



**Università degli studi di Napoli  
Federico II**

**anno 2017/2018**

**PROGETTO INGEGNERIA DEL SOFTWARE**

**Progetto em\_17**

**AUTORI:**

**Gruppo 22**

Giuliano Vanesio N86001585  
Tommaso Pirozzi N86001735  
Marcello Quattromani N86001799  
Vincenzo Torino N86001374

# Sommario

<b>CAPITOLO 1: Documento analisi dei requisiti software .....</b>	<b>3</b>
REQUISITI FUNZIONALI.....	3
APPLICATIVO DESKTOP.....	3
APP MOBILE.....	9
MODELLO FUNZIONALE.....	10
Use Case diagram .....	10
Mock Up .....	12
Tabelle di Cockburn .....	22
MODELLI DI DOMINIO .....	28
Class diagram database .....	28
<b>CAPITOLO 2: Design e Object del Sistema.....</b>	<b>29</b>
ARCHITETTURA ESTERNA DEL SISTEMA.....	29
ARCHITETTURA INTERNA DEL SISTEMA.....	30
ARCHITETTURA INTERNA APP MOBILE .....	31
DESIGN PATTERN.....	32
Pattern ECB .....	32
Pattern DAO .....	32
Pattern Factory .....	33
Pattern Strategy .....	33
CLASS DIAGRAM APPLICATIVO DESKTOP .....	34
CLASS DIAGRAM APP MOBILE .....	36
SEQUENCE DIAGRAM .....	37
CRC CARDS APPLICATIVO DESKTOP .....	44
<b>CAPITOLO 3: Testing e Piano di test .....</b>	<b>50</b>

## CAPITOLO 1: Documento analisi dei requisiti software

Il sistema è composto da:

- un software per la gestione di eventi, clienti, addetti alla sicurezza, turni di lavoro e visualizzazione statistiche;
- un'applicazione mobile utile per gli addetti alla sicurezza, con cui potranno verificare la validità di un biglietto tramite la lettura del QR-CODE su di esso.

### REQUISITI FUNZIONALI

Per requisiti funzionali intendiamo i requisiti che descrivono i servizi o funzioni offerti dal sistema.

#### *APPLICATIVO DESKTOP*

#### Caratteristiche generali di un evento

L'evento avrà le seguenti caratteristiche:

1. il suo ID di identificazione;
2. il titolo;
3. la descrizione;
4. il tipo di evento;
5. il genere di evento in base al tipo;
6. la data;
7. il luogo in cui si svolge.

#### Gestione Eventi

Le operazioni inerenti all'evento sono gestite da parte di un amministratore.

Questo prima di compiere qualsiasi operazione, dovrà autenticarsi tramite il proprio username e la propria password (spiegazione in GESTIONE LOGIN). Dopo avere dunque effettuato l'accesso sarà indirizzato alla schermata principale che avrà delle etichette laterali dalla quale potrà scegliere il campo da gestire.

La prima etichetta è proprio quella inerente agli eventi, dove sarà possibile

- **CERCARE** genericamente un evento tramite qualsiasi suo attributo;
- cercare un evento tramite una **RICERCA AVANZATA**;
- **MODIFICARE** gli attributi di un evento dopo averlo trovato tramite ricerca;
- **CANCELLARE** uno o più eventi dopo averli trovati tramite ricerca;
- **CREARE** un evento.

**Il sistema deve permettere la ricerca di un evento. Tale ricerca può avvenire in due modi:**

1. La parola cercata verrà confrontata con qualsiasi attributo dell'evento così da avere una ricerca generica;
2. Tramite il pulsante ADVANCED SEARCH si può effettuare una ricerca avanzata. Questa permette dunque all'amministratore di eseguire delle ricerche più mirate e più filtrate.

**Il sistema deve permettere la modifica di un evento:**

- Una volta effettuata una ricerca, se questa va a buon fine mostrerà in una tabella i risultati;
- Gli eventi trovati potranno dunque essere modificati;
- Per apportare una modifica ad un campo di un evento basterà cliccare due volte su di esso e cambiare il suo valore con quello desiderato;
- Fatto ciò basterà premere il pulsante invio tramite il quale si darà conferma della modifica;
- Il campo modificato salverà il nuovo valore.

**Il sistema deve permettere la cancellazione di un evento:**

- Una volta effettuata una ricerca, se questa va a buon fine mostrerà in una tabella i risultati;
- Gli eventi trovati potranno dunque essere cancellati;
- Si può selezionare tramite mouse uno o più eventi che si vogliono cancellare;
- Una volta scelti gli eventi si potranno eliminare tramite il pulsante con il simbolo del cestino.

**Il sistema deve permettere la creazione di un evento:**

- Dalla schermata relativa agli eventi si può cliccare sul pulsante CREATE EVENT il quale aprirà una schermata adatta allo scopo.
- Dalla nuova finestra si dovranno riempire i campi relativi all'evento;
- Dopo aver inserito i dati dell'evento questo potrà essere creato cliccando sul tasto CREATE.

## **Caratteristiche generali di un Cliente**

Un cliente dovrà avere le seguenti caratteristiche:

1. ID per identificarlo
2. Username
3. Password
4. Nome
5. Cognome
6. Codice fiscale
7. Anno di nascita
8. Email
9. Numero cellulare

## **Gestione Clienti**

I clienti sono gestiti da un amministratore, previa autenticazione, che può CERCARLI, MODIFICARE i loro attributi, CANCELLARLI.

La ricerca può essere effettuata sia ricercando generalmente per tutti gli attributi, sia effettuando una ricerca avanzata.

Analogamente agli eventi, la cancellazione e la modifica degli eventi avviene dopo la ricerca di questi ultimi.

I clienti saranno già presenti nel database in quanto si creeranno in modo autonomo all'iscrizione sul sito apposito.

## **Il sistema deve permettere la ricerca di un cliente. Tale ricerca può avvenire in due modi:**

1. La parola cercata verrà confrontata con qualsiasi attributo di un cliente così da ottenere una ricerca generica;
2. Tramite il pulsante ADVANCED SEARCH, invece, si può effettuare una ricerca avanzata. Questa permette dunque all'amministratore di eseguire delle ricerche più mirate e più filtrate.

## **Il sistema deve permettere la modifica di un cliente:**

- Una volta effettuata una ricerca, se questa va a buon fine mostrerà in una tabella i risultati;
- I clienti trovati potranno a questo punto essere modificati;
- Per apportare una modifica ad un campo di un cliente basterà cliccare due volte su di esso e cambiare il suo valore con quello desiderato;
- Fatto ciò basterà premere il pulsante invio tramite il quale si darà conferma della modifica;
- Il campo modificato conterrà così il nuovo valore.

### **Il sistema deve permettere la cancellazione di un cliente:**

- Una volta effettuata una ricerca, se questa va a buon fine mostrerà in una tabella i risultati;
- I clienti trovati potranno dunque essere cancellati;
- Si può selezionare tramite mouse uno o più clienti che si vogliono cancellare;
- Una volta scelti si potranno eliminare tramite il pulsante con il simbolo del cestino.

### **Caratteristiche generali di un addetto alla sicurezza**

L'evento avrà le seguenti caratteristiche:

1. ID per identificarlo
2. Username
3. Password
4. Nome
5. Cognome
6. Codice fiscale
7. Anno di nascita
8. Email
9. Numero cellulare
10. Salario

### **Gestione Addetto alla sicurezza**

La gestione di un addetto alla sicurezza avviene in modo totalmente analogo a quella degli eventi.

Si potrà allora:

- **CERCARE** genericamente un addetto tramite qualsiasi suo attributo;
- cercare un addetto tramite una **RICERCA AVANZATA**;
- **MODIFICARE** gli attributi di un addetto dopo averlo trovato tramite ricerca;
- **CANCELLARE** uno o più addetti dopo averli trovati tramite ricerca;
- **CREARE** un addetto.

**Il sistema deve permettere la ricerca di un addetto alla sicurezza. Tale ricerca può avvenire in due modi:**

Analogamente agli eventi

**Il sistema deve permettere la modifica di un addetto alla sicurezza:**

Analogamente agli eventi

**Il sistema deve permettere la cancellazione di un addetto alla sicurezza:**

Analogamente agli eventi

### **Il sistema deve permettere la creazione di un addetto alla sicurezza:**

Analogamente agli eventi

### **Caratteristiche generali di un turno di lavoro**

Un turno di lavoro avrà le seguenti caratteristiche:

1. ID per identificarlo;
2. ID dell'addetto alla sicurezza;
3. ID dell'evento;
4. Orario inizio turno;
5. Orario fine turno.

### **Gestione turno di lavoro**

Un turno di lavoro può essere gestito da un amministratore (previa autenticazione) tramite l'apposita schermata. Si potrà a questo punto gestire i turni di lavoro, ricercandoli sia per un preciso addetto, sia per un preciso evento, sia per entrambi gli attributi. I risultati della ricerca verranno mostrati nella relativa tabella e potranno eventualmente essere rimossi tramite il pulsante del "DELETE" (rappresentato dal cestino) che si attiverà una volta selezionato uno o più turni. Per creare un nuovo turno di lavoro basterà soltanto compilare ogni campo stando attenti a non superare le 8 ore lavorative (in caso contrario apparirà un messaggio di errore) ed infine cliccare sul pulsante "SET WORK".

### **Il sistema deve permettere la ricerca di un turno di lavoro:**

- La ricerca può essere effettuata riempiendo solo il campo COGNOME relativo all'addetto oppure riempiendo solo il campo TITOLO relativo all'evento o ancora riempiendo entrambi i campi suddetti. Una volta rispettati i precedenti punti bisogna cliccare sul pulsante "CERCA" (rappresentato da una lente)

### **Il sistema deve permettere la creazione di un turno di lavoro:**

- Una volta scritto il cognome dell'addetto alla sicurezza, bisogna caricare il codice fiscale tramite il pulsante "🔄" così da essere sicuri di scegliere il giusto addetto in caso di omonimia;
- Stessa procedura avverrà anche per l'evento da voler selezionare;
- Una volta scelto l'addetto e l'evento, bisogna scegliere l'orario di inizio e fine turno;
- Infine per settare il turno di lavoro basta cliccare sul pulsante "SET WORK".

**Il sistema deve permettere la cancellazione di un turno di lavoro:**

- Dopo avere effettuato una ricerca, se questa è andata a buone fine, mostrerà i risultati in una tabella;
- Per cancellare uno o più turni di lavoro basta selezionarli nella tabella e cliccare sul pulsante “DELETE” (rappresentato dal cestino).

**Gestione Statistiche:**

Le statistiche potranno essere visualizzate da parte di un amministratore (previa autenticazione) selezionando il tipo da esaminare.

In particolare avremo quattro tipi di statistica:

1. Percentuale di presenze per tipo evento con età media;
2. Biglietti venduti negli anni;
3. Guadagni per anno;
4. Presenze negli anni per un tipo particolare di evento.

Le statistiche verranno mostrate tramite dei grafici che potranno essere o “PIE CHART” o “LINE CHART”.

**Il sistema deve permettere di mostrare le statistiche in base a quella scelta:**

- Percentuale di presenze per tipo evento con età media: in questo caso verrà generato un grafico cliccando sul pulsante “GENERATE CHART”;
- Biglietti venduti negli anni: bisognerà scegliere il range di anni su cui visualizzare le statistiche;
- Guadagni per anno: analogo al secondo;
- Presenze negli anni per un tipo particolare di evento: oltre alla selezione delle date bisognerà selezionare il tipo di evento sul quale si vuole esaminare le presenze.



## *APP MOBILE*

### **Gestione lettura biglietti tramite qrcode**

L'applicativo mobile deve essere in grado di leggere i QRcode presenti sui biglietti dei clienti che li hanno precedentemente acquistati.

La lettura del biglietto verrà effettuata da un addetto alla sicurezza, e l'applicativo comunicherà a quest'ultimo la validità del biglietto.

L'applicazione mobile dovrà dunque:

- **LEGGERE** un QR code mostrando dei messaggi che notificano la validità del biglietto.

### **Il sistema deve permettere la lettura di un biglietto:**

Dopo aver effettuato il login, un addetto potrà leggere il codice di un biglietto tramite dispositivo per verificare la validità di quest'ultimo.

## MODELLO FUNZIONALE

### *Use Case diagram*

Con il seguente use case diagram vogliamo descrivere quali sono i servizi e le funzioni offerti dal sistema.

Gli attori rilevati sono:

- Amministratore;
- Addetto alla sicurezza.

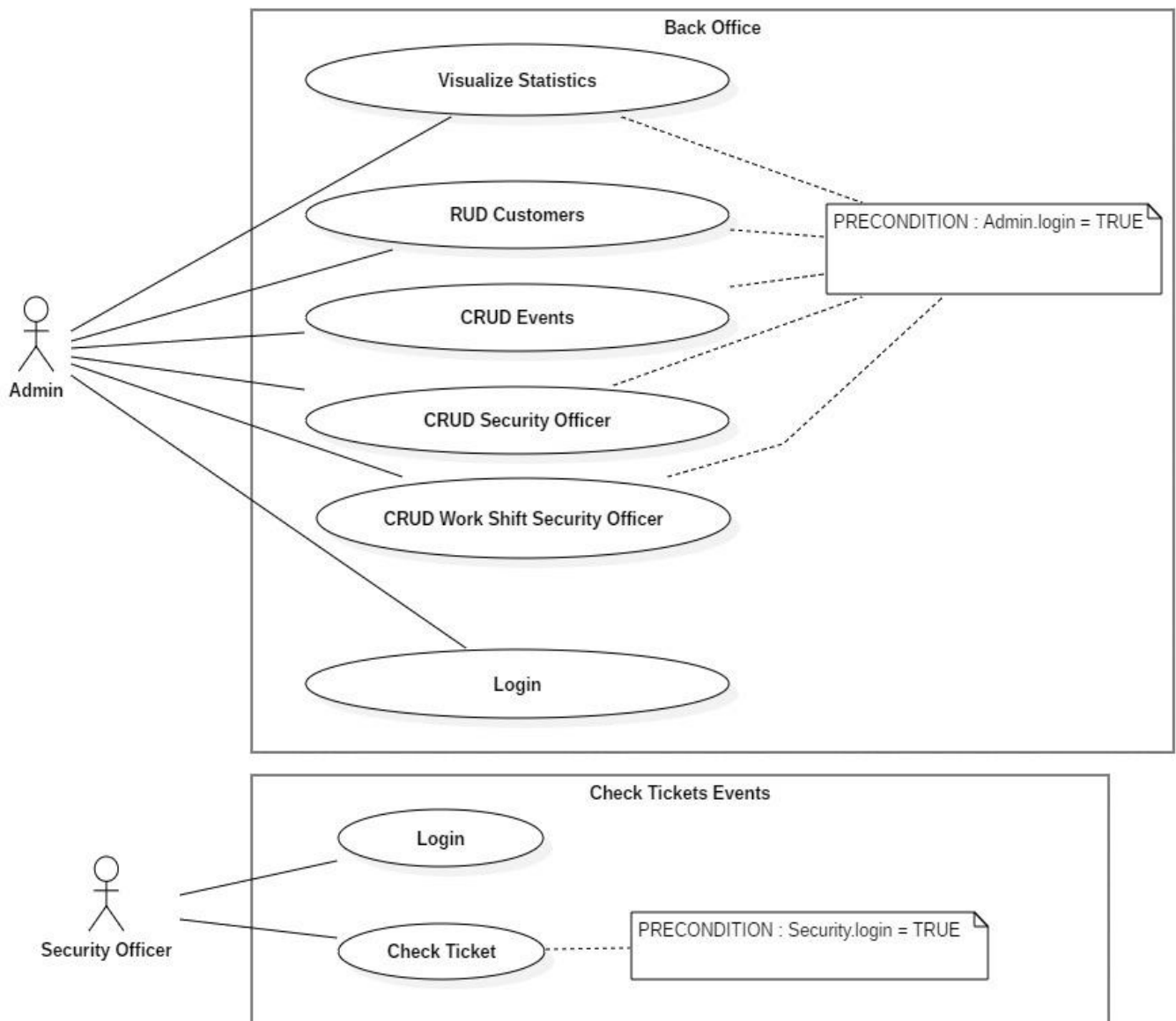
I casi d'uso riscontrati per l'amministratore sono:

1. CRUD evento;
2. CRUD addetto alla sicurezza;
3. RUD cliente;
4. Visualizzazione statistiche;
5. Gestione turni lavoro;
6. Effettua il Login.

I casi d'uso rilevati per l'addetto alla sicurezza sono:

1. Login in app;
2. Lettura biglietto tramite QR-code.

Nella pagina successiva sarà mostrato l'use case diagram realizzato.



## *Mock Up*

Nella seguente sezione viene illustrato come potrebbe presentarsi l'applicativo desktop e l'app mobile. L'idea grafica di base è stata di rendere il più semplice possibile l'utilizzo dei due software, i mock up sono una rappresentazione di massima dei sistemi sviluppati, il loro reale aspetto potrebbe differire.

I mock up presentati saranno:

- Event;
- Customer;
- Create Event;
- Advanced Search Event;
- Work Shift Officer;
- Statistics;
- Error message;
- Success message;
- Confirm message;
- Login app mobile;
- Scanner QRcode app mobile;
- Positive message;
- Negative message.

I mock up relativi all'addetto alla sicurezza non sono stati rappresentati in quanto analoghi a quelli dell'evento.

Di seguito verranno mostrati tutti i mock up rappresentati.

## Mock Up EVENT

EM17DB

File Edit View Help

Events

Clients

Statistics

Create Event

Advanced Search

Name	Type	Date	Places Available	Cod Event	Price	Description
Napoli-Lazio	Sport	12/12/2018	20000	001	40€	Description
Roma-Inter	Sport	12/12/2018	20000	002	30€	Description
JovanottiTour2018	Music	15/10/2018	10000	003	35€	Description

La modifica e la cancellazione di un evento avvengono cliccando due volte sull'evento cercato.

Mock Up CUSTOMER


EM17DB

File Edit View Help

Events

Q Nome

Advanced Search



Clients

Nome	Tipo	CF / P.IVA	Iscrizione	Tot Biglietti Ac.
Giacomo Guilizzoni	Privato	Codice Fiscale	12/12/12	2
Marco Botton	Privato	Codice fiscale	12/12/12	1
Mariah Maclachlan	Privato	Codice fiscale	12/12/12	3
Facebook SRL	Società	P.IVA	12/12/12	4

Security Off

Statistics

WorkShift

I clienti possono essere anche società che hanno dati differenti dalle persone fisiche

Mock Up CREATE EVENT

EM17DB - Create Event

Event Info

Code

Name

Type

Date

/ /

Places

Price

Description

Clear

Back

Create

## Mock Up ADVANCED SEARCH EVENT

EM17DB - Advanced Search


### Advanced Search

Code

Name

Type

Date



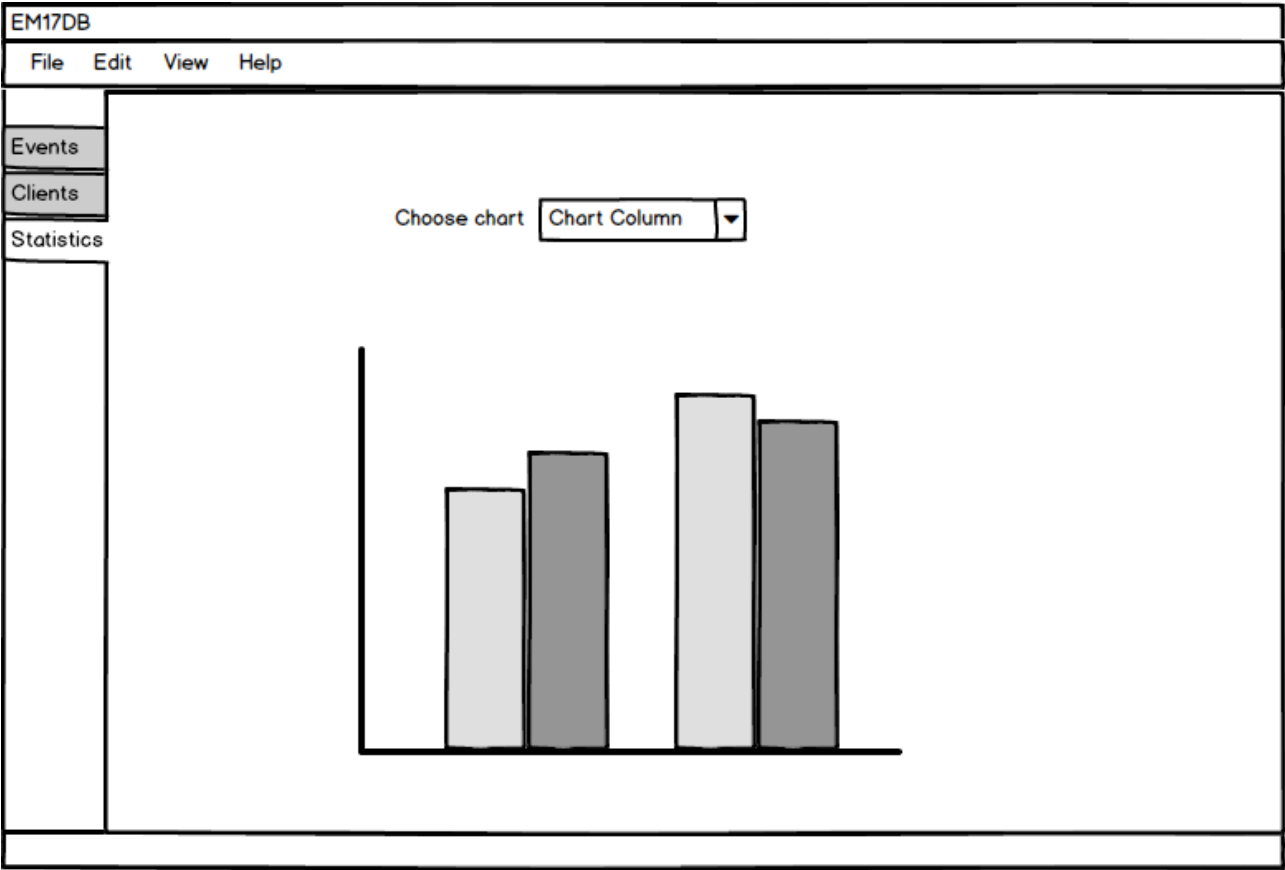
La ricerca avanzata avviene in real time compilando i campi in alto. Modifica e cancellazione avvengono cliccando su una riga



## Mock Up WORK SHIFT

[illegible]

Mock Up STATISTICS



Mock Up MESSAGES

Confirm

Are you sure ?

Cancel

Yes

Error Message

<Type Error> : <Message>

For more information ?

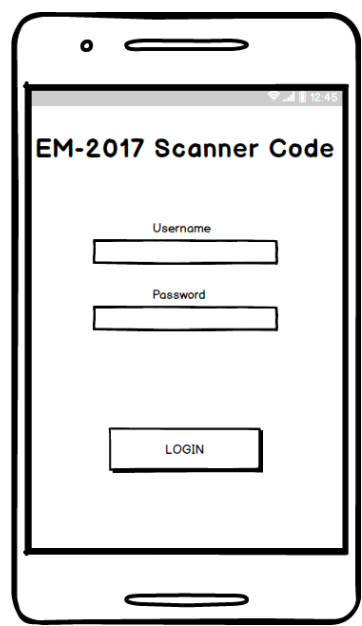
Ok

Success Message

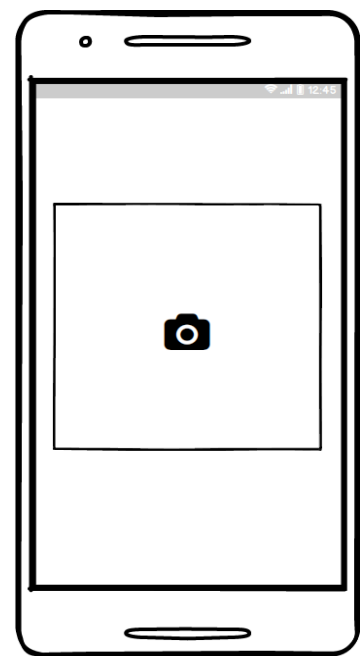
<Type Success> : <Message>

Ok

Mock Up Login app mobile

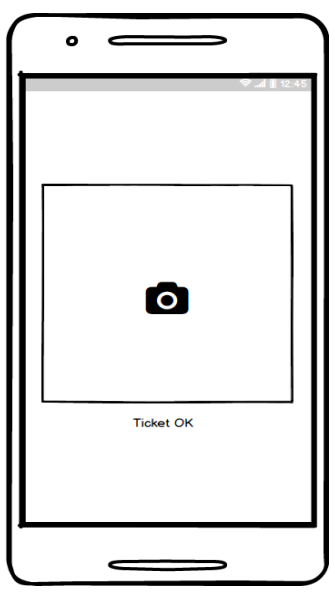


Mock Up Scanner

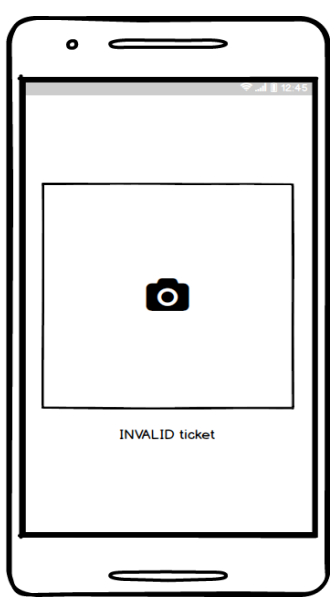


Mock Up Messages App mobile

Positive



Negative



## *Tabelle di Cockburn*

Di seguito verranno descritte le tabelle di Cockburn.

USE CASE #1	Check Tickets		
Goal in Context	L'addetto alla sicurezza vuole verificare il QrCode.		
Scope & Level			
Preconditions	Login effettuato con successo		
Success End Condition	Lettura Qr Code effettuata.		
Failed End Condition	Lettura Qr Code non andata a buon fine .		
Primary Actor	Addetto alla sicurezza		
Trigger	Lettura QR code		
DESCRIPTION	<b>Step</b>	<b>Addetto sicurezza</b>	<b>Sistema</b>
	1	Passa fotocamera su QR code	
	2		Scansiona QR code
	3		Mostra MockUp Positive Response
	4	Preme su schermo	
EXTENSION #1	<b>Step</b>	<b>Addetto sicurezza</b>	<b>Sistema</b>
	1		Mostra Errore Fallimento Apertura Fotocamera
	2	Riavvia applicazione	
SUBVARIATION #1	<b>Step</b>	<b>Addetto sicurezza</b>	<b>Sistema</b>
	3.1		Mostra MockUp Negative Response
	4.1	Preme su schermo	

USE CASE	<i>Creates Event</i>		
Goal in Context	L'amministratore vuole creare l'evento		
Scope & Level			
Preconditions	Login		
Success End Condition	L'evento viene creato		
Failed End Condition	L'evento non viene creato		
Primary Actor	Amministratore		
Trigger	Pressione pulsante "CREATE EVENT"		
DESCRIPTION	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	1	Preme il pulsante "CREATE EVENT"	
	2		Mostra Mock Up "CREATE EVENT"
	3	Riempie campi dell'evento	
	4	Preme pulsante "Create"	
	5		Crea evento
	6		Mostra Mock Up "Success Message"
	7	Preme tasto "OK"	
EXTENSION #1	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	5.1		Errore del sistema: Evento non creato
	6.1	Preme tasto "OK"	
	7.1		Ritorna allo step 2 dello scenario principale
SUBVARIATION #1	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	5.1		Mostra Mock Up "Error Message"
	6.1	Preme tasto "OK"	
	7.1		Ritorna allo step 2 dello scenario principale

USE CASE	Deletes Event		
Goal in Context	L'amministratore deve cancellare uno o più eventi		
Scope & Level			
Preconditions	Ricerca e seleziona eventi		
Success End Condition	Evento/i cancellato/i		
Failed End Condition	L'evento non viene cancellato		
Primary Actor	Amministratore		
Trigger	Preme pulsante "DELETE"		
DESCRIPTION	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	1	Preme pulsante "DELETE"	
	2		Mostra messaggio di conferma
	3	Preme pulsante "SI"	
	4		Elimina evento
	5		Mostra messaggio di avvenuta eliminazione
	6	Preme "OK"	
SUBVARIATION #1	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	3.1	Preme "NO"	



USE CASE	Update Events		
Goal in Context	L'amministratore deve modificare un evento		
Scope & Level			
Preconditions	Ricerca evento		
Success End Condition	L'evento viene modificato		
Failed End Condition	L'evento non viene modificato		
Primary Actor	Amministratore		
Trigger	Seleziona evento da modificare su tabella		
DESCRIPTION	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	1	Seleziona evento da modificare su tabella	
	2	Clicca due volte su attributo da modificare	
	3	Modifica attributo e preme tasto INVIO	
	4		Mostra messaggio di conferma
	5	Preme "SI"	
	6		Evento modificato\
SUBVARIATION #1	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	5.1	Preme "NO"	
SUBVARIATION #2	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	3.1	Preme tasto qualsiasi	
	4.1		L'evento non viene modificato

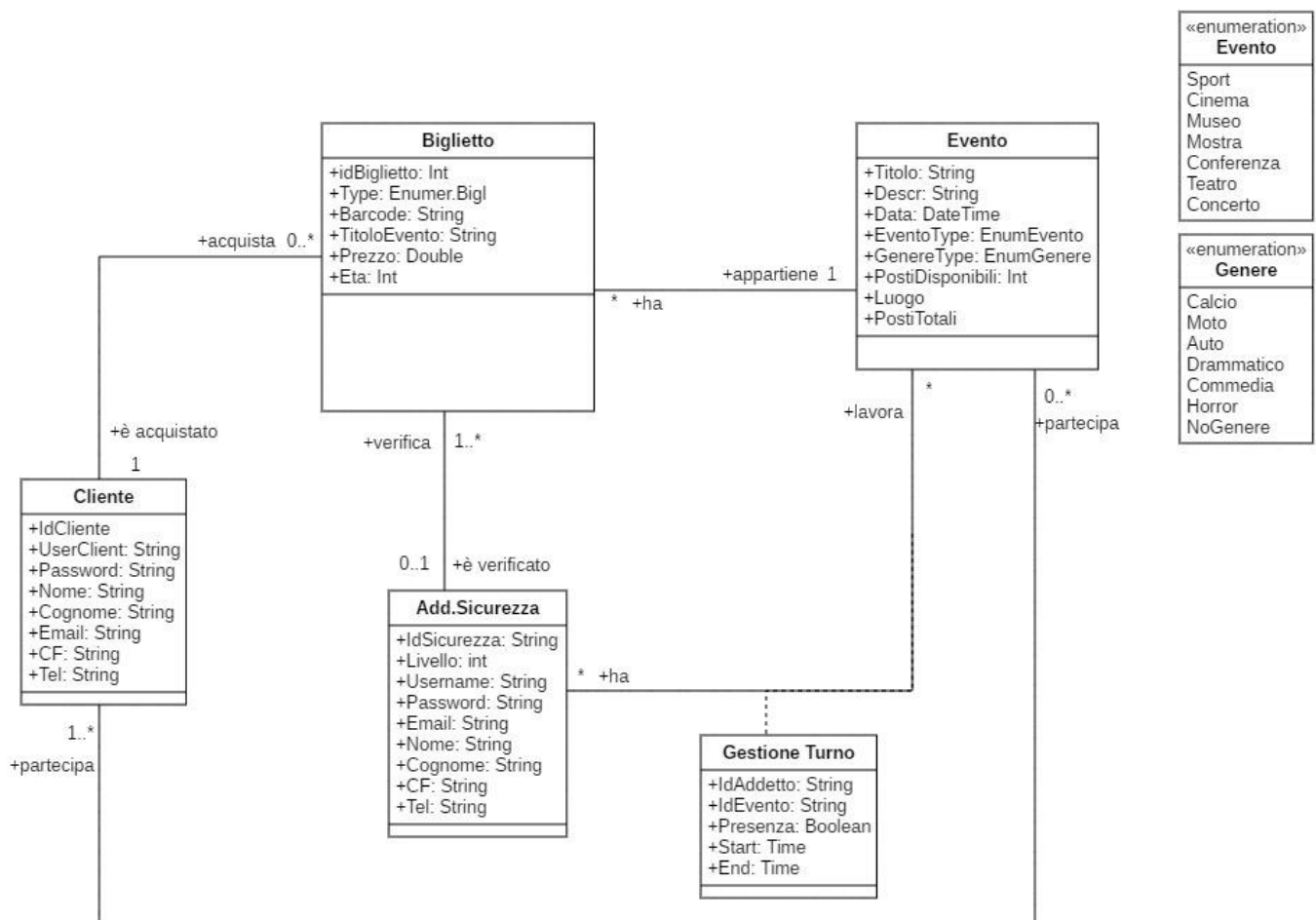
USE CASE	Search Event		
Goal in Context	L'amministratore cercare uno o più eventi		
Scope & Level			
Preconditions	Login		
Success End Condition	L'evento viene trovato		
Failed End Condition	L'evento non viene trovato		
Primary Actor	Amministratore		
Trigger	Preme pulsante "CERCA"		
DESCRIPTION	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	1	Preme pulsante "CERCA"	
	2		Mostra MockUp Advanced Search
	3	Riempie i campi per la ricerca	
	4	Preme tasto "CERCA"	
	5		Mostra risultati ricerca nella tabella
SUBVARIATION #1	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	5.1		Mostra messaggio errore: campi non riempiti correttamente

USE CASE	<i>Login</i>		
Goal in Context	Effettua Login		
Scope & Level			
Preconditions	L'addetto alla sicurezza deve possedere le credenziali		
Success End Condition	L'addetto alla sicurezza effettua il login		
Failed End Condition	Il login fallisce		
Primary Actor	Addetto sicurezza		
Trigger	Apre l'app		
DESCRIPTION	<b>Step n°</b>	<b>Attore 1</b>	<b>Sistema</b>
	1	Apre l'app	
	2	Compila campi "Username" e Password" e clicca bottone Login	
	3		Mostra MockUp Scanner
<b>SUBVARIATION #1</b>	<b>Step</b>	<b>Amministratore</b>	<b>Sistema</b>
	3.1		Mostra messaggio errore: User o password non corretti

## MODELLI DI DOMINIO

### Class diagram database

Di seguito il class diagram, rappresentato tramite UML, inerente alle relazioni delle classi (non dettagliate) presenti nel database.



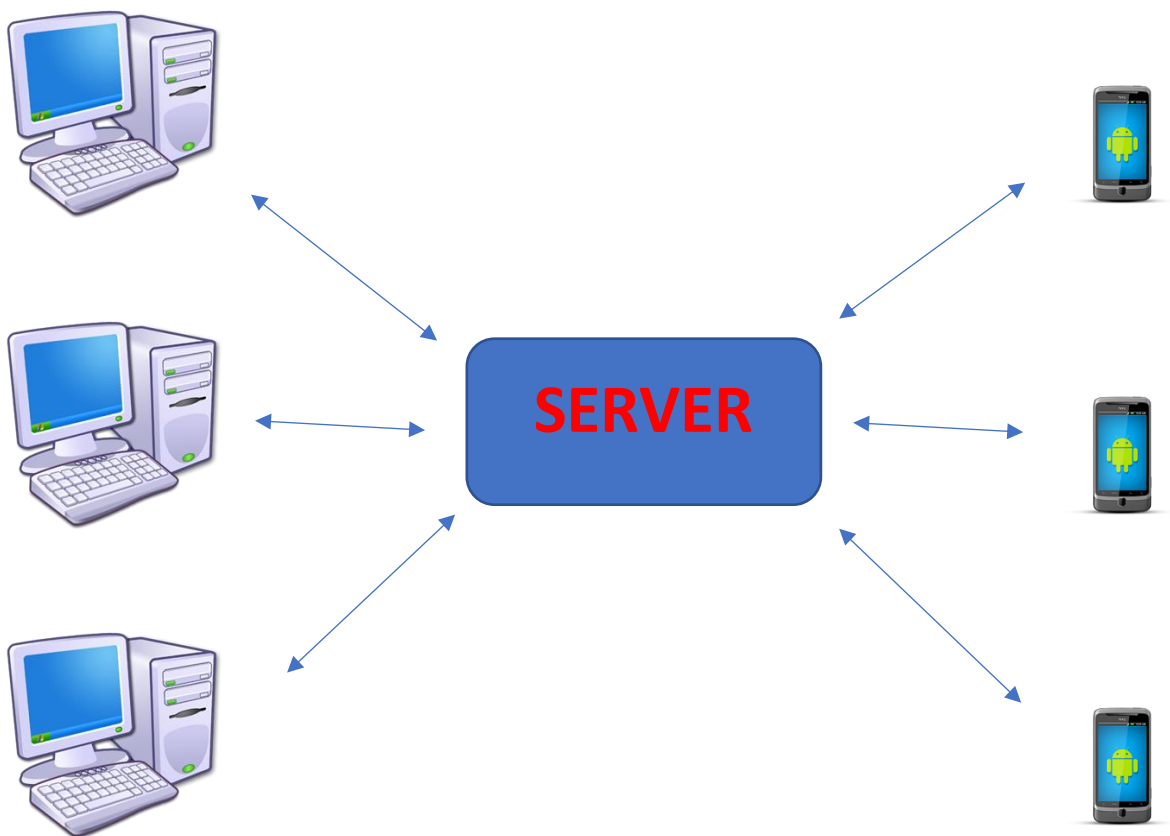
## CAPITOLO 2: Design e Object del Sistema

### ARCHITETTURA ESTERNA DEL SISTEMA

Il sistema presenta un'architettura repository: l'applicativo desktop e l'app mobile accedono e modificano il database memorizzato sul server.

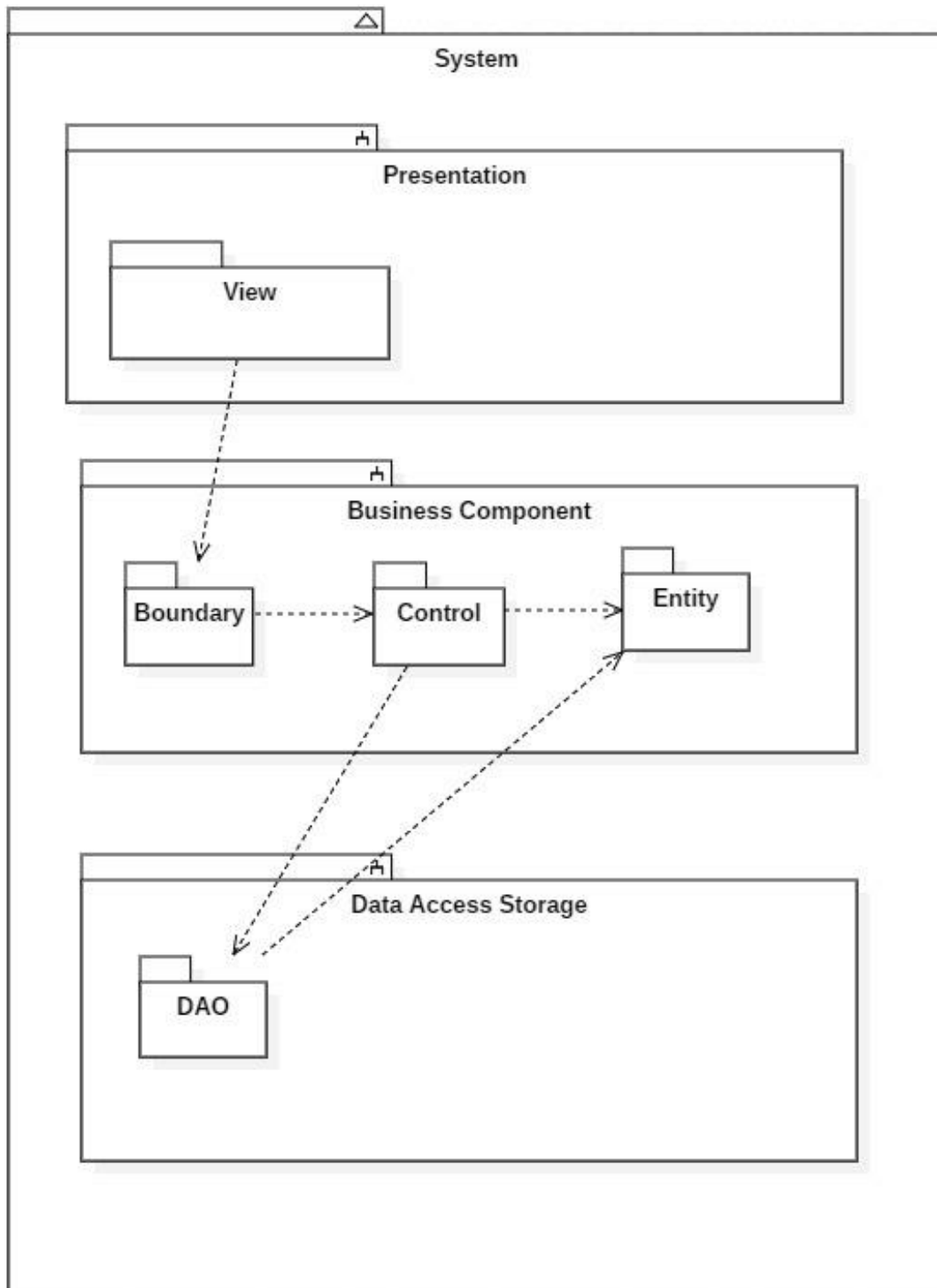
I sottosistemi sono indipendenti tra loro, difatti i due non comunicano mai tra di loro direttamente.

Il seguente schema presenterà il tipo di architettura:



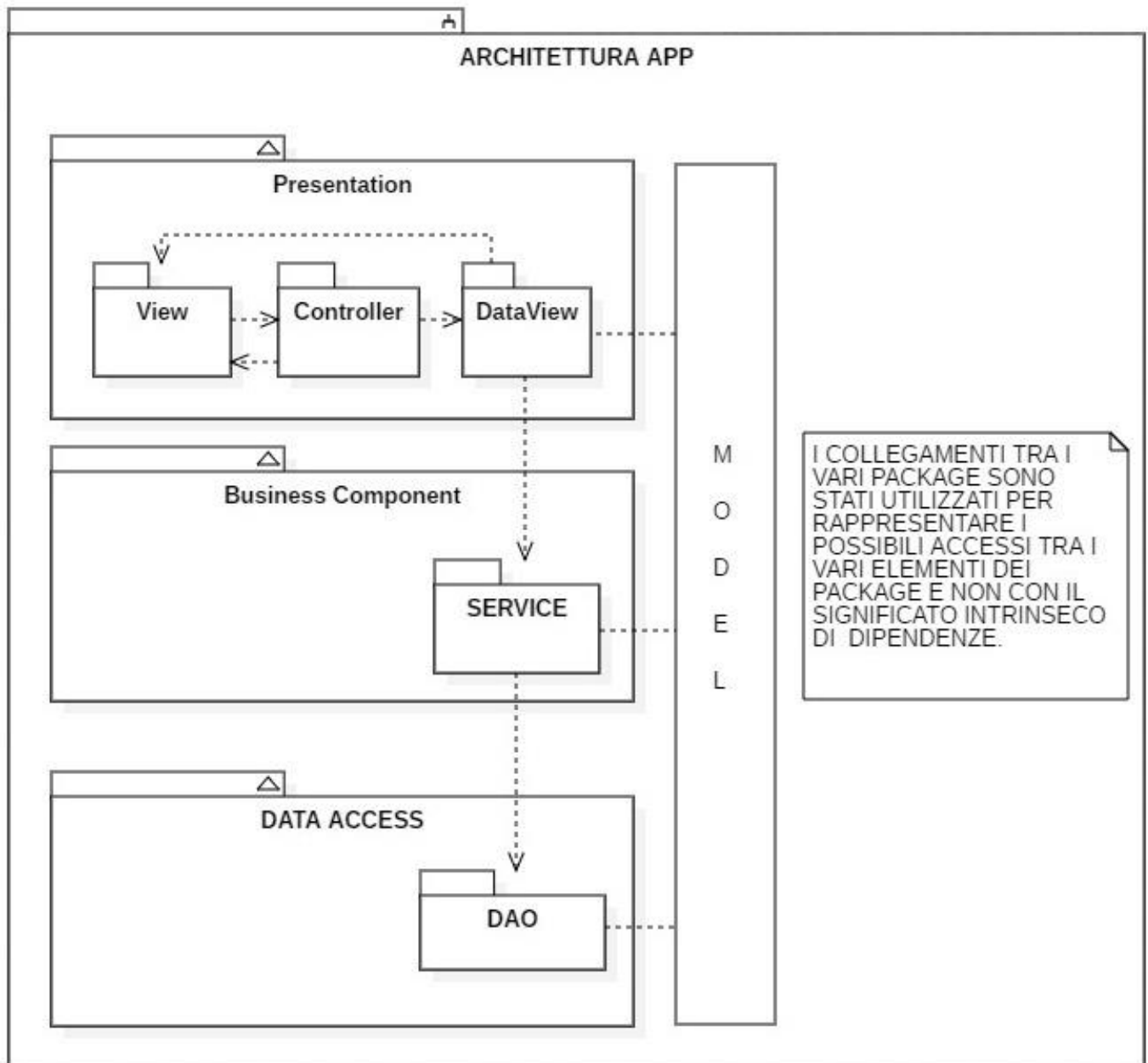
## ARCHITETTURA INTERNA DEL SISTEMA

L'architettura del software è organizzata in layer (per permettere il riutilizzo, la manutenibilità e la portabilità) tramite il seguente schema:



## ARCHITETTURA INTERNA APP MOBILE

L'architettura dell'app mobile è organizzata in layer secondo il seguente schema:



## DESIGN PATTERN

I design pattern utilizzati nel progetto sono:

- Entity Control Boundary (ECB);
- Data Access Object (DAO);
- Factory;
- Strategy.

### *Pattern ECB*

Il pattern ECB è un pattern architetturale composto da:

- Entity: oggetti che rappresentano i dati del dominio;
- Control: oggetti che mediano tra i boundary e gli entity;
- Boundary: oggetti che si interfacciano con gli attori.

Tale pattern è utile a disaccoppiare la logica applicativa dall'interfaccia grafica in modo tale da rendere riutilizzabile la logica funzionale. L'unico elemento non riutilizzabile è dunque il boundary, in quanto strettamente collegato al tipo di interfaccia dell'applicativo.

### *Pattern DAO*

Il pattern Data Access Object (DAO) è un pattern architetturale utilizzato per separare i servizi della logica applicativa dalle operazioni di accesso ai dati.

Il Data Access Object nasconde completamente i dettagli dell'interazione con la sorgente dati. L'interfaccia esposta dal DAO al client non cambia quando l'implementazione dell'origine dei dati sottostante cambia e questo consente al pattern di adattarsi a diversi schemi di archiviazione senza dover modificare nulla sugli altri livelli.

In sostanza, il DAO funge da adattatore tra il componente della logica applicativa e l'origine dati, permettendo tramite esso di disaccoppiare le operazioni di CRUD dalla logica funzionale.



## *Pattern Factory*

Il pattern Factory è un pattern creazionale nel quale possiamo individuare i seguenti componenti:

- **Creator:** dichiara la Factory che avrà il compito di ritornare l'oggetto appropriato;
- **ConcreteCreator:** effettua l'overwrite del metodo della Factory al fine di ritornare l'implementazione dell'oggetto;
- **Product:** definisce l'interfaccia dell'oggetto che deve essere creato dalla Factory;
- **ConcreteProduct:** implementa l'oggetto in base ai metodi definiti dall'interfaccia Product.

Tale pattern è stato utilizzato per la generazione dei grafici relativi alle statistiche.

## *Pattern Strategy*

Il pattern Strategy è un pattern comportamentale di oggetti, utilizzato quando si vogliono implementare una famiglia di algoritmi con un medesimo scopo, ma con diverse varianti. Infatti possiamo creare delle classi di algoritmi che implementano in modo diverso uno stesso algoritmo oppure possiamo creare delle nuove classi di algoritmi.

Tale pattern è stato utilizzato per gli oggetti DAO.

## **CLASS DIAGRAM APPLICATIVO DESKTOP**

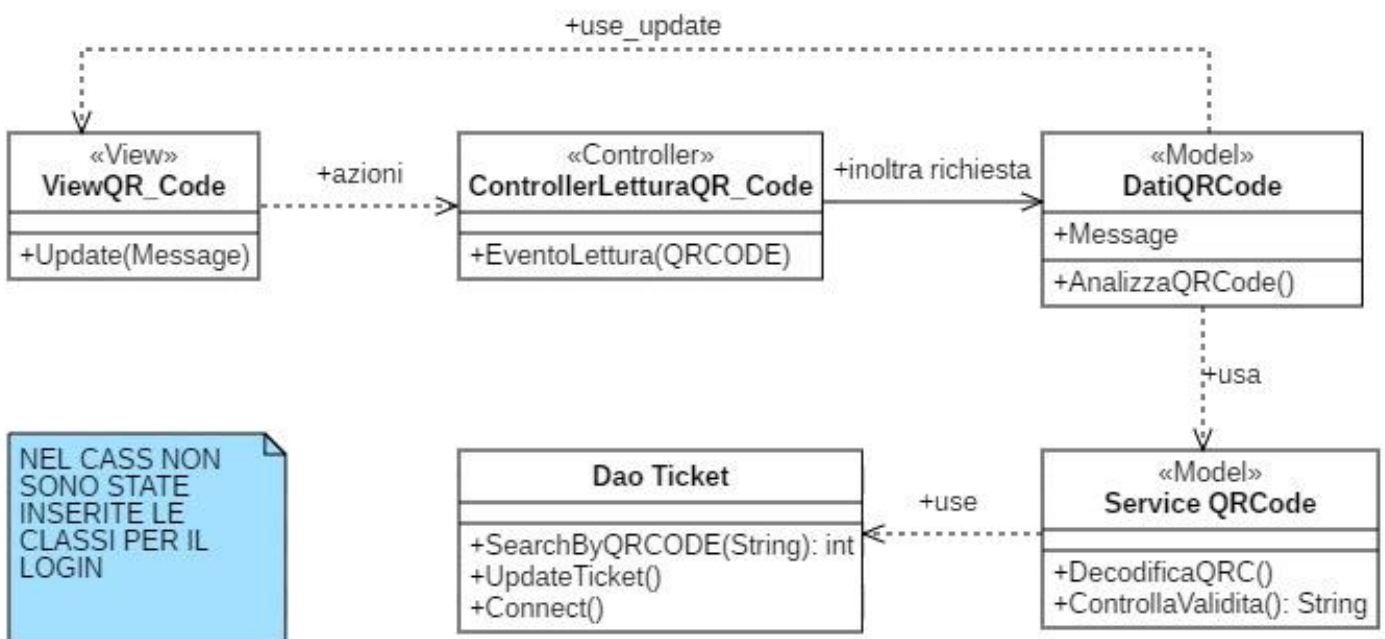
Per quanto riguarda lo sviluppo dell'applicativo desktop sono stati classificati tre tipi di oggetti come Entity, Boundary, Control in quanto è stato utilizzato il pattern architetturale ECB.

Nella pagina successiva viene mostrato il class diagram realizzato.



## CLASS DIAGRAM APP MOBILE

Di seguito il class diagram inerente all'app mobile realizzato utilizzando il pattern architetturale MVC.



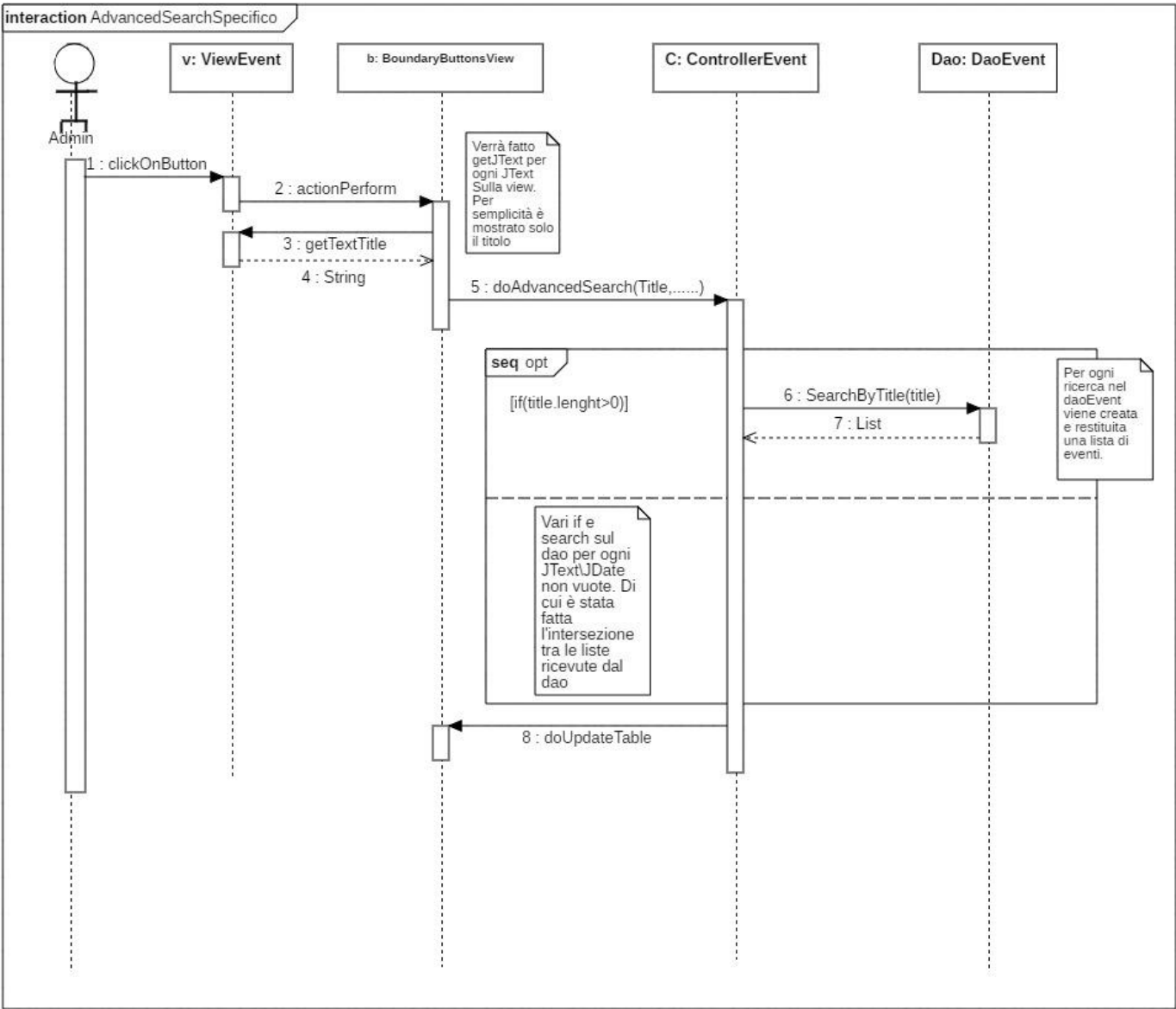
## SEQUENCE DIAGRAM

In questa sezione verranno analizzati i sequence diagram realizzati per l'applicativo desktop e mobile. In particolare verranno dettagliati i sequence diagram relativi al caso d'uso attinente agli eventi.

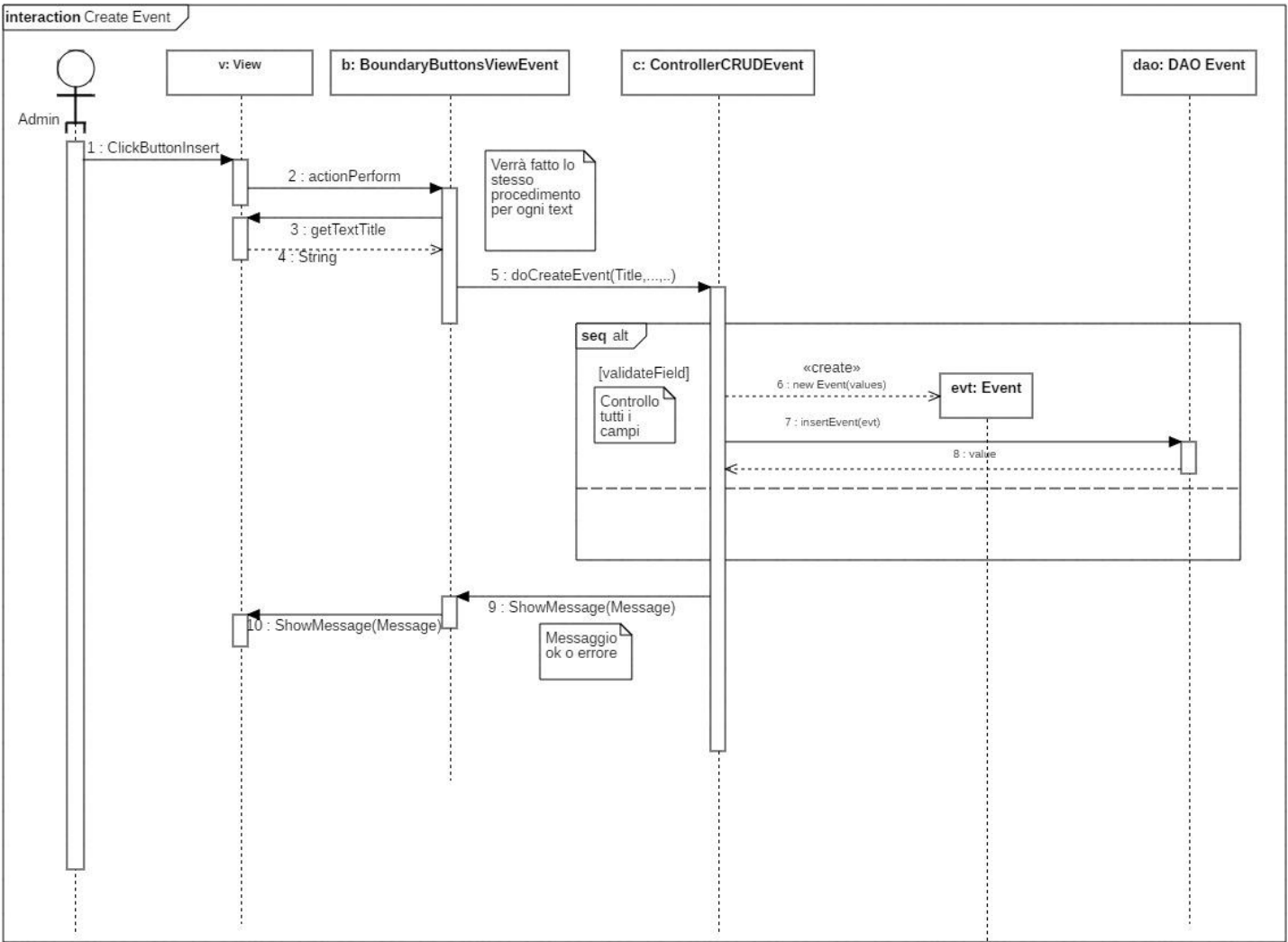
I sequence diagram allegati nelle pagine successive riguardano:

1. Ricerca avanzata;
2. Creazione di un evento;
3. Cancellazione di uno o più eventi;
4. Modifica di un evento;
5. Statistiche;
6. Controllo QR Code (App mobile).

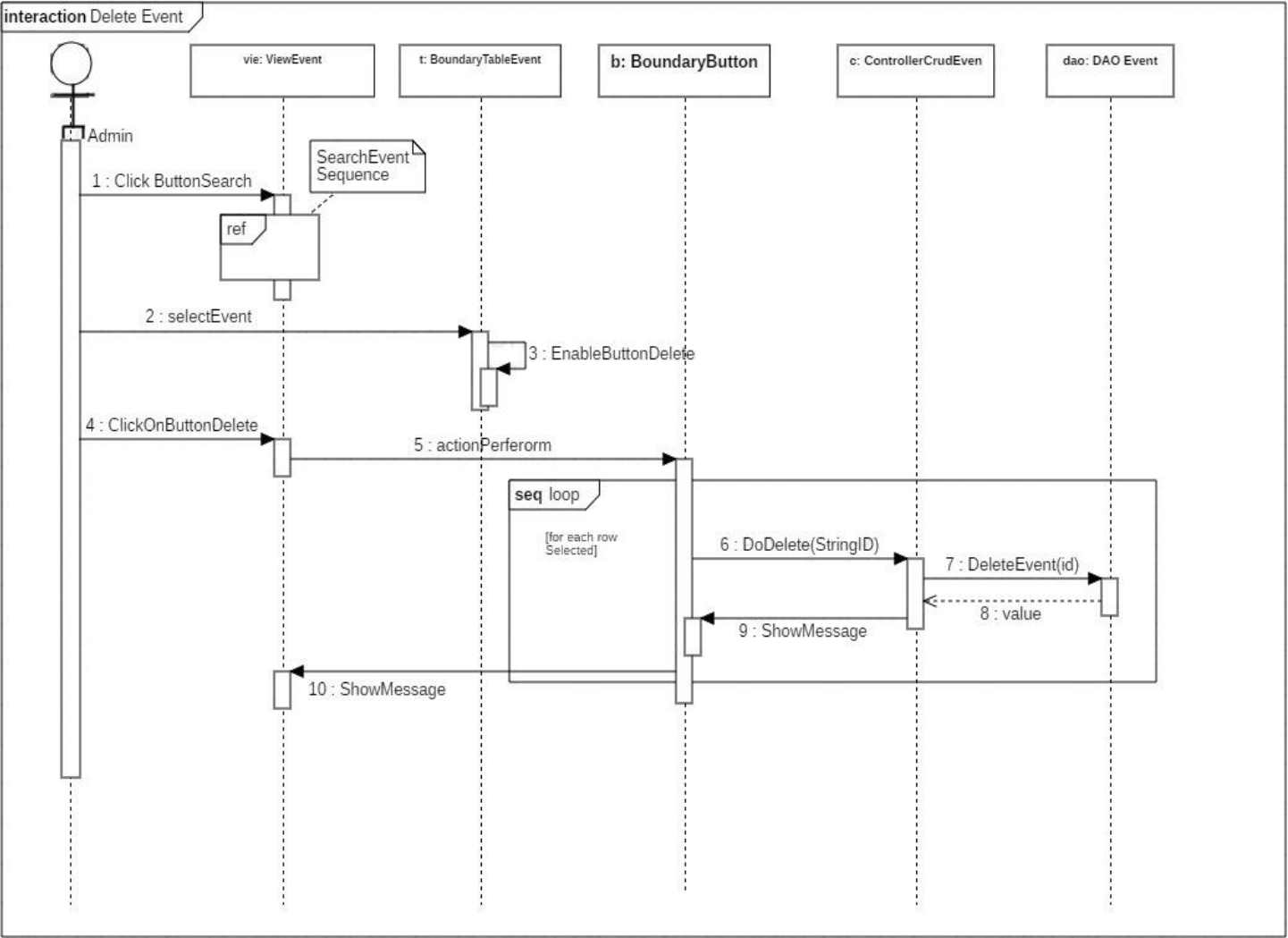
Sequence diagram: Ricerca avanzata



Sequence diagram: Creazione Evento

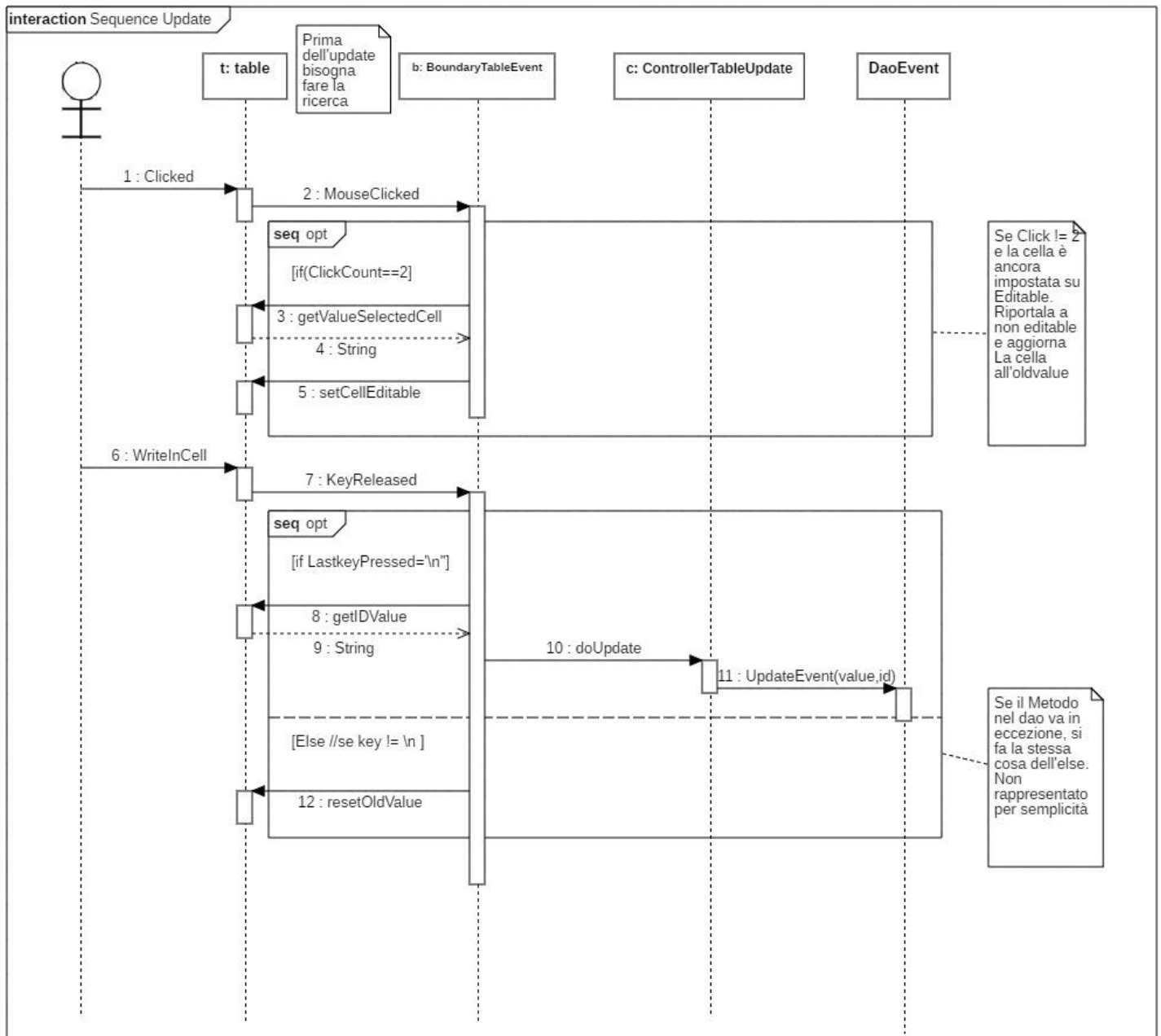


Sequence diagram: Cancellazione evento

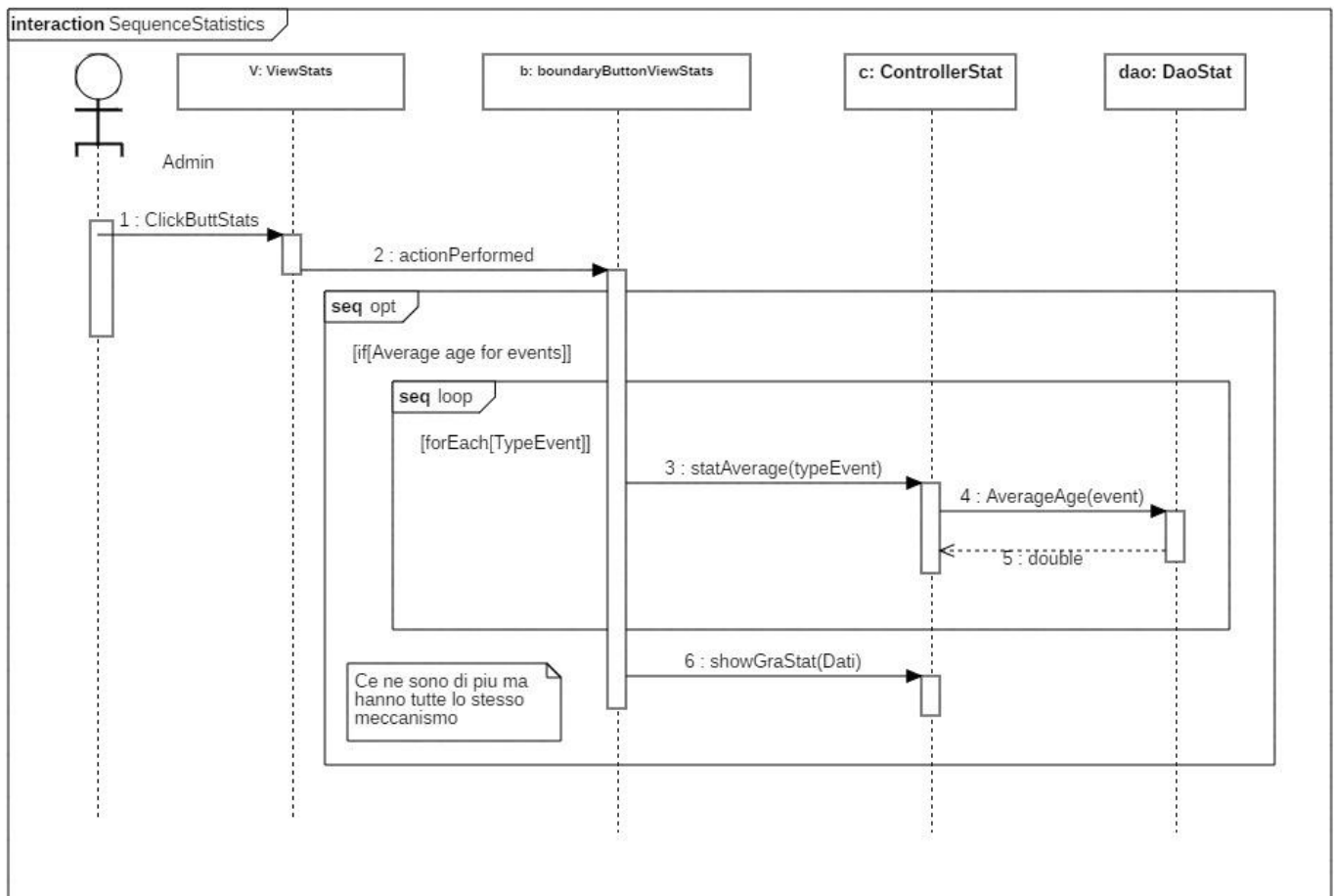




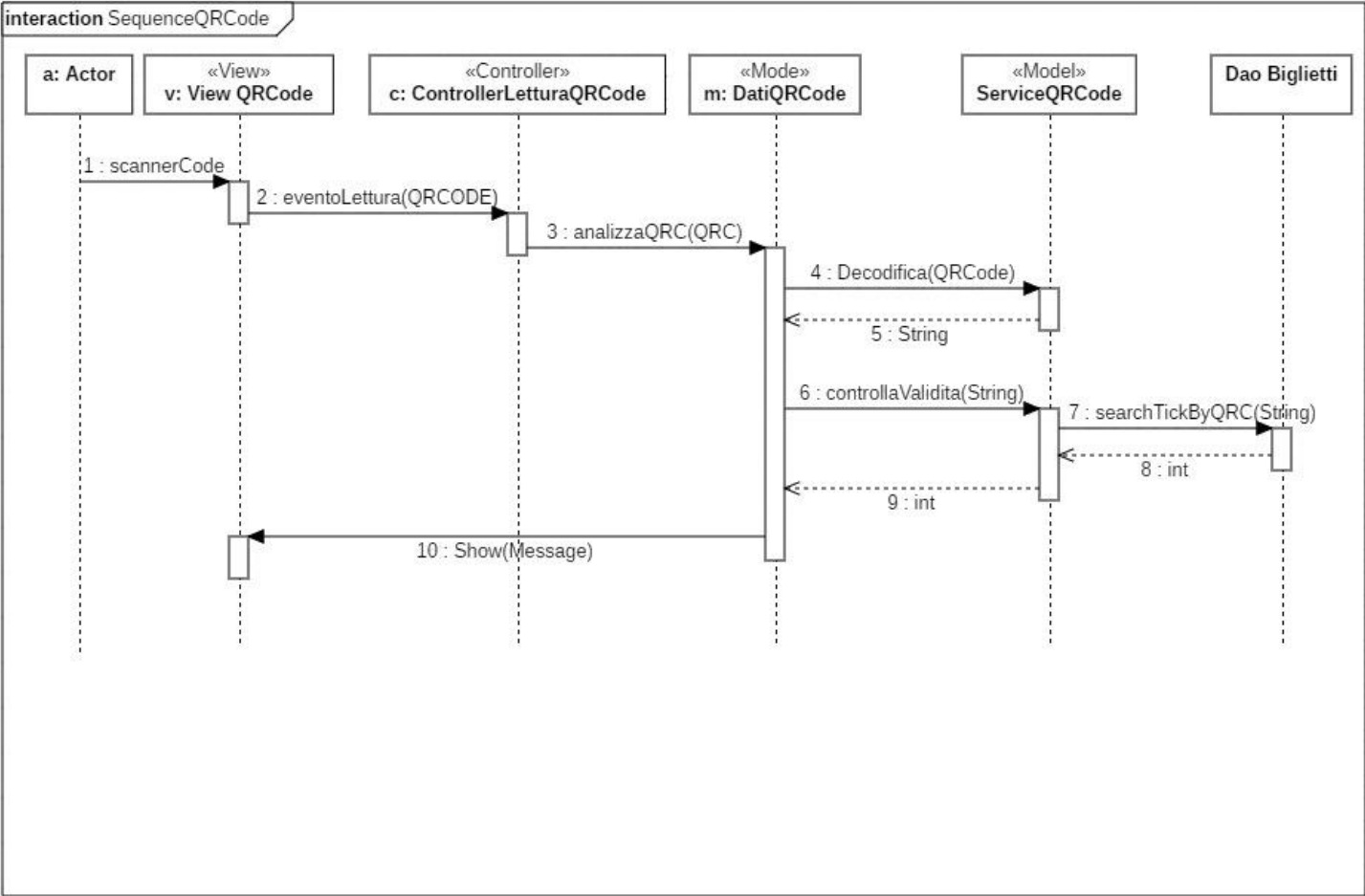
## Sequence diagram: Modifica evento



## Sequence diagram: Statistiche



Sequence diagram: Controllo QR Code



## CRC CARDS APPLICATIVO DESKTOP

Verranno di seguito introdotte le CRC cards (Class Responsibility Collaboration) per dettagliare le collaborazioni tra le classi e le loro responsabilità.

<b>Class name</b>	ImplMySQLEventDao
<b>Super Class</b>	Implements EventDao
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe ImplMySQLEventDao rappresenta i dati correlati agli eventi presenti sul database. E' stata creata seguendo le direttive del design pattern: DataAccessObject.	Event

<b>Class name</b>	ImplMySQLAddettiDao
<b>Super Class</b>	Implements AddettiDao
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe implMySQLAddettiDao rappresenta i dati correlati agli addetti alla sicurezza presenti sul database. E' stata creata seguendo le direttive del design pattern: DataAccessObject.	Addetto

<b>Class name</b>	implMySQLCustomerDao
<b>Super Class</b>	Implements CustomerDao
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe implMySQLCustomer Dao rappresenta i dati correlati ai clienti presenti sul database. E' stata creata seguendo le direttive del design pattern: DataAccessObject.	Customer

<b>Class name</b>	implMySQLTurnDao
<b>Super Class</b>	Implements TurnDao
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe implMySQLTurnDao rappresenta i dati correlati ai turni degli addetti agli eventi presenti sul database. E' stata creata seguendo le direttive del design pattern: DataAccessObject.	Turn Event Addetto

<b>Class name</b>	ListenerPanel (Abstract)
<b>Super Class</b>	Implements KeyListener, ActionListener, MouseListener
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe si occupa di gestire i vari listener dei Panel. Per esempio attivazione e disattivazione di button etc. E' stata resa classe in quanto comune a diversi panel.	GeneralPanel.

<b>Class name</b>	ControllerCRDEvent
<b>Super Class</b>	ListenerPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControllerCRDEvent è un controllore del caso d'uso "CRUDEvent" descritto nella documentazione e si occupa di creare, eliminare e cercare gli eventi. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Control"	EventDao EventPanel Event

<b>Class name</b>	ListenerTable (Abstract)
<b>Super Class</b>	Implements MouseListener, KeyListener
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ListenerTable è una classe che permette di gestire i listener di un oggetto JTable.	JTable

<b>Class name</b>	ControllerUpdateEvent
<b>Super Class</b>	ListenerTable
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControllerUpdateEvent è un controllore del caso d'uso "CRUDEvent" descritto nella documentazione e si occupa di effettuare update degli eventi. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Control"	EventPanel Event Dao Event

<b>Class name</b>	ControllerCRDAddetto
<b>Super Class</b>	ListenerPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleCRDEvent è un controllore del caso d'uso "CRUDAddetti" descritto nella documentazione e si occupa di creare,eliminare e cercare gli addetti. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll"	AddettiDao AddettiPanel. Addetto

<b>Class name</b>	ControllerUpdateAddetto
<b>Super Class</b>	ListenerTable
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleCRDEvent è un controllore del caso d'uso "CRUDAddetti" descritto nella documentazione e si occupa di fare update degli addetti. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll".	AddettiDao AddettiPanel. Addetto

<b>Class name</b>	ControllerCRUDTurn
<b>Super Class</b>	ListenerTable
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleCRUDTurn è un controllore del caso d'uso "CRUDTurn" descritto nella documentazione e si occupa di creare,eliminare e ricercare i turni di un evento o di un addetto. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll".	TurnDao ManagementTurnView Addetto Evento

<b>Class name</b>	ControllerRDCustomer
<b>Super Class</b>	ListenerPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleRDEvent è un controllore del caso d'uso "RUDCustomer" descritto nella documentazione e si occupa di creare ed eliminare i customer. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll".	CustomerDao CustomerPanel Customer

<b>Class name</b>	ControllerUpdateCustomer
<b>Super Class</b>	ListenerTable
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleRDEvent è un controllore del caso d'uso "RUDCustomer" descritto nella documentazione e si occupa degli update di un customer. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll".	Customer Dao. CustomerPanel. Customer

<b>Class name</b>	ControllerStatistics
<b>Super Class</b>	Implements ActionListener,ItemListener
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleStatistics è un controllore del caso d'uso "Check Statistics" descritto nella documentazione .	DaoStatistics PanelStats

<b>Class name</b>	ControllerLogin
<b>Super Class</b>	ListenerPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
ControlleLogin è un controllore del caso d'uso "Login" descritto nella documentazione e si occupa di far effettuare il login al database. Le sue responsabilità sono quelle tipiche di un controllore nella rappresentazione "Entity-Boundary-Controll".	LoginView

<b>Class name</b>	DBConnect
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
La classe si occupa di creare la connessione all'database e, quando richiesto, fornire la connessione creata.	-

<b>Class name</b>	MainView
<b>Super Class</b>	
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Frame che contiene un JTabbedPane, la quale conterrà tutti i panel presenti nel software.	

<b>Class name</b>	GeneralPanel(Abstract)
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Classe astratta che contiene elementi di grafica.	

<b>Class name</b>	EventPanel
<b>Super Class</b>	GeneralPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia grafica per gestire gli eventi.	

<b>Class name</b>	AddettiPanel
<b>Super Class</b>	GeneralPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia grafica per gestire gli addetti.	

<b>Class name</b>	CustomerPanel
<b>Super Class</b>	GeneralPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia grafica per gestire i clienti.	

<b>Class name</b>	ManagementTurnView
<b>Super Class</b>	GeneralPanel
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia grafica per gestire i turni degli addetti.	



<b>Class name</b>	LoginView
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia grafica che permette il login all'applicativo.	

<b>Class name</b>	StatsPanel
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Interfaccia che permette di visualizzare le statistiche.	

<b>Class name</b>	Customer
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Contiene Tutti i dati del singolo customer.	

<b>Class name</b>	Addetto
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Contiene Tutti i dati del singolo Addetto.	

<b>Class name</b>	Event
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Contiene Tutti i dati del singolo evento.	

<b>Class name</b>	Turn
<b>Super Class</b>	-
<b>Sub Class</b>	-
<b>Responsabilità</b>	<b>Collaboratori</b>
Contiene Tutti i dati del singolo turno	

## CAPITOLO 3: Testing e Piano di test

### DAO EVENT TEST

#### PIANO TEST 1: testato tramite JUnit

TEST METODO : SEARCHBYTITLE(String Title) .

#### DESCRIZIONE

L'utente inserisce il titolo dell'evento e la ricerca deve restituire tutte le righe degli eventi che hanno come titolo quello inserito dall'utente.

#### ESITO ATTESO

Eventi con titolo passati in input.

#### ESITO NON ATTESO

Titoli di eventi diversi da quello inserito.

#### CLASSI DI EQUIVALENZA "TITLE"

1. CU1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi , caratteri speciali)
2. CU2: nessun valore.

#### TEST CASES STRATEGIA WECT:

TEST	TITLE	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	Harry Potter	CU1	Tutte le tuple di evento nel database con titolo Harry Potter	OK	-
TC2	Da Paura	CU1	Tutte le tuple di evento con titolo Da Paura	OK	-
TC3	-	CU2	Nessun valore	OK	-
TC4	XXX	CU1	Nessun valore trovato	OK	-
TC5	*****	CU1	Nessun valore trovato	OK	-

## **METODI :**

- **SEARCHBYPLACE**(String place);
- **SEARCHBYTYPEEVENT**(String typeEvent);
- **SEARCHBYKINDEVENT**(String kindEvent).

I piani di test sono analoghi a SEARCHBYTITLE

## **PIANO TEST 2: testato tramite JUnit**

**TEST METODO : CREATE\_EVENT(Event e)**

### **DESCRIZIONE**

Il metodo permette di inserire l'evento e all'interno del database.

### **ESITO ATTESO**

Inserimento effettuato con successo.

### **ESITO NON ATTESO**

Inserimento non effettuato.

### **CLASSI DI EQUIVALENZA "TITLE"**

1. CT1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi)
2. CT2: Caratteri speciali
3. CT3: Nessun valore.

### **CLASSI DI EQUIVALENZA "DATE EVENT"**

4. CD1: Qualsiasi data successive a quella odierna
5. CD2: Data odierna
6. CD3: Qualsiasi data precedente a quella odierna
7. CD4: Nessun Valore

### **CLASSE DI EQUIVALENZA "TYPE\_EVENT"**

8. CTE1: QUALSIASI VALORE TRA 'CINEMA', 'SPORT', 'CONCERT', 'THEATER' E 'OTHER'
9. CTE2: QUALSIASI ALTRO VALORE.

## CLASSE DI EQUIVALENZA "KIND\_EVENT"

10. CKE1: QUALSIASI VALORE TRA "FOOTBALL", 'TENNIS', 'BASKET', 'VOLLEYBALL', 'SWIMMING', 'POPANDROCK', 'METAL', 'MUSICAL', 'CABARET', 'COMEDY', 'DRAMATIC', 'HORROR' E 'OTHER'

11. CKE2: QUALSIASI ALTRO VALORE.

DESCRIZIONE E LUOGO SONO ANALOGHI AL TITLE.

## TEST CASE STRATEGIA MISTO TRA WECT E SECT

TEST	TITLE	DATA	TYPE EVENT	KIND EVENT	CE	RISULTATO ATTESO	RISULTATO TEST
WECT1	Prova1	5/06/1990	Cinema	Commedy	CT1 ∩ CD3 ∩ CTE1 ∩ CKE1	SQLException. The entered date is already passed.	OK
WECT2		5/06/2020	Sport	Football	CT3 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error Invalid title format.	OK
WECT3	*****	Oggi	Sport	Swimming	CT2 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error Invalid title format	OK
WECT4	Harry Potter	Oggi	Ciao		CT1 ∩ CD1 ∩ CTE2 ∩ CKE2	SQLException : Invalid (enumeration) Type Event	BUG FIXED
W*ECT5	Harry Potter 3	7/06/2020	Cinema	Other	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	Inserimento ok	OK
WECT6	Napoli- Juventus	6/03/2019	Sport	Football	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	Inserimento ok	OK
WECT7	Napoli- Juventus	6/03/2019	Sport	Commedy	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error : You must associate Football, Swimming ... to sport	OK
WECT8	Pino Daniele	Oggi	Concert		CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException. Error : You must associate PopAndRock, Metal ... to Concert	OK

WECT9	Prova*_*	Oggi			CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException. Enumeration error on Type_Event	OK
WECT10	Jovanotti		Concert	PoP&Rock	CT1 ∩ CD2 ∩ CTE1 ∩ CKE1	SQLException : Date is null	OK
WECT11	Jovanotti	25/07/2022	Concert	Xxxxxxxx	CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException : Enumeration error Kind_Event	OK
WECT11	Jovanotti	-	Concert	Xxxxxxxx	CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException : Data non può essere null	OK

### PIANO TEST 3: testato tramite JUnit

**METODO:** DELETE\_EVENT(String idEvent)

#### DESCRIZIONE

Il metodo permette di eliminare l'evento con quell'id all'interno del database.

#### ESITO ATTESO

Delete effettuato con successo.

#### ESITO NON ATTESO

Delete non effettuato.

#### CLASSI DI EQUIVALENZA "IDEVENT"

1. CE1 : Valore nominale
2. CE2 : Nessun valore

#### TEST CASE STRATEGIA SECT

TEST CASE	IDEVENT	CE	RISULTATO ATTESO	RISULTATO OTTENUTO	BUG_FIXING
SECT1	Aaaaaaabbcc	CE1	Eliminazione non effettuata (chiave non presente)	OK	
SECT2		CE2	Eliminazione non effettuata (chiave non presente)	OK	
SECT3	2	CE1	Eliminazione effettuata con successo	OK	

## **PIANO DI TEST 4**

**METODO: UPDATE\_EVENT(Event e)**

### **DESCRIZIONE**

Il metodo permetto di modificare sul database i campi dell'evento selezionato.

### **ESITO ATTESO**

Update effettuato con successo.

### **ESITO NON ATTESO**

Update non effettuato.

### **CLASSI DI EQUIVALENZA "TITLE"**

1. CE1 : Valore nominale
2. CE2 : Nessun valore

### **CLASSI DI EQUIVALENZA "DATE"**

3. CD1 : Date successiva di quella odierna
4. CD2 : Nessun valore
5. CD3 : Date precedenti di quella odierna

### **CLASSI DI EQUIVALENZA "Type\_Event"**

6. CTE1 : Valore Consentiti
7. CTE2 : Altri valori

### **CLASSE DI EQUIVALENZA "Kind\_Event"**

8. CKE1 : Valori Consentiti
9. CKE2 : Altri valori

Kind Event e Type event sono enumerazioni.

Descrizione e luogo hanno le stesse caratteristiche degli altri, per semplicità evitiamo di testarli.

**STRATEGIA MISTA TRA WECT E SECT.**

TEST	TITLE	DATA	TYPE EVENT	KIND EVENT	CE	RISULTATO ATTESO	RISULTATO TEST
WECT1	Prova1	5/06/1990	Cinema	Commedy	CT1 ∩ CD3 ∩ CTE1 ∩ CKE1	SQLException. The entered date is already passed.	
WECT2		5/06/2020	Sport	Football	CT3 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error Invalid title format.	
WECT3	*****	Oggi	Sport	Swimming	CT2 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error Invalid title format	
WECT4	Harry Potter	Oggi	Ciao		CT1 ∩ CD1 ∩ CTE2 ∩ CKE2	Error : Invalid Type Event	
W*ECT5	Harry Potter 3	7/06/2020	Cinema	Other	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	Update ok	
WECT6	Napoli- Juventus	6/03/2019	Sport	Football	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	Update ok	
WECT7	Napoli- Juventus	6/03/2019	Sport	Commedy	CT1 ∩ CD1 ∩ CTE1 ∩ CKE1	SQLException. Error : You must associate Football, Swimming ... to sport	
WECT8	Pino Daniele	Oggi	Concert		CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException.: Enumeration error on Kind_Event	
WECT9	Prova*_*	Oggi			CT1 ∩ CD1 ∩ CTE2 ∩ CKE2	SQLException :. Enumeration error on type Event	



WECT10	Jovanotti	15/01/2019	Concert	Prova	CT1 ∩ CD1 ∩ CTE1 ∩ CKE2	SQLException : Enumeration error on kind Event	
WECT11	Jovanotti		Concert	Pop&Rock	CT1 ∩ CD2 ∩ CTE1 ∩ CKE1	SQLException : Date is null.	

## PIANO DI TEST 5

### METODI : SEARCHBYDATE (Date from , Date to)

#### DESCRIZIONE

Il metodo permette di cercare sul database eventi compresi tra due date.

#### ESITO ATTESO

Eventi compresi tra le due date

#### ESITO NON ATTESO

Eventi non compresi tra le due date

#### CLASSI DI EQUIVALENZA "DATE FROM"

1. CDF1 : Qualsiasi Data
2. CDF2 : Nessuna Data

#### CLASSI DI EQUIVALENZA "DATE FROM"

3. CDT1 : Qualsiasi Data
4. CDT2: Nessuna Data

**STRATEGIA SECT**

TEST	DATE FROM	DATE TO	CLASSI EQUIVAL.	RISULTATO ATTESO	RISULTATO OTTENUTO	BUG FIXED
Sect1	1-07-2018	31-07-2018	CDF1 ∩ CDT1	Tutti gli eventi tra le due date	OK	
Sect2			CDF2 ∩ CDT2	Nessun evento trovato	OK	
Sect3	25-03-2018		CDF1 ∩ CDT2	Nessun evento trovato	OK	
Sect4		22-04-2018	CDF2 ∩ CDT1	Nessun evento trovato	OK	

**LOGIN TEST APP:****DESCRIZIONE**

L'addetto per poter effettuare il login deve prima riempire in maniera corretta i campi di testo "username" e "password".

**ESITO ATTESO**

L'utente inserisce utente e password corretti e procede con il login.

**ESITO NON ATTESO**

L'utente inserisce valori NON corretti e non può procedere con il login.

**CLASSI DI EQUIVALEZA "USERNAME"**

CU1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi)

CU2: nessun valore.

**CLASSI DI EQUIVALEZA "PASSWORD"**

CP1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi)

CP2: nessun valore.

**TEST CASES:**

TEST	USERNAME	PASSWORD	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	prove	prove	CU1 U CP1	Login eseguito	Ok	-
TC2	root	root	CU1 U CP1	Username o password errati	Ok	-
TC3	admin	-	CU1 U CP2	Errore	Ok	-
TC4	-	-	CU2 U CP2	Errore	Ok	-
TC5	-	password	CU2 U CP1	Errore	OK	-
TC6	Pippo 20	pluto	CU1 U CU1	Errore	Crash Applicazione	Fixato il 10/08/18
TC7	Pippo10	Pluto 19	CU1 U CU1	Errore	Crash Applicazione	Fixato il 10/08/18

**Nota per test grafico**

Per gli elementi dell'interfaccia grafica si consiglia di rilasciare una versione beta dell'applicativo ad un numero ristretto di clienti, in modo tale da ricevere dei feedback in merito ad eventuali bug puramente grafici.