



Università degli Studi di Salerno

Corso di Ingegneria del Software

Quick Service System Design Document



Data <5/05/2008>

MEMBRI DEL QUICK SERVICE:

Team Members:

Vicidomini Vincenzo

Mercurio Antonio

Iacoletti Alessandro

Pacifico Marta

Pizza Luca Ernesto

INDICE DEI CONTENUTI

1.	INTRODUZIONE	4
1.1	SCOPO DEL SISTEMA	4
1.2	OBIETTIVI DI DESIGN.....	4
1.3	DEFINIZIONI, ACRONIMI ED ABBREVIAZIONI.....	6
1.4	RIFERIMENTI	6
1.5	OVERVIEW	7
2.	SISTEMA SOFTWARE CORRENTE	7
3.	SISTEMA SOFTWARE PROPOSTO	8
3.1	OVERVIEW	8
3.2	DECOMPOSIZIONE IN SOTTOSISTEMI	11
3.3	HARDWARE/SOFTWARE MAPPING	12
3.4	GESTIONE DATI PERSISTENTI	18
3.5	CONTROLLO DEGLI ACCESSI E SICUREZZA	18
3.6	CONTROLLO GLOBALE DEL SOFTWARE.....	19
3.7	CONDIZIONI BOUNDARY	20
4	SERVIZI DEI SOTTOSISTEMI.....	22
5.	GLOSSARIO	

1. Introduzione

1.1 Scopo del sistema

Gli obiettivi del sistema sono di realizzare un software per la gestione del servizio in camera presso una struttura.

Il progetto QUICK TOUCH SERVICE si adatta per le strutture che vogliono consentire una gestione facile, comoda e completa dei servizi volti alla clientela. Con un semplice click del mouse, o meglio ancora con una semplice pressione del monitor (touch screen), e con la rappresentazione grafica dei prodotti.

Il sistema sarà in grado di incrementare l'efficienza di un albergo, mediante meccanismi di gestione che si basano sull'utilizzo di un archivio centrale dove sono organizzate le informazioni.

In fase di design del software è stato necessario compiere delle scelte riguardanti i programmi da utilizzare per lo sviluppo del progetto.

1.2 Obiettivi di Design

I punti fondamentali che abbiamo preso in considerazione sono:

- Il linguaggio di programmazione;
- Portabilità;
- Interfaccia grafica;
- Base di dati;
- Prestazioni;

Il Linguaggio di Programmazione

In funzione della fase di sviluppo sarebbero preferibili una sintassi semplice e flessibile, semplicità e potenza nell'elaborazione di testo, disponibilità di documentazione e possibilità d'estensione ed interfacciamento verso altri linguaggi (JAVA).

Portabilità

Trattandosi di un sistema software gestionale di alto livello è desiderabile una marcata indipendenza dal sistema operativo. Peraltro non vi sono necessità imprescindibili che pongano ostacoli nel design di un sistema software portabile. Inoltre, sebbene il design e l'implementazione di un software multi-piattaforma possano risultare più lunghi e complessi, è a nostro avviso importante compiere subito un deciso passo verso un sistema di questo genere.

Interfaccia Grafica

Il sistema sarà semplificato da interfacce amichevoli, intuitive e di semplice utilizzo.

L'interfaccia utente sarà realizzata in modo da renderne immediato l'utilizzo anche da parte degli utenti meno esperti. Tutte le informazioni vengono presentate in maniera chiara ed intuitiva.

Base di Dati

Sia l'implementazione di un sistema proprio sia l'interfacciamento ad un DBMS presentano ovvi vantaggi e svantaggi. Considerando condizioni operative normali, con svariati record di diverso tipo memorizzati e la necessità di condividere la base di dati fra più client l'unica scelta possibile risulta l'interfacciamento con un DBMS.

Prestazioni

Ovviamente trattandosi di un software interattivo è necessaria una buona velocità di risposta alle richieste dell'utente. D'altra parte ipotizzando un ambiente di utilizzo tipico che coinvolge una base di dati condivisa, accessibile via rete, non bisogna trascurare la latenza ad essa dovuta, quindi è inutile esasperare l'aspetto prestazionale. E' comunque desiderabile che l'utilizzo del software non sia precluso alle macchine di fascia più bassa.

1.3 Definizioni, Acronimi ed Abbreviazioni

QUICK TOUCH SERVICE: Sistema software per la gestione del servizio in camera presso una struttura.

JDBC: Java DataBase Connection;

DBMS: Database Management System, Sistema di gestione del database.

Java: linguaggio di programmazione orientato agli oggetti.

1.4 Riferimenti

- **Object Software Engineering – Using UML, Patterns and Java** *B. Bruegge, A. H. Dutoit.*
- **Rad**
- **SoftwareProjectManagementPlan**

1.5 Overview

Nei successivi paragrafi del documento di System Design verranno descritti:

- **Sistema Software Corrente:** rappresenta l'architettura del software che è stato sostituito o, se esso non esiste, descrive l'architettura dei software per sistemi simili esistenti.
- **Sistema Software proposto:** documenta il modello di System Design del nuovo sistema, in cui vengono descritti i seguenti elementi:
 - Decomposizione in sottosistemi, nella quale viene descritta la suddivisione del sistema in vari sottosistemi. Le operazioni che essi svolgono vengono descritte ad alto livello senza entrare troppo nello specifico.
 - Hardware/Software mapping, in cui vengono prese alcune decisioni per quanto riguarda le piattaforme hardware e software su cui il sistema dovrà girare, una volta decise le piattaforme è necessario mappare le componenti su di esse. Questa operazione potrebbe portare all'introduzione di nuove componenti per interfacciare i sottosistemi su diverse piattaforme.
 - Gestione Dati Persistenti, in cui sono identificati gli oggetti persistenti e scelto il tipo di infrastruttura da usare per memorizzarli.
 - Controllo degli Accessi e Sicurezza, che descrive il modello utente del sistema in termini di matrici di accesso e i problemi di sicurezza, come la scelta di un meccanismo di autenticazione.
 - Controllo Globale del Software, che descrive come il controllo globale del software è implementato, come le procedure di richiesta sono avviate e come si sincronizzano i sottosistemi.
 - Condizioni Boundary in cui sono descritte le condizioni limite del sistema, le quali includono lo start-up, lo shutdown, l'inizializzazione e la gestione di fallimenti come corruzione di dati, caduta di connessione e caduta di componenti; a tale scopo saranno elaborati dei casi d'uso che specificano la sequenza di operazioni in ciascuno dei casi sopra elencati.
- **Servizi dei Sottosistemi:** descrive i servizi forniti da ciascun sottosistema in termini di operazioni.

2. Sistema Software Corrente

Attualmente esistono sistemi software con caratteristiche simili all'applicazione QUICK TOUCH SERVICE, ma il nostro scopo è quello di mirare alla creazione di un software che faciliti ulteriormente la gestione di strutture e infrastrutture strettamente connesse con le attività turistiche.

3. Sistema Software Proposto

3.1 Overview

Il tipo di architettura software che si è scelta è legata (e vincolata) all'analisi del sistema. Infatti, il sistema che si vuole realizzare è un sistema distribuito nel quale esiste un elemento centrale contenente l'archivio dati della libreria a cui i vari attori dalle loro postazioni vogliono accedere. Quindi esistono dei moduli client che richiedono dati ad un modulo centrale che rappresenta il server. Fra loro i vari moduli non comunicano: l'unica comunicazione è rivolta da e verso il server. Naturalmente tali moduli risiedono in macchine differenti.

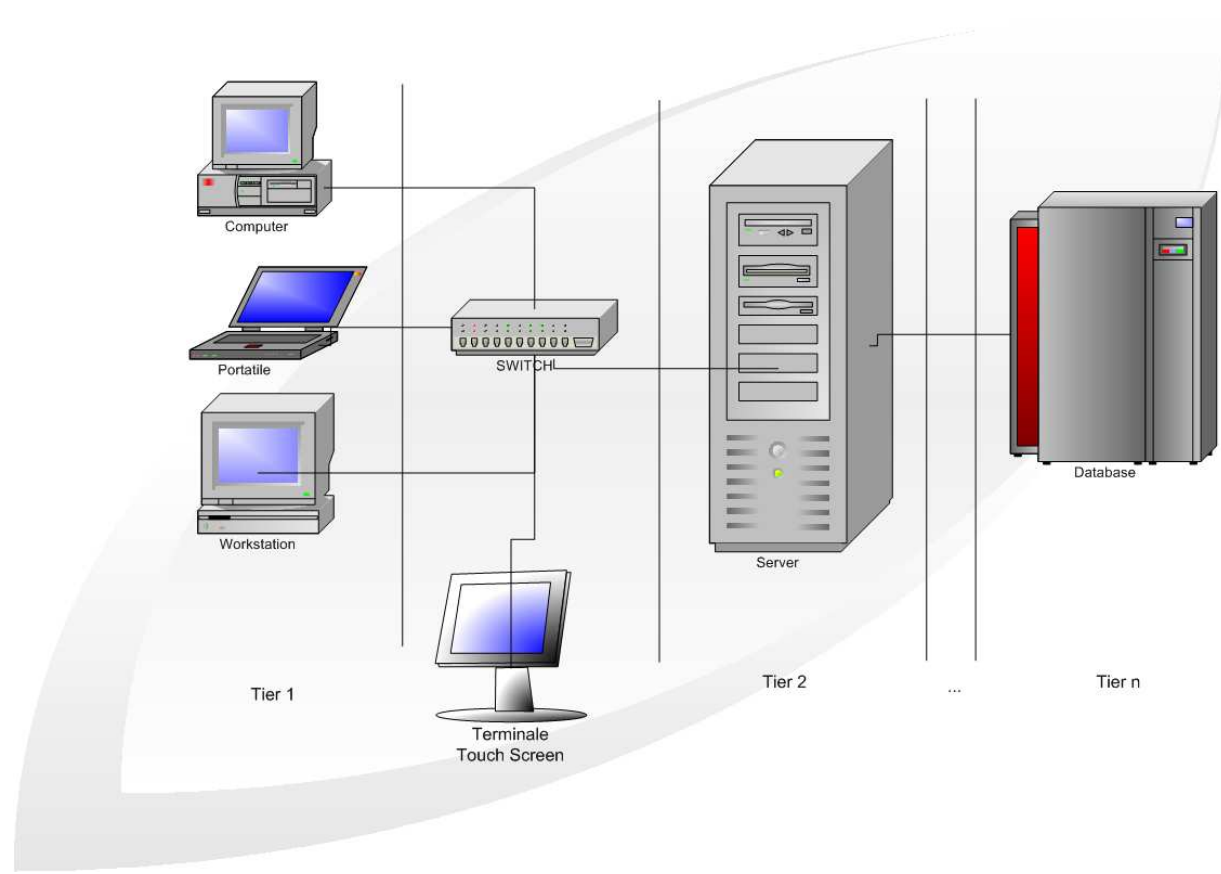
L'architettura scelta è pertanto quella client - server. Il server a sua volta fornisce la comunicazione dei dati con il database.

La comunicazione tra client e server avviene utilizzando l'architettura Three-Tier, evoluzione dell'architettura client/server.

Con riferimento a tali sistemi, infatti, possiamo osservare che esse bene si adattano al nostro sistema di accesso ai dati.

La necessità di accesso da parte del personale della libreria al sistema informativo sta alla base dell'organizzazione *n-tier*. Un'architettura *n-tier* è costituita da n sistemi che contribuiscono in successione, a portare a termine una computazione.

Nella Figura seguente è schematizzata tale struttura.



Tra ogni coppia di livelli esiste una relazione *client - server* e l'interazione viene propagata dal primo all'ultimo livello dell'architettura a viceversa. In ogni passo sono coinvolti programmi che si occupano di rispondere alle richieste eventualmente appoggiandosi ai livelli successivi. In ogni *tier* possono essere coinvolti più calcolatori. Le richieste vengono elaborate dai vari *tier* fino al loro completamento.

Il modello *n-tier* si è affermato proprio perché riesce a identificare "isole" di calcolo che possono essere associate alle reti su cui risiedono le risorse su cui un programma opera.

Le architetture *n-tier* riguardano la struttura di un sistema che è composto di elementi che nel loro piccolo sono del tutto standard. La ricchezza che deriva dalla struttura è principalmente una migliore ingegnerizzazione del sistema che porta ad una sua maggiore manutenibilità. Inoltre la conoscenza delle interfacce tra un livello e l'altro aiuta a rendere più flessibile il sistema, ad esempio accedendo più DB invece che uno solo per distribuire il carico senza che il livello intermedio si accorga del cambiamento. Le applicazioni *three-tier* ($n=3$) prevedono un primo livello

costituito dai clients, questi si connettono al secondo livello in cui vengono elaborate le richieste. Nel livello intermedio si gestisce l'elaborazione della richiesta accedendo a dati che risiedono sul terzo ed ultimo livello.

Client + Front-end + Back-end = three-tier

È importante osservare come l'interfaccia tra i livelli sia ben definita: questo aiuta a effettuare verifiche di sicurezza sulle comunicazioni. Inoltre è possibile separare il livello dei clients dai restanti due utilizzando un *firewall*.

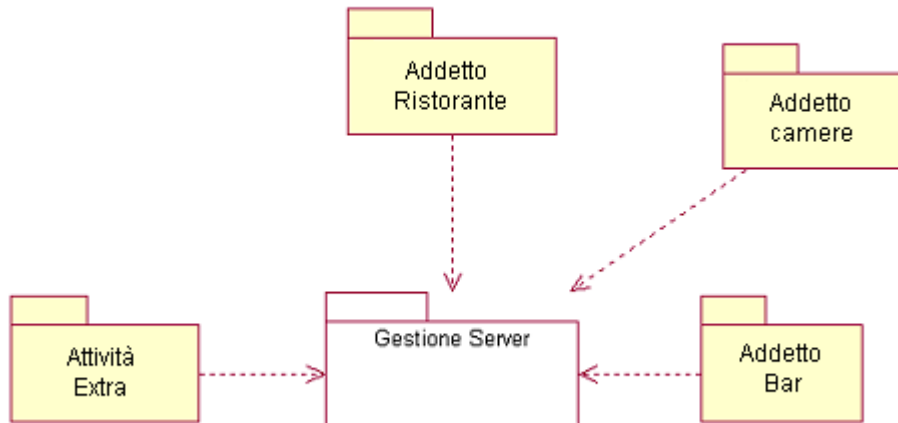
In questo progetto abbiamo realizzato una struttura 3-tier in cui c'è un database, un client ed un server che interagisce sia con il database sia con il client e che svolge il compito di modulo intermedio. Tale modulo intermedio, (rappresentato per noi dalla classe Server e dai driver JDBC-ODBC di Java), si occupa di inoltrare le richieste dell'utente al livello archivio dati e di trasmettere i relativi risultati.

Il principale vantaggio di tale scelta è di rendere il client del tutto indipendente dalle problematiche di accesso ai dati, rendendo il sistema aggiornabile e mantenibile soprattutto per quel che riguarda la parte del sistema che si occupa dell'accesso all'archivio dei dati.

Resta da definire il tipo di protocollo utilizzato per la comunicazione client - server. Si è scelto l'uso del protocollo UDP. Questo protocollo offre un servizio di comunicazione fra client e server senza connessione evitando così la fase di handshaking.

Il sistema è composto da sei componenti principali. Riportiamo il diagramma dei package principali del sistema ed una breve descrizione di ogni package.

3.2 Decomposizione dei sottosistemi



I sottosistemi indicati sono in esecuzione su macchine diverse (La parte cliente è replicata su più terminali disposti nelle camere, mentre le altre parti sono disposte nei vari settori dell'albergo) e comunicano (tutti col sottosistema server) in maniera remota.

Il sistema è stato diviso *orizzontalmente* in tre strati:

- Sottosistema **Presentation Layer**: visualizza all'utente gli oggetti del dominio applicativo. Fanno parte di questo strato gli oggetti Boundary.
- Sottosistema **Application Layer**: responsabile della sequenza di interazioni con l'utente e di notificare ai Presentation Layer i cambiamenti nel modello. Fanno parte di questo strato gli oggetti Control.
- Sottosistema **Data Layer**: mantiene la conoscenza del dominio applicativo. Fanno parte di questo strato gli oggetti Entity.

Il sottosistema **Presentation Layer** è stato diviso in quattro sottosistemi principali. Le partizioni fanno riferimento ai quattro principali attori che accedono al sistema:

- Sottosistema **AddettoBar**: Include tutte le interfacce grafiche a cui l'Addetto Bar può accedere e cioè: `visualizza_ordinazione`, `modifica_ordinazione`, `nuovo_ordine`, `cancella_ordine`, `visualizza_listino`, `inserimento_prodotto`, `modifica_prodotto`, `cancella_prodotto`.
- Sottosistema **AddettoRistorante**: Include tutte le interfacce grafiche l'Addetto Ristorante può accedere e cioè: `annullamento_ordinazione`, `visualizza_ordinazione`,

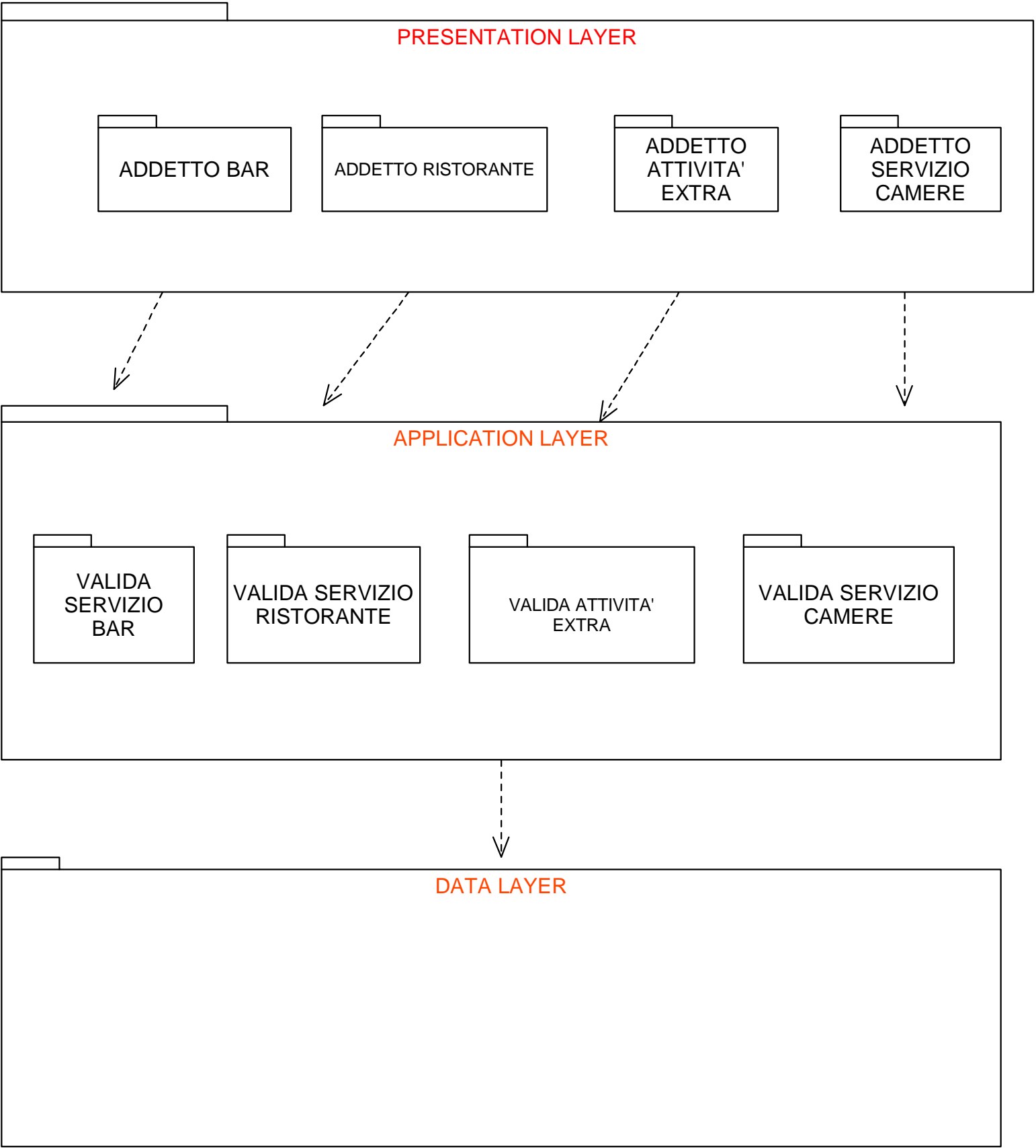
modifica_ordinazione, prenotazione_menù, visualizza_saldo, visualizza_menù, modifica_menù, inserimento_menù, consultazione_prenotazione.

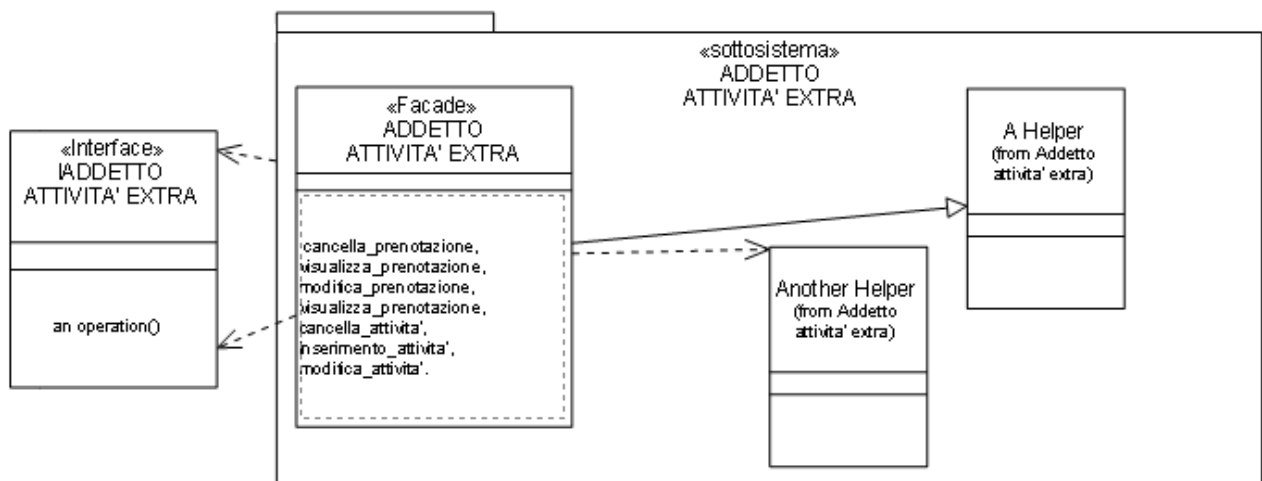
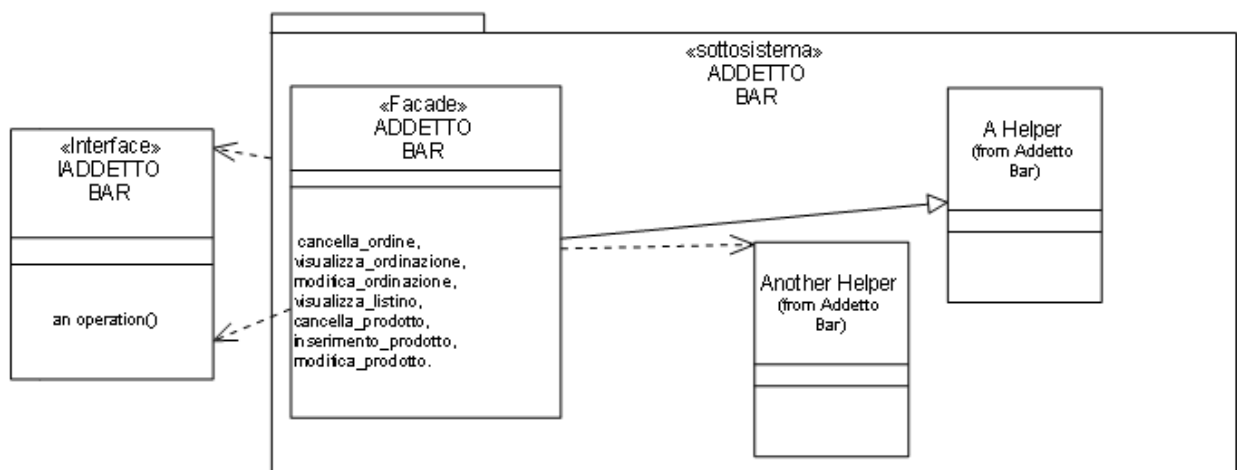
- Sottosistema **AddettoAttivitàExtra**: Include tutte le interfacce grafiche a cui l'Addetto Attività Extra può accedere e cioè: annullamento_prenotazione, visualizza_prenotazione, modifica_prenotazione, nuova_prenotazione, resoconto_prenotazione, modifica_attività, cancella_attività, inserimento_attività.
- Sottosistema **AddettoServizioCamere**: Include tutte le interfacce grafiche a cui l'Addetto Servizio può accedere e cioè: libera_camera, visualizza_camera, modifica_camera, assegna_camera.

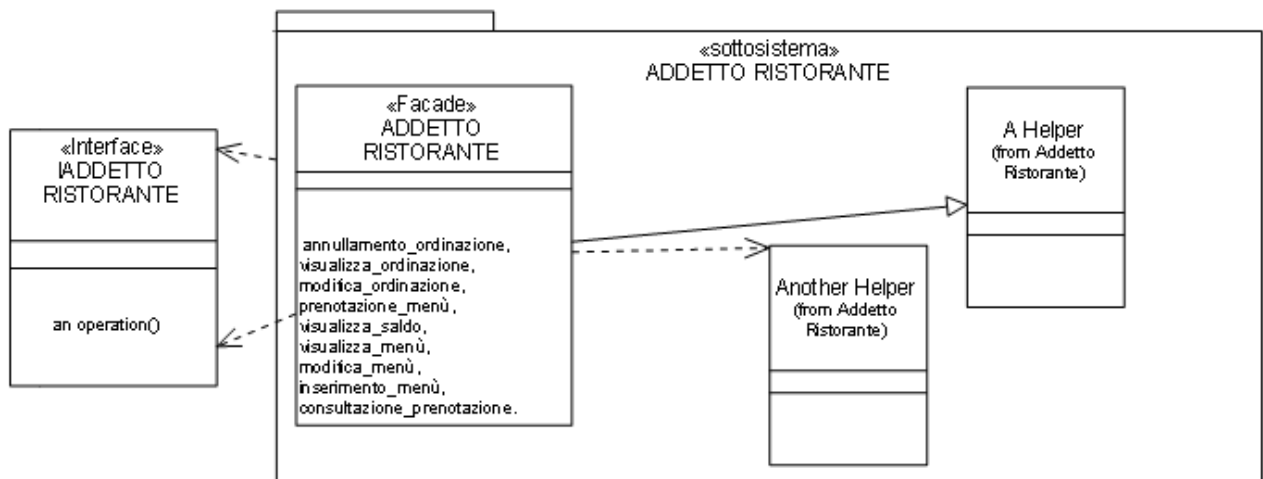
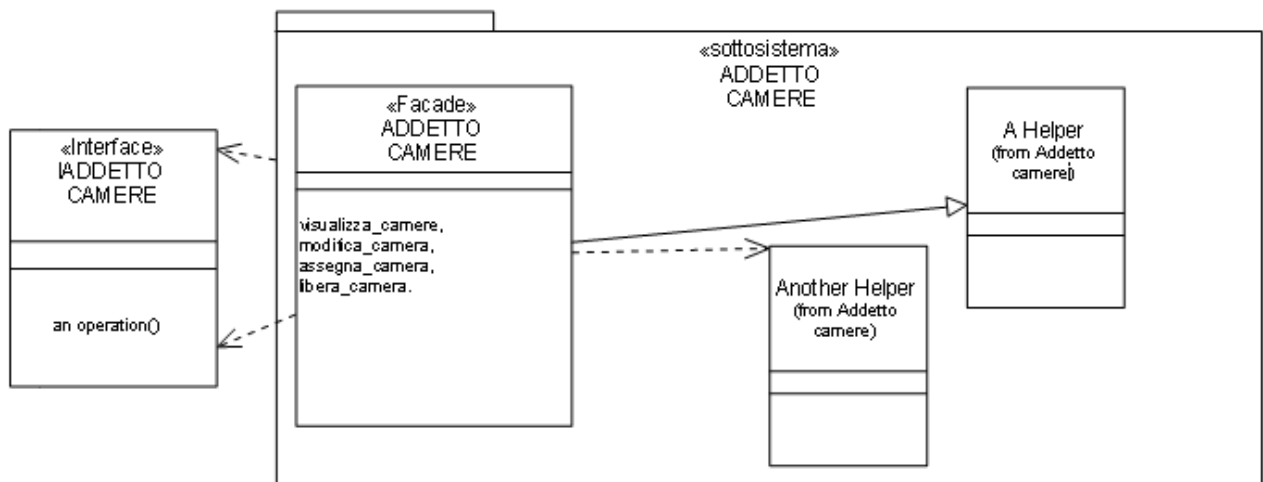
Il sottosistema ***Application Layer*** è stato diviso in cinque sottosistemi principali. Le partizioni fanno riferimento alle cinque principali entità a cui l'utente accede:

- Sottosistema **ValidaServizioBar**: In esso sono incluse tutte le unità di controllo che accedono all'entità Addetto Bar e cioè: login, logout, visualizza/modifica prodotti, aggiungi prodotti, cancella prodotto.
- Sottosistema **ValidaServizioRistorante**: In esso sono incluse tutte le unità di controllo che accedono all'entità Addetto Ristorante e cioè: login, logout, visualizza/modifica menù, aggiungi menù, cancella menù.
- Sottosistema **ValidaAttivitàExtra**: In esso sono incluse tutte le unità di controllo che accedono all'entità Addetto Attività Extra e cioè: login, logout, visualizza/modifica attività, aggiungi attività, cancella attività.
- Sottosistema **ValidaServizioCamere**: In esso sono incluse tutte le unità di controllo che accedono all'entità Addetto Camere e cioè: login, logout, visualizza/modifica prenotazione camera, cancella prenotazione camera.

La decomposizione in sottosistemi descritta è illustrata nella seguente figura







3.3 Hardware-Software mapping

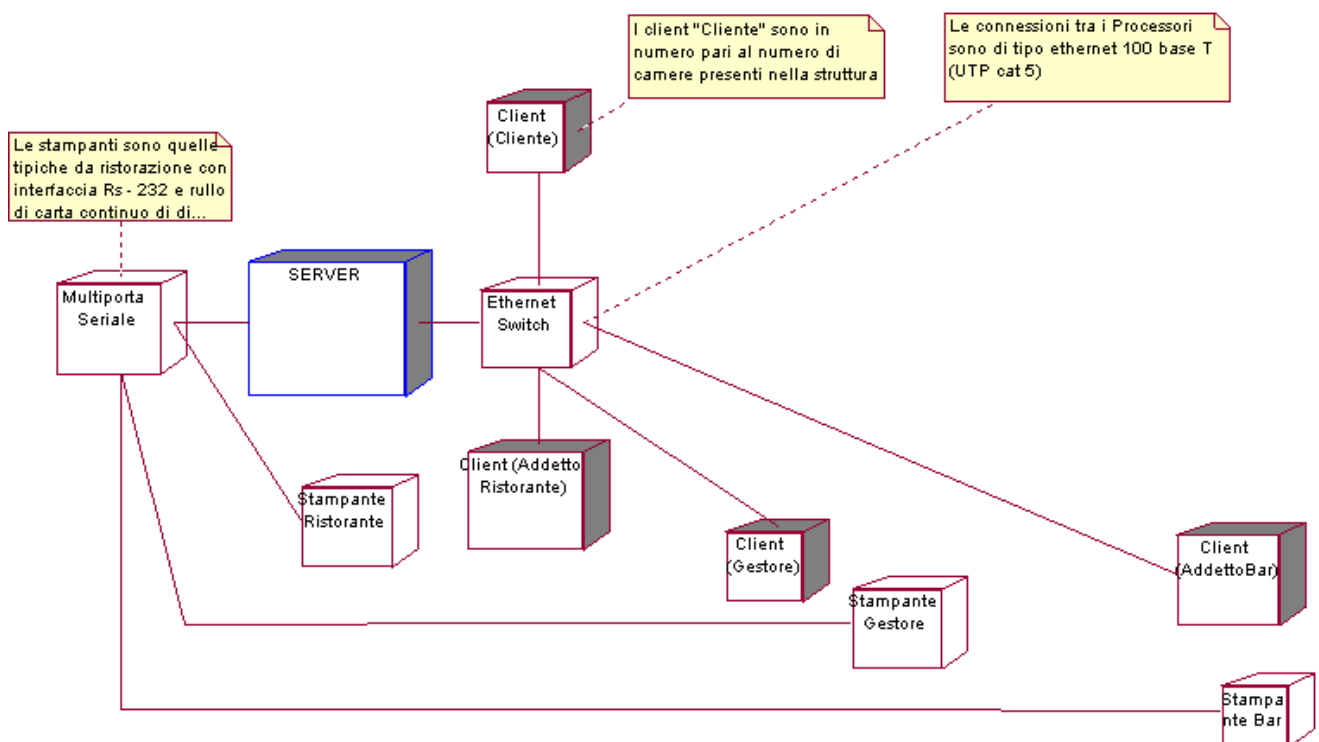
L'intero sistema risulta essere composto da quattro componenti software principali: le cinque interfacce grafiche, le cinque interfacce di controllo, la gestione comunicazione e la gestione dati.

Tali componenti non risiedono sulla stessa macchina ma su macchine differenti.

(Le interfacce Grafiche e le interfacce di controllo illustrate in ciascun package sono accoppiate ovviamente su una sola macchina).

I componenti sono inoltre indipendenti gli uni dagli altri tranne che per il server che è l'unico componente che comunica con gli altri. A parte questa comunicazione gli altri componenti sono fra loro isolati, ovvero non si scambiano messaggi direttamente, ma alcune delle componenti possono ricevere dal SERVER informazioni sullo stato di altre. Mostriamo di seguito come tale struttura venga modellata mediante l'utilizzo del diagramma di deployment.

Diagramma di Deployment del sistema.



L'intero programma è gestito da un server al quale sono collegati quattro client e le rispettive stampanti.

I client sono collegati al server tramite uno “Switch Ethernet” che collega i processori dell’Addetto Ristorante, dell’Addetto Bar, del Gestore e dei Clienti della struttura alberghiera (uno per ogni camera d’albergo). I compiti principali sono svolti dal server che ha il compito di immagazzinare dati, gestire le ordinazioni e le varie operazioni svolte dagli impiegati dell’albergo e soprattutto dei clienti.

Come detto in precedenza, al server è inoltre collegata una “Multiporta seriale” che permette di collegare le varie stampanti utili agli addetti e al gestore per avere resoconti e informazioni riguardanti ad esempio il resoconto delle prenotazioni per un’attività proposta dall’albergo o il saldo di un cliente.

Le stampanti sono quelle tipiche per strutture di questo tipo con interfaccia “RS 232” e rullo di carta continuo che rende più semplice e immediata la stampa delle informazioni.

3.4 Gestione Dati Persistenti

I dati persistenti del sistema rappresentano le informazioni relative al sottosistema Storage. Esso contiene i DatiCliente, DatiUtente, DatiCamere, DatiProdottiBar, DatiAttivitàExtra, DatiMenù, DatiPortateMenù, DatiTipoPortate, DatiCategorie, DatiPrenotazione, DatiDettagliPrenotazioneAttivitàExtra, DatiDettagliPrenotazioneBar, DatiDettagliPrenotazioneRistorante.

Per la memorizzazione di tali dati viene scelto un database relazionale. La scelta è giustificata dal fatto che sugli attributi vengono effettuate un numero cospicuo di query complesse.

3.5 Controllo degli Accessi e Sicurezza

L'utente deve essere autenticato, e l'operazione di login viene eseguita tramite la compilazione del form che compare all'accensione della macchina. Nei casi di login fallito, ogni sottosistema ripropone il form di login. Dato che i compiti svolti dai vari "attori" nel nostro sistema sono differenti ed accedono ad informazioni diverse dell'archivio al fine di evitare:

- la visualizzazione di dati specifici per un certo compito e non utilizzati per altre mansioni;
- la modifica di dati fondamentali per la gestione dell'albergo stesso;
- e che le attività svolte dagli impiegati dell'albergo non possano essere ricondotte ad una specifica persona;

Si è pensato di utilizzare più meccanismi di protezione dei dati e delle attività eseguite.

Il primo livello di protezione è un filtro che si è realizzato nell'accesso e nella modifica dei dati e che viene posto a livello delle funzionalità offerte ad ogni impiegato dal sistema in esecuzione sul proprio terminale, secondo il seguente schema:

1. **Cliente:** può accedere solo in lettura ai dati relativi ai servizi dell'albergo.
Può però anche inserire nuovi dati nel database quando ad esempio decide di ordinare qualcosa al bar oppure desidera prenotarsi per un'escursione

2. **Addetto Bar/Ristorante:** può accedere ai dati del magazzino per visualizzare tutti i dati relativi ai prodotti, agli ordini emessi, all'insieme di tutte le transazioni ed al menu ristorante.

Si è scelto di dare all'addetto bar/ristorante delle funzioni di sola consultazione per isolare l'insieme delle sue attività da quelle di modifica dei dati.

3. **Gestore:** può accedere ad ogni informazione dell'archivio dati per consultazione e per modifiche.

Un secondo livello di protezione viene garantito durante la fase di accesso al sistema dalla necessità di dover inserire una password ed il proprio ID.

Questa operazione preliminare deve essere svolta da tutti gli utilizzatori del sistema ad eccezione del cliente.

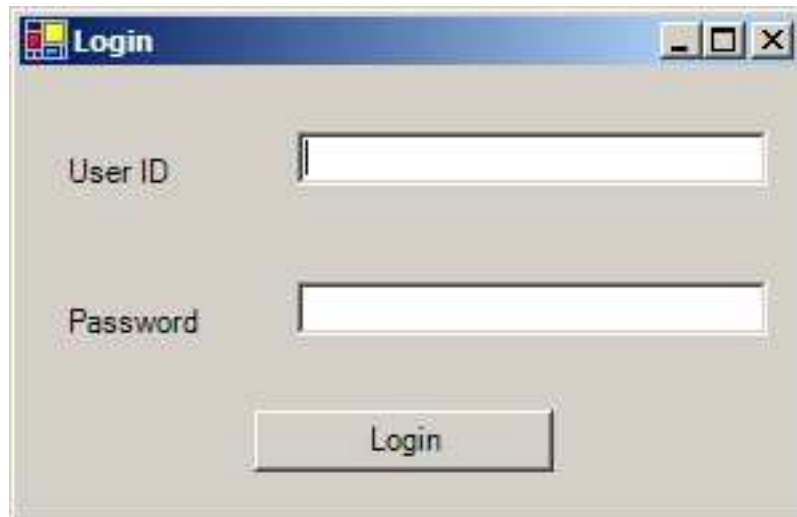
L'utilizzo delle password insieme all'ID garantisce inoltre la possibilità di poter ricondurre ogni operazione di modifica dei dati ad un impiegato.

La gestione delle password inoltre viene effettuata solamente dal Gestore.

3.6 Controllo Globale del Software

Si è ritenuto opportuno utilizzare una gestione del *controllo centralizzata* di tipo **event-driven**. Il sistema prevede una serie di bottoni, form e notifiche che realizzano l'interfaccia grafica. Prevede l'accesso solamente ai terminali predisposti su cui è installato il client di QuickService e collegati alla rete locale, ovvero i terminali installati nelle camere e le postazioni del bar, del ristorante e del gestore. L'accesso a funzionalità di inserimento, modifica e cancellazione di servizi offerti quali prodotti del bar, menu del ristorante, attività extra o gestione camere è consentita solamente dopo che l'utente ha effettuato l'autenticazione, questo assicura che tali operazioni sono svolte solamente dal personale autorizzato.

Per effettuare l'autenticazione il sistema mostra una maschera che richiede l'inserimento del login name e della password.



l'utente inserisce i dati e preme il bottone per la convalida, viene invocato il metodo remoto di convalida che prende i dati inseriti dall'utente controlla la corrispondenza nel database e consente l'accesso se i dati sono corretti altrimenti l'accesso sarà negato.

Per il cliente non è prevista alcuna autenticazione, infatti, può accedere alle funzionalità del sistema in qualsiasi momento, l'utente è riconosciuto in base all'identificativo del client presente nella camera.

3.7 Condizioni Boundary

- **Inizializzazione**

Data la natura distribuita del sistema proposto l'avvio e l'esecuzione delle sue varie parti ossia dei sottosistemi che forniscono le varie funzionalità ai vari "attori" può avvenire indipendentemente.

I sottosistemi quindi non hanno bisogno di accedere a nessun dato, poiché i client conoscono già la loro identità nel sistema.

All'avvio, se si sta avviando un sottosistema che riguardante la gestione (Gestione Ordinazioni Ristorante, Gestione Prenotazioni Attività extra, ecc...), il sistema richiederà una login ed una password e poi alla visualizzazione principale; altrimenti, se l'utente è il cliente dell'albergo, si aprirà subito la visualizzazione principale.

L'interfaccia grafica è molto semplice ed intuitiva, da questa si può accedere a tutti i servizi offerti a quel determinato utente, infatti, essa varia a seconda dei servizi offerti, che sono diversi per ogni tipo di utente (Addetto bar, Addetto Camere, Cliente).

- **Terminazione**

Per il corretto funzionamento di un sottosistema non è richiesto che gli altri siano attivi, quindi se uno di essi termina non intralchierebbe il lavoro degli altri sottosistemi fatta eccezione che per il server. E' richiesto, infatti, che sia prima avviato il server affinché una qualunque postazione client possa usufruire dei servizi offerti da esso. Sebbene questa sia una condizione indispensabile per il corretto funzionamento del sistema (nella sua interezza o in parte) se si verifica l'avvio di un sottosistema senza che il server sia attivo si avrà comunque una gestione di tale situazione da parte del sottosistema stesso.

La terminazione si realizza soltanto quando tutti i sottosistemi sono stati disattivati, infatti, il sistema resta comunque attivo anche se solo due sottosistemi (di cui uno è il server) sono ancora in esecuzione.

La disattivazione di un singolo sottosistema avviene mediante la funzione di exit presente su tutte le varie postazioni. Questa funzionalità sulla postazione del cliente, ha come unico effetto la chiusura del suo terminale.

Essendo quindi ogni sottosistema indipendente, non c'è bisogno di notificare a gli altri che uno di essi è stato terminato, fatta eccezione per il server, infatti, questa situazione genererebbe errore in tutti gli altri.

- **Fallimento**

Quando avviene un blocco generale del sistema, esso viene riavviato, quindi viene effettuata una nuova inizializzazione. Non c'è rischio di perdite di dati, poiché tutte le comande in corso vengono gestite non appena esse sono confermate dall'utente.

Quando invece avviene il collasso di un nodo o di una maglia di comunicazione, il sistema lancia un avviso di errore, il quale oltre ad avvisare l'utente della mancata esecuzione di un'operazione, dà la possibilità di riprovare la stessa in un secondo momento.

3 Servizi dei Sottosistemi

3.1. Application Layer

3.1.1. ValidaServizioBar

Tabella 1: Descrizione e servizi del sottosistema ValidaServizioBar

Sottosistema: <i>ValidaServizioBar</i>	
Descrizione	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un interfaccia che gestisce l'accesso e le operazioni comuni a tutti gli utenti.
Servizi del sottosistema	
<u>Operazioni</u>	<u>Comportamento</u>
Login()	Operazione che offre la possibilità a un Utente Registrato di accedere alla sezione ad egli riservata.
LogOut ()	Operazione che offre la possibilità a un Utente Registrato di uscire dal sistema
VisualizzaProdotti ()	Operazione che offre la possibilità a un Utente Registrato di visualizzare i prodotti del bar
ModificaProdotti ()	Operazione che offre la possibilità ad un Utente Registrato di modificare i prodotti.
AggiungiProdotti ()	Operazione che offre la possibilità ad un Utente Registrato di aggiungere i prodotti.
CancellaProdotti ()	Operazione che offre la possibilità ad un Utente Registrato di cancellare i prodotti.

3.1.2. ValidaServizioRistorante

Tabella 2: Descrizione e servizi del sottosistema ValidaServizioRistorante

Sottosistema: <i>ValidaServizioRistorante</i>	
Descrizione	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un interfaccia che gestisce l'accesso e le operazioni comuni a tutti gli utenti.
<u>Servizi del sottosistema</u>	
<u>Operazioni</u>	<u>Comportamento</u>
Login()	Operazione che offre la possibilità a un Utente Registrato di accedere alla sezione ad egli riservata.
LogOut ()	Operazione che offre la possibilità a un Utente Registrato di uscire dal sistema
VisualizzaMenù()	Operazione che offre la possibilità a un Utente Registrato di visualizzare i menù del ristorante.
ModificaMenù ()	Operazione che offre la possibilità ad un Utente Registrato di modificare i menù
AggiungiMenù ()	Operazione che offre la possibilità ad un Utente Registrato di aggiungere i menù
CancellaMenù ()	Operazione che offre la possibilità ad un Utente Registrato di cancellare i menù.

3.1.3. ValidaAttivitàExtra

Tabella 3: Descrizione e servizi del sottosistema ValidaAttivitàExtra

Sottosistema: <i>ValidaAttivitàExtra</i>	
Descrizione	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un interfaccia che gestisce l'accesso e le operazioni comuni a tutti gli utenti.
Servizi del sottosistema	
<u>Operazioni</u>	<u>Comportamento</u>
Login()	Operazione che offre la possibilità a un Utente Registrato di accedere alla sezione ad egli riservata.
LogOut ()	Operazione che offre la possibilità a un Utente Registrato di uscire dal sistema
VisualizzaAttività()	Operazione che offre la possibilità a un Utente Registrato di visualizzare le attività extra.
ModificaAttività ()	Operazione che offre la possibilità ad un Utente Registrato di modificare le attività
AggiungiAttività ()	Operazione che offre la possibilità ad un Utente Registrato di aggiungere le attività
CancellaAttività ()	Operazione che offre la possibilità ad un Utente Registrato di cancellare le attività

3.1.4. ValidaServizioCamere

Tabella 4: Descrizione e servizi del sottosistema ValidaServizioCamere

Sottosistema: <i>ValidaServizioCamere</i>	
Descrizione	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un interfaccia che gestisce l'accesso e le operazioni comuni a tutti gli utenti.
Servizi del sottosistema	
<u>Operazioni</u>	<u>Comportamento</u>
Login()	Operazione che offre la possibilità a un Utente Registrato di accedere alla sezione ad egli riservata.
LogOut ()	Operazione che offre la possibilità a un Utente Registrato di uscire dal sistema
VisualizzaPrenotazioneCamera()	Operazione che offre la possibilità a un Utente Registrato di visualizzare la prenotazione delle camere.
ModificaPrenotazioneCamera ()	Operazione che offre la possibilità ad un Utente Registrato di modificare la prenotazione delle camere.
AggiungiPrenotazioneCamera ()	Operazione che offre la possibilità ad un Utente Registrato di aggiungere la prenotazione delle camere.
CancellaPrenotazioneCamera ()	Operazione che offre la possibilità ad un Utente Registrato di cancellare la prenotazione delle camere.

3.2. Data Layer

Tabella 5: Descrizione e servizi del sottosistema Data Layer

Sottosistema: <i>Data Layer</i>	
Descrizione	Sottosistema contenente un interfaccia che accede ai dati persistenti dell'applicazione
Servizi del sottosistema	
<u>Operazioni</u>	<u>Comportamento</u>
DatiCliente ()	Operazione che contiene le informazioni sui clienti che accedono al sistema per effettuare la prenotazione e consente anche la modifica e la creazione di clienti.
DatiUtente ()	Operazione che serve per memorizzare le registrazioni di vari addetti che gestiscono il sistema di prenotazione
DatiCamere ()	Operazione che contiene le informazioni della camera con dati riguardanti i numeri di posti disponibili e sul suo stato (occupata o no)
DatiProdottiBar ()	Operazione che mostra i dettagli dei singoli prodotti del bar con la disponibilità e il prezzo
DatiAttivitàExtra ()	Operazione che mostra le varie attività extra dell'albergo come per esempio gite escursioni ecc.
DatiMenù ()	Operazione che serve per mostrare un menu formato dalla data, dal tipo (cena o pranzo).
DatiPortateMenù ()	Operazione collegata alla classe menù, serve per mostrare le singole portate che formano il menù.
DatiTipoPortate ()	Operazione collegata alla classe menù, serve per mostrare le tipologie della portata.
DatiCategorie ()	Operazione che serve per elencare le varie categorie che formano il bar (per esempio: bibite, liquori, ecc.)
DatiPrenotazione ()	Operazione che consente ad un cliente di effettuare le prenotazioni di: attività extra, bar, ristorante
DatiDettagliPrenotazioneAttivitàExtra ()	Operazione che serve per mostrare i dettagli di una singola attività includendo dati come la disponibilità dei posti

DatiDettaglioPrenotazioneBar ()	Operazione che mostra e gestisce le varie prenotazioni che si possono effettuare al bar
DatiDettaglioPrenotazioneRistorante ()	Operazione che gestisce i dettagli riguardanti le prenotazioni del ristorante

5.Glossario

Client: componente che accede ai servizi o alle risorse di un'altra componente, detta server.

Deployment Diagram: Schema che descrive la struttura dinamica del sistema

DBMS: programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.

Event-driven: modello in cui la segnalazione o notifica di un evento ha lo scopo di informare le applicazioni di uno o più domini destinatari che, sulla base del significato del messaggio associato all'evento, producono un cambiamento permanente del valore dei propri oggetti applicativi.

Form: finestra di dialogo incorporata in una pagina Web che consente all'utente di inserire informazioni destinate ad un server. Generalmente richiede un programma sul server che si occupi di esaminare le informazioni inviate. E' composto da spazi (campi) predefiniti, ad esempio menù a tendina, elenchi puntati o caselle di testo libero.

Internet: Un sistema internazionale che connette tra loro piccole e grandi reti telematiche. Le reti connesse tramite Internet usano un particolare standard di comunicazione, conosciuto come TCP/IP.

Java: Linguaggio di programmazione ad oggetti, sviluppato da Sun Microsystem, che permette di generare sia programmi completi che moduli interattivi più piccoli (Applets), leggibili da un browser

JDBC: API per il linguaggio di programmazione Java che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati. È orientata ai database relazionali.

Login: Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.

Layer: E' un insieme di classi con funzionalità simile (tipicamente raggruppati in un unico package).

Logout: Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password (vedi *login*).

Mysql: Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.

Password: E' un metodo di sicurezza che, mediante una stringa di caratteri, permette di identificare un utente specifico. Generalmente le password sono formate da una sequenza di lettere e numeri; digitando correttamente questi caratteri, si può avere accesso al computer o alla rete.

Package: Un Package rappresenta una collezione di classi ed interfacce che possono essere raggruppate in base alla funzione comune da esse svolta

Server: Programma di gestione di un servizio che invia informazioni in un particolare formato ricevuto e interpretato da un programma Client dal lato ricevente.

Shutdown: fase di terminazione del sistema.

Start-up: processo di accensione e di avvio di un computer, di un dispositivo o di un sistema.