



**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

CAPITOLO 1: Documento analisi dei requisiti software .....	3
REQUISITI FUNZIONALI .....	3
MODELLO FUNZIONALE .....	9
Use Case diagram .....	9
Mock Up .....	11
Tabelle di Cockburn .....	19
MODELLI DI DOMINIO .....	25
Class diagram database .....	25
CAPITOLO 2: Design e Object del Sistema .....	26
ARCHITETTURA ESTERNA DEL SISTEMA .....	26
ARCHITETTURA INTERNA DEL SISTEMA .....	27
DESIGN PATTERN .....	28
Pattern ECB .....	28
Pattern DAO .....	28
Pattern Factory .....	29
Pattern Strategy .....	29
CLASS DIAGRAM APPLICATIVO DESKTOP .....	30
CLASS DIAGRAM APP MOBILE .....	32
SEQUENCE DIAGRAM .....	33
CRC CARDS APPLICATIVO DESKTOP .....	40
CAPITOLO 3: Testing e Piano di test .....	46

## 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.

#### 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.



## 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.

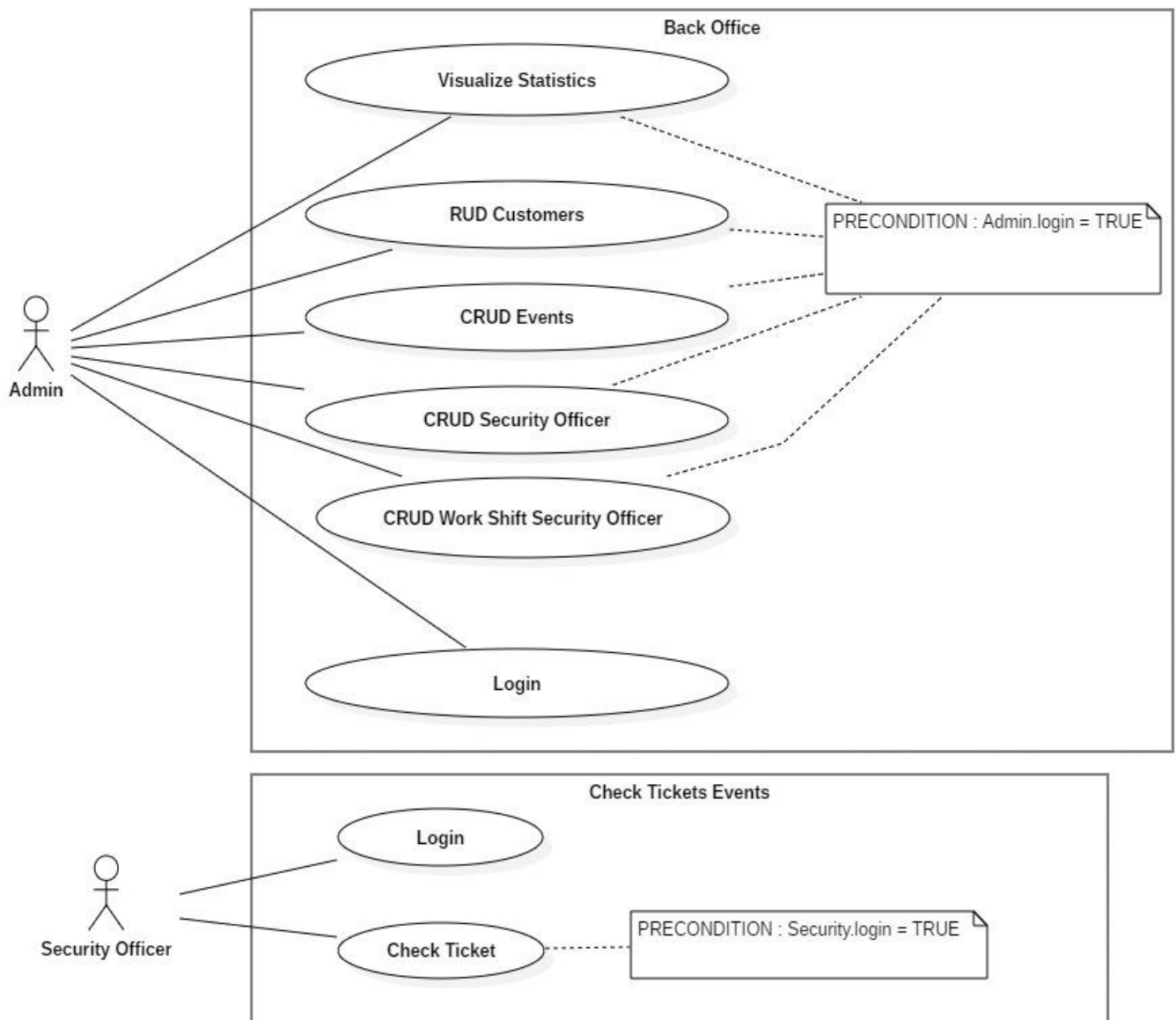
I casi d'uso riscontrati per l'amministratore sono:

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.

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

Security Off	Nome	Tipo	CF / P.IVA	Iscrizione	Tot Biglietti Ac.
Statistics	Giacomo Guilizzoni	Privato	Codice Fiscale	12/12/12	2
WorkShift	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

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

/ /

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			

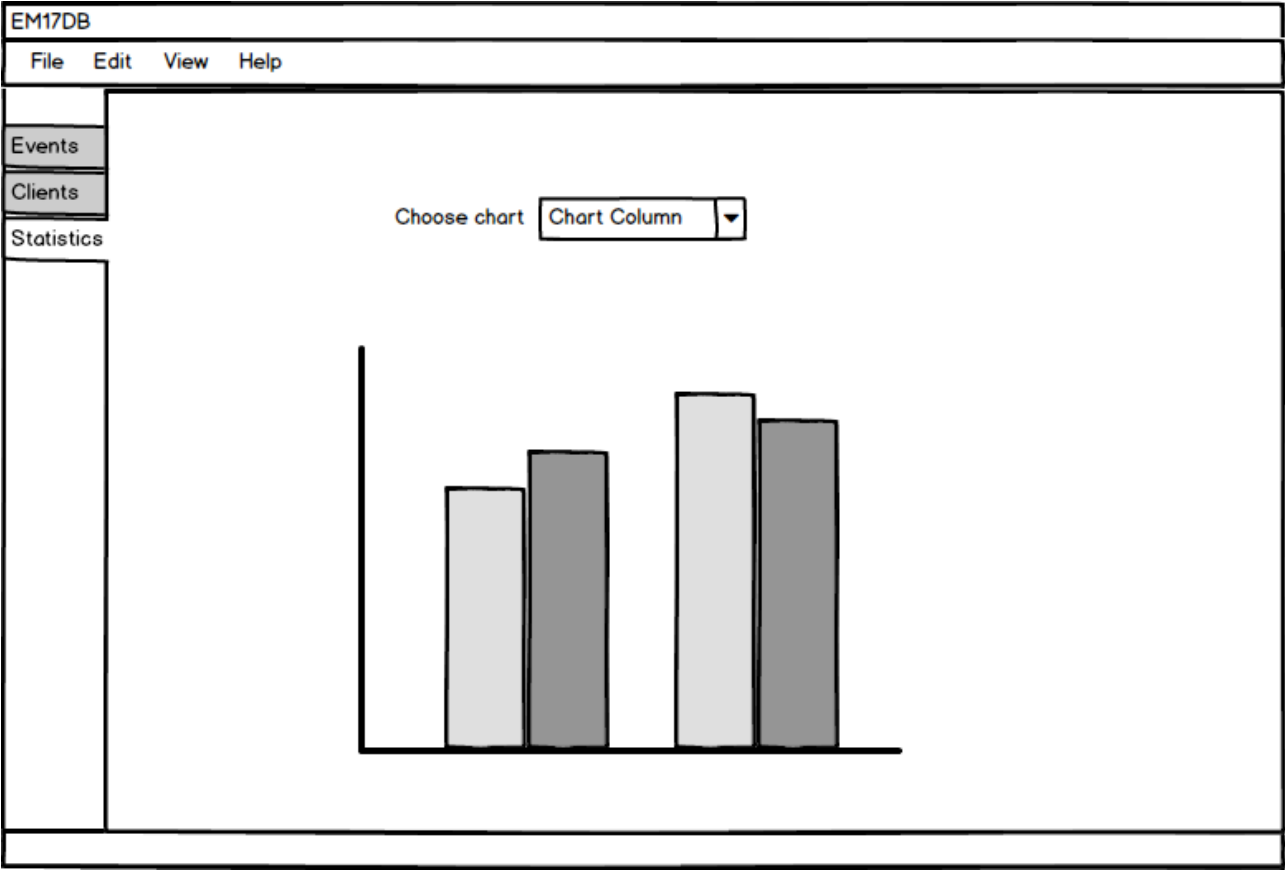
La ricerca avanzata avviene in real time compilando i campi in alto. Modifica e cancellazione avvengono cliccando su una riga

## 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

## *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

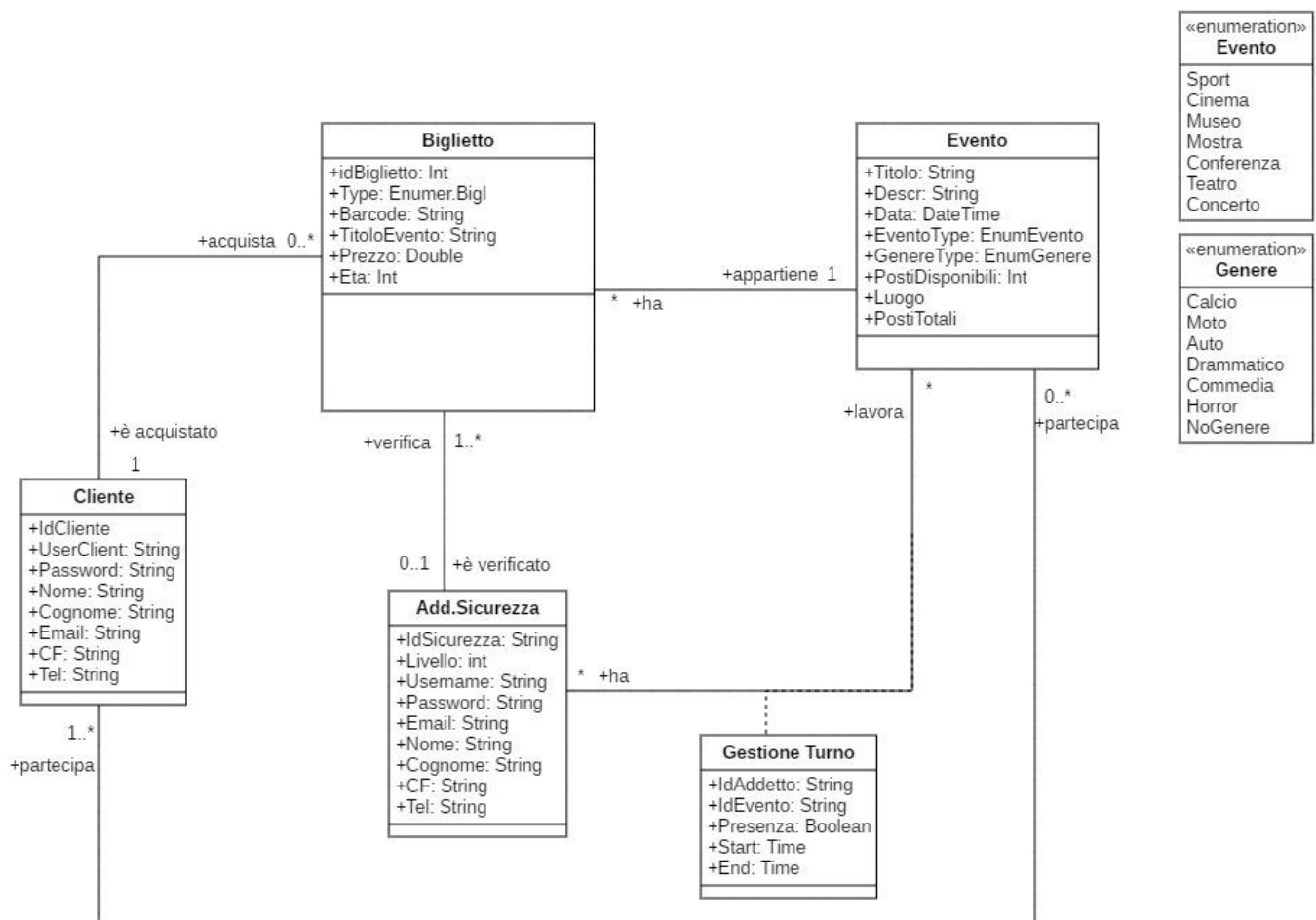
USE CASE	<i>Login</i>		
Goal in Context	Effettua Login		
Scope & Level			
Preconditions	L'addetto alla sicurezza deve possedere le credenziali		
Success End Condition	L'addetto alla sicurezza effettua il login		
Failed End Condition	Il login fallisce		
Primary Actor	Addetto sicurezza		
Trigger	Apre l'app		
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.



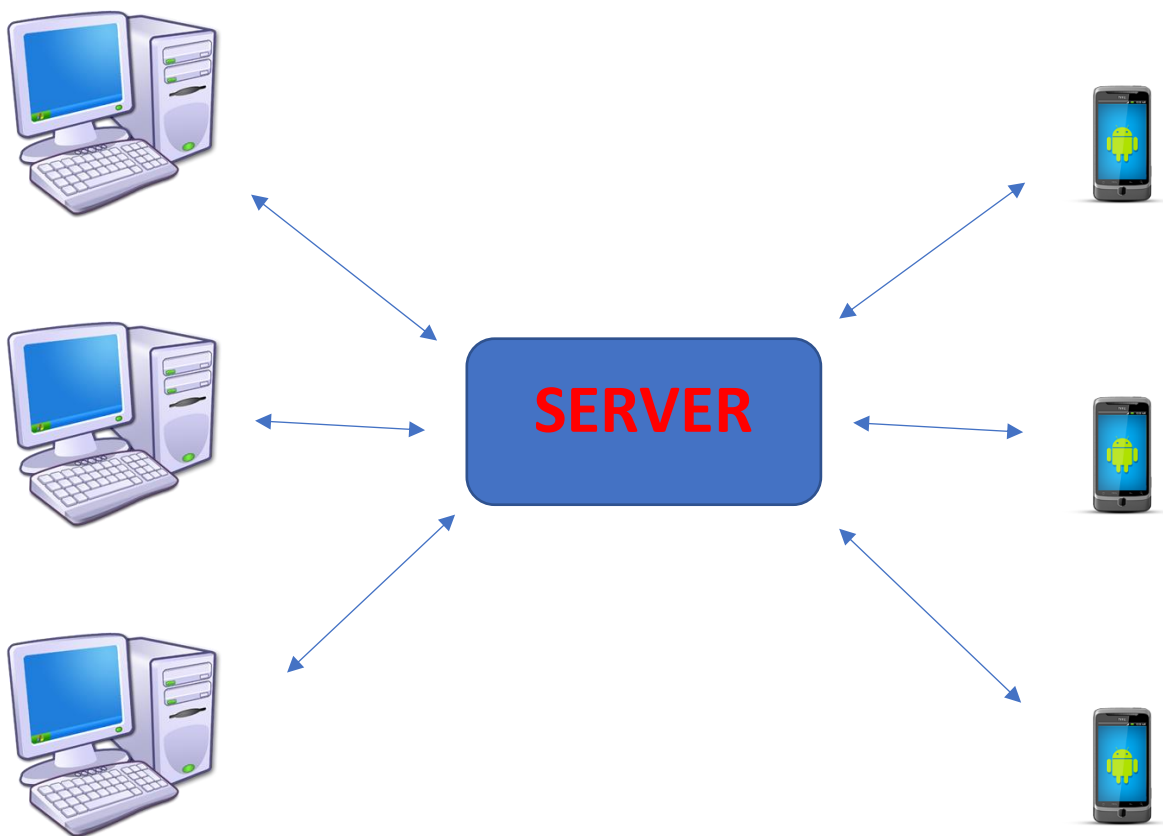
## CAPITOLO 2: Design e Object del Sistema

### 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.

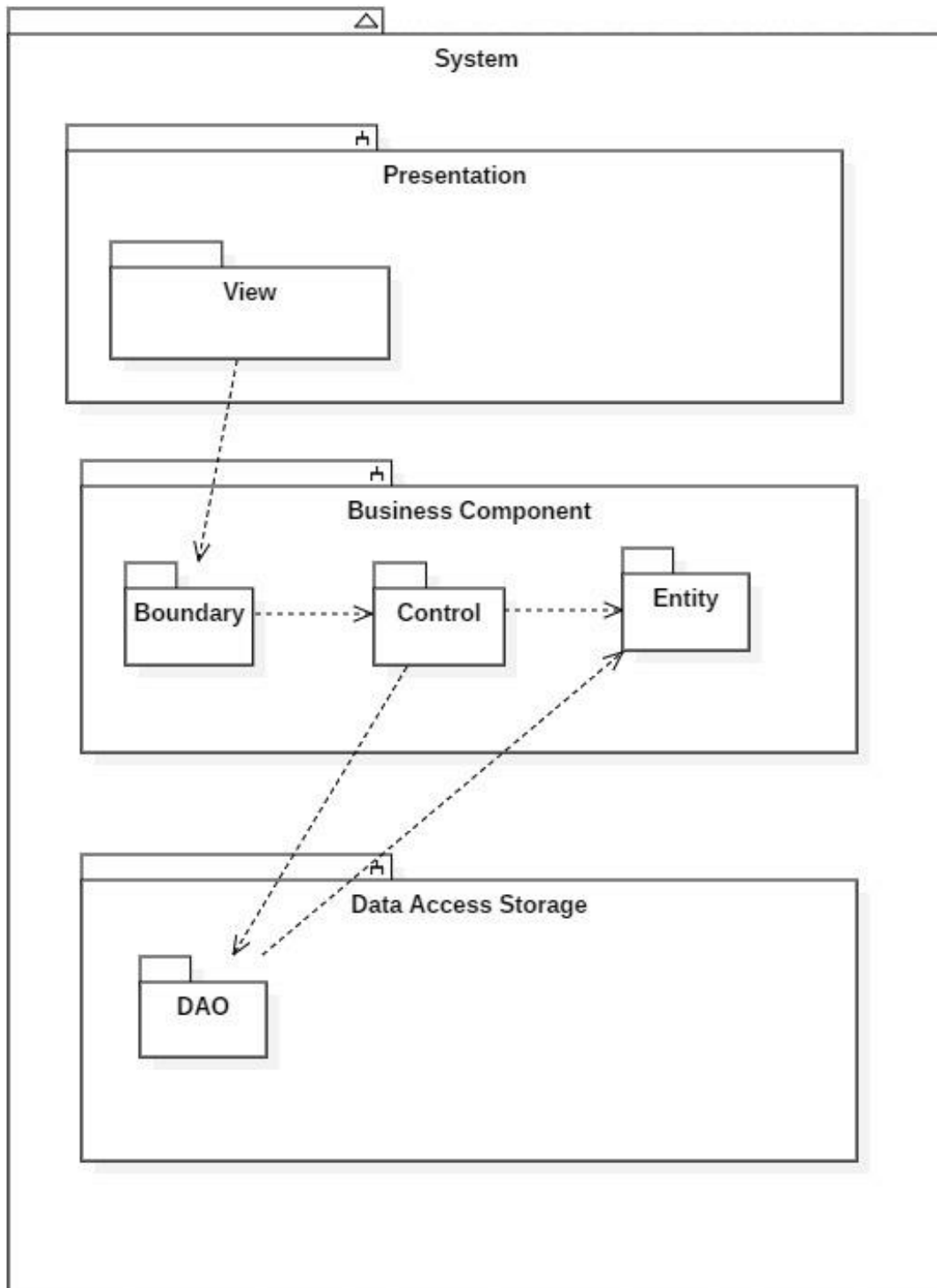
I sottosistemi sono indipendenti tra loro, difatti i due non comunicano mai tra di loro direttamente.

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:



## 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**

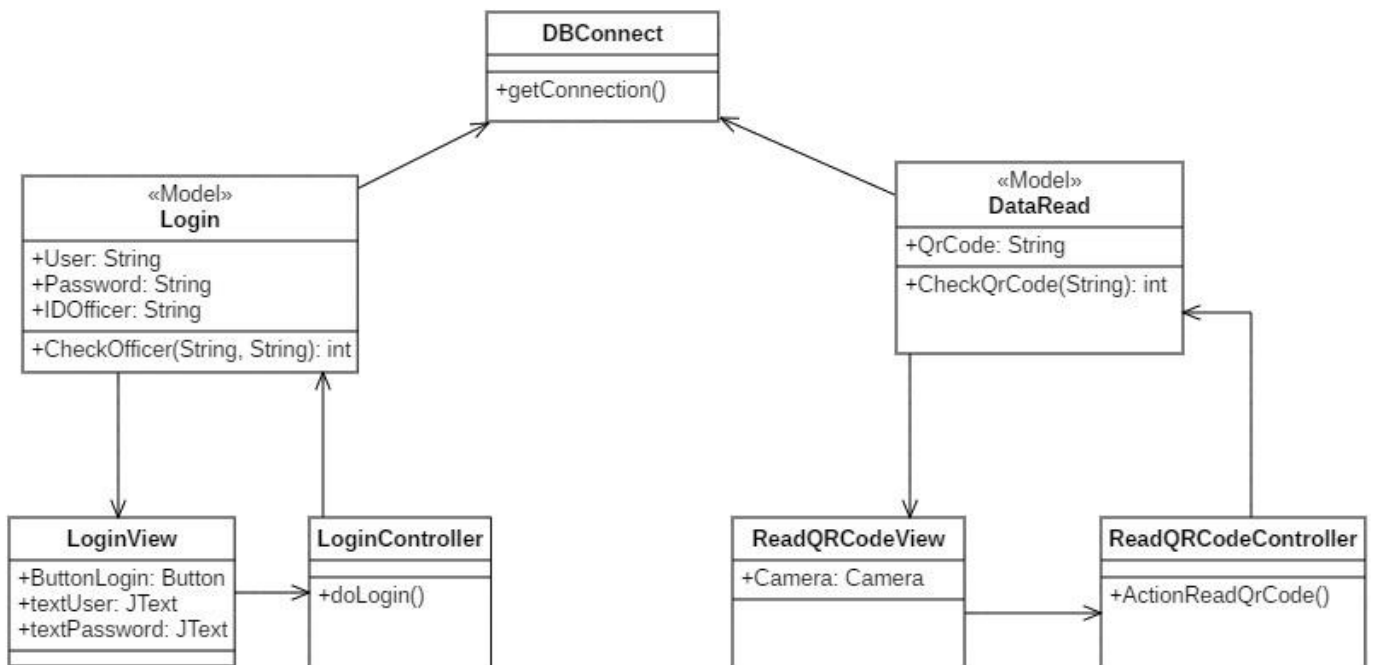
Per quanto riguarda lo sviluppo dell'applicativo desktop sono stati classificati tre tipi di oggetti come Entity, Boundary, Control in quanto è stato utilizzato il pattern architetturale ECB.

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.





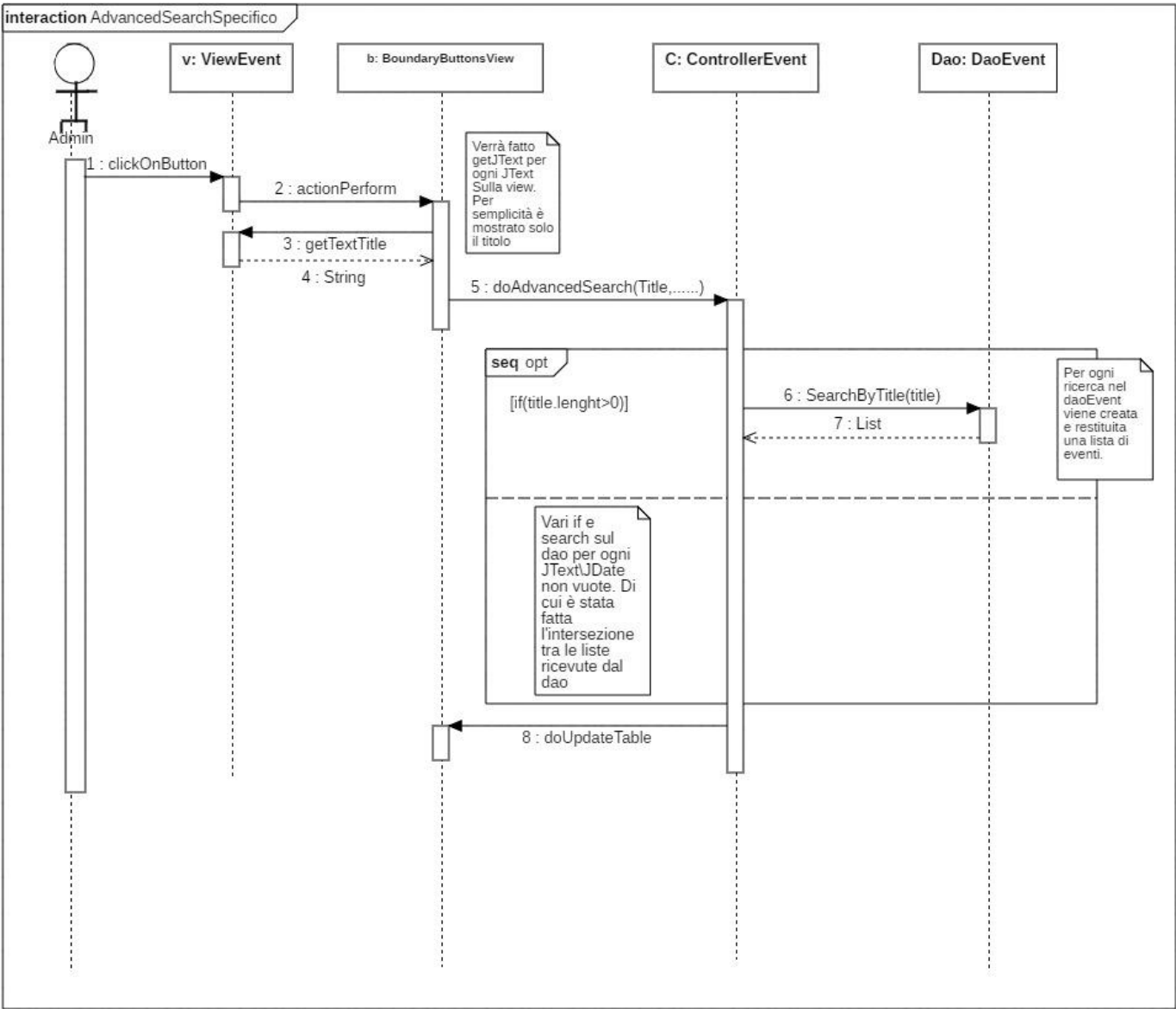
## SEQUENCE DIAGRAM

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.

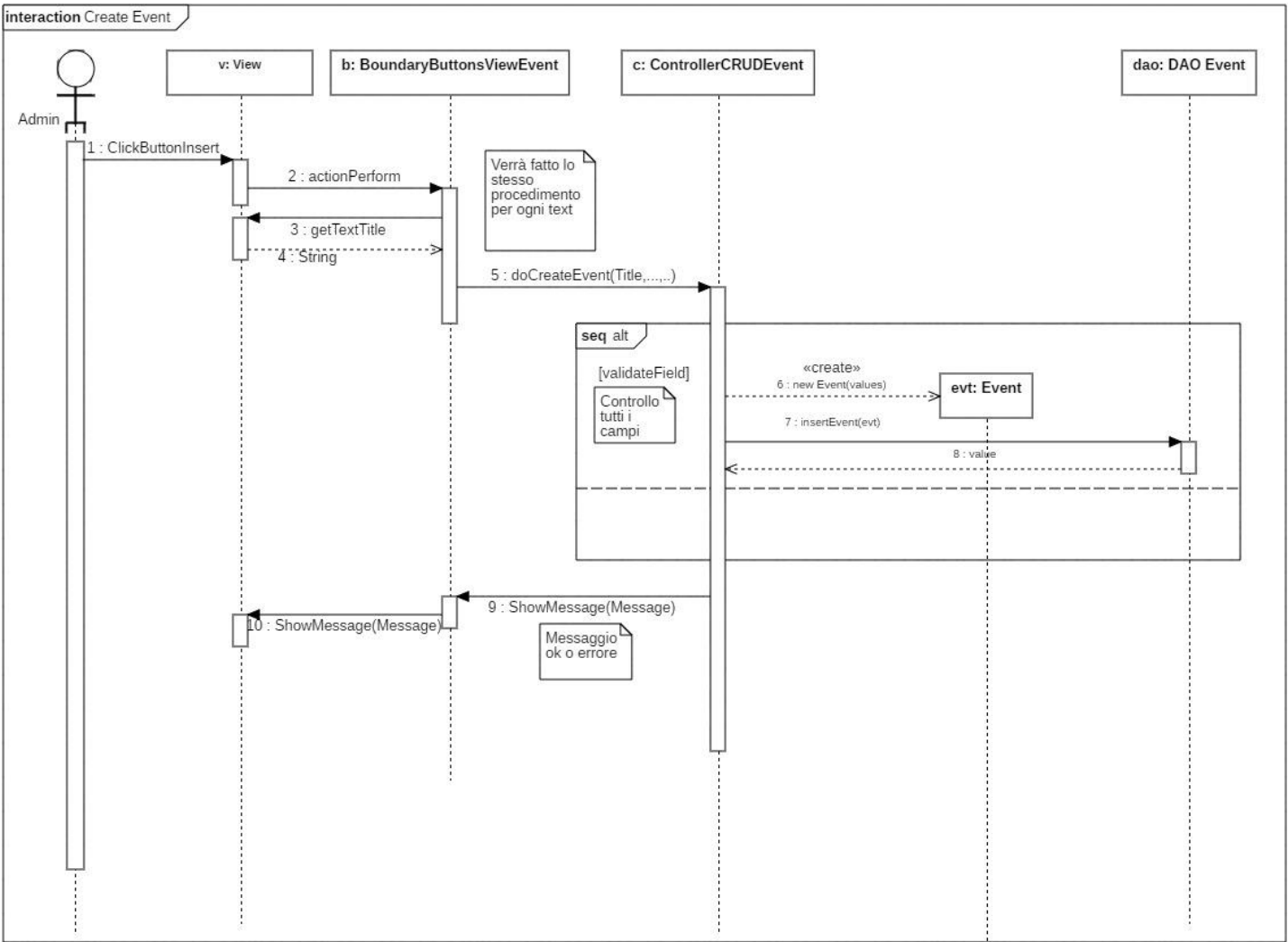
I sequence diagram allegati nelle pagine successive riguardano:

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).

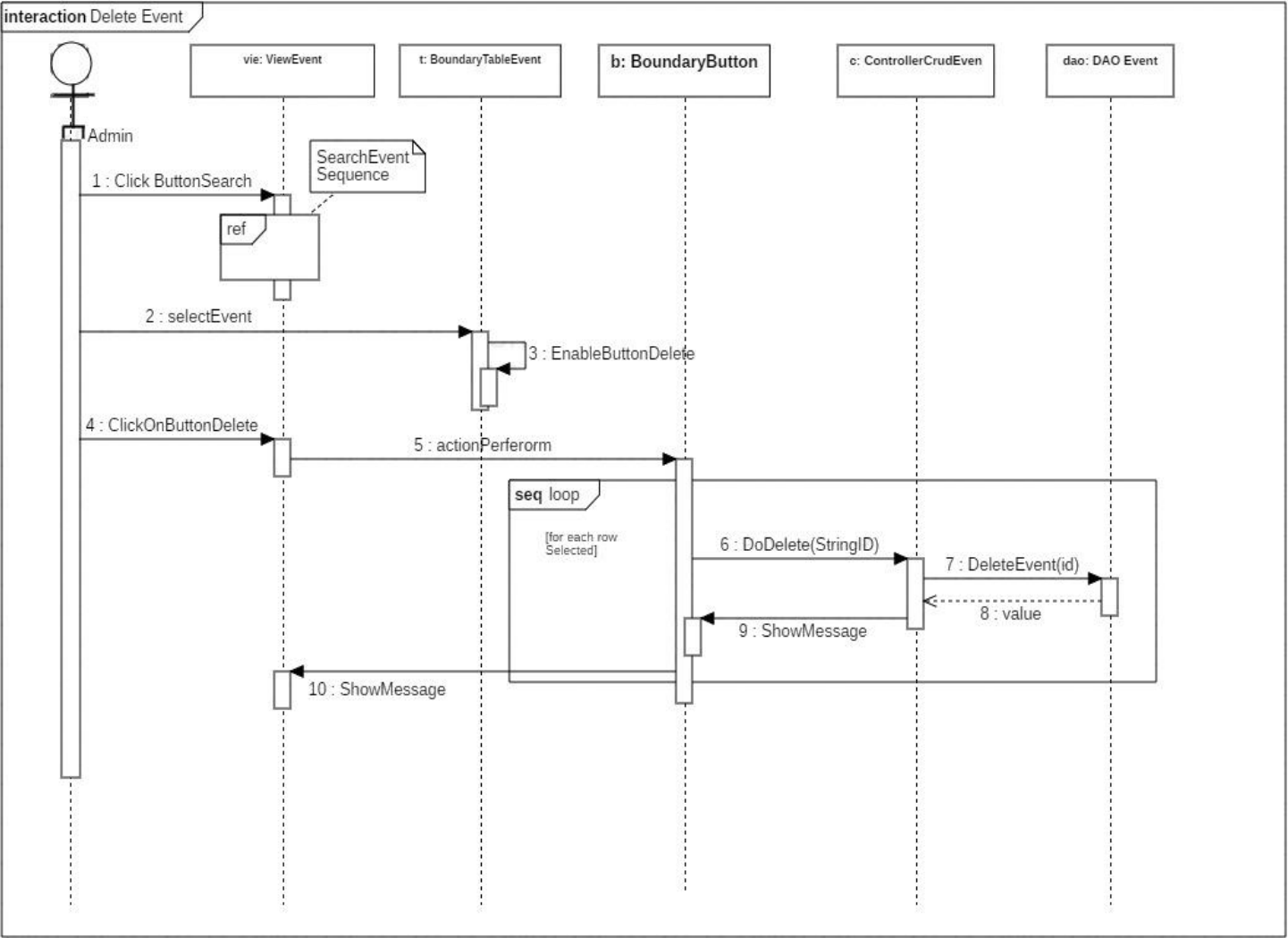
Sequence diagram: Ricerca avanzata



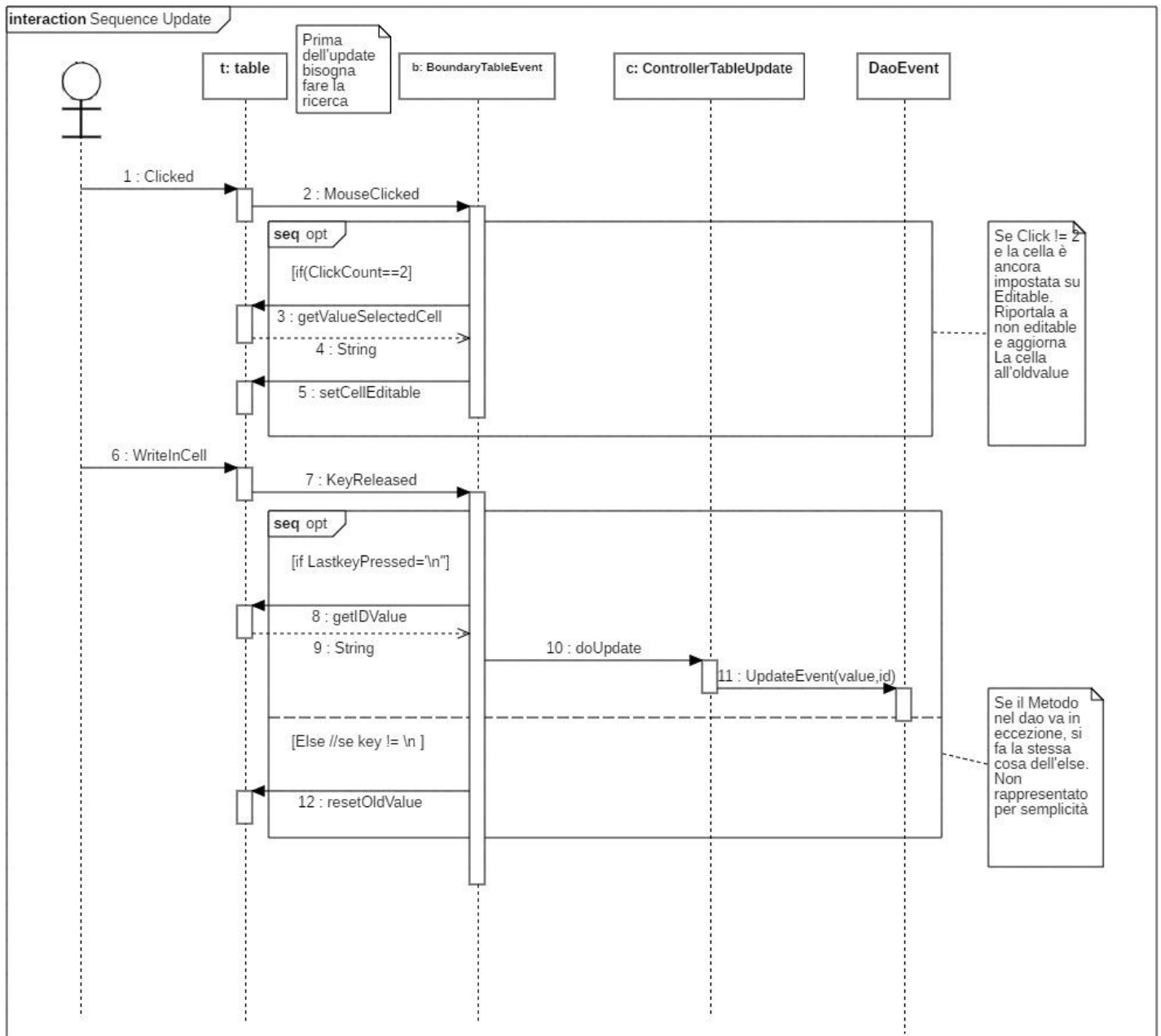
Sequence diagram: Creazione Evento



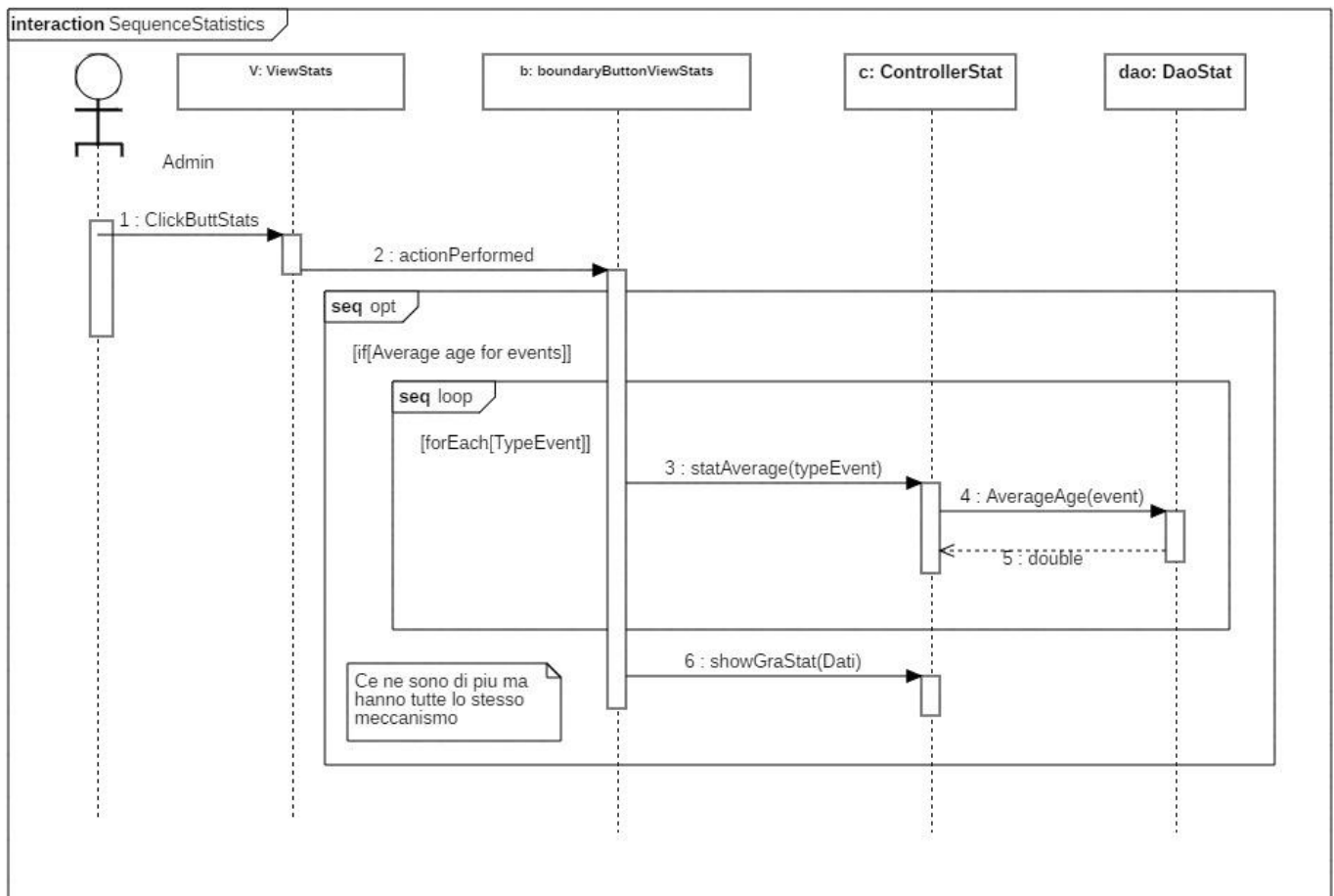
Sequence diagram: Cancellazione evento



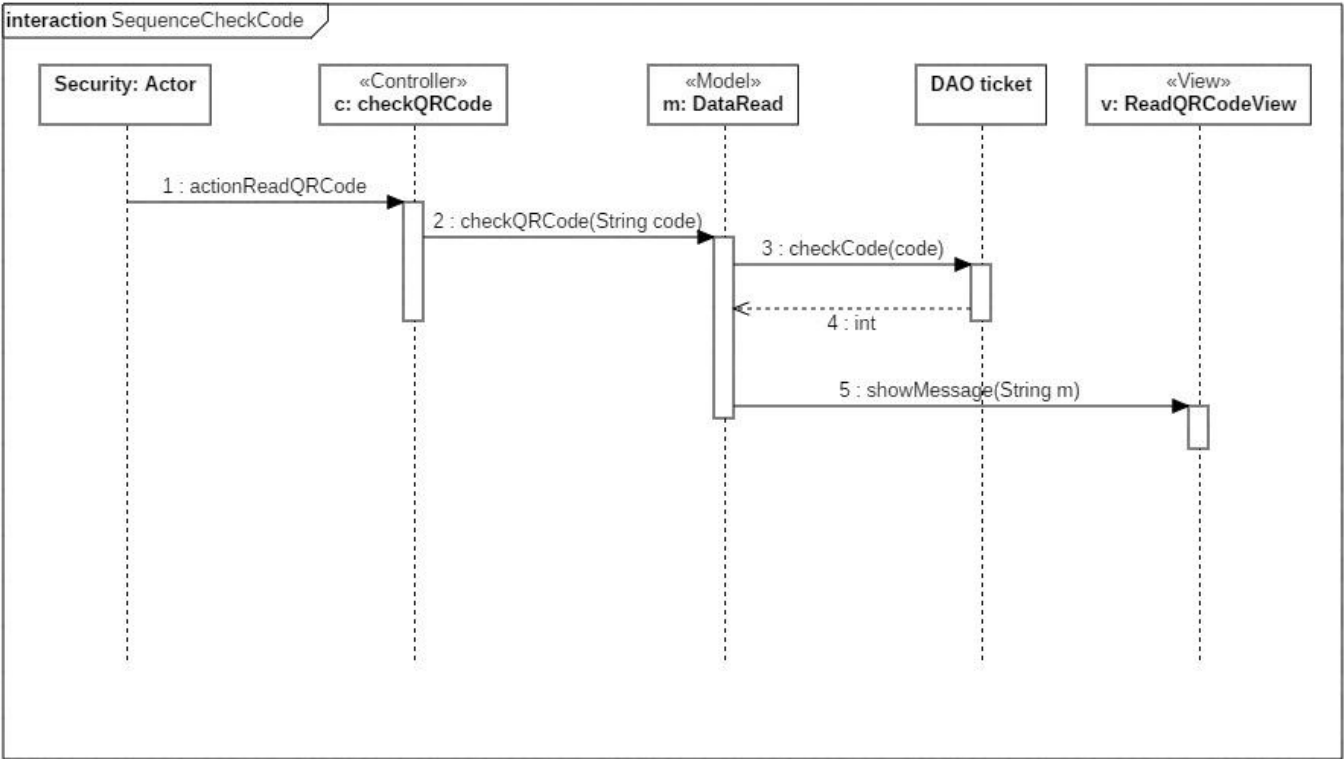
## Sequence diagram: Modifica evento



## 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

### LOGIN TEST:

#### DESCRIZIONE

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

Procederemo con l'individuazione di classi di equivalenza per i valori che l'utente può inserire

#### ESITO ATTESO

L'utente inserisce due valori coerenti e procede con il login.

#### ESITO NON ATTESO

L'utente inserisce valori NON coerenti 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 STRATEGIA SECT:

TEST	USERNAME	PASSWORD	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	Giuliano	ingegneria	CU1 U CP1	Login eseguito	-	-
TC2	Tommaso	progettoing	CU1 U CP1	Login eseguito	-	-
TC3	Marcello	xxx	CU1 U CP1	Username o password errati	-	-
TC4	Vincenzo	-	CU1 U CP2	Errore riempi tutti i campi	-	-
TC5	-	password	CU2 U CP1	Errore riempi tutti i campi	-	-
TC6	-	-	CU2 U CP2	Errore riempi tutti i campi	-	-

## **CREATE EVENT TEST:**

### **DESCRIZIONE**

L'utente può creare un evento compilando le opportune caselle di testo e successivamente cliccando il button CREATE EVENT.

### **ESITO ATTESO**

L'evento viene creato

### **ESITO NON ATTESO**

L'evento non viene creato

### **CLASSI DI EQUIVALEZA "TITOLO"**

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

### **CLASSI DI EQUIVALEZA "DATA"**

- 4 CD1: valore nominale. (Qualsiasi data inserita dall'odierna in poi)
- 5 CD2: nessun valore.

### **CLASSI DI EQUIVALENZA "LUOGO"**

- 6 CL1: valore nominale. (Qualsiasi carattere alfanumerico e/o composizione di essi)
- 7 CL2: caratteri speciali
- 8 CL3: nessun valore

**TEST CASE STRATEGIA WECT:**

TEST	TITOLO	DATA	LUOGO	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	Pino in concerto	15-lug-2018	Napoli	$CT1 \cap CD1 \cap CL1$	Successfull insert	-	
TC2	Pino in concerto	15-lug-2018	-	$CT1 \cap CD1 \cap CL3$	ERROR: Inserisci tutti i campi	-	
TC3	Pino in concerto	-	Napoli	$CT1 \cap CD2 \cap CL1$	ERROR: Inserisci tutti i campi	-	
TC4	-	15-lug-2018	Napoli	$CT3 \cap CD1 \cap CL1$	ERROR: Inserisci tutti i campi	-	
TC5	!^	15-lug-2018	Napoli	$CT2 \cap CD1 \cap CL1$	ERROR: Titolo non valido	-	
TC6	Pino in concerto	15-lug-2018	!!!	$CT2 \cap CD1 \cap CL2$	ERROR: Luogo non valido	-	
TC7	-	-	-	$CT2 \cap CD2 \cap CL3$	ERROR: Inserisci tutti i campi	-	



## **ADVANCED SEARCH EVENT TEST:**

### **DESCRIZIONE**

L'utente può cercare un evento per campi specifici.

### **ESITO ATTESO**

La ricerca restituirà uno o più eventi che saranno mostrati in una tabella .

### **ESITO NON ATTESO**

Problemi con il database.

### **CLASSI DI EQUIVALEZA "TITOLO"**

- 1 CT1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi + caratteri speciali)
- 2 CT2: nessun valore.

### **CLASSI DI EQUIVALEZA "DATA FROM"**

- 3 CDF1: valore nominale. (Qualsiasi data inserita dall'odierna in poi)
- 4 CDF2: nessun valore.

### **CLASSI DI EQUIVALEZA "DATA TO"**

- 5 CDT1: valore nominale. (Qualsiasi data inserita dall'odierna in poi)
- 6 CDT2: nessun valore.

**TEST CASE STRATEGIA SECT:**

TEST	TITOLO	DATA FROM	DATA TO	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	Harry Potter	10-lug-2018	12-lug-2018	$CT1 \cap CDF1 \cap CDT1$	Ricerca Effettuata	Ok	
TC2	Potter	10-lug-2018	-	$CT1 \cap CDF1 \cap CDT2$	INFORMATION: Fill both date fields. The search will not consider the date fields.	OK	
TC3	007	-	10-lug-2018	$CT1 \cap CDF2 \cap CDT1$	INFORMATION: Fill both date fields. The search will not consider the date fields.	OK	
TC4	Non aprire quel software	-	-	$CT1 \cap CDF2 \cap CDT2$	Ricerca solo per titolo	OK	
TC5	-	10-lug-2018	12-lug-2018	$CT2 \cap CDF1 \cap CDT1$	Ricerca gli eventi nel range di date	-	
TC6	-	10-lug-2018	-	$CT2 \cap CDF1 \cap CDT2$	ERRORE: bisogna inserire data to	-	
TC7	-	-	10-lug-2018	$CT2 \cap CDF2 \cap CDT1$	ERRORE: bisogna inserire data from	-	
TC8	-	-	-	$CT2 \cap CDF2 \cap CDT2$	ERRORE: Inserire almeno un parametro	-	

## **UPDATE EVENT TEST:**

### **DESCRIZIONE**

L'utente può modificare un campo specifico di un evento dalla tabella tramite un doppio click sulla cella selezionata e confermando premendo INVIO dalla tastiera (Vedi TEST TABELLA). Tutti i campi modificabili lo sono in modo analogo, ma bisogna porre attenzione sul campo TYPE in quanto se cambio il tipo di evento allora il genere (KIND) dell'evento verrà settato a 'null' in quanto strettamente correlato. Procederemo con l'individuazione delle classi di equivalenza per TITOLO, TIPO, GENERE. Tutti gli altri casi sono equivalenti al TITOLO.

### **ESITO ATTESO**

Il dato in questione verrà modificato.

### **ESITO NON ATTESO**

Il dato in questione non verrà modificato.

### **CLASSI DI EQUIVALEZA "TITOLO"**

- 1 CT1: valori nominali. (Qualsiasi carattere alfanumerico e/o composizione di essi + caratteri speciali)
- 2 CT2: nessun valore.

### **CLASSI DI EQUIVALEZA "TIPO"**

- 3 CP1: valori dell'enumerazione: SPORT, CINEMA, THEATER, CONCERT, OTHER
- 4 CP2: nessun valore
- 5 CP3: qualsiasi altra stringa diversa dalla classe di equivalenza CP1

### **CLASSI DI EQUIVALEZA "GENERE"**

- 6 CG1: valore nominale: il genere dipende dal tipo dell'evento o nel caso questo cambi il genere può essere null.
- 7 CG2: nessun valore.
- 8 CG3: qualsiasi altra stringa diversa dalla classe di equivalenza CG1

**TEST CASE STRATEGIA WECT: il campo modificato sarà definito dal colore ROSSO**

TEST	TITOLO	TIPO	GENERE	CE	RISULTATO ATTESO	RISULTATO TEST	BUG FIXING
TC1	Harry Potter	Sport	Tennis	$CT1 \cap CP1 \cap CG1$	Il campo titolo verrà modificato	-	
TC2	Pino in concerto	Sport	Horror	$CT1 \cap CP1 \cap CG1$	ERRORE: il genere modificato non è inerente al tipo [in questo caso: FOOTBALL, BASKET, VOLLEYBALL, TENNIS, OTHER]	-	
TC3	-----	Sport	Tennis	$CT2 \cap CP1 \cap CG1$	ERRORE: titolo non valido	-	
TC4	Software	Theater	Tennis	$CT1 \cap CP1 \cap CG1$	Il genere viene settato a OTHER	-	
TC5	Software	----	Tennis	$CT1 \cap CP2 \cap CG1$	Tipo inserito non valido: deve essere uno tra SPORT, CINEMA, THEATER, CONCERT, OTHER	-	
TC6	Software	ciao	Tennis	$CT1 \cap CP3 \cap CG1$	Tipo inserito non valido: deve essere uno tra SPORT, CINEMA, THEATER, CONCERT, OTHER	-	

TC7	Software	Sport	-----	$CT1 \cap CP1 \cap CG2$	ERRORE: il genere modificato non è inerente al tipo [in questo caso: FOOTBALL, BASKET, VOLLEYBALL, TENNIS, OTHER]	-	
TC8	Partita	Sport	ciao	$CT1 \cap CP1 \cap CG3$	ERRORE: il genere modificato non è inerente al tipo [in questo caso: FOOTBALL, BASKET, VOLLEYBALL, TENNIS, OTHER]	-	
TC9	Software	Cinema	ciao	$CT1 \cap CP1 \cap CG3$	ERRORE: il genere modificato non è inerente al tipo [in questo caso: HORROR, COMEDY, DRAMATIC, OTHER]	-	

### 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.