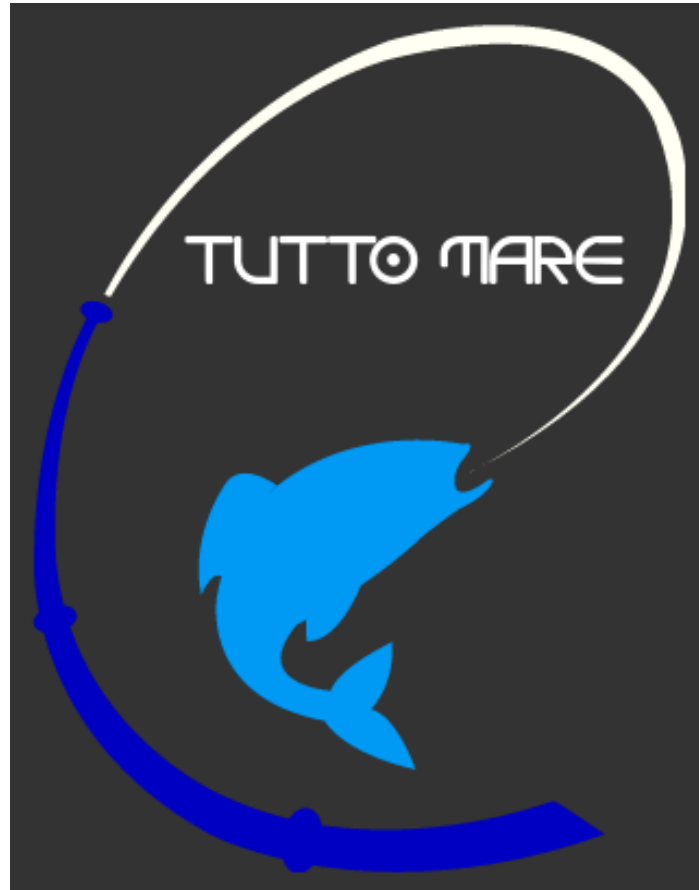


System Design Document



Di:

Sara Guerriero 0512103956
Torre Francesca 0512102366

Indice degli argomenti

Indice degli argomenti:

1.0 Introduzione	3
1.1 Descrizione del sistema	3
1.2 Scopo del sistema	3
1.3 Obiettivi e Criteri di successo del sistema	3
1.4 Dependability criteria	3
1.5 Performance criteria	4
1.6 Maintance criteria	4
1.7 End user criteria	5
1.8 Trade-offs	5
2.0 Definitions, acronyms and abbreviation	5
3.0 Current software architecture	6
4.0 Proposed software architecture	6
4.1 Overview	6
4.2 Subsystem decomposition	6
4.3 Schema Suddivisione Generale	6
4.4 Hardware/ software mapping	11
4.5 Persistent data management	11
4.6 Schema ER	12
4.7 Mapping	13
4.8 Dizionario dei dati	14
4.9 Modello logico	15
4.9.1 Access control and security	16
4.9.2 Global software control	17
4.9.3 Boundary conditions	17
4.9.4 Terminazione	17
4.9.5 Fallimento	17
4.9.6 Use Case condition limits	17
5.0 Servizi dei Sottosistemi	19
6.0 Glossario	19

1.0 Introduzione

1.1 Descrizione del sistema

Vogliamo creare una guida che consenta all'utente di poter interagire con il sito, utile alla vendita del materiale da pesca.

All'interno del sistema possiamo trovare diverse funzionalità come: l'accesso al sistema, la descrizione delle pagine presenti e l'utilizzo di esse con i relativi comandi, permettendo all'utente di svolgere le sue più usuali operazioni in maniera molto elementare.

1.2 Scopo del sistema

Il sistema è mirato ad offrire un insieme di servizi per migliorare la consultazione e nell'acquisto dei vari prodotti, svolte da parte dei clienti.

Al fine di permettere gli acquisti, essi sono stati divisi in utenti generici e registrati;

- Qualsiasi utente si potrà collegare al sito e sarà visto come un utente generico.

Gli utenti generici accede al sistema tramite web browser per visualizzare i vari prodotti e informazioni.

Gli utenti registrato accede al sistema tramite una login per usufruire delle varie funzionalità.

L'amministratore accede al sistema tramite la login per gestire il sistema.

1.3 Obiettivi e Criteri di successo del sistema

Questo progetto ha lo scopo di permettere agli utenti registrati la vendita on-line di oggetti per la pesca, scegliendo diversi tipi di pagamento e sul trasporto del prodotto a destinazione in tempi brevi.

È finalizzato a registrare i vari utenti e il sistema mira ad automatizzare vari operazioni come l'archiviazione in maniera persistente dei dati anagrafici dei vari clienti.

Il sistema permette, inoltre, diverse operazioni in dipendenza dal tipo di utente che si collega al sito. Il sistema prevede una messa in evidenza dei nuovi prodotti utili a facilitare gli acquisti, sono oltre presenti dei filtri che aiuteranno l'utente nella ricerca degli oggetti da acquistare.

1.4 Dependability criteria

Il sito garantisce il proprio corretto funzionamento, gestendo i vari errori.

Quindi, l'applicazione TuttoMare rispetterà i seguenti requisiti di qualità, relativi all'affidabilità:

- **Robustezza:** se l'utente inserisce all'interno del sistema dati sbagliati, esso farà visualizzare dei messaggi di errore all'utente, avvisandolo così del problema esistente.
- **Affidabilità:** garantisce il corretto svolgimento delle funzionalità del sistema.
- **Disponibilità:** il sistema dovrà funzionare in ogni momento (al di fuori dei periodi di manutenzione o di aggiornamento del sistema stesso).
- **Fault tolerance:** l'applicazione garantisce una tolleranza degli errori, cercando di risolverli nel minor tempo possibile.
- **Sicurezza:** per accedere al sito ci sarà un controllo di autenticazione che garantirà ad ogni utente, che è registrato sulla piattaforma, di poter accedere alla propria area riservata.

1.5 Performance criteria

Il sistema deve essere performante, sia in tempi di risposta, sia in ambito degli errori, etc...

Saranno quindi rispettati i seguenti requisiti di qualità:

- **Tempo di risposta:** il sistema garantisce un tempo di risposta del minor tempo possibile (entro i 5 secondi).
- **Throughput:** l'applicazione deve essere in grado di completare ogni attività nel minor tempo possibile.
- **Memoria:** il sistema deve essere scalabile e performante per evitare di incorrere in problemi di rallentamento dovuti ad un eccessivo utilizzo della memoria; dovrà quindi supportare un consistente numero di query.

1.6 Maintance criteria

Il sistema deve essere facilmente mantenibile e deve rispettare i seguenti requisiti:

- **Estensibilità:** il sistema è progettato per poter inserire altre funzionalità utili a migliorare le prestazioni dell'applicazione.
- **Modificabilità:** il sistema dovrà essere realizzato in modo da poter supportare eventuali aggiornamenti.
- **Tracciabilità dei requisiti:** sarà facile risalire ad ogni requisito o altre parti della documentazione grazie al linguaggio chiaro e improntato in maniera schematica.

1.7 End user criteria

Devono essere garantiti i requisiti di qualità:

- **Utilità:** tramite la raccolta dei requisiti, il sistema sarà in grado di soddisfare le esigenze degli utenti.
- **Usabilità:** il sistema deve essere facile da utilizzare per qualsiasi tipo di utente.

1.8 Trade-offs

- **Spazio-Velocità:** il sistema deve occupare meno memoria possibile e deve essere altrettanto veloce ed efficiente.

2.0 Definitions, acronyms and abbreviation

ACROMI	DESCRIZIONE
RAD	Requirement Analysis Document
SDD	System Design Document
HW	Hardware
SW	Software
SQL	Structured Query Language
DBMS	Database Management System
GUI	Graphical User Interface

Il documento sarà diviso in quattro fasi:

- **Introduzione:** viene descritto il sistema con le sue funzionalità e le sue caratteristiche.
- **Proposed software architecture:** viene data una visuale dell'architettura che abbiamo utilizzato per creare l'applicazione.
- **Servizi dei sottosistemi:** vengono descritti i sottosistemi e viene specificato il servizio che ognuno svolge.
- **Glossario:** raccolta di termini per aiutare il cliente a capire la documentazione.

3.0 Current software architecture

La progettazione e lo sviluppo del software inizia da zero.

4.0 Proposed software architecture

4.1 Overview

L'architettura che abbiamo scelto per il sistema è di tipo Three-layer.

L'utente può interagire con l'application layer tramite le interfacce.

L'applicaton layer dovrà comunicare con il database per memorizzare i dati persistenti.

4.2 Subsystem decomposition

Abbiamo adottato tre livelli per l'architettura:

- **interface layer** (gestisce l'interfaccia utente, vengono inclusi gli oggetti boundary, le pagine web, i form, etc...);
- **aplication layer** (comprende le componenti logihe e vengono inclusi gli oggetti entity e control);
- **data storage layer** (si occupa di effettuare le operazioni di inserimento, cancellazione e aggiornamento, inoltre si occupa di rendere disponibili i dati che si trovano all'interno del database);

In ogni strato esistono sottosistemi specifici che possono essere:

- Account;
- Amministratore;
- ServiziUtente;

Sistema Account

Sistema rivolto alle funzionalità dedicate a tutti i tipi di utente: Utente Generico, Utente Registrato e Amministratore.

Sistema Amministratore

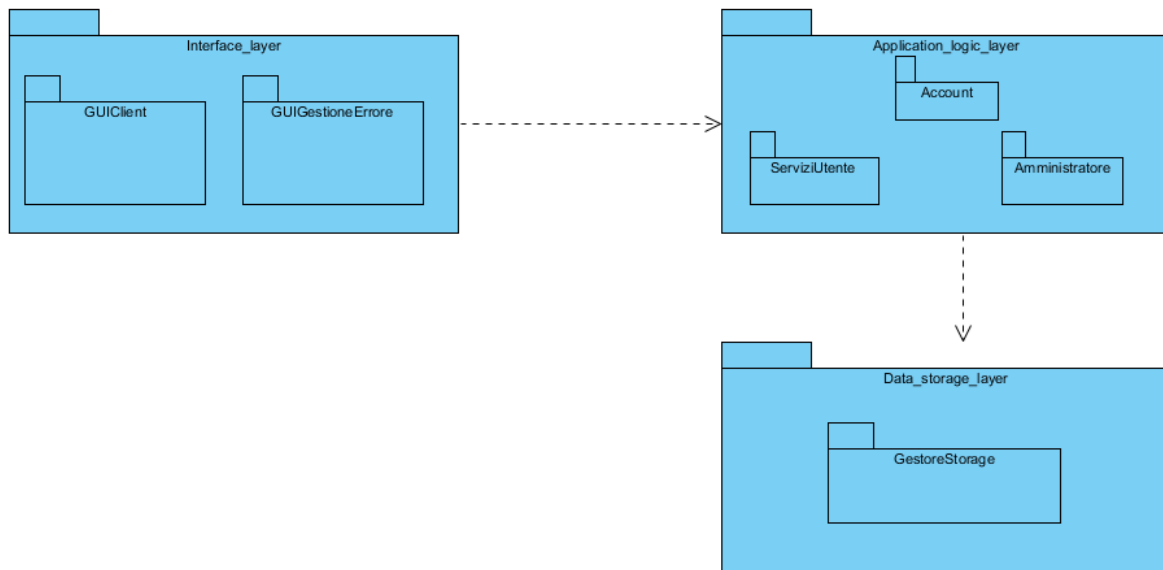
Sistema rivolto alle funzionalità dedicate all'Amministratore. Ad esempio permette di aggiungere o rimuovere un prodotto, di aggiungere o rimuovere una categoria ecc.

Sistema ServiziUtente

Sistema rivolto alle funzionalità dedicate all'utente registrato. Permette ad esempio di acquistare un prodotto, di modificare il proprio account ecc.

4.3 Schema Suddivisione Generale

Di seguito è riportato lo schema della suddivisione e successivamente andremo a descrivere tutti i sottosistemi nel dettaglio.

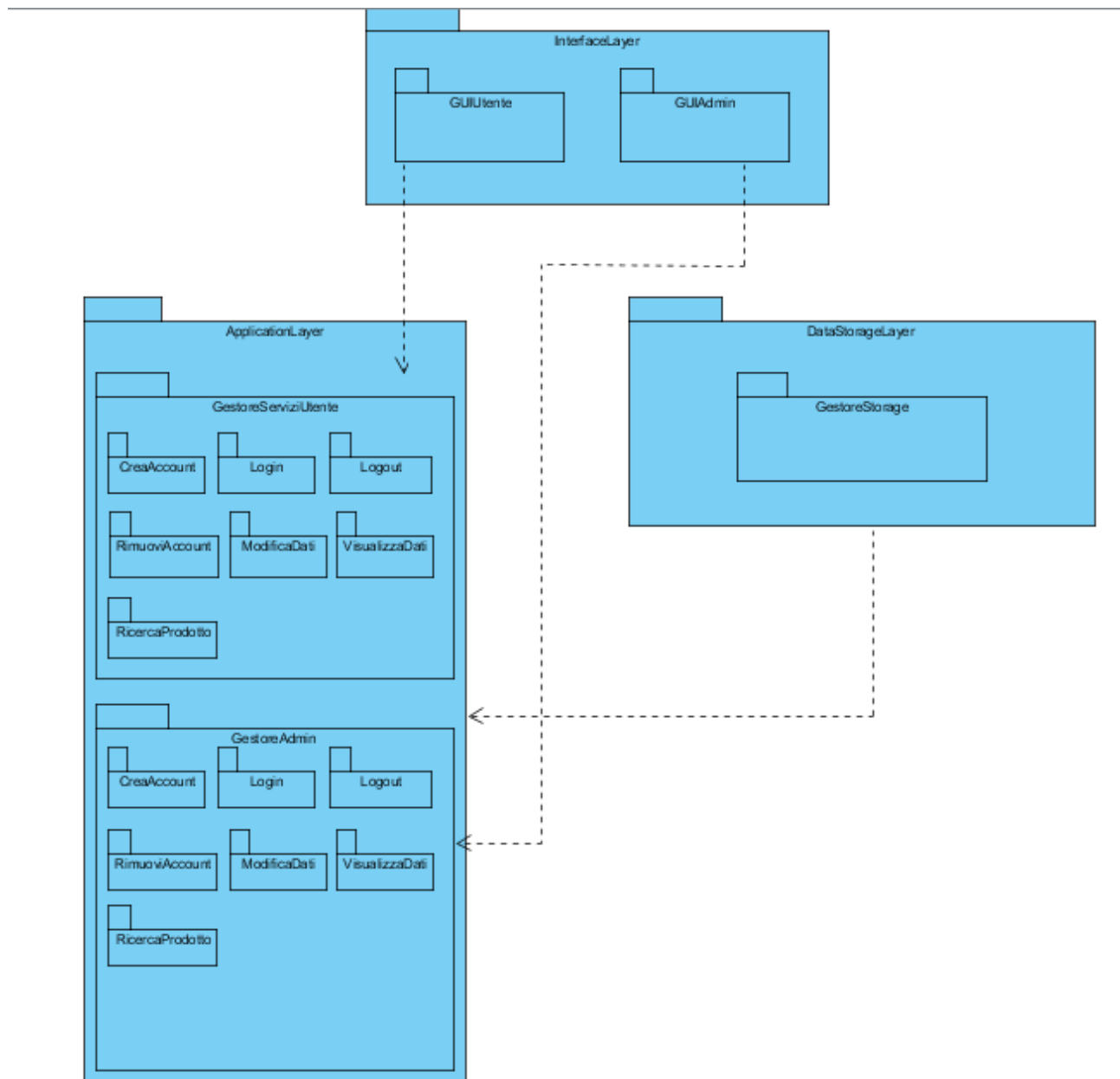


Interface Layer: GUIClient e GUIGestioneErrori indicano i sottosistemi che contengono oggetti boundary;

Application Logic Layer: contiene i tre sottosistemi individuati (Account, Servizi Utente e Amministratore);

DataStorageLayer: è il sottosistema che si occupa di effettuare le operazioni verso il database.

Account:

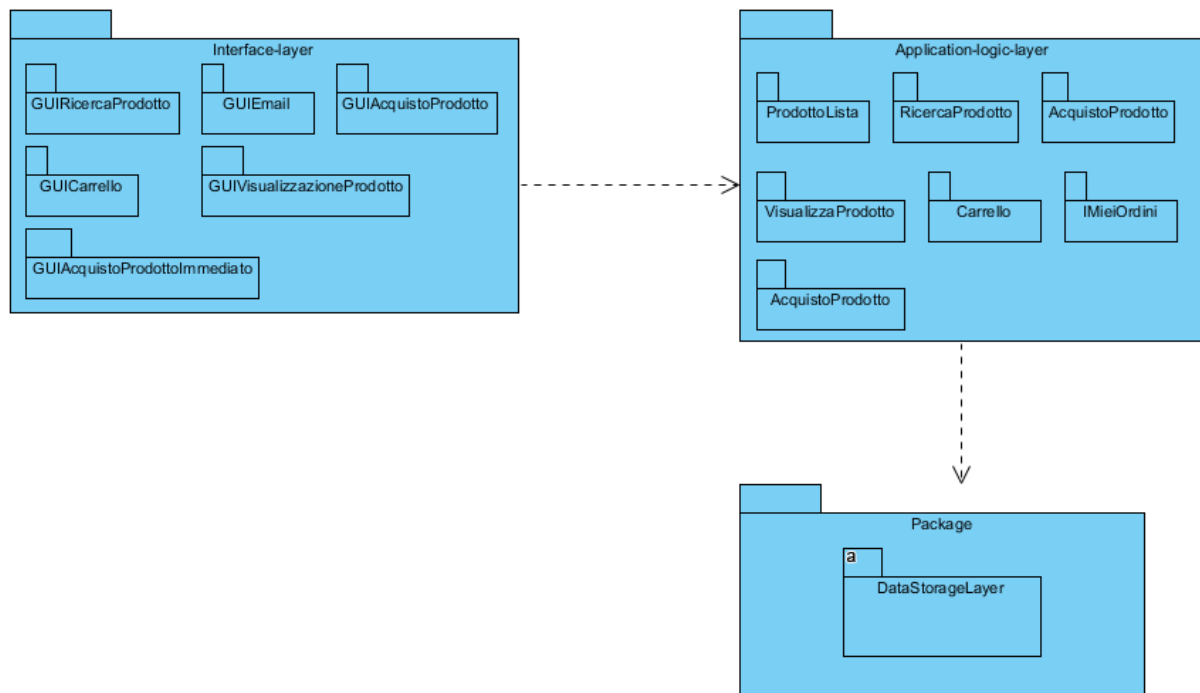


Interface Layer: GUIUtente e GUIAdmin indicano i sottosistemi che contengono oggetti boundary;

Application Logic Layer: contiene due sottosistemi individuati (GestioneAdmin, GestioneServiziUtente) che a loro volta contengono rispettivamente altri sei sottosistemi uguali fra loro (CreaAccount, Login, Logout, RimuoviAccount, ModificaDati, VisualizzaDati, RicercaProdotto);

DataStorageLayer: è il sottosistema che si occupa di effettuare le operazioni verso il database.

ServiziUtente

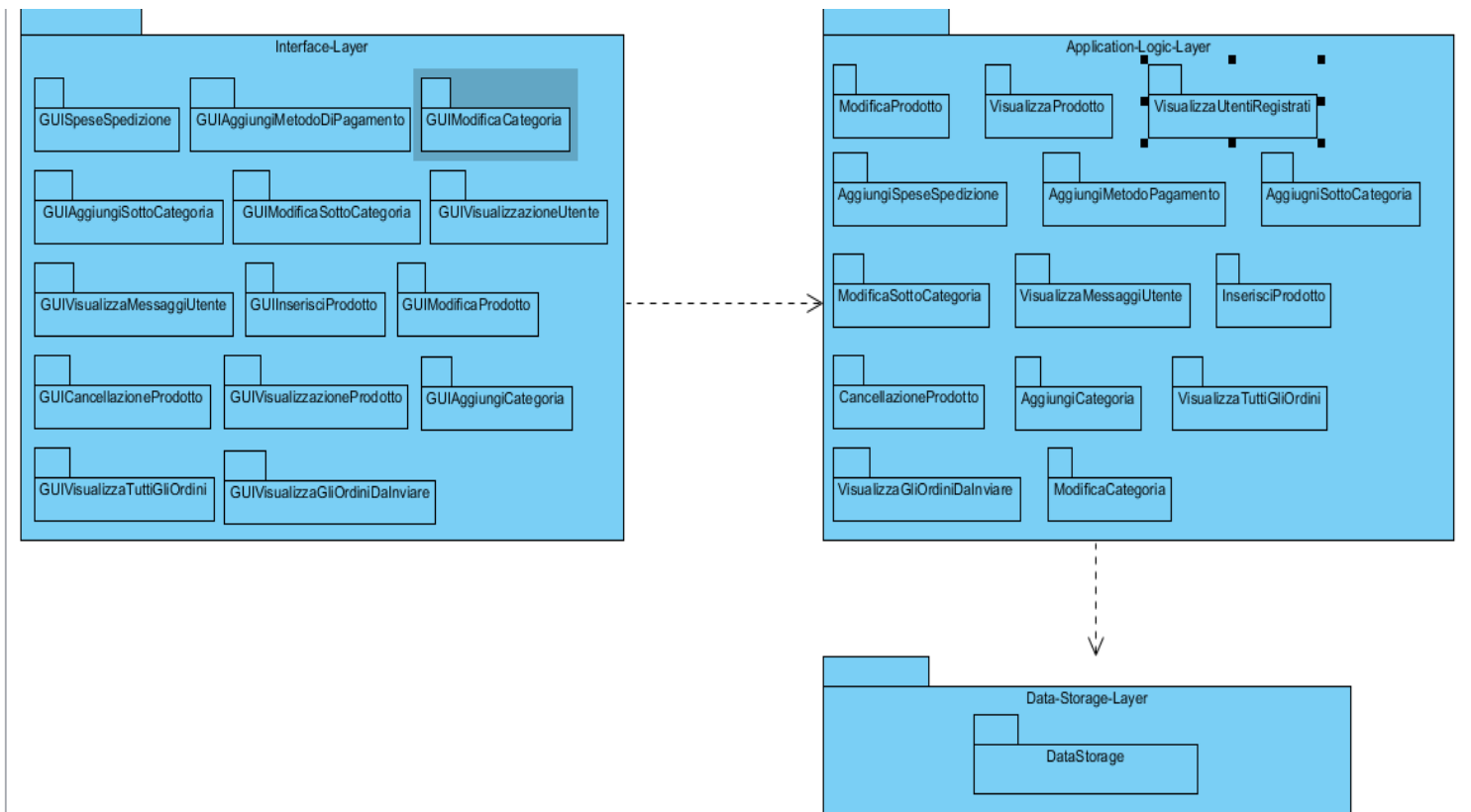


Interface Layer: GUIRicercaProdotto, GUIEmail, GUIAcquistoProdotto, GUICarrello, GUIVisualizzazioneProdotto e GUIAcquistoProdottoImmediato indicano i sottosistemi che contengono oggetti boundary;

Application Logic Layer: contiene sette sottosistemi individuati (ProdottoLista, RicercaProdotto, AcquistoProdotto, VisualizzaProdotto, Carrello, ImieiOrdini, AcquistoProdotto);

DataStorageLayer: è il sottosistema che si occupa di effettuare le operazioni verso il database.

Amministratore



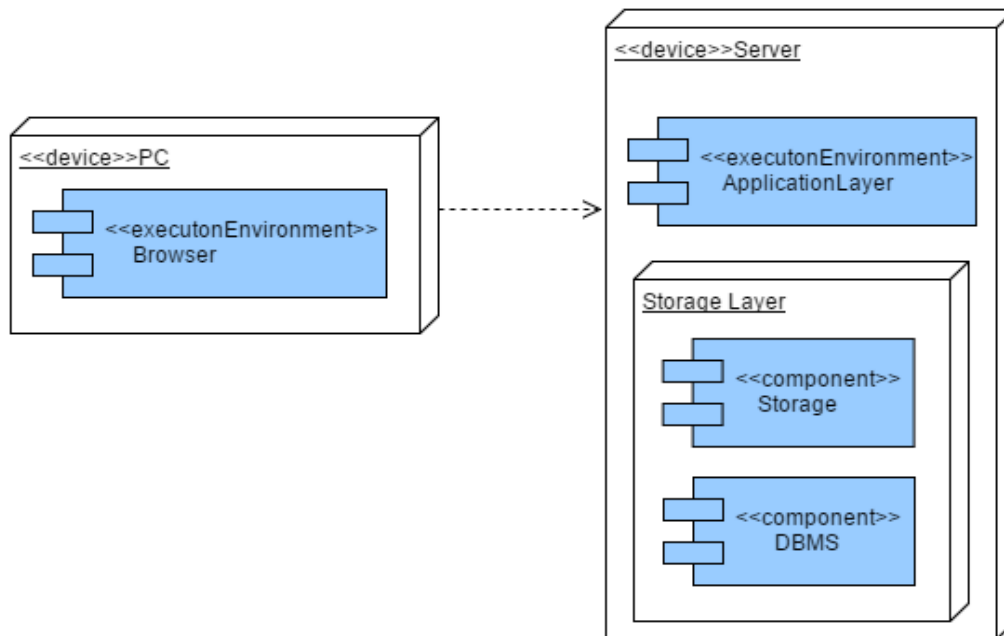
Interface Layer: GUISpeseDiSpedizione, GUIAggiungiMetodoDiPagamento, GUIModificaCategoria, GUIAggiungiSottoCategoria, GUIVisualizzazioneUtente, GUIVisualizzaMessaggiUtente, GUIInserisciProdotto, GUIModificaProdotto, GUICancellazioneProdotto, GUIVisualizzazioneProdotto, GUIAggiungiCategoria, GUIVisualizzaTuttiGliOrdini, GUIVisualizzaGliOrdiniDaInviare.

Application Logic Layer: contiene quattordici sottosistemi individuati (SpeseDiSpedizione, AggiungiMetodoDiPagamento, ModificaCategoria, AggiungiSottoCategoria, VisualizzaUtentiRegistrati, VisualizzaMessaggiUtente, InserisciProdotto, ModificaProdotto, CancellazioneProdotto, VisualizzaProdotto, AggiungiCategoria, VisualizzaTuttiGliOrdini, VisualizzaGliOrdiniDaInviare);

DataStorageLayer: è il sottosistema che si occupa di effettuare le operazioni verso il database.

4.4 Hardware/ software mapping

Il sistema sviluppato sarà installato su un solo server e utilizzerà un DBMS MySQL per la gestione dei dati persistenti. Il sistema è diviso con un'architettura di tipo client-server. Questo tipo di diagramma mette in evidenza le componenti utilizzate all'interno del sistema per aiutare lo sviluppatore a capire al meglio la relazione tra le componenti hardware e software.

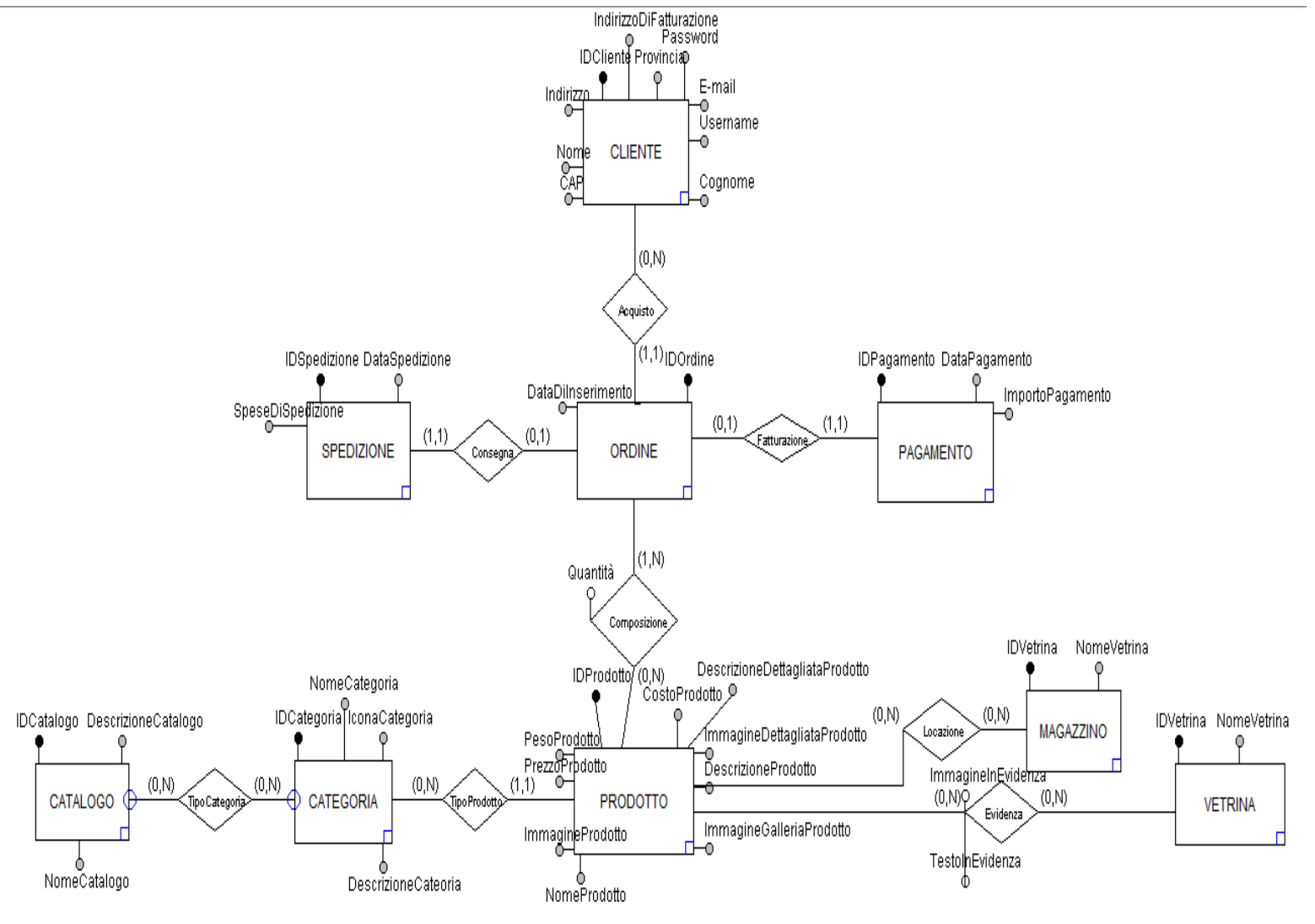


4.5 Persistent data management

L'applicazione utilizza servizi di storage su Database e le risorse saranno salvate sul server tramite apposite cartelle.

Il DBMS da noi utilizzato è MySQL .

4.6 Schema ER



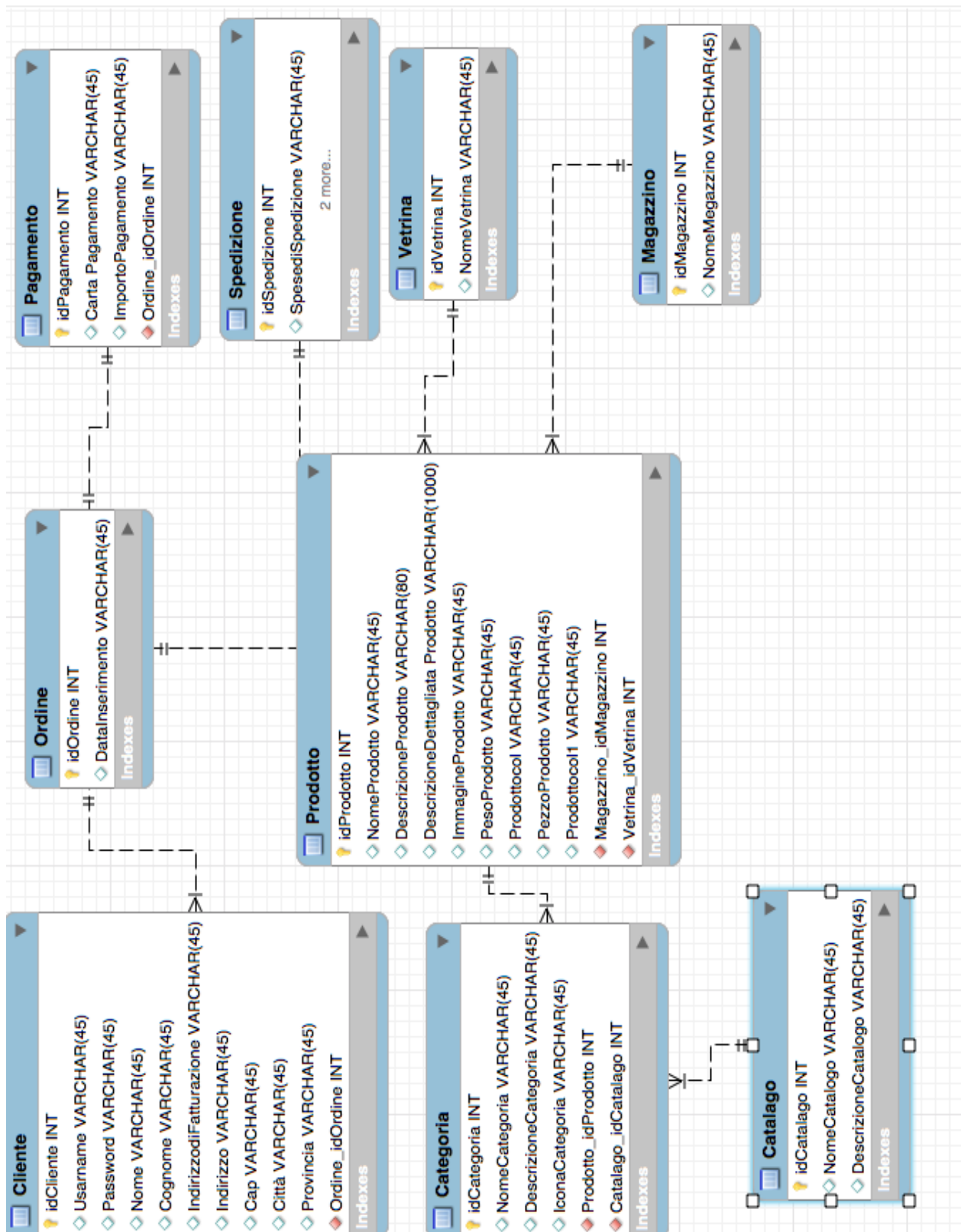
4.7 Mapping

ENTITA'	DATI
Cliente	<u>IDCliente</u> , Username, Password, E-mail, Telefono, Nome, Cognome, IndirizzoDiFatturazione, Indirizzo, CAP, Città, Provincia, ID Ordine.
Ordine	<u>IDOrdine</u> , Data Inserimento.
Spedizione	<u>ID Spedizione</u> , Data Spedizione, ID Ordine, Data Di Inserimento, Spese Di Spedizione.
Pagamento	<u>ID Pagamento</u> , Carta Pagamento, Importo Pagamento, ID Ordine, Data Di Inserimento.
Catalogo	<u>ID Catalogo</u> , Nome Catalogo, Descrizione Catalogo.
Categoria	<u>ID Categoria</u> , Nome Categoria, Descrizione Categoria, Icona Categoria, ID Catalogo, ID Prodotto.
Prodotto	<u>ID Prodotto</u> , Nome Prodotto, Descrizione Prodotto, Descrizione Dettagliata Prodotto, Immagine prodotto, Peso Prodotto, Immagine Dettagliata Prodotto, Immagine Galleria Prodotto, Prezzo Prodotto, Peso Prodotto, Costo Prodotto, ID Magazzino, ID Vetrina.
Magazzino	<u>ID Magazzino</u> , Nome Magazzino.
Vetrina	<u>ID Vetrina</u> , Nome Vetrina, ImmagineinEvidenza, TestodiEvidenza.

4.8 Dizionario dei dati

Entità	Descrizione	Attributi	Identificatore
Cliente	Utente registrato sul sistema	Username, Password, E-mail, Telefono, Nome, Cognome, IndirizzoDiFatturazione, Indirizzo, CAP, Città, Provincia, ID Ordine.	IdCliente
Ordine	Elenco degli ordini	Data Inserimento	<u>IDOrdine</u>
Spedizione	Elenco delle spedizioni effettuate	Data Spedizione, ID Ordine, Spese Di Spedizione.	<u>ID Spedizione</u>
Pagamento	Elenco dei pagamenti effettuati	Carta Pagamento, Importo Pagamento, ID Ordine.	<u>ID Pagamento</u>
Catalogo	Elenco dei cataloghi	Nome Catalogo, Descrizione Catalogo.	<u>ID Catalogo</u>
Categoria	Elenco delle categorie pubblicate	Nome Categoria, Descrizione Categoria, Icona Categoria, ID Catalogo, ID Prodotto	<u>ID Categoria</u>
Prodotto	Elenco dei prodotti disponibili	Nome Prodotto, Descrizione Prodotto, Descrizione Dettagliata Prodotto, Immagine prodotto, Peso Prodotto, Immagine Dettagliata Prodotto, Immagine Galleria Prodotto, Prezzo Prodotto, Peso Prodotto, ID Magazzino, ID Vetrina.	<u>ID Prodotto</u>
Magazzino	Elenco dei magazzini	Nome Magazzino.	<u>ID Magazzino</u>
Vetrina	Elenco delle vetrine	Nome Vetrina.	<u>ID Vetrina</u>

4.9 Modello logico



4.9.1 Access control and security

Ogni utente o amministratore può collegarsi al sistema grazie al controllo degli accessi e tramite l'inserimento dell'e-mail e della password (che saranno richieste ad ogni singolo accesso dell'utente). La sessione termina quando l'utente effettua il logout.

Se l'accesso al sistema non produce un risultato positivo, verrà inviata una notifica di errore all'interno del form di inserimento delle proprie credenziali (e-mail e password).

Se, invece, l'utente non ha effettuato il login all'interno del sistema o non si è mai registrato, potrà usufruire dei servizi messi a disposizione dell' "utente generico" come la visualizzazione dei prodotti, la ricerca di questi ultimi, ed un eventuale richiesta da parte dell'utente tramite un "commento".

Il sistema fornisce più viste a seconda del tipo di utente che si interfaccia con il sistema, in modo che ognuno potrà accedere alle proprie funzionalità.

Sottosistema Attori	Gestione		
	Account	ServiziUtenti	Amministratore
Amministratore	ModificaProfilo() SessionUser() RimuoviAccount())	AcquistoProdotto() AcquistiEffettuati() AcquistoDiretto() CarrelloServlet() SchedaProdotto() Ricerca()	AggiungiSpeseSpedizioni() Catalogo() ListaUtenteServlet() ModificaCategoriaS() ModificaProdottoS() ModificaQuantita() ModificaSottoCS() TuttiGliOrdini() VisualizzaMessaggiClienti() VisualizzaMetodoDiPagamento()
Utente Registrato	ModificaProfilo() SessionUser() RimuoviAccount())	AcquistoProdotto() AcquistiEffettuati() AcquistoDiretto() CarrelloServlet() SchedaProdotto() Ricerca() Contattaci()	
Utente Generico	FormRegistrazione()	SchedaProdotto() Ricerca() Contattaci()	

4.9.2 Global software control

Il server è sempre in funzione ed aspetta che l'utente gli invii una o più richieste da poter gestire contemporaneamente; nel caso in cui ci sia un errore, il sistema dovrà risolverlo il prima possibile.

4.9.3 Boundary conditions

Il server è sempre attivo e in caso di errore l'applicazione garantisce una tolleranza degli errori, cercando di risolverli nel minor tempo possibile.

4.9.4 Terminazione

Quando la pagina web verrà chiusa, senza effettuare il logout, si ha la terminazione del sistema con un regolare logout.

4.9.5 Fallimento

Per gli errori visualizzati dall'utente, verrà visualizzata una pagina che ci avverte dell'errore e il prima possibile dovranno essere risolti.

4.9.6 Use Case condition limits

Sono qui riportate alcune condizioni limite a cui il sistema può andare incontro:

4.9.6.1 - *StartServer*

Nome Use Case	<i>StartServer</i>
Partecipanti	Amministratore Primario Vuole avviare il sistema
Condizioni d'ingresso	Il server è spento
Flusso degli Eventi	<ol style="list-style-type: none">1. L'amministratore del Sistema accede alla macchina su cui è installato il sistema.2. L'amministratore avvia il server.3. Il server segnala che l'avvio è avvenuto con successo e i suoi servizi vengono messi a disposizione degli utenti.
Condizioni d'uscita	Il server è stato avviato
Requisiti di qualità	Il sistema notifica l'avvenuto avviamento del server

4.9.6.2 - *Terminazione*

Nome Use Case	ShutDownServer
Partecipanti	Amministratore Primario Vuole arrestare il sistema
Condizioni d'ingresso	Il server è acceso

Flusso degli Eventi	<ol style="list-style-type: none"> 1. L'amministratore del Sistema accede alla macchina su cui è installato il sistema. 2. L'amministratore esegue la terminazione del server. 3. Il Sistema comunica ai client la sua terminazione. 4. Il server si spegne.
Condizioni d'uscita	Il server è stato arrestato
Requisiti di qualità	Il sistema notifica l'avvenuto terminazione del server

4.9.6.3 *Crash Server*

Nome Use Case	ServerCrash
Partecipanti	Sistema, Amministratore Primario
Condizioni d'ingresso	Il server è in crash
Flusso degli Eventi	<ol style="list-style-type: none"> 1. La schermata del client visualizza un messaggio di errore che comunica il guasto del sistema e che tutti i servizi di quest'ultimo non saranno disponibili fino a quando il problema tecnico non sarà risolto. 2. L'Amministratore Primario provvede al ripristino del Server manualmente. 3. Il Server è ripristinato dal punto in cui ha avuto il crash utilizzando i punti di ripristino precedentemente creati. 4. Il Sistema comunica la sua disponibilità ai client ad offrire i propri servizi.
Condizioni d'uscita	Il server è stato ripristinato.
Requisiti di qualità	Il sistema notifica l'avvenuto ripristino del server.

5.0 Servizi dei Sottosistemi

Abbiamo diviso il sistema in due interfacce:

- **GUIUtente:** Interfaccia che gestisce l'interazione dell'utente con il sistema.
- **GUIAdmin:** Interfaccia che gestisce l'interazione dell'amministratore con il sistema.

Si è deciso di dividere il sistema in tre sottosistemi.

Account: offre i propri servizi agli utenti non registrati, tra cui la ricerca dei prodotti, la creazione del proprio account. L'utente registrato e l'amministratore, oltre alle funzionalità che ereditano dall'utente non registrato, possono effettuare il login, il logout, può modificare i propri dati o rimuoverli dal sistema.

ServiziUtenti: offre i propri servizi agli utenti non registrati, tra cui la ricerca di un prodotto, la sua visualizzazione oppure può inviare un commento al sistema. L'utente registrato e l'amministratore, oltre alle funzionalità che ereditano dall'utente non registrato, possono visualizzare il proprio carrello ed effettuare acquisti direttamente o tramite carrello e visualizzare i propri ordini.

Amministratore: offre i propri servizi solo all'Amministratore, tra cui l'aggiunta delle spese di spedizioni e dei metodi di pagamento, la visualizzazione degli utenti e degli ordini da inviare, può aggiungere o modificare i prodotti, una categoria e una sottocategoria. Può inoltre visualizzare i commenti inviati dai vari utenti.

6.0 Glossario

Dependability criteria: quantificazione dello sforzo utilizzato per risolvere problemi di crash del sistema.

Performance criteria: requisiti imposti sul sistema in termini di spazio e velocità.

Query: interrogazione di un database per estrarre o aggiornare i dati.

Maintance criteria: requisiti imposti sul sistema in termini di mantenibilità.

End user criteria: qualità non incluse nei criteri di performance e affidabilità.

Trade-offs: compromessi.

Persistent data management: strategie relative alla gestione dei dati persistenti.

Schema ER: è un modello per la rappresentazione concettuale dei dati.

Dizionario dei dati: permette di inserire all'interno dello schema ER altre informazioni riguardanti la descrizione delle entità.

Tavola dei volumi: specifica il numero stimato di istanze per ogni entità e associazione.

Modello logico: è il modello che discende da quello concettuale e disegna un'architettura che tiene conto delle proprie strutture su quel particolare database.

Boundary conditions: soluzioni a problemi fisici.