



Università degli Studi di Salerno
Corso di Ingegneria del Software

eRockOn
System Design Document
Versione 2.0

eRockOn
Share Your Soul...

30/03/2016

Progetto: eRockOn	Versione : 2.0
Documento: System Design Document	Data : 03/12/2015

Partecipanti:

Nome	Matricola
Russo Luigi[RL]	0512103314
Prisco Roberto[PR]	0512101988

Scritto da:	Prisco Roberto
--------------------	----------------

Indice

- **Introduzione** 4
 - Purpose of the system
 - Design goals
 - Definizioni, acronimi e abbreviazioni
 - Overview
- **Sistema corrente** 6
- **Architettura software proposta**7
 - Overview
 - Identificazione dei sottosistemi
 - Hardware/software mapping
 - Gestione dei dati persistenti
 - Identificazione degli oggetti persistenti
 - Definizione Sql del Database
 - Access control and security
 - Global software control
 - Boundary conditions
- **Servizi dei sottosistemi**20

1. Introduzione

1.1 Purpose of the System

Il sistema che l'azienda ha richiesto è una piattaforma web-based per la gestione dei turni di lavoro dei propri dipendenti, che in particolare permette ad ogni utente del servizio di accedere al sistema, previa autenticazione, e gli permetta di controllare i turni e gli orari lavorativi. Così facendo si ottiene la possibilità di interagire istantaneamente con l'amministrazione affinché vi sia una corretta e veloce gestione dei turni onde evitare perdite di tempo e probabili gap amministrativi.

1.2 Design Goals

Gli obiettivi di qualità che il sistema dovrà rispettare sono:

- **Prestazioni**
 - **Tempo di risposta:** Ci si aspetta che una richiesta possa essere soddisfatta in meno di 2 secondi. Oltre questo limite il sistema è da considerarsi scarso.
 - **Throughput:** Questo criterio è trascurabile, perché l'applicazione effettua poco lavoro computazionale.
- **Criteri di Affidabilità**
 - **Robustezza:** il sistema software dovrà evitare malfunzionamenti a seguito dei difetti nel codice. La maggior parte dei problemi riguardanti il codice dovrà essere individuata nella fase di testing. Inoltre il sistema dovrà sopravvivere agli input non validi immessi dall'utente.
 - **Affidabilità:** L'affidabilità è definita come la probabilità che il sistema funzioni senza errori per un dato intervallo di tempo, in un dato ambiente e per un determinato scopo. Useremo il numero di difetti (bug) che verranno scoperti durante la fase di test per predire l'affidabilità e sarà migliorata rimuovendo i fault (comportamento run-time inaspettato (ed errato) osservato da un utente del sistema) che compaiono nelle parti del sistema che sono più frequentemente usate.
 - **Disponibilità:** Il sistema sarà sempre disponibile a compiere nuove operazioni, in quanto il costo computazionale delle operazioni è molto basso.
 - **Tolleranza ai guasti:** Capacità di operare sotto condizioni di errore, affinché il sistema funzioni correttamente sotto tali condizioni viene utilizzata la tecnica della ridondanza. Questa tecnica si basa sulla

presenza degli stessi dati su diversi server , così che se uno di questi risulti non raggiungibile viene sostituito.

- **Sicurezza del sistema:** Capacità di resistere ad attacchi da parte di malintenzionati. La possibilità di resistere ad attacchi da parte di malintenzionati sarà garantita da un apposito sistema di gestione degli account. Inoltre un'ulteriore sicurezza dei dati è garantita dal DBMS e dal protocollo https. Per motivi legati alla sicurezza, l'area utente per la gestione dei dipendenti è accessibile solo in locale.
- **Usabilità:** eRockOn rende ogni funzione di semplice uso, garantendo all'utente un'ottima esperienza di utilizzo del sistema; grazie ad interfacce grafiche intuitive.
- **Compromessi di design**
 - **Spazio vs. velocità:** Se il software non rispetta i requisiti di tempo di risposta, è possibile utilizzare più memoria per velocizzare il sistema.
 - **Tempo di rilascio vs. funzionalità:** Se i tempi di rilascio sono stringenti, possono essere rilasciate meno funzionalità di quelle richieste, ma nei tempi giusti.
 - **Tempo di rilascio vs. qualità:** Se i tempi di rilascio sono stretti, il project manager può decidere di rilasciare il software nei tempi prefissati ma con dei bug noti e, successivamente, di correggere gli errori.

1.3 Definizioni, acronimi e abbreviazioni

- **DBMS:** Database management system;
- **GUI:** Graphical User Interface;
- **Fault:** Comportamento run-time inaspettato ed errato osservato da un utente del sistema;
- **Web-Based :** Web-based ovvero "basato sul web".

Quando si parla di un software Web-based si intende un programma in cui tutte le funzioni sono accessibili con un normale web-browser come Explorer ,quindi non necessita di alcun software di installazione sul computer degli utenti. Le applicazioni Web-based permettono ,quindi, all'utente di interagire con il sistema tramite un web-browser. Tale caratteristica permette all'utente di interagire da qualsiasi sede e di rendere quindi una postazione generica una vera e propria postazione di lavoro.

1.4 Overview

Il sistema che si vuole realizzare è un sistema distribuito nel quale esiste un nodo Server contenente l'archivio dati a cui i vari attori, dalle loro postazioni, vogliono accedere. Pertanto esistono dei moduli client che richiedono dati ad un modulo 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.

L'architettura è di tipo three-tier, molto adatta per sistemi dove c'è la necessità di un'organizzazione stratificata che delega ogni livello a un ristretto campo d'azione. In questo caso utilizzata per avere una netta distinzione tra interfaccia grafica e logica applicativa.

2. Sistema corrente

Non vi sono sistemi precedenti.

3. Architettura software proposta

3.1 Overview

Nella decomposizione del sistema si è scelto uno stile architetturale Three-tier.

Tale architettura organizza i sottosistemi in tre layers:

- **Il layer di interfaccia:** è lo strato che implementa la GUI del sistema, dove vi sono tutti quegli elementi che permettono all'utente di interagire con il sistema (form , button, text box, etc.)
- **Il layer di business logic:** racchiude tutta la logica applicativa.
- **Il layer di storage:** i dati acceduti dalla business logic sono presenti in maniera persistente in un DBMS.

3.2 Identificazione dei sottosistemi

Per prima cosa dobbiamo evidenziare l'interfaccia del nostro sistema che è caratterizzata da tre componenti. In primis troviamo il sottosistema “Autenticazione” ,comune sia per l'utente regolare che per l'amministratore, che a sua volta indirizza nella pagina principale.

In primo luogo abbiamo distinto cinque parti principali del sistema e-RockOn: la sezione di esplorazione, la sezione per l'inserimento di un annuncio, la sezione per la visualizzazione di un annuncio, la pagina personale e la pagina dell'amministratore.

La sezione di esplorazione si suddivide in quattro sottosistemi:

- **RicercaStrumentoPerNome:** Si occupa della ricerca di uno strumento filtrando il nome.
- **RicercaStrumentoPerDescrizione:** Si occupa della ricerca di uno strumento filtrando la descrizione.
- **RicercaStrumentoPerProvincia:** Si occupa della ricerca di uno strumento filtrando la provincia.
- **RicercaSalaPerProvincia:** Si occupa della ricerca di uno sala di registrazione filtrando la provincia.

La sezione per l'inserimento di un annuncio si suddivide in due sottosistemi:

- **InserisciAnnuncioStrumento:** Si occupa dell'inserimento di un annuncio per uno strumento musicale .
- **InserisciAnnuncioSala:** Si occupa dell'inserimento di un annuncio per uno sala di registrazione.

La sezione per la visualizzazione di un annuncio si suddivide in tre sottosistemi:

- **NoleggiaStrumento:** Si occupa di inviare una richiesta di noleggio di uno strumento al rispettivo proprietario.
- **NoleggiaSala:** Si occupa di inviare una richiesta di noleggio di uno sala registrazioni al rispettivo proprietario.

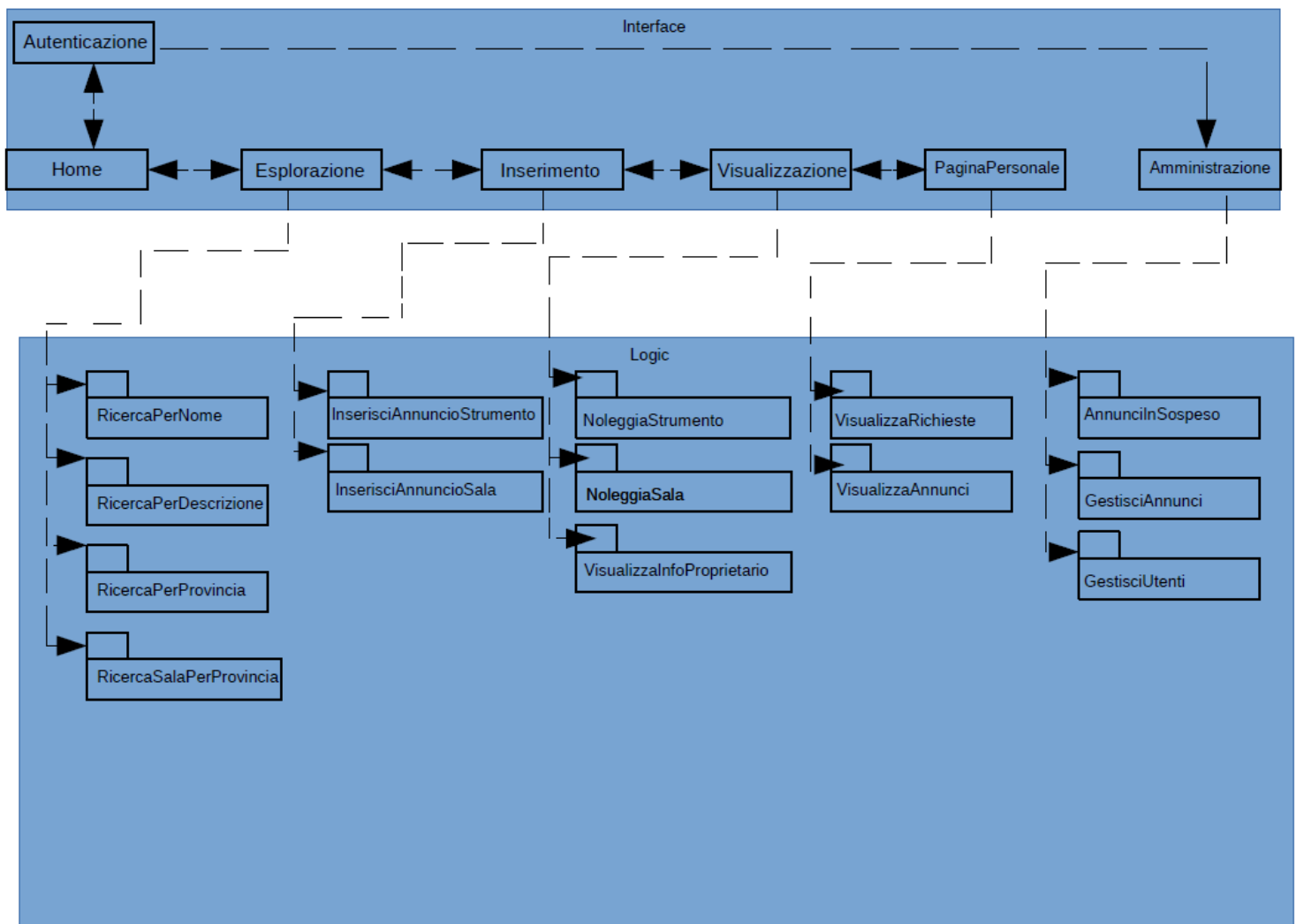
- **VisualizzaInformazioniProprietario:** Si occupa della visualizzazione delle informazioni del proprietario di una sala o di uno strumento.

La pagina personale si suddivide in due sottosistemi:

- **VisualizzaRichieste:** Si occupa della visualizzazione e della risoluzione delle richieste di noleggio in sospeso
- **VisualizzaAnnunci:** Si occupa della visualizzazione degli annunci pubblicati dall'utente.

Infine è presente la sezione amministrativa, suddivisa in tre sottosistemi:

- **AnnunciInSospeso:** Si occupa dell'approvazione o dell'eliminazione di un annuncio.
- **GestisciAnnunci:** Si occupa della visualizzazione e dell'eventuale eliminazione degli annunci.
- **GestisciUtenti:** Si occupa della visualizzazione delle informazioni e dell'eventuale eliminazione degli utenti.



Servizi dei sottosistemi

Sottosistema	Autenticazione
Descrizione	Questo sottosistema permette l'accesso alla propria pagina personale.
Servizi offerti	
Servizio	Autenticazione
Descrizione	Permette di accedere alla pagina principale del sito e da qui usare le varie funzionalità.

Sottosistema	RicercaPerNome
Descrizione	Questo sottosistema permette di effettuare una ricerca di uno strumento, filtrando il nome.
Servizi offerti	
Servizio	Descrizione
	Effettua la ricerca di uno strumento.

Sottosistema	RicercaPerDescrizione
Descrizione	Questo sottosistema permette di effettuare una ricerca di uno strumento, filtrando la descrizione.
Servizi offerti	
Servizio	Descrizione
	Effettua la ricerca di uno strumento

Sottosistema	RicercaPerProvincia
Descrizione	Questo sottosistema permette di effettuare una ricerca di uno strumento, filtrando la provincia.
Servizi offerti	
Servizio	Descrizione
	Effettua la ricerca di uno strumento

Sottosistema	RicercaSalaPerProvincia
Descrizione	Questo sottosistema permette di effettuare una ricerca di una sala registrazioni, filtrando la provincia.
Servizi offerti	
Servizio	Descrizione
	Effettua la ricerca di una sala.

Sottosistema	InserisciAnnuncioStrumento
Descrizione	Questo sottosistema permette di inserire un annuncio per uno strumento musicale .
Servizi offerti	
Servizio	Descrizione
gestoreRichiestaFerie	Permette di inserire un annuncio.

Sottosistema	InserisciAnnuncioSala
Descrizione	Questo sottosistema permette di inserire un annuncio per uno sala registrazioni .
Servizi offerti	
Servizio	Descrizione
gestoreAvvisoMalattia	Permette di inserire un annuncio.

Sottosistema	NoleggiaStrumento
Descrizione	Questo sottosistema permette di inviare una richiesta di noleggio di uno strumento al rispettivo proprietario.
Servizi offerti	
Servizio	Descrizione
	Permette di inviare una richiesta di noleggio

Sottosistema	NoleggiaSala
Descrizione	Questo sottosistema permette di inviare una richiesta di noleggio di una sala registrazioni al rispettivo proprietario.
Servizi offerti	
Servizio	Descrizione
	Permette di inviare una richiesta di noleggio

Sottosistema	VisualizzaInfoProprietario
Descrizione	Questo sottosistema visualizza le informazioni del

	proprietario di una sala o di uno strumento.
Servizi offerti	
Servizio	Descrizione
	Mostra le informazioni di un utente

Sottosistema	VisualizzaRichieste
Descrizione	Questo sottosistema visualizza le richieste di noleggio in attesa ricevute dall'utente.
Servizi offerti	
Servizio	Descrizione
gestoreNotificheCambioLinea	Visualizza le richieste di noleggio in attesa.

Sottosistema	VisualizzaAnnunci
Descrizione	Questo sottosistema visualizza gli annunci pubblicati dall'utente.
Servizi offerti	
Servizio	Descrizione
	Visualizza gli annunci di un utente.

Sottosistema	AnnunciInSospeso
Descrizione	Questo sottosistema permette all'amministratore di approvare o negare gli annunci inseriti dagli utente.
Servizi offerti	
Servizio	Descrizione
gestoreNotificheFerie	Permette di approvare o eliminare un annuncio.

Sottosistema	GestisciAnnunci
Descrizione	Questo sottosistema permette all'amministratore di eliminare o visualizzare un annuncio.

Servizi offerti	
Servizio	Descrizione
gestoreNuovoTurno	Permette di eliminare un annuncio o di visualizzare le informazioni di un annuncio.

Sottosistema	GestisciUtenti
Descrizione	Questo sottosistema permette all'amministratore di eliminare un utente o visualizzarne le informazioni.
Servizi offerti	
Servizio	Descrizione
	Permette di eliminare un utente o di visualizzare le informazioni di un utente.

3.3 Hardware/Software mapping

Il sistema presenta principalmente l'interfaccia grafica dell'autenticazione. Ogni utente avrà la possibilità, mediante un dispositivo personale, di connettersi al sistema tramite una form di login. Come abbiamo precedentemente accennato il nostro sistema si basa su una architettura a tre livelli; il primo di questi è il livello di presentazione che è stato creato tramite HTML e Javascript per manipolare e gestire eventi ed animazioni degli elementi del DOM in maniera molto semplice. eRockOn è un sistema multiplatforma ovvero accessibile da qualsiasi web-browser, poiché tutto il codice è presente su un web-server che lo esegue. I linguaggi di programmazione utilizzati sono :

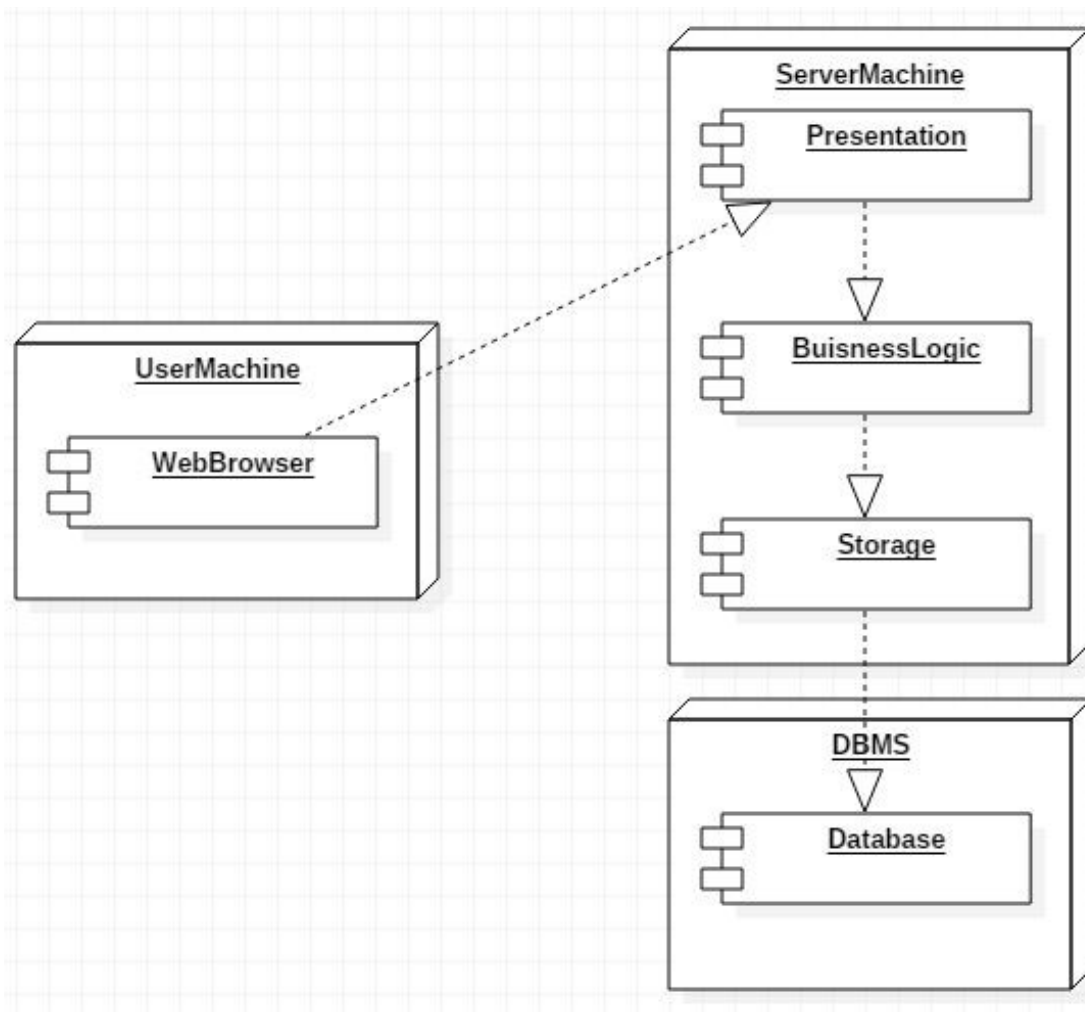
PHP per la gestione del login, la gestione delle sessioni dei vari utenti e per effettuare query al database;

JavaScript per prelevare i dati dalle form compilate dagli utenti e gestire eventi grafici nella sezione di amministrazione.

AJAX per visualizzare in tempo reale le informazioni degli utenti.

Il webserver utilizzato è Apache.

Rivolgiamo ora la nostra attenzione alla gestione della persistenza.



3.4 Gestione dei dati persistenti

Dopo un'analisi accurata siamo giunti alla conclusione che è necessario utilizzare un database relazionale ai fini di tener traccia della grande quantità di dati da gestire.

E' stato scelto un DBMS MySQL, che ci permette di abbattere i costi e i tempi, poiché non si deve progettare ed implementare tutta la logica per rendere persistenti i dati.

La connessione al database viene effettuata tramite il linguaggio php con il driver nativo MySQL.

Per effettuare delle query sul database si devono eseguire le seguenti istruzioni:

- `mysql_connect()`
- `mysql_select_db()`
- `mysql_query()`
- `mysql_fetch_assoc()`

L'istruzione `mysql_connect()` necessita dell'indirizzo(IP) del server e del nome utente e password; come risultato restituisce un numero che costituisce l'identificativo di connessione MySQL ("MySQL link identifier") in caso di successo, oppure FALSE in caso di fallimento.

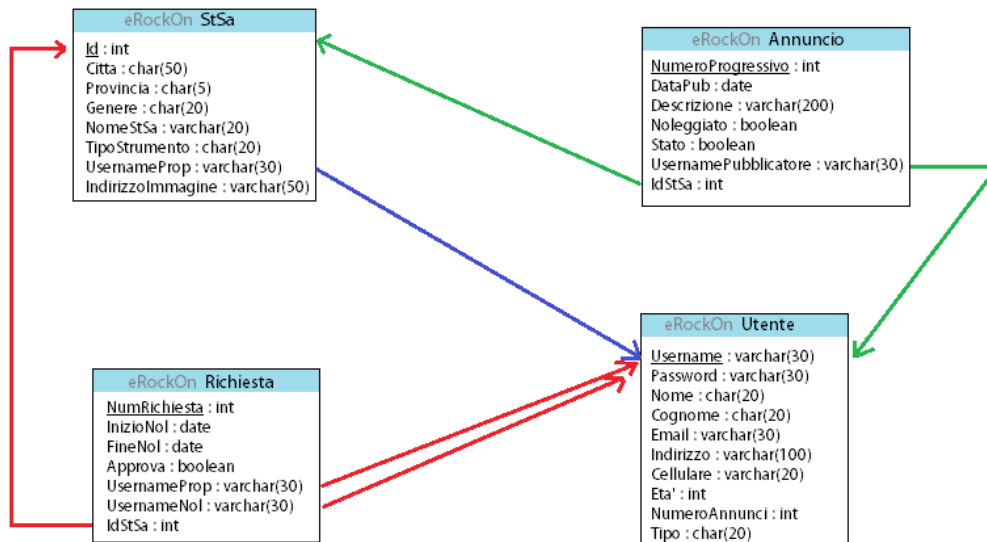
Dopo aver effettuato la connessione a MySQL si deve selezionare il database sulla quale si deve operare, tramite l'istruzione `mysql_select_db()`, che restituisce TRUE o FALSE come esito della selezione.

Per effettuare una query al database si utilizza la funzione `mysql_query()` che prende come parametro una query sql e la connessione effettuata con `mysql_connect()`, dopodiché si recuperano i risultati con `mysql_fetch_assoc()`; se tale funzione ha restituito TRUE.

Infine si deve terminare la connessione con il database con l'istruzione `mysql_close()`, passando in input il *MySQL link identifier* ottenuto da `mysql_connect()`.

3.4.1 Identificazione degli oggetti persistenti

I dati persistenti sono quei dati che devono sopravvivere alla singola esecuzione del sistema.



In particolare i dati identificati come persistenti sono:

- **Utente**: Questo oggetto memorizza tutti gli utenti registrati

#	Nome	Tipo	Null	Predefinito	Extra
1	<u>Username</u>	varchar(30)	No	Nessuno	
2	Password	varchar(30)	No	Nessuno	
3	Nome	char(20)	No	Nessuno	
4	Cognome	char(20)	No	Nessuno	
5	Email	varchar(30)	No	Nessuno	
6	Indirizzo	varchar(100)	No	Nessuno	
7	Cellulare	varchar(20)	No	Nessuno	
8	Età	int	No	Nessuno	
9	NumeroAnnunci	int	No	Nessuno	
10	Tipo	char(20)	No	Nessuno	

- **StSa:** Questo oggetto memorizza i dati di tutti gli strumenti/sale registrati

#	Nome	Tipo	Null	Predefinito	Extra
1	<u>Id</u>	int	No	Nessuno	
2	Citta	char(50)	No	Nessuno	
3	Provincia	char(5)	No	Nessuno	
4	Genere	char(20)	No	Nessuno	
5	NomeStSa	varchar(20)	No	Nessuno	
6	TipoStrumento	char(20)	No	Nessuno	
7	UsernameProp	varchar(30)	No	Nessuno	
8	IndirizzoImmagine	varchar(50)	No	Nessuno	

- **Richiesta:** Questo oggetto memorizza i dati di tutte le richieste effettuate

#	Nome	Tipo	Null	Predefinito	Extra
1	<u>NumRichiesta</u>	int	No	Nessuno	AUTO_INCREMENT
2	InizioNol	date	No	Nessuno	
3	FineNol	date	No	Nessuno	
4	Approva	boolean	No	Nessuno	
5	UsernameProp	varchar(30)	No	Nessuno	
6	UsernameNol	varchar(30)	No	Nessuno	
7	IdStSa	int	No	Nessuno	

- **Annuncio:** Questo oggetto memorizza i dati di tutti gli annunci presenti nel sistema

#	Nome	Tipo	Null	Predefinito	Extra
1	<u>NumeroProgressivo</u>	int	No	Nessuno	AUTO_INCREMENT
2	DataPub	date	No	Nessuno	
3	Descrizione	varchar(200)	No	Nessuno	
4	Noleggiato	boolean	No	Nessuno	
5	Stato	boolean	No	Nessuno	
6	UsernamePubblicatore	varchar(30)	No	Nessuno	
7	IdStSa	int	No	Nessuno	

3.4.2 Definizione SQL del database

Si rimanda al documento "Query SQL".

3.5 Access control and security

eRockOn è un sistema multi-user , in cui differenti attori possono accedere a differenti funzionalità e a differenti dati.

Gli attori del sistema sono :

- L'**utente**, accede alla piattaforma principalmente per visualizzare annunci di strumenti o sale, inoltre può inserire egli stesso annunci relativi a sale o strumenti , inviare richieste di noleggio e visualizzare tutte le richieste ricevute.
- L'**amministratore**, gestisce tutti gli annunci (approvati e non) degli utenti e gestisce gli utenti stessi.

Da questa breve descrizione si evince la necessità di capire in ogni momento “chi, può accedere a che cosa”.

Per rispondere alla prima parte della domanda, è necessario un modo per identificare l'utente che sta accedendo al sistema, quindi, nel momento in cui un generico utente tenta di accedere al sistema, si deve autenticare. L'autenticazione richiede un username ed una corrispondente password che è conosciuta solo da chi ha i permessi per l'accesso all'applicazione.

3.6 Global software control

Come abbiamo già detto l'architettura del nostro sistema è di tipo three-tier (tre livelli), in tal caso il controllo del flusso nei vari strati è indipendente dagli altri tranne che per la comunicazione col database. Il funzionamento del server è piuttosto semplice : esso è sempre in funzione, attende la richiesta di un servizio da parte di un sottosistema così da poterla subito soddisfare.

3.7 Boundary conditions

Il sistema garantisce l'esecuzione indipendente delle diverse funzionalità, essendo però un'applicazione dipendente da un server, il corretto funzionamento del sistema è strettamente legato alla sua attivazione; ognuno dei sottosistemi, infatti, usufruisce dei servizi offerti da questo ultimo, tra i quali l'utilizzo del database. Sono stati individuati due casi d'uso aggiuntivi:

StarteRockOnServer e StopeRockOnServer.

StarteRockOnServer : L'amministratore del sistema avvia la macchina server. Non appena il sistema si è avviato , gli utenti e gli amministratori possono accedervi ed utilizzare tutte le funzionalità.

StopeRockOnServer : L'amministratore del sistema arresta la macchina server, di conseguenza nessun utente può accedere alla piattaforma. Non è prevista nessuna attività di notifica ai sottosistemi attivi qualora uno di questi venga disattivato. Nella fase di terminazione inoltre, non è prevista un'operazione di aggiornamento del database poiché ogni singola postazione non gestisce operazioni sui dati in locale, ma effettua le modifiche in tempo reale.

Fallimenti

La prevenzione di eventuali danni causati dall'usura dell'hardware dedicato alla memorizzazione di dati persistenti è gestita da più unità di backup. Un eventuale crash del server comporta il riavvio di ognuno dei sottosistemi ed il ripristino del database allo stato precedente al crash stesso.