|  |  |
| --- | --- |
|  | Laurea magistrale in ingegneria informatica  Corso di ingegneria del software  **Studenti:**  **Fabrizio Giorgione 399000116**  **Alfredo Nazzaro 399000109**  **Professore:**  **M.Di Penta** |

Sommario

[1.0 Analisi applicazione Counter Shock 3](#_Toc461787995)

[1.1 Problem statement 3](#_Toc461787996)

[1.2 Requisiti funzionali 3](#_Toc461787997)

[1.3 Requisiti non funzionali 3](#_Toc461787998)

[1.4 Attori con relative funzioni 4](#_Toc461787999)

[1.5 User stories 4](#_Toc461788000)

[1.6 Principali casi d’uso 5](#_Toc461788001)

[1.7 Tabella del ranking 5](#_Toc461788002)

[2.0 Design architetturale 6](#_Toc461788003)

[2.1 Diagramma dei casi d’uso 6](#_Toc461788004)

[2.2 Use case diagram 7](#_Toc461788005)

[2.2.1 Login con account google: Activity diagram 7](#_Toc461788006)

[2.2.2 Login con account google: Sequence diagram 8](#_Toc461788007)

[2.2.3 Inserimento percorso tramite google maps: Activity diagram 8](#_Toc461788008)

[2.2.4 Inserimento percorso tramite google maps: Sequence diagram 9](#_Toc461788009)

[2.2.5 Inserimento commenti: activity diagram 9](#_Toc461788010)

[2.2.6 Modifica/eliminazione commenti: activity diagram 10](#_Toc461788011)

[2.2.7 Visualizzazione numero di fossi del percorso: activity diagram 11](#_Toc461788012)

[2.2.8 Visualizzazione numero di fossi del percorso: sequence diagram 12](#_Toc461788013)

[3.0 Principali design pattern 12](#_Toc461788014)

[4.0 Testing 12](#_Toc461788015)

# 1.0 Analisi applicazione Counter Shock

## 1.1 Problem statement

L’applicazione si prefigge come scopo quello di creare un’applicazione che definito un percorso tramite un’interfaccia user-friendly sarà capace di monitore la situazione del manto stradale e alla fine del percorso fornirà il numero di fossi l’utente ha preso. Vi sarà, inoltre, la possibilità di visualizzare un elenco di percorsi inseriti da altri utenti attraverso i quali l’utente potrà visionare la condizione di altre strade. Vi sarà anche la possibilità di inserire un commento, modificarlo e cancellarlo.

## 1.2 Requisiti funzionali

**- Login utente**

L’utente potrà loggarsi dopo aver effettuato la registrazione al sistema tramite i relativi layout.

**- Aggiunta commenti**

L’utente loggato visualizzerà una lista di percorsi e potrà accedere ai dettagli dei singoli. Una volta aperto potrà inserire un commento sulla base dei dati a disposizione, ogni commento avrà un identificativo utente, ciò consentirà di eliminare o modificare il commento da lui inserito (presente dalla milestone 0.2)

**- Inserimento del percorso desiderato**

L’utente loggato potrà accedere ad un menù che permetterà di aprire un layout nel quale saranno presenti tutte le informazioni necessarie per inserire il percorso desiderato. Dal milestone 0.2 durante il percorso l’app conterà il numero di fossi presi e alla fine potrà inserire anche un voto sul manto stradale. Una volta inserito il voto e un commento l’utente potrà salvare.

## 1.3 Requisiti non funzionali

* **Tolleranza a malfunzionamenti**

Il sistema dovrà garantire un funzionamento minimale in caso di malfunzionamenti, ovvero dovrà consentire la registrazione e la visualizzazione dei percorsi agli utenti.

## 1.4 Attori con relative funzioni

- Utente anonimo:

* visualizza percorsi e feedback

- Utente registrato:

* Accesso tramite login
* Inserimento percorso desiderato
* Possibilità di inserire commenti e dare un voto al manto stradale

## 1.5 User stories

1) Come utente anonimo, posso visualizzare i percorsi effettuati da altri utenti e visualizzare eventuali commenti. Posso effettuare l’accesso con le mie credenziali oppure, effettuare la registrazione.

**Punteggio: 7**

2)Come utente registrato, posso inserire un percorso desiderato oppure visualizzare un elenco di percorsi. Posso inserire dei commenti tramite l’apposita finestra presente all’interno di ogni percorso salvato.

**Punteggio: 5**

3) Come utente registrato, ho la possibilità di salvare il percorso da me effettuato; questi conterrà il numero di fossi contati dall’accelerometro e il numero di km effettuati.

**Punteggio: 5**

4) Come utente registrato, una volta inserito il percorso potrò visualizzare il numero di fossi che l’utente prenderà durante il tragitto. Tale implementazione sarà effettuata tramite l’uso di un accelerometro

**Punteggio: 10**

N.B. Il punteggio varia in base alle difficoltà implementative

## 1.6 Principali casi d’uso

Definizione dei principali casi d’uso con relativa descrizione:

1) Login

2) Inserimento percorso

3) Inserimento di commenti

4) Visualizzazione informazioni del percorso

## 1.7 Tabella del ranking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Casi d’uso | a. | b. | c. | d. | Totale |
| Login | 5 | 5 | 5 | 3 | 18 |
| Inserimento percorso | 5 | 5 | 4 | 3 | 17 |
| Inserimento commenti | 1 | 3 | 2 | 2 | 8 |
| Visualizzazione informazione del percorso | 5 | 4 | 2 | 2 | 13 |

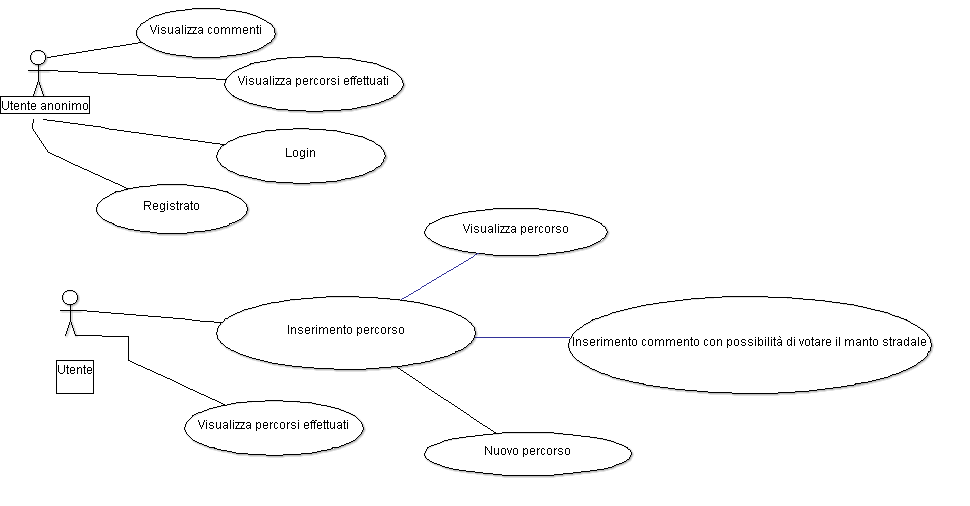
Legenda:

* a. Importanza architetturale
* b. Maggior numero di funzionalità base
* c. Alto rischio
* d. Difficoltà di realizzazione

I valori sono compresi da 1 a 5

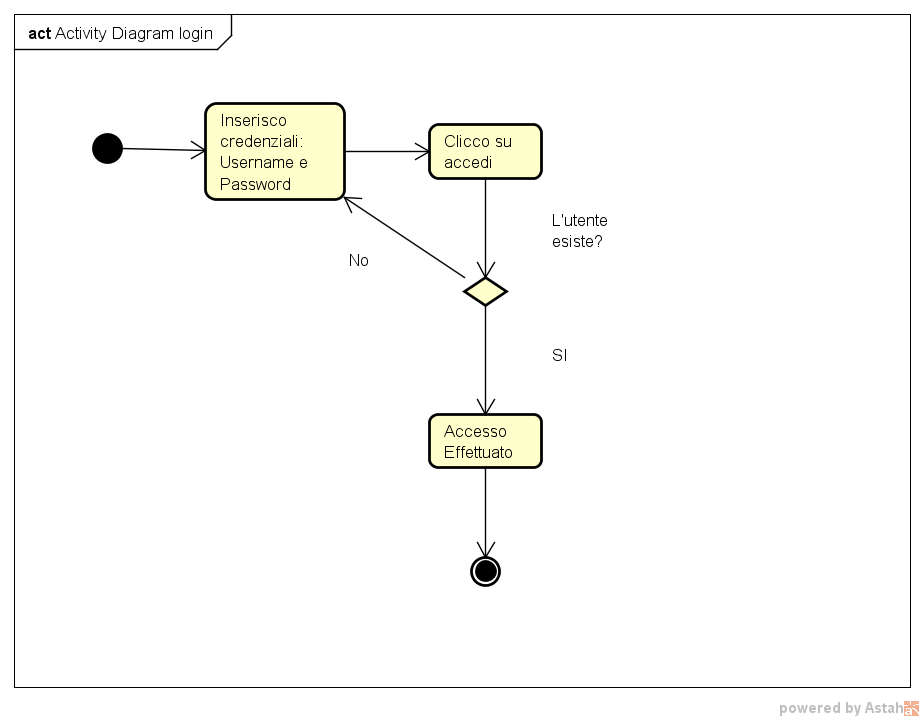
# 2.0 Design architetturale

## 2.1 Diagramma dei casi d’uso

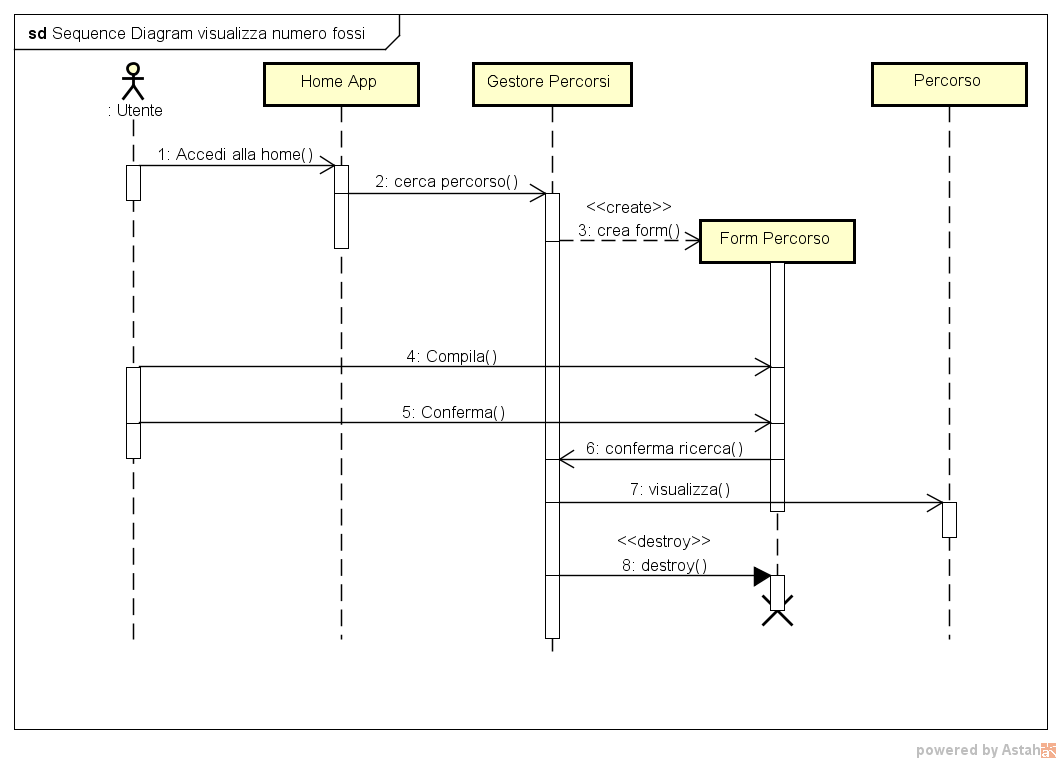


## 2.2 Use case diagram

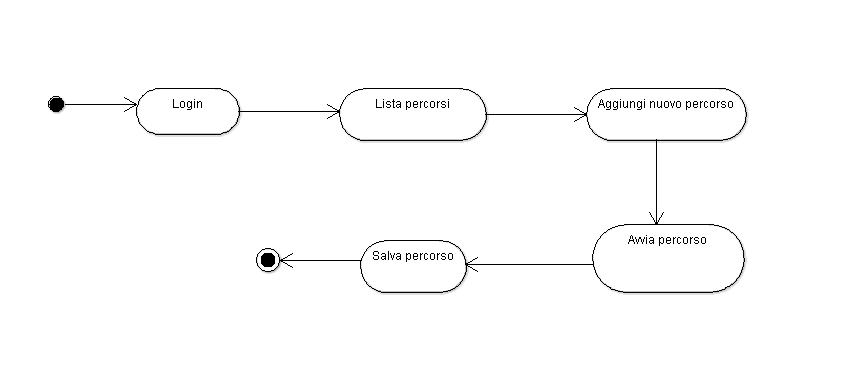
### 2.2.1 Login: Activity diagram



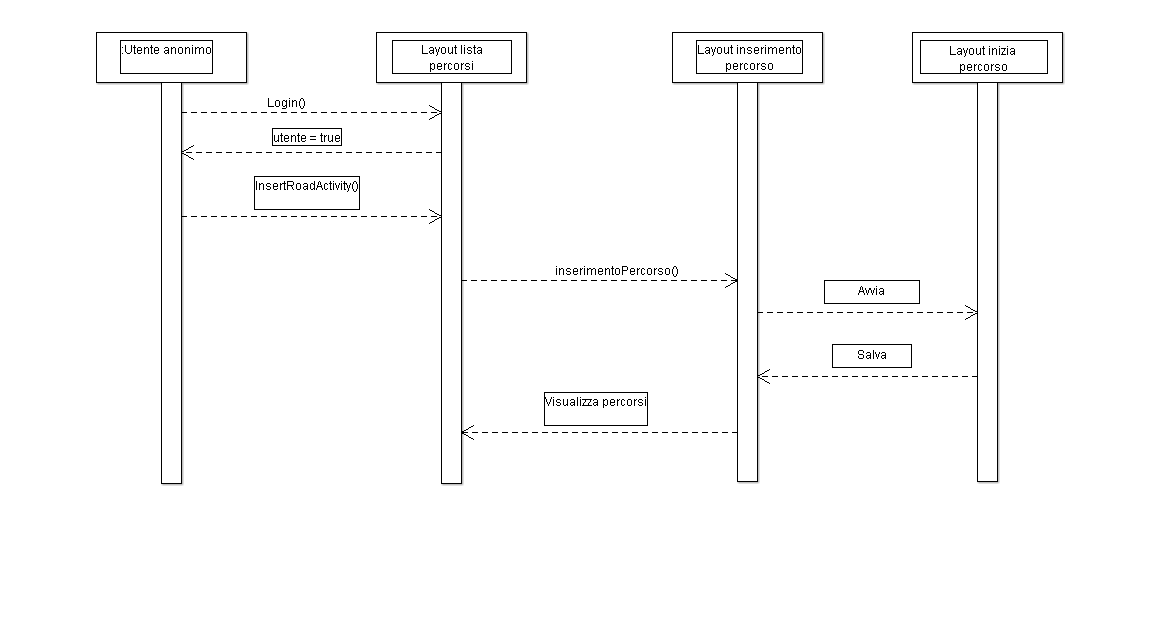
### 2.2.2 Login: Sequence diagram



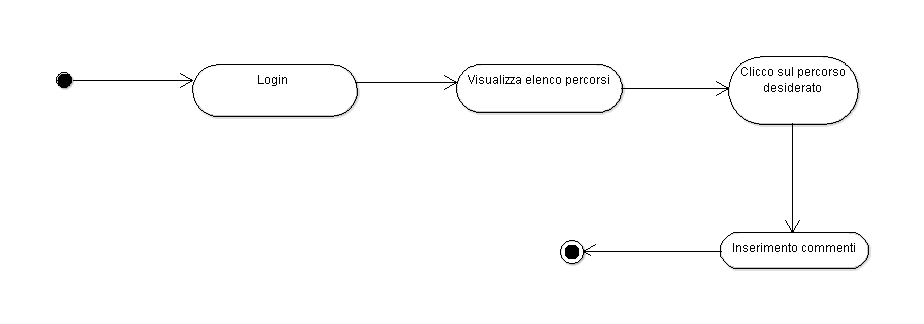
### 2.2.3 Inserimento percorso: Activity diagram



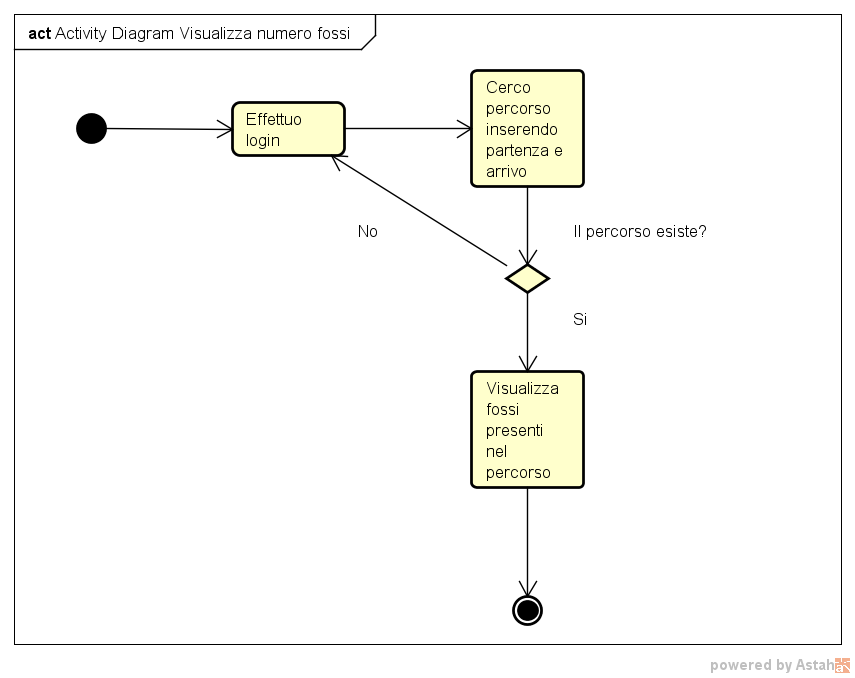
### 2.2.4 Inserimento percorso: Sequence diagram



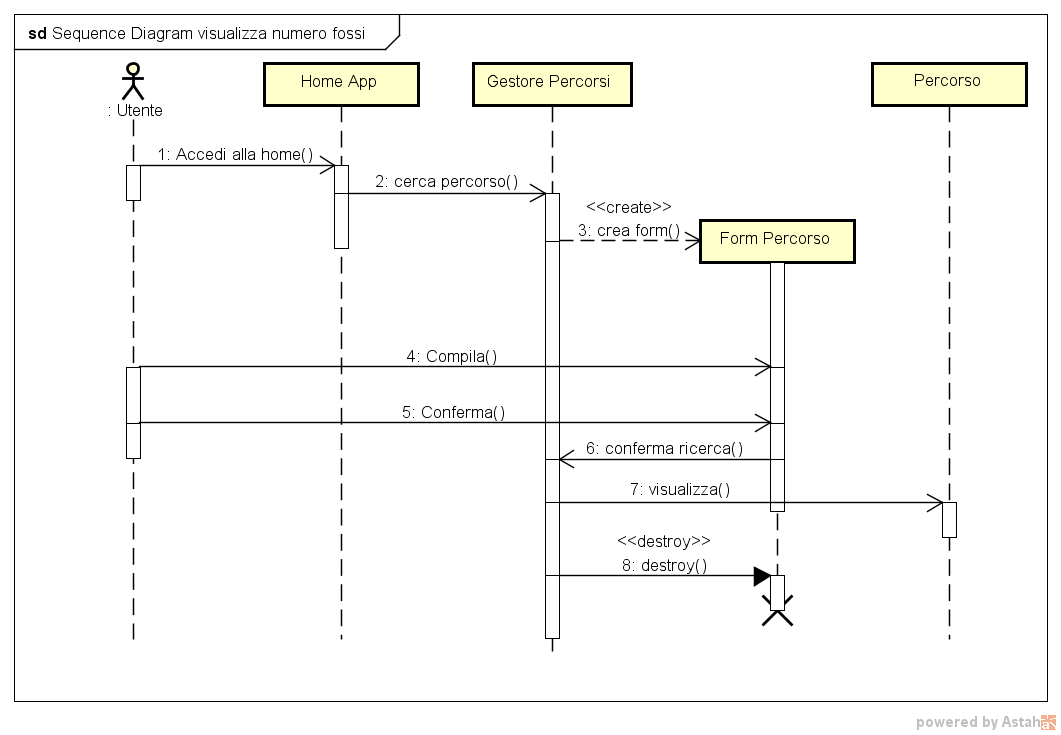
### 2.2.5 Inserimento commenti: activity diagram



### 2.2.7 Visualizzazione informazioni del percorso: activity diagram



### 2.2.8 Visualizzazione numero di fossi del percorso: sequence diagram



# 3.0 Principali design pattern

I design pattern utilizzati appartengono alla categorie creazionale e strutturale

Nell’ordine sono: Singleton e Facade

**SINGLETON**

Le classi che gestiscono il login e il mantenimento della sessione sono implementate secondo il pattern singleton. Il pattern singleton si assicura che una determinata classe abbia una sola istanza e fornisca per essa un solo punto di accesso. L’utilizzo di questo pattern ha come conseguenza, nel nostro progetto, la gestione di un accesso controllato alla singola istanza.

**FACADE**

E’ stato utilizzato il facade per fornire un’interfaccia unificata ad un insieme di interfacce del sottosistema allo scopo di renderlo più semplice da utilizzare. Come conseguenza , l’utente accede in maniera agevolata alle componenti del sottosistema , riducendo l’accoppiamento del sistema.

# 4.0 Testing