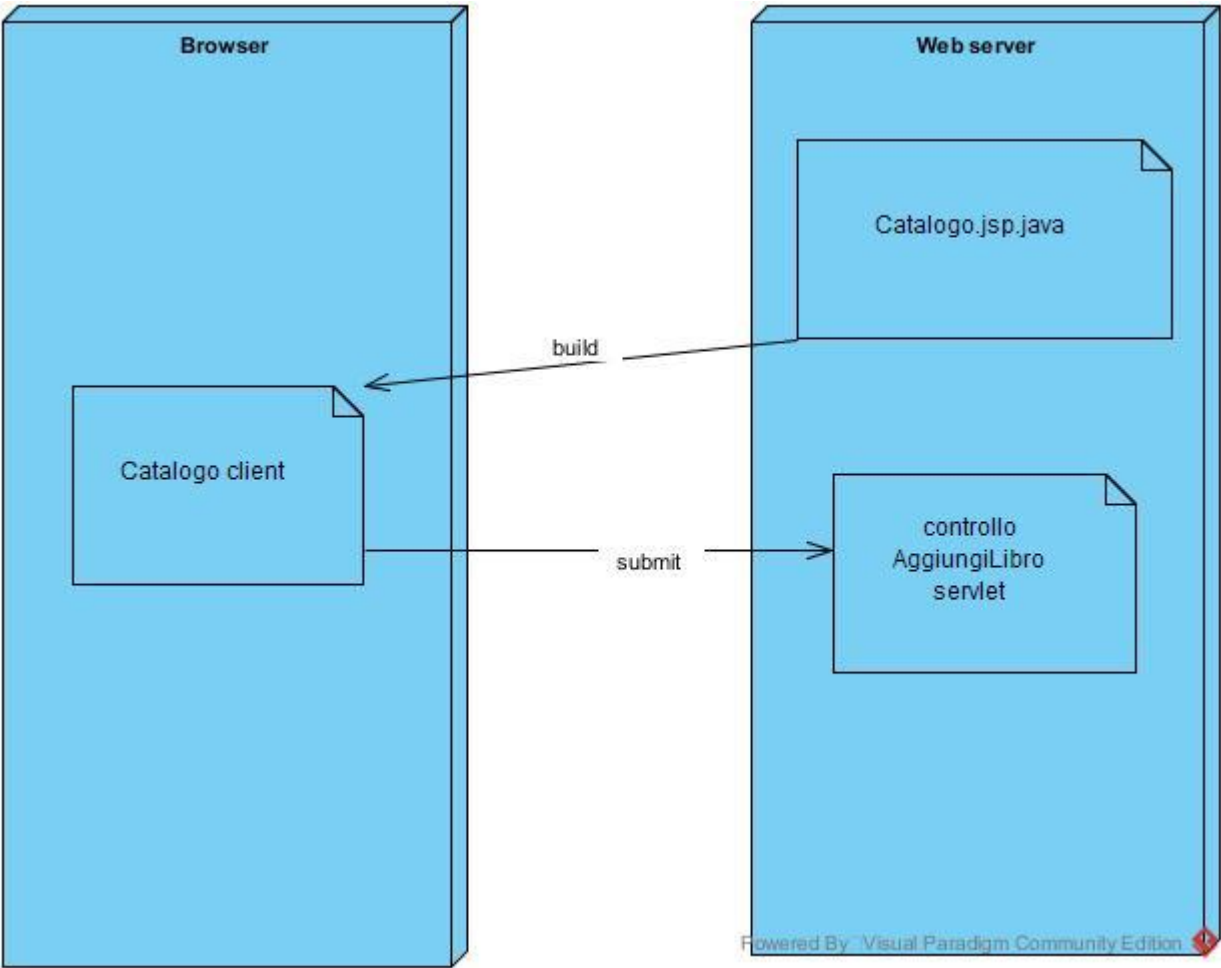
Visualizza prestiti amministratore deployment diagram



Aggiungi Libro al catalogo component diagram

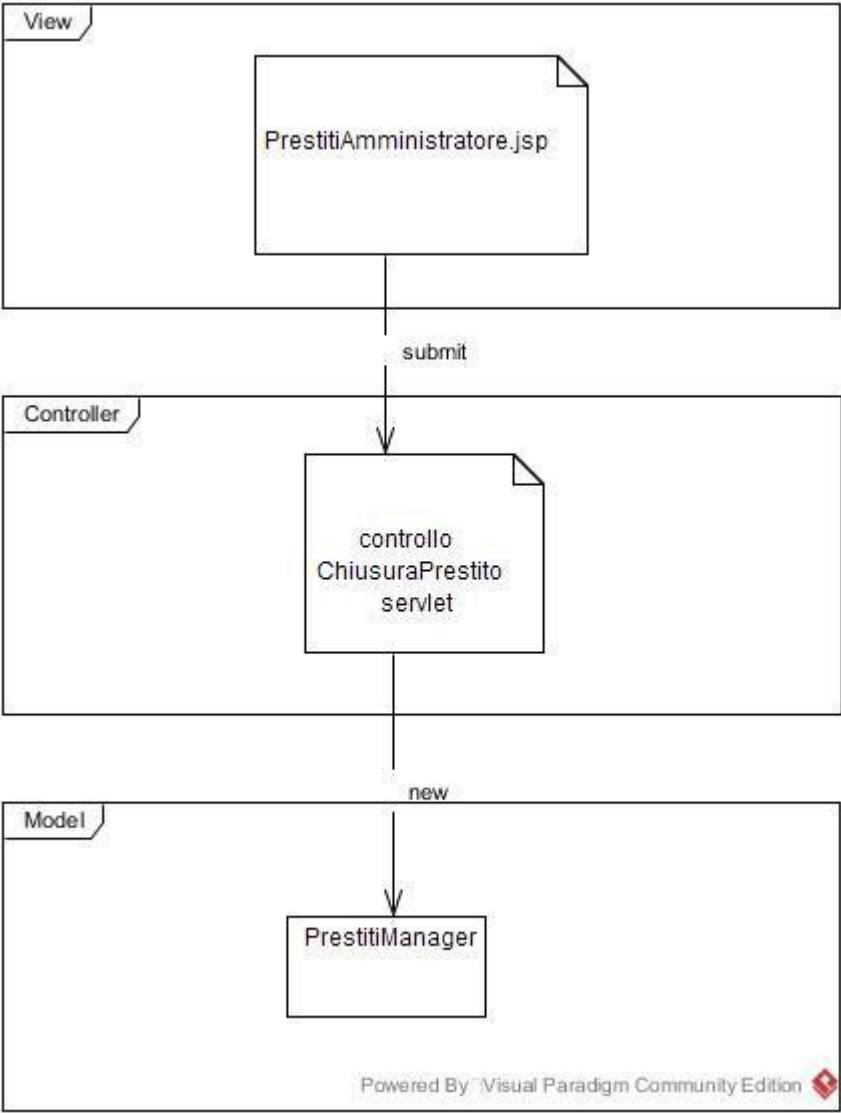


Aggiungi Libro al catalogo deployment diagram

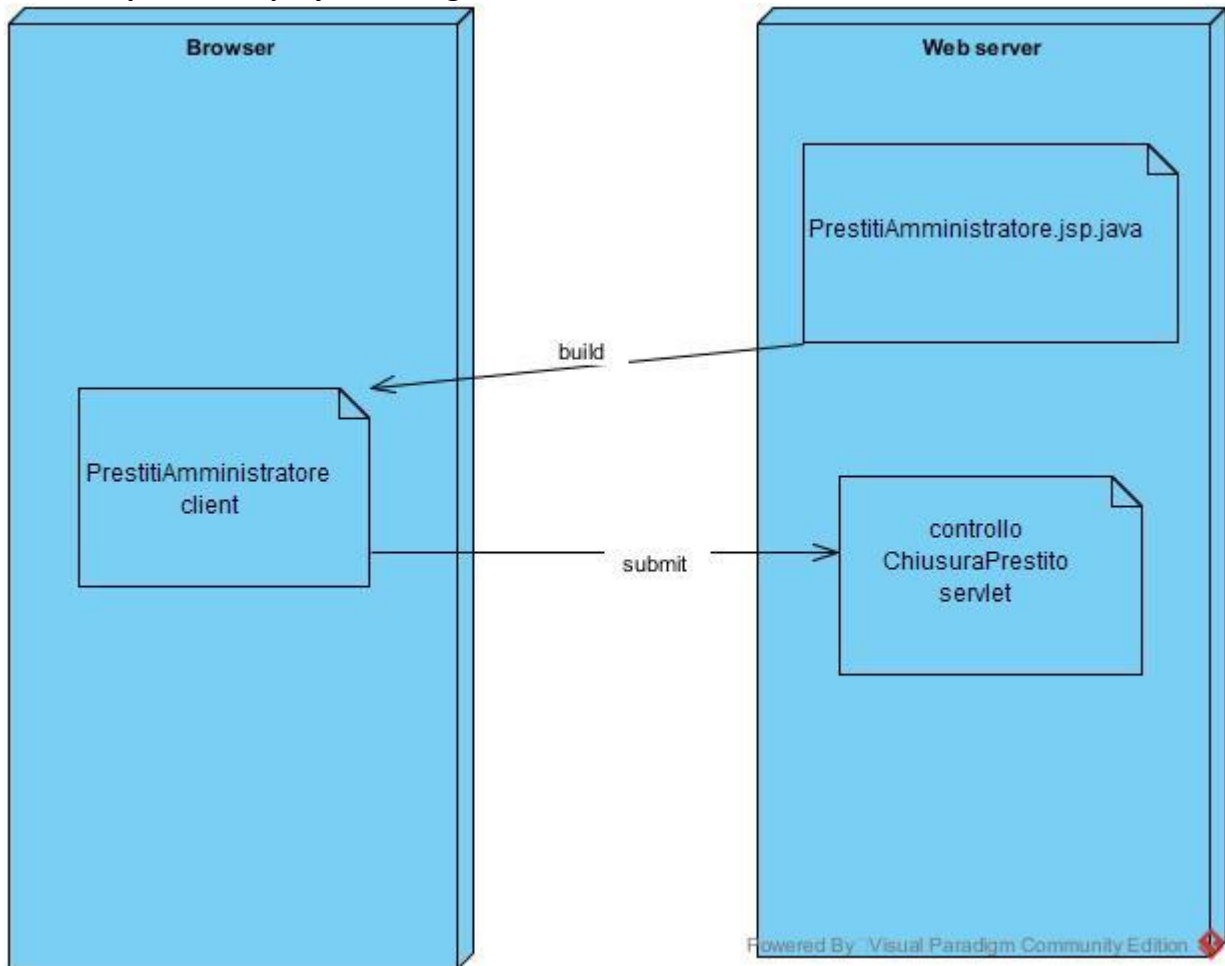




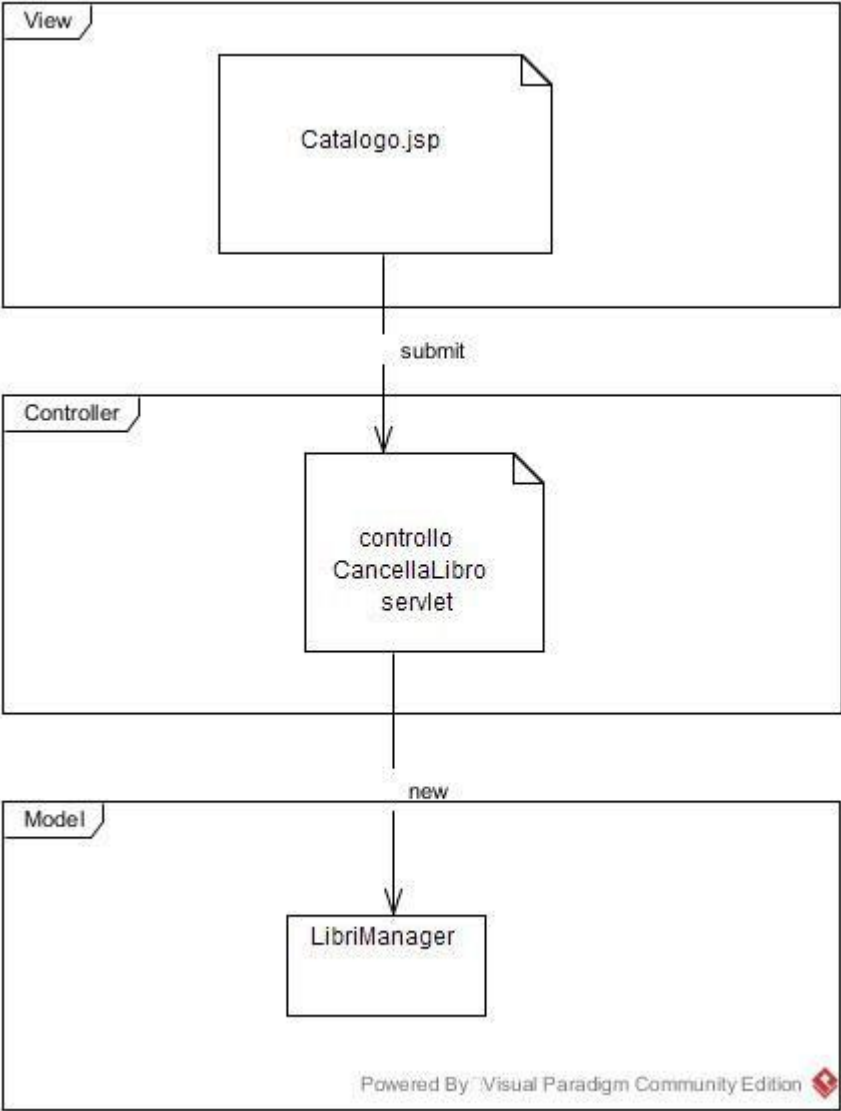
Chiusura prestito component diagram



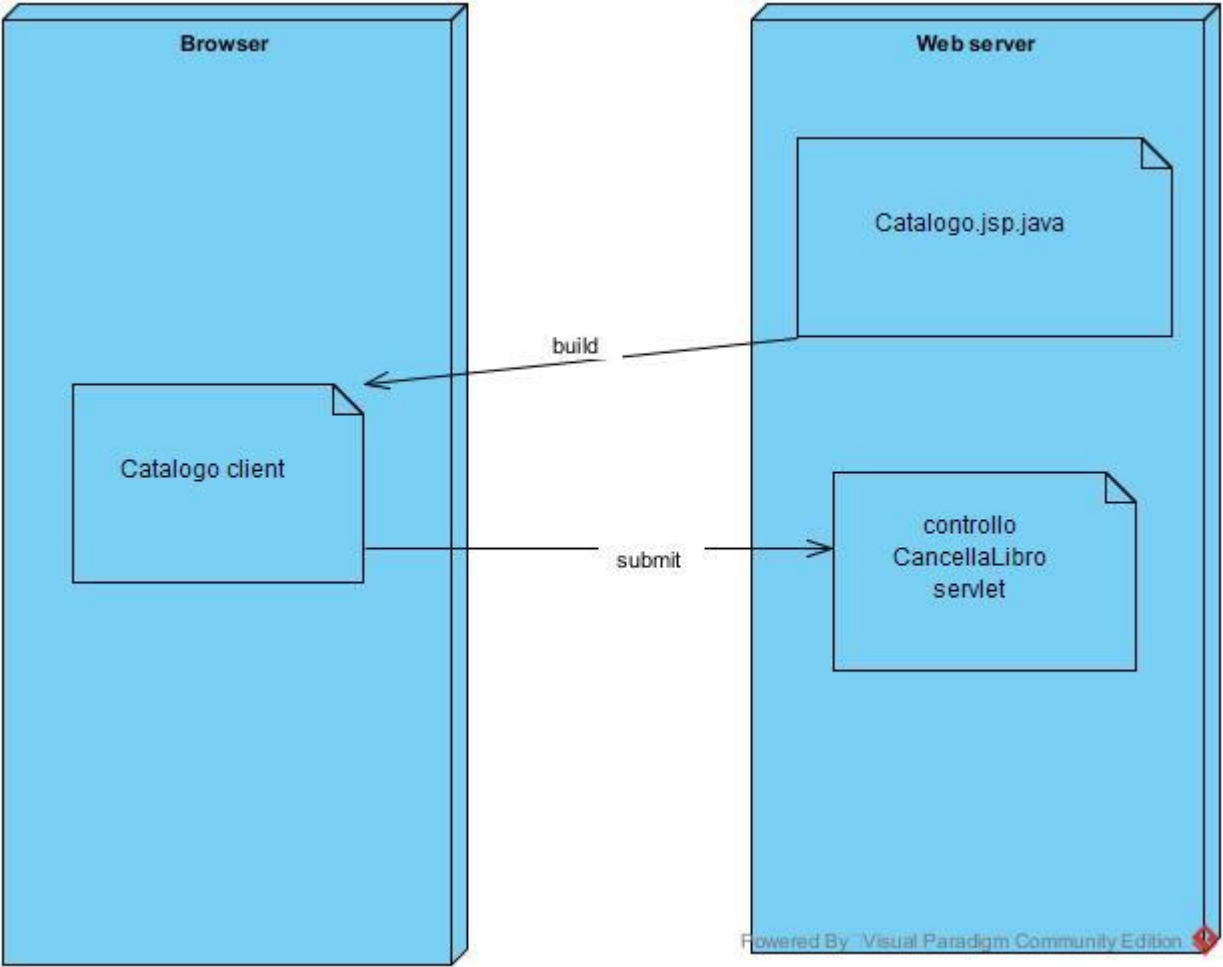
### Chiusura prestito deployment diagram



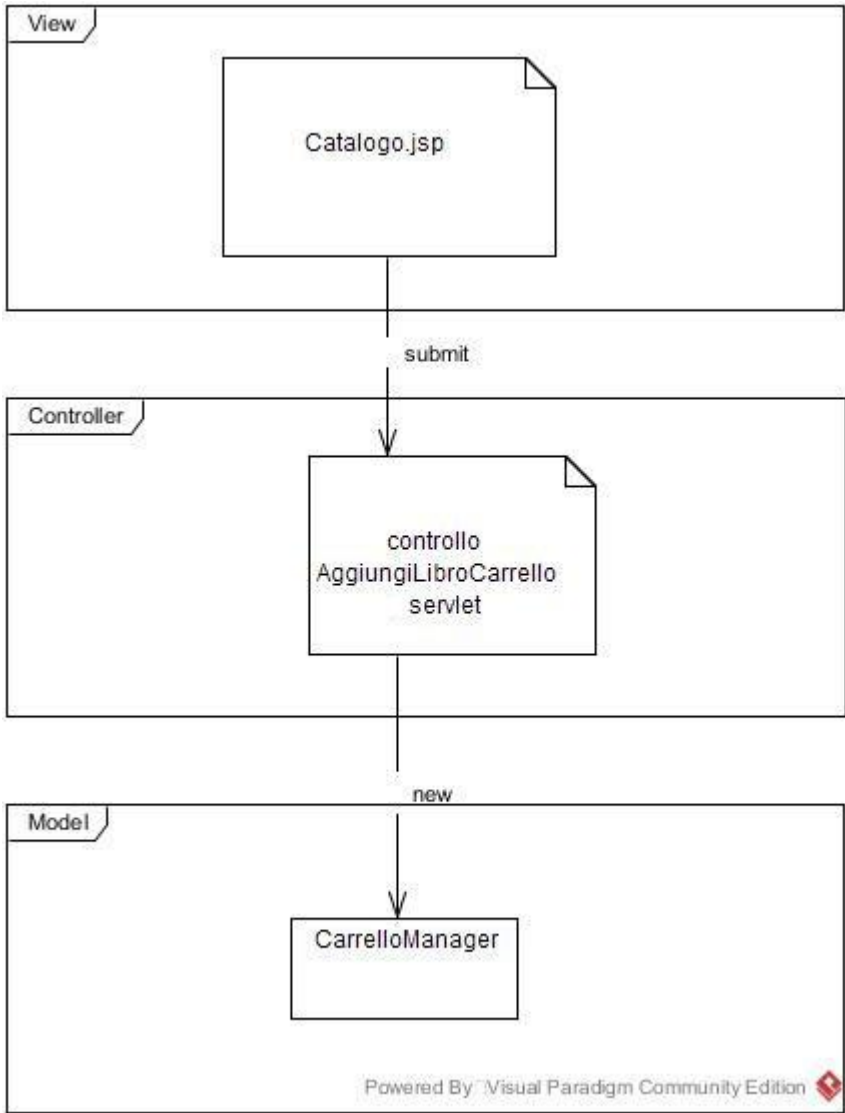
Cancella Libro dal catalogo component diagram



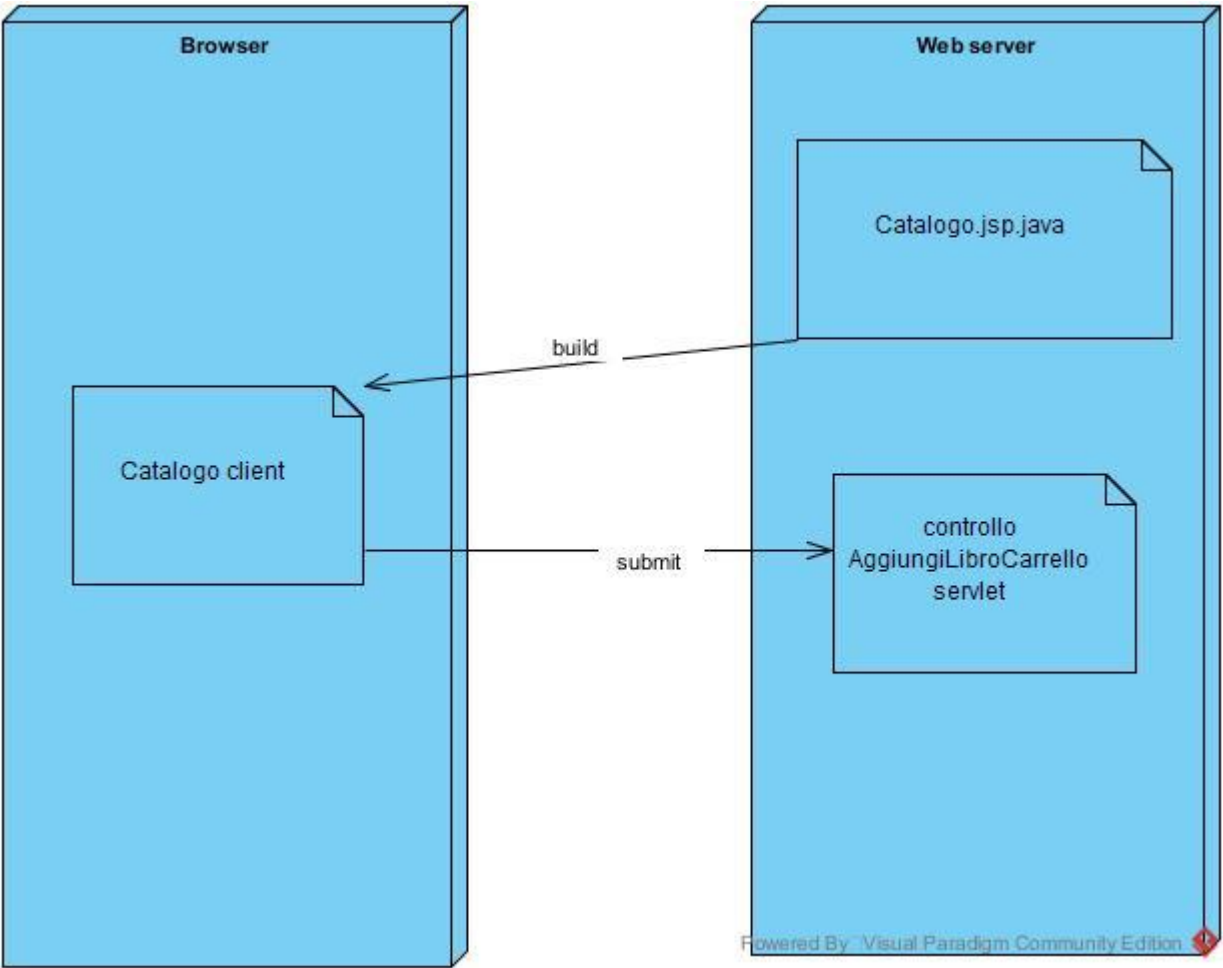
Cancella Libro dal catalogo deployment diagram



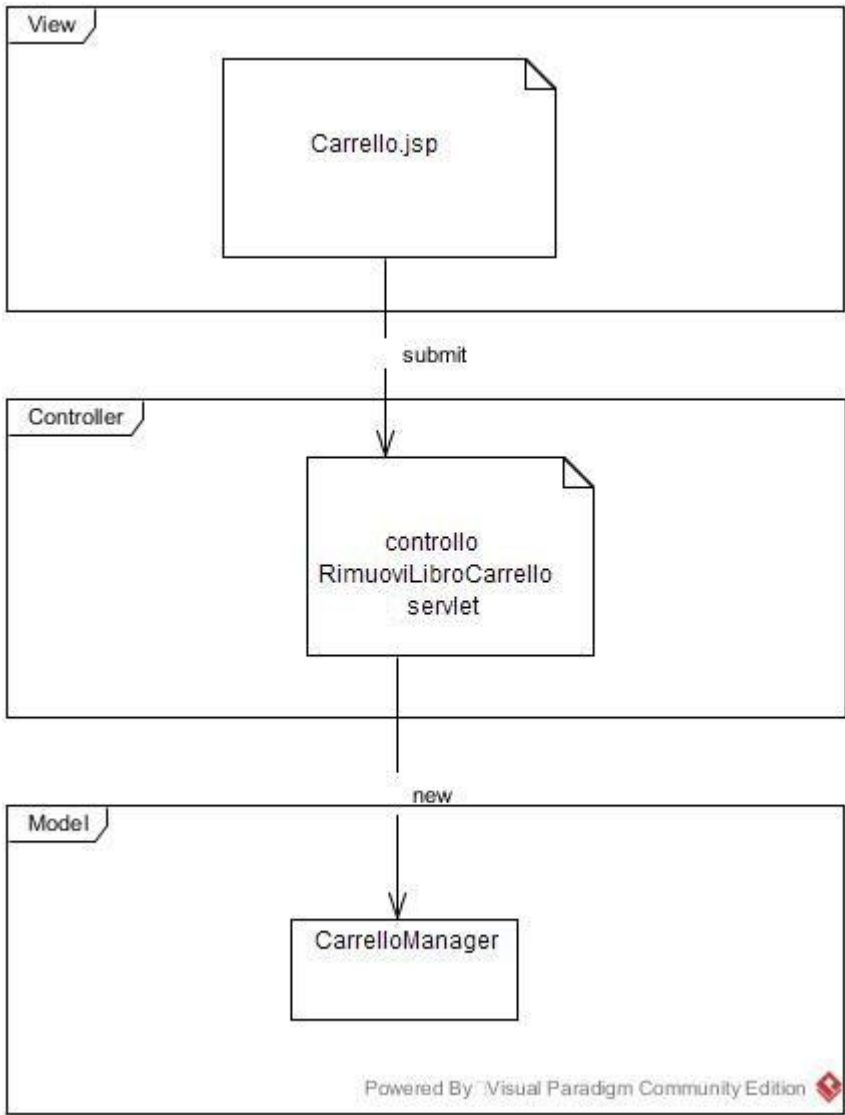
Aggiungi Libro al carrello component diagram



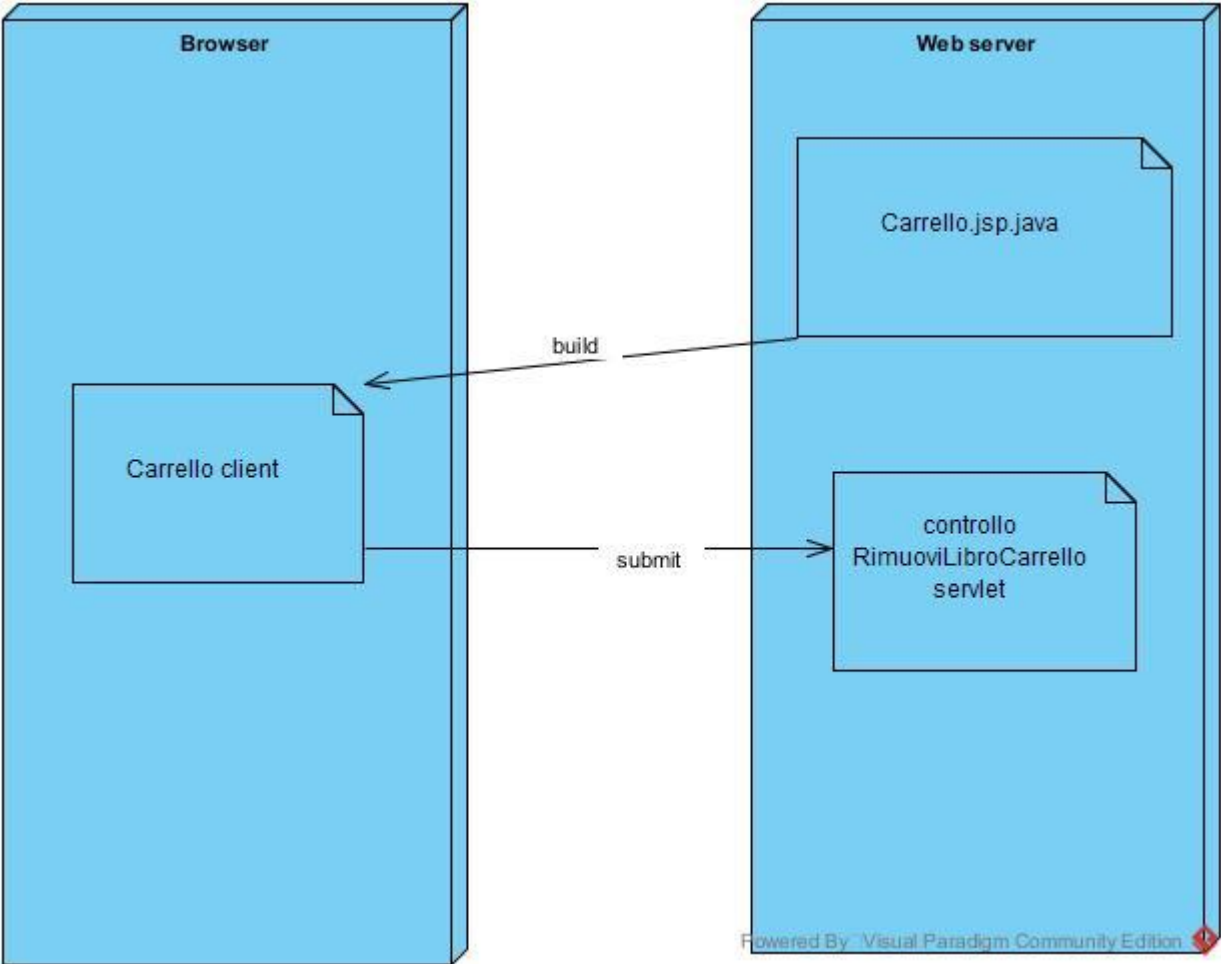
Aggiungi Libro al carrello deployment diagram



Rimuovi Libro dal carrello component diagram

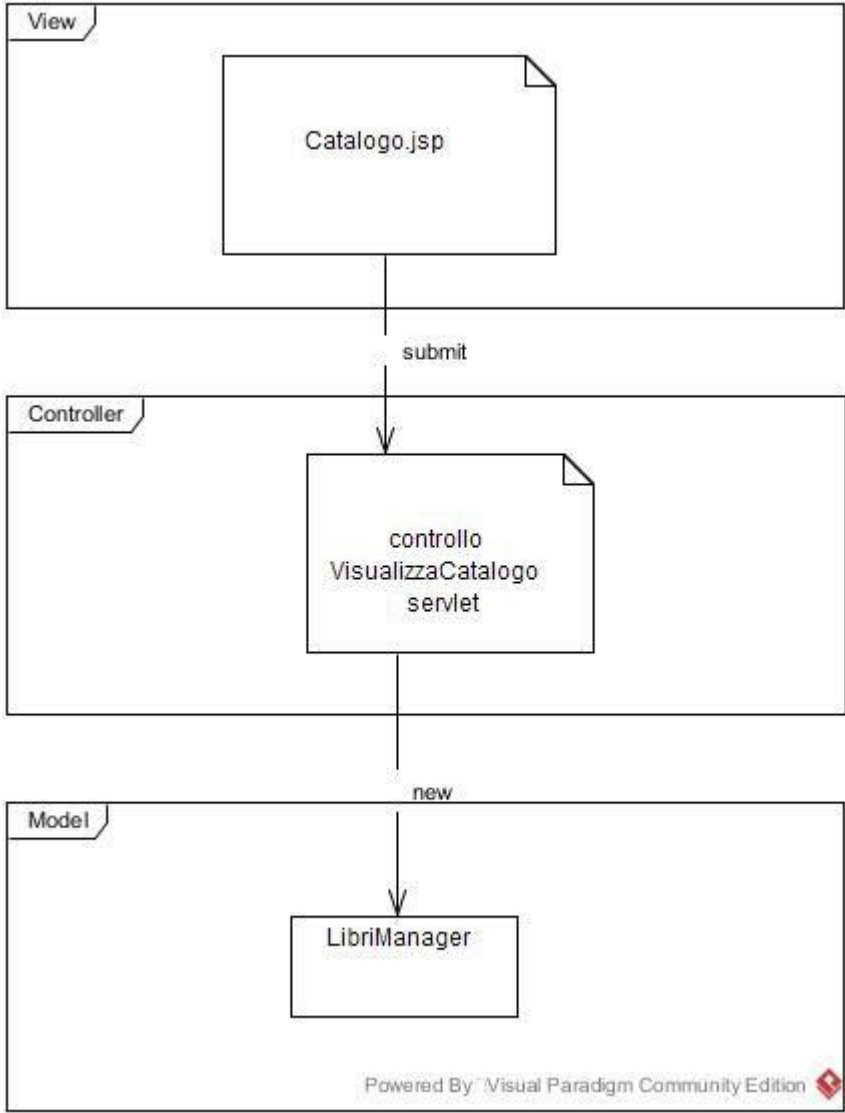


Rimuovi Libro dal carrello deployment diagram

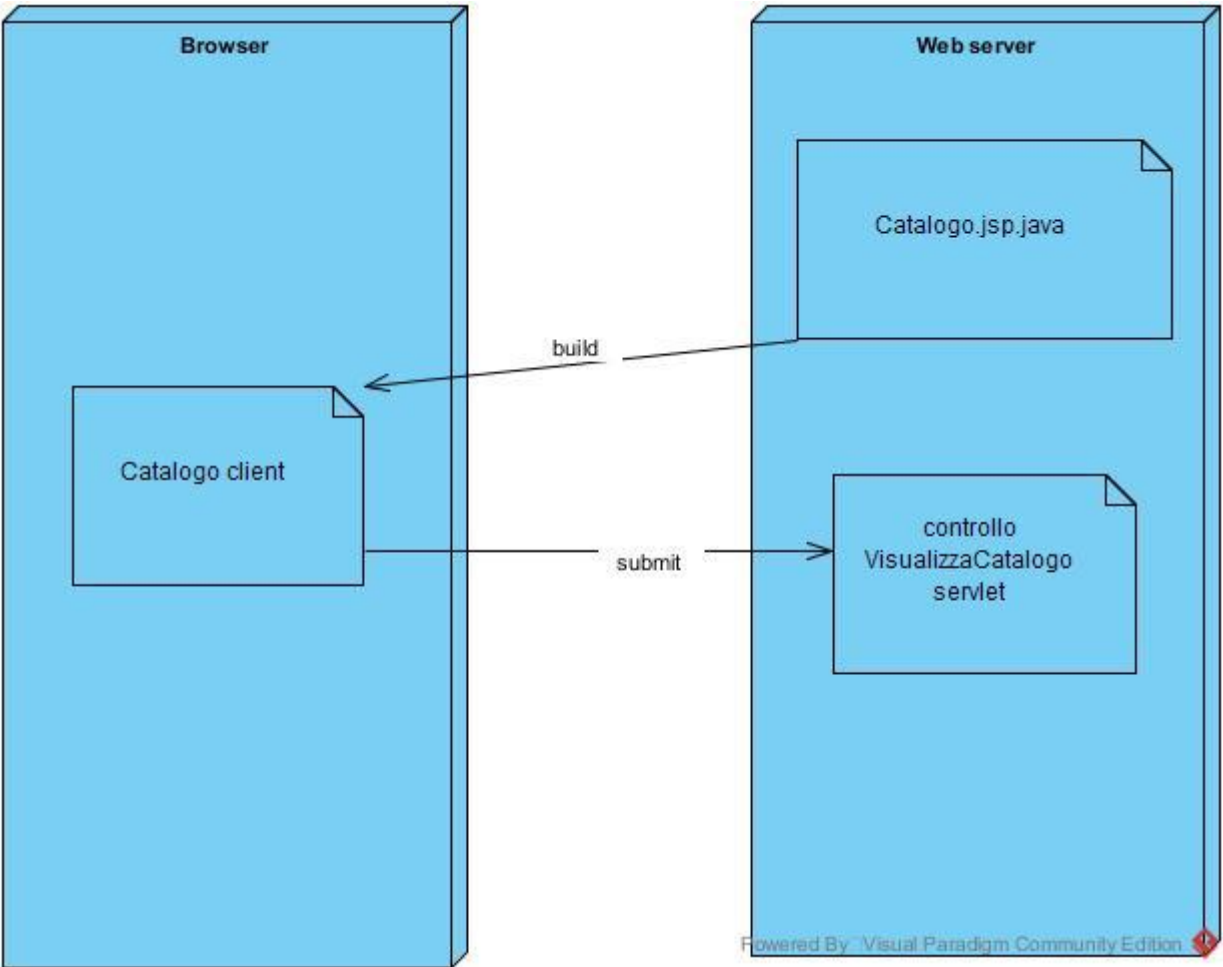




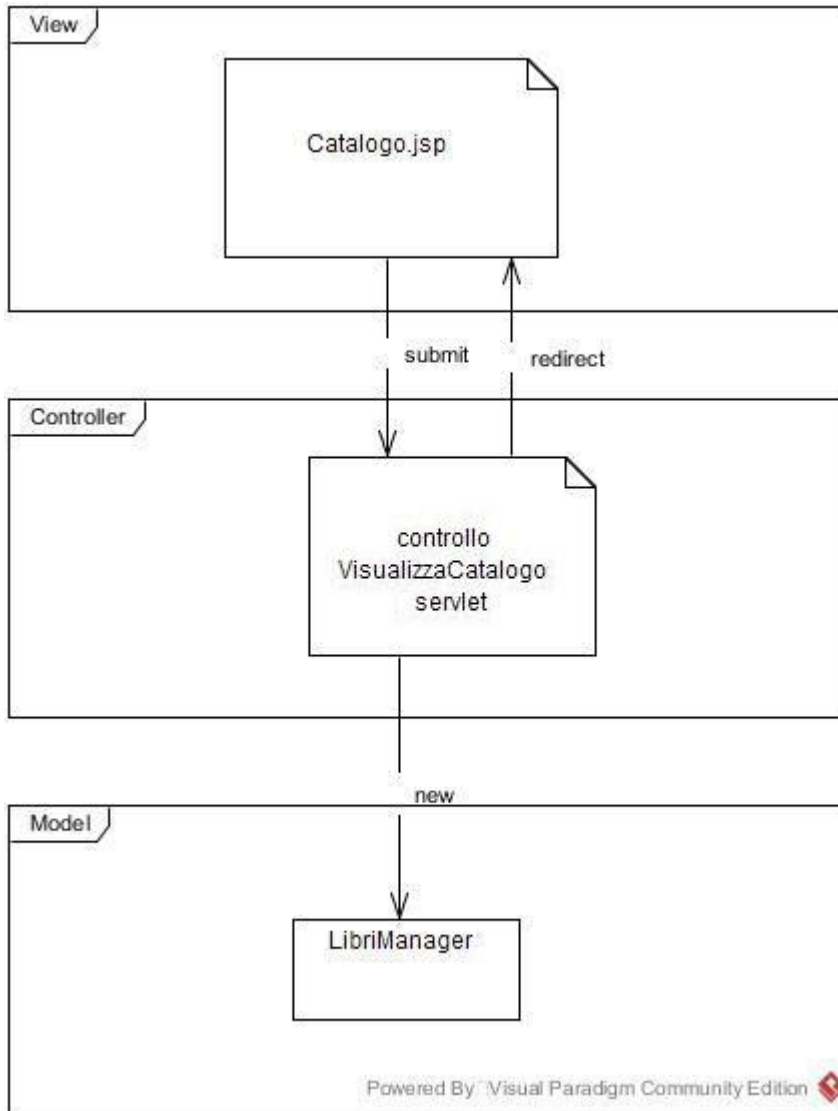
Visualizza catalogo component diagram



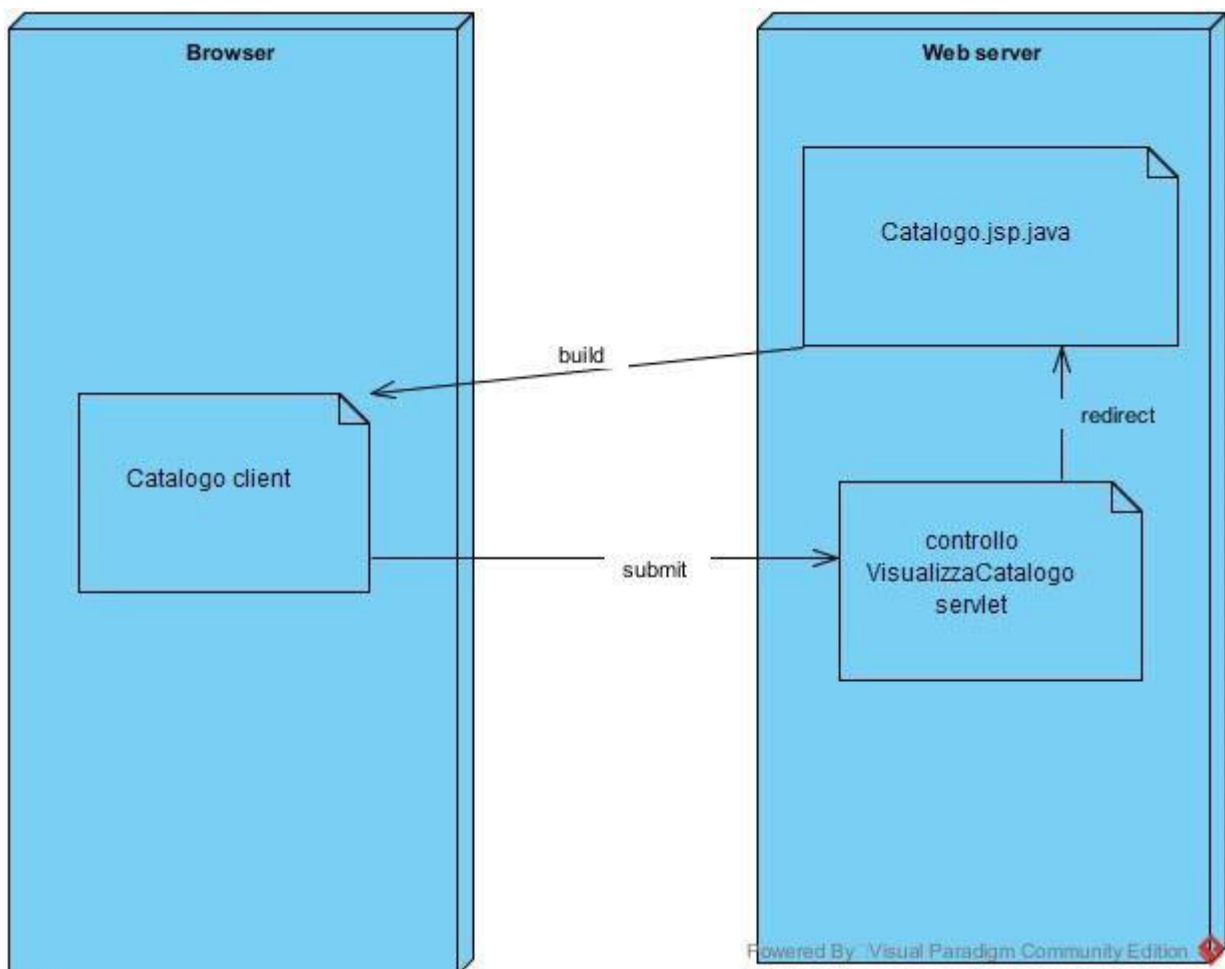
Visualizza catalogo deployment diagram



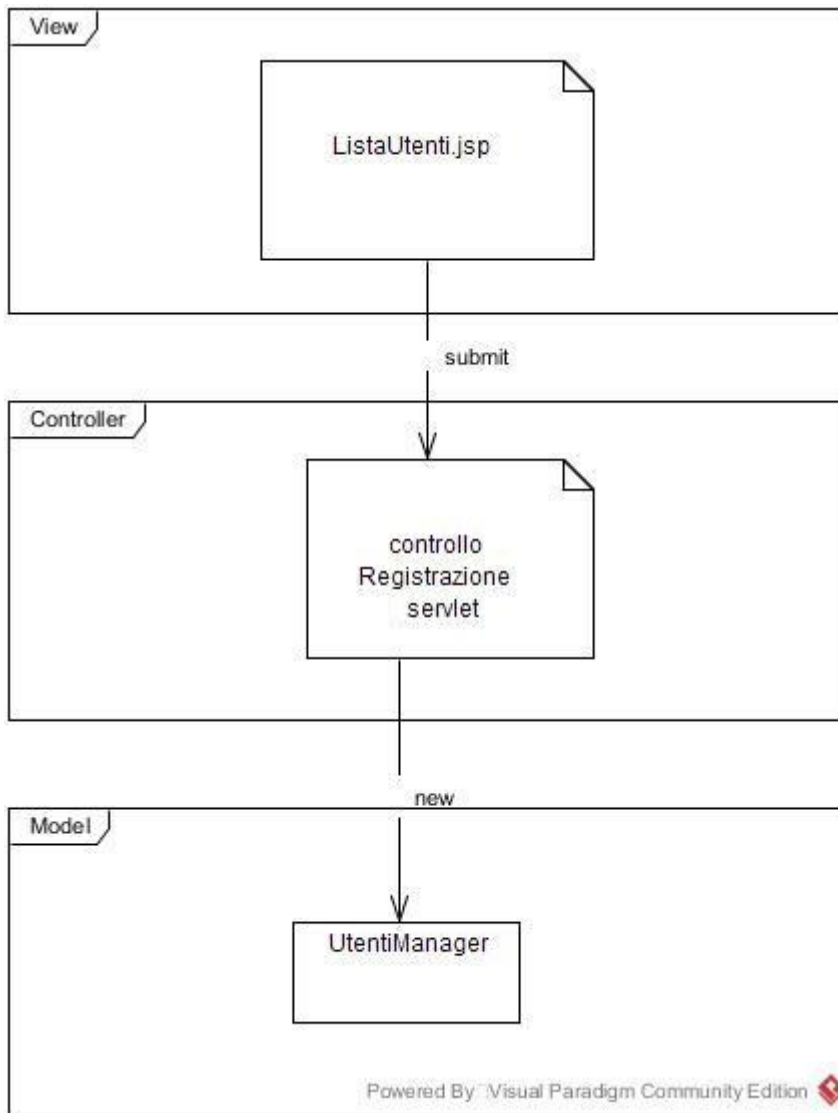
### Ricerca libro catalogo component diagram



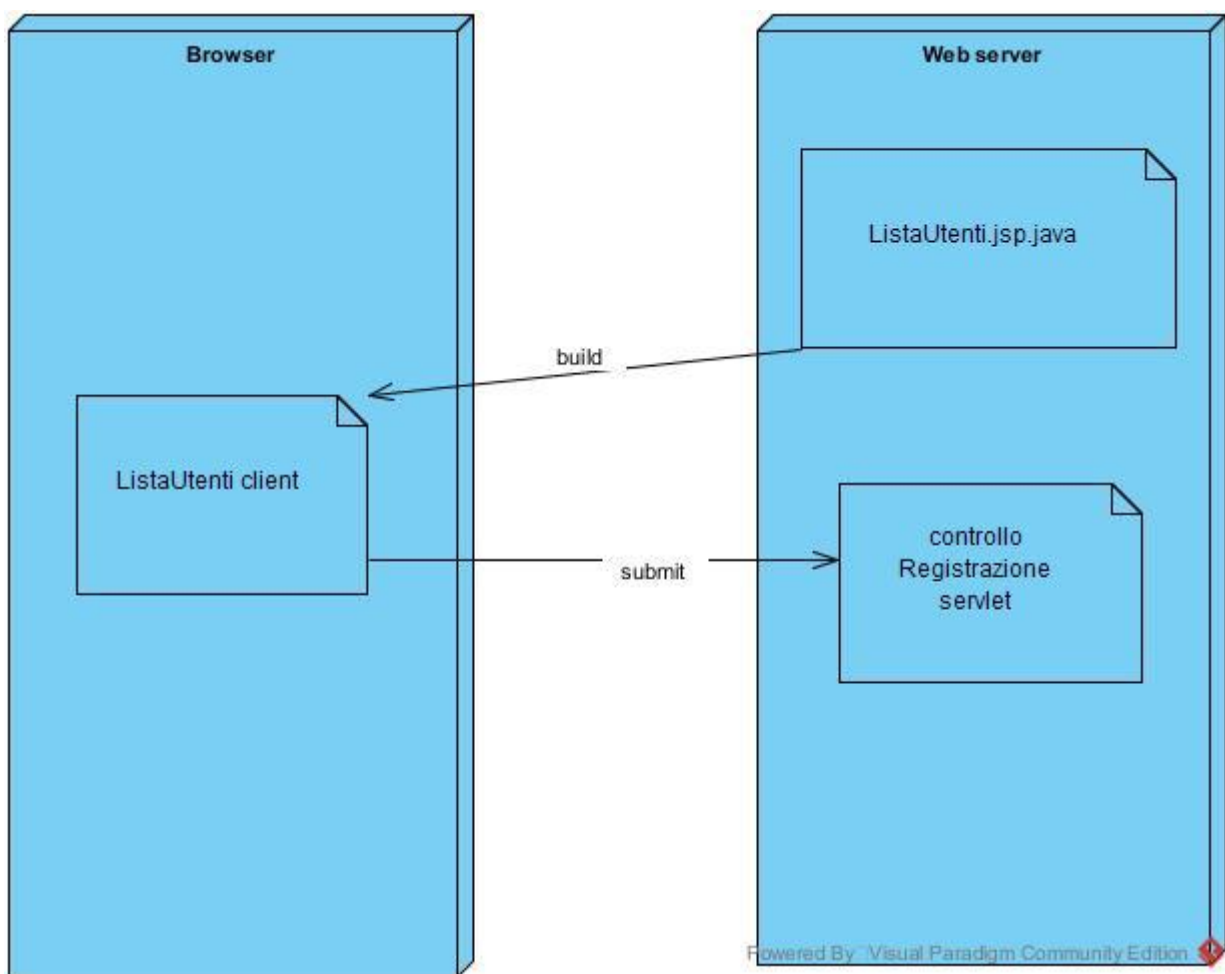
### Ricerca libro catalogo deployment diagram



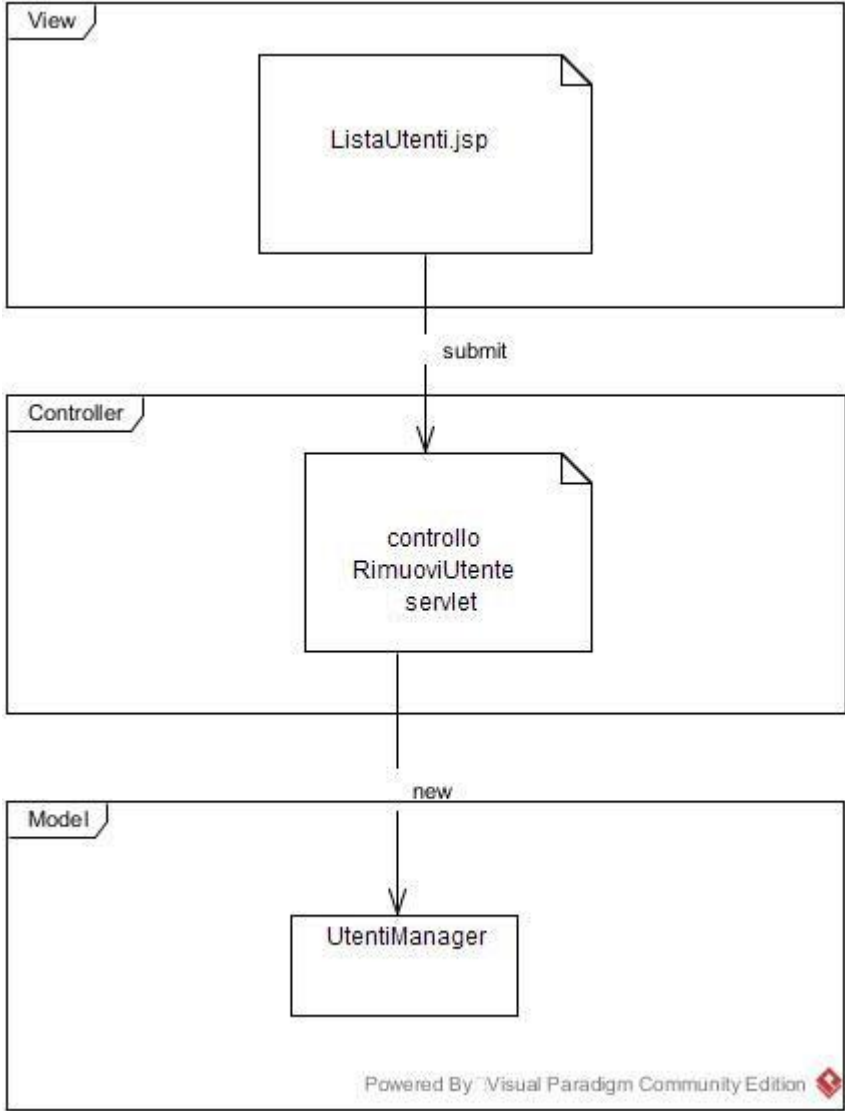
## Registrazione nuovo utente component diagram



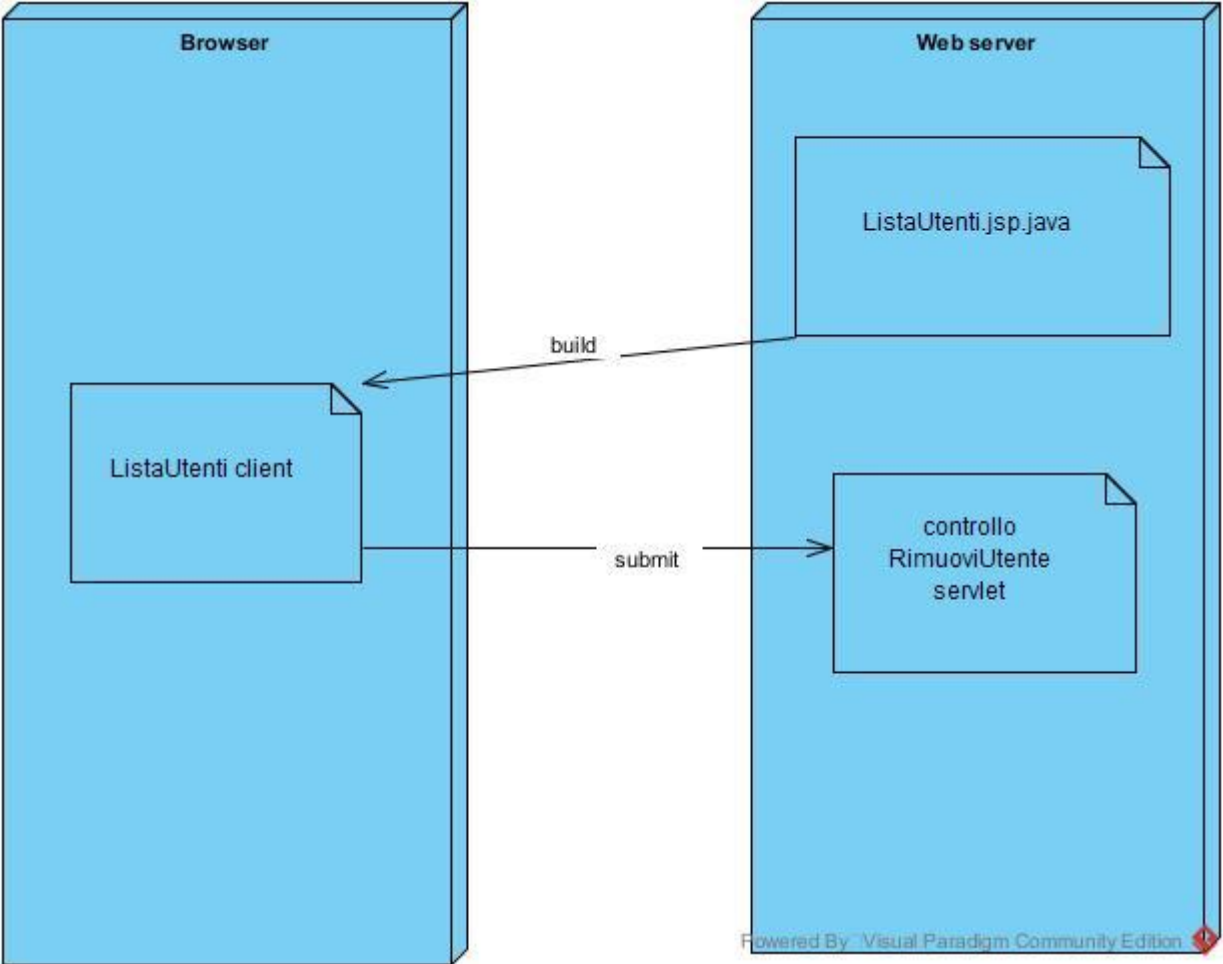
### Registrazione nuovo utente deployment diagram



Cancellazione utente component diagram

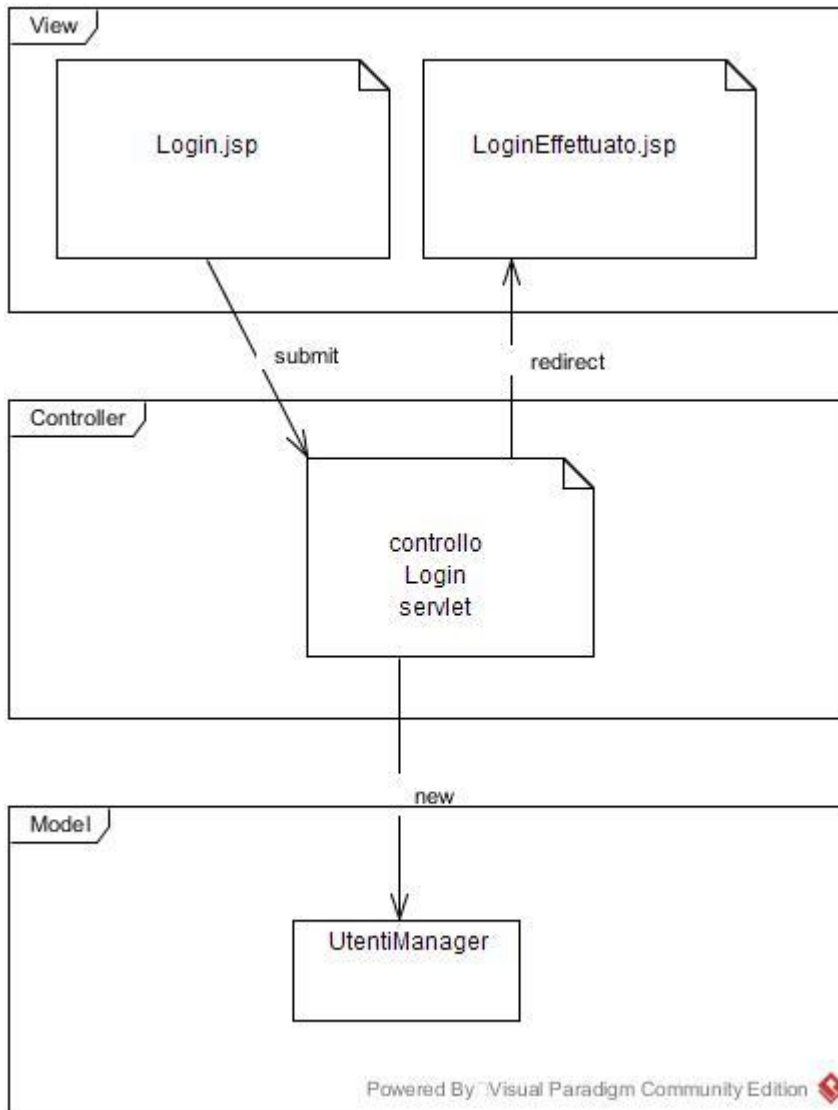


Cancellazione utente deployment diagram

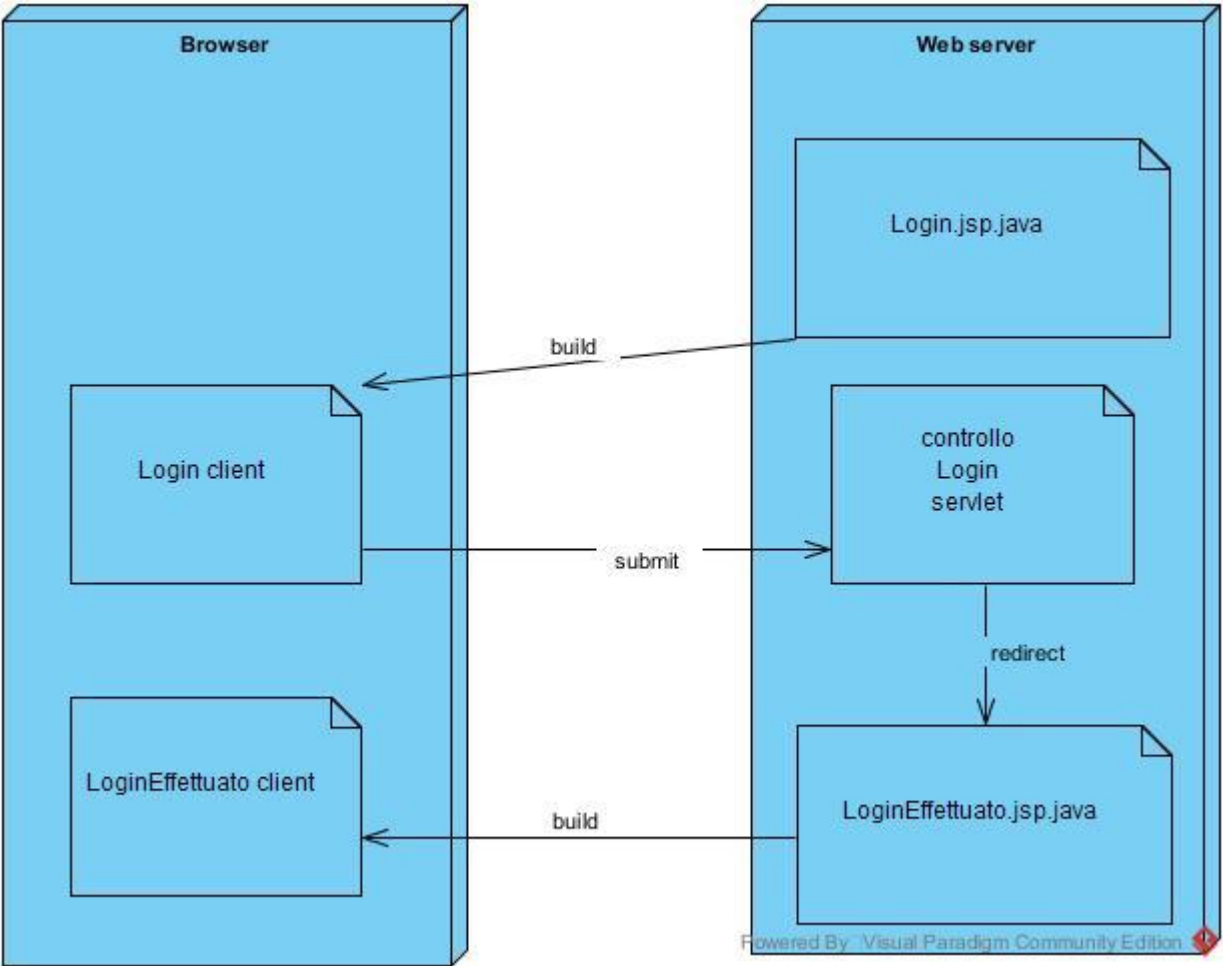




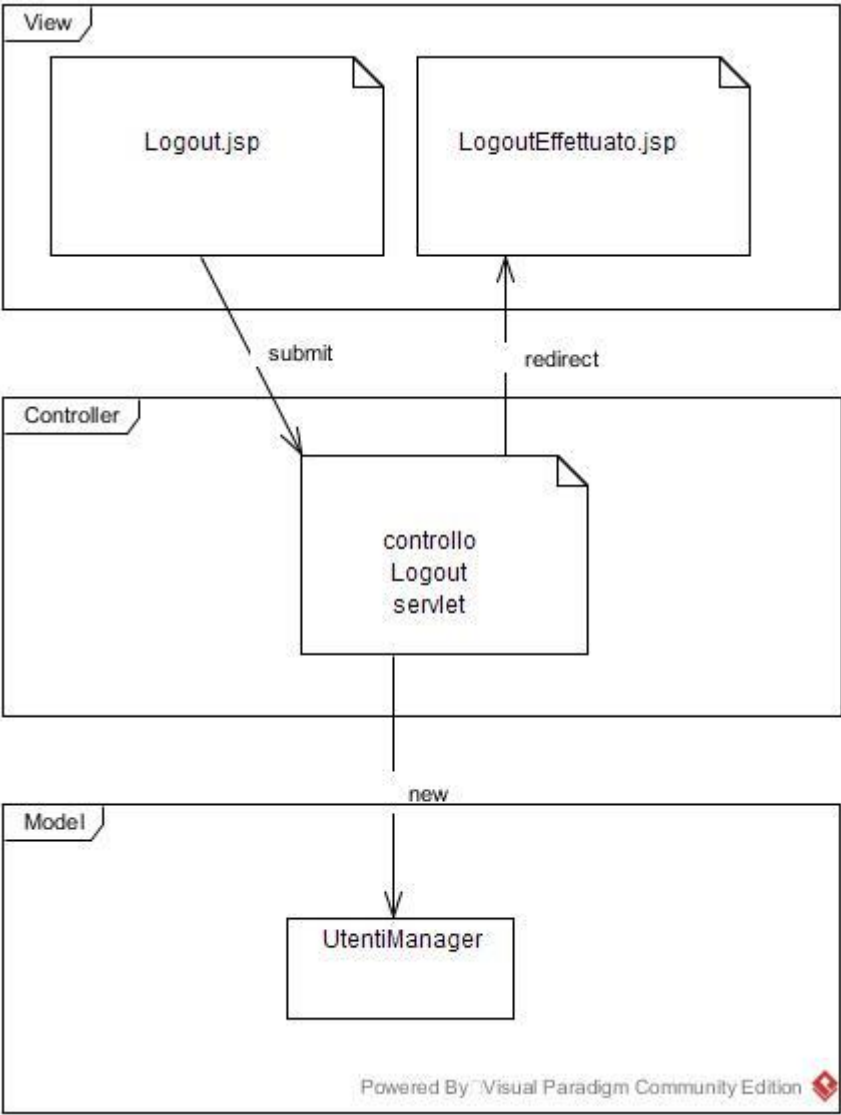
## Login component diagram



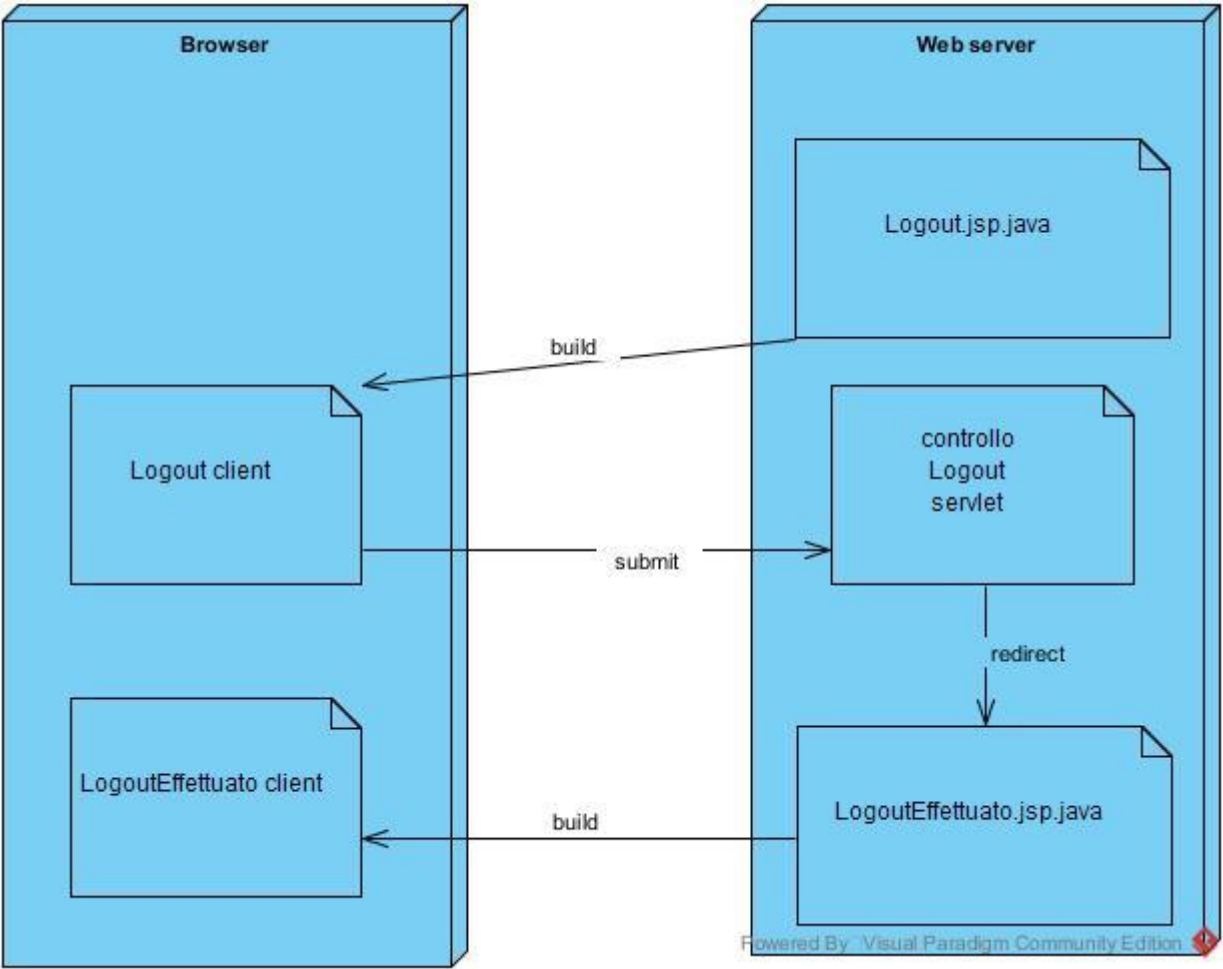
Login deployment diagram



Logout component diagram



Logout deployment diagram



## 5.5 GESTIONE DATI PERSISTENTI

*Il sistema utilizza un database di tipo relazionale.*

*Sono state individuate queste classi che devono essere persistenti:*

- Utente
- Libro
- Prestito

I dati sono gestiti persistentemente dal DBMS MySQL tramite tabelle.

Tabella della classe “Utente”:

Attributo	Tipo di dato	Vincoli
nome	stringa	Non nullo
cognome	Stringa	Non nullo
Data di nascita	Stringa	Non nullo
Indirizzo	Stringa	Non nullo
Numero tessera	Stringa	Vincolo di unicità (chiave della tabella)
Password	Stringa	Non nullo

Tabella della classe “Libro”:

Attributo	Tipo di dato	Vincoli
<b>ISBN</b>	<b>stringa</b>	Vincolo di unicità (chiave della tabella)
titolo	stringa	Non nullo
lingua	Stringa	Non nullo
Anno di pubblicazione	intero	Non nullo
categoria	Stringa	Non nullo
Autore	Stringa	Non nullo
Casa editrice	Stringa	Non nullo
Codice prestito	intero	Chiave esterna con riferimento a tabella “prestito”

Tabella della classe “Prestito”:

Attributo	Tipo di dato	Vincoli
Codice Prestito	stringa	Vincolo di unicità (chiave della tabella)
Data inizio	data	Non nullo
Data fine	data	Non nullo
Codice utente	Stringa	Chiave esterna con riferimento a tabella “utente”

## 5.6 CONTROLLO ACCESSI E SICUREZZA

Gli utenti vengono divisi in utente registrato, utente non registrato e amministratore.

L'utente non registrato può consultare il catalogo dei libri e può avere accesso ai contatti della biblioteca.

L'utente registrato aggiunge alle funzionalità dell'utente non registrato, l'accesso alla propria pagina prestiti, l'aggiunta e la rimozione di libri dal carrello e la possibilità di effettuare un prestito.

L'amministratore può registrare nuovi utenti, eliminare utenti, accedere alla lista dei prestiti e chiudere i prestiti.

Matrice degli accessi:

Attore	Gestione Utente	Gestione Libro	Gestione Carrello	Gestione Prestito
Utente non registrato		Ricerca Libro, Visualizza catalogo		
Utente Registrato	Login,logout	Ricerca Libro, Visualizza catalogo	Visualizza carrello, Aggiunta di libri al carrello, Rimozione dei libri dal carrello	Invio prestito, visualizzare i propri prestiti
Amministratore	Login, Logout , registra utenti, elimina utenti	Ricerca Libro, Visualizza catalogo,aggiungi libro,elimina libro		Chiusura prestito, visualizzare prestiti

## 5.7 SOFTWARE DI CONTROLLO GLOBALE

Il sito web ha un flusso guidato dagli eventi (event-driven): non esiste una sequenza di operazioni prestabilita, è l'utente a scegliere di volta in volta l'operazione da eseguire. L'architettura software è composta da un web server che rimane in ascolto in attesa di una richiesta da parte di un web browser. Appena riceve una richiesta di erogazione di un servizio del sistema, elabora il risultato invocando opportuni metodi sui sottosistemi coinvolti nell'operazione e invia l'output al client che provvederà a renderlo all'utente finale. Ogni qualvolta sarà possibile le operazioni saranno realizzate in maniera asincrona per ridurre i tempi di risposta.

## 5.8 BOUNDARY CONDITIONS

### CONFIGURATION

Per ogni oggetto persistente bisogna definire le fasi in cui viene creato, modificato, distrutto, se è possibile, e archiviato nel database.

**Utente registrato:** l'oggetto Utente registrato viene creato dall'Amministratore eseguendo il caso d'uso "Registrazione utente" e archiviato subito dopo la creazione. La rimozione dal sistema avviene su richiesta dell'amministratore quando chiede di eseguire il caso d'uso "Cancella Utente".

**Libro:** l'oggetto Libro viene creato dall'amministratore quando chiede eseguire il caso d'uso "aggiungi libro al catalogo" e archiviato al termine dell'inserimento. La rimozione dal sistema avviene su richiesta dell'amministratore eseguendo il corrispondente caso d'uso "Rimozione libro catalogo".

**Prestito:** l'oggetto ordine viene creato dall'Utente registrato eseguendo il caso d'uso "invio prestito" e archiviato subito dopo la creazione. Viene rimosso dal sistema dall'amministratore eseguendo il caso d'uso "chiudi prestito".

**Amministratore:** l'oggetto amministratore è creato dallo sviluppatore, che lo inserisce manualmente all'interno del db. Poi fornisce le credenziali dell'accesso al bibliotecario, che sarà "l'amministratore" del sito. Non può essere modificato o eliminato.

### START-UP E SHUTDOWN USE CASES

Easybook è divisa tra tre component run time: il web browser, cioè il client, il server e il db.

Il web browser viene acceso e spento dall'utente.

Il server viene acceso e spento dall'amministratore.

La connessione al db viene gestita dal server. Quando avviene lo start-up del server, il server apre la connessione al DBMS e accede al db.

### EXCEPTION USE CASES

L'uso Easybook comporta due tipi di eccezione:

- **Fallimento hardware:** avviene quando il server e/o il db crasha;
- **Fallimento software:** avviene in caso di bug o di inserimento di input errati.

Nel caso di fallimento hardware, per evitare la perdita di dati, bisogna effettuare copie di dati di backup periodiche. Il dispositivo che ha causato il crash viene chiuso. Se tale dispositivo è il server,

viene riavviato.

Nel caso di fallimento software, il sistema dovrà provvedere a risolvere il bug.

## 6.SUBSYSTEM SERVICES:

Nome Sottosistema	Gestore Utente
Descrizione	Si occupa della gestione degli utenti registrati.

Servizio offerto	Descrizione
Registrazione utente	Il sottosistema permette all'amministratore di registrare un nuovo utente.
Login	Il sottosistema permette di autenticarsi e accedere al proprio profilo.
Logout	Il sottosistema permette di effettuare il logout e di non visualizzare più la pagina del profilo.
Cancellazione utente	Il sottosistema permette all'amministratore di cancellare un utente registrato.

Nome Sottosistema	Gestore Libri
Descrizione	Si occupa della gestione dei libri, della loro ricerca e della visualizzazione del catalogo.

Servizio Offerto	Descrizione
------------------	-------------



Visualizza Catalogo	Il sottosistema permette di visualizzare il catalogo, con le informazioni relative ad ogni libro.
Ricerca libri	Il sottosistema permette di ricercare libri dal catalogo.
Aggiunta libro al catalogo	Il sottosistema permette di aggiungere libri al catalogo.
Cancellazione del libro dal catalogo	Il sottosistema permette di rimuovere i libri dal catalogo.

Nome sottosistema	Gestore carrello
Descrizione	È il sottosistema che si occupa della gestione del carrello.

Servizio offerto	Descrizione
Aggiunta libro al carrello	Servizio che permette all'utente l'inserimento di un libro nel carrello
Rimozione libro dal carrello	Servizio che permette all'utente la rimozione di un libro dal carrello
Visualizza carrello	Servizio che permette all'utente di visualizzare il carrello

Nome sottosistema	Gestore Prestito
Descrizione	È il sottosistema che si occupa della creazione e chiusura di un prestito

Servizio offerto	Descrizione
Creazione Prestito	Servizio che permette la creazione di un nuovo prestito
Chiusura prestito	Servizio che permette la chiusura di un prestito
Visualizza tuoi prestiti	Servizio che permette di visualizzare i propri prestiti attivi