**Proposta progettuale**

**Componenti del Team:**

Carlos BORGES, matr: 0522500240

Aniello SAULINO, matr: 0522500219

Luigi SAVINELLI, matr: 0522500223

Valentino VIVONE, matr: 0522500260

**Progetto:**

*Tipologia:* Migrazione verso il web

*Titolo:* GPC – Gestione Polo Conciario

Con il software GPC ci si pone l’obiettivo di gestire e di controllare l’andamento di un polo conciario, affrontando il rapporto con fornitori, clienti e magazzino. Il sistema si basa sulla gestione di un database con il quale si può interagire tramite interfacce che varieranno a seconda dell’utilizzatore del sistema, controllando in modo accurato che non ci siano delle violazioni di dominio (es: possibilità di accedere a dati riservati ad altre persone) da parte dell’applicazione.

Il sistema proposto è diviso in due parti, appartenenti a due gruppi di lavoro separati: il primo si propone di gestire gli attori che utilizzeranno il sistema, il secondo di gestire il magazzino. Nonostante siano entrambe documentate, il codice in nostro possesso riguarda solo la prima parte (Sottosistema Gestore), che sarà l’oggetto della nostra migrazione.

**Architettura:**

Il Sistema è stato decomposto utilizzando l’architettura “Plan-Do-Check-Act”.

Il livello “Plan” consente di:

* Definire il problema / impostare il progetto;
* Documentare la situazione di partenza;
* Analizzare il progetto;
* Pianificare le operazioni da realizzare.

Il livello “Do” consente di:

* Identificare soluzioni e miglioramenti;
* Gestire il miglioramento come un vero e proprio progetto;
* Adottare strumenti di analisi quantitativa delle grandezze coinvolte nel

miglioramento.

Il livello “Check” consente di:

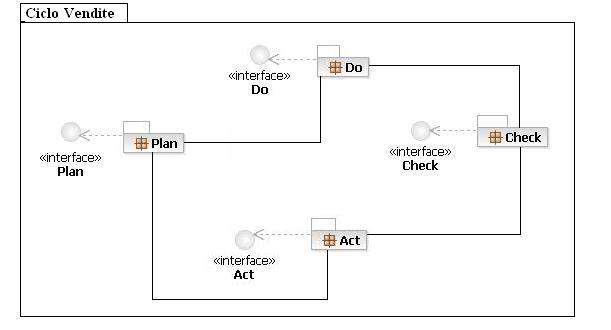
* Verificare i risultati ottenuti rispetto agli obiettivi prefissati;
* Se si è raggiunto l’obiettivo, passare a fase 1 ACT;
* Se non si è raggiunto l’obiettivo, passare a fase 2 ACT.

Il livello “Act” consente di:

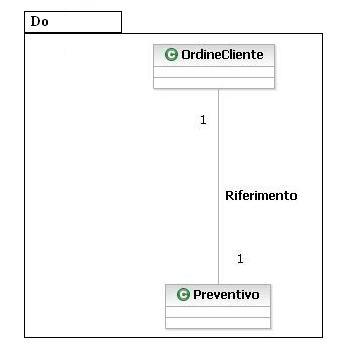
* A obiettivo raggiunto, standardizzare, consolidare e addestrare gli operatori;
* A obiettivo non raggiunto, ripetere il ciclo PDCA sullo stesso problema, analizzando criticamente le varie fasi del ciclo precedente ai fini di individuare le cause del non raggiungimento degli obiettivi.

Il sistema è diviso nei seguenti sottosistemi:

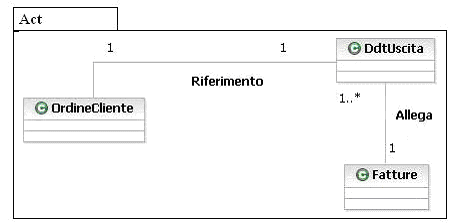
* Sottosistema “Ciclo Delle Vendite”;
* Sottosistema “Ciclo Degli Acquisti”;
* Sottosistema per il Login Utente;
* Sottosistema del Magazzino (Questo non è oggetto di migrazione poiché il codice non è presente).

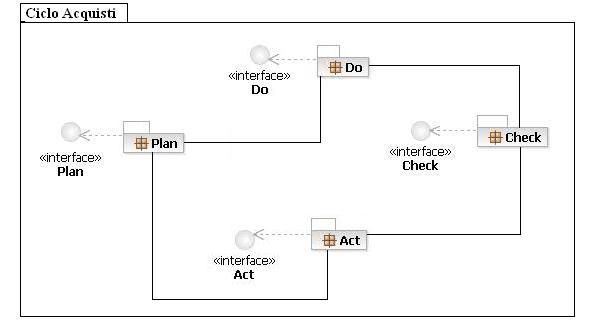


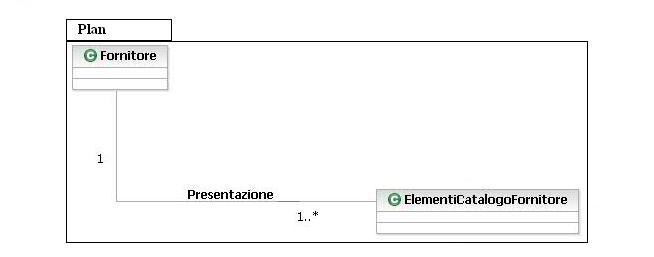




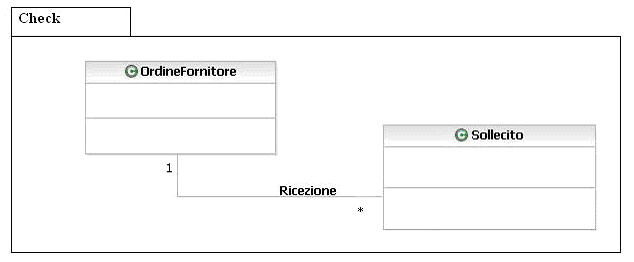


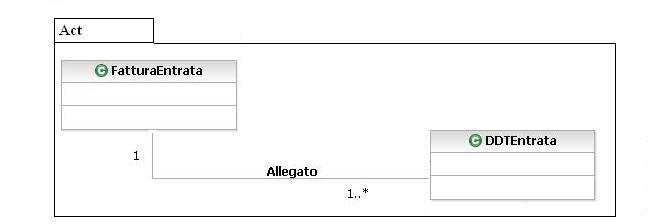








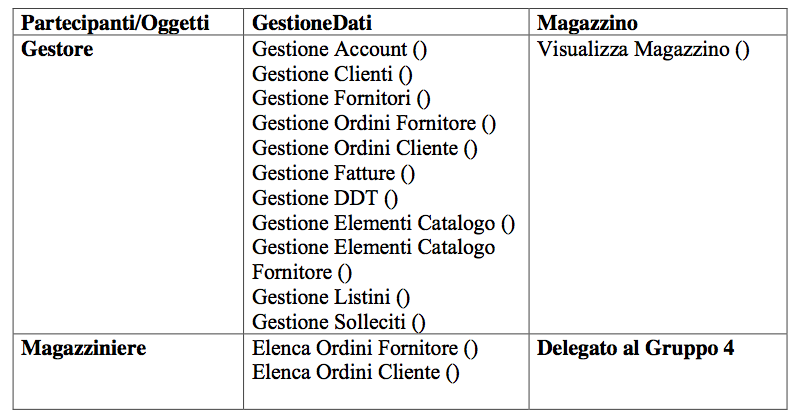




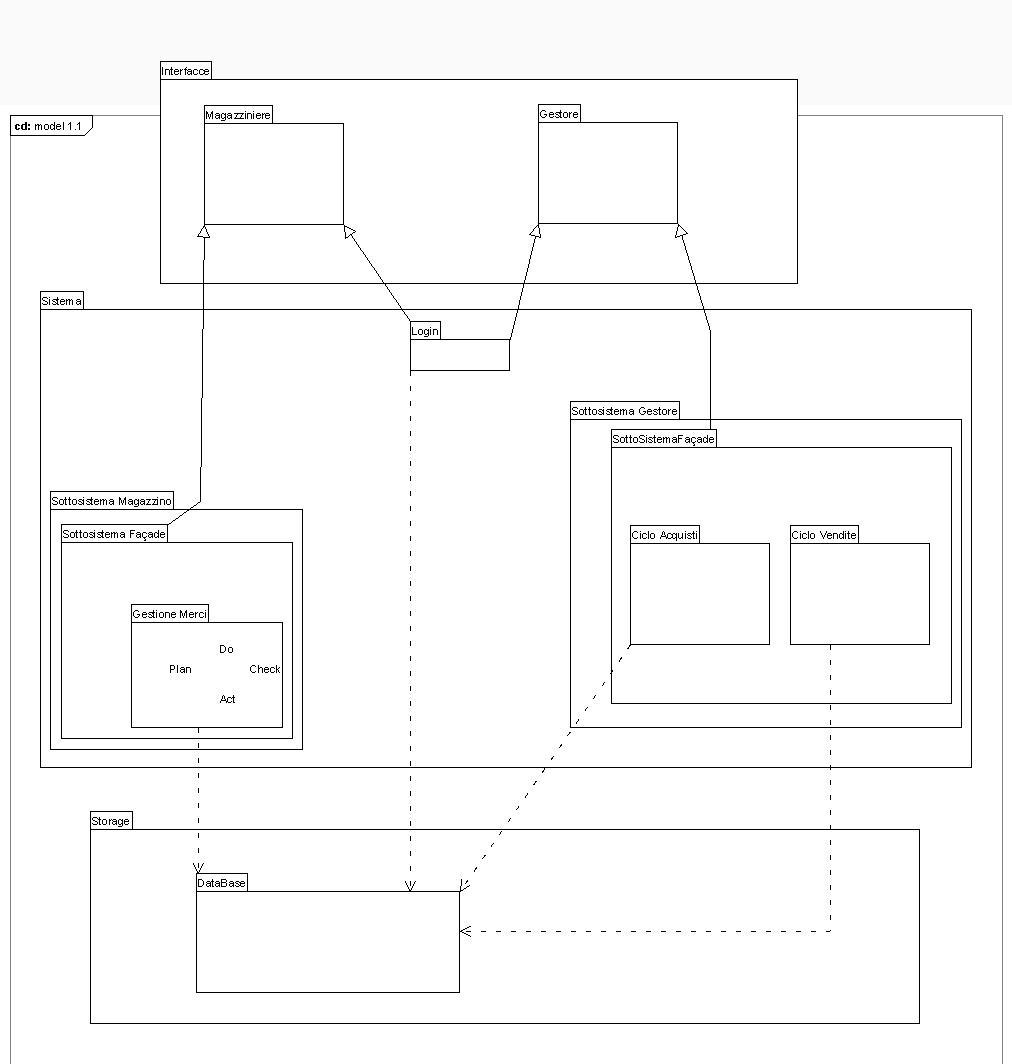
**Controllo Accessi**

Gli accessi al Sistema sono controllati appena si tenta di utilizzare una specifica modalità. Comparirà infatti una finestra dove, per accedere alla modalità desiderata, dovrà essere inserita una login e una password personale.

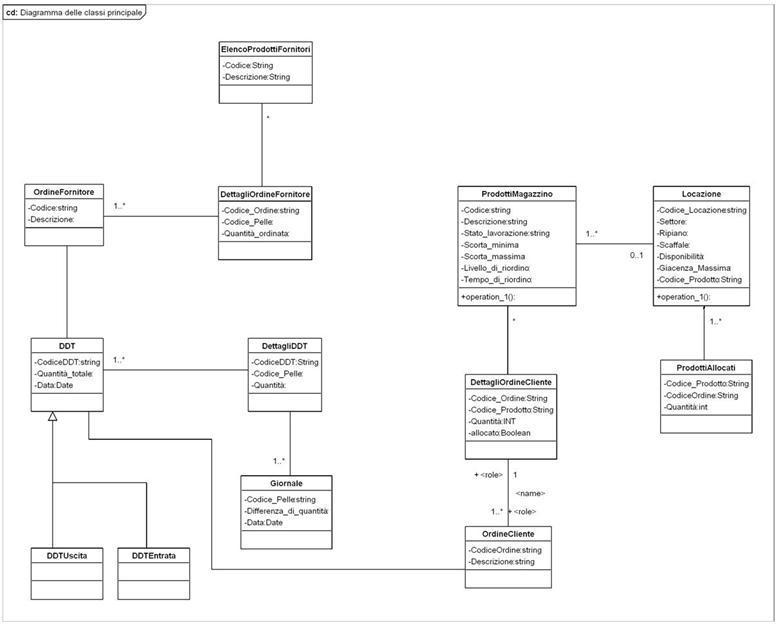
È presente una matrice degli accessi che spiega meglio la politica di controllo degli accessi utilizzata:



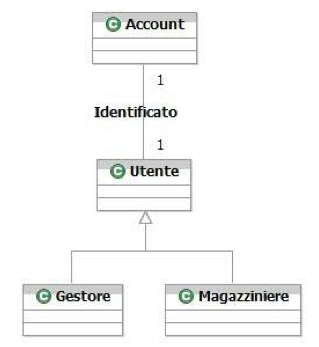
Il sistema è stato decomposto usando l’architettura su tre livelli “Three – tier” e poi integrato con l’approccio “Plan-Do-Check-Act”. Il Livello Superiore (layer interface) è implementato dai sottosistemi “Gestore” e “Magazziniere”, che comprendono tutti i boundary accessibili dagli utenti che utilizzano il Sistema. Il Livello intermedio (layer application logic) include tutti i sottosistemi che controllano le entità presenti nel sistema. Infine il Livello Inferiore (layer storage) è implementato dal sottosistema “Storage” che permette la memorizzazione delle informazioni persistenti e l’interrogazione. L’architettura Three-tier permette di individuare facilmente una gerarchia tra i sottosistemi, organizzati secondo i layer dell’architettura stessa e il modello PDCA.I sottosistemi del layer application logic sono tutti accessibili dal layer interface e tutti accedono al layer storage per ottenere o conservare le informazioni collegate alle entità. Inoltre, all’interno del layer application logic alcuni sottosistemi comunicano tra loro per ottenere informazioni.



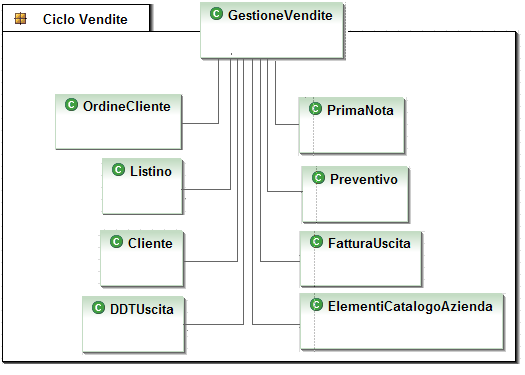
**Class Diagram**

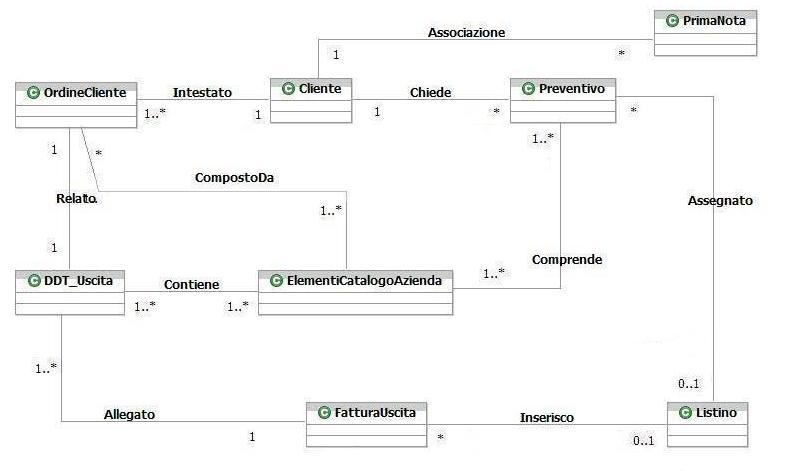


**Diagramma delle classi del Sistema Login**

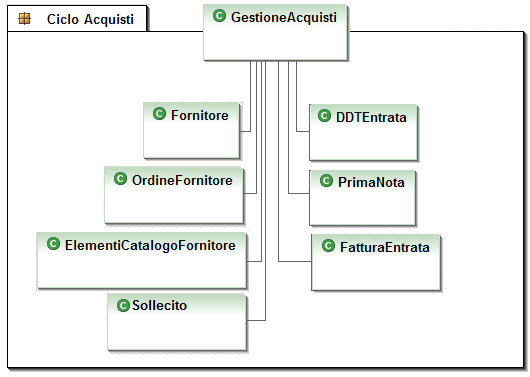
****

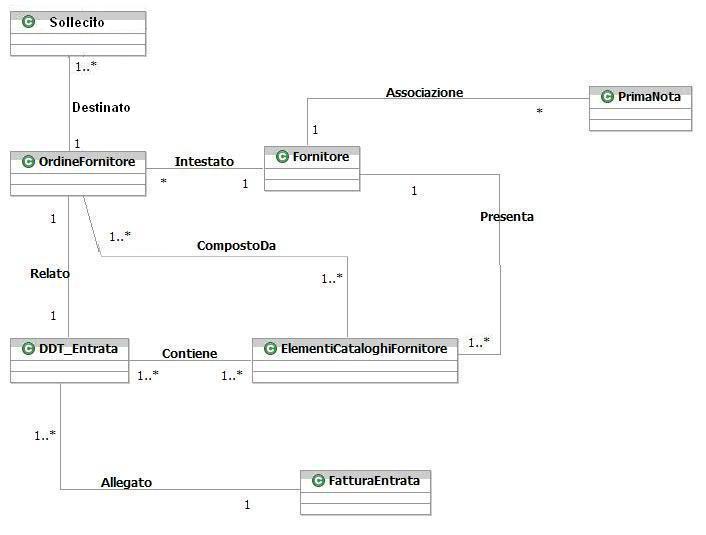
**Diagramma delle classi del Sistema Ciclo delle Vendite**





**Diagramma delle classi del Sistema Ciclo degli Acquisti**





**Documentazione Esistente**

La documentazione rilasciata consta dei seguenti artefatti:

* Problem Statement;
* RAD;
* ODD;
* SDD.

Non sono presenti documenti di Testing, che verranno creati dopo la migrazione.

**Codice esistente**

Il codice esistente presenta le seguenti mancanze:

* Il sottosistema Login va completamente ristrutturato: GPC presenta infatti i beans per la creazione degli account, ma l’accesso viene effettuato in modo statico. Non c’è connessione al database, ma solo un utente memorizzato all’interno del codice per far partire l’applicativo;
* I packages Ciclo Acquisti e Ciclo Vendite presentano solo i relativi beans. Sarà necessario ultimare la connessione tra questi al fine di rendere completamente funzionante il sistema;
* La Gestione dei Clienti/Fornitori (con le relative operazioni di storing, modifca e cancellazione) sono implementate.