

**Fiorazon**

**SYSTEM DESIGN DOCUMENT VERSIONE 0.1**

**INGEGNERIA DEL SOFTWARE 2016-2017**

**25/11/2016**

**SOMMARIO**

[1. INTRODUZIONE 3](#_Toc467241635)

[1.1. SCOPO DEL SISTEMA 3](#_Toc467241636)

[1.2. OBIETTIVI DI PROGETTAZIONE 3](#_Toc467241637)

[1.3. DEFINIZIONI, ACRONIMI E ABBREVIAZIONI 3](#_Toc467241638)

[1.4. RIFERIMENTI 3](#_Toc467241639)

[1.5. PANORAMICA 3](#_Toc467241640)

[2. ARCHITETTURA SOFTWARE CORRENTE 3](#_Toc467241641)

[3. ARCHITETTURA SOFTWARE PROPOSTA 3](#_Toc467241642)

[3.1. PANORAMICA 3](#_Toc467241643)

[3.2. DECOMPOSIZIONE SUBSYSTEM 3](#_Toc467241644)

[3.3. HARDWARE / SOFTWARE DI MAPPATURA 3](#_Toc467241645)

[3.4. GESTIONE DATI PERSISTENTI 3](#_Toc467241646)

[3.5. CONTROLLO ACCESSI E SICUREZZA 3](#_Toc467241647)

[3.6. SOFTWARE DI CONTROLLO GLOBALE 3](#_Toc467241648)

[3.7. BOUNDARY CONDITIONS 3](#_Toc467241649)

[4. SUBSYSTEM SERVICES GLOSSARY 3](#_Toc467241650)

[5. TEAM 3](#_Toc467241651)

1. **INTRODUZIONE**

Il sistema è concepito per aiutare un fioraio, in maniera tale da gestire l’ordine e la vendita dei prodotti via web. La gestione di un negozio fisico comporta diverse spese: il mantenumento di un sitoweb ne comporta molto meno, e inoltre l’utente troverà più comodo ordinare su un sito piuttosto che recarsi di persona al negozio. Il fioraio riesce anche a gestire gli ordini in maniera più semplice. La fama del fioraio, poi, naturalmente aumenta, in qunato via web è più facile da raggiungere.

* 1. **SCOPO DEL SISTEMA**

“Fiorazon” è una piattaforma di e-commerce per la pubblicizzazione e la vendita di fiori online. Il sistema ha l’obiettivo di gestire la visualizzazione del catalogo dei fiori, gestire l’inserimento, rimozione e, più in generale, l’interazione del cliente con il carrello e di gestire il corretto acquisto del catalogo mediante l’apposita procedura. Deve inoltre garantire le operazioni di filtraggio della ricerca dei fiori del catalogo.

* 1. **OBIETTIVI DI PROGETTAZIONE**

Il sistema deve rispettare i seguenti obiettivi di progettazione:

**1.2.3 Criteri di performance**

**Tempo di risposta**

Chiaramente, i tempi di risposta per le operazione che deve eseguire l’utente, deve essere inferiore ai 3 secondi.

**Throughput**

Le prestazioni del sistema hanno come unica limitazione l'utilizzo di un database relazionale utilizzato come repository centrale e del tipo di web server utilizzato per la gestione del portale.

**Memoria**

Il sistema ha bisogno di un database relazionale di tipo MySQL per memorizzare i dati relativi agli utenti, agli ordini e ai prodotti.

**1.2.4 Criteri di affidabilità**

**Robustezza**

Il sistema garantisce il funzionamento anche nel caso in cui vengano inseriti input errati: in questo caso, l’utente verrà avvisato dell’errore nell’inserimento.

**Affidabilità**

Il sistema deve portare a termine ogni sua operazione. Le operazioni, quindi, devono essere atomiche.

**Sicurezza**

L’accesso utente e amministratore sarà protetto dall’inserimento di un username e password.

**1.2.5 Criteri di costo**

**Development cost** Non sono presenti costi di sviluppo, in quanto utilizzeremo Apache Tomcat come web server e MySQL come DBMS e i compnenti lavorano gratuitamente.

**Deployment cost**Non sono previsti costi di deployment.

**1.2.6 Criteri di mantenimento**

**Estendibilità** L’utilizzo di linguaggi, quali HTML, JAVASCRIPT, CSS, SQL e JAVA, permetterà la facile introduzione di nuove funzionalità.

**Leggibilità**Grazie ad appositi commenti al codice e documentazione, si potrà facilmente intervenire su di esso.

**Modificabilità**Chiaramente, il codice sarà modificabile, in maniera tale da migliorarlo e da correggere eventuali imperfezioni.

**Tracciabilità**In tutte le fasi di sviluppo, fino alla codifica, si potrà risalire a requisiti funzionali per ogni funzione.

**1.2.7 Criteri dell’utente finale**

**Usability** Il sistema dovrà essere user-friendly, cioè non richiede competenze specifiche. Le interfacce dovranno essere intuitive. Ogni bottone avrà un’etichetta che ne specifica la funzione, in maniera tale da non confondere l’utente.

**Utility**Grazie ad opportuni messaggi, l’utente sarà sempre avvisato degli errori.

* 1. **DEFINIZIONI, ACRONIMI E ABBREVIAZIONI**

|  |  |
| --- | --- |
| **Acronimo** | **Definizione** |
| DB | Database (sistema di memorizzazione per l’archiviazione dei dati permanenti) |
| DBMS | Database Management System (gestore del sistema di memorizzazione) |
| HTML | HyperText Markup Language (linguaggio di markup utilizzato per la definizione della struttura della pagina web) |
| HTTP | HyperTextTransferProtocol (Protocollo per la gestione di richieste e risposte scambiate tra client e server) |
| JAVA | Linguaggio orientato agli oggetti |
| JAVASCRIPT | Linguaggio di scripting utilizzato lato client per rendere il portale web dinamico |
| Layout | Impaginazione struttura grafica del portale e dell'applicazione |
| MYSQL | DBMS utilizzato |
| SQL | Structured Query Language (linguaggio per l'invocazione delle richieste al database) |
| Query | Interrogazione al database |
| RAD | Requirementes Analysis Document |
| SDD | System Design Document |

* 1. **RIFERIMENTI**

Per realizzare il progetto, sono stati utilizzati:

* slide del docente;
* B. BRUEGGE, A.H. DUTOIT, OBJECT ORIENTED SOFTWARE ENGINEERING – USING UML, PATTERNS AND JAVA, PRENTICE HALL, 3D EDITION, 2009;
* RAD di Fiorazon.
  1. **PANORAMICA**

Nella prima parte del documento vendono esposti gli obiettivi di progettazione.   
Al secondo e al terzo punto del documento vengono esposte l’architettura software orrente e quella da noi proposta.

Poi, viene mostrata la decomposizione del sistema in sottosistemi. Ad ogni sottositema è associata una funzione. Sono analizzati i dati persistenti necessari e l’approccio alla loro gestione. Sono individuati le categorie di utenti che useranno il sistema e viene formalizzata la politica di accesso. Vengono formalizzate le condizione di eccezione e i casi limite.

1. **ARCHITETTURA SOFTWARE CORRENTE**

Il sistema non sostituirà nessun sistema precedente, ma sono presenti altri siti di e-commerce simili, come [www.floraqueen.it](http://www.floraqueen.it) e [www.floraexpress.it](http://www.floraexpress.it) Molti di questi prevedono spedizioni in tutta Italia con spedizioni anche in giornata. In tutti c’è bisogno della creazione di un account e in alcuni vi è la gestione di un carrello. Si basano su un’architettura client-server, che utilizzeremo come “ispirazione” per il nostro progetto.

1. **ARCHITETTURA SOFTWARE PROPOSTA**
   1. **PANORAMICA**

Fiorazon è un sito web e, come tale, si basa su di un'architettura client server in cui il client è il browser dell'utente: il client chiede l’erogazione di servizi al server che provvederà a fornire una risposta. Per memorizzare i dati persistenti è stato scelto di utilizzare un database relazionale. Si è preferito l’utilizzo di un DBMS per garantire la consistenza dei dati e un rapido accesso agli stessi. Trattandosi di un applicazione web abbiamo optato per un’architettura three-tier: vengono coiè individuati i 3 livelli di presentazione delle informazioni, logica dell’applicazione e sistema di memorizzazione.

* 1. **DECOMPOSIZIONE SUBSYSTEM**

Subsystem decomposition describes the decomposition into subsystems and the responsibilities of each. This is the main product of system design.

* 1. **HARDWARE / SOFTWARE DI MAPPATURA**

Hardware/software mapping describes how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.

* 1. **GESTIONE DATI PERSISTENTI**

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. This section typically includes the description of data schemes, the selection of a database, and the description of the encapsulation of the database.

* 1. **CONTROLLO ACCESSI E SICUREZZA**

Gli utenti vengono divisi in utente registrato, utente non registrato e amministratore.  
L’utente non registrato può consultare il catalogo dei fiori, può avere accesso ai contatti del fiorario, può iscriversi, può aggiungere prodotti al carrello e modificarlo.  
L’utente registrato aggiunge alle funzionalità dell’utente non registrato, l’accesso alla propria pagina utente, il procedimento al pagamento e alla cancellazione del proprio account.  
L’amministratore può gestire il catalogo e accedere alla propria pagina utente.

Matrice degli accessi:

Gestione Magazzino

Gestione Ordini

Gestione profilo utente

Gestione carrello

Gestione pagamento

Gestione catalogo

* 1. **SOFTWARE DI CONTROLLO GLOBALE**

Il sito web ha un flusso guidato dagli eventi (event-driven): non esiste una sequenza di operazioni prestabilita, è l’utente a scegliere di volta in volta l’operazione da eseguire. L’architettura software è composta da un web server che rimane in ascolto in attesa di una richiesta da parte di un web browser. Appena riceve una richiesta di erogazione di un servizio del sistema, elabora il risultato invocando opportuni metodi sui sottosistemi coinvolti nell’operazione e invia l’output al client che provvederà a renderlo all’utente finale. Alcune funzionalità saranno eseguite in maniera sincrona, ad esempio la registrazione o l’acquisto dell’carrello. Ogni qualvolta sarà possibile le operazioni saranno realizzate in maniera asincrona per ridurre i tempi di risposta.

* 1. **BOUNDARY CONDITIONS**

Boundary conditions describes the start-up, shutdown, and error behavior of the system. (If new use cases are discovered for system administration, these should be included in the requirements analysis document, not in this section.)

**CONFIGURATION**

Per ogni oggetto persistente bisogna definire le fasi in cui viene creato, modificato, distrutto, se è possibile, e archiviato nel database.

Utente registrato: l’oggetto Utente registrato viene creato dall’Utente non registrato eseguendo il caso d’uso “Registrazione utente” e archiviato subito dopo la creazione. La modifica avviene su richiesta dell’utente registrato quando chiede di eseguire il caso d’uso “modifica dati profilo” e vengono aggiornati i dati archiviati. La rimozione dal sistema avviene su richiesta dell’utente registrato quando chiede di eseguire il caso d’uso “Cancellazione Profilo Utente”.

Prodotto: l’oggetto Prodotto viene creato dall’amministratore quando chiede eseguire il caso d’uso “aggiungi del prodotto al catalogo” e archiviato al termine dell’inserimento. La modifica avviene su richiesta dell’amministratore quando chiede di eseguire il caso d’uso “modifica prodotto catalogo” e vengono aggiornati i dati archiviati. La rimozione dal sistema avviene su richiesta dell’amministratore eseguendo il corrispondente caso d’uso “Rimozione prodotto catalogo”.

Ordine: l’oggetto ordine viene creato dall’Utente registrato eseguendo il caso d’uso “pagamento” e archiviato subito dopo la creazione. Non viene rimosso dall’ordine, ma viene modificato il suo stato da “Da spedire” a “Spedito” dall’amministratore con il caso d’uso “Avanzamento dello stato dell’ordine” e da “Spedito” ad “Arrivato” dall’utente con il caso d’uso “Conferma dello stato dell’ordine”.

1. **SUBSYSTEM SERVICES GLOSSARY**

The fourth section, Subsystem services, describes the services provided by each subsystem in terms of operations. Although this section is usually empty or incomplete in the first versions of the SDD, this section serves as a reference for teams for the boundaries between their subsystems. The interface of each subsystem is derived from this section and detailed in the Object Design Document.

1. **TEAM**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giuseppe Siani | 0512102958 |
| Carmine D’ Alessandro | 0512103402 |
| Aleandro Giuseppe Libano | 0512102976 |

1. **STORIA DELLE VERSIONI**

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSIONE | DESCRIZIONE | AUTORE |
| 25/11/2016 | 0.1 | Prima Bozza | Carmine D’Alessandro, Aleandro Giuseppe Libano, Giuseppe Siani |