

Università degli Studi di Salerno
Corso di Ingegneria del Software

MyHotel

SDD

Versione 1.4



Partecipanti:

Nome	Matricola
Marco Avagliano	0512102880
Alessandro Criscuolo	0512103132

Scritto da:	Marco Avagliano,Alessandro Criscuolo
--------------------	--------------------------------------

Revision History

Data	Versione	Descrizione	Autore
05/12/2016	1.0	Introduzione,obiettivi del sistema,design goal	Alessandro Criscuolo
06/12/2016	1.1	Criteri di performance,Criteri di affidabilità,Criteri di costo,Criteri di mantenimento,Criteri dell'utente finale	Marco Avagliano
07/12/2016	1.2	Architettura corrente del software,Architettura software proposta	Alessandro Criscuolo
09/12/2016	1.3	Decomposizione del sistema in sottosistemi,component diagrams,gestione dei dati persistenti,matrice del controllo degli accessi	Marco Avagliano, Alessandro Criscuolo
10/12/2016	1.4	Flusso di controllo globale,hardware e software mapping, condizioni limite,start-up e shutdown use cases,exception use cases, servizi dei sottosistemi ,glossario	Marco Avagliano, Alessandro Criscuolo

Indice

INTRODUZIONE.....	5
OBIETTIVI DEL SISTEMA.....	5
DESIGN GOAL.....	6
CRITERI DI PERFORMANCE.....	6
TEMPO DI RISPOSTA.....	6
THROUGHPUT.....	6
MEMORIA.....	6
CRITERI DI AFFIDABILITA'.....	6
ROBUSTEZZA.....	6
AFFIDABILITA'.....	7
SICUREZZA.....	7
TOLLERANZA.....	7
DISPONIBILITA'.....	7
CRITERI DI COSTO.....	7
DEVELOPMENT COST.....	7
DEPLOYMENT COST.....	7
CRITERI DI MANTENIMENTO.....	8
ESTENDIBILITA'.....	8
LEGGIBILITA'.....	8
MODIFICABILITA'.....	8
CRITERI DELL'UTENTE FINALE.....	8
USABILITY.....	8
UTILITY.....	8
DEFINIZIONI,ACRONIMI E ABBREVIAZIONI.....	9
RIFERIMENTI.....	10
PANORAMICA.....	10
ARCHITETTURA CORRENTE DEL SOFTWARE.....	10
ARCHITETTURA SOFTWARE PROPOSTA.....	11
DECOMPOSIZIONE DEL SISTEMA IN SOTTOSITEMI.....	11
COMPONENT DIAGRAMS.....	13

GESTIONE DATI PERSISTENTI.....	14
TABELLA UTENTE.....	14
TABELLA CAMERE.....	15
TABELLA PRENOTAZIONI.....	15
TABELLA SERVIZI.....	16
TABELLA CARRELLO.....	16
MATRICE DI CONTROLLO DEGLI ACCESSI.....	16
FLUSSO DI CONTROLLO GLOBALE.....	18
HARDWARE E SOFTWARE MAPPING.....	19
CONDIZIONI LIMITE.....	20
START-UP E SHUTDOWN USE CASES.....	21
EXCEPTION USE CASES.....	22
SERVIZI DEI SOTTOSISTEMI.....	23
GLOSSARIO.....	25

Introduzione

Il sistema nasce per rendere quanto più semplice possibile la prenotazione online di una vacanza presso un hotel. I servizi presenti online per la prenotazione di vacanze sono molteplici, ma ognuno riporta qualche mancanza. Ad esempio, ci sono piattaforme che permettono di prenotare le stanze inserendo solo il periodo di permanenza, o altri ancora che non permettono di scegliere i servizi di cui si può usufruire. Nel nostro caso, la piattaforma prende in gestione un hotel in modo tale che tutti gli utenti che vogliono passare la propria vacanza in quest'ultimo possano creare una vacanza ad hoc, specificando tutte le proprie necessità.

Obiettivi del Sistema

MyHotel è una piattaforma che permette ad un utente che vuole prenotare una vacanza presso un hotel di specificare tutte le proprie necessità, dal periodo di permanenza ai trattamenti di pensione completa, mezza pensione o B&B, dalla scelta dei servizi alla possibilità di scegliere una camera specifica. Ogni utente potrà dare un'occhiata a

tutte le funzionalità anche prima di essersi registrato, ma per usufruirne dovrà prima registrarsi e poi effettuare l'autenticazione.

Design Goal

Il sistema software a cui abbiamo pensato cerca di rispettare tutti gli obiettivi di design di seguito elencati:

Criteri di performance

Tempo di risposta

Attraverso un'analisi accurata siamo giunti alla conclusione che il tempo di risposta deve essere inferiore ai 4 secondi. I tempi di risposta alla sottomissione di dati saranno limitati in quanto parte dei controlli verrà effettuata lato client e l'utente avrà subito la segnalazione dell'eventuale non correttezza dell'input inserito.

Troughput

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 avrà in memoria i dati relative agli utenti registrati e le informazioni riguardanti le prenotazioni effettuate. Inoltre dovrà contenere tutti i dettagli di ciascuna camera come foto, prezzo ecc. Come sistema di memorizzazione verrà usato un database relazionale di tipo MySQL.

Criteri di affidabilità

Robustezza

Il sistema garantisce il corretto funzionamento anche nel caso in cui vengano inseriti input invalidi. Il software avvertirà l'utente qualora sbagli

o ometta l'inserimento di un campo, spiegando dettagliatamente cosa ha provocato l'errore nel sistema.

Affidabilità

Il sistema deve essere in grado di effettuare le operazioni nella loro atomicità, ovvero devono essere realizzate tutte le fasi di quest'ultima o nessuna fase in modo da garantire la consistenza dei dati. Inoltre, deve essere in grado di stabilire il valore dei dati ad una situazione precedente ad un crash o comunque in un loro stato consistente.

Sicurezza

Ogni utente potrà accedere con una email e password personale decisa al momento della registrazione.

Tolleranza

Vengono previsti dei backup periodici per evitare la perdita totale dei dati in caso di crash del sistema.

Disponibilità

Il sistema sarà disponibile per l'utilizzo sette giorni su sette, 24 ore su 24.

Criteri di costo

Development cost

Il DBMS utilizzato per il sistema sarà MySQL che è open source. Inoltre i costi di sviluppo prevedono la retribuzione per il lavoro dei singoli componenti del team. In aggiunta sono previsti dei costi per licenze software.

Deployment cost

Non sono previsti costi di deployment.

Criteri di mantenimento

Estensibilità

La progettazione del sistema sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità utilizzando linguaggi di scripting e markup (HTML, JAVASCRIPT, CSS, JAVA, JSP, SERVLET) per la piattaforma web.

Leggibilità

Il codice sarà ben strutturato e documentato per semplificare eventuali interventi su di esso. Per documentazione intendiamo la documentazione di ausilio che l'utente può consultare usando appositi link nel footer mostrato nella homepage e i commenti al codice.

Modificabilità

Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile per facilitarne la modifica.

Criteri dell'utente finale

Usability

L'utilizzo del sistema non richiede competenze tecniche. Deve essere possibile apprendere il funzionamento del sistema in breve tempo e senza neanche consultare la documentazione d'ausilio. Il sistema è tuttavia provvisto di documentazione online per aiutare l'utente ad apprendere come utilizzare il sistema. Le interfacce mostrate all'utente saranno intuitive e autoesplicative. Ogni pulsante avrà un'etichetta che ne specifica la funzione, le caselle di testo presenteranno del testo illustrativo per aiutare l'utente nell'immissione dei dati.

Utility

Eventuali errori verranno spiegati all'utente attraverso opportuni messaggi.

Definizioni,acronimi e abbreviazioni

Acronimo	Definizione
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)
SQL	Structured Query Language (linguaggio per l'invocazione delle richieste al database)
CSS	Cascading Style Sheet (linguaggio utilizzato per la resa grafica del portale)
DB	Database (sistema di memorizzazione per l'archiviazione dei dati permanenti)
JAVA	Linguaggio orientato agli oggetti
MYSQL	DBMS Utilizzato
JAVASCRIPT	Linguaggio di scripting utilizzato lato client per rendere la piattaforma web dinamica
Query	Interrogazione al database
Layout	Impaginazione struttura grafica del portale e dell'applicazione
HTTP	HyperTextTransferProtocol (Protocollo per la gestione di richieste e risposte scambiate tra client e server)
RAD	Requirements Analysis Document
SDD	System Design Document

Riferimenti

L'insieme del materiale di riferimento utilizzato per la realizzazione del progetto e per la stesura di questo stesso documento comprende:

- Slide del docente, reperibili sulla piattaforma E-Learning.
- Libro di testo "Object-Oriented Software Engineering – Using UML, Patterns and Java" di Bernd Bruegge e Allen H. Dutoit, edito da Prentice Hall.
- Documento RAD di MyHotel.

Panoramica

Il documento si compone di una prima parte in cui vengono introdotti i design goals. Viene brevemente spiegata l'architettura corrente del software e quella a cui stiamo pensando per la definizione del nostro sistema. Segue la decomposizione del sistema in sottosistemi e l'individuazione dei servizi offerti da ogni sottosistema. Vengono individuati i dati persistente necessari per il funzionamento di MyHotel e l'approccio per la gestione dei dati persistenti. Vengono individuate le categorie di utenti che possono usare il sistema e viene definita la politica di accesso. Vengono disciplinate le condizioni di eccezione e casi limite.

Architettura corrente del software

Ricercando sul web, abbiamo trovato molti sistemi simili a quello che stiamo progettando. Infatti la maggior parte degli hotel possiede una piattaforma online che permette la prenotazione. Esplorandone alcune ci siamo accorti che solo in rari casi permettono di scegliere i servizi aggiuntivi al momento della prenotazione. La maggior parte delle piattaforme, infatti, si limitano soltanto ad effettuare una prenotazione

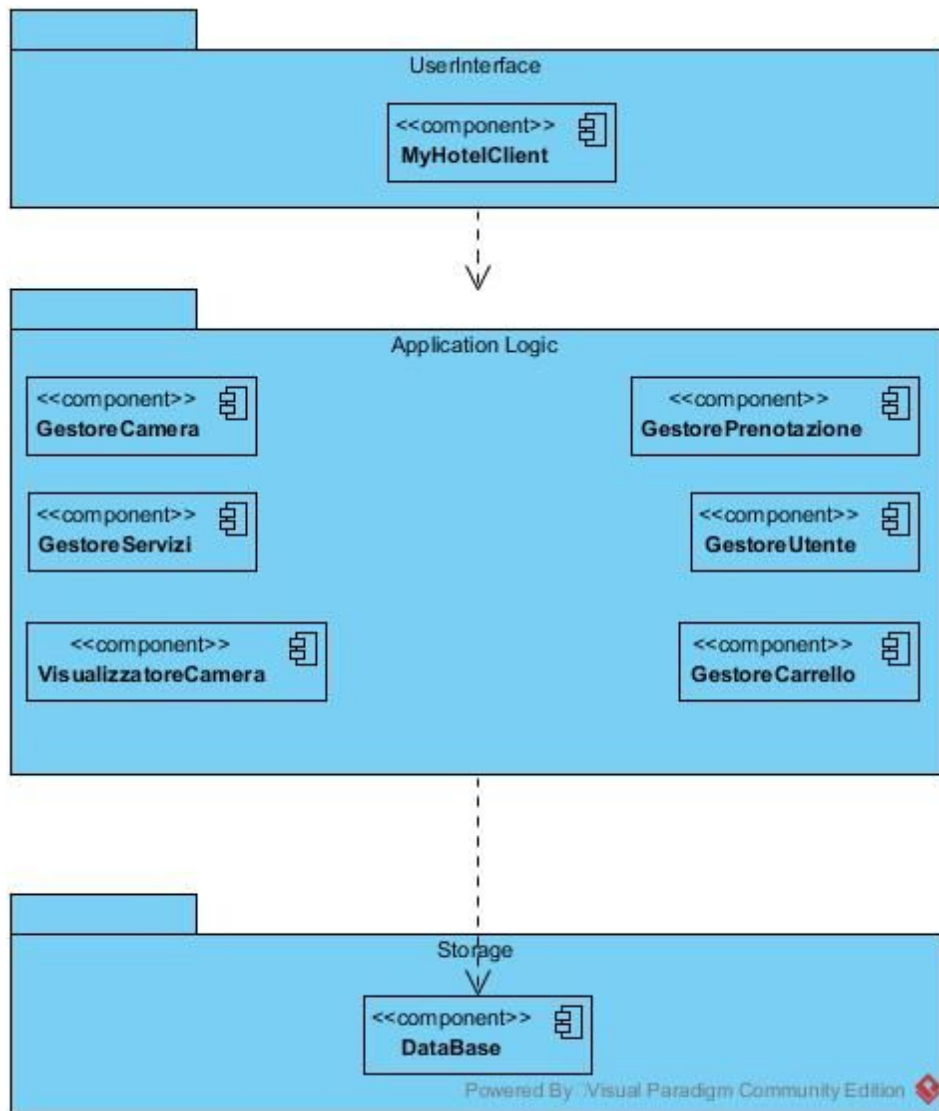
permettendo di inserire solo il periodo di permanenza non permettendo così all'utente la possibilità di inserire eventuali altre richieste.

Architettura software proposta

MyHotel è una piattaforma web basata su un'architettura client-server: il client chiede l'erogazione di servizi al server che provvederà a fornire le risposte in tempi accettabili. Per memorizzare i dati persistenti è necessario disporre di un sistema di memorizzazione e abbiamo optato per un database relazionale in modo che possa essere gestito agevolmente l'accesso concorrente ai dati e l'utilizzo del DBMS garantisca la consistenza dei dati stessi. Il database viene utilizzato solo dal server: il client richiede le informazioni al server che provvederà a prelevarle dal database mediante opportune query. Gli utenti non hanno modo di interagire tra di loro. Il client si occupa solo di rendere graficamente le richieste e i risultati all'utente finale; tutta la logica dell'applicazione risiede sul server. A partire da questa osservazione abbiamo optato per un'architettura three-tier: vengono individuati i livelli di interfaccia, logica dell'applicazione e sistema di memorizzazione.

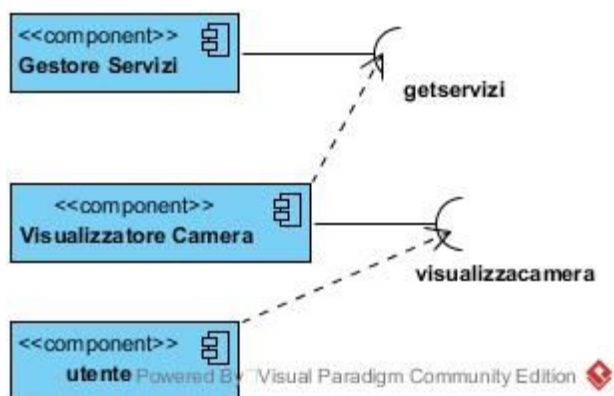
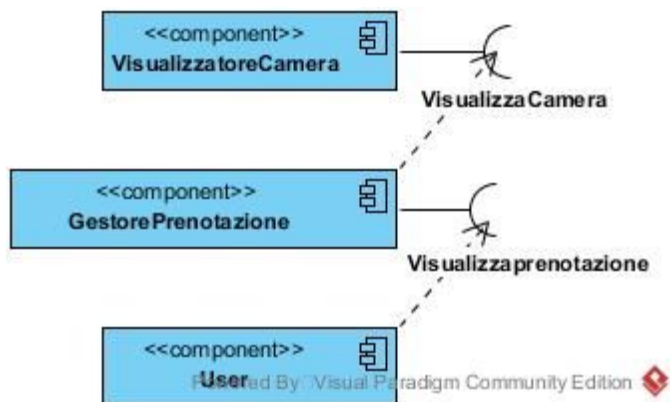
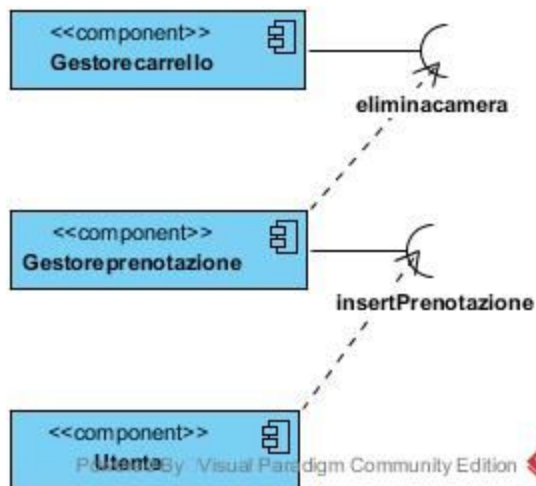
Decomposizione del sistema in sottosistemi

Durante la fase di analisi abbiamo individuato una prima divisione logica del sistema in sottosistemi e individuato le relazioni tra di loro. La fase di System Design eredita dalla fase di analisi il seguente schema:



Ogni sottosistema offre dei servizi specifici riguardanti il concetto relativo al dominio dell'applicazione che astrae. Per quanto riguarda le relazioni tra i sottosistemi, ogni sottosistema può interagire con il Database attraverso il Manager contenuto in esso

Component Diagrams:



Gestione dei dati persistenti

Il sistema utilizza un database di tipo relazionale, con cui si ottiene una maggiore affidabilità, maggiore facilità di gestione, maggiore velocità di accesso e trasmissione dei dati.

I dati persistenti individuati anche dallo studio dei casi d'uso sono:

1. Tabella Utente, contenente l'elenco dei dati degli utenti registrati al sistema;
2. Tabella Camere, contenente l'elenco di tutte le camere dell'albergo e i dati relativi ad esse;
3. Tabella Prenotazioni, contenente i dati relativi alle prenotazioni effettuate dagli utenti.
4. Tabella Servizi, contenente l'elenco di tutti i servizi messi a disposizione dall'hotel per le camere.
5. Tabella Carrello, contenente l'elenco di tutte le prenotazioni effettuate dagli utenti registrati alla piattaforma.

Questi dati devono essere permanenti e accessibili in un qualsiasi momento. Abbiamo scelto di usare il DBMS open source MySQL per la gestione del database relazionale. All'interno del database abbiamo bisogno di memorizzare:

Tabella Utente

Campi	Vincoli
email	Chiave primaria
password	not null
nome	not null
cognome	not null
dataNascita	not null

ruolo	enum{Cliente,Amministratore}
-------	------------------------------

Tabella Camere

Campi	Vincoli
numeroCamera	Chiave Primaria
tipologia	enum{Singola, Doppia, Suite}
immagine	
prezzo	not null
descrizione	

Tabella Prenotazioni

Campi	Vincoli
idPrenotazione	Chiave Primaria
email	Chiave Esterna
numeroCamera	Chiave Esterna
totale	Not null
dataInizio	Not null
dataFine	Not null

Tabella Servizi

Campi	Vincoli
nomeServizio	Chiave Primaria
descrizione	Not null

Tabella Carrello

Campi	Vincoli
email	Chiave Esterna
numeroCamera	Chiave Esterna
dataInizio	Not null
dataFine	Not null
totale	Not null

Matrice di controllo degli accessi

Le categorie di utenti di MyHotel sono due: utente non registrato e utente registrato.

L'utente non registrato ha accesso alla visualizzazione delle camere e dei servizi offerti dalla struttura.

L'utente registrato è un un utente che, oltre alla possibilità di visualizzare le camere, può effettuare tutte le operazioni offerte dalla piattaforma. In particolare potrà accedere al proprio profilo, modificare i propri dati personali, effettuare una prenotazione, visualizzare il carrello e, eventualmente, eliminare un prodotto da quest'ultimo.

Un particolare tipo di utente registrato è l'amministratore, che risulta essere un utente che ha la possibilità di visualizzare le prenotazioni effettuate dai clienti, modificare la disponibilità delle camere, visualizzare

la sua area personale, aggiungere una nuova camera o eventualmente eliminarla.

Attore\Oggetto	Gestore Utente	Visualizzatore e Camere	Gestore Camere	Gestore Prenotazioni	Gestore Carrello	Gestore Servizi
Utente non registrato	Registrazione	Filtra camera, Visualizza camera	-	-	-	Visualizza ServiziCamera
Utente registrato	Login/Logout , Visualizza Profilo, Modifica Profilo, Elimina Profilo	Filtra camera, Visualizza camera	-	Aggiungi Prenotazione, Visualizza Prenotazioni Utente	Aggiungi camera, Rimuovi camera, Svuotacarrello, Visualizza carrello	Visualizza ServiziCamera
Amministratore	Login/Logout , Visualizza Profilo, Modifica Profilo, Elimina Profilo, Visualizza Utenti	Visualizza camera	Aggiungi camera Elimina camera Modifica camera	Visualizza Prenotazioni, Rimuovi Prenotazioni, Filtra Prenotazioni	-	Visualizza ServiziCamera, Visualizza Servizi, Aggiungi Servizio Camera

Un *Utente non registrato* può usare il servizio di registrazione offerto dal GestoreUtente, inoltre può visualizzare e filtrare tutte le camere presenti nel database sfruttando il VisualizzatoreCamere. Nell'accedere ad una pagina dedicata alla visualizzazione di una camera l'utente usa involontariamente il GestoreServizi per visualizzare gli eventuali servizi disponibili per quella camera.

L'*Utente Registrato* ha invece a disposizione, come è normale che sia, molti più servizi infatti può effettuare le procedure di login e logout sfruttando le credenziali fornite durante la registrazione, inoltre può gestire il suo profilo modificando i suoi dati o addirittura eliminandolo. Tutti questi servizi sono offerti dal GestoreUtente.

Mentre per quanto riguarda il VisualizzatoreCamere ha a disposizione le stesse funzionalità di un utente non registrato.

Per effettuare una prenotazione l'utente registrato sfrutta il GestorePrenotazione che salva in modo permanente ogni singola prenotazione e da anche la possibilità di visualizzare lo storico delle prenotazioni effettuate dall'utente che lo richiede.

Prima di effettuare una prenotazione l'utente registrato deve prima aggiungere una o più camere al carrello, per effettuare questa operazione si deve far uso del GestoreCarrello che offre oltre alla possibilità di aggiungere o eliminare una camera dal carrello anche quella di svuotarlo o semplicemente di visualizzarlo per procedere alla prenotazione.

L'*Amministratore* può effettuare login e logout, può visualizzare, modificare o eliminare qualunque profilo presente all'interno del database poichè ha accesso alle informazioni di tutti gli utenti. Può effettuare queste operazioni utilizzando il GestoreUtente.

Può visualizzare le informazioni di una singola camera sfruttando il VisualizzatoreCamere e inoltre ha la possibilità di aggiungere, modificare eliminare una camera dal database grazie ai servizi offerti dal GestoreCamera a cui solo l'amministratore ha accesso.

Per quanto riguarda il GestorePrenotazione l'amministratore può filtrare le prenotazioni di tutti gli utenti visualizzando solo quelle a cui è interessato e inoltre può rimuoverle singolarmente.

L'amministratore non dovendo effettuare prenotazioni non ha accesso al GestoreCarrello, però ha accesso al GestoreServizi infatti oltre a visualizzare i servizi di una singola camera quest'ultimo può visualizzare tutti i servizi offerti dalla struttura ed eventualmente aggiungerne uno più ad una singola camera scegliendo fra quelli disponibili.

Flusso di controllo globale

Il sito web ha un flusso guidato dagli eventi, ovvero non esiste una sequenza di operazioni prestabilita, ma è l'utente a scegliere di volta in

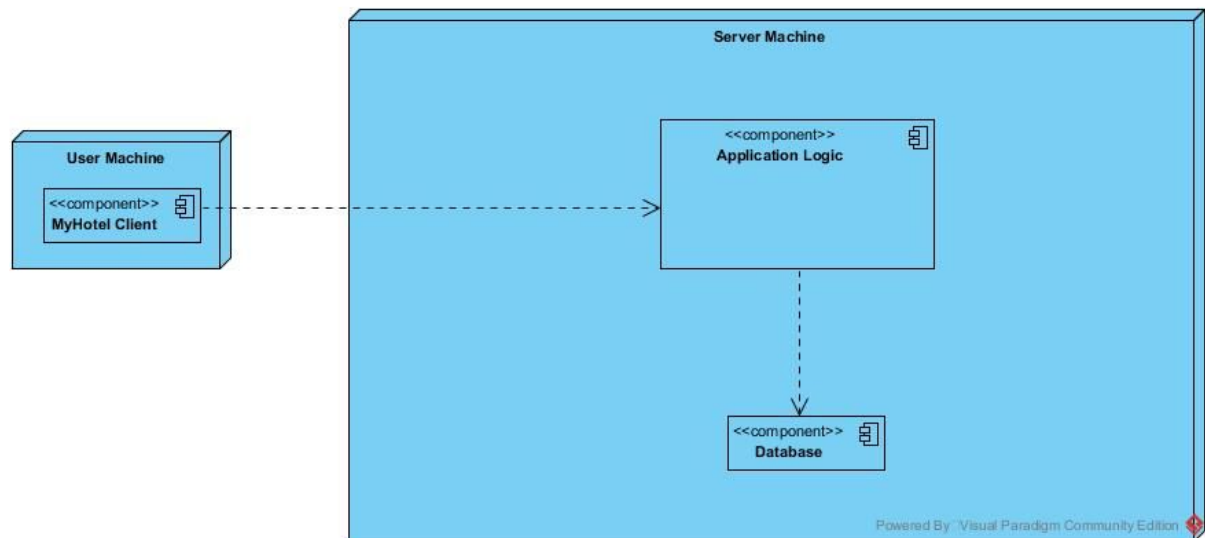
volta l'operazione da eseguire. L'architettura software è composta da un web server che rimane in attesa di una richiesta da parte di un web browser. Appena il web server riceve la richiesta di erogazione di un servizio il sistema elabora il risultato invocando i metodi sui sottosistemi coinvolti nell'operazione e invia l'output al client che provvederà a fornirlo (in maniera grafica) all'utente. Alcune funzionalità saranno eseguite in maniera sincrona, ad esempio la registrazione. Ogni volta che sarà possibile le operazioni saranno realizzate in maniera asincrona per ridurre i tempi di risposta.

Il nostro sistema si baserà essenzialmente sul pattern MVC, quindi sarà orientato ad eventi, ovvero, ogni qualvolta un utente esegue operazioni sul nostro sistema, tali operazioni verranno reindirizzate e gestite attraverso dei dispatcher. Quindi, per natura, il nostro sistema sarà decentralizzato, ossia ogni oggetto provvederà a risolvere una singola classe di problemi, quindi per poter eseguire delle operazioni, verranno chiamate in causa vari oggetti, su cui ognuno di essi eseguirà azioni a lui competenti.

Hardware e Software Mapping

MyHotel è un sistema costituito da una piattaforma web. Per usufruire dei servizi offerti sono necessari: un dispositivo in possesso dell'Utente che provvederà all'interfaccia utente e uno o più dispositivi Server su cui verrà eseguita la logica dell'applicazione e il sistema di memorizzazione che, in generale, provvederanno all'erogazione dei servizi del sistema. Il MyHotel client è localizzato sul dispositivo del client. Sul server vengono posizionati tutti gli altri sottosistemi individuati nella fase di decomposizione. Per la realizzazione della piattaforma abbiamo scelto le SERVLET per la definizione dei sottosistemi lato server; HTML, JAVASCRIPT, JSP e CSS per la definizione del sottosistema lato client. Il client del sistema è un web browser e il MyHotel server e tutti gli altri sottosistemi sono accessibili mediante web server. Il sistema di memorizzazione dei dati è un database relazionale che risiede sul dispositivo server e a cui i sottosistemi lato server possono accedere

mediante un DBMS. Come DBMS abbiamo scelto MySQL. Le query al database vengono definite usando il linguaggio SQL. Il protocollo usato per la comunicazione tra client e server è http.



Condizioni Limite

Configuration

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

Utente: l'oggetto Utente viene creato dall'utente non registrato eseguendo il caso d'uso "Registrazione dell'utente" e archiviato subito dopo la creazione. La rimozione dal sistema viene effettuata su richiesta dell'utente registrato utilizzando il caso d'uso "Rimuovi profilo".

Camere: l'oggetto Camere viene creato dall'utente con privilegi di amministratore, eseguendo il caso d'uso "L'amministratore aggiunge una nuova camera" e archiviato subito dopo la creazione. La rimozione dal sistema di un oggetto Camere avviene sempre per mano dell'amministratore che utilizza il caso d'uso "L'amministratore elimina

una camera". Nel caso in cui si debba cambiare la disponibilità della camera oppure un qualsiasi dato di quest'ultima, l'amministratore utilizzerà il caso d'uso "L'amministratore modifica la disponibilità delle camere".

Prenotazione: l'oggetto Prenotazione viene creato dall'utente registrato tramite il caso d'uso "L'utente acquista un soggiorno". Un utente ha la possibilità di rimuovere un oggetto Prenotazione se non ha più la volontà di pernottare nell'hotel tramite il caso d'uso "Disdici Prenotazione". L'oggetto prenotazione può essere acceduto anche dall'amministratore, che può visualizzare tutte le prenotazioni effettuate dagli utenti, tramite il caso d'uso "L'amministratore visualizza le prenotazioni effettuate".

Start-Up e Shutdown Use Cases

La fase di avviamento del sistema parte dal caricamento sul database dei dati, presi in fasi in cui lo stesso sistema era in funzione. La UserInterface carica gli script necessari alla visualizzazione di pagine del profilo utente. Da qui, attraverso la form di login, l'utente può accedere alle operazioni del sistema e ai relativi dati delle entità in database.

StartServer

Attore: SystemAdmin

Entry condition: Il SystemAdmin ha effettuato il login sul server

Flusso di eventi:

Il SystemAdmin esegue il comando di startup del sistema

Il sistema esegue le operazioni di inizializzazione, andando a leggere dal database i dati necessari alle normali funzionalità

Exit condition: Il sistema è inizializzato correttamente ed è disponibile

Per quanto riguarda la terminazione del sistema, dato che i sottosistemi sono decentralizzati e gestiti ad eventi, si vanno ad effettuare controlli sulle operazioni in modo tale da non renderle disponibili all'utilizzo. Di conseguenza, si gestisce un cambiamento comportamentale del

sistema, in modo tale da causare delle failure quando questo non è disponibile.

ShutdownServer

Attore: SystemAdmin

Entry condition: Il SystemAdmin ha effettuato il login sul server

Flusso di eventi:

Il SystemAdmin esegue il comando di shutdown del sistema

Il sistema esegue le operazioni di

terminazione, rilasciando tutte le risorse del utilizzate e salvando i dati rimanenti sul database per recuperarli in seguito.

Exit condition: Il sistema è spento correttamente.

Exception Use Cases

MyHotel può essere affetto da due tipi di eccezioni:

1. fallimento hardware, crash del sistema, sia server che DBMS;
2. errore nel software.

Nel caso di un fallimento hardware si cerca di mantenere i dati in uno stato consistente, ma se è il DBMS a fallire si avrà una perdita totale dei dati. Per ovviare a ciò si effettueranno delle copie di backup periodiche.

Nel caso in cui sia il server o il DBMS a fallire si segnala l'errore al client e si chiude il dispositivo che ha provocato il fallimento. Se è il server che ha smesso di funzionare si cerca di riavviarlo. Se è il DBMS che ha smesso di funzionare il server cercherà di avviarlo nuovamente o recupererà i dati dalla copia di backup. Non viene considerato un'eccezione il fallimento di un client perché non può danneggiare il corretto funzionamento del sistema. Nel caso di cambio di ambiente operativo il sistema non può far altro che segnalare all'utente il problema che si è verificato e aspettare che il dispositivo abbia di nuovo la connessione ad Internet. Nel caso di errore software il sistema dovrà provvedere a risolvere il bug. Come garanzia ogni sottosistema effettuerà opportuni controlli di dati provenienti dall'utente o da altri sottosistemi per prevenire gli errori.

Extension Use Case: fallimento hardware
ID: fallimento hardware
Esteso da: start-up server, start-up database, shutdown server, shutdown database.
Il sistema hardware smette di funzionare correttamente. Il sistema manda un messaggio di errore all'utente. Si cerca di memorizzare i dati in maniera consistente. Il dispositivo che ha provocato il fallimento viene chiuso.

Servizi dei Sottosistemi

Nome Sottosistema	Gestore Utenti
Descrizione	E' il sottosistema che si occupa della gestione dei dati di ogni utente registrato.

Servizio	Descrizione
Registrazione	Il sottosistema permette ad ogni utente che sta effettuando la registrazione di creare un proprio profilo utente: nel profilo confluiranno tutti i dati inseriti nel form di registrazione dall'utente stesso.
Login	Il sottosistema permette ad ogni utente registrato di poter effettuare il login e usufruire di tutti i servizi della piattaforma.
Logout	Il sottosistema permette ad ogni utente loggato di effettuare il logout e non usufruire di tutti i servizi della piattaforma.
Visualizza Profilo	Il sottosistema permette ad ogni utente loggato di visualizzare i dati che ha inserito nel form durante la fase di registrazione alla piattaforma.
Modifica Profilo	Il sottosistema permette ad ogni utente loggato di visualizzare i dati che ha inserito nel form durante la fase di registrazione alla piattaforma e poterli modificare sempre tramite un form.

Elimina Profilo	Il sottosistema permette ad ogni utente loggato che non vuole più usufruire della piattaforma di eliminare il proprio profilo da quest'ultima.
-----------------	--

Nome Sottosistema	Gestore Camere
Descrizione	E' il sottosistema che si occupa di tutte le operazioni che possono essere eseguite sulle camere.

Servizio	Descrizione
Aggiungi Camera	Il sottosistema permette all'amministratore della piattaforma di aggiungere nuove camere a quelle già esistenti.
Modifica Camera	Il sottosistema permette all'amministratore della piattaforma di modificare la disponibilità o i dati delle camere.
Elimina Camera	Il sottosistema permette all'amministratore della piattaforma di eliminare una camera.

Nome Sottosistema	Visualizzatore camera
Descrizione	E' il sottosistema che permette di visualizzare le camere con le

Servizio	Descrizione
Visualizza camera	Il sottosistema permette ad ogni utente(registrato e non) di visualizzare le camere presenti all'interno della piattaforma.
Filtra	Il sottosistema permette ad ogni utente(registrato e non) di filtrare la visualizzazione delle camere tramite un form.
Ordina	Il sottosistema permette di ordinare le camere a seconda dei parametri inseriti dall'utente.

Nome sottosistema	Gestore Prenotazioni
Descrizione	E' il sottosistema che si occupa della gestione dei dati relativi alle prenotazioni effettuate dagli utenti.

Servizio	Descrizione
----------	-------------

Aggiungi Prenotazione	Il sottosistema permette di aggiungere una prenotazione all'utente(registrato).
Rimuovi Prenotazione	Il sottosistema permette di rimuovere una prenotazione all'utente(registrato e amministratore).
Visualizza Prenotazione	Il sottosistema permette all'amministratore della piattaforma di visualizzare lo storico delle prenotazioni effettuate da tutti gli utenti. Inoltre permette ad un utente di visualizzare lo storico delle proprie prenotazioni.
Filtra Prenotazioni	Il sottosistema permette all'amministratore della piattaforma di filtrare le prenotazioni.

Nome sottosistema	Gestore Carrello
Descrizione	E' il sottosistema che si occupa di tutte le attività legate al carrello di ogni utente.

Servizio	Descrizione
Visualizza carrello	Il sottosistema permette all'utente di poter visualizzare tutte le sue prenotazioni aggiunte al carrello e non ancora pagate.
Aggiungi al carrello	Il sottosistema permette all'utente di poter aggiungere una prenotazione al carrello.
Svuota carrello	Il sottosistema permette all'utente di poter svuotare il carrello di tutte le prenotazioni aggiunte precedentemente.

Nome sottosistema	Gestore Servizi
Descrizione	E' il sottosistema che si occupa della gestione dei servizi offerti dalla piattaforma per ogni camera dell'hotel.

Servizio	Descrizione
Visualizza servizi	Il sottosistema permette all'amministratore di visualizzare tutti i servizi disponibili dell'hotel.
Visualizza servizi camera	Il sottosistema permette all'utente di visualizzare i servizi che sono disponibili presso una camera.

Inserisci servizi	Il sottosistema permette all'amministratore di aggiungere nuovi servizi ad una camera.
-------------------	--

Glossario

Camera: E' un oggetto del sistema che contiene informazioni sulla camera dell'albergo come prezzo, foto, descrizione ecc... .Quando un utente visualizza la pagina dedicata ad una camera ha accesso a tutte queste informazioni. Se l'utente è autenticato può aggiungerla al carrello e successivamente effettuare una prenotazione per una o più camere.

Servizi: Sono i servizi aggiuntivi che l'hotel offre ai propri clienti. L'utente può visionare i servizi disponibili e una volta autenticato può richiederli al momento della prenotazione di una camera.

Form: è una "scheda di compilazione" per l'inserimento dei dati, dove l'utente inserisce i dati e li invia al server.

Login: è la procedura di accesso (procedura di autenticazione) ad un sistema informatico o ad un'applicazione informatica con le credenziali registrate al sito web.

Logout: è la procedura di uscita da un sistema informatico o un'applicazione informatica.

Password: è una parola chiave con cui l'utente accede al sito web associandola al proprio indirizzo email.

Profilo: E' l'insieme delle informazioni dell'utente: email, username e l'immagine del profilo. Quando un utente chiederà di visualizzare il proprio profilo visualizzerà una pagina che contiene tali informazioni e avrà la possibilità di accedere allo storico delle sue prenotazioni. Inoltre dei link gli forniranno la possibilità di modificare il proprio profilo (le informazioni contenute nel profilo compresa la password) .

Registrazione: è la procedura di salvataggio dei dati dell'utente nel database con i quali accede a tutte le funzionalità che il sistema offre.

Sito web: è un insieme di pagine web correlate tra di loro.

Piattaforma web: indica una base software e/o hardware su cui sono sviluppate e/o eseguite applicazioni.

Utente: è un utente che si può specializzare in utente registrato e utente non registrato.

Utente non registrato: è colui che può usufruire di poche funzionalità del sistema in quanto non è registrato al sito web.

Utente registrato: è colui che ha accesso a tutte le funzionalità del sistema ovviamente dopo aver effettuato il login.