Università degli Studi di Salerno

Corso di Ingegneria Del Software

Eat & Reorder

Requirements analysis document

**Docente:**

Prof. Andrea De Lucia

**Studenti:**

Abate Francesco

Dello Buono Marco

De Martino Vincenzo

Gagliardi Rosario

Sommario

[1. Introduzione 3](#_Toc25831082)

[1.1. Scopo del sistema 3](#_Toc25831083)

[1.2. Design Goals 4](#_Toc25831084)

[1.2.1. DG1 Criteri di Usabilità 4](#_Toc25831085)

[1.2.2. DG2 Criteri di Affidabilità 4](#_Toc25831086)

[1.2.3. DG1 Criteri di Performance 4](#_Toc25831087)

[1.2.4. DG4 Criteri di Affidabilità 4](#_Toc25831088)

[1.2.5. DG5 Criteri di Implementazione 4](#_Toc25831089)

[1.3. Definizioni, acronimi ed abbreviazioni 4](#_Toc25831090)

[1.4. Riferimenti 4](#_Toc25831091)

[1.5. Panoramica 5](#_Toc25831092)

[2. Architettura del Sistema 5](#_Toc25831093)

[2.1. Panoramica 5](#_Toc25831094)

# Introduzione

## 1.1. Scopo del sistema

Il nostro progetto permette all’utenza di ordinare online piatti dalle aziende della propria città. L’obiettivo principale è quello di agevolare le ordinazioni per il cliente e l’azienda, fornendo dei fattorini che si occuperanno della consegna. La piattaforma sarà estremamente semplice da utilizzare, offrendo un’interfaccia minimale ed intuitiva, in modo tale che qualsiasi utente riesca ad effettuare ogni operazione senza alcun problema. Per rendere possibile ciò, verrà creata una Web App compatibile con qualsiasi dispositivo dotato di browser web, in modo da assicurare un semplice utilizzo a chiunque la provi.

Il nostro sistema fornisce determinate funzioni in base al tipo di utenza.

Gestione Utente:

Utente non registrato:

* Registrazione

Cliente, Azienda, Fattorino:

* + Modifica del profilo
  + Visualizzazione del profilo

Cliente:

* + Inserimento di una segnalazione

Moderatore:

* + Gestione di una segnalazione

Gestione Acquisto:

Cliente, Utente non Registrato:

* + Ricerca di un’azienda
  + Visualizzazione del listino di un’azienda
  + Visualizzazione di un prodotto

Cliente:

* + Aggiunta di un prodotto al carrello
  + Modifica della quantità di un prodotto al carrello
  + Rimozione di un prodotto dal carrello
  + Visualizzazione del carrello
  + Creazione dell’ordine

Gestione Ristorante:

Azienda:

* + Visualizzazione dei prodotti
  + Inserimento di un prodotto nel listino
  + Modifica di un prodotto nel listino
  + Rimozione di un prodotto dal listino
  + Visualizzazione degli ordini commissionati
  + Visualizzazione dei dettagli di un ordine

Gestione Consegna:

Fattorino:

* + Visualizzazione delle consegne
  + Visualizzazione dei dettagli di una consegna
  + Conferma del ritiro di un ordine
  + Conferma della consegna di un ordine

## 1.2. Design Goals

La Web App Eat&Reorder punterà all’essere il più lineare ed intuitiva possibile, in modo da offrire un’interfaccia grafica lineare ed intuitiva. Il sistema sottolinea, quindi, tramite la sua semplice interfaccia grafica, tutte le funzioni disponibili all’utente, il quale difficilmente troverà difficoltà nell’utilizzo di esse, evitando problemi ed inserimenti di dati errati.

Il sistema proposto rispetterà i seguenti criteri di design:

### 1.2.1. DG1 Criteri di Usabilità

* + DG1.1 – Semplicità: il sistema permetterà all’utente di effettuare qualsiasi operazione ad egli inerente senza riscontrare problemi e con estrema semplicità.
  + DG1.2 – Interfaccia grafica: il sistema offrirà un’interfaccia grafica semplice e minimale, onde evitare confusione nell’utilizzo della piattaforma.

### 1.2.2. DG2 Criteri di Affidabilità

* + DG2.1 – Consistenza dei dati: il sistema registra dati e transazioni, garantendo il loro salvataggio in un database e la loro usabilità nell’applicazione.
  + DG2.2 – Il sistema deve garantire che i dati inseriti dall’utente siano corretti, in modo da non immettere dati errati all’interno del database.
  + DG2.3 – Correttezza: il sistema produrrà sempre l’output atteso in base all’operazione svolta dall’utente, evitando errori indesiderati.
  + DG2.4 – Disponibilità: il sistema è disponibile agli utenti 24 ore su 24 e 7 giorni su 7 con alcune limitazioni. I clienti potranno effettuare acquisti prezzo l’azienda scelta solamente durante i giorni e gli orari indicati da quest’ultima.
  + DG2.5 – Sicurezza: il sistema garantisce all’utenza lo svolgimento delle operazioni inerenti solamente alla tipologia di utente che lo utilizza.

### 1.2.3. DG1 Criteri di Performance

* + DG3.1 – Tempi di risposta: il sistema deve garantire tempi di risposta relativamente brevi per ogni funzionalità. Mediamente una richiesta dovrà esser soddisfatta in meno di 5 secondi. Ovviamente tale tempo può dipendere dalla velocità della connessione.
  + DG3.2 – Throughput: il sistema sarà capace di gestire una media di 100 utenti, consentendo loro di effettuare operazioni senza alcun rallentamento.
  + DG3.3 – Dati: i dati salvati dal sistema verranno ricavati nel minor tempo possibile.

### 1.2.4. DG4 Criteri di Affidabilità

* + DG4.1 – Modificabilità: il sistema permette di apportare modifiche alle funzionalità già implementate senza la necessità di modificare i sottosistemi.

### 1.2.5. DG5 Criteri di Implementazione

* + DG5.1 – Implementazione: il sistema, almeno nella sua prima versione, implementerà solamente alcuni requisiti funzionali definiti in fase di analisi.

## 1.3. Definizioni, acronimi ed abbreviazioni

DA COMPILARE DOPO

## 1.4. Riferimenti

Eat&Reorder RAD

Object Oriented Software Engineering Using UML

## 1.5. Panoramica

Il documento si compone di 3 parti fondamentali:

1. Introduzione: vengono illustrati gli obiettivi del sistema proposto e vengono descritte le varie funzionalità messe a disposizione per i diversi attori.
2. Architettura del sistema proposto: viene illustrata la decomposizione in sottosistemi, la gestione dei dati persistenti, il mapping hardware/software, il controllo degli accessi, sicurezza e condizioni limite.
3. Servizi dei sottosistemi e glossario: viene illustrata la descrizione per ogni servizio fornito dal sottosistema e un glossario contenente una raccolta dei termini contenuti nel sistema proposto.

# 2. Architettura del Sistema

## 2.1. Panoramica

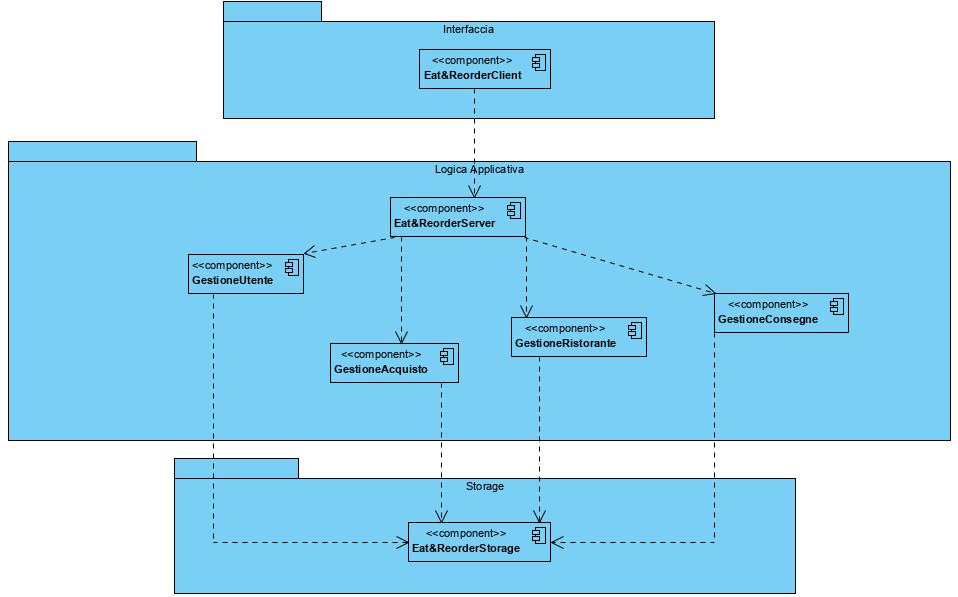
Il sistema prevederà l’utilizzo di un’architettura ibrida tra Client-Server ed MVC: il lato Client si occuperà delle pagine di presentazione del sistema, quindi delle view; il lato Server si occuperà della logica applicativa e di controllo, quindi del model e del controller.

## 2.2. Decomposizione in sottosistemi

### 2.2.1. Macro decomposizione in sistemi

La macro decomposizione divide il sistema nei seguenti sottosistemi:

* Gestione Utente
* Gestione Acquisti
* Gestione Ristorante
* Gestione Consegna

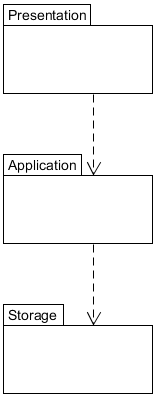


Gli utenti utilizzeranno il sistema dal proprio dispositivo interagendo con l’interfaccia del Server Web, la quale elaborerà dati interagendo a sua volta con il database. Il database verrà gestito da un DBMS che si occupa di inserire, cercare e aggiornare in concorrenza i dati presenti al suo interno, elaborando le richieste ricevute dagli utenti.

### 2.2.2. Micro decomposizione in sottosistemi

Per semplificare la progettazione e lo sviluppo dell’applicazione, i sottosistemi saranno decomposti secondo lo schema previsto dall’architettura software a tre livelli, ovvero:

* Presentation: si occupa della gestione dell’interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l’utente;
* Application: si occupa dell’elaborazione degli input degli utenti, di elaborarli e di fornire dei risultati che verranno mostrati graficamente;
* Storage: si occupa dell’interazione con il database.



#### 2.2.2.1. Gestione Utente

Questo sottosistema si occupa della gestione dell’utenza.

Il sottosistema Gestione Utente fornisce le seguenti funzioni:

* Registrazione account
* Modifica del profilo
* Visualizzazione del profilo
* Segnalazione
  + Inserimento di una segnalazione
  + Gestione di una segnalazione

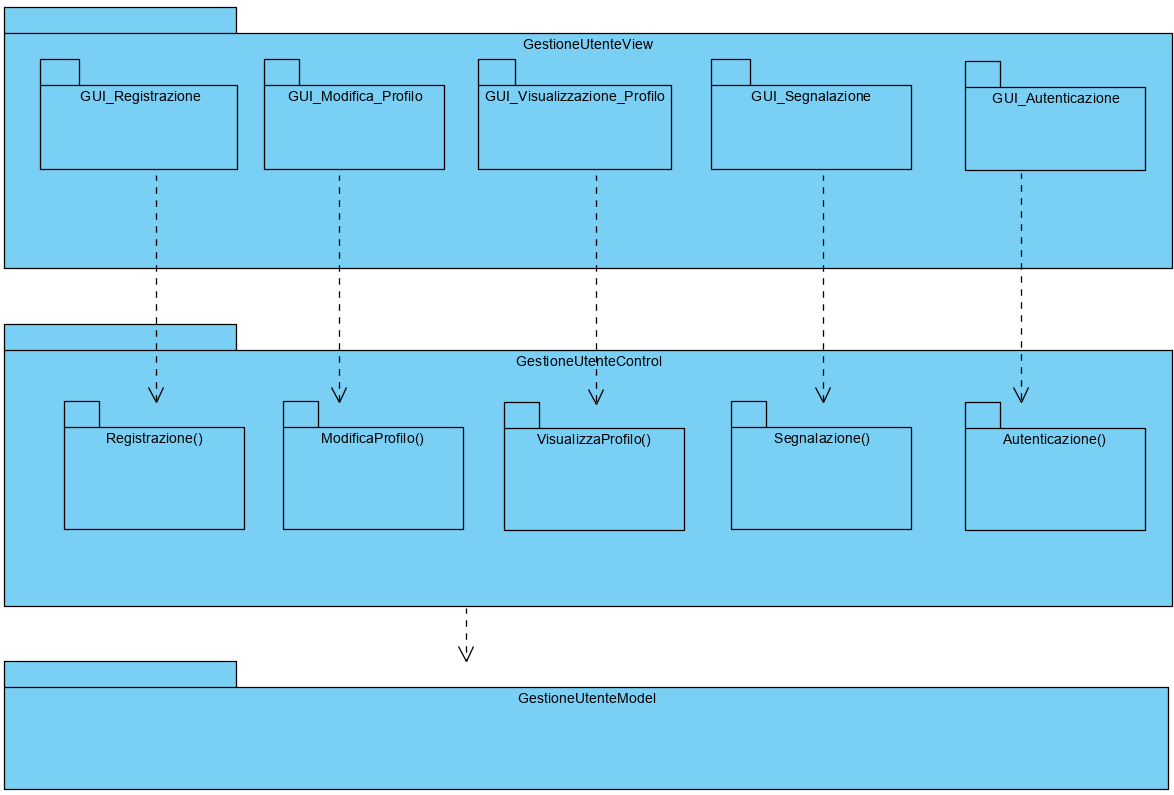
La GestioneUtenteView comprende tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account:

* GUI\_Registrazione – comprende le interfacce che consentono all’utente non registrato di registrarsi al sistema;
* GUI\_Modifica\_Profilo – comprende le interfacce che consentono all’utente registrato di modificare il proprio profilo;
* GUI\_Visualizzazione\_Profilo – comprende le interfacce che consentono all’utente registrato di visualizzare il proprio profilo;
* GUI\_Segnalazione – comprende le interfacce che consentono al cliente di effettuare una segnalazione e al moderatore di gestirla;
* GUI\_Autenticazione – comprende le interfacce che consentono al cliente di effettuare il login nel sistema ed il logout.

La GestioneUtenteControl offre le seguenti funzioni:

* Registrazione() – Registra l’utente non registrato nel sistema
* ModificaProfilo() – Modifica il profilo dell’utente nel sistema
* VisualizzaProfilo() – Visualizza il profilo dell’utente
* Segnalazione() – Crea o gestisce una segnalazione
* Autenticazione() – Effettua il login o il logout

Il GestioneUtenteModel si occupa di rendere reperibili i dati presenti all’interno del database, relativi agli account registrati al sistema.



#### 2.2.2.2. Gestione Acquisti

Questo sottosistema si occupa di:

* Ricerca di Un Prodotto
* Gestione del Carrello
* Ordinazione

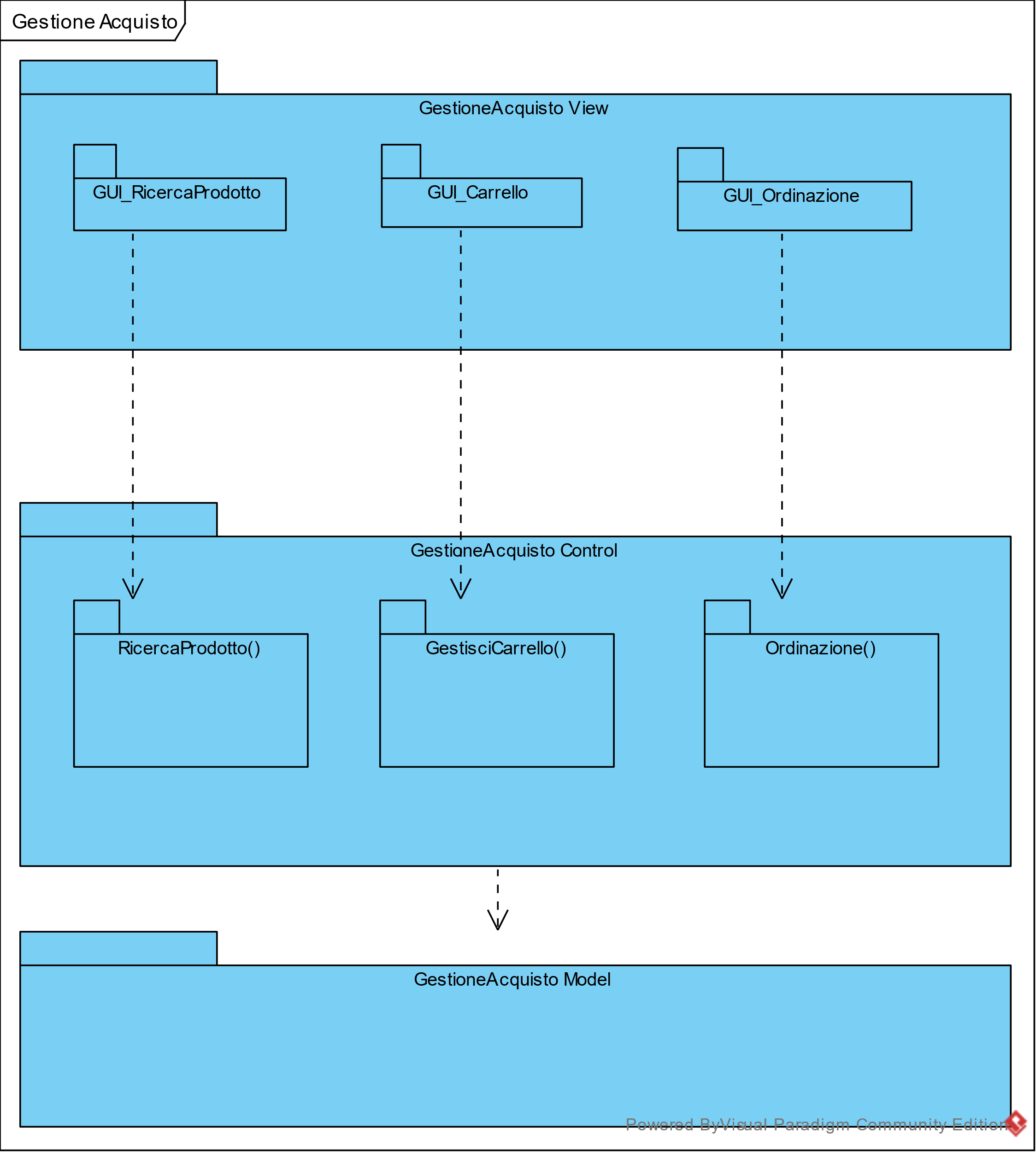
La GestioneAcquistoView comprende tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione di un acquisto:

* GUI\_RicercaProdotto: comprende le interfacce che consentono al cliente di cercare un prodotto
* GUI\_Carrello: comprende le interfacce che consentono al cliente l’aggiunta, l’inserimento e la rimozione di un prodotto nel carrello e la sua visualizzazione.
* GUI\_Ordinazione: comprende le interfacce che consentono al cliente la creazione di una ordinazione

La GestioneCarrelloControl offre le seguenti funzioni:

* RicercaProdotto() consente al cliente di cercare un prodotto.
* GestisciCarrello() gestisce la visualizzazione del carrello, l’aggiunta e la rimozione di un prodotto, la modifica della quantità di un prodotto nel carrello
* Ordinazione() consente al cliente di creare una ordinazione utilizzando i prodotti del proprio carrello.

Il GestioneAcquistoModel si occupa di rendere reperibili i dati presenti all’interno del database, relativi alle aziende, ai prodotti delle aziende e alle ordinazioni.



#### 2.2.2.3. Gestione Ristorante

Questo sottosistema si occupa di :

* Visualizzare i prodotti di una Azienda
* Gestire i prodotti dell’Azienda
* Gestire gli ordini commissionati all’Azienda

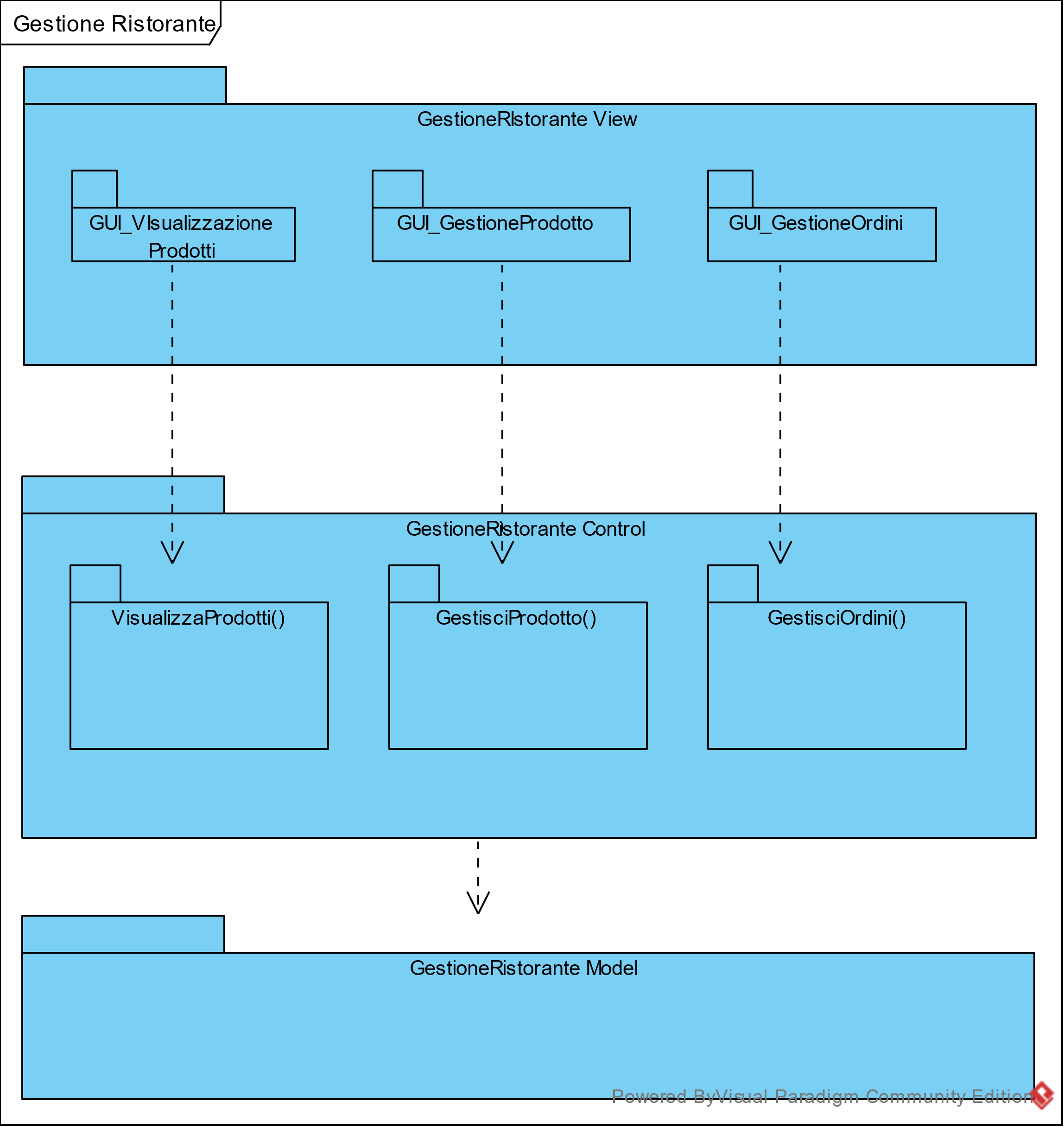
La Gestione Ristorante View comprende tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’azienda:

* GUI\_Visualizzazione\_Prodotti: comprende le interfacce che consentono la visualizzazione dei prodotti offerti dall’Azienda
* GUI\_GestioneProdotto: comprende le interfacce che consentono di inserire, rimuovere e modificare un prodotto presente nel listino dell’Azienda
* GUI\_GestioneOrdini: comprende le interfacce che consentono la visualizzazione dei dettagli degli ordini commissionati all’azienda

La Gestione RistoranteControl offre le seguenti funzioni:

* VisualizzaProdotti() consente all’azienda di visualizzare i prodotti del suo listino
* GestioneProdotto() consente l’aggiunta, la rimozione o la modifica di un prodotto nel listino dell’azienda
* GestioneOrdini() consente la visualizzazione degli ordini commissionati all’Azienda

Il GestioneRistoranteModel si occupa di rendere reperibili i dati presenti all’interno del database, relativi ai prodotti dell’Azienda e agli ordini commissionati.



#### 2.2.2.4. Gestione Consegna

Questo sottosistema si occupa di :

* Visualizzare le consegne e i dettagli delle consegne
* Gestire le consegne

La Gestione Cosegna View comprende tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’azienda:

* GUI\_VisualizzaConsegne comprende le interfacce per la visualizzazione delle consegne e dei rispettivi dettagli
* GUI\_GestisciConsegne comprende le interfacce per conferma del ritiro e della consegna di un ordine

La Gestione Consegna Control offre le seguenti funzioni:

* VisualizzaConsegne() consente al fattorino di visualizzare le consegne e i relativi dettagli
* GestisciConsegne() consente al fattorino di confermare il ritiro e la consegna di un ordine

Il GestioneConsegneModel si occupa di rendere reperibili i dati presenti all’interno del database, relativi alle consegne di un fattorino e alla conferma degli stati degli ordini.

