Università degli Studi di Salerno

Corso di Ingegneria Del Software

Eat & Reorder

System Design 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 del sistema.
3. Servizi dei sottosistemi e glossario: viene illustrata la descrizione dei servizi forniti dal sistema.

# 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 del sistema