**Progetto:** “Prezzi Pazzi”

**Componenti del gruppo:**

* Andrea Valenti
* Giuseppe Papaleo
* Ruggero Tammaro

**RAD** - **R**equirements **A**nalysis **D**ocument:

**1.Introduzione**  
 1.1 Scopo del sistema  
 1.2 Visibilità del sistema  
 1.3 Obiettivi e criteri di successo del progetto  
 1.4 Definizioni, acronimi e abbreviazioni  
**2. Sistema corrente  
3. Sistema proposto**  
 3.1 Overview (Visione d’insieme)  
 3.2 Requisiti funzionali  
 3.3 Requisiti non funzionali  
 3.4 Modelli di sistema  
 3.4.1 Scenari  
 3.4.2 Casi d’uso  
 3.4.3 Modello a oggetti  
 3.4.4 Modello dinamico

**1.Introduzione**

1.1 Scopo del sistema

L’Obiettivo del progetto “Prezzi Pazzi” è quello di sviluppare un’applicazione in grado di:

1. Gestire un negozio virtuale.
2. Ridurre i costi di gestione.
3. Ottimizzare e velocizzare gli acquisti dei prodotti.
4. Offrire un area dedicata all’utente in cui può visualizzare il catalogo del negozio a suo piacimento

1.2 Visibilità del sistema  
  
Il sistema sarà un applicazione java, realizzata a scopo di didattico quindi è un prototipo. Può essere eseguita solo da chi la installa sul proprio computer, non è connessa in alcun modo ad internet.

1.3 Obiettivi e criteri di successo del progetto

I principali obiettivi che il sistema vuole raggiungere sono: realizzare un prototipo veloce ed accattivante di un possibile e-commerce che rispetti i criteri e i punti di forza del linguaggio OO con cui è verrà scritto.

1.4 Definizioni, acronimi e abbreviazioni

RAD: **R**equirements **A**nalysis **D**ocument

**2. Sistema corrente**

Il sistema da noi proposto è implementato dal nulla, non va a modificare un sistema già esistente. La raccolta dei requisiti viene fatta esclusivamente esaminando un ipotetico documento di richieste fornito da un cliente, che ci è stato fornito dal docente. Sul web è comunque possibile trovare numerosi altri siti simili che verranno esaminati per capirne limiti e punti di forza, queste informazioni saranno usate per migliorare il nostro servizio.

**3. Sistema proposto**

3.1 Visione d’insieme

L’ipotetico cliente del sistema sarà agevolato da un’interfaccia semplice, chiara ed intuitiva. Al termine del progetto, il sistema verrà presentato al docente per valutarne la qualità e la fattibilità implementativa in un contesto reale.

3.2 Requisiti funzionali

Il sistema da noi proposto sarà in grado di:

Attore: Admin,Cliente

* RF-1.1 Login: consente ad un utente registrato al sistema di autenticarsi e svolgere operazioni nella sua area di competenza. Esegue un'autenticazione specificando:
  + Username (stringa alfanumerica)
  + Password (stringa alfanumerica).
* RF1.2 - Logout: consente all'utente di uscire dal sistema.

Attore: Admin

* RF-2.1 Visualizzazione delle offerte: permette all’admin di visionare lo stato attuale del catalogo. Tale funzionalità risulta molto utile per validare le operazioni eseguite sul catalogo.
* RF-2.2 Inserisci offerte: funzionalità che consente di inserire un nuova offerta nel catalogo tra Cena/Vacanza/Bene di Consumo/Prestazione d’opera.
* RF-2.3 Rimuovi Offerte: operazione che consente all’admin di rimuovere delle offerte dal catalogo per riparare ad errori di inserimento, oppure per rimuovere offerte che non si desidera avere nel catalogo.
* RF-2.4 Polita sconti: feature molto utile per abilitare disabilitare delle politiche di sconto a seconda dell’esigenza.

Attore: Cliente

* RF-3.1 Inserire credito: funzionalità fondamentale che permette di effettuare delle ricariche di credito per l’utente, utili per acquistare prodotti nel catalogo.
* RF-3.2 Visualizza Offerte: feature core del sistema, offre all’utente la possibilità di visionare il catalogo in maniera chiara e poter scegliere le offerte che più lo soddisfano
* RF-3.3 Scelta ordinamento: funzionalità accessoria, che permette di scegliere l’ordinamento che più soddisfa le proprie esigenze. Si potrà scegliere tra ordinamento per id crescente oppure un si potrà selezionare un ordinamento cronologico.
* RF-3.4 Visualizza offerte in scadenza: funzionalità molto importante, poiché in caso di sconti permette di visualizzare in maniera veloce le offerte che sono in scadenza e quindi soggette a sconti.
* RF-3.5 Visualizza fornitore: feature di contorno, che può essere utilizzata per offrire all’utente la possibilità di poter visualizzare solo le offerte inserite da fornitori che rispettano un certo livello di qualità

Attore: Utente non registrato

* RF-4.1 Registrazione: L’utente non registrato può eseguire la registrazione al sistema.

3.3 Requisiti non funzionali

*3.3.1 Usabilità*Il sistema verrà utilizzato da diversi tipi di utenti perché ognuno avrà bisogno di accedere a determinate informazioni ed effettuare determinate azioni.

*3.3.2 Performance*Il sistema deve rispondere alle richieste impartite dagli utenti in tempo reale.  
L’acquisto, ad esempio, potrà essere effettuata in poco tempo, indicando l’id e la quantità del prodotto.

*3.3.3 Implementazione*Il sistema sarà implementato utilizzando il linguaggio di programmazione JAVA e a supporto del sistema ci saranno dei file utili per garantire la persistenza.

*3.3.4 Interfaccia*

Il sistema dovrà avere un’interfaccia semplice, piacevole ed intuitiva per rendere la navigazione facile all’utente.

3.4Modelli di sistema:

3.4.1 Scenari

* **Identificazione degli attori**

1. **AMMINISTRATORE DEL SISTEMA:** è colui che ha accesso al back-end del sistema e può inserire/rimuovere offerte.
2. **Cliente**: può effettuare la registrazione ed accedere al sistema e fare acquisti. Inoltrè può fare delle ricariche di credito e scegliere tra tanti funzioni accessorie per visualizzare tutte le offerte del catalogo secondo le sue esigenze.
3. **Utente non registrato:** persona che vuole accedere al sistema ma che non ha le credenziali. Potrà solo visionare la pagina di autenticazione, oppure consultare la pagina di registrazione.

* **Identificazione degli scenari**

|  |  |
| --- | --- |
| **Scenario** | **Inserimento offerta** |
| **Attore** | **Andrea: Amministratore del sistema** |
| **Flusso degli eventi** | 1. **Andrea accede con il suo account amministratore alla piattaforma;** 2. **Il sistema risponde presentando sulla pagina le azioni che Andrea potrà compiere** 3. **Andrea selezionerà la voce “Inserisci Offerta”** 4. **Andrea seleziona un tipo di offerta** 5. **Andrea compilerà il form per l’aggiunta di quel tipo di offerta** 6. **Il sistema notifica l’avvenuta aggiunta** 7. **Andrea si disconnette.** |

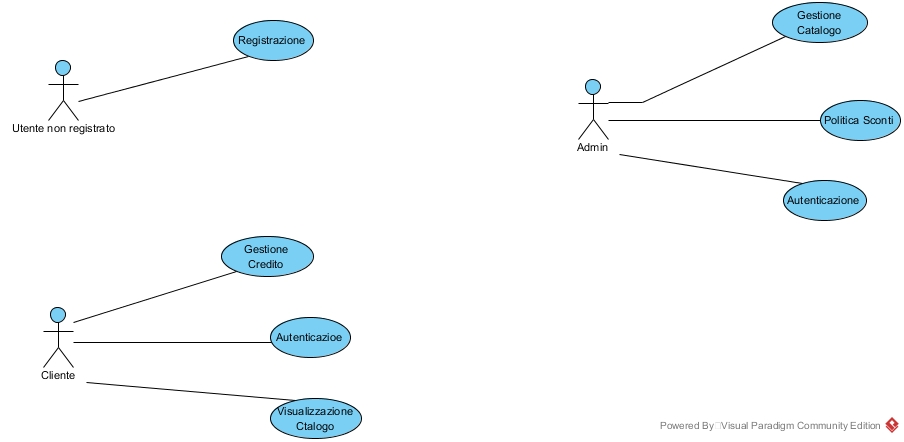
|  |  |
| --- | --- |
| **Scenario** | **Acquisto** |
| **Attore** | **Pino: Cliente** |
| **Flusso degli eventi** | 1. **Pino accede con il suo account alla piattaforma inserendo i dati;** 2. **Il sistema risponde presentando sulla pagina le azioni che Pino potrà compiere;** 3. **Pino selezionerà la voce “Visualizza catalogo” e il sistema gli mostrerà l’intero catalogo;** 4. **Pino inserisce l’id dell’offerta e la quantità e in fondo alla pagina premerà su acquista;** 5. **Il sistema notifica l’avvenuto acquisto** 6. **Pino si disconnette.** |

|  |  |
| --- | --- |
| **Scenario** | **Ricatica credito** |
| **Attore** | **Pino : Cliente** |
| **Flusso degli eventi** | 1. **Pino accede con il suo account alla piattaforma inserendo i dati;** 2. **Il sistema risponde presentando sulla pagina le azioni che Pino potrà compiere;** 3. **Pino selezionerà la voce “Area personale” e il sistema gli mostrerà altre azioni che può compiere;** 4. **Pino seleziona “Ricarica credito”;** 5. **Pino inserisce l’importo che vuole ricaricare e preme ricarica;** 6. **Il sistema notifica l’avvenuta ricarica** 7. **Pino si disconnette.** |

|  |  |
| --- | --- |
| **Scenario** | **Iscrizione** |
| **Attore** | **Antonio: Utente non registrato** |
| **Flusso degli eventi** | 1. **Antonio accede all’ Home page del sito e seleziona la voce “Registrati”;** 2. **Il sistema risponde presentando i campi che Antonio dovrà compilare;** 3. **Antonio dovrà inserire i propri dati personali all’ interno degli opportuni campi ed infine cliccare sulla voce “Registrati”.** 4. **Il sistema controlla la veridicità dei dati e confermerà l’avvenuta registrazione;** |

* **Diagramma casi d’uso**

Il sistema da noi proposto è suddiviso in sottosistemi che utilizzano funzioni simili per scopi diversi. Di seguito è presentato il pattern che verrà utilizzato per tali funzioni:



3.4.2Casi D’ Uso

|  |  |
| --- | --- |
| **UC\_ID** | **ADD\_01** |
| **Nome :** | **Aggiungi Offerta;** |
| **Attori :** | **Amministratore del Sistema** |
| **Condizione d’ Entrata:** | **Uno dei negozi che aderisce alla piattaforma impartisce il compito all’ Amministratore di aggiungere un nuovo prodotto al Sistema.** |
| **Flusso degli eventi :** | 1. **L’ Amministratore accede al sistema con le proprie credenziali.** 2. **L’Amministratore è stato autenticato dal sistema e verrà indirizzato verso la propria pagina personale.** 3. **L’ Amministratore selezionerà la voce “Aggiungi Offerta” .** 4. **Il sistema mostra la pagina di aggiunta Offerte;** 5. **L’Amministratore seleziona il tipo di Offerta.** 6. **Il sistema mostra i campi relativi all’offerta scelta.** 7. **L’Amministratore inserisce i valori e preme su “Aggiungi”** 8. **Il sistema notificherà l’ avvenuta aggiunta.** |
| **Condizioni d’ uscita :** | **Il prodotto sarà visibile nel catalogo.** |
| **Flussi alternativi :** | * **Mancata compilazione di almeno un campo nella fase 7.** * **Continua al punto 5.** |

|  |  |
| --- | --- |
| **UC\_ID** | **REM\_01** |
| **Nome :** | **Elimina Offerta;** |
| **Attori :** | **Amministratore del Sistema** |
| **Condizione d’ Entrata:** | **Uno dei negozi che aderisce alla piattaforma impartisce il compito all’ Amministratore di eliminare un offerta.** |
| **Flusso degli eventi :** | 1. **L’ Amministratore accede al sistema con le proprie credenziali.** 2. **L’Amministratore è stato autenticato dal sistema e verrà indirizzato verso la propria pagina personale.** 3. **L’ Amministratore selezionerà la voce “Elimina Offerta”.** 4. **Il sistema mostra un campo in cui inserire l’id dell’offerta da eliminare** 5. **L’Amministratore inserisce l’id e premerà su elimina.** 6. **Il sistema chiederà una conferma di rimozione.** 7. **IL sistema notificherà l’ avvenuta rimozione.** |
| **Condizioni d’ uscita :** | **L’offerta è stata eliminata con successo.** |
| **Flussi alternativi :** | * **Non accettazione di conferma di rimozione :** * **Id Non valido** * **Continua al punto 5.** |

|  |  |
| --- | --- |
| **UC\_ID** | **REG\_01** |
| **Nome :** | **Registrazione;** |
| **Attori :** | **Utente non registrato;** |
| **Condizione d’ Entrata:** | **L’utente decide di registrarsi al sistema.** |
| **Flusso degli eventi :** | 1. **L’utente accede al sistema** 2. **L’utente clicca sul pulsante “Registrati”.** 3. **Il sistema mostra la pagina di registrazione.** 4. **L’utente inserisce i dati nel form di registrazione;** 5. **Il sistema controlla che tutti i dati necessari siano inseriti;** 6. **L'account viene registrato dal sistema nel file utenti;** 7. **Il sistema notifica l’avvenuta registrazione** |
| **Condizioni d’ uscita :** | **L’utente è stato registrato con successo al sistema.** |
| **Flussi alternativi :** | * **Mancata compilazione di almeno un campo :** * **Il sistema notifica la mancata compilazione di un campo.** * **Continua al punto 4.** |

|  |  |
| --- | --- |
| **UC\_ID** | **CAT \_01** |
| **Nome :** | **Visualizza catalogo;** |
| **Attori :** | **Cliente.** |
| **Condizione d’ Entrata:** | **Il cliente decide di consultare il catalogo.** |
| **Flusso degli eventi :** | 1. **Il cliente accede alla piattaforma;** 2. **Il cliente si logga al sistema.** 3. **Il sistema indirizza Il cliente verso la propria area utente.** 4. **Il cliente seleziona la voce “Visualizza catalogo” .** 5. **Il sistema indirizza il cliente verso una nuova pagina nella quale visualizzerà tutto il catalogo.** 6. **Trovate le informazioni che cercava il cliente effettua il Log-out .** |
| **Condizioni d’ uscita :** | **Consulta del catalogo avvenuta con successo.** |
| **Flussi alternativi :** |  |

3.4.3Diagramma degli oggetti

Object Model: Acquista prodotto

* Entity: Cliente-> un utente del sito che vuole acquistare un prodotto
* Boundary: Catalogo-> una lista contenente le possibili offerte da acquistare
* CampoIdOfferta-> casella in cui il cliente deve inserire l’id del prodotto che vuole acquistare
* CampoQuantità-> casella in cui il cliente deve inserire la quantita che necessita di quel prodotto
* BottoneAcquista-> la pressione di questo tasto, genera l’acquisto per quell’utente che verrà registrato nella sua area personale
* Control: ConfirmPurchase -> una funzione che controlla rispettivamente
  + Se l’utente ha il credito sufficiente per effettuare l’acquisto
  + Se la quantità di quel prodotto è disponibile per soddisfare la richiesta dell’utente
  + E se l’offerta è acquistabile secondo dei criteri che cambiano a seconda del tipo (A supporto di questo punto viene utilizzata un interfaccia comune che contiene un metodo eAcquistabile, che viene scritto a seconda del tipo. Ad esempio una vacanza è acquistabile se la data di partenza non è passata)

Object Model: Visualizza Offerte in scadenza

* Entity: Cliente-> un utente del sito che vuole visionare delle offerte nell’ultima settimana di validità poiché sono scontate
* Boundary: Catalogo-> una lista contenente le offerte che rispettano il criterio dell’utente
* Bottone Visualizza offerte in scadenza -> la pressione di questo tasto, genera la richiesta del catalogo passando un parametro per effettuare confronti sulle date (A supporto di questa funzione c’è un interfaccia comparable che permette di confrontare oggetti del catalogo)

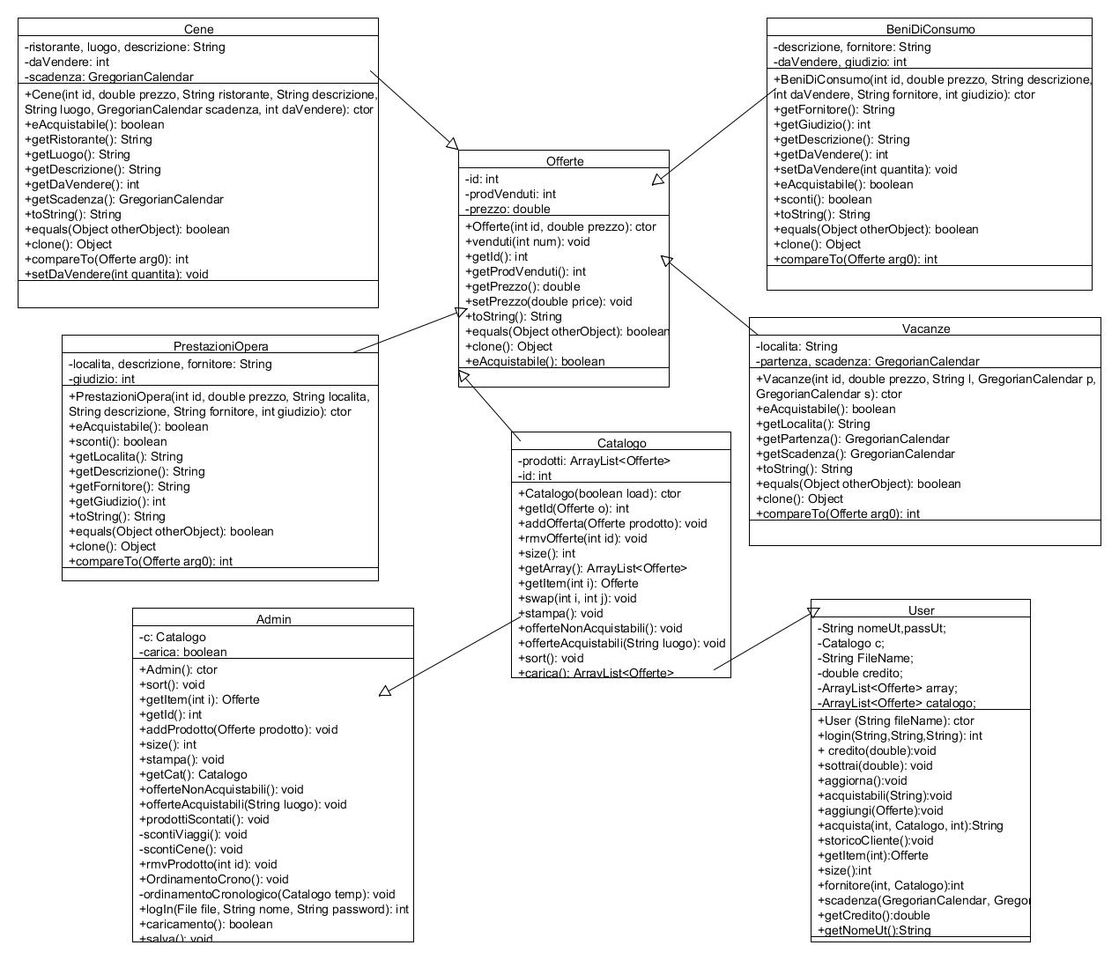
Object Model: Visualizza Catalogo

* Entity: Cliente-> un utente del sito che vuole acquistare un prodotto
* Boundary: Catalogo-> una lista contenente le possibili offerte da acquistare
* Bottone Visualizza catalogo -> la pressione di questo tasto, genera la richiesta del catalogo da parte della view al model e se ci sono offerte disponibili vengono mostrate.

Object Model: Applica sconto

* Entity: Admin-> utente della piattaforma con privilegi per inserire/cancellare offerte.
* Boundary: Catalogo-> una lista contenente le possibili offerte da acquistare
* Bottone Applica Sconto -> la pressione di questo tasto, va a modificare il catalogo applicando le riduzioni di prezzo previste dallo sconto
* Control Applica Sconto -> una funzione che va ad applicare lo sconto pattuito con i negozi. Scorre tutto il catalogo e controlla se delle offerte rientrano in uno sconto. Ad esempio per una Cena si applica uno sconto se si trova nell’ultima settimana di validità.

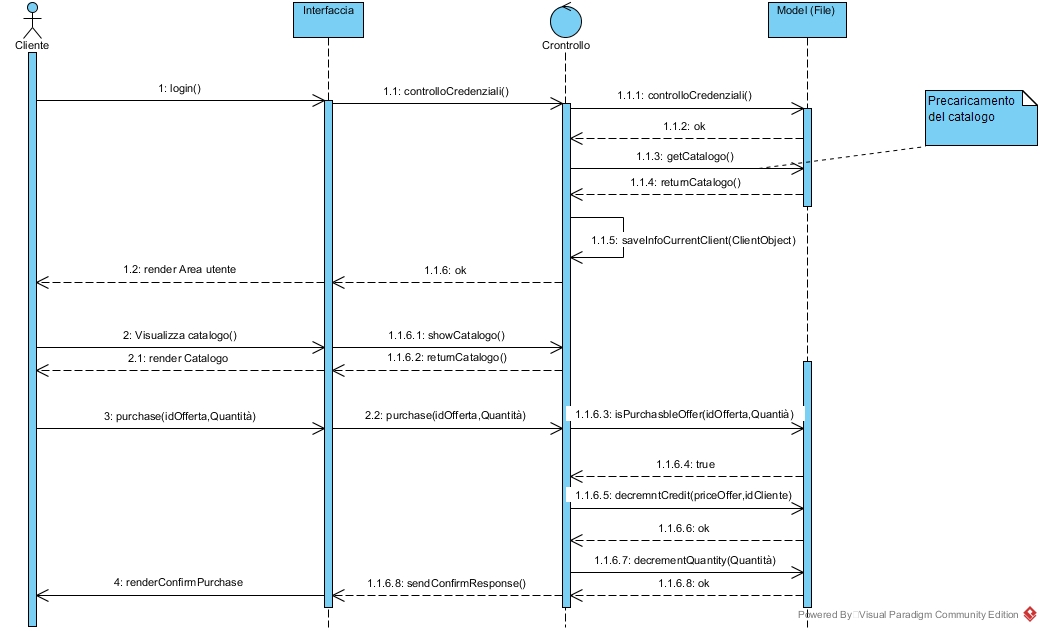
**Diagramma delle classi**



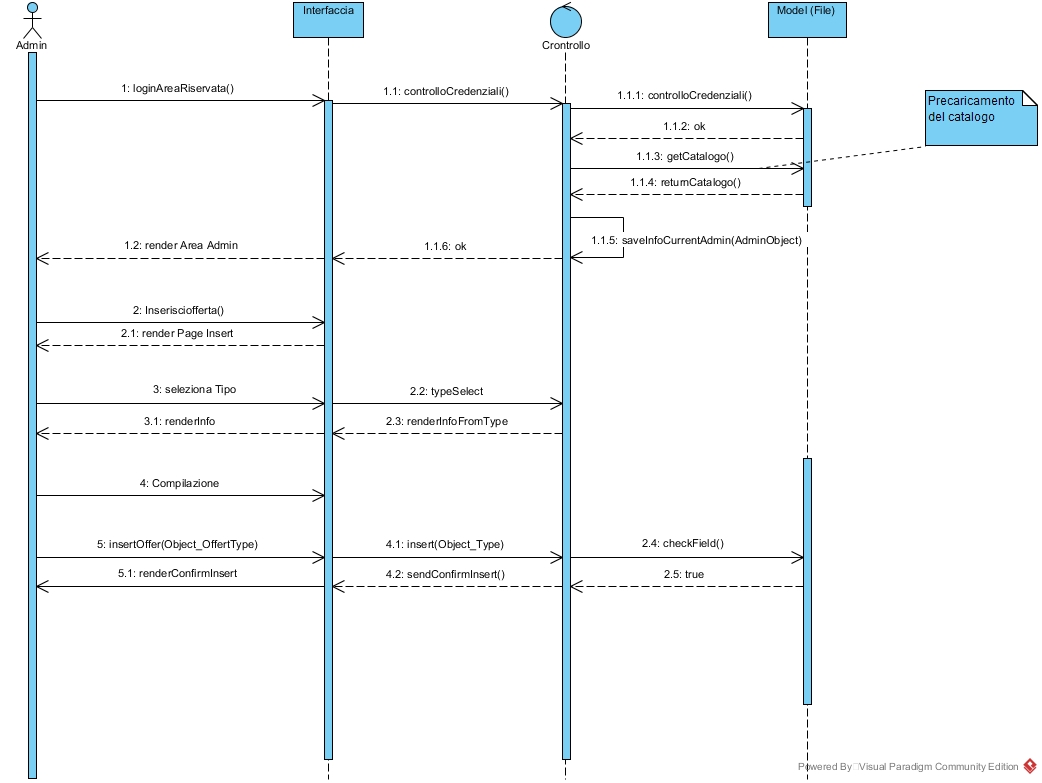
* + 1. Modello dinamico

**Sequence diagram**

* + - Acquisto



* + - Inserisci Offerta



* + - Registrazione

