UNIVERSITÀ DEGLI STUDI DI SALERNO

Ingegneria del Software

ANNO ACCADEMICO 2015/2016

LOGO

SYSTEM DESIGN DOCUMENT

Versione 1.2

Top Manager:

Nome
Prof. De Lucia Andrea

Partecipanti:

Partecipante	Matricola
Carfora Andrea	0512102638
Grappone Renato	0512102536

Revision History:

Data	Versione	Descrizione	Autore
16/10/2015	1.0		Carfora Andrea Grappone Renato
2/12/2015	1.1		Carfora Andrea Grappone Renato
23/12/2015	1.2		Carfora Andrea Grappone Renato

Indice

1. Introduzione

1.1 Purpose of the System

1.2 Design goals

1.3 Definitions, acronyms and abbreviation

1.4 References

1.5 Overview

2. Current software architecture

3. Proposed software architecture

3.1 Overview

3.2 Subsystem decomposition

3.3 Hardware/software mapping

3.4 President data management

3.5 Access control and security

3.6 Global software control

3.7 Boundary conditions

4. Subsystem Service

1. Introduzione

1.1 Purpose of the System

Lo scopo del sistema è quello di progettare uno Store virtuale operante nel settore della distribuzione di articoli sportivi per varie società sportive nazionali, intraprendendo un'azione di marketing altamente aggressiva, permettendo ai cliente di accedere al sistema da un browser Firefox o Chrome, con il proprio account creato tramite l'apposita interfaccia offerta dal sito. Il cliente può visualizzare i prodotti, selezionarli ed effettuare ordini, visualizzare gli ordini effettuati e contattare il fornitore in caso di problemi o eventuali richieste. Il proprieario del magazzino ha un proprio account impostato gia dal primo avvio del sistema con cui potrà aggiornare il proprio magazzino avare una gestione delle proprie finanze e avere pieno controllo della lista dei propri clienti. L'azienda principale si è posta, inoltre, lo scopo di concedere ai propri clienti un portale e-commerce apposito, distacandosi dai grandi siti di e-commerce come eBay in quanto si vuole che i clienti possano avere diretto contatto con il database dell'azienda per poter consultare il catalogo dei prodotti disponibili presso il magazzino principale ed acquistarli.

1.2 Design goals

Possiamo selezionare gli obiettivi di design da una lista di qualità desiderabili, mediante criteri suddivisi in cinque gruppi: *Criteri Performance, Affidabilità, Costi, Mantenimento, Criteri End User.*

Criteri di Performance

Tempo di risposta:
Il sistema risponderà alle richieste impartite dai componenti in tempo reale. Se la pagina dei prodotti ricercati da un componente è di grandi dimensioni, il sistema la visualizzerà con leggero ritardo (impiegherà qualche attimo in più) rispetto ad una pagina contenente pochi prodotti. Per quanto concerne l'iscrizione del cliente, si vuole che dopo la sottomissione della richiesta al cliente venga data una risposta (Account creato con successo oppure no) in non più di 5 secondi.
Memoria:
La dimensione della memoria è dinamica, in quanto dipende dalla grandezza del DataBase.

Criteri di Affidabilità

Robustezza:
Prima della conferma dell' iscrizione del cliente vi sarà una attenta verifica di attendibilità dei dati immessi dall'utente (controllo della partita iva, esistenza reale della persona fisica, esistenza reale della società, ecc...).

Attendibilità:
I risultati prodotti dalle pagine dinamiche (le servlet) riguardo la disponibilità o meno di prodotti in maggazzino sono attendibili nel senso che rispecchiano istante per istante la reale situazione del magazzino stesso; non si deve mai verificare che venga visualizzata la disponibilità di un determinato prodotto se questo non c'è effettivamente.

Disponibilità:
Una volta che il sistema è stato realizzato sarà disponibile ogni qualvolta il cliente ne richiederà l'utilizzo (a meno di guasti temporanei).

Sicurezza:
La sicurezza è garantita nei limiti da una login e di password non criptata.

Criteri di Mantenimento

Estendibilità:
E' consentito, in quanto è possibile aggiungere in futuro, nuove funzionalità al sistema, oppure creare nuove classi, con l'estensione di quelle già esistenti.

Modificabilità:
Il sistema in futuro potrà estendere il proprio campo di azione in tutta Europa oppure sbilanciarsi persino in tutto il mondo. Quindi l'internazionalizzazione, ossia la capacità di cambiare il sistema per aggiungere convenzioni internazionali (come linguaggi, unità di misura e formati numerici) al sito, potrà essere presa in considerazione in un successivo momento. Il sistema è gestito da un server.

Adattabilità:
Il sistema può essere facilmente portato su diversi domini di applicazione (multipiattaforma).

Portabilità:
Tutto il software relativamente associato al sistema sarà scritto usando il linguaggio di markup HTML 5, e linguaggi correlati come CSS e JavaScript, per conformarsi alla politica corrente della compagnia. Nessun vincolo è imposto dalla piattaforma hardware. Quindi, in virtù della portabilità del linguaggio, l'intero sistema è portabile su differenti piattaforme ed eseguibile da diversi browser.

Leggibilità:
Mediante la lettura del codice il sistema è comprensibile. Infatti semplici istruzioni consentono di comprendere le funzionalità del sistema e con l'ausilio di commenti e documentazione il lavoro di comprensione è ulteriormente semplificato.

Tracciabilità dei requisiti:
In questa fase non siamo in grado di definire la facilità della mappatura del codice nei requisiti specifici, in virtù del fatto che non esiste ancora la stesura del codice.

Criteri End User

Utilità:
Il lavoro dell'utente verrà supportato nel miglior modo possibile dal sistema. Infatti l'utente compierà le operazioni ad egli consentite, senza aver il minimo problema. Il sistema lo seguirà passo passo.

Usabilità:
Il cliente può accedere al sistema facilmente. Altrettanto facile sarà la visualizzazione dei prodotti più in voga, oltre all'iscrizione come cliente. Quando egli ha l'intenzione di visualizzare i prodotti, egli deve richiedere la pagina relativa alla loro viisualizzazione. Quando vuole iscriversi gli viene richiesto di inserire i suoi dati personali all'interno di un form. Il cliente, dopo aver inserito la sua login e la sua password, può richiedere la pagina relativa al catalogo, avendo poi la possibilità di interrogarlo, eventualmente acquistando dei prodotti. Infine quest'ultimo ha la possibilità di effettuare il logout premendo uno specifico pulsante.

1.3 Definitions, acronyms and abbreviation

Acronimi :

RAD: Requirements Analysis Document
SDD: System Design Document
ODD: Object Design Document
DB: DataBase
DBMS: DataBase Management System
BROWSER: Firefox, Chrome
WebBrowser: Cliente (utente che accede al sistema), Venditore (proprietario del magazzino che ha un account già preimpostato all'interno del sistema) WebServer: Server su cui sono memorizzate le risorse.

1.4 References 1.5 Overview

2.Current software architecture

Il sistema è implementato dal nulla in quanto nessun sistema esiste in precedenza, per cui questa fase è di "Greenfield Engineering" e la raccolta dei requisisti viene fatta esclusivamente colloquiando con il cliente e cercando di estrarre i requisiti dalle sue richieste.

3.Proposed software architecture

3.1 Overview

L'obiettivo del progetto è quello di apportare al cliente un beneficio dal punto di vista dei costi (risparmiando sull'acquisto di carta) oltre che un notevole vantaggio dal punto di vista dell'efficienza nella gestione del proprio magazzino. Uno dei vantaggi é la facilità della catalogazione dei propri items riuscendo a sapere in tempo reale le quantità comprendendo un sistema di notifiche che avvisa l'utente in caso di raggiungimento di una soglia minima di un item. Il sistema è capace di gestire più corsi contemporaneamente. Questa è una

caratteristica fondamentale del sistema: semplificare ed avere sotto controllo l'intera attività.

Il sistema punta ad un miglioramento dell'affidabilità; è implementato in modo tale da mettere a conoscenza l'utente di eventi accidentali.

Il sistema è anche in grado di creare delle statistiche basate sull'andamento del bilancio aziendale, permettendo all' utente di capire sia l'andamento dell'attività nel corso degli anni, sia l'andamento della vendita degli articoli. Infine il sistema avrà anche una parte Client che permetterà agli acquirenti di entrare in contatto con il loro fornitore, visualizzare i prodotti, i prodotti presenti in magazzino , ordinarli e acquistarli.

3.2 Subsystem decomposition

Per realizzare il sistema SportsWear Reseller è stato utilizzato lo stile architetturale *three-tier* in versione Client/server. Questa scelta è stata fatta considerando il tipo di tecnologie utilizzate: infatti, essendo il sistema una piattaforma web, tale stile architetturale risponde esaustivamente a tutte le caratteristiche richieste e rispetta tutti i requisiti non funzionali. L' architettura three-tier ("a tre strati") indica una particolare architettura software di tipo multi-tier per l'esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre diversi moduli o strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale (business logic) e alla gestione dei dati persistenti. Tale architettura va tipicamente a mappare a livello fisico-infrastrutturale quella del sistema informatico ospitante l'applicazione da eseguire.

In particolare i tre strati si occupano di differenti funzionalità del sistema di seguito descritte:

Interface Layer

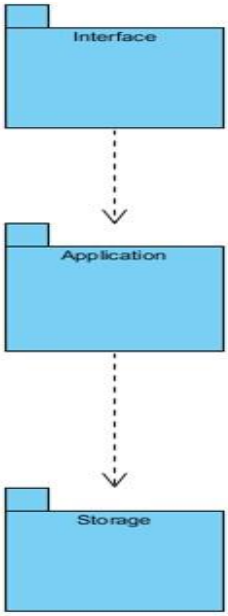
Include tutte le interfacce grafiche e in generale i boundary objects, come le form e i link, con cui interagisce l'utente. L' *interfaccia* verso l'utente è rappresentata da un Web server e da eventuali contenuti statici (es. pagine HTML)

Application Layer

Include tutti gli oggetti relativi al controllo e alle entità del sistema. Tale strato, detto anche *business logic* corrisponde ad una serie di moduli integrati in un application server per la generazione di contenuti dinami

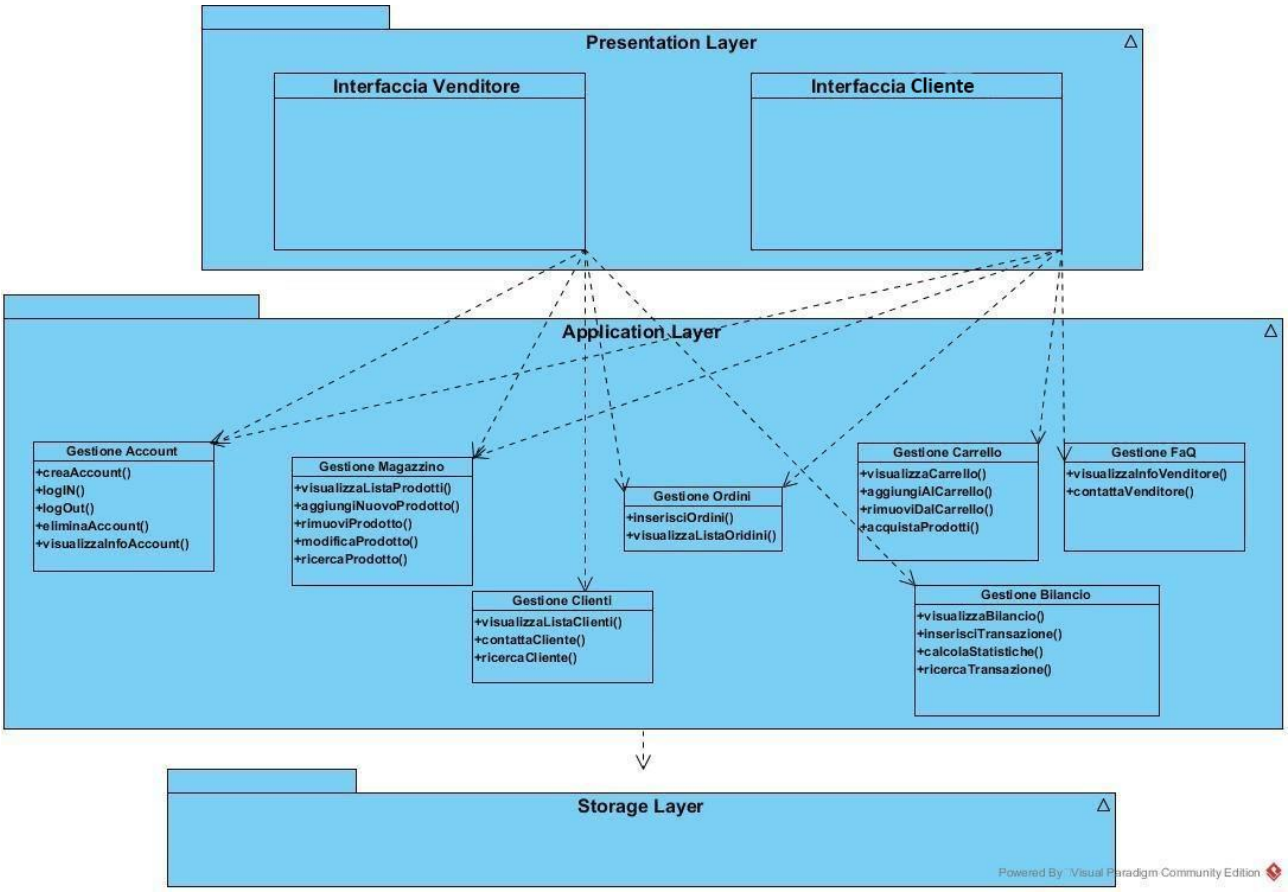
Storage Layer

Effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti. I *dati,* i quali possono essere acceduti dalla business logic, sono depositati in maniera persistente su un DBMS (*data layer*). Può risiedere sulla stessa macchina host dell'application server oppure su una macchina host dedicata e separata. Nel sistema tale gestione sarà assegnata all'application server.



Andiamo ora a vedere nello specifico le funzionalità e gli obiettivi di ogni singolo strato implementato nel sistema.
Nello stile architetturale del software SportsWear Reseller il Presentation layer è caratterizzato dalle interfacce grafiche degli utenti, ovvero dall'interfaccia dell'*utente registrato*, dello *cliente* e del *Venditore*.
La ***logica applicativa*** contiene sette sottosistemi: ognuno di esso è legato all'interfaccia del sottosistema presente allo strato superiore in base alle funzionalità a cui gli stessi possono accedere.
Lo ***Storage Layer*** conterrà la base di dati dove verranno resi persistenti i dati, e comunicherà con l'application layer attraverso il framework JDBC.
Il compito del System Design è quello di decomporre il sistema in sottosistemi che mantengano tra di loro un giusto equilibrio tra coesione ed accoppiamento, permettendo così ai team di lavorare

sui sottosistemi in modo individuale, con un minimo overhead di comunicazione. La suddivisione nell'application layer è stata fatta seguendo proprio tali criteri. A tal proposito si è deciso di suddividere il sistema in dodici aree, ognuna delle quali si occupa di un determinato aspetto della struttura architetturale. Analizziamo ora i singoli sottosistemi presenti nell'application layer:



Gestione Account

Questa Funzionalità raccoglie tutte le operazioni necessarie per consentire agli utenti di gestire le informazioni registrate sul loro account, di autenticarsi e di creare un nuovo Account.

Gestione Magazzino

Questa Funzionalità consente al Venditore di effettuare tutte le operazioni necessarie per gestire al meglio il proprio magazzino.

Gestione Bilancio

Questa Funzionalità consente al Venditore di tenere sotto controllo le finanze della propria attività.

Gestione Cliente
Questa Funzionalità permette al Venditore di avere a disposizione una rubrica dei propri clienti con le relative informazioni e la possibilità di contattarli.

Gestione Ordini
Questa funzionalità permette al cliente di inserire eliminare o modifcare un ordine, mentre permette al Venditore di visualizzare gli ordini che ogni cliente ha confermato

Gestione Carrello
Questa Funzionalità permette ad un cliente di effettuare tutte quelle operazioni necessarie per visualizzare i prodotti aggiunti al carrello ed acquistarli.

Gestione Faq
Questa Funzionalità permette ad un Cliente di visualizzare le informazioni relative al fornitore e contattarlo tramite e-mail.

L'obiettivo principale della *decomposizione in sottosistemi* è ridurre la complessità mentre avvengono cambiamenti. Un sistema grande è di solito decomposto in sottosistemi usando sia gli strati che le partizioni. Il *partizionamento* e la *stratificazione* sono tecniche usate per ottenere una maggiore efficienza dell'intero sistema. In particolare è stata utilizzata la stratificazione per ottenere un Basso Coupling, ovvero che un cambiamento in un sottosistema non incide sugli altri.

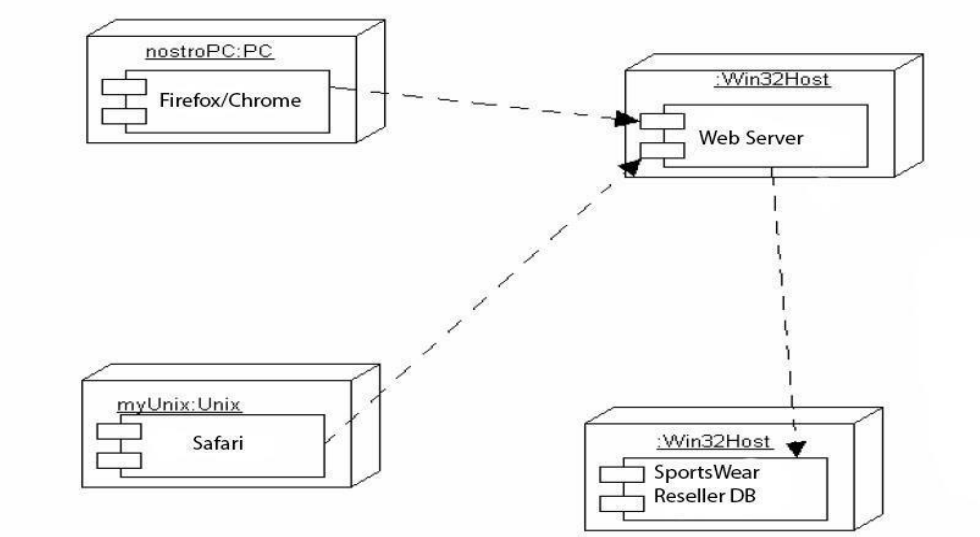
3.3 Hardware/software mapping

Per il sistema, basato su un'architettura distribuita client/server, sono state scelte questi tipi di configurazioni:
PIATTAFORMA: Qualsiasi che supporta un browser java2 compatibile (client) Window NT/2000/XP con almeno 128mb di ram (server)
COMPUTAZIONE: Monoprocessore (server) almeno un i586(pentium) intel 100% compatibile intel con 128MB per WinNT/2000 e 256MB per WinXP
CONNETTIVITA: Tasso di trasmissione standard (minimo 56 kbps) protocollo:
HTTP,TCP/IP
Interazioni: Sincrone, eccetto per la conferma dell'iscrizione del franchiser (il cui tempo di risposta è previsto entro le 24 ore).
COMPONENTI: Memorizzazione dati: DBMS Borland Interbase requisiti minimi: JSDK 1.1 e 10MB circa liberi sul disco
Interazione : CLIENT-SERVER/APACHE TOMCAT v.5 requisiti minimi: processore ì586 compatibile intel 100%, JSDK 1.4.x e 50MB circa liberi sul disco

Piattaforma virtuale: JSDK 1.4.2 requisiti minimi: Pentium 166MHz o microprocessore più veloce con almeno 48MB di RAM, raccomandati per applicazioni avviate all'interno di un browser che usa il Java Plug-in e con 84MB circa liberi sul disco.

3.4 Persistent data management

UML DEPLOYMENT DATA



3.5 Access control and security

Il sistema presenta diversi use-case, di cui solo due accessibili senza autenticazione. La ricerca dei prodotti e l'iscrizione di un nuovo cliente è accessibile a tutti, poiché sono operazioni di dominio pubblico. Le operazioni invece relative al Venditore (che possono essere effettuate solo se autorizzati) richiedono il processo di autenticazione tramite la apposite login e password, strettamente univoche ed inviategli tramite posta elettronica in contemporanea all'acquisto del sistema. Il Venditore può autenticarsi tramite un form apposito nella home page in cui inserire i dati richiesti (login-password), e tramite la pressione di un bottone login.

3.6 Global software control

Il sistema è caratterizzato da un portale accessibile da browser e da un WebServer, attivo 24h, che deve provvedere a gestire gli accessi concorrenti da parte di Clienti e Venditore. Quando un cliente si logga e sottomette i propri dati, vi è un accesso al database (query di interrogazione) che permette di controllare l'esistenza del soggetto. Dopo la conferma, il cliente può accedere a diverse operazioni messe a disposizione dal sistema. Ogni operazione (ad eccezione dell'acquisto) è indipendente dalle altre ed è attivabile dalla pressione di un bottone (submit).

VisualizzazioneProdotto consiste di una ricerca di un singolo prodotto o di una classe di prodotti simili nel tipo o nel prezzo; poiché tale accesso è in sola lettura non c'è bisogno della gestione della concorrenza (infatti non vi è modifica di dati).

AcquistaProdotto richiede un flusso di operazioni molto più complesso. Per prima cosa vi è la chiamata allo UseCase VisualizzazioneProdotto, in modo da ricercare ciò a cui si è davvero interessati. Prima di confermare l'acquisto si entra in una sezione critica ove il DBMS lato server gestisce la concorrenza di più clienti, per evitare che vi siano accavallamenti di decremento di disponibilità (se per esempio due clienti acquistano contemporaneamente lo stesso prodotto). Il cliente può accedere al sistema (in questo caso però le operazioni disponibili sono logicamente divise da quelle accessibili all'Venditore) solo per visualizzare i prodotti o per iscriversi.

3.7 Boundary conditions

Inizializzazione : Il sistema lato server sarà attivo 24h su 24 ed una volta attivato non sarà più stoppato.

Il sistema lato Client è inizializzato ogni volta che un utente accede al portale inserendo l'URL del sito web nel browser. L'interfaccia è quella del sito.

Terminazione: Il sistema lato server non può terminare salvo imprevisti (guasto della macchina server) .

Il sistema lato Client è terminato alla chiusura del browser .

Fallimento: Il sistema lato server può fallire solo a causa di condizioni eccezionali quali mancanza di elettricità o guasti all'hardware (hard disk danneggiato etc che danneggiano permanentemente il sistema) o in caso di crash di sistema (attacchi esterni al server) . Il sistema lato client può fallire a causa di guasti temporanei (hardware o software) o per la caduta della linea telefonica ,ma ciò non influisce sul server che rimarrà comunque attivo e protetto , anche in caso di connessione attiva al DB, poichè c'è il DBMS a gestire le transazioni. Per recuperare un " Fallimento" lato serve occorre riavviare il web-server, ma nel frattempo i client non possono usufruire del sistema.

4 Subsystem Service

SS_ACN: Gestione Account

Application Layer	Presentation Layer	Storage Layer
CreaAccount	GUIRegistrazione	/ /
Login	GUILogin	/ /
Logout	GUILogin GUILogout	/ /
Elimina Account	GUIEliminaAccount	//
Visualizza Info Account	GUIVisualizzaInfo	//

SS_MGZ: Gestione Magazzino

Application Layer	Presentation Layer	Storage Layer
Visualizza Prodotti	GUIVisProdotti	/ /
Aggiungi Nuovi Prodotti	GUIAddProdotti	/ /
Rimuovi Prodotti	GUIRimProdotti	/ /
Modifica Prodotti	GUIModProdotti	/ /
Ricerca Prodotto	GUIRicercaProdotto	//

SS_BIL: Gestione Bilancio

Application Layer	Presentation Layer	Storage Layer
Visualizza Bilancio	GUIVisInfo	/ /

Inserisci Transazione	GUITransazioni	/ /
Statistiche	GUIStatistiche	/ /
RicercaTransazione	GUIRicercaTransazione	//

SS_CLI: Gestione Cliente

Application Layer	Presentation Layer	Storage Layer
Visualizza Clienti	GUIClienti	/ /
Contatta Cliente	GUIContattaCliente	/ /
RicercaCliente	GUIRicercaCliente	//

SS_ORD: Gestione Ordini

Application Layer	Presentation Layer	Storage Layer
Inserisci Ordine	GUIInserOrdine	/ /
Visualizza Ordini	GUIVisualizzaOrdini	/ /

SS_CAR: Gestione Carrello

Application Layer	Presentation Layer	Storage Layer
Visualizza Carrello	GUIVIsualizzaCarrello	/ /
AggiungiProdottoAlCarrello	GUIAggiungiAlCarrello	/ /

RimuoviProdottoAlCarrello	GUIRimuoviDalCarrello	/ /
AcquistaProdotti	GUIAcquisto	/ /

SS_FAQ: Gestione Faq

Application Layer	Presentation Layer	Storage Layer
ContattaFornitore	GUIContattaFornitore	/ /
VisualizzaInfoVenditore	GUIVisualizzaFornitore	/ /